

Security

Oct 13, 2015

Security

The following topics cover configuration and installation information for NetScaler security features. Most of these features are policy based.

Authentication, Authorization, Auditing (AAA)	Keeps unauthorized users out of the network, denies users access to tasks for which they are not authorized, and tracks the resources used during user sessions.
Application Firewall	Prevents security breaches, data loss, and possible unauthorized modifications to web sites that access sensitive business or customer information.
Content Filtering	Blocks inappropriate HTML requests, preventing the requests from reaching the Web servers.
HTTP Denial-of-Service Protection	Prevents hackers from attacking your Web site with large numbers of HTTP requests.
Priority Queuing	Detects high-priority connections and allows those connections to proceed ahead of other connections, guaranteeing unimpeded access to those users.
SureConnect	Serves all incoming connections with either the requested content or a custom Web page that displays information about a delay in the request being serviced.
Surge Protection	Detects any rapid rise in connection attempts and adjusts the rate at which connections are allowed to proceed to the server, preventing server overload.

AAA Application Traffic

Many companies restrict web site access to valid users only, and control the level of access permitted to each user. The AAA feature allows a site administrator to manage access controls with the NetScaler appliance instead of managing these controls separately for each application. Doing authentication on the appliance also permits sharing this information across all web sites within the same domain that are protected by the appliance.

The AAA feature supports authentication, authorization, and auditing for all application traffic. To use AAA, you must configure authentication virtual servers to handle the authentication process and traffic management virtual servers to handle the traffic to web applications that require authentication. You also configure your DNS to assign FQDNs to each virtual server. After configuring the virtual servers, you configure a user account for each user that will authenticate via the NetScaler appliance, and optionally you create groups and assign user accounts to groups. After creating user accounts and groups, you configure policies that tell the appliance how to authenticate users, which resources to allow users to access, and how to log user sessions. To put the policies into effect, you bind each policy globally, to a specific virtual server, or to the appropriate user accounts or groups. After configuring your policies, you customize user sessions by configuring session settings and binding your session policies to the traffic management virtual server. Finally, if your intranet uses client certs, you set up the client certificate configuration.

Before configuring AAA, you should be familiar with and understand how to configure load balancing, content switching, and SSL on the NetScaler appliance.

How AAA Works

AAA provides security for a distributed Internet environment by allowing any client with the proper credentials to connect securely to protected application servers from anywhere on the Internet. This feature incorporates the three security features of authentication, authorization, and auditing. Authentication enables the NetScaler ADC to verify the client's credentials, either locally or with a third-party authentication server, and allow only approved users to access protected servers. Authorization enables the ADC to verify which content on a protected server it should allow each user to access. Auditing enables the ADC to keep a record of each user's activity on a protected server.

To understand how AAA works in a distributed environment, consider an organization with an intranet that its employees access in the office, at home, and when traveling. The content on the intranet is confidential and requires secure access. Any user who wants to access the intranet must have a valid user name and password. To meet these requirements, the ADC does the following:

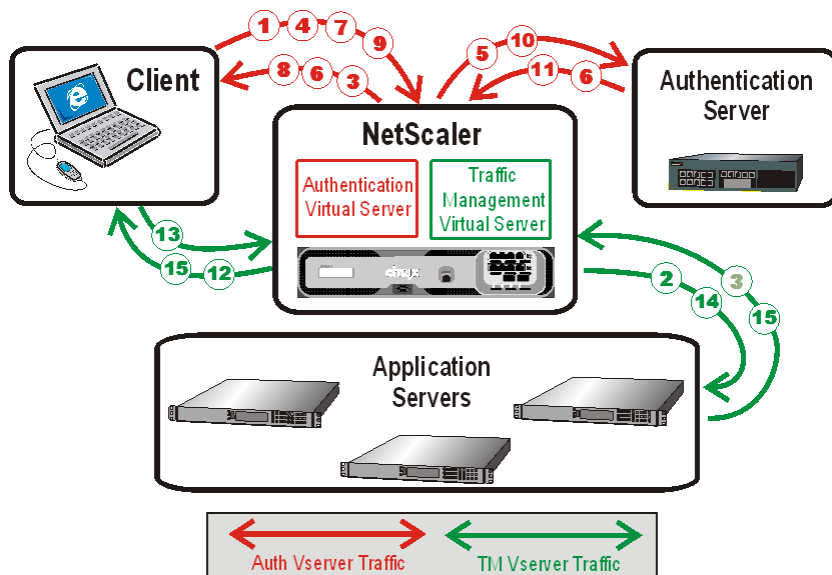
- Redirects the user to the login page if the user accesses the intranet without having logged in.
- Collects the user's credentials, delivers them to the authentication server, and caches them in a directory that is accessible through LDAP.
- Verifies that the user is authorized to access specific intranet content before delivering the user's request to the application server.
- Maintains a session timeout after which users must authenticate again to regain access to the intranet. (You can configure the timeout.)
- Logs the user accesses, including invalid login attempts, in an audit log.

Authentication requires that several entities—the client, the NetScaler appliance, the external authentication server if one is used, and the application server—respond to each other when prompted by performing a complex series of tasks in the correct order. If you are using an external authentication server, this process can be broken down into the following fifteen steps.

- The client sends a GET request for a URL on the application server.
- The NetScaler appliance's traffic management virtual server redirects the request to the application server.
- The application server determines that the client has not been authenticated, and therefore sends an HTTP 200 OK response via the TM vserver to the client. The response contains a hidden script that causes the client to issue a POST request for /cgi/tm.
- The client sends a POST request for /cgi/tm.
- The NetScaler appliance's authentication virtual server redirects the request to the authentication server.
- The authentication server creates an authentication session, sets and caches a cookie that consists of the initial URL and the domain of the traffic management virtual server, and then sends an HTTP 302 response via the authentication virtual server, redirecting the client to /vpn/index.html.
- The client sends a GET request for /vpn/index.html.
- The authentication virtual server redirects the client to the authentication server login page.
- The client sends a GET request for the login page, enters credentials, and then sends a POST request with the credentials back to the login page.
- The authentication virtual server redirects the POST request to the authentication server.
- If the credentials are correct, the authentication server tells the authentication virtual server to log the client in and redirect the client to the URL that was in the initial GET request.
- The authentication virtual server logs the client in and sends an HTTP 302 response that redirects the client to the initially requested URL.
- The client sends a GET request for their initial URL.
- The traffic management virtual server redirects the GET request to the application server.
- The application server responds via the traffic management virtual server with the initial URL.

If you use local authentication, the process is similar, but the authentication virtual server handles all authentication tasks instead of forwarding connections to an external authentication server. The following figure illustrates the authentication process.

Figure 1. Authentication Process Traffic Flow



When an authenticated client requests a resource, the ADC, before sending the request to the application server, checks the user and group policies associated with the client account, to verify that the client is authorized to access that resource. The ADC handles all authorization on protected application servers. You do not need to do any special configuration of your protected application servers.

AAA-TM handles password changes for users by using the protocol-specific method for the authentication server. For most protocols, neither the user nor the administrator needs to do anything different than they would without AAA-TM. Even when an LDAP authentication server is in use, and that server is part of a distributed network of LDAP servers with a single designated domain administration server, password changes are usually handled seamlessly. When an authenticated client of an LDAP server changes his or her password, the client sends a credential modify request to AAA-TM, which forwards it to the LDAP server. If the user's LDAP server is also the domain administration server, that server responds appropriately and AAA-TM then performs the requested password change. Otherwise, the LDAP server sends AAA-TM an LDAP_REFERRAL response to the domain administration server. AAA-TM follows the referral to the indicated domain administration server, authenticates to that server, and performs the password change on that server.

When configuring AAA-TM with an LDAP authentication server, the system administrator must keep the following conditions and limitations in mind:

- AAA-TM assumes that the domain administration server in the referral accepts the same bind credentials as the original server.
- AAA-TM only follows LDAP referrals for password change operations. In other cases AAA-TM refuses to follow the referral.
- AAA-TM only follows one level of LDAP referrals. If the second LDAP server also returns a referral, AAA-TM refuses to follow the second referral.

The ADC supports auditing of all states and status information, so you can see the details of what each user did while logged on, in chronological order. To provide this information, the appliance logs each event, as it occurs, either to a designated audit log file on the appliance or to a syslog server. Auditing requires configuring the appliance and any syslog server that you use.

Enabling AAA

To use the AAA - Application Traffic feature, you must enable it. You can configure AAA entities—such as the authentication and traffic management virtual servers—before you enable the AAA feature, but the entities will not function until the feature is enabled.

To enable AAA by using the command line interface

At the command prompt, type the following commands to enable AAA and verify the configuration:

- enable ns feature AAA
- show ns feature

Example

```
> enable feature AAA
Done
```

```
> show ns feature
```

	Feature -----	Acronym -----	Status -----
1)	Web Logging	WL	OFF
2)	Surge Protection	SP	ON
.			
.			
15)	AAA	AAA	ON
.			
.			
23)	HTML Injection	HTMLInjection	ON
24)	NetScaler Push	push	OFF

```
Done
```

To enable AAA by using the configuration utility

1. Navigate to System > Settings.
2. In the details pane, under Modes and Features, click Change basic features.
3. In the Configure Basic Features dialog box, select the Authentication, Authorization and Auditing check box.
4. Click OK.

Setting up AAA Virtual Servers and DNS

You can configure AAA by using the built-in wizard, or manually. To use the wizard, in the main AAA pane of the configuration utility, you click AAA - Application Traffic wizard and follow the prompts.

To configure AAA manually, you first configure an authentication virtual server, which involves binding an SSL certificate-key pair. You then associate the authentication virtual server with a new or existing traffic management virtual server. (Either a load balancing virtual server or a content switching virtual server can serve as a traffic management virtual server.) To complete the initial configuration, you configure DNS to assign hostnames to both the authentication virtual server and the traffic management virtual server, and verify that your virtual servers are UP and configured correctly.

Caution: Both virtual servers must have hostnames in the same domain, or the AAA configuration will not work.

Configuring the Authentication Virtual Server

To configure AAA, first configure an authentication virtual server to handle authentication traffic. Next, bind an SSL certificate-key pair to the virtual server to enable it to handle SSL connections. For additional information about configuring SSL and creating a certificate-key pair, see the *Citrix NetScaler Traffic Management Guide* at "[Traffic Management](#)."

To configure an authentication virtual server by using the command line interface

To configure an authentication virtual server and verify the configuration, at the command prompt type the following commands in the order shown:

- o add authentication vserver <name> ssl <ipaddress>
- o show authentication vserver <name>
- o bind ssl certkey <certkeyName>
- o show authentication vserver <name>
- o set authentication vserver <name> -authenticationDomain <FQDN>
- o show authentication vserver <name>

Example

```
> add authentication vserver Auth-Vserver-2 SSL 10.102.29.77 443
Done
> show authentication vserver Auth-Vserver-2
Auth-Vserver-2 (10.102.29.77:443) - SSL Type: CONTENT
State: DOWN[Certkey not bound]
Client Idle Timeout: 180 sec
Down state flush: DISABLED
Disable Primary Vserver On Down : DISABLED
Authentication : ON
Current AAA Users: 0
Done
> bind ssl certkey Auth-Vserver-2 Auth-Cert-1
Done
> show authentication vserver Auth-Vserver-2
Auth-Vserver-2 (10.102.29.77:443) - SSL Type: CONTENT
State: UP
Client Idle Timeout: 180 sec
Down state flush: DISABLED
Disable Primary Vserver On Down : DISABLED
Authentication : ON
Current AAA Users: 0
Authentication Domain: myCompany.employee.com
Done
> set authentication vserver Auth-Vserver-2 -AuthenticationDomain myCompany.employee.com
Done
> show authentication vserver Auth-Vserver-2
Auth-Vserver-2 (10.102.29.77:443) - SSL Type: CONTENT
State: DOWN[Certkey not bound]
Client Idle Timeout: 180 sec
Down state flush: DISABLED
Disable Primary Vserver On Down : DISABLED
Authentication : ON
Current AAA Users: 0
Authentication Domain: myCompany.employee.com
Done
```

To configure an authentication virtual server by using the configuration utility

1. Navigate to Security > AAA - Application Traffic > Virtual Servers.
2. In the details pane, do one of the following:
 - o To create a new authentication virtual server, click Add.
 - o To modify an existing authentication virtual server, select the virtual server, and then click Edit.The Configuration dialog opens with the Basic Settings area expanded.
3. Specify values for the parameters as follows (asterisk indicates a required parameter):
 - o Name*â€"name (Cannot be changed for a previously created virtual server)
 - o IP Address*â€"ipaddress

- o Domain*â€"authenticationDomain
- o Failed login timeoutâ€"failedLoginTimeout (Seconds allowed before login fails and user must start login process again.)
- o Max login attemptsâ€"maxLoginAttempts (Number of login attempts allowed before user is locked out

Note: The authentication virtual server uses only the SSL protocol and port 443, so those options are greyed out. Any options that are not mentioned are not relevant and should be ignored.

4. Click Continue to display the Certificates area.
5. In the Certificates area, configure any SSL certificates you want to use with this virtual server.
 - o To configure a CA certificate, click the arrow on the right of CA Certificate to display the CA Cert Key dialog box, select the certificate you want to bind to this virtual server, and click Save.
 - o To configure a server certificate, click the arrow on the right of Server Certificate, and follow the same process as for CA certificate.
6. Click Continue to display the Advanced Authentication Policies area.
7. If you want to bind an advanced authentication policy to the virtual server, click the arrow on the right side of the line to display the Authentication Policy dialog box, choose the policy that you want to bind to the server, set the priority, and then click OK.
8. Click Continue to display the Basic Authentication Policies area.
9. If you want to create a basic authentication policy and bind it to the virtual server, click the plus sign to display the Policies dialog box, and follow the prompts to configure the policy and bind it to this virtual server.
10. Click Continue to display the 401-Based Virtual Servers area.
11. In the 401-Based Virtual Servers area, configure any load balancing or content switching virtual servers that you want to bind to this virtual server.
 - o To bind a load balancing virtual server, click the arrow to the right of LB virtual server to display the LB Virtual Servers dialog box, and follow the prompts.
 - o To bind a content switching virtual server, click the arrow to the right of CS virtual server to display the CS Virtual Servers dialog box, and follow the same process as to bind an LB virtual server.
12. If you want to create or configure a group, in the Groups area click the arrow to display the Groups dialog box, and follow the prompts.
13. Review your settings, and when you are finished, click Done. The dialog box closes. If you created a new authentication virtual server, it now appears in the Configuration window list.

Configuring a Traffic Management Virtual Server

After you have created and configured your authentication virtual server, you next create or configure a traffic management virtual server and associate your authentication virtual server with it. You can use either a load balancing or content switching virtual server for a traffic management virtual server. For more information about creating and configuring either type of virtual server, see the *Citrix NetScaler Traffic Management Guide* at [Traffic Management](#).

Note: The FQDN of the traffic management virtual server must be in the same domain as the FQDN of the authentication virtual server for the domain session cookie to function correctly.

You configure a traffic management virtual server for AAA by enabling authentication and then assigning the FQDN of the authentication server to the traffic management virtual server. You can also configure the authentication domain on the traffic management virtual server at this time. If you do not configure this option, the NetScaler appliance assigns the traffic management virtual server an FQDN that consists of the FQDN of the authentication virtual server without the hostname portion. For example, if domain name of the authentication vserver is `tm.xyz.bar.com`, the appliance assigns `xyz.bar.com` as the authentication domain.

To configure a TM virtual server for AAA by using the command line interface

At the command prompt, type one of the following sets of commands to configure a TM virtual server and verify the configuration:

- o `set lb vserver <name> -auth="authentication ON -authenticationhost <FQDN> [-authenticationdomain <authdomain>]`
- o `show lb vserver <name>`
- o `set cs vserver <name> -auth="authentication ON -authenticationhost <FQDN> [-authenticationdomain <authdomain>]`
- o `show cs vserver <name>`

Example

```
> set lb vserver vs-cont-sw -Authentication ON -AuthenticationHost mywiki.index.com
Done
> show lb vserver vs-cont-sw
vs-cont-sw (0.0.0.0:0) - TCP      Type: ADDRESS
State: DOWN
Last state change was at Wed Aug 19 10:03:15 2009 (+410 ms)
Time since last state change: 5 days, 20:00:40.290
Effective State: DOWN
Client Idle Timeout: 9000 sec
Down state flush: ENABLED
Disable Primary Vserver On Down : DISABLED
No. of Bound Services : 0 (Total)      0 (Active)
Configured Method: LEASTCONNECTION
Mode: IP
Persistence: NONE
Connection Failover: DISABLED
Authentication: ON      Host: mywiki.index.com
Done
```

To configure a TM virtual server for AAA by using the configuration utility

1. In the navigation pane, do one of the following.
 - o Navigate to Traffic Management > Load Balancing > Virtual Servers.
 - o Navigate to Traffic Management > Content Switching > Virtual ServersThe AAA configuration process for either type of virtual server is identical.
2. In the details pane, select the virtual server on which you want to enable authentication, and then click Edit.
3. In the Domain text box, type the authentication domain.
4. In the Advanced menu on the right, select Authentication.
5. Choose either Form Based Authentication or 401 Based Authentication., and fill in the Authentication information.
 - o For Form Based Authentication, enter the Authentication FQDN (the fully-qualified domain name of the authentication server), the Authentication VServer (the IP address of the authentication virtual server), and the Authentication Profile (the profile to use for authentication).
 - o For 401 Based Authentication, enter the Authentication VServer and the Authentication Profile only.
6. Click OK. A message appears in the status bar, stating that the vserver has been configured successfully.

Configuring DNS

For the domain session cookie used in the authentication process to function correctly, you must configure DNS to assign both the authentication and the traffic management virtual servers to FQDNs in the same domain. For information about how to the configure DNS address records, see the *Citrix NetScalerTraffic Management Guide* at "[Traffic Management](#)".

Verifying Your Setup for AAA

After you configure authentication and traffic management virtual servers and before you create user accounts, you should verify that both virtual servers are configured correctly and are in the UP state.

To verify authentication virtual server setup by using the command line interface

At the command prompt, type the following command:
show authentication vserver <name>

Example

```
> show authentication vserver Auth-Vserver-2
Auth-Vserver-2 (10.102.29.77:443) - SSL Type: CONTENT
State: UP
Client Idle Timeout: 180 sec
Down state flush: DISABLED
Disable Primary Vserver On Down : DISABLED
Authentication : ON
Current AAA Users: 0
Authentication Domain: myCompany.employee.com

Done
```

To verify your AAA virtual server setup by using the configuration utility

1. Navigate to Security > AAA - Application Traffic > Virtual Servers.
2. Review the information in the AAA Virtual Servers pane to verify that your configuration is correct and your authentication virtual server is accepting traffic. You can select a specific virtual server to view detailed information in the details pane.

Authentication Profile

An authentication profile specifies the authentication virtual server, authentication host, authentication domain, and authentication level in one location. You can then use the profile to configure authentication for different traffic management virtual servers that use the same authentication virtual server.

To configure an authentication profile by using the command line

To configure an authentication profile, bind that authentication profile to a traffic management virtual server, and verify the configuration, at the command prompt type the following commands:

- o add authentication authnProfile <name> {-authnVsName <string>} {-authenticationHost <string>} {-authenticationDomain <string>} [-authenticationLevel <positive_integer>]
- o set lb vserver <vserverName> -authnProfile <authnProfileName>
- o show authentication authnProfile <name>
- o show authentication vserver <name>

Example

```
> add authentication authnProfile authProfile -authnVsName authVS
    -authenticationHost authnVS.example.com -authenticationDomain example.com
    -authenticationLevel 1
Done
```

To configure an authentication profile by using the configuration utility

1. Navigate to Security > AAA - Application Traffic > Authentication Profile.
2. In the details pane, do one of the following:
 - o To create a new authentication profile, click Add.
 - o To modify an existing authentication profile, select the profile, and then click Edit.
3. In the Create Authentication Profile or the Configure Authentication Profile dialog, fill in the fields as described below:
 - o Nameâ€™Name for the authentication profile. Must begin with a letter, number, or the underscore character (_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (#), space , at (@), equals (=), colon (:), and underscore characters. Can be from 1 to 127 characters long. Cannot be changed for existing authentication profiles.
 - o Authentication VServer Nameâ€™NetScaler name of the authentication virtual server to which this profile applies.
 - o Authentication Hostâ€™Hostname of the authentication virtual server.
 - o Authentication Domainâ€™Domain of the authentication virtual server.
 - o Authentication Levelâ€™How many authentication levels this authentication host performs.
4. Click Create or OK.

Configuring Users and Groups

After configuring the AAA basic setup, you create users and groups. You first create a user account for each person who will authenticate via the NetScaler appliance. If you are using local authentication controlled by the NetScaler appliance itself, you create local user accounts and assign passwords to each of those accounts.

You also create user accounts on the NetScaler appliance if you are using an external authentication server. In this case, however, each user account must exactly match an account for that user on the external authentication server, and you do not assign passwords to the user accounts that you create on the NetScaler. The external authentication server manages the passwords for users that authenticate with the external authentication server.

If you are using an external authentication server, you can still create local user accounts on the NetScaler appliance if, for example, you want to allow temporary users (such as visitors) to log in but do not want to create entries for those users on the authentication server. You assign a password to each local user account, just as you would if you were using local authentication for all user accounts.

Each user account must be bound to policies for authentication and authorization. To simplify this task, you can create one or more groups and assign user accounts to them. You can then bind policies to groups instead of individual user accounts.

To create a local AAA user account by using the command line interface

At the command prompt, type the following commands to create a local AAA user account and verify the configuration:

- add aaa user <username> [â€“password <password>]
- show aaa user

Example

```
> add aaa user user-2 -password emptybag
Done
> show aaa user
1)      UserName: user-1
2)      UserName: user-2
Done
```

To change the password for an existing AAA local user account by using the command line interface

At the command prompt, type the following command and, when prompted, type the new password:
set aaa user <username>

Example

```
> set aaa user user-2
Enter password:
Done
```

To configure AAA local users by using the configuration utility

1. Navigate to Security > AAA - Application Traffic > Users
2. In the details pane, do one of the following:
 - To create a new user account, click Add.
 - To modify an existing user account, select the user account, and then click Open.
3. In the Create AAA User dialog box, in the User Name text box, type a name for the user.
4. If creating a locally authenticated user account, clear the External Authentication check box and provide a local password that the user will use to log on.
5. Click Create or OK, and then click Close. A message appears in the status bar, stating that the user has been configured successfully.

To create AAA local groups and add users to them by using the command line interface

At the command prompt, type the following commands. Type the first command one time, and type the second command once for each user:

- o add aaa group <groupname>
- o show aaa group

Example

```
> add aaa group group-2
Done
> show aaa group
1)      GroupName: group-1
2)      GroupName: group-2
Done
```

- o bind aaa group <groupname> -username <username>

Example

```
> bind aaa group group-2 -username user-2
Done
> show aaa group group-2
      GroupName: group-2

      UserName: user-2
Done
```

To remove users from an AAA group by using the command line interface

At the command prompt, unbind users from the group by typing the following command once for each user account that is bound to the group:

```
unbind aaa group <groupname> -username <username>
```

Example

```
> unbind aaa group group-hr -username user-hr-1
Done
```

To remove an AAA group by using the command line interface

First remove all users from the group. Then, at the command prompt, type the following command to remove an AAA group and verify the configuration:

- o rm aaa group <groupname>
- o show aaa group

Example

```
> rm aaa group group-hr
Done
> show aaa group
1)      GroupName: group-1
2)      GroupName: group-finance
Done
```

To configure AAA local groups and add users to them by using the configuration utility

1. Navigate to Security > AAA - Application Traffic > Groups
2. In the details pane, do one of the following:
 - o To create a new group, click Add.
 - o To modify an existing group, select the group, and then click Edit.
3. If you are creating a new group, in the Create AAA Group dialog box, in the Group Name text box, type a name for the group.
4. In the Advanced area to the right, click AAA Users.

- a. To add a user to the group, select the user, and then click Add.
 - b. To remove a user from the group, select the user, and then click Remove.
 - c. To create a new user account and add it to the group, click the Plus icon, and then follow the instructions in ["To configure AAA local users by using the configuration utility."](#)
5. Click Create or OK. The group that you created appears in the AAA Groups page.

Configuring AAA Policies

After you set up your users and groups, you next configure authentication policies, authorization policies, and audit policies to define which users are allowed to access your intranet, which resources each user or group is allowed to access, and what level of detail AAA will preserve in the audit logs. An authentication policy defines the type of authentication to apply when a user attempts to log on. If external authentication is used, the policy also specifies the external authentication server. Authorization policies specify the network resources that users and groups can access after they log on. Auditing policies define the audit log type and location.

You must bind each policy to put it into effect. You bind authentication policies to authentication virtual servers, authorization policies to one or more user accounts or groups, and auditing policies both globally and to one or more user accounts or groups.

When you bind a policy, you assign a priority to it. The priority determines the order in which the policies you define are evaluated. You can set the priority to any positive integer. In the NetScaler operating system, policy priorities work in reverse order: the higher the number, the lower the priority. For example, if you have three policies with priorities of 10, 100, and 1000, the policy assigned a priority of 10 is performed first, then the policy assigned a priority of 100, and finally the policy assigned an order of 1000. The AAA feature implements only the first of each type of policy that a request matches, not any additional policies of that type that a request might also match, so policy priority is important for getting the results you intend.

You can leave yourself plenty of room to add other policies in any order, and still set them to evaluate in the order you want, by setting priorities with intervals of 50 or 100 between each policy when you bind the policies. You can then add additional policies at any time without having to reassign the priority of an existing policy.

For additional information about binding policies on the NetScaler, see the *Citrix NetScaler Traffic Management Guide* at "[Traffic Management](#)."

Authentication Policies

The NetScaler ADC can authenticate users with local user accounts or by using an external authentication server. The appliance supports the following authentication types:

LOCAL

Authenticates to the NetScaler by using a password, without reference to an external authentication server. User data is stored locally on the NetScaler appliance.

RADIUS

Authenticate to an external Radius server.

LDAP

Authenticates to an external LDAP authentication server.

TACACS

Authenticates to an external Terminal Access Controller Access-Control System (TACACS) authentication server.

After a user authenticates to a TACACS server, the NetScaler ADC connects to the same TACACS server for all subsequent authorizations. When a primary TACACS server is unavailable, this feature prevents delays while the ADC waits for the first TACACS server to time out before resending the authorization request to the second TACACS server.

Note: When authenticating through a TACACS server, AAA-TM logs only successfully executed TACACS commands, to prevent the logs from showing TACACS commands that were entered by users who were not authorized to execute them.

CERT

Authenticates to the NetScaler appliance by using a client certificate, without reference to an external authentication server.

NEGOTIATE

Authenticates to a Kerberos authentication server. If there is an error in Kerberos authentication, NetScaler uses NTLM authentication.

SAML

Authenticates to a server that supports the Security Assertion Markup Language (SAML).

SAMLIDP

Configures the NetScaler ADC to serve as a Security Assertion Markup Language (SAML) Identity Provider (IdP).

WEB

Authenticates to a web server, providing the credentials that the web server requires in an HTTP request and analyzing the web server response to determine that user authentication was successful.

An authentication policy is comprised of an expression and an action. Authentication policies use NetScaler expressions.

After creating an authentication action and an authentication policy, bind it to an authentication virtual server and assign a priority to it. When binding it, also designate it as either a primary or a secondary policy. Primary policies are evaluated before secondary policies. In configurations that use both types of policy, primary policies are normally more specific policies while secondary policies are normally more general policies intended to handle authentication for any user accounts that do not meet the more specific criteria.

To add an authentication action by using the command line interface

If you do not use LOCAL authentication, you need to add an explicit authentication action. To do this, at the command prompt, type the following command:

```
add authentication tacacsAction <name> -serverip <IP> [-serverPort <port>] [-authTimeout <positive_integer>] [ ... ]
```

Example

```
> add authentication tacacsaction Authn-Act-1 -serverip 10.218.24.65 -serverport 1812
```

To configure an authentication action by using the command line interface

To configure an existing authentication action, at the command prompt, type the following command:

```
set authentication tacacsAction <name> -serverip <IP> [-serverPort <port>] [-authTimeout <positive_integer>] [ ... ]
```

Example

```
> set authentication tacacsaction Authn-Act-1 -serverip 10.218.24.65 -serverport 1812
```

To remove an authentication action by using the command line interface

To remove an existing RADIUS action, at the command prompt, type the following command:

```
rm authentication radiusAction <name>
```

Example

```
> rm authentication tacacsaction Authn-Act-1 Done
```

To configure an authentication server by using the configuration utility

Note: In the configuration utility, the term server is used instead of action, but refers to the same task.

1. Navigate to Security > AAA - Application Traffic > Policies > Authentication.
2. In the details pane, on the Servers tab, do one of the following:
 - o To create a new authentication server, click Add.
 - o To modify an existing authentication server, select the server, and then click Open.
3. In the Create Authentication Server or Configure Authentication Server dialog box, type or select values for the parameters.
 - o Name*â€”radiusActionName (Cannot be changed for a previously configured action)
 - o Authentication Type*â€”authtype (Set to RADIUS, cannot be changed)
 - o IP Address*â€”serverip <IP>
 - o IPV6*â€”Select the checkbox if the server IP is an IPv6 IP. (No command line equivalent.)
 - o Port*â€”serverPort
 - o Time-out (seconds)*â€”authTimeout
4. Click Create or OK, and then click Close. The policy that you created appears in the Authentication Policies and Servers page.

To create and bind an authentication policy by using the command line interface

At the command prompt, type the following commands in the order shown to create and bind an authentication policy and verify the configuration:

- o add authentication negotiatePolicy <name> <rule> <reqAction>
- o show authentication localPolicy <name>
- o bind authentication vserver <name> -policy <policyname> [-priority <priority>] [-secondary]]
- o show authentication vserver <name>

Example

```
> add authentication localPolicy Authn-Pol-1 ns_true Done > show authentication local
```

To modify an existing authentication policy by using the command line interface

At the command prompt, type the following commands to modify an existing authentication policy:

```
set authentication localPolicy <name> <rule> [-reqaction <action>]
```

Example

```
> set authentication localPolicy Authn-Pol-1 'ns_true' Done
```

To remove an authentication policy by using the command line interface

At the command prompt, type the following command to remove an authentication policy:

```
rm authentication localPolicy <name>
```

Example

```
> rm authentication localPolicy Authn-Pol-1 Done
```

To configure and bind authentication policies by using the configuration utility

1. Navigate to Security > AAA - Application Traffic > Policies > Authentication, and then select the type of policy that you want to create.
2. In the details pane, on the Policies tab, do one of the following:
 - o To create a new policy, click Add.
 - o To modify an existing policy, select the action, and then click Edit.
3. In the Create Authentication Policy or Configure Authentication Policy dialog, type or select values for the parameters.
 - o Name*â€”polycname (Cannot be changed for a previously configured action)
 - o Authentication Type*â€”authtype
 - o Server*â€”authVsName
 - o Expression*â€”rule (You enter expressions by first choosing the type of expression in the leftmost drop-down list beneath the Expression window, and then by typing your expression directly into the expression text area, or by clicking Add to open the Add Expression dialog box and using the drop-down lists in it to construct your expression.)
4. Click Create or OK. The policy that you created appears in the Policies page.
5. Click the Servers tab, and in the details pane do one of the following:
 - o To use an existing server, select it, and then click .
 - o To create a new server, click Add, and follow the instructions.
6. If you want to designate this policy as a secondary authentication policy, on the Authentication tab, click Secondary. If you want to designate this policy as a primary authentication policy, skip this step.
7. Click Insert Policy.
8. Choose the policy you want to bind to the authentication virtual server from the drop-down list.
9. In the Priority column to the left, modify the default priority as needed to ensure that the policy is evaluated in the proper order.
10. Click OK. A message appears in the status bar, stating that the policy has been configured successfully.

LDAP Authentication Policies

As with other types of authentication policies, a Lightweight Directory Access Protocol (LDAP) authentication policy is comprised of an expression and an action. After creating an authentication policy, you bind it to an authentication virtual server and assign a priority to it. When binding it, you also designate it as either a primary or a secondary policy. In addition to standard authentication functions, LDAP can search other active directory (AD) servers for user accounts for users that do not exist locally. This function is called referral support or referral chasing.

Normally you configure the NetScaler ADC to use the IP address of the authentication server during authentication. With LDAP authentication servers, you can also configure the ADC to use the FQDN of the LDAP server instead of its IP address to authenticate users. Using an FQDN can simplify an otherwise much more complex AAA configuration in environments where the authentication server might be at any of several IP addresses, but always uses a single FQDN. To configure authentication by using a server's FQDN instead of its IP address, you follow the normal configuration process except when creating the authentication action. When creating the action, you use the **serverName** parameter instead of the **serverIP** parameter, and substitute the server's FQDN for its IP address.

Before you decide whether to configure the ADC to use the IP or the FQDN of your LDAP server to authenticate users, consider that configuring AAA to authenticate to an FQDN instead of an IP address adds an extra step to the authentication process. Each time the ADC authenticates a user, it must resolve the FQDN. If a great many users attempt to authenticate simultaneously, the resulting DNS lookups might slow the authentication process.

LDAP referral support is disabled by default and cannot be enabled globally. It must be explicitly enabled for each LDAP action. You must also make sure that the AD server accepts the same `binddn` credentials that are used with the referring (GC) server. To enable referral support, you configure an LDAP action to follow referrals, and specify the maximum number of referrals to follow.

If referral support is enabled, and the NetScaler ADC receives an LDAP_REFERRAL response to a request, AAA follows the referral to the active directory (AD) server contained in the referral and performs the update on that server. First, AAA looks up the referral server in DNS, and connects to that server. If the referral policy requires SSL/TLS, it connects via SSL/TLS. It then binds to the new server with the `binddn` credentials that it used with the previous server, and performs the operation which generated the referral. This feature is transparent to the user.

Note: These instructions assume that you are already familiar with the LDAP protocol and have already configured your chosen LDAP authentication server.

For more information about setting up authentication policies in general, see "[Authentication Policies](#)". For more information about NetScaler expressions, which are used in the policy rule, see the *Citrix NetScaler Policy Configuration and Reference Guide* at "[Policies and Expressions](#)."

To enable LDAP referral support by using the command line interface

At the command prompt, type the following commands:

- o `set authentication ldapAction <name> -followReferrals ON`
- o `set authentication ldapAction <name> -maxLDAPReferrals <integer>`

Example

```
> set authentication ldapAction ldapAction-1 -followReferrals ON
set authentication ldapAction ldapAction-1 -maxLDAPReferrals 2
```

To enable LDAP referral support by using the configuration utility

Note: In the configuration utility, the term server is used instead of action, but refers to the same task.

1. Navigate to Security > AAA - Application Traffic > Policies > LDAP.
2. In the details pane, on the Servers tab, select the LDAP server that you want to configure, and then click Edit.
3. In the Configure Authentication Server dialog, scroll down to the Referrals check box, and select it.
4. In the Maximum Referral Level text box, type the maximum number of referrals to allow.
5. Click OK, and then click Close.

Negotiate Authentication Policies

As with other types of authentication policies, a Negotiate authentication policy is comprised of an expression and an action. After creating an authentication policy, you bind it to an authentication virtual server and assign a priority to it. When binding it, you also designate it as either a primary or a secondary policy.

In addition to standard authentication functions, the Negotiate Action command can now extract user information from a keytab file instead of requiring you to enter that information manually. If a keytab has more than one SPN, AAA selects the correct SPN. You can configure this feature at the NetScaler command line, or by using the configuration utility..

Note: These instructions assume that you are already familiar with the LDAP protocol and have already configured your chosen LDAP authentication server.

For more information about setting up authentication policies in general, see "[Authentication Policies](#)". For more information about NetScaler expressions, which are used in the policy rule, see the *Citrix NetScaler Policy Configuration and Reference Guide* at "[Policies and Expressions](#)".

To configure AAA to extract user information from a keytab file by using the command line interface

At the command prompt, type the appropriate command:

- add authentication negotiateAction <name> [-keytab <string>]
- set authentication negotiateAction <name> [-keytab <string>]

Example

```
> set authentication negotiateAction negotiateAction-1 -keytab keytab-1
```

To configure AAA to extract user information from a keytab file by using the configuration utility

Note: In the configuration utility, the term server is used instead of action, but refers to the same task.

1. Navigate to Security > AAA - Application Traffic > Policies > Negotiate.
2. In the details pane, on the Servers tab, do one of the following:
 - If you want to create a new Negotiate action, click Add.
 - If you want to modify an existing Negotiate action, in the data pane select the action, and then click Edit.
3. If you are creating a new Negotiate action, in the Name text box, type a name for your new action. The name can be from one to 127 characters in length and can consist of upper- and lowercase letters, numbers, and the hyphen (-) and underscore (_) characters. If you are modifying an existing Negotiate action, skip this step. The name is read-only; you cannot change it.
4. Under Negotiate, if the Use Keytab file check box is not already checked, check it.
5. In the Keytab file path text box, type the full path and filename of the keytab file that you want to use.
6. In the Default authentication group text box, type the authentication group that you want to set as default for this user.
7. Click Create or OK to save your changes.

RADIUS Authentication Policies

As with other types of authentication policies, a Remote Authentication Dial In User Service (RADIUS) authentication policy is comprised of an expression and an action. After creating an authentication policy, you bind it to an authentication virtual server and assign a priority to it. When binding it, you also designate it as either a primary or a secondary policy. However, setting up a RADIUS authentication policy has certain special requirements that are described below.

Normally you configure the NetScaler ADC to use the IP address of the authentication server during authentication. With RADIUS authentication servers, you can now configure the ADC to use the FQDN of the RADIUS server instead of its IP address to authenticate users. Using an FQDN can simplify an otherwise much more complex AAA configuration in environments where the authentication server might be at any of several IP addresses, but always uses a single FQDN. To configure authentication by using a server's FQDN instead of its IP address, you follow the normal configuration process except when creating the authentication action. When creating the action, you substitute the **serverName** parameter for the **serverIP** parameter.

Before you decide whether to configure the ADC to use the IP or the FQDN of your RADIUS server to authenticate users, consider that configuring AAA to authenticate to an FQDN instead of an IP address adds an extra step to the authentication process. Each time the ADC authenticates a user, it must resolve the FQDN. If a great many users attempt to authenticate simultaneously, the resulting DNS lookups might slow the authentication process.

Note: These instructions assume that you are already familiar with the RADIUS protocol and have already configured your chosen RADIUS authentication server.

For more information about setting up authentication policies in general, see "[Authentication Policies](#)." For more information about NetScaler expressions, which are used in the policy rule, see the *Citrix NetScaler Policy Configuration and Reference Guide* at "[Policies and Expressions](#)."

To add an authentication action for a RADIUS server by using the command line interface

If you authenticate to a RADIUS server, you need to add an explicit authentication action. To do this, at the command prompt, type the following command:

```
add authentication radiusAction <name> [-serverip <IP> | -serverName] <FQDN> [-serverPort <port>] [-authTimeout <positive_integer>] [-radKey } [-radNASip ( ENABLED | DISABLED )] [-radNASid <string>] [-radVendorID <positive_integer>] [-radAttributeType <positive_integer>] [-radGroupsPrefix <string>] [-radGroupSeparator <string>] [-passEncoding <passEncoding>] [-ipVendorID <positive_integer>] [-ipAttributeType <positive_integer>] [-accounting ( ON | OFF )] [-pwdVendorID <positive_integer>] [-pwdAttributeType <positive_integer>]] [-defaultAuthenticationGroup <string>] [-callingstationid ( ENABLED | DISABLED )]
```

Example

The following example adds a RADIUS authentication action named **Authn-Act-1**, with the server IP **10.218.24.65**, the server port **1812**, the authentication timeout **15** minutes, the radius key **WareTheLorax**, NAS IP disabled, and NAS ID **NAS1**.

```
> add authentication radiusaction Authn-Act-1 -serverip 10.218.24.65 -serverport 1812
    -authtimeout 15 -radkey WareTheLorax -radNASip DISABLED -radNASid NAS1
Done
```

The following example adds the same RADIUS authentication action, but using the server FQDN **rad01.example.com** instead of the IP.

```
> add authentication radiusaction Authn-Act-1 -serverName rad01.example.com -serverport 1812
    -authtimeout 15 -radkey WareTheLorax -radNASip DISABLED -radNASid NAS1
Done
```

To configure an authentication action for an external RADIUS server by using the command line

To configure an existing RADIUS action, at the NetScaler command prompt, type the following command:

```
set authentication radiusAction <name> [-serverip <IP> | -serverName] <FQDN> [-serverPort <port>] [-authTimeout <positive_integer>] [-radKey } [-radNASip ( ENABLED | DISABLED )] [-radNASid <string>] [-radVendorID <positive_integer>] [-radAttributeType <positive_integer>] [-radGroupsPrefix <string>] [-radGroupSeparator <string>] [-passEncoding <passEncoding>] [-ipVendorID <positive_integer>] [-ipAttributeType <positive_integer>] [-accounting ( ON | OFF )] [-pwdVendorID <positive_integer>] [-pwdAttributeType <positive_integer>]] [-defaultAuthenticationGroup <string>] [-callingstationid ( ENABLED | DISABLED )]
```

To remove an authentication action for an external RADIUS server by using the command line interface

To remove an existing RADIUS action, at the command prompt, type the following command:

```
rm authentication radiusAction <name>
```

Example

```
> rm authentication radiusaction Authn-Act-1  
Done
```

To configure a RADIUS server by using the configuration utility

Note: In the configuration utility, the term server is used instead of action, but refers to the same task.

1. Navigate to Security > AAA - Application Traffic > Policies > Authentication > Radius
2. In the details pane, on the Servers tab, do one of the following:
 - o To create a new RADIUS server, click Add.
 - o To modify an existing RADIUS server, select the server, and then click Edit.
3. In the Create Authentication RADIUS Server or Configure Authentication RADIUS Server dialog, type or select values for the parameters. To fill out parameters that appear beneath Send Calling Station ID, expand Details.
 - o Name*â€”radiusActionName (Cannot be changed for a previously configured action)
 - o Authentication Type*â€”authtype (Set to RADIUS, cannot be changed)
 - o Server Name / IP Address*â€”Choose either Server Name or Server IP
 - Server Name*â€”serverName <FQDN>
 - IP Address*â€”serverIp <IP> If the server is assigned an IPv6 IP address, select the IPv6 check box.
 - o Port*â€”serverPort
 - o Time-out (seconds)*â€”authTimeout
 - o Secret Key*â€”radKey (RADIUS shared secret.)
 - o Confirm Secret Key*â€”Type the RADIUS shared secret a second time. (No command line equivalent.)
 - o Send Calling Station IDâ€”callingstationid
 - o Group Vendor Identifierâ€”radVendorID
 - o Group Attribute Typeâ€”radAttributeType
 - o IP Address Vendor Identifierâ€”ipVendorID
 - o pwdVendorIDâ€”pwdVendorID
 - o Password Encodingâ€”passEncoding
 - o Default Authentication Groupâ€”defaultAuthenticationGroup
 - o NAS IDâ€”radNASid
 - o Enable NAS IP address extractionâ€”radNASip
 - o Group Prefixâ€”radGroupsPrefix
 - o Group Separatorâ€”radGroupSeparator
 - o IP Address Attribute Typeâ€”ipAttributeType
 - o Password Attribute Typeâ€”pwdAttributeType
 - o Accountingâ€”accounting
4. Click Create or OK. The policy that you created appears in the Servers page.

SAML Authentication Policies

The Security Assertion Markup Language (SAML) is an XML-based standard for exchanging authentication and authorization tokens between servers which authenticate users (the *Identity Provider* or *IdP*) and servers that host user applications (*Service Providers*). The NetScaler ADC supports SAML authentication and authorization with HTTP POST-binding, in which the ADC responds to user requests with a 200 OK that contains a form-auto post with the required authentication token.

The NetScaler ADC supports attribute extraction from SAML assertions, and encrypted SAML assertions. The NetScaler implementation of SAML allows signing certificates of less than 2048 bits, but displays a warning message. It also supports the SHA256 hash algorithm for signatures and digests. Citrix recommends that all signing certificates be of at least 2048 bits, and that you use SHA256 as SHA-1 is no longer considered secure.

As with other types of NetScaler authentication policies, a SAML authentication policy is comprised of an expression and an action. After creating an authentication policy, you bind it to an authentication virtual server and assign a priority to it. When binding it, you also designate it as either a primary or a secondary policy. However, setting up a SAML authentication policy has certain special requirements that are described below.

Note: These instructions assume that you are already familiar with the SAML protocol and have already configured your chosen SAML authentication server.

For more information about setting up authentication policies in general, see "Authentication Policies". For more information about NetScaler expressions, which are used in the policy rule, see the *Citrix NetScaler Policy Configuration and Reference Guide* at "Policies and Expressions".

To add an authentication action for an external SAML server by using the command line

If you authenticate to a SAML server, you need to add an explicit authentication action. To do this, at the command prompt, type the following command:

```
add authentication samlaction <name> [-samlIdPCertName <string>] [-samlSigningCertName <string>] [-samlRedirectUrl <string>] [-samlACSIndex <positive_integer>] [-samlUserField <string>] [-samlRejectUnsignedAssertion ( ON | OFF )] [-samlIssuerName <string>] [-samlTwoFactor ( ON | OFF )] [-defaultAuthenticationGroup <string>] [-Attribute1 <string>] [-Attribute2 <string>] [-Attribute3 <string>] [-Attribute4 <string>] [-Attribute5 <string>] [-Attribute6 <string>] [-Attribute7 <string>] [-Attribute8 <string>] [-Attribute9 <string>] [-Attribute10 <string>] [-Attribute11 <string>] [-Attribute12 <string>] [-Attribute13 <string>] [-Attribute14 <string>] [-Attribute15 <string>] [-Attribute16 <string>] [-signatureAlg ( RSA-SHA1 | RSA-SHA256 )] [-digestMethod ( SHA1 | SHA256 )]
```

Example

The following example adds a SAML authentication action named Authn-Act-1.

```
> add authentication samlaction Authn-Act-1 -samlIdPCertName samlcert1
   -samlSigningCertName ssigncert1 -samlRedirectUrl https://login.example.com/login
   -samlUserField userfield1 -samlRejectUnsignedAssertion ON -samlIssuerName Issuer
   -samlTwoFactor ON -defaultAuthenticationGroup group -signatureAlg RSA-SHA256
   -digestMethod SHA256
```

Done

To configure an authentication action for an external SAML server by using the command line

To configure an existing SAML action, at the command prompt, type the following command:

```
set authentication samlaction <name> [-samlIdPCertName <string>] [-samlSigningCertName <string>] [-samlRedirectUrl <string>] [-samlUserField <string>] [-samlRejectUnsignedAssertion ( ON | OFF )] [-samlIssuerName <string>] [-samlTwoFactor ( ON | OFF )] [-defaultAuthenticationGroup <string>] [-Attribute1 <string>] [-Attribute2 <string>] [-Attribute3 <string>] [-Attribute4 <string>] [-Attribute5 <string>] [-Attribute6 <string>] [-Attribute7 <string>] [-Attribute8 <string>] [-Attribute9 <string>] [-Attribute10 <string>] [-Attribute11 <string>] [-Attribute12 <string>] [-Attribute13 <string>] [-Attribute14 <string>] [-Attribute15 <string>] [-Attribute16 <string>] [-signatureAlg ( RSA-SHA1 | RSA-SHA256 )] [-digestMethod ( SHA1 | SHA256 )]
```

Example

```
> add authentication samlaction Authn-Act-1 -samlIdPCertName samlcert1
   -samlSigningCertName ssigncert1 -samlRedirectUrl https://login.example.com/login
   -samlUserField userfield1 -samlRejectUnsignedAssertion ON -samlIssuerName Issuer
```

```
-samlTwoFactor ON -defaultAuthenticationGroup group -signatureAlg RSA-SHA256  
-digestMethod SHA256
```

Done

To remove an authentication action for an external SAML server by using the command line

To remove an existing SAML action, at the command prompt, type the following command:

```
rm authentication samlaction <name>
```

Example

```
> rm authentication samlaction Authn-Act-1  
Done
```

To configure a SAML server by using the configuration utility

Note: In the configuration utility, the term server is used instead of action, but refers to the same task.

1. Navigate to AAA - Application Traffic > Policies > Authentication > SAML.
2. In the details pane, on the Servers tab, do one of the following:
 - o To create a new SAML server, click Add.
 - o To modify an existing SAML server, select the server, and then click Edit.
3. In the Create Authentication Server or Configure Authentication Server dialog box, type or select values for the parameters:
 - o Name*â€”policyname (Cannot be changed for a previously configured action)
 - o Authentication Type*â€”authtype (Set to SAML, cannot be changed)
 - o IDP Certificate Name*â€”samlIDPcertname
 - o Redirect URL*â€”samlRedirectUrl
 - o User Fieldâ€”samlUserField
 - o Signing Certificate Nameâ€”samlSigningcertname
 - o Issuer Nameâ€”samlIssuerName
 - o Default Authentication Groupâ€”defaultAuthenticationGroup
 - o Two Factorâ€”samlTwoFactor
 - o Reject Unsigned Assertionâ€”samlRejectUnsignedAssertion
 - o ACS Indexâ€”samlACSIndex
 - o Attribute 1-Attribute 16â€”attribute1-attribute16 (Used to extract attributes from the SAML assertion.)
4. Click Create or OK. The policy that you created appears in the Servers page.

SAML IDP Authentication Policies

As with other types of authentication policies, a Security Assertion Markup Language (SAML) intrusion detection policy (IdP) authentication policy is comprised of an expression and an action. After creating an authentication policy, you bind it to an authentication virtual server and assign a priority to it. When binding it, you also designate it as either a primary or a secondary policy. However, setting up a SAML IdP authentication policy has certain special requirements that are described below.

Note: These instructions assume that you are already familiar with the SAML protocol, understand SAML IdP, and have already configured your chosen SAML authentication server.

For more information about setting up authentication policies in general, see "Authentication Policies". For more information about NetScaler expressions, which are used in the policy rule, see the *Citrix NetScaler Policy Configuration and Reference Guide* at ""Policies and Expressions".

To configure a SAML IdP authentication policy by using the configuration utility

1. Navigate to Security > AAA - Application Traffic > Policies > Authentication > SAML IDP.
2. In the details pane, do one of the following:
 - To add a new policy, click Add.
 - To configure an existing policy, select the policy and then click Edit.The Create SAML IDP Policy or the Configure SAML IDP Policy dialog box opens. The dialog boxes are nearly identical.
3. If you are creating a new policy, in the Create Create SAML IDP Policy dialog box, type a name for your new policy.

The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space (), at (@), equals (=), colon (:), and underscore (_) symbols.

4. Select the action (profile) that you want to bind to the policy from the list of available actions, or click the plus sign and create a new action.
5. Add a policy expression (rule) in the Expression window. You enter expressions by first choosing the type of expression in the leftmost drop-down list beneath the Expression window, and then by typing your expression directly into the expression text area, or by clicking Add to open Add Expression dialog box and using the drop-down lists in it to construct your expression.
6. Click Create or OK.

Web Authentication Policies

AAA-TM is now able to authenticate a user to a web server, providing the credentials that the web server requires in an HTTP request and analyzing the web server response to determine that user authentication was successful. As with other types of authentication policies, a Web authentication policy is comprised of an expression and an action. After creating an authentication policy, you bind it to an authentication virtual server and assign a priority to it. When binding it, you also designate it as either a primary or a secondary policy. I

To set up web-based authentication with a specific web server, first you create a web authentication action. Since authentication to web servers does not use a rigid format, you must specify exactly which information the web server requires and in which format when creating the action. To do this, you create an expression in NetScaler default syntax that contains the following items:

- **Server IP**—The IP address of the authentication Web server.
- **Server Port**—The port of the authentication Web server.
- **Authentication Rule**—An expression in NetScaler default syntax that contains the user's credentials in the format that the Web server expects.
- **Scheme**—HTTP (for unencrypted web authentication) or HTTPS (for encrypted web authentication).
- **Success Rule**—An expression in NetScaler default syntax that matches the web server response string that signifies that the user authenticated successfully.

For all other parameters, follow the normal rules for the add authentication action command.

Next you create a policy associated with that action. The policy is similar to an LDAP policy, and like LDAP policies uses NetScaler classic syntax.

Note: These instructions assume that you are already familiar with the authentication requirements of the web server(s) to which you want to authenticate, and have already configured the web authentication server.

For more information about setting up authentication policies in general, see "[Authentication Policies](#)". For more information about NetScaler expressions, which are used in the policy rule, see the *Citrix NetScaler Policy Configuration and Reference Guide* at "[Policies and Expressions](#)."

To configure a Web authentication action by using the command line interface

To create a web authentication action at the command line, at the command line type the following command:

```
add authentication webAuthAction <name> -serverIP <ip_addr|ipv6_addr*> -serverPort <port*> [-fullReqExpr <string>] -
scheme ( http | https ) -successRule <expression> [-defaultAuthenticationGroup <string>] [-Attribute1 <string>] [-Attribute2
<string>] [-Attribute3 <string>] [-Attribute4 <string>] [-Attribute5 <string>] [-Attribute6 <string>] [-Attribute7 <string>] [-Attribute
<string>] [-Attribute9 <string>] [-Attribute10 <string>] [-Attribute11 <string>] [-Attribute12 <string>] [-Attribute13 <string>] [-
Attribute14 <string>] [-Attribute15 <string>] [-Attribute16 <string>]
```

Example

```
> add authentication webAuthAction webauth1 -serverIP 10.214.56.31 -serverPort 80 -
```

To configure a Web authentication action by using the configuration utility

Note: In the configuration utility, the term server is used instead of action, but refers to the same task.

1. Navigate to Security > AAA - Application Traffic > Policies > LDAP.
2. In the details pane, on the Servers tab, do one of the following:
 - If you want to create a new web authentication action, click Add.
 - If you want to modify an existing web authentication action, in the data pane select the action, and then click Edit.
3. If you are creating a new web authentication action, in the Create Authentication Web server dialog box, Name text box, type a name for the new web authentication action. The name can be from one to 127 characters in length, and can consist of upper- and lowercase letters, numbers, and the hyphen (-) and underscore (_) characters. If you are modifying an existing web authentication action, skip this step. The name is read-only; you cannot change it.
4. In the Web Server IP Address text box, type the IPv4 or IPv6 IP address of the authentication web server. If the address is an IPv6 IP address, select the IPv6 check box first.
5. In the Port text box, type the port number on which the web server accepts connections.
6. Select HTTP or HTTPS in the Protocol drop-down list.
7. In the HTTP Request Expression text area, type a PCRE-format regular expression that creates the web server request that contains the user's credentials in the exact format expected by the authentication web server.

8. In the Expression to validate the Authentication text area, type a NetScaler default syntax expression that describes the information in the web server response that indicates that user authentication was successful.
9. Fill out the remaining fields as described in the general authentication action documentation.
10. Click OK.

Configuring Advanced Authentication Policies

If you know exactly how you want an authentication policy to be configured, you can use the advanced authentication policy dialog to create the policy quickly.

To configure an advanced authentication policy by using the configuration utility

1. Navigate to Security > AAA - Application Traffic > Policies > Authentication > Advanced Policies, and then select Policies.
2. In the details pane do one of the following:
 - To create a new policy, click Add.
 - To modify an existing policy, select the policy, and then click Edit.
3. In the Create Authentication Policy or Configure Authentication Policy dialog box, type or select values for the parameters.
 - Name*â€”The policy name. Cannot be changed for a previously configured policy.
 - Action Type*â€”The policy type: Cert, Negotiate, LDAP, RADIUS, SAML, SAMLIDP, TACACS, or WEBAUTH.
 - Action*â€”The authentication action (profile) to associate with the policy. You can choose an existing authentication action, or click the plus and create a new action of the proper type.
 - Log Action*â€”The audit action to associate with the policy. You can choose an existing audit action, or click the plus and create a new action.
 - Expression*â€”The rule that selects connections to which you want to apply the action that you specified. The rule can be simple ("true" selects all traffic) or complex. You enter expressions by first choosing the type of expression in the leftmost drop-down list beneath the Expression window, and then by typing your expression directly into the expression text area, or by clicking Add to open Add Expression dialog box and using the drop-down lists in it to construct your expression.)
 - Comment*â€”You can type a comment that describes the type of traffic that this authentication policy will apply to. Optional.
4. Click Create or OK, and then click Close. If you created a policy, that policy appears in the Authentication Policies and Servers page.

Authorization Policies

After you create authentication policies, you next create any authorization policies you need. Authorization policies, like other policies, consist of an expression and action. There are only two actions for authorization policies: ALLOW and DENY. ALLOW permits users to access the specified resource; DENY blocks access. The default setting for authorization when no specific policy exists is to deny access to network resources. This means that a user or group can access a particular resource only if an authorization policy explicitly allows access. For optimum security, the best practice is not to change the default setting and to create specific authorization policies for users who need access to specific resources.

Authorization use both default syntax expressions and classic expressions. These expressions are described in detail in the *Citrix NetScaler Policy Configuration and Reference Guide* at "[Policies and Expressions](#)."

After you create an authorization policy, you bind it to the appropriate user accounts or groups to put it into effect.

To create an authorization policy

At the NetScaler command prompt, type the following commands to create an authorization policy and verify the configuration:

- o add authorization policy <name> <rule> <action>
- o show authorization policy <name>

Example

```
> add authorization policy authz-pol-1 "HTTP.REQ.URL.SUFFIX.EQ(\"gif\")" DENY
Done
> show authorization policy authz-pol-1
1)      Name: authz-pol-1      Rule: HTTP.REQ.URL.SUFFIX.EQ("gif")
      Action: DENY
Done
>
```

To modify an authorization policy

At the command prompt, type the following command to modify an authorization policy:
set authorization policy <name> [-rule <expression>] -action <action>

Example

```
> set authorization policy authz-pol-1 -rule "HTTP.REQ.URL.SUFFIX.EQ(\"gif\")" -action ALLOW
Done
> show authorization policy authz-pol-1
1)      Name: authz-pol-1      Rule: HTTP.REQ.URL.SUFFIX.EQ("gif")
      Action: ALLOW
Done
>
```

To bind an authorization policy to a user account or group

At the command prompt, type one of the following commands to bind an authorization policy to a user account or group and verify the configuration:

- o bind aaa user <userName> [-policy <polycname> [-priority <priority>]] [-intranetApplication <appname>] [-urlName <urlname>] [-intranetIP <intranetip> [<netmask>]]
- o show aaa user <userName>
- o bind aaa group <groupName> [-policy <polycname> [-priority <priority>]] [-intranetApplication <appname>] [-urlName <urlname>] [-intranetIP <intranetip> [<netmask>]]
- o show aaa group <name>

Example

```
> bind aaa user user-hr-1 -policy authz-pol-1
Done
> show aaa user user-hr-1
      UserName: user-hr-1
```

```

Policy: authz-pol-1, Priority: 0
Done
> bind aaa group group-1 -policy authz-pol-1
Done
> show aaa group group-1
    GroupName: group-1
        UserName: user-2
        UserName: user-1
Policy: authz-pol-1, Priority: 0
Done

```

To unbind an authorization policy from a user account or group

At the command prompt, type one of the following commands to unbind an authorization policy from a user account or group:

- o unbind aaa user <userName> -policy <policyname>
- o unbind aaa group <groupName> -policy <policyname>

Example

```

> unbind aaa user aaa-user-1 -policy auth-pol-1
Done

```

To remove an authorization policy

First unbind the policy from all user accounts and groups, and then, at the NetScaler command prompt, type the following command to remove an authorization policy:

```
rm authorization policy <name>
```

To configure and bind authorization policies by using the configuration utility

1. Navigate to Security > AAA - Application Traffic > Authorization.
2. Navigate to Security > AAA - Application Traffic > Policies > Authorization.
3. In the details pane, do one of the following:
 - o To create a new authorization policy, click Add.
 - o To modify an existing authorization policy, select the policy, and then click Edit.
4. In the Create Authorization Policy or Configure Authorization Policy dialog, type or select values for the parameters.
 - o Name*â€™policyname(Cannot be changed for a previously configured policy.)
 - o Action*â€™ action
 - o Expression*â€™rule (By default, the Expression box accepts default syntax policies. To switch to the classic syntax view, click Switch to Classic Syntax.)
5. Click Create or OK. The policy that you created appears on the Authorization Policies page.
6. To bind an authorization policy to a user account or group, in the navigation pane, under AAA - Application Traffic, click either Users or Groups, as appropriate, and then add that policy to the user account list:
 - a. In the details pane, select the appropriate user account, and then click Edit.
 - b. In the Advanced menu on the right, click Authorization Policies.
 - c. Click the arrow to the right of the list of authorization policies to open the Bind Policy dialog.
 - d. Select the policy you want to bind to the user account or group.
 - e. In the Priority column, modify the default priority as needed to ensure that the policy is evaluated in the proper order.
 - f. Click OK.

A message appears in the status bar, stating that the policy has been configured successfully.

Auditing Policies

After you create authentication policies, you next create any auditing policies you need. The NetScaler ADC allows auditing of all states and status information, so you can see the event history for any user in chronological order. When you configure auditing on the ADC, you can choose to store the log files locally on the ADC or to send them to a syslog server.

To put your auditing policies into effect, you bind them globally, to a specific authentication virtual server, or to specific user accounts or groups.

To create an auditing policy by using the command line interface

At the command prompt, type the following commands to create an auditing policy and verify the configuration:

- o add audit nslogPolicy <name> [-rule <rule>] [-action <action>]
- o show audit nslogPolicy

Example

```
> add audit nslogPolicy audit-1 ns_true audit_server
Done
> show audit nslogPolicy
1)      Name: audit-pol Rule: ns_true
        Action: audit_server
2)      Name: audit-1   Rule: ns_true
        Action: audit_server
Done
```

To modify an auditing policy by using the command line interface

At the command prompt, type the following commands to modify an auditing policy and verify the configuration:

- o set audit nslogPolicy <name> [-rule <expression>] [-action <string>]
- o show audit nslogPolicy

Example

```
> set audit nslogPolicy audit-1 ns_true audit_server
Done
> show audit nslogPolicy
1)      Name: audit-pol Rule: ns_true
        Action: audit_server
2)      Name: audit-1   Rule: ns_true
        Action: audit_server
Done
```

To globally bind an auditing policy by using the command line interface

At the command prompt, type the following commands to globally bind an auditing policy:

bind tm global [-policyName <string> [-priority <positive_integer>]]

Example

```
> bind tm global -policyName Audit-Pol-1 -priority 1000
Done
```

To bind an auditing policy to an authentication virtual server by using the command line interface

At the command prompt, type the following commands to bind an auditing policy to an authentication virtual server and verify the configuration:

- o bind authentication vserver <name> [-policy <string> [-priority <positive_integer>] [-secondary] [-groupExtraction]]

- o show authentication vserver [<name>]

Example

```
> bind authentication Vserver Auth-Vserver-2 -policy Authn-Pol-1
Done
> show authentication Vserver Auth-Vserver-2
Auth-Vserver-2 (10.102.29.77:443) - SSL Type: CONTENT
State: UP
Client Idle Timeout: 180 sec
Down state flush: DISABLED
Disable Primary Vserver On Down : DISABLED
Authentication : ON
Current AAA Users: 0
Authentication Domain: myCompany.employee.com

1) Primary authentication policy name: Authn-Pol-1 Priority: 0
Done
```

To bind an auditing policy to a user account or a group by using the command line interface

At the command prompt, type one of the following commands to bind an auditing policy to a user account or a group:

- o bind audit <logtype> user <userName> -policy <policyname> [-priority <priority>]
- o bind audit <logtype> user <userName> -policy <policyname> [-priority <priority>]

Example

```
> bind audit nslogPolicy user aaa-user-1 -policyName Audit-Pol-1 -priority 1000
Done
```

To unbind a globally bound auditing policy by using the command line interface

At the command prompt, type the following commands to unbind a globally-bound auditing policy:
unbind audit <logtype> global -policy <policyname>

Example

```
> unbind audit nslogPolicy global -policy Audit-Pol-1
Done
```

To unbind an auditing policy from an authentication virtual server by using the command line interface

At the command prompt, type the following commands to unbind an auditing policy from an authentication virtual server:
unbind authentication vserver <name> [-policy <string> [-secondary][[-groupExtraction]]]

Example

```
> unbind authentication vserver auth-vserver-1 -policyName Audit-Pol-1
Done
```

To unbind an auditing policy from a user account or a group by using the command line interface

At the command prompt, type one of the following commands to unbind an auditing policy from a user account or a group:

- o unbind audit <logtype> user <userName> -policy <policyname>
- o unbind audit <logtype> group <groupName> -policy <policyname>

Example

```
> unbind audit nslogPolicy group aaa-group-1 -policyName Audit-Pol-1
Done
```

To remove an auditing policy by using the command line interface

First unbind the policy from all users and groups, and then, at the command prompt, type the following command to remove an auditing policy:

```
rm audit <logtype> <policyname>
```

To configure and bind auditing policies by using the configuration utility

1. Navigate to Security > AAA - Application Traffic > Policies > Auditing.
2. Choose the type of auditing policy that you want to create.
 - o To create a policy that logs to syslog, expand Syslog.
 - o To create a policy that logs to nslog, expand Nslog.

Note: The dialogs for the two types of policies are nearly identical.
3. In the details pane, do one of the following:
 - o To create a new auditing policy, click Add.
 - o To modify an existing auditing policy, select the policy, and then click Edit.
4. In the Create Audit Policy or Configure Audit Policy dialog, type or select values for the parameters.
 - o Name*â€”policyname (Cannot be changed for a previously configured policy.)
 - o Server*â€”action

Note: The Auditing type list box is read-only; you cannot change it.
5. Click Create or OK. The policy that you created appears in the Auditing Policies page.
6. Click OK.
7. To globally bind an auditing policy, in the details pane, click Global Bindings and fill in the Bind/Unbind Audit Policies to Global dialog box.
 - a. Select the name of the audit policy you want to globally bind.
 - b. Click OK.

A message appears in the status bar, stating that the policy has been configured successfully.
8. To bind an auditing policy, select Action > Global Bindings. You can bind an auditing policy to an authentication virtual server, a user account, or a group.

Session Settings

After you configure your authentication, authorization, and auditing profiles, you configure session settings to customize your user sessions. The session settings are:

The session timeout.

Controls the period after which the user is automatically disconnected and must authenticate again to access your intranet.

The default authorization setting.

Determines whether the NetScaler appliance will by default allow or deny access to content for which there is no specific authorization policy.

The single sign-on setting.

Determines whether the NetScaler appliance will log users onto all web applications automatically after they authenticate, or will pass users to the web application logon page to authenticate for each application.

The credential index setting.

Determines whether the NetScaler appliance will use primary or the secondary authentication credentials for single signon.

To configure the session settings, you can take one of two approaches. If you want different settings for different user accounts or groups, you create a profile for each user account or group for which you want to configure custom sessions settings. You also create policies to select the connections to which to apply particular profiles, and you bind the policies to users or groups. You can also bind a policy to the authentication virtual server that handles the traffic to which you want to apply the profile.

If you want the same settings for all sessions, or if you want to customize the default settings for sessions that do not have specific profiles and policies configured, you can simply configure the global session settings.

Session Profiles

To customize your user sessions, you first create a session profile. The session profile allows you to override global settings for any of the session parameters.

Note: The terms “session profile” and “session action” mean the same thing.

To create a session profile by using the command line interface

At the command prompt, type the following commands to create a session profile and verify the configuration:

- add tm sessionAction <name> [-sessTimeout <mins>] [-defaultAuthorizationAction (ALLOW | DENY)] [-SSO (ON | OFF)] [-ssoCredential (PRIMARY | SECONDARY)] [-ssoDomain <string>] [-httpOnlyCookie (YES | NO)] [-persistentCookie (ENABLED | DISABLED)] [-persistentCookieValidity <minutes>]
- show tm sessionAction <name>

Example

```
> add tm sessionAction session-profile -sessTimeout 30 -defaultAuthorization ALLOW
Done
> show tm sessionAction session-profile
1)      Name: session-profile
        Authorization action : ALLOW
        Session timeout: 30 minutes
Done
```

To modify a session profile by using the command line interface

At the command prompt, type the following commands to modify a session profile and verify the configuration:

- set tm sessionAction <name> [-sessTimeout <mins>] [-defaultAuthorizationAction (ALLOW | DENY)] [-SSO (ON | OFF)] [-ssoCredential (PRIMARY | SECONDARY)] [-ssoDomain <string>] [-httpOnlyCookie (YES | NO)] [-persistentCookie (ENABLED | DISABLED)] [-persistentCookieValidity <minutes>]
- show tm sessionAction

Example

```
> set tm sessionAction session-profile -sessTimeout 30 -defaultAuthorization ALLOW
Done
> show tm sessionAction session-profile
1)      Name: session-profile
        Authorization action : ALLOW
        Session timeout: 30 minutes
Done
```

To remove a session profile by using the command line interface

At the command prompt, type the following command to remove a session profile:

```
rm tm sessionAction <name>
```

To configure session profiles by using the configuration utility

1. Navigate to Security > AAA - Application Traffic > Session.
2. Navigate to Security > AAA - Application Traffic > Policies > Session.
3. In the details pane, click the Profiles tab.
4. On the Profiles tab, do one of the following:
 - To create a new session profile, click Add.
 - To modify an existing session profile, select the profile, and then click Edit.
5. In the Create TM Session Profile or Configure TM Session Profile dialog, type or select values for the parameters.
 - Name* “actionname (Cannot be changed for a previously configured session action.)
 - Session Time-out “sesstimeout
 - Default Authorization Action “defaultAuthorizationAction
 - Single Signon to Web Applications “sso
 - Credential Index “ssocredential
 - Single Sign-on Domain “ssoDomain

- o HTTPOnly Cookieâ€”httpOnlyCookie
 - o Enable Persistent Cookieâ€”persistentCookie
 - o Persistent Cookie Validityâ€”persistentCookieValidity
6. Click Create or OK. The session profile that you created appears in the Session Policies and Profiles pane.

Session Policies

After you create one or more session profiles, you create session policies and then bind the policies globally or to an authentication virtual server to put them into effect.

To create a session policy by using the command line interface

At the command prompt, type the following commands to create a session policy and verify the configuration:

- o add tm sessionPolicy <name> <rule> <action>
- o show tm sessionPolicy <name>

Example

```
> add tm sessionPolicy session-pol "URL == /*.gif" session-profile
Done
> show tm sessionPolicy session-pol
1)      Name: session-pol      Rule: URL == '/*.gif'
      Action: session-profile
Done
```

To modify a session policy by using the command line interface

At the command prompt, type the following commands to modify a session policy and verify the configuration:

- o set tm sessionPolicy <name> [-rule <expression>] [-action <action>]
- o show tm sessionPolicy <name>

Example

```
> set tm sessionPolicy session-pol "URL == /*.gif" session-profile
Done
> show tm sessionPolicy session-pol
1)      Name: session-pol      Rule: URL == '/*.gif'
      Action: session-profile
Done
```

To globally bind a session policy by using the command line interface

At the command prompt, type the following commands to globally bind a session policy and verify the configuration:
bind tm global -policyName <policyname> [-priority <priority>]

Example

```
> bind tm global -policyName session-pol
Done

> show tm sessionPolicy session-pol
1)      Name: session-pol      Rule: URL == '/*.gif'
      Action: session-profile
      Policy is bound to following entities
      1) TM GLOBAL      PRIORITY : 0
Done
```

To bind a session policy to an authentication virtual server by using the command line interface

At the command prompt, type the following command to bind a session policy to an authentication virtual and verify the configuration:

bind authentication vserver <name> -policy <policyname> [-priority <priority>]

Example

```
> bind authentication vserver auth-vserver-1 -policyName Session-Pol-1 -priority 1000
Done
```

To unbind a session policy from an authentication virtual server by using the command line interface

At the command prompt, type the following commands to unbind a session policy from an authentication virtual server and verify the configuration:

```
unbind authentication vserver <name> -policy <policyname>
```

Example

```
> unbind authentication vserver auth-vserver-1 -policyName Session-Pol-1
Done
```

To unbind a globally bound session policy by using the command line interface

At the command prompt, type the following commands to unbind a globally-bound session policy:

```
unbind tm global -policyName <policyname>
```

Example

```
> unbind tm global -policyName Session-Pol-1
Done
```

To remove a session policy by using the command line interface

First unbind the session policy from global, and then, at the command prompt, type the following commands to remove a session policy and verify the configuration:

```
rm tm sessionPolicy <name>
```

Example

```
> rm tm sessionPolicy Session-Pol-1
Done
```

To configure and bind session policies by using the configuration utility

1. Navigate to Security > AAA - Application Traffic > Session.
2. Navigate to Security > AAA - Application Traffic > Policies > Session.
3. In the details pane, on the Policies tab, do one of the following:
 - o To create a new session policy, click Add.
 - o To modify an existing session policy, select the policy, and then click Edit.
4. In the Create Session Policy or Configure Session Policy dialog, type or select values for the parameters.
 - o Name*â€™policyname (Cannot be changed for a previously configured session policy.)
 - o Request Profile*â€™actionname
 - o Expression*â€™rule (You enter expressions by first choosing the type of expression in the leftmost drop-down list beneath the Expression text area and then typing your expression directly into the expression text area, or by clicking Add to open the Add Expression dialog box and using the drop-down lists in it to construct your expression.)
5. Click Create or OK. The policy that you created appears in the details pane of the Session Policies and Profiles page.
6. To globally bind a session policy, in the details pane, select Global Bindings from the Action drop-down list, and fill in the dialog.
 - a. Select the name of the session policy you want to globally bind.
 - b. Click OK.
7. To bind a session policy to an authentication virtual server, in the navigation pane, click Virtual Servers, and add that policy to the policies list.
 - a. In the details pane, select the virtual server, and then click Edit.
 - b. In the Advanced selections to the right of the detail area, click Policies.
 - c. Select a policy, or click the plus icon to add a policy.
 - d. In the Priority column to the left, modify the default priority as needed to ensure that the policy is evaluated in the proper order.
 - e. Click OK.

A message appears in the status bar, stating that the policy has been configured successfully.

Global Session Settings

In addition to or instead of creating session profiles and policies, you can configure global session settings. These settings control the session configuration when there is no explicit policy overriding them.

To configure the session settings by using the command line interface

At the command prompt, type the following commands to configure the global session settings and verify the configuration:

```
set tm sessionParameter [-sessTimeout <mins>] [-defaultAuthorizationAction ( ALLOW | DENY )] [-SSO ( ON | OFF )] [-ssoCredential ( PRIMARY | SECONDARY )] [-ssoDomain <string>] [-httpOnlyCookie ( YES | NO )] [-persistentCookie ( ENABLED | DISABLED )] [-persistentCookieValidity <minutes>]
```

Example

```
> set tm sessionParameter -sessTimeout 30
Done
> set tm sessionParameter -defaultAuthorizationAction DENY
Done
> set tm sessionParameter -SSO ON
Done
> set tm sessionParameter -ssoCredential PRIMARY
Done
```

To configure the session settings by using the configuration utility

1. Navigate to Security > AAA - Application Traffic
2. In the details pane, under Settings, click Change global settings.
3. In the Global Session Settings dialog, type or select values for the parameters.
 - o Session Time-outâ€”sessTimeout
 - o Default Authorization Actionâ€”defaultAuthorizationAction
 - o Single Sign-on to Web Applicationsâ€”sso
 - o Credential Indexâ€”ssoCredential
 - o Single Sign-on Domainâ€”ssoDomain
 - o HTTPOnly Cookieâ€”httpOnlyCookie
 - o Enable Persistent Cookieâ€”persistentCookie
 - o Persistent Cookie Validity (minutes)â€”persistentCookieValidity
 - o Home Pageâ€”homepage
4. Click OK.

Traffic Settings

If you use forms-based or SAML single sign-on (SSO) for your protected applications, you configure that feature in the Traffic settings. SSO enables your users to log on once to access all protected applications, rather than requiring them to log on separately to access each one.

Forms-based SSO allows you to use a web form of your own design as the sign-on method instead of a generic pop-up window. You can therefore put your company logo and other information you might want your users to see on the logon form. SAML SSO allows you to configure one NetScaler appliance or virtual appliance instance to authenticate to another NetScaler appliance on behalf of users who have authenticated with the first appliance.

To configure either type of SSO, you first create a forms or SAML SSO profile. Next, you create a traffic profile and link it to the SSO profile you created. Next, you create a policy, link it to the traffic profile. Finally, you bind the policy globally or to an authentication virtual server to put your configuration into effect.

Traffic Profiles

After creating at least one forms or SAML sso profile, you must next create a traffic profile.

Note: In this feature, the terms "profile" and "action" mean the same thing.

To create a traffic profile by using the command line interface

At the command prompt, type:

```
add tm trafficAction <name> [-appTimeout <mins>] [-SSO ( ON | OFF )] [-formSSOAction <string>]] [-persistentCookie (
ENABLED | DISABLED )] [-InitiateLogout ( ON | OFF )]
```

Example

```
add tm trafficAction Traffic-Prof-1 "appTimeout 10 -SSO ON -formSSOAction SSO-Prof-1"
```

To modify a session profile by using the command line interface

At the command prompt, type:

```
set tm trafficAction <name> [-appTimeout <mins>] [-SSO ( ON | OFF )] [-formSSOAction <string>]] [-persistentCookie (
ENABLED | DISABLED )] [-InitiateLogout ( ON | OFF )]
```

Example

```
set tm trafficAction Traffic-Prof-1 "appTimeout 10 -SSO ON -formSSOAction SSO-Prof-1"
```

To remove a session profile by using the command line interface

At the command prompt, type:

```
rm tm trafficAction <name>
```

Example

```
rm tm trafficAction Traffic-Prof-1
```

To configure traffic profiles by using the configuration utility

1. Navigate to Security > AAA - Application Traffic > Traffic.
2. Navigate to Security > AAA - Application Traffic > Policies > Traffic.
3. In the details pane, click the Profiles tab.
4. On the Profiles tab, do one of the following:
 - o To create a new traffic profile, click Add.
 - o To modify an existing traffic profile, select the profile, and then click Edit.
5. In the Create Traffic Profile or Configure Traffic Profile dialog box, specify values for the parameters.
 - o Name "name (Cannot be changed for a previously configured session action.)
 - o AppTimeout "appTimeout
 - o Single Sign-On "SSO
 - o Form SSO Action "formSSOAction
 - o SAML SSO Action "samlSSOAction
 - o Enable Persistent Cookie "persistentCookie
 - o Initiate Logout "InitiateLogout
6. Click Create or OK. The traffic profile that you created appears in the Traffic Policies, Profiles, and either the Form SSO Profiles or SAML SSO Profiles pane, as appropriate.

Traffic Policies

After you create one or more form SSO and traffic profiles, you create traffic policies and then bind the policies, either globally or to a traffic management virtual server, to put them into effect.

To create a traffic policy by using the command line interface

At the command prompt, type:

```
add tm trafficPolicy <name> <rule> <action>
```

Example

```
add tm trafficPolicy Traffic-Pol-1 "HTTP.REQ.HEADER("Cookie").CONTAINS("login=true)" Traffic-
```

To modify a traffic policy by using the command line interface

At the command prompt, type:

```
set tm trafficPolicy <name> <rule> <action>
```

Example

```
set tm trafficPolicy Traffic-Pol-1 "HTTP.REQ.HEADER("Cookie").CONTAINS("login=true)" Traffic-
```

To globally bind a traffic policy by using the command line interface

At the command prompt, type:

```
bind tm global -policyName <string> [-priority <priority>]
```

Example

```
bind tm global -policyName Traffic-Pol-1
```

To bind a traffic policy to a load balancing or content switching virtual server by using the command line interface

At the command prompt, type one of the following commands:

- `bind lb vserver <name> -policy <policyName> [-priority <priority>]`
- `bind cs vserver <name> -policy <policyName> [-priority <priority>]`

Example

```
bind authentication vserver auth-vserver-1 -policyName Traffic-Pol-1 -priority 1000
```

To unbind a globally bound traffic policy by using the command line interface

At the command prompt, type:

```
unbind tm global -policyName <policyname>
```

Example

```
unbind tm global -policyName Traffic-Pol-1
```

To unbind a traffic policy from a load balancing or content switching virtual server by using the command line interface

At the command prompt, type one of the following commands:

- `unbind lb vserver <name> -policy <policyname>`
- `unbind cs vserver <name> -policy <policyname>`

Example


```
unbind authentication vserver auth-vserver-1 -policyName Traffic-Pol-1
```

To remove a traffic policy by using the command line interface

First unbind the session policy from global, and then, at the command prompt, type:

```
rm tm trafficPolicy <name>
```

Example

```
rm tm trafficPolicy Traffic-Pol-1
```

To configure and bind traffic policies by using the configuration utility

1. Navigate to Security > AAA - Application Traffic > Traffic.
2. Navigate to Security > AAA - Application Traffic > Policies > Traffic.
3. In the details pane, do one of the following:
 - o To create a new session policy, click Add.
 - o To modify an existing session policy, select the policy, and then click Edit.
4. In the Create Traffic Policy or Configure Traffic Policy dialog, specify values for the parameters.
 - o Name*â€”policyName (Cannot be changed for a previously configured session policy.)
 - o Profile*â€”actionName
 - o Expression*â€”rule (You enter expressions by first choosing the type of expression in the leftmost drop-down list beneath the Expression text area and then typing your expression directly into the expression text area, or by clicking Add to open the Add Expression dialog box and using the drop-down lists in it to construct your expression.)
5. Click Create or OK. The policy that you created appears in the details pane of the Session Policies and Profiles page.

Form SSO Profiles

To enable and configure forms-based SSO, you first create an SSO profile.

Note:

- Forms-based single sign-on does not work if the form is customized to include Javascript.
- In this feature, the terms "profile" and "action" mean the same thing.

To create a form SSO profile by using the command line interface

At the command prompt, type:

- add tm formSSOAction <name> -actionURL <URL> -userField <string> -passwdField <string> -ssoSuccessRule <expression> [-nameValuePair <string>] [-responsesize <positive_integer>] [-nvtype (STATIC | DYNAMIC)] [-submitMethod (GET | POST)]
- show tm formSSOAction [<name>]

Example

```
add tm formSSOAction SSO-Prof-1 -actionURL "/logon.php"
-userField "loginID" -passwdField "passwd"
-nameValuePair "loginID passwd" -responsesize "9096"
-ssoSuccessRule "HTTP.RES.HEADER("Set-Cookie").CONTAINS("LogonID")"
-nvtype STATIC -submitMethod GET
&quot;sessTimeout 10 -defaultAuthorizationAction ALLOW
```

To modify a form SSO by using the command line interface

At the command prompt, type:

```
set tm formSSOAction <name> -actionURL <URL> -userField <string> -passwdField <string> -ssoSuccessRule
<expression> [-nameValuePair <string>] [-responsesize <positive_integer>] [-nvtype ( STATIC | DYNAMIC )] [-submitMethod
( GET | POST )]
```

Example

```
set tm formSSOAction SSO-Prof-1 -actionURL "/logon.php"
-userField "loginID" -passwdField "passwd"
-ssoSuccessRule "HTTP.RES.HEADER("Set-Cookie").CONTAINS("LogonID")"
-nameValuePair "loginID passwd" -responsesize "9096"
-nvtype STATIC -submitMethod GET
&quot;sessTimeout 10 -defaultAuthorizationAction ALLOW
```

To remove a form SSO profile by using the command line interface

At the command prompt, type:

```
rm tm formSSOAction <name>
```

Example

```
rm tm sessionAction SSO-Prof-1
```

To configure form SSO profiles by using the configuration utility

- Navigate to Security > AAA - Application Traffic > Policies > Traffic.
- In the details pane, click the Form SSO Profiles tab.
- On the Form SSO Profiles tab, do one of the following:
 - To create a new form SSO profile, click Add.
 - To modify an existing form SSO profile, select the profile, and then click Edit.
- In the Create Form SSO Profile or Configure Form SSO Profile dialog, specify values for the parameters:
 - Name* "name (Cannot be changed for a previously configured session action.)
 - Action URL* "actionURL
 - User Name Field* "userField
 - Password Field* "passField
 - Expression* "ssoSuccessRule
 - Name Value Pair* "nameValuePair

- Response Sizeâ€™responsesize
 - Extractionâ€™nvtype
 - Submit Methodâ€™submitMethod
5. Click Create or OK, and then click Close. The form SSO profile that you created appears in the Traffic Policies, Profile , and Form SSO Profiles pane.

SAML SSO Profiles

To enable and configure SAML-based SSO, you first create a SAML SSO profile.

To create a SAML SSO profile by using the command line interface

At the command prompt, type:

```
add tm samlSSOProfile <name> -samlSigningCertName <string> -assertionConsumerServiceURL <URL> -relaystateRule <expression> -sendPassword (ON|OFF) [-samlIssuerName <string>]
```

Example

```
add tm samlSSOProfile saml-SSO-Prof-1 -samlSigningCertName "Example, Inc." -assertionConsum
```

To modify a SAML SSO by using the command line interface

At the command prompt, type:

```
set tm samlSSOProfile <name> -samlSigningCertName <string> -assertionConsumerServiceURL <URL> -relaystateRule <expression> -sendPassword (ON|OFF) [-samlIssuerName <string>]
```

Example

```
set tm samlSSOProfile saml-SSO-Prof-1 -samlSigningCertName "Example, Inc." -assertionConsum
```

To remove a SAML SSO profile by using the command line interface

At the command prompt, type:

```
rm tm samlSSOProfile <name>
```

Example

```
rm tm sessionAction saml-SSO-Prof-1
```

To configure a SAML SSO profile by using the configuration utility

1. Navigate to Security > AAA - Application Traffic > Policies > Traffic.
2. In the details pane, click the SAML SSO Profiles tab.
3. On the SAML SSO Profiles tab, do one of the following:
 - o To create a new SAML SSO profile, click Add.
 - o To modify an existing SAML SSO profile, select the profile, and then click OpenEdit.
4. In the Create SAML SSO Profiles or the Configure SAML SSO Profiles dialog box, set the following parameters:
 - o Name*
 - o Signing Certificate Name*
 - o ACS URL*
 - o Relay State Rule*
 - o Send Password
 - o Issuer Name
5. Click Create or OK, and then click Close. The SAML SSO profile that you created appears in the Traffic Policies, Profiles, and SAML SSO Profiles pane.

Session Timeout for OWA 2010

You can now force OWA 2010 connections to time out after a specified period of inactivity. OWA sends repeated keepalive requests to the server to prevent timeouts. Keeping the connections open can interfere with single sign-on.

To force OWA 2010 to timeout after a specified period by using the command line interface

At the command prompt, type the following commands:

- `add tm trafficAction <actname> [-forcedTimeout <forcedTimeout> -forcedTimeoutVal <mins>]`
For <actname>, substitute a name for your traffic policy. For <mins>, substitute the number of minutes after which to initiate a forced timeout. For <forcedTimeout>, substitute one of the following values:
STARTâ€”Starts the timer for forced timeout if a timer has not already been started. If a running timer exists, has no effect.
STOPâ€”Stops a running timer. If no running timer is found, has no effect.
RESETâ€”Restarts a running timer. If no running timer is found, starts a timer just as if the START option had been used.
- `add tm trafficPolicy <polname> <rule> <actname>`
For <polname>, substitute a name for your traffic policy. For <rule>, substitute a rule in NetScaler default syntax.
- `bind lb vserver <vservname> â€”policyName <name> -priority <number>`
For <vservname>, substitute the name of the AAA traffic management virtual server. For <priority>, substitute an integer that designates the policy's priority.

Example

```
add tm trafficAction act-owa2010timeout -forcedTimeout RESET -forcedTimeoutVal 10
add tm trafficPolicy pol-owa2010timeout true act-owa2010timeout
bind lb vserver vs-owa2010 -policyName pol-owa2010timeout -priority 10
```

Authenticating with Client Certificates

Web sites that contain sensitive content, such as online banking websites or websites with employee personal information, sometimes require client certificates for authentication. To configure AAA to authenticate users on the basis of client-side certificate attributes, you first enable client authentication on the traffic management virtual server and bind the root certificate to the authentication virtual server. Then, you implement one of two options. You can configure the default authentication type on the authentication virtual server as CERT, or you can create a certificate action that defines what the NetScaler ADC must do to authenticate users on the basis of a client certificate. In either case, your authentication server must support CRLs. You configure the ADC to extract the user name from the SubjectCN field or another specified field in the client certificate.

When the user tries to log on to an authentication virtual server for which an authentication policy is not configured, and a global cascade is not configured, the user name information is extracted from the specified field of the certificate. If the required field is extracted, the authentication succeeds. If the user does not provide a valid certificate during the SSL handshake, or if the user name extraction fails, authentication fails. After it validates the client certificate, the ADC presents a logon page to the user.

The following procedures assume that you have already created a functioning AAA configuration, and therefore they explain only how to enable authentication by using client certificates. These procedures also assume that you have obtained your root certificate and client certificates and have placed them on the ADC in the /nsconfig/ssl directory.

To configure the AAA client certificate parameters by using the command line interface

At the command prompt, type the following commands, in the order shown, to configure the certificate and verify the configuration:

- o add ssl certKey <certkeyName> -cert <certFile> -key <keyFile> -password -inform <inform> -expiryMonitor <expiryMonitor> -notificationPeriod <notificationPeriod>
- o bind ssl certKey <certkeyName> -vServer <certkeyName> -CA -crlCheck Mandatory
- o show ssl certKey [<certkeyName>]
- o set aaa parameter -defaultAuthType CERT
- o show aaa parameter
- o set aaa certParams -userNameField "Subject:CN"
- o show aaa certParams

To configure the AAA client certificate parameters by using the configuration utility

1. Navigate to Security > AAA - Application Traffic > Virtual Servers.
2. In the details pane, select the virtual server that you want to configure to handle client certificate authentication, and then click Edit.
3. On the Configuration page, under Certificates, click the right arrow (>) to open the CA Cert Key installation dialog.
4. In the CA Cert Key dialog box, click Insert.
5. In the CA Cert Key - SSL Certificates dialog box, click Install.
6. In the Install Certificate dialog box, set the following parameters, whose names correspond to the CLI parameter names as shown:
 - o Certificate-Key Pair Name*â€™certkeyName
 - o Certificate File Nameâ€™certFile
 - o Key File Nameâ€™keyFile
 - o Certificate Formatâ€™inform
 - o Passwordâ€™password
 - o Certificate Bundleâ€™bundle
 - o Notify When Expiresâ€™expiryMonitor
 - o Notification Periodâ€™notificationPeriod
7. Click Install, and then click Close.
8. In the CA Cert Key dialog box, in the Certificate list, select the root certificate.
9. Click Save.
10. Click Back to return to the main configuration screen.
11. Navigate to Security > AAA - Application Traffic > Policies > Authentication > CERT.
12. In the details pane, select the policy you want to configure to handle client certificate authentication, and then click Edit.
13. In the Configure Authentication CERT Policy dialog, Server drop-down list, select the virtual server you just configured to handle client certificate authentication.
14. Click OK. A message appears in the status bar, stating that the configuration completed successfully.

Client Certificate Pass-Through

The NetScaler ADC can now be configured to pass client certificates through to protected applications that require client certificates for user authentication. The ADC first authenticates the user, then inserts the client certificate into the request and sends it to the application. This feature is configured by adding appropriate SSL policies.

The exact behavior of this feature when a user presents a client certificate depends upon the configuration of the VPN virtual server.

- If the VPN virtual server is configured to accept client certificates but not require them, the ADC inserts the certificate into the request and then forwards the request to the protected application.
- If the VPN virtual server has client certificate authentication disabled, the ADC renegotiates the authentication protocol and reauthenticates the user before it inserts the client certificate in the header and forwards the request to the protected application.
- If the VPN virtual server is configured to require client certificate authentication, the ADC uses the client certificate to authenticate the user, then inserts the certificate in the header and forwards the request to the protected application.

In all of these cases, you configure client certificate pass-through as follows.

To create and configure client certificate pass-through by using the command line interface

At the command prompt, type the following commands:

- `add vpn vserver <name> SSL <IP> 443`
For <name>, substitute a name for the virtual server. The name must contain from one to 127 ASCII characters, beginning with a letter or underscore (_), and containing only letters, numbers, and the underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. For <IP>, substitute the IP address assigned to the virtual server.
- `set ssl vserver <name> -clientAuth ENABLED -clientCert <clientcert>`
For <name>, substitute the name of the virtual server that you just created. For <clientCert>, substitute one of the following values:
 - disabled – disables client certificate authentication on the VPN virtual server.
 - mandatory – configures the VPN virtual server to require client certificates to authenticate.
 - optional – configures the VPN virtual server to allow client certificate authentication, but not to require it.
- `bind vpn vserver <name> -policy local`
For <name>, substitute the name of the VPN virtual server that you created.
- `bind vpn vserver <name> -policy cert`
For <name>, substitute the name of the VPN virtual server that you created.
- `bind ssl vserver <name> -certkeyName <certkeyname>`
For <name>, substitute the name of the virtual server that you created. For <certkeyName>, substitute the client certificate key.
- `bind ssl vserver <name> -certkeyName <cacertkeyname> -CA -ocspCheck Optional`
For <name>, substitute the name of the virtual server that you created. For <cacertkeyName>, substitute the CA certificate key.
- `add ssl action <actname> -clientCert ENABLED -certHeader CLIENT-CERT`
For <actname>, substitute a name for the SSL action.
- `add ssl policy <polname> -rule true -action <actname>`
For <polname>, substitute a name for your new SSL policy. For <actname>, substitute the name of the SSL action that you just created.
- `bind ssl vserver <name> -policyName <polname> -priority 10`
For <name>, substitute the name of the VPN virtual server.

Example

```
add vpn vserver vs-certpassthru SSL 10.121.250.75 443
set ssl vserver vs-certpassthru -clientAuth ENABLED -clientCert optional
bind vpn vserver vs-certpassthru -policy local
bind vpn vserver vs-certpassthru -policy cert
bind ssl vserver vs-certpassthru -certkeyName mycertKey
bind ssl vserver vs-certpassthru -certkeyName mycertKey -CA -ocspCheck Optional
add ssl action act-certpassthru -clientCert ENABLED -certHeader CLIENT-CERT
```

```
add ssl policy pol-certpassthru -rule true -action act-certpassthru
bind ssl vserver vs-certpassthru -policyName pol-certpassthru -priority 10
```


Configuring AAA with Commonly Used Protocols

Configuring the NetScaler for Authentication, Authorization, and Auditing (AAA) needs a specific setup on the NetScaler and clients' browsers. The configuration varies with the protocol used for AAA.

For more information about configuring the NetScaler for Kerberos authentication, see [Handling Authentication, Authorization and Auditing with Kerberos/NTLM](#).

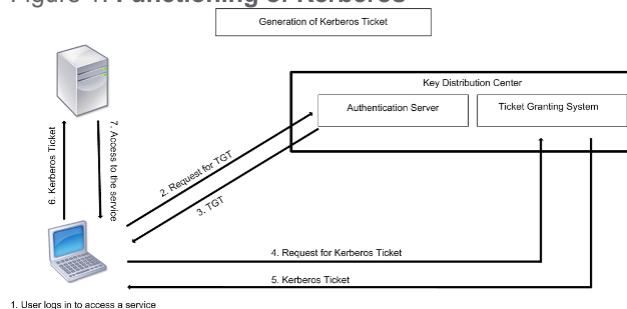
Handling Authentication, Authorization and Auditing with Kerberos/NTLM

Kerberos, a computer network authentication protocol, provides secure communication over the Internet. Designed primarily for client-server applications, it provides for mutual authentication by which the client and server can each ensure the other's authenticity. Kerberos uses a trusted third party, referred to as Key Distribution Center (KDC). A KDC consists of an Authentication Server (AS), which authenticates a user, and a Ticket Granting Server (TGS).

Each entity on the network (client or server) has a secret key that is known only to itself and the KDC. The knowledge of this key implies authenticity of the entity. For communication between two entities on the network, the KDC generates a session key, referred to as the Kerberos ticket or service ticket. The client makes a request to the AS for credentials for a specific server. The client then receives a ticket, referred to as Ticket Granting Ticket (TGT). The client then contacts the TGS, using the TGT it received from the AS to prove its identity, and asks for a service. If the client is eligible for the service, the TGS issues a Kerberos ticket to the client. The client then contacts the server hosting the service (referred to as the service server), using the Kerberos ticket to prove that it is authorized to receive the service. The Kerberos ticket has a configurable lifetime. The client authenticates itself with the AS only once. If it contacts the physical server multiple times, it reuses the AS ticket.

The following figure shows the basic functioning of the Kerberos protocol.

Figure 1. **Functioning of Kerberos**



Kerberos authentication has the following advantages:

- Faster authentication. When a physical server gets a Kerberos ticket from a client, the server has enough information to authenticate the client directly. It does not have to contact a domain controller for client authentication, and therefore the authentication process is faster.
- Mutual authentication. When the KDC issues a Kerberos ticket to a client and the client uses the ticket to access a service, only authenticated servers can decrypt the Kerberos ticket. If the virtual server on the NetScaler is able to decrypt the Kerberos ticket, you can conclude that both the virtual server and client are authenticated. Thus, the authentication of the server happens along with the authentication of the client.
- Single sign-on between Windows and other operating systems that support Kerberos.

Kerberos authentication may have the following disadvantages:

- Kerberos has strict time requirements; the clocks of the involved hosts must be synchronized with the Kerberos server clock to ensure that the authentication does not fail. You can mitigate this disadvantage by using the Network Time Protocol daemons to keep the host clocks synchronized. Kerberos tickets have an availability period, which you can configure.
- Kerberos needs the central server to be available continuously. When the Kerberos server is down, no one can log on. You can mitigate this risk by using multiple Kerberos servers and fallback authentication mechanisms.
- Because all the authentication is controlled by a centralized KDC, any compromise in this infrastructure, such as the user's password for a local workstation being stolen, can allow an attacker to impersonate any user. You can mitigate this risk to some extent by using only a desktop machine or laptop that you trust, or by enforcing preauthentication by means of a hardware-token.

To use Kerberos authentication, you must configure it on the NetScaler appliance and on each client.

How NetScaler Implements Kerberos Authentication

Note: Kerberos/NTLM authentication is supported only in the NetScaler 9.3 nCore release or later, and it can be used only for AAA traffic management (AAA-TM) virtual servers.

NetScaler handles the components involved in Kerberos authentication in the following way:

Key Distribution Center (KDC)

In the Windows 2000 Server or later versions, the Domain Controller and KDC are part of the Windows Server. If the Windows Server is UP and running, it indicates that the Domain Controller and KDC are configured. The KDC is also the Active Directory server.

Note: All Kerberos interactions are validated with the Windows Kerberos Domain Controller.

Authentication Service and Protocol Negotiation

A NetScaler appliance supports Kerberos authentication on the AAA-TM authentication virtual servers. If the Kerberos authentication fails, the NetScaler uses the NTLM authentication.

By default, Windows 2000 Server and later Windows Server versions use Kerberos for AAA. If you create an authentication policy with NEGOTIATE as the authentication type, the NetScaler attempts to use the Kerberos protocol for AAA and if the client's browser fails to receive a Kerberos ticket, the NetScaler uses the NTLM authentication. This process is referred to as negotiation.

The client may fail to receive a Kerberos ticket in any of the following cases:

- Kerberos is not supported on the client.
- Kerberos is not enabled on the client.
- The client is in a domain other than that of the KDC.
- The Access Directory on the KDC is not accessible to the client.

For Kerberos/NTLM authentication, the NetScaler does not use the data that is present locally on the NetScaler appliance.

Authorization

The traffic management virtual server can be a load balancing virtual server or a content switching virtual server.

Auditing

The NetScaler appliance supports auditing of Kerberos authentication with the following audit logging:

- Complete audit trail of the traffic management end-user activity
- SYSLOG and high performance TCP logging
- Complete audit trail of system administrators
- All system events
- Scriptable log format

Supported Environment

Kerberos authentication does not need any specific environment on the NetScaler. The client (browser) must provide support for Kerberos authentication.

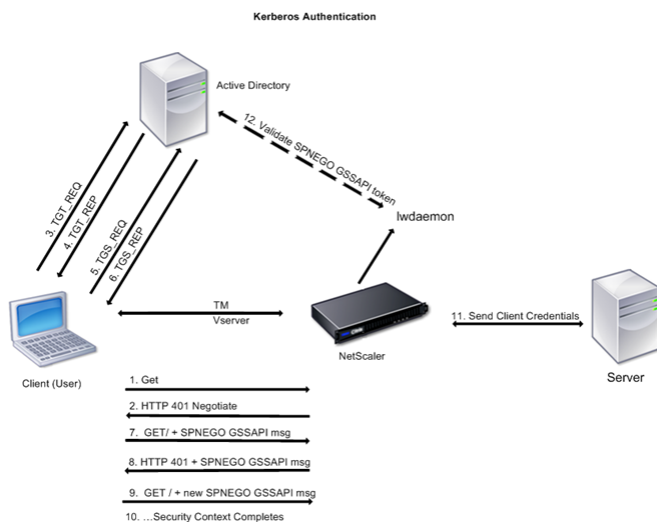
High Availability

In a high availability setup, only the active NetScaler joins the domain. In case of a failover, the NetScaler lwagent daemon joins the secondary NetScaler appliance to the domain. No specific configuration is required for this functionality.

Kerberos Authentication Process

The following figure shows a typical process for Kerberos authentication in the NetScaler environment.

Figure 1. Kerberos Authentication Process on NetScaler



The Kerberos authentication occurs in the following stages:

Client authenticates itself to the KDC.

1. The NetScaler appliance receives a request from a client.
2. The traffic management (load balancing or content switching) virtual server on the NetScaler sends a challenge to the client.
3. To respond to the challenge, the client gets a Kerberos ticket.
 - o The client sends the Authentication Server of the KDC a request for a ticket-granting ticket (TGT) and receives the TGT. (See 3, 4 in the figure, Kerberos Authentication Process.)
 - o The client sends the TGT to the Ticket Granting Server of the KDC and receives a Kerberos ticket. (See 5, 6 in the figure, Kerberos Authentication Process.)

Note: The above authentication process is not necessary if the client already has a Kerberos ticket whose lifetime has not expired. In addition, clients such as Web Services, .NET, or J2EE, which support SPNEGO, get a Kerberos ticket for the target server, create an SPNEGO token, and insert the token in the HTTP header when they send an HTTP request. They do not go through the client authentication process.

Client requests a service.

1. The client sends the Kerberos ticket containing the SPNEGO token and the HTTP request to the traffic management virtual server on the NetScaler. The SPNEGO token has the necessary GSSAPI data.
2. The NetScaler establishes a security context between the client and the NetScaler. If the NetScaler cannot accept the data provided in the Kerberos ticket, the client is asked to get a different ticket. This cycle repeats till the GSSAPI data is acceptable and the security context is established. The traffic management virtual server on the NetScaler acts as an HTTP proxy between the client and the physical server.

NetScaler completes the authentication.

1. After the security context is complete, the traffic management virtual server validates the SPNEGO token.
2. From the valid SPNEGO token, the virtual server extracts the user ID and GSS credentials, and passes them to the authentication daemon.
3. A successful authentication completes the Kerberos authentication.

Kerberos Authentication - Configuration on the NetScaler Appliance

To configure Kerberos authentication on the NetScaler appliance, perform the following tasks:

1. Enable the Authentication, Authorization, and Auditing (AAA) feature on the NetScaler appliance.
2. On the Active Directory, add a user for Kerberos authentication, map the HTTP service to this user, and generate a keytab file and import it to the NetScaler appliance. You can map more than one service if the Kerberos authentication is required for more than one service. The keytab file should contain entries for every service that is bound to the traffic management virtual server on the NetScaler. The keytab file is necessary for decrypting the secret received from the client during Kerberos authentication. The authentication details of all the services are stored in a single keytab file on the NetScaler.
3. Add a DNS server.
Note: The NetScaler must obtain the IP address of the domain controller from the fully qualified domain name (FQDN). Therefore, Citrix recommends configuring the NetScaler with a DNS server. A less preferred alternative is to create a static DNS entry.
4. Create an authentication negotiation policy with a negotiation action.
5. Configure an authentication server and bind the authentication policy to the authentication virtual server.
6. Configure an authentication service and a traffic management virtual server, and bind the service to the virtual server. You can use either a load balancing or a content switching virtual server.
7. Verify the configuration.

Enabling AAA on the NetScaler

Enable authentication of the traffic on the NetScaler appliance.

To enable Authentication, Authorization, and Auditing (AAA) by using the command line interface

At the command prompt, type the following commands to enable AAA and verify the configuration:

- enable ns feature AAA
- show ns feature

Example

```
> enable feature aaa
Done
> show ns feature
Feature Acronym Status
-----
1) Web Logging WL ON
   â€¦
3) Load Balancing LB ON
4) Content Switching CS ON
5) Cache Redirection CR ON
   â€¦
14) SSL VPN SSLVPN ON
15) AAA AAA ON
   â€¦
26) CloudBridge CloudBridge OFF
Done
```

To enable Authentication, Authorization, and Auditing (AAA) on NetScaler by using the configuration utility

1. In the navigation pane, expand System, and then click Settings.
2. In the details pane, under Modes and Features, click Configure basic features.
3. In the Configure Basic Features dialog box, select the Authentication, Authorization and Auditing check box.
4. Click OK.
5. In the confirmation dialog box, click Yes. A message appears in the status bar to indicate that the feature is enabled.

Adding a Keytab file

The keytab file contains information about services necessary for Kerberos authentication. The keytab file is necessary for decrypting the secret received from the client during Kerberos authentication. You can map more than one service if the Kerberos authentication is required for more than one service. The keytab file should contain entries for every service that is bound to the traffic management virtual server on the NetScaler. The authentication details of all the services are stored in a single keytab file on the NetScaler.

To generate a keytab file and import it to the NetScaler appliance, follow the procedure described below:

Note: You can generate the keytab file and import it onto the NetScaler only from the command line.

1. Log onto the Active Directory server and create a user for Kerberos authentication.

For example, type the following command:

```
net user Kerb-SVC-Account freebsd!@#456 /add
```

2. In the User Properties section, ensure the following settings:
 - The Change password at next logon option is not selected.
 - The Password does not expire option is selected.
3. Map the HTTP service to the above user and export the keytab file. For example, run the following command on the Active Directory server:

```
ktutil /out keytabfile /princ HTTP/owa.newacp.com@NEWACP.COM /pass freebsd!@#456  
/mapuser newacp\dummy /ptype KRB5_NT_PRINCIPAL
```

Note: If you want to map more services, repeat the above command for every service. You can give the same name or different names for the output file.

4. Transfer the keytab file to the NetScaler by using the unix ftp command or any other file transfer utility of your choice.
5. Log onto the NetScaler appliance, and run the ktutil utility to verify the keytab file. The keytab file has an entry for the HTTP service after it is imported.

Example

```
root@ns# ktutil  
ktutil: rkt /var/keytabfile  
ktutil: list  
slot KVNO Principal  
-----  
  
ktutil: wkt /etc/ krb5.keytab  
ktutil: list  
slot KVNO Principal  
-----  
1 2 HTTP/owa.newacp.com@NEWACP.COM  
ktutil: quit
```

Adding a DNS Server

The NetScaler appliance should obtain the IP address of the domain controller from the fully qualified domain name (FQDN). Therefore, Citrix recommends configuring the NetScaler with a DNS server. A less preferred alternative is to create a static DNS entry.

To add a DNS server by using the command line interface

At the command prompt, type the following command:

```
add dns nameserver <IP>
```

Note: Alternatively, you can add static host entries or use any other means so that the NetScaler can resolve the FQDN name of the domain controller to an IP address.

Example

```
add dns nameserver 1.2.3.4
```

To add a DNS server by using the NetScaler configuration utility

1. Navigate to Traffic Management > DNS > Name Servers.
2. In the details pane, click Add.
3. In the IP Address box, type the IP address.
4. Click Create, and then Close.
5. Verify that the details pane shows the newly added DNS server.

Creating an Authentication Negotiation Policy

Create a negotiation policy with a negotiation action for Kerberos authentication of services.

To create an authentication negotiation policy by using the command line interface

At the command prompt, type the following commands:

- `add authentication negotiateAction <name> -domain <domainName> -domainUser <domainUsername> -domainUserPasswd <domainUserPassword> -encrypted`
- `add authentication negotiatePolicy <name> <rule> <reqAction>`

Example

```
add authentication negotiateAction negact -domain newacp.com -domainUser Administrator -domainUserPasswd <password> -encrypted
add authentication negotiatePolicy negopol ns_true negact
```

To create an authentication negotiation policy by using the NetScaler configuration utility

1. Navigate to Security > AAA-Application Traffic > Policies > Authentication.
2. In the details pane, on the Policies tab, click Add.
3. In the Create Authentication Policy dialog box, set the following parameters:
 - Name
 - Authentication Type - Select NEGOTIATE.
 - Server - Select an existing server from the dropdown list. To add a new authentication server, click New, and in the Create Authentication Server dialog box, set the following parameters:
 - Domain Name
 - User Name
 - Password
 - Confirm Password - Retype the password.
 - Expression - In the Named Expression list, select General and select True Value from the dropdown list, and then click Add Expression.
4. Click Create, and then click Close.
5. Verify that the policy you created appears in the Authentication Policies and Servers pane.

Creating an Authentication Virtual Server

Configure an authentication virtual server and bind the authentication negotiation policy to the authentication virtual server.

To create an authentication virtual server and bind the negotiation policy by using the command line interface

At the command prompt, type the following commands:

- o add authentication vserver <name> SSL <ipAuthVserver> 443 -authenticationDomain <domainName>
- o bind authentication vserver <name> -policy <negotiatePolicyName>

Example

```
add authentication vserver authen1 SSL 10.102.113.166 443 -authenticationDomain newacp.com
add ssl certKey cert1 -cert "/nsconfig/ssl/complete/server/server_rsa_2048.pem" -key "/nscon
bind ssl vserver authen1 -certkeyName cert1
bind authentication vserver authen1 -policy negopol
```

To create an authentication virtual server and bind the negotiation policy by using the NetScaler configuration utility

1. Navigate to Security > AAA-Application Traffic > Virtual Servers.
2. In the details pane, click Add.
3. In the Create Virtual Server (Authentication) dialog box, set the following parameters:
 - o Name
 - o IP Address
 - o Protocol - Select SSL
 - o Domain - Type the fully qualified domain name added while creating the keytab file.

Note: For AAA, the protocol must be SSL protocol and port must be 443. Therefore, these options are not provided.
4. On the Authentication tab, click Insert Policy. In the Authentication Policies group, from the Policy Name dropdown list select the negotiate authentication policy you added for Kerberos authentication.
5. On the Certificates tab, select an SSL certificate from the list of available certificates, and then click Add. If the certificate you want to bind is not displayed in the Available Certificates list, click Install..., and then select the certificate file.
6. Click Create, and then click Close. The new authentication virtual server appears in the Authentication Virtual Servers pane.

Configuring a Traffic Management Virtual Server

Configure an authentication service and a traffic management virtual server, and bind the service to the virtual server. You can use either a load balancing or a content switching virtual server.

To create a traffic management virtual server and service, and bind the service by using the command line interface

At the command prompt, type the following commands:

- o add service <name>@ <ipBackendWebserver> HTTP 80
 - o add lb vserver <name>@ SSL <ipAddressLbVserver> 443 -authn401 ON -authnVsName <authVserverName>
 - o bind lb vserver <name>@ <serviceName>
- Note: Use a similar procedure for using a content switching virtual server as the traffic management virtual server.

Example

```
add service svc1 10.217.28.92 HTTP 80
add lb vserver v2 HTTP 10.102.113.164 80 -persistenceType NONE -cltTimeout 180 -authn401 ON
bind lb vserver v2 svc1
```

To create a traffic management virtual server and service, and bind the service by using the NetScaler configuration utility

1. Navigate to Traffic Management > Load Balancing > Services.
 2. In the details pane, click Add.
 3. In the Create Service dialog box, set the following parameters:
 - o Service Name
 - o Server
 - o Protocol - Select HTTP.
 - o Port - Select 80.
 4. In the navigation pane, expand Load Balancing and click Virtual Servers.
 5. In the details pane, click Add.
 6. In the Create Virtual Server (Load balancing) dialog box, set values for the following parameters:
 - o Name
 - o IP Address
 - o Protocol
 - o Port
 7. In the Create Virtual Server (Load balancing) dialog box, on the Services tab, select the service you created in Step 3 to Step 5.
 8. In the Create Virtual Server (Load balancing) dialog box, on the Advanced tab, expand Authentication Settings, and then select the 401 Based Authentication check box.
 9. Click Create, and then click Close. The new load balancing virtual server appears in the Load Balancing Virtual Servers pane.
 10. In the details pane, verify the settings of the virtual server.
- Note: Use a similar procedure to create a content switching virtual server.
- Note: For more information, see [Setting up basic load balancing](#).

Verifying the configuration for Kerberos Authentication

Ensure that you completed the following tasks and verify whether the configuration is complete and correct.

- Enable the AAA feature
- Import the keytab file
- Configure the DNS server
- Configure negotiation policies and actions
- Configure authentication virtual server
- Configure traffic management virtual server

To verify the configuration:

1. Access the load balancing virtual server, using the FQDN. For example, <http://owa.newacp.com>.
2. View the AAA session on the NetScaler. `show aaa session`

Example

```
ClientIp (ClientPort) ->ServerIp(ServerPort)
-----
PE id : 4
User name: john.smith@NEWACP.COM Session Type: TM
Done
```

Configuration of Kerberos Authentication on a Client

Kerberos support must be configured on the browser to use Kerberos for authentication. You can use any Kerberos-compliant browser. Instructions for configuring Kerberos support on Internet Explorer and Mozilla Firefox follow. For other browsers, see the documentation of the browser.

To configure Internet Explorer for Kerberos authentication

1. In the Tools menu select Internet Options.
2. On the Security tab, click Local Intranet, and then click Sites.
3. In the Local Intranet dialog box, make sure that the Automatically detect intranet network option is selected, and then click Advanced.
4. In the Local Intranet dialog box, add the web sites of the domains of the traffic management virtual server on the NetScaler. The specified sites become local intranet sites.
5. Click Close or OK to close the dialog boxes.

To configure Mozilla Firefox for Kerberos authentication

1. Make sure that you have Kerberos properly configured on your computer.
 2. Type `about:config` in the URL bar.
 3. In the filter text box, type `network.negotiate`.
 4. Change `network.negotiate-auth.delegation-uris` to the domain that you want to add.
 5. Change `network.negotiate-auth.trusted-uris` to the domain that you want to add.
- Note: If you are running Windows, you also need to enter `sspi` in the filter text box and change the `network.auth.use-sspi` option to `False`.

Offloading Kerberos Authentication from Physical Servers

The NetScaler appliance can offload authentication tasks from servers. Instead of the physical servers authenticating the requests from clients, the Netscaler authenticates all the client requests before it forwards them to any of the physical servers bound to it. The user authentication is based on Active Directory tokens.

There is no authentication between the NetScaler and the physical server, and the authentication offload is transparent to the end users. After the initial logon to a Windows computer, the end user does not have to enter any additional authentication information in a pop-up or on a logon page.

In the current NetScaler release, Kerberos authentication is available only for Authentication, Authorization, and Auditing (AAA) Traffic Management Virtual Servers. Kerberos authentication is not supported for SSL VPN in the NetScaler Gateway Enterprise Edition appliance or for NetScaler appliance management.

Kerberos authentication requires configuration on the NetScaler appliance and on client browsers.

To configure Kerberos authentication on the NetScaler appliance

1. Create a user account on Active Directory. When creating a user account, verify the following options in the User Properties section:
 - o Make sure that you do not select the Change password at next logon option.
 - o Be sure to select the Password does not expire option.
2. On the NetScaler appliance, at the CLI command prompt, type:
 - o `ktpass -princ HTTP/kerberos.crete.lab.net@crete.lab.net -ptype KRB5_NT_PRINCIPAL -mapuser kerbuser@crete.lab.net -mapop set -pass Citrix1 -out C:\kerbtabfile.txt`Note: Be sure to type the above command on a single line. The output of the above command is written into the `C:\kerbtabfile.txt` file.
3. Upload the `kerbatabfile.txt` file to the `/etc` directory of the NetScaler appliance by using a Secure Copy (SCP) client.
4. Run the following command to add a DNS server to the NetScaler appliance.
 - o `add dns nameserver 1.2.3.4`

The NetScaler appliance cannot process Kerberos requests without the DNS server. Be sure to use the same DNS server that is used in the Microsoft Windows domain.

5. Switch to the shell prompt and run the following commands from the shell prompt:
 - o `ktutil # rkt /etc/kerbtabfile.txt`
 - o `# wkt /etc/krb5.keytab`
 - o `# list`

The `list` command displays the user account details that you created in the Active Directory. A sample screen of the output of the `list` command is shown below.

Figure 1. Sample Output of the `list` Command

```
> shell
Copyright (c) 1992-2008 The FreeBSD Project.
Copyright (c) 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992,
The Regents of the University of California. All rights reserved.

root@ns# cd /etc
root@ns# ls -la *.txt
-rw-r--r--  1 root  wheel  82 Apr  4 00:43 kerbtabfile.txt
root@ns# ktutil
ktutil: rkt /etc/kerbtabfile.txt
ktutil: wkt /etc/krb5.keytab
ktutil: list
slot KVO Principal
-----
1      3 HTTP/kerberos.crete.example.com@crete.example.com
ktutil: quit
root@ns#
```

6. Switch to the command line interface of NetScaler.
7. Run the following command to create a Kerberos authentication server:
 - o `add authentication negotiateAction KerberosServer -domain "crete.lab.net" -domainUser kerbuser -domainUserPasswd Citrix1`
8. Run the following command to create a negotiation policy:
 - o `add authentication negotiatePolicy Kerberos-Policy "REQ.IP.DESTIP == 192.168.17.200" KerberosServer`
9. Run the following command to create an authentication virtual server.
 - o `add authentication vserver Kerb-Auth SSL 192.168.17.201 443 -AuthenticationDomain crete.example.com`
10. Run the following command to bind the Kerberos policy to the authentication virtual server:
 - o `bind authentication vserver Kerb-Auth -policy Kerberos-Policy -priority 100`

11. Run the following command to bind an SSL certificate to the authentication virtual server. You can use one of the test certificates, which you can install from the GUI NetScaler appliance. Run the following command to use the ServerTestCert sample certificate.
 - o bind ssl vserver Kerb-Auth -certkeyName ServerTestCert
12. Create an HTTP load balancing virtual server with the IP address, 192.168.17.200.

Ensure that you create a virtual server from the command line interface for NetScaler 9.3 releases if they are older than 9.3.47.8.

13. Run the following command to configure an authentication virtual server:
 - o set lb vserver <name>-authn401 ON -authnVsName Kerb-Auth
14. Enter the host name `http://www.crete.example.com` in the address bar of the Web browser.

The Web browser displays an authentication dialog box because the Kerberos authentication is not set up in the browser.

Note: Kerberos authentication requires a specific configuration on the client. Ensure that the client can resolve the hostname, which results in the Web browser connecting to an HTTP virtual server.

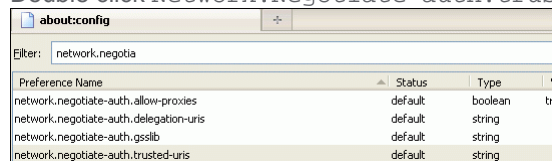
15. Configure Kerberos on the Web browser of the client computer.
 - o For configuring on Internet Explorer, see "[Configuring Internet Explorer for Kerberos authentication.](#)"
 - o For configuring on Mozilla Firefox, see "[Configuring Mozilla Firefox for Kerberos authentication.](#)"
16. Verify whether you can access the backend physical server without authentication.

To configure Internet Explorer for Kerberos authentication

1. Select Internet Options from the Tools menu.
2. Activate the Security tab.
3. Select Local Intranet from the Select a zone to view change security settings section.
4. Click Sites.
5. Click Advanced.
6. Specify the URL, `http://www.crete.example.com` and click Add.
7. Restart Internet Explorer.

To configure Mozilla Firefox for Kerberos authentication

1. Enter `about:config` in the address bar of the browser.
2. Click the warning disclaimer.
3. Type `Network.Negotiate-auth.trusted-uris` in the Filter box.
4. Double click `Network.Negotiate-auth.trusted-uris`. A sample screen is shown below.



Preference Name	Status	Type
network.negotiate-auth.allow-proxies	default	boolean
network.negotiate-auth.delegation-uris	default	string
network.negotiate-auth.gsslib	default	string
network.negotiate-auth.trusted-uris	default	string

5. In the Enter String Value dialog box, specify `www.crete.example.com`.
6. Restart Firefox.

NetScaler Kerberos Single Sign-On

NetScaler appliances now support single sign-on (SSO) using the Kerberos 5 protocol. Users log on to a proxy, the Application Delivery Controller (ADC), which then provides access to protected resources.

The NetScaler Kerberos SSO implementation requires the user's password for SSO methods that rely on basic, NTLM, or forms-based authentication. The user's password is not required for Kerberos SSO, although if Kerberos SSO fails and the NetScaler appliance has the user's password, it uses the password to attempt NTLM SSO.

If the user's password is available, the KCD account is configured with a realm, and no delegated user information is present, the NetScaler Kerberos SSO engine impersonates the user to obtain access to authorized resources. Impersonation is also called unconstrained delegation.

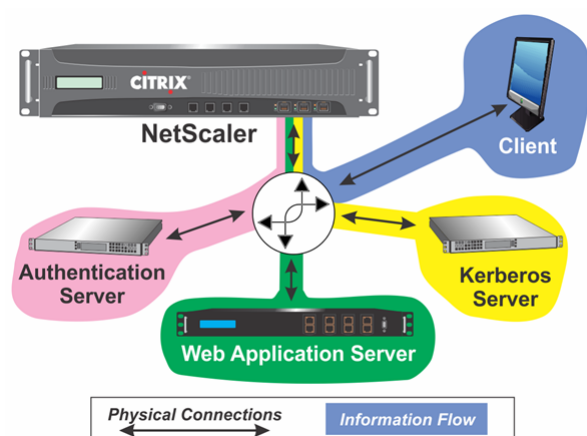
The NetScaler Kerberos SSO engine can also be configured to use a delegated account to obtain access to protected resources on the user's behalf. This configuration requires delegated user credentials, a keytab, or a delegated user certificate and matching CA certificate. Configuration that uses a delegated account is called constrained delegation.

An Overview of NetScaler Kerberos SSO

To use the NetScaler Kerberos SSO feature, users first authenticate with Kerberos or a supported third-party authentication server. Once authenticated, the user requests access to a protected web application. The web server responds with a request for proof that the user is authorized to access that web application. The user's browser contacts the Kerberos server, which verifies that the user is authorized to access that resource, and then provides the user's browser with a service ticket that provides proof. The browser resends the user's request to the web application server with the service ticket attached. The web application server verifies the service ticket, and then allows the user to access the application.

AAA-TM implements this process as shown in the following diagram. The diagram illustrates the flow of information through the NetScaler appliance and AAA-TM, on a secure network with LDAP authentication and Kerberos authorization. AAA-TM environments that use other types of authentication have essentially the same information flow, although they might differ in some details.

Figure 1. A Secure Network with LDAP and Kerberos



NetScaler AAA-TM authentication and authorization in a Kerberos environment requires that the following actions take place.

1. The client sends a request for a resource to the traffic management virtual server on the NetScaler appliance.
2. The traffic management virtual server passes the request to the authentication virtual server, which authenticates the client and then passes the request back to the traffic management virtual server.
3. The traffic management virtual server sends the client's request to the web application server.
4. The web application server responds to the traffic management virtual server with a 401 Unauthorized message that requests Kerberos authentication, with fallback to NTLM authentication if the client does not support Kerberos.
5. The traffic management virtual server contacts the Kerberos SSO daemon.
6. The Kerberos SSO daemon contacts the Kerberos server and obtains a ticket-granting ticket (TGT) allowing it to request service tickets authorizing access to protected applications.
7. The Kerberos SSO daemon obtains a service ticket for the user and sends that ticket to the traffic management virtual server.
8. The traffic management virtual server attaches the ticket to the user's initial request and sends the modified request back to the web application server.
9. The web application server responds with a 200 OK message.

These steps are transparent to the client, which just sends a request and receives the requested resource.

Integration of NetScaler Kerberos SSO with Authentication Methods

All AAA-TM authentication mechanisms support NetScaler Kerberos SSO. AAA-TM supports the Kerberos SSO mechanism with the Kerberos, CAC (Smart Card) and SAML authentication mechanisms with any form of client authentication to the NetScaler appliance. It also supports the HTTP-Basic, HTTP-Digest, Forms-based, and NTLM (versions 1 and 2) SSO mechanisms if the client uses either HTTP-Basic or Forms-Based authentication to log on to the NetScaler appliance.

The following table shows each supported client-side authentication method, and the supported server-side authentication method for that client-side method.

Table 1. Supported Authentication Methods

Client-Side Authentication Method	Server-Side Authentication Method	Server-Side Authentication Method	Server-Side Authentication Method
Basic/Digest/NTLM	Kerberos Constrained Delegation	User Impersonation	

CAC (Smart Card): at SSL/TLS Layer	Â	X	X
Forms-Based (LDAP/RADIUS/TACACS)	X	X	X
HTTP Basic (LDAP/RADIUS/TACACS)	X	X	X
Kerberos	Â	X	Â
NTLM v1/v2	Â	X	X
SAML	Â	X	Â
SAML Two-Factor	X	X	X
Certificate Two-Factor	X	X	X

Setting up NetScaler SSO

You can configure NetScaler SSO to work in one of two ways: by impersonation or by delegation. SSO by impersonation is a simpler configuration than SSO by delegation, and is therefore usually preferable when your configuration allows it. To configure NetScaler SSO by impersonation, you must have the user's user name and password.

To configure NetScaler SSO by delegation, you must have the delegated user's credentials in one of the following formats: the user's user name and password, the keytab configuration that includes the user name and an encrypted password, or the delegated user certificate and the matching CA certificate.

Prerequisites

Before you configure NetScaler SSO, you need to have your NetScaler appliance fully configured to manage traffic to and authentication for your web application servers. Therefore, you must configure either load balancing or content switching, and then AAA, for these web application servers. You should also verify routing between the appliance, your LDAP server, and your Kerberos server.

If your network is not already configured in this manner, perform the following configuration tasks:

- Configure a server and service for each web application server.
- Configure a traffic management virtual server to handle traffic to and from your web application server.

Following are brief instructions and examples for performing each of these tasks from the NetScaler command line. For further assistance, see "" and ["Setting up AAA Virtual Servers and DNS."](#)

To create a server and service by using the NetScaler command line

For NetScaler SSO to obtain a TGS (service ticket) for a service, either the FQDN assigned to the server entity on the NetScaler appliance must match the FQDN of the web application server, or the server entity name must match the NetBios name of the web application server. You can take either of the following approaches:

- Configure the NetScaler server entity by specifying the FQDN of the web application server.
- Configure the NetScaler server entity by specifying the IP address of the web application server, and assign the server entity the same name as the NetBios name of the web application server.

At the command prompt, type the following commands:

- `add server name <serverFQDN>`
- `add service name serverName serviceType port`

For the variables, substitute the following values:

- **serverName** – A name for the NetScaler appliance to use to refer to this server.
- **serverFQDN** – The FQDN of the server. If the server has no domain assigned to it, use the server's IP address and make sure that the server entity name matches the NetBios name of the web application server.
- **serviceName** – A name for the NetScaler appliance to use to refer to this service.
- **type** – The protocol used by the service, either HTTP or MSSQLSVC.
- **port** – The port on which the service listens. HTTP services normally listen on port 80. Secure HTTPS services normally listen on port 443.

Example

The following examples add server and service entries on the NetScaler appliance for the web application server `was1.example.com`. The first example uses the FQDN of the web application server; the second uses the IP address.

To add the server and service using the web application server FQDN, `was1.example.com`, you would type the following commands:

```
add server was1 was1.example.com
add service was1service was1 HTTP 80
```

To add the server and service using the web application server IP and NetBios name, where the web application server IP is `10.237.64.87` and its NetBios name is `WAS1`, you would type the following commands:

```
add server WAS1 10.237.64.87
add service was1service WAS1 HTTP 8
```

To create a traffic management virtual server by using the NetScaler command line

The traffic management virtual server manages traffic between the client and the web application server. You can use either a load balancing or a content switching virtual server as the traffic management server. The SSO configuration is the same for either type.

To create a load balancing virtual server, at the command prompt, type the following command:

```
add lb vserver <vserverName> <type> <IP> <port>
```

For the variables, substitute the following values:

- o **vserverName** – A name for the NetScaler appliance to use to refer to this virtual server.
- o **type** – The protocol used by the service, either HTTP or MSSQLSVC.
- o **IP** – The IP address assigned to the virtual server. This would normally be an IANA-reserved, non-public IP address on your LAN.
- o **port** – The port on which the service listens. HTTP services normally listen on port 80. Secure HTTPS services normally listen on port 443.

Example

To add a load balancing virtual server called `tmvserver1` to a configuration that manages HTTP traffic on port 80, assigning it a LAN IP address of `10.217.28.20` and then binding the load balancing virtual server to the `wasservice1` service, you would type the following commands:

```
add lb vserver tmvserver1 HTTP 10.217.28.20 80
bind lb vserver tmvserv1 wasservice1
```

To create an authentication virtual server by using the NetScaler command line

The authentication virtual server manages authentication traffic between the client and the authentication (LDAP) server. To create an authentication virtual server, at the command prompt type the following commands:

- o `add authentication vserver <authvserverName> SSL <IP> 443`
- o `set authentication vserver <authvservername> authenticationdomain <domain>`

For the variables, substitute the following values:

- o **authvserverName** – A name for the NetScaler appliance to use to refer to this authentication virtual server. Must begin with a letter, number, or the underscore character (`_`), and must contain only letters, numbers, and the hyphen (`-`), period (`.`), pound (`#`), space (), at (`@`), equals (`=`), colon (`:`), and underscore characters. Can be changed after the authentication virtual server is added by using the `rename authentication vserver` command.
- o **IP** – The IP address assigned to the authentication virtual server. As with the traffic management virtual server, this address would normally be an IANA-reserved, non-public IP on your LAN.
- o **domain** – The domain assigned to the virtual server. This would usually be the domain of your network. It is customary, though not required, to enter the domain in all capitals when configuring the authentication virtual server.

Example

To add an authentication virtual server called `authvserver1` to your configuration and assign it the LAN IP `10.217.28.21` and the domain `EXAMPLE.COM`, you would type the following commands:

```
add authentication vserver authvserver1 SSL 10.217.28.21 443
set authentication vserver authvserver1 authenticationdomain EXAMPLE.COM
```

To configure a traffic management virtual server to use an authentication profile

The authentication virtual server can be configured to handle authentication for a single domain or for multiple domains. If it is configured to support authentication for multiple domains, you must also specify the domain for NetScaler SSO by creating an authentication profile, and then configuring the traffic management virtual server to use that authentication profile.

Note: The traffic management virtual server can be either a load balancing (`lb`) or content switching (`cs`) virtual server. The following instructions assume that you are using a load balancing virtual server. To configure a content switching virtual server, simply substitute `set cs vserver` for `set lb vserver`. The procedure is otherwise the same.

To create the authentication profile, and then configure the authentication profile on a traffic management virtual server, type the following commands:

- o add authentication authnProfile <authnProfileName> {-authvserverName <string>} {-authenticationHost <string>} {-authenticationDomain <string>}
- o set lb vserver <vserverName> -authnProfile <authnprofileName>

For the variables, substitute the following values:

- o **authnprofileName** – A name for the authentication profile. Must begin with a letter, number, or the underscore character (_), and must consist of from one to thirty-one alphanumeric or hyphen (-), period (.) pound (#), space (), at (@), equals (=), colon (:), and underscore characters.
- o **authvserverName** – The name of the authentication virtual server that this profile uses for authentication.
- o **authenticationHost** – Host name of the authentication virtual server.
- o **authenticationDomain** – Domain for which NetScaler SSO handles authentication. Required if the authentication virtual server performs authentication for more than one domain, so that the correct domain is included when the NetScaler appliance sets the traffic management virtual server cookie.

Example

To create an authentication profile named `authnProfile1` for authentication of the `example.com` domain, and to configure the load balancing virtual server `vserver1` to use the authentication profile `authnProfile1`, you would type the following commands:

```
add authentication authnProfile authnProfile1 -authnvsName authvsesrver1
    -authenticationHost authvsesrver1 -authenticationDomain example.com
set lb vserver vserver1 -authnProfile authnProfile1
```

Configuring SSO

Configuring NetScaler SSO to authenticate by impersonation is simpler than configuring SSO to authenticate by delegation, and is therefore usually preferable when your configuration allows it. You just create a KCD account. You can use the user's password.

If you do not have the user's password, you can configure NetScaler SSO to authenticate by delegation. Although somewhat more complex than configuring SSO to authenticate by impersonation, the delegation method provides flexibility in that a user's credentials might not be available to the NetScaler appliance in all circumstances.

For either impersonation or delegation, you must also enable integrated authentication on the web application server.

Enabling Integrated Authentication on the Web Application Server

To set up NetScaler Kerberos SSO on each web application server that Kerberos SSO will manage, use the configuration interface on that server to configure the server to require authentication. Select Kerberos (negotiate) authentication by preference, with fallback to NTLM for clients that do not support Kerberos.

Following are instructions for configuring Microsoft Internet Information Server (IIS) to require authentication. If your web application server uses software other than IIS, consult the documentation for that web server software for instructions.

To configure Microsoft IIS to use integrated authentication

1. Log on to the IIS server and open Internet Information Services Manager.
2. Select the web site for which you want to enable integrated authentication. To enable integrated authentication for all IIS web servers managed by IISM, configure authentication settings for the Default Web Site. To enable integrated authentication for individual services (such as Exchange, Exadmin, ExchWeb, and Public), configure these authentication settings for each service individually.
3. Open the Properties dialog box for the default web site or for the individual service, and click the Directory Security tab.
4. Beside Authentication and Access Control, select Edit.
5. Disable anonymous access.
6. Enable Integrated Windows authentication (only). Enabling integrated Windows authentication should automatically set protocol negotiation for the web server to `Negotiate, NTLM`, which specifies Kerberos authentication with fallback to NTLM for non-Kerberos capable devices. If this option is not automatically selected, manually set protocol negotiation to `Negotiate, NTLM`.

Setting Up SSO by Impersonation

You can configure the KCD account for NetScaler SSO by impersonation. In this configuration, the NetScaler appliance obtains the user's username and password when the user authenticates to the authentication server and uses those credentials to impersonate the user to obtain a ticket-granting ticket (TGT). If the user's name is in UPN format, the appliance obtains the user's realm from UPN. Otherwise, it obtains the user's name and realm by extracting it from the SSO domain used during initial authentication, or from the session profile.

When configuring the KCD account, you must set the realm parameter to the realm of the service that the user is accessing. The same realm is also used as the user's realm if the user's realm cannot be obtained from authentication with the Netscaler appliance or from the session profile.

To create the KCD account for SSO by impersonation with a password

At the command prompt, type the following command:

```
add aaa kcdaccount <accountname> -realmStr <realm>
```

For the variables, substitute the following values:

- o **accountname**—The KCD account name.
- o **realm**—The domain assigned to NetScaler SSO.

Example:

To add a KCD account named `kcdaccount1`, and use the keytab named `kcdvserver.keytab`, you would type the following command:

```
add aaa kcdAccount kcdaccount1 -keytab kcdvserver.keytab
```

Configuring SSO by Delegation

To configure SSO by Delegation, you need to perform the following tasks:

- If you are configuring delegation by delegated user certificate, install the matching CA certificates on the NetScaler appliance and add them to the NetScaler configuration.
- Create the KCD account on the appliance. The appliance uses this account to obtain service tickets for your protected applications.
- Configure the Active Directory server.

Installing the Client CA Certificate on the NetScaler appliance

If you are configuring NetScaler SSO with a client certificate, you must copy the matching CA certificate for the client certificate domain (the client CA certificate) to the NetScaler appliance, and then install the CA certificate. To copy the client CA certificate, use the file transfer program of your choice to transfer the certificate and private-key file to the NetScaler appliance, and store the files in `/nsconfig/ssl`.

To install the client CA certificate on the NetScaler appliance

At the command prompt, type the following command:

```
add ssl certKey <certKeyName> -cert <cert> [(-key <key> [-password]) | -fipsKey <fipsKey>] [-inform ( DER | PEM )] [-expiryMonitor ( ENABLED | DISABLED | UNSET ) [-notificationPeriod <positive_integer>]] [-bundle ( YES | NO )]
```

For the variables, substitute the following values:

- **certKeyName**—A name for the client CA certificate. Must begin with an ASCII alphanumeric or underscore (_) character, and must consist of from one to thirty-one characters. Allowed characters include the ASCII alphanumerics, underscore, hash (#), period(.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the certificate-key pair is created. If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my cert" or 'my cert').
- **cert**—Full path name and file name of the X509 certificate file used to form the certificate-key pair. The certificate file must be stored on the NetScaler appliance, in the `/nsconfig/ssl/` directory.
- **key**—Full path name and file name of the file that contains the private key to the X509 certificate file. The key file must be stored on the NetScaler appliance in the `/nsconfig/ssl/` directory.
- **password**—If a private key is specified, the passphrase used to encrypt the private key. Use this option to load encrypted private keys in PEM format.
- **fipsKey**—Name of the FIPS key that was created inside the Hardware Security Module (HSM) of a FIPS appliance, or a key that was imported into the HSM.
Note: You can specify either a `key` or a `fipsKey`, but not both.
- **inform**—Format of the certificate and private-key files, either PEM or DER.
- **passplain**—Pass phrase used to encrypt the private key. Required when adding an encrypted private-key in PEM format.
- **expiryMonitor**—Configure the NetScaler appliance to issue an alert when the certificate is about to expire. Possible values: ENABLED, DISABLED, UNSET.
- **notificationPeriod**—If expiryMonitor is ENABLED, number of days before the certificate expires to issue an alert.
- **bundle**—Parse the certificate chain as a single file after linking the server certificate to its issuer's certificate within the file. Possible values: YES, NO.

Example

The following example adds the specified delegated user certificate `customer-cert.pem` to the NetScaler configuration along with the key `customer-key.pem`, and sets the password, certificate format, expiration monitor, and notification period.

To add the delegated user certificate, you would type the following commands:

```
add ssl certKey customer -cert "/nsconfig/ssl/customer-cert.pem"  
-key "/nsconfig/ssl/customer-key.pem" -password "dontUseDefaultPWs!"  
-inform PEM -expiryMonitor ENABLED [-notificationPeriod 14]
```

Creating the KCD Account

If you are configuring NetScaler SSO by delegation, you can configure the KCD account to use the user's log-on name and password, to use the user's log-on name and keytab, or to use the user's client certificate. If you configure SSO with user name and password, the NetScaler appliance uses the delegated user account to obtain a Ticket Granting Ticket (TGT), and then uses the TGT to obtain service tickets for the specific services that each user requests. If you configure SSO with keytab file, the NetScaler appliance uses the delegated user account and keytab information. If you configure SSO with a delegated user certificate, the NetScaler appliance uses the delegated user certificate.

To create the KCD account for SSO by delegation with a password

At the command prompt, type the following commands:

```
add aaa kcdaccount <accountname> -delegatedUser root -kcdPassword <password> -realmStr <realm>
```

For the variables, substitute the following values:

- o **accountname**—A name for the KCD account.
- o **password**—A password for the KCD account.
- o **realm**—The realm of the KCD account, usually the domain for which SSO is active.

Example (UPN Format)

To add a KCD account named `kcdaccount1` to the NetScaler appliance configuration with a password of `password1` and a realm of `EXAMPLE.COM`, specifying the delegated user account in UPN format (as root), you would type the following commands:

```
add aaa kcdaccount kcdaccount1 -delegatedUser root  
-kcdPassword password1 -realmStr EXAMPLE.COM
```

Example (SPN Format)

To add a KCD account named `kcdaccount1` to the NetScaler appliance configuration with a password of `password1` and a realm of `EXAMPLE.COM`, specifying the delegated user account in SPN format, you would type the following commands:

```
add aaa kcdAccount kcdaccount1 -realmStr EXAMPLE.COM  
-delegatedUser "host/kcdvserver.example.com" -kcdPassword password1
```

Creating the KCD account for SSO by delegation with a keytab

If you plan to use a keytab file for authentication, first create the keytab. You can create the keytab file manually by logging onto the AD server and using the `ktpass` utility, or you can use the NetScaler configuration utility to create a batch script, and then run that script on the AD server to generate the keytab file. Next, use FTP or another file transfer program to transfer the keytab file to the NetScaler appliance and place it in the `/nsconfig/krb` directory. Finally, configure the KCD account for NetScaler SSO by delegation and provide the path and file name of the keytab file to the NetScaler appliance.

To create the keytab file manually

Log on to the AD server command line and, at the command prompt, type the following command:

```
ktpass /princ <SPN> /ptype KRB5_NT_PRINCIPAL /mapuser <DOMAIN>\<username> /pass <password> -out <File_Path>
```

For the variables, substitute the following values:

- o **SPN**—The service principal name for the KCD service account.
- o **DOMAIN**—The domain of the Active Directory server.
- o **username**—The KSA account username.
- o **password**—The KSA account password.
- o **path**—The full path name of the directory in which to store the keytab file after it is generated.

To use the NetScaler configuration utility to create a script to generate the keytab file.

1. Navigate to Security > AAA - Application Traffic
2. In the data pane, under Kerberos Constrained Delegation, click Batch file to generate Keytab.
3. In the Generate KCD (Kerberos Constrained Delegation) Keytab Script dialog box, set the following parameters:
 - o **Domain User Name**—The KSA account username.
 - o **Domain Password**—The KSA account password.
 - o **Service Principal**—The service principal name for the KSA.
 - o **Output File Name**—The full path and file name to which to save the keytab file on the AD server.

4. Clear the Create Domain User Account check box.
5. Click Generate Script.
6. Log on to the Active Directory server and open a command line window.
7. Copy the script from the Generated Script window and paste it directly into the Active Directory server command-line window. The keytab is generated and stored in the directory under the file name that you specified as **Output File Name**.
8. Use the file transfer utility of your choice to copy the keytab file from the Active Directory server to the NetScaler appliance and place it in the `/nsconfig/krb` directory.

To create the KCD account

At the command prompt, type the following command:

```
add aaa kcdaccount <accountname> â€"keytab <keytab>
```

Example:

To add a KCD account named `kcdaccount1`, and use the keytab named `kcdvserver.keytab`, you would type the following commands:

```
add aaa kcdaccount kcdaccount1 â€"keytab kcdvserver.keytab
```

To create the KCD account for SSO by delegation with a delegated user cert

At the command prompt, type the following command:

```
add aaa kcdaccount <accountname> -realmStr <realm> -delegatedUser <user_name/SPN> -usercert <cert> -cacert <cacert>
```

For the variables, substitute the following values:

- o **accountname**â€"A name for the KCD account.
- o **realmStr**â€"The realm for the KCD account, usually the domain for which SSO is configured.
- o **delegatedUser**â€"The delegated user name, in SPN format.
- o **usercert**â€"The full path and name of the delegated user certificate file on the NetScaler appliance. The delegated user certificate must contain both the client certificate and the private key, and must be in PEM format. If you use smart card authentication, you might need to create a smart card certificate template to allow certificates to be imported with the private key.
- o **cacert**â€"The full path to and name of the CA certificate file on the NetScaler appliance.

Example:

To add a KCD account named `kcdaccount1`, and use the keytab named `kcdvserver.keytab`, you would type the following command:

```
add aaa kcdaccount kcdaccount1 -realmStr EXAMPLE.COM
    -delegatedUser "host/kcdvserver.example.com" -usercert /certs/usercert
    -cacert /cacerts/cacert
```

Setting up Active Directory for NetScaler SSO

When you configure SSO by delegation, in addition to creating the KCDAccount on the NetScaler appliance, you must also create a matching Kerberos Service Account (KSA) on your LDAP active directory server, and configure the server for SSO. To create the KSA, use the account creation process on the active directory server. To configure SSO on the active directory server, open the properties window for the KSA. In the Delegation tab, enable the following options: Trust this user for delegation to specified services only and Use any Authentication protocol. (The Kerberos only option does not work, because it does not enable protocol transition or constrained delegation.) Finally, add the services that NetScaler SSO will manage.

Note: If the Delegation tab is not visible in the KSA account properties dialog box, before you can configure the KSA as described, you must use the Microsoft `setspn` command-line tool to configure the active directory server so that the tab is visible.

To configure delegation for the Kerberos service account

1. In the LDAP account configuration dialog box for the Kerberos service account that you created, click the Delegation tab.
2. Choose "Trust this user for delegation to the specified services only".
3. Under "Trust this user for delegation to the specified services only," choose "Use any authentication protocol".
4. Under "Services to which this account can present delegated credentials," clickAdd.

5. In the Add Services dialog box, click Users or Computers, choose the server that hosts the resources to be assigned to the service account, and then click OK.
Note: Constrained delegation does not support services hosted in domains other than the domain assigned to the account, even though Kerberos might have a trust relationship with other domains
6. Back in the Add Services dialog box, in the Available Services list, choose the services assigned to the service account. NetScaler SSO supports the HTTP and MSSQLSVC services.
7. Click OK.

Generating the KCD Keytab Script

The KCD Keytab Script dialog box generates the keytab script, which in turn generates the keytab file necessary to configure KCD on the NetScaler ADC.

To generate the KCD keytab script by using the configuration utility

1. Navigate to Security > AAA - Application Traffic
2. In the details pane, under Kerberos Constrained Delegation, click Batch file to generate keytab.
3. In the Generate KCD (Kerberos Constrained Delegation) Keytab Script dialog box, fill out the fields as described below.
 - o **Domain User Name:** The name of the domain user.
 - o **Domain Password:** The password for the domain user.
 - o **Service Principal:** The service principal.
 - o **Output File Name:** A filename for the KCD script file.
 - o **Create Domain User Account:** Select this check box to create the specified domain user account.
4. Click Generate Script to generate the script. The script is generated, and appears in the Generated Script text box below the Generate Script button.
5. Copy the script, and save it as a file on your AD domain controller. You must now run this script on the domain controller to generate the keytab file, and then copy the keytab file to the `/nsconfig/krb/` directory on the NetScaler appliance.
6. Click OK.

SAML IdP

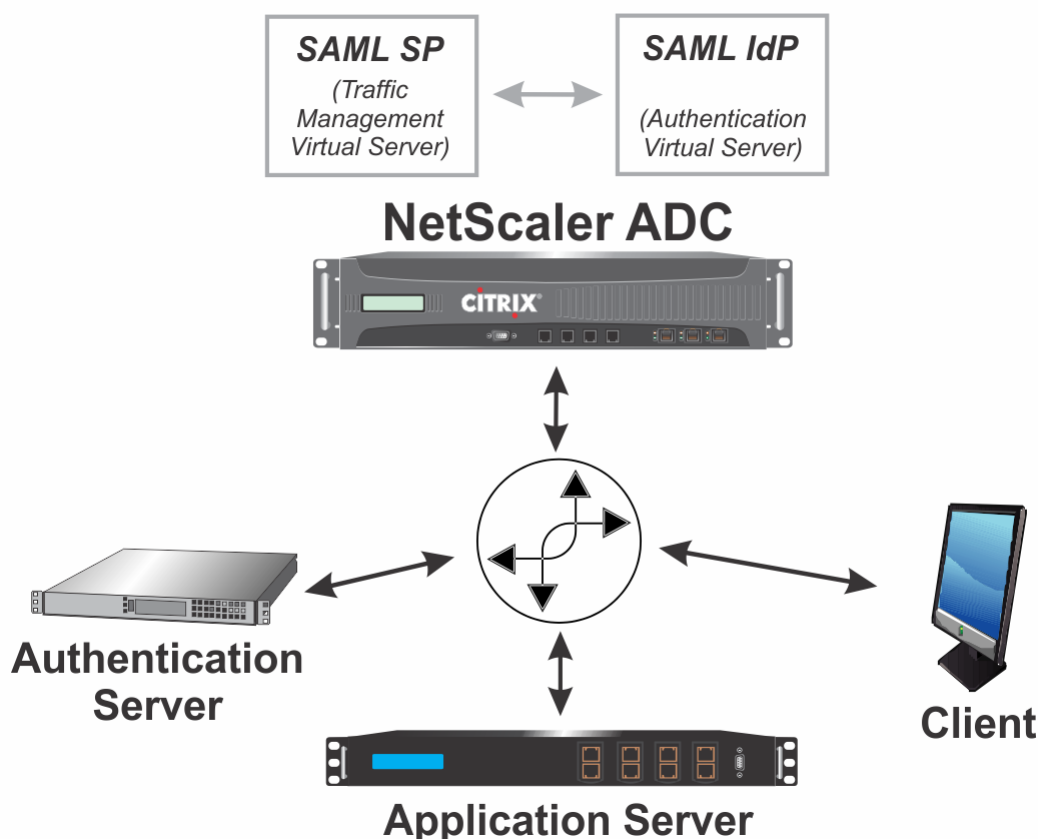
The NetScaler ADC can now act as a SAML identity provider (IdP). As an IdP, the ADC accepts SAML requests from users that request access to a protected application, redirecting users to the logon page to authenticate. After the user authenticates, the ADC generates a SAML assertion that grants access to the protected resource and redirects the user to the resource that requested authentication. When the user logs out or is logged out by any SP, the ADC sends logout requests to all other SPs that the user accessed during the current session and terminates the session.

Note: From NetScaler 11 onwards, the SAML IdP can be configured to encrypt the assertions that are sent to the SAML SP. It is recommended to encrypt assertions as they can include sensitive information. The assertions are encrypted by using the public key of the SAML SP. To encrypt assertions, you must enable encryption and specify the encryption algorithm, in the SAML IdP profile.

The SAML protocol supports a variety of bindings (profiles) which handle the authentication and authorization process in different ways. The NetScaler ADC currently supports only the SP-initiated model with POST binding. An SP-initiated SAML flow usually occurs when a protected resource redirects an unauthenticated client's request to the SAML SP. The SP evaluates the client's request and, if the client has not already authenticated, redirects the request to the IdP.

The NetScaler ADC supports every authentication method with SAML IdP that it supports with traditional logins. It supports both single-factor and two-factor authentication, and also supports SAML authentication with external SPs. The ADC supports attribute extraction from SAML assertions, and encrypted SAML assertions. The NetScaler implementation of SAML allows signing certificates of less than 2048 bits, but displays a warning message. It also supports the SHA256 hash algorithm for signatures and digests. Citrix recommends that all signing certificates be of at least 2048 bits, and that you use SHA256 as SHA-1 is no longer considered secure.

In any SAML configuration of the ADC, the load balancing virtual server functions as the SP. When the NetScaler ADC is configured to serve as the IdP, the authentication virtual server functions as the IdP. In this configuration, most of the SAML conversation takes place entirely within the ADC, as illustrated in the following diagram.



The authentication process occurs in the following sequence:

1. The client requests a protected application.
2. The traffic management virtual server receives the request, and responds to the client with `HTTP 200 OK` and a request for a SAML assertion.
3. If the client does not have a SAML assertion, the client requests to authenticate.
4. The traffic management virtual server sends an authentication request to the authentication virtual server by posting it to `/saml/login`, including the following two XML-based attachments and signing it:
 - o **SAMLRequest**. Standard SAML authentication request with appropriate information.
 - o **RelayState**. Internal NetScaler information.

5. The authentication virtual server receives the signed SAML authentication request and performs the following steps:
 - a. Evaluates its SAML IdP policies to choose the correct profile.
 - b. Uses that profile to validate the SAML authentication request.
 - c. Responds with HTTP 302 Found, and then loads the logon page.
6. The client types his or her credentials into the logon form, and sends them to the authentication virtual server.
7. The authentication virtual server sends the credentials to the authentication server.
8. The authentication server authenticates the client to the authentication virtual server.
9. The authentication virtual server generates a SAML assertion, signs it, and sends the assertion to the client.
10. The client resends the initial request with the SAML assertion.
11. The traffic management virtual server responds with HTTP 302 found and loads the requested resource.

When the ADC is configured to authenticate users with an external IdP, the conversation described above changes slightly. If the ADC is configured to authenticate all users with the external IDP, after the authentication virtual server validates the authentication request it redirects the request to the external IdP, which displays its own login page. If the ADC is configured for fallback authentication, it displays its own logon page and redirects the user to the external IdP only when the user fails to authenticate locally.

When the ADC is configured to use two-factor authentication, it first authenticates the user with the IdP. After the user successfully authenticates with the IdP and the authentication virtual server has validated the SAML assertion, it redirects the user to the second authentication logon page.

Configuring the NetScaler ADC to serve as SAML IdP requires adding a SAML IdP profile and policy to the standard SAML configuration. As with most NetScaler commands, with the following commands you can substitute "set" for "add" to modify an existing SAML IdP profile or policy.

To add a SAML IdP profile from the command line interface

At the command prompt, type:

```
add authentication samlIdPProfile <name> -samlSigningCertName <samlSigningCertName> -samlIdPCertName
<samlIdPCertName> -assertionConsumerServiceURL <URL> [-sendPassword ( ON | OFF )] [-samlIssuerName <string>]
```

Example:

```
> add authentication samlIdPProfile act-auth-samlidp -samlSigningCertName myCert -samlIdPCer
```

To configure a SAML IdP profile by using the configuration utility

1. Navigate to Security > AAA - Application Traffic > Policies > Authentication > Basic Policies > SAML IDP.
2. In the details pane, click the Profiles tab.
3. On the Profiles tab, do one of the following:
 - o To create a new SAML IdP profile, click Add.
 - o To modify an existing SAML IdP profile, select the profile, and then click Edit.
4. In the Create Authentication SAML IDP Profile or the Configure Authentication SAML IdP Profile dialog box, specify values for the required parameters.
5. Click Create or OK, and then click Close.

Application Firewall

The following topics cover installation and configuration of the Citrix Application Firewall feature.

Introduction	An overview of web application security and how the application firewall works.
Configuration	How to configure the application firewall to protect a web site, a web service, or a web 2.0 site.
Signatures	A detailed description of the signatures feature and how to configure the signatures, add signatures from a supported vulnerability scanning tool, and define your own signatures, with examples.
Advanced Protections	A detailed description of all of the application firewall security checks, with configuration information and examples.
Profiles	A description of how profiles are configured and used in the application firewall.
Policies	A description of how policies are used when configuring the application firewall, with examples of useful policies.
Imports	A description of how the application firewall uses different types of imported files, and how to import and export files.
Global Configuration	A description of application firewall features that apply to all profiles, and how to configure them.
Use Cases	Extended examples that demonstrate how to set up the application firewall to best protect specific types of more complex web sites and web services.
Logs, Statistics, and Reports	How to access and use the application firewall logs, the statistics, and the reports to assist in configuring the application firewall.

Introduction

The Citrix NetScaler Application Firewall prevents security breaches, data loss, and possible unauthorized modifications to web sites that access sensitive business or customer information. It does so by filtering both requests and responses, examining them for evidence of malicious activity, and blocking those that exhibit such activity. Your site is protected not only from common types of attacks, but also from new, as yet unknown attacks. In addition to protecting web servers and web sites from unauthorized access and misuse by hackers and malicious programs, the application firewall provides protection against security vulnerabilities in legacy CGI code or scripts, other web frameworks, web server software, and the underlying operating systems.

The NetScaler Application Firewall is available as a stand-alone appliance, or as a feature on a Citrix NetScaler application delivery controller (ADC) or Citrix NetScaler virtual appliance (VPX). In the application firewall documentation, the term NetScaler ADC refers to the platform on which the application firewall is running, regardless of whether that platform is a dedicated firewall appliance, a NetScaler ADC on which other features have also been configured, or a NetScaler VPX.

To use the application firewall, you must create at least one security configuration to block connections that violate the rules that you set for your protected web sites. The number of security configurations that you might want to create depends on the complexity of your web site. In some cases, a single configuration is sufficient. In other cases, particularly those that include interactive web sites, web sites that access database servers, online stores with shopping carts, you might need several different configurations to best protect sensitive data without wasting significant effort on content that is not vulnerable to certain types of attacks. You can often leave the defaults for the global settings, which affect all security configurations, unchanged. However, you can change the global settings if they conflict with other parts of your configuration or you prefer to customize them.

Web Application Security

Web application security is network security for computers and programs that communicate by using the HTTP and HTTPS protocols. This is an extremely broad area in which security flaws and weaknesses abound. Operating systems on both servers and clients have security issues and are vulnerable to attack. Web server software and web site enabling technologies such as CGI, Java, JavaScript, PERL and PHP have underlying vulnerabilities. Browsers and other client applications that communicate with web-enabled applications also have vulnerabilities. Web sites that use any technology but the simplest of HTML, including any site that allows interaction with visitors, often have vulnerabilities of their own.

In the past, a breach in security was often just an annoyance, but today that is seldom the case. For example, attacks in which a hacker gained access to a web server and made unauthorized modifications to (defaced) a web site used to be common. They were usually launched by hackers who had no motivation beyond demonstrating their skills to fellow hackers or embarrassing the targeted person or company. Most current security breaches, however, are motivated by a desire for money. The majority attempt to accomplish one or both of the following goals: to obtain sensitive and potentially valuable private information, or to obtain unauthorized access to and control of a web site or web server.

Certain forms of web attacks focus on obtaining private information. These attacks are often possible even against web sites that are secure enough to prevent an attacker from taking full control. The information that an attacker can obtain from a web site can include customer names, addresses, phone numbers, social security numbers, credit card numbers, medical records, and other private information. The attacker can then use this information or sell it to others. Much of the information obtained by such attacks is protected by law, and all of it by custom and expectation. A breach of this type can have extremely serious consequences for customers whose private information is compromised. At best, these customers will have to exercise vigilance to prevent others from abusing their credit cards, opening unauthorized credit accounts in their name, or appropriating their identities outright (identity theft). At worst, the customers may face ruined credit ratings or even be blamed for criminal activities in which they had no part.

Other web attacks are aimed at obtaining control of (or *compromising*) a web site or the server on which it operates, or both. A hacker who gains control of a web site or server can use it to host unauthorized content, act as a proxy for content hosted on another web server, provide SMTP services to send unsolicited bulk email, or provide DNS services to support such activities on other compromised web servers. Most web sites that are hosted on compromised web servers promote questionable or outright fraudulent businesses. For example, the majority of phishing web sites and child exploitation web sites are hosted on compromised web servers.

Protecting your web sites and web services against these attacks requires a multilayered defense capable of both blocking known attacks with identifiable characteristics and protecting against unknown attacks, which can often be detected because they look different from the normal traffic to your web sites and web services.

Known Web Attacks

Updated: 2014-10-08

The first line of defense for your web sites is protection against the large number of attacks that are known to exist and have been observed and analyzed by web security experts. Common types of attacks against HTML-based web sites include:

- **Buffer overflow attacks.** Sending an extremely long URL, extremely long cookie, or other extremely long bit of information to a web server in hopes of causing it or the underlying operating system to hang, crash, or provide the attacker with access to the underlying operating system. A buffer overflow attack can be used to gain access to unauthorized information, to compromise a web server, or both.
- **Cookie security attacks.** Sending a modified cookie to a web server, usually in hopes of obtaining access to unauthorized content by using falsified credentials.
- **Forceful browsing.** Accessing URLs on a web site directly, without navigating to the URLs by means of hyperlinks on the home page or other common start URLs on the web site. Individual instances of forceful browsing may simply indicate a user who bookmarked a page on your web site, but repeated attempts to access nonexistent content, or content that users should never access directly, often represent an attack on web site security. Forceful browsing is normally used to gain access to unauthorized information, but can also be combined with a buffer overflow attack in an attempt to compromise your server.
- **Web form security attacks.** Sending inappropriate content to your web site in a web form. Inappropriate content can include modified hidden fields, HTML or code in a field intended for alphanumeric data only, an overly long string in a field that accepts only a short string, an alphanumeric string in a field that accepts only an integer, and a wide variety of other data that your web site does not expect to receive in that web form. A web form security attack can be used either to obtain unauthorized information from your web site or to compromise the web site outright, usually when combined with a buffer overflow attack.

Two specialized types of attacks on web form security deserve special mention:

- o **SQL injection attacks.** Sending an active SQL command or commands in a web form or as part of a URL, with the goal of causing an SQL database to execute the command or commands. SQL injection attacks are normally used to obtain unauthorized information.
- o **Cross-site scripting attacks.** Using a URL or a script on a web page to violate the same-origin policy, which forbids any script from obtaining properties from or modifying any content on a different web site. Since scripts can obtain information and modify files on your web site, allowing a script access to content on a different web site can provide an attacker the means to obtain unauthorized information, to compromise a web server, or both.

Attacks against XML-based web services normally fall into at least one of the following two categories: attempts to send inappropriate content to a web service, or attempts to breach security on a web service. Common types of attacks against XML-based web services include:

- o **Malicious code or objects.** XML requests that contain code or objects that can either directly obtain sensitive information or can give an attacker control of the web service or underlying server.
- o **Badly-formed XML requests.** XML requests that do not conform to the W3C XML specification, and that can therefore breach security on an insecure web service.
- o **Denial of service (DoS) attacks.** XML requests that are sent repeatedly and in high volume, with the intent of overwhelming the targeted web service and denying legitimate users access to the web service.

In addition to standard XML-based attacks, XML web services and Web 2.0 sites are also vulnerable to SQL injection and cross-site scripting attacks, as described below:

- o **SQL injection attacks.** Sending an active SQL command or commands in an XML-based request, with the goal of causing an SQL database to execute that command or commands. As with HTML SQL injection attacks, XML SQL injection attacks are normally used to obtain unauthorized information.
- o **Cross-site scripting attacks.** Using a script included in an XML based application to violate the same-origin policy, which does not allow any script to obtain properties from or modify any content on a different application. Since scripts can obtain information and modify files by using your XML application, allowing a script access to content belonging to a different application can give an attacker the means to obtain unauthorized information, to compromise the application, or both.

Known web attacks can usually be stopped by filtering web site traffic for specific characteristics (signatures) that always appear for a specific attack and should never appear in legitimate traffic. This approach has the advantages of requiring relatively few resources and posing relatively little risk of false positives. It is therefore a valuable tool in fighting attacks on web sites and web services, and configuring basic signature protections that intercept most known web attacks is easy to do.

Unknown Web Attacks

The greatest threat against web sites and applications does not come from known attacks, but from unknown attacks. Most unknown attacks fall into one of two categories: newly-launched attacks for which security firms have not yet developed an effective defense (zero-day attacks), and carefully-targeted attacks on a specific web site or web service rather than many web sites or web services (spear attacks). These attacks, like known attacks, are usually intended to obtain sensitive private information, compromise the web site or web service and allow it to be used for further attacks, or both of those goals.

Zero-day attacks are a major threat to all users. These attacks are usually of the same types as known attacks; zero-day attacks often involve injected SQL, a cross-site script, a cross-site request forgery, or another type of attack similar to known attacks. In most cases, they target vulnerabilities that the developers of the targeted software, web site, or web service either are unaware of or have just learned about. Security firms have therefore usually not developed defenses against these attacks, and even if they have, users have usually not obtained and installed the patches or performed the workarounds necessary to protect against these attacks. The time between discovery of a zero-day attack and availability of a defense (the vulnerability window) is shrinking, but perpetrators can still count on hours or even days in which many web sites and web services lack any specific protection against the attack.

Spear attacks are a major threat, but to a more select group of users. A common type of spear attack, a spear phish, is usually targeted at customers of a specific bank or financial institution, or (less commonly) at employees of a specific company or organization. Unlike other phishes, which are often crudely written forgeries that a user with any familiarity with the actual communications of that bank or financial institution can recognize, spear phishes are letter perfect and extremely convincing. They can contain information specific to the individual that, at first look, no stranger should know or be able to obtain. The spear phisher is therefore able to convince his or her target to provide the requested information, which the phisher can then use to loot accounts, to process illegitimately obtained money from other sources, or to gain access to other, even more sensitive information.

Both of these types of attack have certain characteristics that can usually be detected, although not by using static patterns that look for specific characteristics, as do standard signatures. Detecting these types of attacks requires more sophisticated and more resource-intensive approaches, such as heuristic filtering and positive security model systems. Heuristic filtering looks, not for specific patterns, but for patterns of behaviors. Positive security model systems model the normal behavior of the web site or web service that they are protecting, and then block connections that do not fit within that model of normal use. URL based and web-form based security checks profile normal use of your web sites, and then control how users interact with your web sites, using both heuristics and positive security to block anomalous or unexpected traffic. Both heuristic and positive security, properly designed and deployed, can catch most attacks that signatures miss. However, they require considerably more resources than do signatures, and you must spend some time configuring them properly to avoid false positives. They are therefore usually used, not as the primary line of defense, but as backups to signatures or other less resource-intensive approaches.

By configuring these advanced protections in addition to signatures, you create a hybrid security model, which enables the application firewall to provide comprehensive protection against both known and unknown attacks.

How The Application Firewall Works

When you install the application firewall, you create an initial security configuration, which consists of a policy, a profile, and a signatures object. The policy is a rule that identifies the traffic to be filtered, and the profile identifies the patterns and types of behavior to allow or block when the traffic is filtered. The simplest patterns, which are called signatures, are not specified within the profile, but in a signatures object that is associated with the profile.

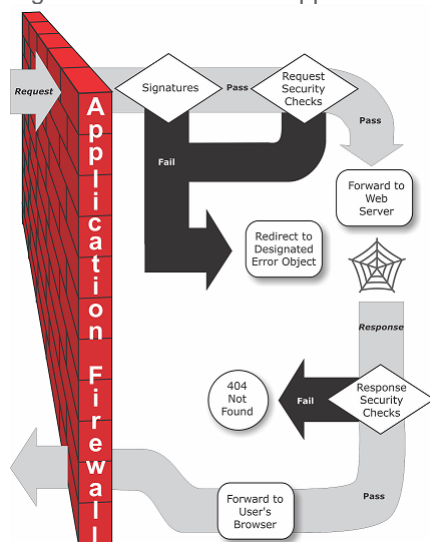
A signature is a string or pattern that matches a known type of attack. The application firewall contains over a thousand signatures in seven categories, each directed at attacks on specific types of web servers and web content. Citrix updates the list with new signatures as new threats are identified. During configuration, you specify the signature categories that are appropriate for the web servers and content that you need to protect. Signatures provide good basic protection with low processing overhead. If your applications have special vulnerabilities or you detect an attack against them for which no signature exists, you can add your own signatures.

The more advanced protections are called security checks. A security check is a more rigorous, algorithmic inspection of a request for specific patterns or types of behavior that might indicate an attack or constitute a threat to your protected web sites and web services. It can, for example, identify a request that attempts to perform a certain type of operation that might breach security, or a response that includes sensitive private information such as a social security number or credit card number. During configuration, you specify the security checks that are appropriate for the web servers and content that you need to protect. The security checks are restrictive. Many of them can block legitimate requests and responses if you do not add the appropriate exceptions (relaxations) when configuring them. Identifying the needed exceptions is not difficult if you use the adaptive learning feature, which observes normal use of your web site and creates recommended exceptions.

The application firewall can be installed as either a Layer 3 network device or a Layer 2 network bridge between your servers and your users, usually behind your company's router or firewall. It must be installed in a location where it can intercept traffic between the web servers that you want to protect and the hub or switch through which users access those web servers. You then configure the network to send requests to the application firewall instead of directly to your web servers, and responses to the application firewall instead of directly to your users. The application firewall filters that traffic before forwarding it to its final destination, using both its internal rule set and your additions and modifications. It blocks or renders harmless any activity that it detects as harmful, and then forwards the remaining traffic to the web server. The following figure provides an overview of the filtering process.

Note: The figure omits the application of a policy to incoming traffic. It illustrates a security configuration in which the policy is to process all requests. Also, in this configuration, a signatures object has been configured and associated with the profile, and security checks have been configured in the profile.

Figure 1. A Flowchart of Application Firewall Filtering



As the figure shows, when a user requests a URL on a protected web site, the application firewall first examines the request to ensure that it does not match a signature. If the request matches a signature, the application firewall either displays the error object (a web page that is located on the application firewall appliance and which you can configure by using the imports feature) or forwards the request to the designated error URL (the error page). Signatures do not require as many resources as do security checks, so detecting and stopping attacks that are detected by a signature before running any of the security checks reduces the load on the server.

If a request passes signature inspection, the application firewall applies the request security checks that have been enabled. The request security checks verify that the request is appropriate for your web site or web service and does not contain material that might pose a threat. For example, security checks examine the request for signs indicating that it might be of an unexpected type, request unexpected content, or contain unexpected and possibly malicious web form data, SQL commands, or scripts. If the request fails a security check, the application firewall either sanitizes the request

and then sends it back to the NetScaler appliance (or NetScaler virtual appliance), or displays the error object. If the request passes the security checks, it is sent back to the NetScaler appliance, which completes any other processing and forwards the request to the protected web server.

When the web site or web service sends a response to the user, the application firewall applies the response security checks that have been enabled. The response security checks examine the response for leaks of sensitive private information, signs of web site defacement, or other content that should not be present. If the response fails a security check, the application firewall either removes the content that should not be present or blocks the response. If the response passes the security checks, it is sent back to the NetScaler appliance, which forwards it to the user.

Application Firewall Features

Updated: 2013-09-03

The basic application firewall features are policies, profiles, and signatures, which provide a hybrid security model as described in "[Known Web Attacks](#)," "[Unknown Web Attacks](#)," and "[How the Application Firewall Works](#)." Of special note is the learning feature, which observes traffic to your protected applications and recommends appropriate configuration settings for certain security checks.

The imports feature manages files that you upload to the application firewall. These files are then used by the application firewall in various security checks, or when responding to a connection that matches a security check.

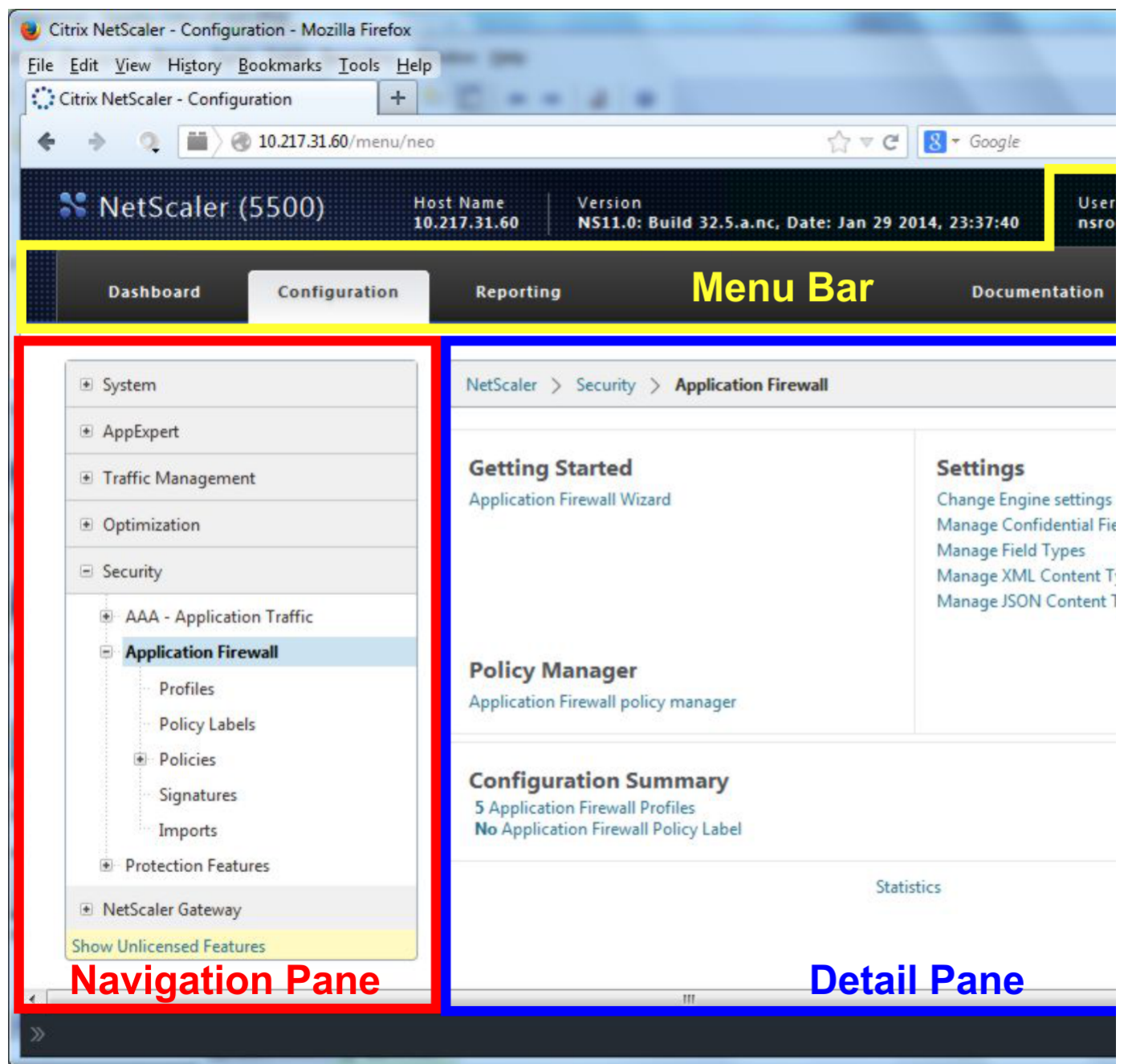
You can use the logs, statistics, and reports features to evaluate the performance of the application firewall and identify possible needs for additional protections.

The Application Firewall Configuration Interfaces

All hardware and virtual versions of the Citrix NetScaler application delivery controller (ADC) can be configured and managed from the Citrix NetScaler command line interface or the web-based configuration utility. All features of most NetScaler features can be configured using either of these tools. The Citrix Application Firewall is an exception: not all application firewall configuration tasks can be performed at the command line. Inexperienced users also find the configuration utility easier to use. In particular, the application firewall wizard considerably reduces the complexity of configuring the application firewall. Unlike most NetScaler wizards, the application firewall wizard can serve as your primary interface to the application firewall.

The command line interface is a modified UNIX shell based on the FreeBSD `bash` shell. To configure the application firewall from the command line interface, you type commands at the prompt and press the Enter key, just as you do with any other Unix shell. For instructions for using the command line interface, see "Command Reference."

The configuration utility is a web-based GUI interface to the ADC. The application firewall configuration section is found under Security > Application Firewall. Figure 1 shows the navigation pane expanded to display the application firewall screens, and in the detail pane the main application firewall screen.



The configuration utility has two main areas on all screens. The panel on the left, called the navigation pane, contains a navigation tree, with which you navigate to the screens on which you configure the features that are installed on your appliance. The screens to which you navigate appear to the right of the navigation pane, in the details pane.

When you access the configuration utility, the details pane displays the System Overview screen. If, in the navigation pane, you click plus sign next to the application firewall folder, the Application Firewall node expands to include the main application firewall elements that you can configure. If you click the first element, Profiles, the details pane displays the configured profiles, if any profiles have been configured. At the bottom of the details pane, you can click Add to configure a new profile. Other buttons at the bottom of the details pane are grayed out until you select an existing profile. Screens for the other elements work in the same way.

If, instead of expanding the application firewall node, you click the node itself, the details pane displays different options, one of which is the application firewall wizard, as shown in Figure 1. Citrix recommends that you use the wizard for initial configuration, and many users use it almost exclusively. It includes most of the functionality that is available elsewhere in the configuration utility.

For information and instructions on accessing the configuration utility, see "[Citrix NetScaler Getting Started Guide](#)."

Configuring the Application Firewall

You can configure the Citrix Application Firewall (application firewall) by using any of the following methods:

- **Application Firewall Wizard.** A dialog box consisting of a series of screens that step you through the configuration process.
- **Citrix Web Interface AppExpert Template.** A NetScaler AppExpert template (a set of configuration settings) that are designed to provide appropriate protection for web sites. This AppExpert template contains appropriate Application Firewall configuration settings for protecting many web sites.
- **Citrix NetScaler Configuration Utility.** The NetScaler web-based configuration interface.
- **Citrix NetScaler Command Line Interface.** The NetScaler command line configuration interface.

Citrix recommends that you use the Application Firewall Wizard. Most users will find it the easiest method to configure the application firewall, and it is designed to prevent mistakes. If you have a new Citrix NetScaler ADC or VPX that you will use primarily to protect web sites, you may find the Web Interface AppExpert template a better option because it provides a good default configuration, not just for the application firewall, but for the entire appliance. Both the configuration utility and the command line interface are intended for experienced users, primarily to modify an existing configuration or use advanced options.

The Application Firewall Wizard

The application firewall wizard is a dialog box that consists of several screens that prompt you to configure each part of a simple configuration. The application firewall then creates the appropriate configuration elements from the information that you give it. This is the simplest and, for most purposes, the best way to configure the application firewall.

To use the wizard, connect to the configuration utility with the browser of your choice. When the connection is established, verify that the application firewall is enabled, and then run the application firewall wizard, which prompts you for configuration information. You do not have to provide all of the requested information the first time you use the wizard. Instead, you can accept default settings, perform a few relatively straightforward configuration tasks to enable important features, and then allow the application firewall to collect important information to help you complete the configuration.

For example, when the wizard prompts you to specify a rule for selecting the traffic to be processed, you can accept the default, which selects all traffic. When it presents you with a list of signatures, you can enable the appropriate categories of signatures and turn on the collection of statistics for those signatures. For this initial configuration, you can skip the advanced protections (security checks). The wizard automatically creates the appropriate policy, signatures object, and profile (collectively, the security configuration) , and binds the policy to global. The application firewall then begins filtering connections to your protected web sites, logging any connections that match one or more of the signatures that you enabled, and collecting statistics about the connections that each signature matches. After the application firewall processes some traffic, you can run the wizard again and examine the logs and statistics to see if any of the signatures that you have enabled are matching legitimate traffic. After determining which signatures are identifying the traffic that you want to block, you can enable blocking for those signatures. If your web site or web service is not complex, does not use SQL, and does not have access to sensitive private information, this basic security configuration will probably provide adequate protection.

You may need additional protection if, for example, your web site is dynamic. Content that uses scripts may need protection against cross-site scripting attacks. Web content that uses SQL such as shopping carts, many blogs, and most content management systems may need protection against SQL injection attacks. Web sites and web services that collect sensitive private information such as social security numbers or credit card numbers may require protection against unintentional exposure of that information. Certain types of web-server or XML-server software may require protection from types of attacks tailored to that software. Another consideration is that specific elements of your web sites or web services may require different protection than do other elements. Examining the application firewall logs and statistics can help you identify the additional protections that you might need.

After deciding which advanced protections are needed for your web sites and web services, you can run the wizard again to configure those protections. Certain security checks require that you enter exceptions (relaxations) to prevent the check from blocking legitimate traffic. You can do so manually, but it is usually easier to enable the adaptive learning feature and allow it to recommend the necessary relaxations. You can use the wizard as many times as necessary to enhance your basic security configuration and/or create additional security configurations.

The wizard automates some tasks that you would have to perform manually if you did not use the wizard. It automatically creates a policy, a signatures object, and a profile, and assigns them the name that you provided when you were prompted for the name of your configuration. The wizard also adds your advanced-protection settings to the profile, binds the signatures object to the profile, associates the profile with the policy, and puts the policy into effect by binding it to Global.

A few tasks cannot be performed in the wizard. You cannot use the wizard to bind a policy to a bind point other than Global. If you want the profile to apply to only a specific part of your configuration, you must manually configure the binding. You cannot configure the engine settings or certain other global configuration options in the wizard. While you can configure any of the advanced protection settings in the wizard, if you want to modify a specific setting in a single security check, it may be easier to do so on the manual configuration screens in the configuration utility.

For more information on using the Application Firewall Wizard, see ["The Application Firewall Wizard."](#)

The Citrix Web Interface AppExpert Template

AppExpert Templates are a different and simpler approach to configuring and managing complex enterprise applications. The AppExpert display in the configuration utility consists of a table. Applications are listed in the left-most column, with the NetScaler features that are applicable to that application appearing each in its own column to the right. (In the AppExpert interface, those features that are associated with an application are called *application units*.) In the AppExpert interface, you configure the interesting traffic for each application, and turn on rules for compression, caching, rewrite, filtering, responder and the application firewall, instead of having to configure each feature individually.

The Web Interface AppExpert Template contains rules for the following application firewall signatures and security checks:

- o **"Deny URL check."** Detects connections to content that is known to pose a security risk, or to any other URLs that you designate.
- o **"Buffer Overflow check."** Detects attempts to cause a buffer overflow on a protected web server.
- o **"Cookie Consistency check."** Detects malicious modifications to cookies set by a protected web site.
- o **"Form Field Consistency check."** Detects modifications to the structure of a web form on a protected web site.
- o **"CSRF Form Tagging check."** Detects cross-site request forgery attacks.
- o **"Field Formats check."** Detects inappropriate information uploaded in web forms on a protected web site.
- o **"HTML SQL Injection check."** Detects attempts to inject unauthorized SQL code.
- o **"HTML Cross-Site Scripting check."** Detects cross-site scripting attacks.

For information on installing and using an AppExpert Template, see ["AppExpert Applications and Templates."](#)

The Citrix NetScaler Configuration Utility

The NetScaler configuration utility is a web-based interface that provides access to all configuration options for the application firewall feature, including advanced configuration and management options that are not available from any other configuration tool or interface. Specifically, many advanced Signatures options can be configured only in the configuration utility. You can review recommendations generated by the learning feature only in the configuration utility. You can bind policies to a bind point other than Global only in the configuration utility.

For a description of the configuration utility, see ["The Application Firewall Configuration Interfaces."](#) For more information on using the configuration utility to configure the application firewall, see ["Manual Configuration By Using the Configuration Utility."](#)

For instructions on configuring the application firewall by using the configuration utility, see ["Manual Configuration By Using the Configuration Utility."](#) For information on the Citrix NetScaler Configuration Utility, see ["The Application Firewall Configuration Interfaces."](#)

The Citrix NetScaler Command Line Interface

The Citrix NetScaler command line interface is a modified UNIX shell based on the FreeBSD `bash` shell. To configure the Application Firewall from the command line interface, you type commands at the prompt and press the Enter key, just as you do with any other Unix shell. You can configure most parameters and options for the application firewall by using the NetScaler command line. Exceptions are the signatures feature, many of whose options can be configured only by using the configuration utility or the Application Firewall wizard, and the learning feature, whose recommendations can only be reviewed in the configuration utility.

For instructions on configuring the application firewall by using the NetScaler command line, see ["Manual Configuration By Using the Command Line Interface."](#)

Enabling the Application Firewall

Before you can create an application firewall security configuration, you must make sure that the application firewall feature is enabled.

- If you are configuring a dedicated Citrix Application Firewall ADC or are upgrading an existing Citrix NetScaler ADC or VPX, the feature is already enabled. You do not have to perform either of the procedures described here.
- If you have a new NetScaler ADC or VPX, you need to enable the application firewall feature before you configure it.
- If you are upgrading a NetScaler ADC or VPX from a previous version of the NetScaler operating system to the current version, you might need to enable the application firewall feature before you configure it.

Note: If you are upgrading a NetScaler ADC or VPX from a previous version, you might also need to update the licenses on your ADC or VPX before you can enable the application firewall. Check with your Citrix representative or reseller to obtain the correct license.

You can enable the application firewall by using the command line or the configuration utility.

To enable the application firewall by using the command line interface

At the command prompt, type the following command:

```
enable ns feature AppFW
```

To enable the application firewall by using the configuration utility

1. Navigate to System > Settings.
2. In the details pane, click Configure Basic Features.
3. In the Configure Basic Features dialog box, check the Application Firewall check box.
4. Click OK.

The Application Firewall Wizard

Unlike most wizards, the Application Firewall wizard is designed not just to simplify the initial configuration process, but also to modify previously created configurations and to maintain your Application Firewall setup. A typical user runs the wizard multiple times, skipping some of the screens each time.

Opening the Wizard

To run the Application Firewall wizard, first open the configuration utility. Next, in the navigation pane, expand Application Firewall, and then in the details pane click Application Firewall Wizard. (For more information about the configuration utility, see "[The Application Firewall User Interfaces](#).") Then:

1. Navigate to Security > Application Firewall.
2. In the details pane, under Getting Started, click Application Firewall Wizard. The first screen of the wizard appears.
3. To advance to the next screen, click Next.

The Wizard Screens

The Application Firewall wizard displays the following screens, in the following order:

1. **Introduction screen.** Provides an introduction to the Application Firewall wizard. There is nothing that you can configure on this screen.
2. **Specify Name screen.** On this screen, when creating a new security configuration, you specify the name that the wizard is to assign to the configuration. The name can begin with a letter, number, or the underscore symbol, and can consist of from 1 to 31 letters, numbers, and the hyphen (-), period (.), pound (#), space (), at (@), equals (=), colon (:), and underscore (_) symbols. Choose a name that makes it easy for others to tell what content your new security configuration protects.

Note: Because the wizard uses this name for both the policy and the profile, it is limited to 31 characters. Manually created policies can have names up to 127 characters in length.

When creating an existing configuration, you select Modify Existing Configuration and then, in the Name drop-down list, select the name of the existing configuration that you want to modify.

Note: Only policies that are bound to global or to a bind point appear in this list; you cannot modify an unbound policy by using the Application Firewall wizard. You must either manually bind it to Global or a bind point, or modify it manually. (For manual modification, in the configuration utility's Application Firewall --> Policies --> Firewall pane, select the policy and click Open).

You also select a profile type on this screen. The profile type determines the types of advanced protection (security checks) that can be configured. Because certain kinds of content are not vulnerable to certain types of security threats, restricting the list of available checks saves time during configuration. The types of Application Firewall profiles are:

- **Web Application (HTML).** Any HTML-based Web site that does not use XML or Web 2.0 technologies.
- **XML Application (XML, SOAP).** Any XML-based Web service.
- **Web 2.0 Application (HTML, XML, REST).** Any Web 2.0 site that combines HTML and XML-based content, such as an ATOM-based site, a blog, an RSS feed, or a wiki.

Note: If you are unsure which type of content is used on your Web site, you can choose Web 2.0 Application to ensure that you protect all types of Web application content.

3. **Specify Rule screen.** On this screen, you specify the policy rule (*expression*) that defines the traffic to be examined by this security configuration. If you are creating an initial configuration to protect your Web sites and Web services, you can simply accept the default value, `true`, which selects all web traffic .

If you want this security configuration to examine, not all HTTP traffic that is routed through the appliance, but specific traffic, you can write a policy rule specifying the traffic that you want it to examine. Rules are written in Citrix NetScaler expressions language, which is a fully functional object-oriented programming language.

- For a simple description of using the NetScaler expressions syntax to create Application Firewall rules, and a list of useful rules, see "[Firewall Policies](#)."
- For a detailed explanation of how to create policy rules in NetScaler expressions syntax, see "[Policies and Expressions](#)."

Note: In addition to the default expressions syntax, for backward compatibility the NetScaler operating system supports the NetScaler classic expressions syntax on NetScaler Classic and nCore appliances and virtual appliances.

Classic expressions are not supported on NetScaler Cluster appliances and virtual appliances. Current users who want to migrate their existing configurations to the NetScaler cluster must migrate any policies that contain classic expressions to the default expressions syntax.

4. **Select Signature Protections screen.** On this screen, you select the categories of signatures that you want to use to protect your web sites and web services. The default categories are:

- **CGI.** Protection against attacks on web sites that use CGI scripts in any language, including PERL scripts, Unix shell scripts, and Python scripts.
- **Cold Fusion.** Protection against attacks on web sites that use the Adobe Systems® ColdFusion® Web development platform.
- **FrontPage.** Protection against attacks on web sites that use the Microsoft® FrontPage® Web development platform.
- **PHP.** Protection against attacks on web sites that use the PHP open-source Web development scripting language.
- **Client side.** Protection against attacks on client-side tools used to access your protected web sites, such as Microsoft Internet Explorer, Mozilla Firefox, the Opera browser, and the Adobe Acrobat Reader.
- **Microsoft IIS.** Protection against attacks on Web sites that run the Microsoft Internet Information Server (IIS).
- **Miscellaneous.** Protection against attacks on other server-side tools, such as Web servers and database servers.

If you are creating a new security configuration, the signature categories that you select are enabled, and by default they are recorded in a new signatures object. The new signatures object is assigned the same name that you entered on the Specify name screen as the name of the security configuration.

If you have previously configured signatures objects and want to use one of them as the signatures object associated with the security configuration that you are creating, click Select Existing Signature and select a signatures object from the Signatures list.

If you are modifying an existing security configuration, you can click Select Existing Signature and assign a different signatures object to the security configuration.

5. **Select Signature Actions screen.** On this screen, you select the actions associated with the signature categories that you selected on the Select signature protections screen. If you are creating an initial configuration, you might want to accept the defaults, which enable the Log and Stats actions but not the Block action. You can decide later, after reviewing the collected logs and statistics, which signatures you should use to block traffic, and then enable the Block action for those signatures. Signatures are designed to catch specific known attacks on your web sites, and therefore they have extremely low false positive rates. However, with any new configuration, you should probably observe how the settings you chose are working before you use them to block traffic.

If you select More for one of the signature categories, the Configure Actions for Signatures dialog box appears. Its contents are the same as the contents of the Modify Signatures Object dialog box, as described in "[To Configure a Signatures Object](#)."

If the signatures object has already logged connections, you can click Logs to display the Syslog Viewer with the logs, as described in "[Logs, Statistics, and Reports](#)." If a signature is blocking legitimate access to your protected web site or web service, you can create and implement a relaxation for that signature by selecting a log that shows the unwanted blocking, and then clicking Deploy.

6. **Select Advanced Protections screen.** On this screen, you choose the advanced protections (also called *security checks* or simply *checks*) that you want to use to protect your web sites and web services. The checks are divided into categories. Which categories are available (and which checks are available within a category) depends on the profile type that you chose on the Specify Name screen. All checks are available for Web 2.0 Application profiles. If you chose that profile type, the Select advanced protections screen displays the following categories of security checks:

- Top--level protections (Some checks appear at the top level, not in any category.)
- Data Leak Prevention Protections
- Advanced Form Protections
- URL Protections
- XML Protections

To display the individual checks in a category, click the icon to the left of the category. To apply a security check to your filtered data, select the check box next to the name of the security check. For descriptions of the security checks see ["Advanced Protections"](#) and its subtopics.

7. **Select Advanced Actions screen.** On this screen, you configure the actions for the advanced protections that you have enabled.

Note: If no advanced protections are enabled, the Wizard skips the Advanced Actions screen and goes directly to the Summary screen.

The actions that you can configure are:

- o **Block.** Block connections that match the signature. Disabled by default.
- o **Log.** Log connections that match the signature for later analysis. Enabled by default.
- o **Stats.** Maintain statistics, for each signature, that show how many connections it matched and provide certain other information about the types of connections that were blocked. Disabled by default.
- o **Learn.** Observe traffic to this Web site or Web service, and use connections that repeatedly violate this check to generate recommended exceptions to the check, or new rules for the check. Available only for some checks.

To enable or disable an action for a check, in the list, select or clear the check box for that action to the right of that check.

To configure other parameters for those checks that have them, in the list, click the blue chevron to the far right of that check. In the dialog box that appears, configure the parameters. These vary from check to check. You can also select a check and, at the bottom of the dialog box, click Open to display a dialog box for modifying any of the options for that check. These dialog boxes also vary from check to check. Most of them include a Checks tab and a General tab. If the check supports relaxations, the Checks tab includes an Add button, which opens yet another dialog box, in which you can specify a relaxation for the check. A relaxation is a rule for exempting specified traffic from the check.

For information about the settings available for a check, see the detailed description of that check.

To review the recommendations generated by the learning engine for a specific check, select that check and then click Learned Violations to open the Manage Learned Rules dialog box for that check. For more information on how learning works and how to configure exceptions (relaxations) or deploy learned rules for a check, see ["Manual Configuration By Using the Configuration Utility"](#) under To configure and use the learning feature

To view all logs for a specific check, select that check, and then click Logs to display the Syslog Viewer, as described in ["Logs, Statistics, and Reports."](#) If a security check is blocking legitimate access to your protected web site or web service, you can create and implement a relaxation for that security check by selecting a log that shows the unwanted blocking, and then clicking Deploy.

8. **Summary screen.** On this screen, you review your configuration choices to verify that they are what you want. If you want to make changes, you click Back until you have returned to the appropriate screen, and make your changes. If the configuration is as you want it, you click Finish to save it, and then click Exit to close the Application Firewall wizard.

Following are four procedures that show how to perform specific types of configuration by using the Application Firewall wizard.

To configure the Application Firewall: Initial Configuration

1. Navigate to Security > Application Firewall.
2. In the details pane, under Getting Started, click Application Firewall Wizard.
3. On the Application Firewall wizard, Introduction screen, in the lower right-hand corner, click Next.
4. On the Specify Name screen, in the Name text box, type a name for your new security configuration, and from the Type drop-down list, select the type of security configuration. Then, click Next.
5. On the Specify Rule screen, click Next again.
Note: The default rule, `true`, protects all Web traffic that is sent via your NetScaler appliance or virtual appliances. You can create specific security configurations to protect specific parts of your Web sites or Web applications later.
6. On the Select Signature Protections screen, select check boxes to specify the groups of signatures that are appropriate for protecting the content on your protected web sites, and then click Next.

For more information about signatures, see ["Signatures."](#)

7. On the Select Signature Actions screen, select or clear the associated check boxes to choose the signature actions that you want for each signature category that you selected in the previous step, and then click Next.
8. On the Select Advanced Protections screen, click Next again.

You typically do not need to configure the security checks during initial configuration.

9. On the Summary screen, review your choices to verify that they are what you want. Then, click Finish, or click Back to return to a previous screen and make changes. When you are finished, click Exit to close the Application Firewall wizard.

To configure the Application Firewall: Enabling Blocking for Signatures

1. Navigate to Security > Application Firewall.
2. In the details pane, under Getting Started, click Application Firewall Wizard.
3. On the Application Firewall wizard, Introduction screen, in the lower right-hand corner, click Next.
4. On the Specify Name screen, select Modify Existing Configuration and, in the Name drop-down list, choose the security configuration that you created during simple configuration, and then click Next.
5. In the Specify Rule screen, click Next again.
6. In the Select Signature Protections screen, click Next again.
7. In the Select Signature Actions screen, enable blocking for your chosen signatures by selecting the Block check box to the left of each of those signature.

For more information about which signatures to consider for blocking and how to determine when you can safely enable blocking for a signature, see "[Signatures](#)."

8. In the Select advanced protections screen, click Next.
9. On the Summary screen, review your choices to verify that they appear correct. Then, click Finish, or click Back to return to the Select Signature Actions screen and make changes. When you are finished, click Exit to close the Application Firewall wizard.

To configure the Application Firewall: Enabling and Configuring advanced protection

1. Navigate to Security > Application Firewall.
2. In the details pane, under Getting Started, click Application Firewall Wizard.
3. On the Application Firewall wizard, Introduction screen, in the lower right-hand corner, click Next.
4. On the Specify Name screen, select Modify Existing Configuration and, in the Name drop-down list, choose the security configuration that you created during simple configuration. Then, click Next.
5. On the Specify Rule screen, click Next again.
6. On the Select Signature Protections screen, click Next.
7. On the Select Signature Actions screen, click Next again.
8. On the Select advanced protections screen, select the check box beside each security check that you want to enable, and then click Next.

For information about the security checks, see "[Advanced Protections](#)" and its subtopics.

9. On the Select Deep Actions screen, select check boxes to specify the actions that you want the Application Firewall to perform for each security check, and then click Next.

For general information about the actions, see "[Advanced Protections](#)" and its subtopics. For information about the learning feature, which is available for some security checks, see "[To configure and use the Learning feature](#)".

10. On the Summary screen, review your choices to verify that they appear correct. Then, click Finish, or click Back to return to the Select Signature Actions screen and make changes. When you are finished, click Exit to close the Application Firewall wizard.

To configure the Application Firewall: Creating A Policy

The following procedure describes how to use the Application Firewall wizard to create a specialized security configuration to protect only specific content. In this case, you create a new security configuration instead of modifying the initial configuration. This type of security configuration requires a custom rule, so that the policy applies the configuration to only the selected Web traffic.

1. Navigate to Security > Application Firewall.
2. In the details pane, under Getting Started, click Application Firewall Wizard.
3. On the Application Firewall wizard, Introduction screen, in the lower right-hand corner, click Next.
4. On the Specify Name screen, type a name for your new security configuration in the Name text box, select the type of security configuration from the Type drop-down list, and then click Next.
5. On the Specify Rule screen, enter a rule that matches only that content that you want this Web application to protect, and then click Next.

For a description of policies and policy rules, see "[Policies](#)."

6. On the Select Signature Protections screen, choose the appropriate groups of signatures to protect the content on your protected web sites by selecting the check box beside each group of signatures, and then click Next.

For detailed information about signatures, see "[Signatures](#)."

7. On the Select Signature Actions screen, select or clear the associated check boxes to choose the signature actions that you want for each signature category that you selected in the previous step, and then click Next. For a detailed description of actions, see "[Signatures](#)."
8. In the Select Advanced Protections screen, select the check box beside each security check that you want to enable, and then click Next.

For detailed information about the security checks, see "[Advanced Protections](#)" and its subtopics.

9. In the Select Advanced Actions screen, select check boxes to specify the actions that you want the Application Firewall to perform for each security check. Then, click Next.

For information about each security check to help you determine which actions to enable, see the Advanced Protections section.

10. On the Summary screen, review your choices to verify that they appear correct. Then, click Finish, or click Back to return to the Select Signature Actions screen and make changes. When you are finished, click Exit to close the wizard.

Manual Configuration

If you want to bind a profile to a bind point other than Global, you must manually configure the binding. Also, certain security checks require that you either manually enter the necessary exceptions or enable the learning feature to generate the exceptions that your Web sites and Web services need. Some of these tasks cannot be performed by using the application firewall wizard.

If you are familiar with how the application firewall works and prefer manual configuration, you can manually configure a signatures object and a profile, associate the signatures object with the profile, create a policy with a rule that matches the web traffic that you want to configure, and associate the policy with the profile. You then bind the policy to Global, or to a bind point, to put it into effect, and you have created a complete security configuration.

For manual configuration, you can use the configuration utility (a graphical interface) or the command line. Citrix recommends that you use the configuration utility. Not all configuration tasks can be performed at the command line. Certain tasks, such as enabling signatures and reviewing learned data, must be done in the configuration utility. Most other tasks are easier to perform in the configuration utility.

Manual Configuration By Using the Configuration Utility

If you need to configure the Application Firewall feature manually, Citrix recommends that you use the configuration utility. For a description of the configuration utility, see ["The Application Firewall User Interfaces."](#)

To create and configure a signatures object

Before you can configure the signatures, you must create a new signatures object from the appropriate default signatures object template. Assign the copy a new name, and then configure the copy. You cannot configure or modify the default signatures objects directly. The following procedure provides basic instructions for configuring a signatures object. For more detailed instructions, see ["Manually Configuring the Signatures Feature."](#) If you need to create your own, user defined signatures, see ["The Signatures Editor."](#)

1. Navigate to Security > Application Firewall > Signatures.
2. In the details pane, select the signatures object that you want to use as a template, and then click Add.

Your choices are:

- o *** Default Signatures.** Contains the signatures rules, the SQL injection rules, and the cross-site scripting rules.
 - o *** XPath Injection.** Contains all of the items in the * Default Signatures, and in addition contains the XPath injection rules.
3. In the Add Signatures Object dialog box, type a name for your new signatures object, click OK, and then click Close. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 31 letters, numbers, and the hyphen (-), period (.) pound (#), space (), at (@), equals (=), and underscore (_) symbols.
 4. Select the signatures object that you created, and then click Open.
 5. In the Modify Signatures Object dialog box, set the Display Filter Criteria options at the left to display the filter items that you want to configure.

As you modify these options, the results that you specify are displayed in the Filtered Results window at the right. For more information about the categories of signatures, see ["Signatures."](#)

6. In the Filtered Results area, configure the settings for a signature by selecting and clearing the appropriate check boxes.
7. When finished, finished, click Close.

To create an application firewall profile by using the configuration utility

Creating an application firewall profile requires that you specify only a few configuration details.

1. Navigate to Security > Application Firewall > Profiles.
2. In the details pane, click Add.
3. In the Create Application Firewall Profile dialog box, type a name for your profile.

The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 31 letters, numbers, and the hyphen (-), period (.) pound (#), space (), at (@), equals (=), colon (:), and underscore (_) symbols.

4. Choose the profile type from the drop-down list.
5. Click Create, and then click Close.

To configure an application firewall profile by using the configuration utility

1. Navigate to Security > Application Firewall > Profiles.
2. In the details pane, select the profile that you want to configure, and then click Edit.
3. In the Configure Application Firewall Profile dialog box, on the Security Checks tab, configure the security checks.
 - o To enable or disable an action for a check, in the list, select or clear the check box for that action.
 - o To configure other parameters for those checks that have them, in the list, click the blue chevron to the far right of that check. In the dialog box that appears, configure the parameters. These vary from check to check.

You can also select a check and, at the bottom of the dialog box, click Open to display the Configure Relaxation dialog box or Configure Rule dialog box for that check. These dialog boxes also vary from check to check. Most of them include a Checks tab and a General tab. If the check supports relaxations or user-defined rules, the Checks tab includes an Add button, which opens yet another dialog box, in which you can specify a relaxation or rule for the check. (A relaxation is a rule for exempting specified traffic from the check.) If relaxations have already been configured, you can select one and click Open to modify it.

- To review learned exceptions or rules for a check, select the check, and then click Learned Violations. In the Manage Learned Rules dialog box, select each learned exception or rule in turn.
 - To edit the exception or rule, and then add it to the list, click Edit & Deploy.
 - To accept the exception or rule without modification, click Deploy.
 - To remove the exception or rule from the list, click Skip.
 - To refresh the list of exceptions or rules to be reviewed, click Refresh.
 - To open the Learning Visualizer and use it to review learned rules, click Visualizer.
 - To review the log entries for connections that matched a check, select the check, and then click Logs. You can use this information to determine which checks are matching attacks, so that you can enable blocking for those checks. You can also use this information to determine which checks are matching legitimate traffic, so that you can configure an appropriate exemption to allow those legitimate connections. For more information about the logs, see "[Logs, Statistics, and Reports](#)."
 - To completely disable a check, in the list, clear all of the check boxes to the right of that check.
4. On the Settings tab, configure the profile settings.
- To associate the profile with the set of signatures that you previously created and configured, under Common Settings, choose that set of signatures in the Signatures drop-down list.
Note: You may need to use the scroll bar on the right of the dialog box to scroll down to display the Common Settings section.
 - To configure an HTML or XML Error Object, select the object from the appropriate drop-down list.
Note: You must first upload the error object that you want to use in the Imports pane. For more information about importing error objects, see "[Imports](#)."
 - To configure the default XML Content Type, type the content type string directly into the Default Request and Default Response text boxes, or click Manage Allowed Content Types to manage the list of allowed content types. "[>>More...](#)"
5. If you want to use the learning feature, click Learning, and configure the learning settings for the profile, as described in "[Configuring and Using the Learning Feature](#)".
6. Click OK to save your changes and return to the Profiles pane.

Configuring an Application Firewall Rule or Relaxation

Updated: 2014-06-12

You configure two different types of information in this dialog box, depending upon which security check you are configuring. In the majority of cases, you configure an exception (or relaxation) to the security check. If you are configuring the Deny URL check or the Field Formats check, you configure an addition (or rule). The process for either of these is the same.

To configure a relaxation or rule by using the configuration utility

1. Navigate to Security > Application Firewall > Profiles.
2. In the Profiles pane, select the profile you want to configure, and then click Edit.
3. In the Configure Application Firewall Profile dialog box, click the Security Checks tab. The Security Checks tab contains the complete list of application firewall security checks, also called *advanced protections* in some places.
4. In the Security Checks tab, click the check that you want to configure, and then click Open. The Modify Check dialog box for the check that you chose is displayed, with the Checks tab selected. The Checks tab contains a list of existing relaxations or rules for this check. The list might be empty if you have not either manually added any relaxations or approved any relaxations that were recommended by the learning engine. Beneath the list is a row of buttons that allow you to add, modify, delete, enable, or disable the relaxations on the list.
5. To add or modify a relaxation or a rule, do one of the following:
 - To add a new relaxation, click Add.
 - To modify an existing relaxation, select the relaxation that you want to modify, and then click Open.

The Add Check Relaxation or Modify Check Relaxation dialog box for the selected check is displayed. Except for the title, these dialog boxes are identical.

6. Fill in the dialog box as described below. The dialog boxes for each check are different; the list below covers all elements that might appear in any dialog box.

- **Enabled check box**—Select to place this relaxation or rule in active use; clear to deactivate it.
- **Attachment Content Type**—The Content-Type attribute of an XML attachment. In the text area, enter a regular expression that matches the Content-Type attribute of the XML attachments to allow.
- **Action URL**—In the text area, enter a PCRE-format regular expression that defines the URL to which data entered into the web form is delivered.

- **Cookie**—In the text area, enter a PCRE-format regular expression that defines the cookie.
- **Field Name**—A web form field name element may be labeled Field Name, Form Field, or another similar name. In the text area, enter a PCRE-format regular expression that defines the name of the form field.
- **Form Origin URL**—In the text area, enter a PCRE-format regular expression that defines the URL that hosts the web form.
- **Form Action URL**—In the text area, enter a PCRE-format regular expression that defines the URL to which data entered into the web form is delivered.
- **Name**—An XML element or attribute name. In the text area, enter a PCRE-format regular expression that defines the name of the element or attribute.
- **URL**—A URL element may be labeled Action URL, Deny URL, Form Action URL, Form Origin URL, Start URL, or simply URL. In the text area, enter a PCRE-format regular expression that defines the URL.
- **Format**—The format section contains multiple settings that include list boxes and text boxes. Any of the following can appear:
 - **Type**—Select a field type in the Type drop-down list. To add a new field type definition, click Manage
 - **Minimum Length**—Type a positive integer that represents the minimum length in characters if you want to force users to fill in this field. Default: 0 (Allows field to be left blank.)
 - **Maximum length**—To limit the length of data in this field, type a positive integer that represents the maximum length in characters. Default: 65535
- **Location**—Choose the element of the request that your relaxation will apply to from the drop-down list. For HTML security checks, the choices are:
 - **FORMFIELD**—Form fields in web forms.
 - **HEADER**—Request headers.
 - **COOKIE**—Set-Cookie headers.

For XML security checks, the choices are:

 - **ELEMENT**—XML element.
 - **ATTRIBUTE**—XML attribute.

- **Maximum Attachment Size**—The maximum size in bytes allowed for an XML attachment.
- **Comments**—In the text area, type a comment. Optional.

Note: For any element that requires a regular expression, you can type the regular expression, use the Regex Tokens menu to insert regular expression elements and symbols directly into the text box, or click Regex Editor to open the Add Regular Expression dialog box, and use it to construct the expression.

7. To remove a relaxation or rule, select it, and then click Remove.
 8. To enable a relaxation or rule, select it, and then click Enable.
 9. To disable a relaxation or rule, select it, and then click Disable.
 10. To configure the settings and relationships of all existing relaxations in an integrated interactive graphic display, click Visualizer, and use the display tools.
- Note: The Visualizer button does not appear on all check relaxation dialog boxes.
11. To review learned rules for this check, click Learning and perform the steps in ["To configure and use the Learning feature."](#)
 12. Click OK.

To configure the Learning feature by using the configuration utility

1. Navigate to Security > Application Firewall > Profiles.
2. In the Profiles pane, select the profile, and then click Edit.
3. Click the Learning tab. At the top of the Learning tab is list of the security checks that are available in the current profile and that support the learning feature.
4. To configure the learning thresholds, select a security check, and then type the appropriate values in the following text boxes:

- o **Minimum number threshold.** Depending on which security checkâ€™s learning settings you are configuring, the minimum number threshold might refer to the minimum number of total user sessions that must be observed, the minimum number of requests that must be observed, or the minimum number of times a specific form field must be observed, before a learned relaxation is generated. Default: 1
 - o **Percentage of times threshold.** Depending on which security checkâ€™s learning settings you are configuring, the percentage of times threshold might refer to the percentage of total observed user sessions that violated the security check, the percentage of requests, or the percentage of times a form field matched a particular field type, before a learned relaxation is generated. Default: 0
5. To remove all learned data and reset the learning feature, so that it must start its observations again from the beginning, click Remove All Learned Data.
Note: This button removes only learned recommendations that have not been reviewed and either approved or skipped. It does not remove learned relaxations that have been accepted and deployed.
 6. To restrict the learning engine to traffic from a specific set of IPs, click Trusted Learning Clients, and add the IP addresses that you want to use to the list.
 - a. To add an IP address or IP address range to the Trusted Learning Clients list, click Add.
 - b. In the Add Trusted Learning Clients dialog box, Trusted Clients IP list box, type the IP address or an IP address range in CIDR format.
 - c. In the Comments text area, type a comment that describes this IP address or range.
 - d. Click Create to add your new IP address or range to the list.
 - e. To modify an existing IP address or range, click the IP address or range, and then click Open. Except for the name, the dialog box that appears is identical to the Add Trusted Learning Clients dialog box.
 - f. To disable or enable an IP address or range, but leave it on the list, click the IP address or range, and then click Disable or Enable, as appropriate.
 - g. To remove an IP address or range completely, click the IP address or range, and then click Remove.
 7. Click Close to return to the Configure Application Firewall Profile dialog box.
 8. Click Close to close the Configure Application Firewall Profile dialog box, and return to the Application Firewall Profile screen.

To create and configure a policy by using the configuration utility

1. Navigate to Security > Application Firewall > Policies.
2. In the details pane, do one of the following:
 - o To create a new firewall policy, click Add. The Create Application Firewall Policy is displayed.
 - o To edit an existing firewall policy, select the policy, and then click Edit.
 The Create Application Firewall Policy or Configure Application Firewall Policy is displayed.
3. If you are creating a new firewall policy, in the Create Application Firewall Policy dialog box, Policy Name text box, type a name for your new policy.

The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 128 letters, numbers, and the hyphen (-), period (.) pound (#), space (), at (@), equals (=), colon (:), and underscore (_) symbols.

If you are configuring an existing firewall policy, this field is read-only. You cannot modify it.

4. Select the profile that you want to associate with this policy from the Profile drop-down list. You can create a new profile to associate with your policy by clicking New, and you can modify an existing profile by clicking Modify.
5. In the Expression text area, create a rule for your policy.
 - o You can type a rule directly into the text area.
 - o You can click Prefix to select the first term for your rule, and follow the prompts. See ["To Create an Application Firewall Rule \(Expression\)"](#) for a complete description of this process.
 - o You can click Add to open the Add Expression dialog box, and use it to construct the rule. See ["The Add Expression Dialog Box"](#) for a complete description of this process.
6. Click Create or OK, and then click Close.

To create or configure an Application Firewall rule (expression)

The policy rule, also called the *expression*, defines the web traffic that the application firewall filters by using the profile associated with the policy. Like other NetScaler policy rules (or *expressions*), application firewall rules use NetScaler expressions syntax. This syntax is powerful, flexible, and extensible. It is too complex to describe completely in this set of instructions. You can use the following procedure to create a simple firewall policy rule, or you can read it as an overview of the policy creation process.

1. If you have not already done so, navigate to the appropriate location in the Application Firewall wizard or the NetScaler configuration utility to create your policy rule:
 - o If you are configuring a policy in the Application Firewall wizard, in the navigation pane, click Application Firewall, then in the details pane click Application Firewall Wizard, and then navigate to the Specify Rule screen.
 - o If you are configuring a policy manually, in the navigation pane, expand Application Firewall, then Policies, and then Firewall. In the details pane, to create a new policy, click Add. To modify an existing policy, select the policy, and then click Open.
2. On the Specify Rule screen, the Create Application Firewall Profile dialog box, or the Configure Application Firewall Profile dialog box, click Prefix, and then choose the prefix for your expression from the drop-down list. Your choices are:
 - o **HTTP.** The HTTP protocol. Choose this if you want to examine some aspect of the request that pertains to the HTTP protocol.
 - o **SYS.** The protected Web site(s). Choose this if you want to examine some aspect of the request that pertains to the recipient of the request.
 - o **CLIENT.** The computer that sent the request. Choose this if you want to examine some aspect of the sender of the request.
 - o **SERVER.** The computer to which the request was sent. Choose this if you want to examine some aspect of the recipient of the request.

After you choose a prefix, the application firewall displays a two-part prompt window that displays the possible next choices at the top, and a brief explanation of what the selected choice means at the bottom.

3. Choose your next term.

If you chose HTTP as your prefix, your only choice is REQ, which specifies the Request/Response pair. (The application firewall operates on the request and response as a unit instead of on each separately.) If you chose another prefix, your choices are more varied. For help on a specific choice, click that choice once to display information about it in the lower prompt window.

When you have decided which term you want, double-click it to insert it into the Expression window.

4. Type a period after the term you just chose. You are then prompted to choose your next term, as described in the previous step. When a term requires that you type a value, fill in the appropriate value. For example, if you choose HTTP.REQ.HEADER(" "), type the header name between the quotation marks.
5. Continue choosing terms from the prompts and filling in any values that are needed, until your expression is finished.

Following are some examples of expressions for specific purposes.

- o **Specific web host.** To match traffic from a particular web host:

```
HTTP.REQ.HEADER("Host").EQ("shopping.example.com")
```

For shopping.example.com, substitute the name of the web host that you want to match.

- o **Specific web folder or directory.** To match traffic from a particular folder or directory on a Web host:

```
HTTP.REQ.URL.STARTSWITH("https://www.example.com/folder")
```

For www.example.com, substitute the name of the web host. For folder, substitute the folder or path to the content that you want to match. For example, if your shopping cart is in a folder called /solutions/orders, you substitute that string for folder.

- o **Specific type of content: GIF images.** To match GIF format images:

```
HTTP.REQ.URL.ENDSWITH(".gif")
```

To match other format images, substitute another string in place of .gif.

- o **Specific type of content: scripts.** To match all CGI scripts located in the CGI-BIN directory:

```
HTTP.REQ.URL.STARTSWITH("https://www.example.com/CGI-BIN")
```

To match all JavaScripts with .js extensions:

```
HTTP.REQ.URL.ENDSWITH(".js")
```

For more information about creating policy expressions, see "Policies and Expressions."

Note: If you use the command line to configure a policy, remember to escape any double quotation marks within NetScaler expressions. For example, the following expression is correct if entered in the configuration utility:

```
HTTP.REQ.HEADER("Host").EQ("shopping.example.com")
```

If entered at the command line, however, you must type this instead:

```
HTTP.REQ.HEADER(\"Host\").EQ(\"shopping.example.com\")
```

To add a firewall rule (expression) by using the Add Expression dialog box

The Add Expression dialog box (also referred to as the Expression Editor) helps users who are not familiar with the NetScaler expressions language to construct a policy that matches the traffic that they want to filter.

1. If you have not already done so, navigate to the appropriate location in the Application Firewall wizard or the NetScaler configuration utility:
 - o If you are configuring a policy in the Application Firewall wizard, in the navigation pane, click Application Firewall, then in the details pane click Application Firewall Wizard, and then navigate to the Specify Rule screen.
 - o If you are configuring a policy manually, in the navigation pane, expand Application Firewall, then Policies, and then Firewall. In the details pane, to create a new policy, click Add. To modify an existing policy, select the policy, and then click Open.
2. On the Specify Rule screen, in the Create Application Firewall Profile dialog box, or in the Configure Application Firewall Profile dialog box, click Add.
3. In the Add Expression dialog box, in the Construct Expression area, in the first list box, choose one of the following prefixes:
 - o **HTTP**. The HTTP protocol. Choose this if you want to examine some aspect of the request that pertains to the HTTP protocol. The default choice.
 - o **SYS**. The protected Web site(s). Choose this if you want to examine some aspect of the request that pertains to the recipient of the request.
 - o **CLIENT**. The computer that sent the request. Choose this if you want to examine some aspect of the sender of the request.
 - o **SERVER**. The computer to which the request was sent. Choose this if you want to examine some aspect of the recipient of the request.
4. In the second list box, choose your next term. The available terms differ depending on the choice you made in the previous step, because the dialog box automatically adjusts the list to contain only those terms that are valid for the context. For example, if you selected **HTTP** in the previous list box, the only choice is **REQ**, for requests. Because the application firewall treats requests and associated responses as a single unit and filters both, you do not need to specify responses separately. After you choose your second term, a third list box appears to the right of the second. The Help window displays a description of the second term, and the Preview Expression window displays your expression.
5. In the third list box, choose the next term. A new list box appears to the right, and the Help window changes to display a description of the new term. The Preview Expression window updates to display the expression as you have specified it to that point.
6. Continue choosing terms, and when prompted filling in arguments, until your expression is complete. If you make a mistake or want to change your expression after you have already selected a term, you can simply choose another term. The expression is modified, and any arguments or additional terms that you added after the term that you modified are cleared.
7. When you have finished constructing your expression, click OK to close the Add Expression dialog box. Your expression is inserted into the Expression text area.

To bind an application firewall policy by using the configuration utility

1. Do one of the following:
 - o Navigate to Security > Application Firewall, and in the details pane, click Application Firewall policy manager.
 - o Navigate to Security > Application Firewall > Policies > Firewall Policies, and in the details pane, click Policy Manager.
2. In the Application Firewall Policy Manager dialog, choose the bind point to which you want to bind the policy from the drop-down list. The choices are:
 - o **Override Global**. Policies that are bound to this bind point process all traffic from all interfaces on the NetScaler appliance, and are applied before any other policies.

- **LB Virtual Server.** Policies that are bound to a load balancing virtual server are applied only to traffic that is processed by that load balancing virtual server, and are applied before any Default Global policies. After selecting LB Virtual Server, you must also select the specific load balancing virtual server to which you want to bind this policy.
 - **CS Virtual Server.** Policies that are bound to a content switching virtual server are applied only to traffic that is processed by that content switching virtual server, and are applied before any Default Global policies. After selecting CS Virtual Server, you must also select the specific content switching virtual server to which you want to bind this policy.
 - **Default Global.** Policies that are bound to this bind point process all traffic from all interfaces on the NetScaler appliance.
 - **Policy Label.** Policies that are bound to a policy label process traffic that the policy label routes to them. The policy label controls the order in which policies are applied to this traffic.
 - **None.** Do not bind the policy to any bind point.
3. Click Continue. A list of existing application firewall policies appears.
 4. Select the policy you want to bind by clicking it.
 5. Make any additional adjustments to the binding.
 - To modify the policy priority, click the field to enable it, and then type a new priority. You can also select Regenerate Priorities to renumber the priorities evenly.
 - To modify the policy expression, double click that field to open the Configure Application Firewall Policy dialog box, where you can edit the policy expression.
 - To set the Goto Expression, double click field in the Goto Expression column heading to display the drop-down list, where you can choose an expression.
 - To set the Invoke option, double click field in the Invoke column heading to display the drop-down list, where you can choose an expression
 6. Repeat steps 3 through 6 to add any additional application firewall policies you want to globally bind.
 7. Click OK. A message appears in the status bar, stating that the policy has been successfully bound.

Manual Configuration By Using the Command Line Interface

You can configure many application firewall features from the NetScaler command line. There are important exceptions, however. You cannot enable signatures from the command line. There are over 1,000 default signatures in seven categories; the task is simply too complex for the command line interface. You can configure the check actions and parameters for security checks from the command line, but cannot enter manual relaxations. While you can configure the adaptive learning feature and enable learning from the command line, you cannot review learned relaxations or learned rules and approve or skip them. The command line interface is intended for advanced users who are thoroughly familiar with the NetScaler appliance and the application firewall feature.

To manually configure the application firewall by using the NetScaler command line, use a telnet or secure shell client of your choice to log on to the NetScaler command line.

To create a profile by using the command line interface

At the command prompt, type the following commands:

- o add appfw profile <name> [-defaults (**basic** | **advanced**)]
- o set appfw profile <name> -type (**HTML** | **XML** | **HTML XML**)
- o save ns config

Example

The following example adds a profile named `pr-basic`, with basic defaults, and assigns a profile type of `HTML`. This is the appropriate initial configuration for a profile to protect an HTML Web site.

```
add appfw profile pr-basic -defaults basic
set appfw profile pr-basic -type HTML
save ns config
```

To configure a profile by using the command line interface

At the command prompt, type the following commands:

- o set appfw profile <name> <arg1> [<arg2> ...] where <arg1> represents a parameter and <arg2> represents either another parameter or the value to assign to the parameter represented by <arg1>. For descriptions of the parameters to use when configuring specific security checks, see [Advanced Protections](#) and its subtopics. For descriptions of the other parameters, see "Parameters for Creating a Profile."
- o save ns config

Example

The following example shows how to configure an HTML profile created with basic defaults to begin protecting a simple HTML-based Web site. This example turns on logging and maintenance of statistics for most security checks, but enables blocking only for those checks that have extremely low false positive rates and require no special configuration. It also turns on transformation of unsafe HTML and unsafe SQL, which prevents attacks but does not block requests to your Web sites. With logging and statistics enabled, you can later review the logs to determine whether to enable blocking for a specific security check.

```
set appfw profile -startURLAction log stats
set appfw profile -denyURLAction block log stats
set appfw profile -cookieConsistencyAction log stats
set appfw profile -crossSiteScriptingAction log stats
set appfw profile -crossSiteScriptingTransformUnsafeHTML ON
set appfw profile -fieldConsistencyAction log stats
set appfw profile -SQLInjectionAction log stats
set appfw profile -SQLInjectionTransformSpecialChars ON
set appfw profile -SQLInjectionOnlyCheckFieldsWithSQLChars ON
set appfw profile -SQLInjectionParseComments checkall
set appfw profile -fieldFormatAction log stats
set appfw profile -bufferOverflowAction block log stats
set appfw profile -CSRFtagAction log stats
save ns config
```

To create and configure a policy

At the command prompt, type the following commands:

- add appfw policy <name> <rule> <profile>
- save ns config

Example

The following example adds a policy named `p1-blog`, with a rule that intercepts all traffic to or from the host `blog.example.com`, and associates that policy with the profile `pr-blog`. This is an appropriate policy to protect a blog hosted on a specific hostname.

```
add appfw policy p1-blog "HTTP.REQ.HOSTNAME.DOMAIN.EQ("blog.example.com")" pr-blog
```

To bind an Application Firewall policy

At the command prompt, type the following commands:

- bind appfw global <policyName> <priority>
- save ns config

Example

The following example binds the policy named `p1-blog` and assigns it a priority of 10.

```
bind appfw global p1-blog 10
save ns config
```

Signatures

The application firewall signatures function provides specific, configurable rules that protect your web sites against known attacks. A signature represents a pattern that is a component of a known attack on an operating system, a web server, a web site, an XML-based web service, or any other server that is connected to a web site or web service. A signature can be used to check either requests or responses. A signature can consist of a literal string or a PCRE-compliant regular expression.

To specify how the application firewall is to use signatures, you configure a signatures object, which specifies the signatures to apply to your traffic and the actions to be taken when the signatures match the traffic. A signatures object also contains the SQL injection and cross-site scripting patterns, and may also contain XPath injection patterns. These patterns are not actually signatures but are used by some of the advanced protection checks. The SQL Injection and Cross-Site Scripting patterns contain the SQL special symbols and keywords, the cross-site scripting allowed tags and attributes, and the denied patterns for the HTML and XML SQL Injection and Cross-Site Scripting checks. The XPath injection patterns contain the XPath (XML Path Language) denied patterns.

Note: If you use the wizard to configure signatures, the signatures object is created automatically.

The application firewall examines requests to your protected web sites and web services to determine whether a request matches a signature. Matching requests are handled as you specify when configuring the Signatures actions. By default, matching requests are logged so that you can examine them later. If you enabled blocking, the application firewall displays an error page or error object. If you enabled statistics, the application firewall also includes the request in the statistics that it maintains about requests that match an application firewall signature or security check.

If you want to configure signatures manually, you must create a signatures object from a template or import a signatures object file. There are two default templates that you can use: the *Default Signatures template and the *XPath Injection template. The *Default Signatures template contains over 1,000 signatures, in addition to the complete list of SQL injection and cross-site scripting allowed and denied patterns. The *XPath Injection template contains all of those, and in addition contains 57 Xpath keywords and special strings.

In addition to using its native signatures format, the application firewall can create a signatures object by using a built-in template for any supported external signatures format, or by importing an external signatures file in a supported format. The supported formats are as follows:

- **Cenzic**—Signatures files, produced by Cenzic products, that use Cenzic Hailstorm technology.
- **IBM AppScan**—Signatures files produced by IBM AppScan Enterprise and IBM AppScan Standard.
- **Qualys**—Qualys WAS signatures files produced by QualysGuard products. Only Qualys WAS 1.0 files are supported for importing as signatures. WAS 2.0 is not supported.
Note: Qualys classifies a single SQL special character in a URL as a security threat, even when no SQL keywords are present. The SQL injection check does not consider the presence of a single SQL special character a threat unless an SQL keyword is present. For that reason, a Qualys scanner continues to report such requests as containing SQL injection vulnerabilities, but the application firewall does not detect or block these requests because they pose no actual threat to your protected web sites and web services.
- **Trend Micro**—Signatures files produced by the Trend Micro Vulnerability Scanner (TMVS).
- **Whitehat**—WASC 1.0, WASC 2.0, and best practices signatures produced by Whitehat Sentinel products.

WASC signatures include information about many vulnerabilities. The application firewall generates blocking signatures from all WASC vulnerabilities. However, only certain vulnerabilities are appropriate for the web application firewall environment. For a list of appropriate Whitehat signatures, see [Whitehat WASC Signature Types for WAF Use](#).

Once you have created a signatures object, you can configure all parts of it, including the signatures rules, the XML SQL Injection and Cross-Site Scripting rules, and the XPath injection rules. You can manually create and modify your own custom signatures in the signatures editor. You can also add new SQL injection, cross-site scripting, and XPath injection patterns, modify existing patterns, and remove patterns.

Regardless of whether you use the wizard for initial configuration or configure your signatures object manually, you should regularly apply the Citrix updates to keep your signatures current. Citrix regularly updates the default application firewall signatures. You can apply those updates manually, or you can enable automatic signature updates so that the application firewall can update the signatures from the Cloud-based application firewall updates service. You can obtain the correct URL for either type of updates from your Citrix service representative or reseller.

Manually Configuring the Signatures Feature

To use signatures to protect your web sites, you must review the rules, and enable and configure the ones that you want to apply. The rules are disabled by default. Citrix recommends that you enable all rules that are applicable to the type of content that your web site uses.

To manually configure the signatures feature, use a browser to connect to the configuration utility. Then, create a signatures object from a built-in template, an existing signatures object, or by importing a file. Next, configure the new signatures object.

Note: The following procedures do not address adding user-defined signatures to a signatures object. To create your own signatures, see "[The Signatures Editor](#)."

Adding or Removing a Signatures Object

You can add a new signatures object to the application firewall by:

- Copying a built-in template.
- Copying an existing signatures object.
- Importing a signatures object from an external file.

You must use the configuration utility to copy a template or existing signatures object. You can use either the configuration utility or the command line to import a signatures object. You can also use either the configuration utility or the command line to remove a signatures object.

To create a signatures object from a template

1. Navigate to Security > Application Firewall > Signatures.
2. In the details pane, select the signatures object that you want to use as a template.

Your choices are:

- *** Default Signatures.** Contains the signatures rules, the SQL injection rules, and the cross-site scripting rules.
- *** XPath Injection.** Contains the XPath injection patterns.
- **Any existing signatures object.**

Attention: If you do not choose a signatures type to use as a template, the application firewall will prompt you to create signatures from scratch.

3. Click Add.
4. In the Add Signatures Object dialog box, type a name for your new signatures object, and then click OK. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 31 letters, numbers, and the hyphen (-), period (.) pound (#), space (), at (@), equals (=), and underscore (_) symbols.
5. Click Close.

To create a signatures object by importing a file

1. Navigate to Security > Application Firewall > Signatures.
2. In the details pane, click Add.
3. In the Add Signatures Object dialog box, select the format of the signatures you want to import.
 - To import a NetScaler format signatures file, select the Native Format tab.
 - To import an external signatures format file, select the External Format tab.
4. Choose the file that you want to use to create your new signatures object.
 - To import a native NetScaler format signatures file, in the Import section select either Import from Local File or Import from URL, then type or browse to the path or URL to the file.
 - To import a Cenzic, IBM AppScan, Qualys, or Whitehat format file, in the XSLT section select Use Built-in XSLT File, Use Local File, or Reference from URL. Next, if you chose Use Built-in XSLT File, select the appropriate file format from the drop-down list. If you chose Use Local File or Reference from URL, then type or browse to the path or URL to the file.
5. Click Add, and then click Close.

To create a signatures object by importing a file by using the command line

At the command prompt, type the following commands:

- `import appfw signatures <src> <name> [-xslt <string>] [-comment <string>] [-overwrite] [-merge] [-sha1 <string>]`
- `save ns config`

Example #1

The following example creates a new signatures object from a file named `signatures.xml` and assigns it the name `MySignatures`.

```
import appfw signatures signatures.xml MySignatures
save ns config
```

To remove a signatures object by using the configuration utility

1. Navigate to Security > Application Firewall > Signatures.
2. In the details pane, select the signatures object that you want to remove.
3. Click Remove.

To remove a signatures object by using the command line

At the command prompt, type the following commands:

- `rm appfw signatures <name>`
- `save ns config`

Configuring or Modifying a Signatures Object

You configure a signatures object after creating it, or modify an existing signatures object, to enable or disable signature categories or specific signatures, and configure how the application firewall responds when a signature matches a connection.

To configure or modify a signatures object

1. Navigate to Security > Application Firewall > Signatures.
2. In the details pane, select the signatures object that you want to configure, and then click Open.
3. In the Modify Signatures Object dialog box, set the Display Filter Criteria options at the left to display the filter items that you want to configure.

As you modify these options, the results that you requested are displayed in the Filtered Results window at the right.

- o To display only selected categories of signatures, check or clear the appropriate signature-category check boxes. The signature categories are:

Name	Type of Attack that this Signature Protects Against
cgi	CGI scripts. Includes Perl and UNIX shell scripts.
client	Browsers and other clients.
coldfusion	Web sites that use the Adobe Systems ColdFusion application server.
frontpage	Web sites that use Microsoft's FrontPage server.
iis	Web sites that use the Microsoft Internet Information Server (IIS).
misc	Miscellaneous attacks.
php	Web sites that use PHP
web-activex	Web sites that contain ActiveX controls.
web-struts	Web sites that contain Apache struts, which are java-ee based applets.

- o To display only signatures that have specific check actions enabled, select the ON check box for each of those actions, clear the ON check boxes for the other actions, and clear all of the OFF check boxes. To display only signatures that have a specific check action disabled, select their respective OFF check boxes and clear all of the ON check boxes. To display signatures regardless of whether they have a check action enabled or disabled, select or clear both the ON and the OFF check boxes for that action. The check actions are:

Criterion	Description
Enabled	The signature is enabled. The application firewall checks only for signatures that are enabled when it processes traffic.
Block	Connections that match this signature are blocked.
Log	A log entry is produced for any connection that matches this signature.
Stats	The application firewall includes any connection that matches this signature in the statistics that it generates for that check.

- o To display only signatures that contain a specific string, type the string into the text box under the filter criteria, and then click Search.
 - o To reset all display filter criteria to the default settings and display all signatures, click Show All.
4. For information about a specific signature, select the signature, and then click the blue double arrow in the More field. The Signature Rule Vulnerability Detail message box appears. It contains information about the purpose of the signature and provides links to external web-based information about the vulnerability or vulnerabilities that this signature addresses. To access an external link, click the blue double arrow to the left of the description of that link.
 5. Configure the settings for a signature by selecting the appropriate check boxes.
 6. If you want to add a local signature rule to the signatures object, or modify an existing local signature rule, see "[The Signatures Editor](#)."
 7. If you have no need for SQL injection, cross-site scripting, or Xpath injection patterns, click OK, and then click Close. Otherwise, in the lower left-hand corner of the details pane, click Manage SQL/XSS Patterns.
 8. In the Manage SQL/XSS Patterns dialog box, Filtered Results window, navigate to the pattern category and pattern that you want to configure. For information about the SQL injection patterns, see "[HTML SQL Injection Check](#)." For information about the cross-site scripting patterns, see "[HTML Cross-Site Scripting Check](#)."

9. To add a new pattern:
 - a. Select the branch to which you want to add the new pattern.
 - b. Click the Add button directly below the lower section of the Filtered Results window.
 - c. In the Create Signature Item dialog box, fill in the Element text box with the pattern that you want to add. If you are adding a transformation pattern to the transform rules branch, under Elements, fill in the From text box with the pattern that you want to change and the To text box with the pattern to which you want to change the previous pattern.
 - d. Click OK.
10. To modify an existing pattern:
 - a. In the Filtered Results window, select the branch that contains the pattern that you want to modify.
 - b. In the detail window beneath the Filtered Results window, select the pattern that you want to modify.
 - c. Click Modify.
 - d. In the Modify Signature Item dialog box, Element text box, modify the pattern. If you are modifying a transformation pattern, you can modify either or both patterns under Elements, in the From and the To text boxes.
 - e. Click OK.
11. To remove a pattern, select the pattern that you want to remove, then click the Remove button below the details pane beneath the Filtered Results window. When prompted, confirm your choice by clicking Close.
12. To add the patterns category to the XSS branch:
 - a. Select the branch to which you want to add the patterns category.
 - b. Click the Add button directly below the Filtered Results window.
Note: Currently you can add only one category, named patterns, to the XSS branch, so after you click Add, you must accept the default choice, which is `patterns`.
 - c. Click OK.
13. To remove a branch, select that branch, and then click the Remove button directly below the Filtered Results window. When prompted, confirm your choice by clicking OK.
Note: If you remove a default branch, you remove all of the patterns in that branch. Doing so can disable the security checks that use that information.
14. When you are finished modifying the SQL injection, cross-site scripting, and XPath injection patterns, click OK, and then click Close to return to the Modify Signatures Object dialog box.
15. Click OK at any point to save your changes, and when you are finished configuring the signatures object, click Close.

Updating a Signatures Object

You should update your signatures objects frequently to ensure that your application firewall is providing protection against current threats. You should regularly update both the default application firewall signatures and any signatures that you import from a supported vulnerability scanning tool.

Citrix regularly updates the default signatures for the application firewall. You can update the default signatures manually or automatically. In either case, ask your Citrix representative or Citrix reseller for the URL to access the updates. You can enable automatic updates of the Citrix native format signatures in the "Engine Settings" and "Signature Auto Update Settings" dialog boxes.

Most makers of vulnerability scanning tools regularly update the tools. Most web sites also change frequently. You should update your tool and rescan your web sites regularly, exporting the resulting signatures to a file and importing them into your application firewall configuration.

Note: When you update the application firewall signatures from the NetScaler command line, you must first update the default signatures, and then issue additional update commands to update each custom signatures file that is based on the default signatures. If you do not update the default signatures first, a version mismatch error prevents updating of the custom signatures files.

To update the application firewall signatures from the source by using the command line

At the command prompt, type the following commands:

- `update appfw signatures <name> [-mergedefault]`
- `save ns config`

Example

The following example updates the signatures object named `MySignatures` from the default signatures object, merging new signatures in the default signatures object with the existing signatures. This command does not overwrite any user-created signatures or signatures imported from another source, such as an approved vulnerability scanning tool.

```
update appfw signatures MySignatures -mergedefault
save ns config
```

Updating a Signatures Object from a Citrix Format File

Updated: 2014-10-06

Citrix regularly updates the signatures for the Application Firewall. You should regularly update the signatures on your Application Firewall to ensure that your Application Firewall is using the most current list. Ask your Citrix representative or Citrix reseller for the URL to access the updates.

To update a signatures object from a Citrix format file by using the command line

At the command prompt, type the following commands:

- `update appfw signatures <name> [-mergeDefault]`
- `save ns config`

To update a signatures object from a Citrix format file by using the configuration utility

1. Navigate to Security > Application Firewall > Signatures.
2. In the details pane, select the signatures object that you want to update.
3. In the Action drop-down list, select Merge.
4. In the Update Signatures Object dialog box, choose one of the following options.
 - **Import from URL**—Choose this option if you download signature updates from a web URL.
 - **Import from Local File**—Choose this option if you import signature updates from a file on your local hard drive, network hard drive, or other storage device.
5. In the text area, type the URL, or type or browse to the local file.
6. Click Update. The update file is imported, and the Update Signatures dialog box changes to a format nearly identical to that of the Modify Signatures Object dialog box. The Update Signatures Object dialog box displays all branches with new or modified signature rules, SQL injection or cross-site scripting patterns, and XPath injection patterns if there are any.

7. Review and configure the new and modified signatures.
8. When you are finished, click OK, and then click Close.

Updating a Signatures Object from a Supported Vulnerability Scanning Tool

Updated: 2014-01-17

Note: Before you update a signatures object from a file, you must create the file by exporting signatures from the vulnerability scanning tool.

To import and update signatures from a vulnerability scanning tool

1. Navigate to Security > Application Firewall > Signatures.
2. In the details pane, select the signatures object that you want to update, and then click Merge.
3. In the Update Signatures Object dialog box, on the External Format tab, Import section, choose one of the following options.
 - o **Import from URL**—Choose this option if you download signature updates from a Web URL.
 - o **Import from Local File**—Choose this option if you import signature updates from a file on your local or a network hard drive or other storage device.
4. In the text area, type the URL, or browse or type the path to the local file.
5. In the XSLT section, choose one of the following options.
 - o **Use Built-in XSLT File**—Choose this option if you want to use a built-in XSLT files.
 - o **Use Local File**—Choose this option to use an XSLT file on your local computer.
 - o **Reference from URL**—Choose this option to import an XSLT file from a web URL.
6. If you chose Use Built-in XSLT File, in the Built-In XSLT drop-down list choose the built-in XSLT file that you want to use.
 - o To use the Cenzic XSLT file, select Cenzic.
 - o To use the IBM AppScan Standard XSLT file, select IBM AppScan Standard.
 - o To use the IBM AppScan Enterprise XSLT file, select IBM AppScan Enterprise.
 - o To use the Qualys XSLT file, select Qualys.
 - o To use the Trend Microsystems XSLT file, select Trend Micro.
 - o To use the Whitehat XSLT file, select Whitehat.
7. Click Update. The update file is imported, and the Update Signatures dialog box changes to a format nearly identical to that of the Modify Signatures Object dialog box, which is described in "[Configuring or Modifying a Signatures Object](#)." The Update Signatures Object dialog box displays all branches with new or modified signature rules, SQL injection or cross-site scripting patterns, and XPath injection patterns if there are any.
8. Review and configure the new and modified signatures.
9. When you are finished, click OK, and then click Close.

Exporting a Signatures Object to a File

You export a signatures object to a file so that you can import it to another NetScaler ADC.

To export a signatures object to a file

1. Navigate to Security > Application Firewall > Signatures.
2. In the details pane, select the signatures object that you want to configure.
3. In the Actions drop-down list, select Export.
4. In the Export Signatures Object dialog box, Local File text box, type the path and name of the file to which you want to export the signatures object, or use the Browse dialog to designate a path and name.
5. Click OK.

The Signatures Editor

You can use the signatures editor, which is available in the configuration utility, to add a new user-defined (local) signature rule to an existing signatures object, or to modify a previously configured local signature rule. Except that it is defined by the user (you), a local signature rule has the same attributes as a default signature rule from Citrix, and it functions in the same way. You enable or disable it, and configure the signature actions for it, just as you do for a default signature.

Add a local rule if you need to protect your web sites and services from a known attack that the existing signatures do not match. For example, you might discover a new type of attack and determine its characteristics by examining the logs on your web server, or you might obtain third-party information about a new type of attack.

At the heart of a signature rule are the rule *patterns*, which collectively describe the characteristics of the attack that the rule is designed to match. Each pattern can consist of a simple string, a PCRE-format regular expression, or the built-in SQL injection or cross-site scripting patterns.

You might want to modify a signature rule by adding a new pattern or modifying an existing pattern to match an attack. For example, you might find out about changes to an attack, or you might determine a better pattern by examining the logs on your web server, or from third-party information.

To add or modify a local signature rule by using the Signatures Editor

1. Navigate to Security > Application Firewall > Signatures.
2. In the details pane, select the signatures object that you want to edit, and then click Open.
3. In the Modify Signatures Object dialog box, in the middle of the screen beneath the Filtered Results window, do one of the following:
 - o To add a new local signature rule, click Add.
 - o To modify an existing local signature rule, select that rule, and then click Open.
4. In the Add Local Signature Rule or the Modify Local Signature Rule dialog box, configure the actions for a signature by selecting the appropriate check boxes.
 - o **Enabled.** Enables the new signature rule. If you do not select this, this new signature rule is added to your configuration, but is inactive.
 - o **Block.** Blocks connections that violate this signature rule.
 - o **Log.** Logs violations of this signature rule to the NetScaler log.
 - o **Stat.** Includes violations of this signature rule in the statistics.
 - o **Remove.** Strips information that matches the signature rule from the response. (Applies only to response rules.)
 - o **X-Out.** Masks information that matches the signature rule with the letter X. (Applies only to response rules.)
 - o **Allow Duplicates.** Allows duplicates of this signature rule in this signatures object.
5. Choose a category for the new signature rule from the Category drop-down list.

You can also create a new category by clicking the icon to the right of the list and using the Add Signature Rule Category dialog box to add a new category to the list. The rule you are modifying is automatically added to the new category. For instructions, see ["To add a signature rule category."](#)

6. In the **LogString** text box, type a brief description of the signature rule to be used in the logs.
7. In the **Comment** text box, type a comment. (Optional)
8. Click More..., and modify the advanced options.
 - a. To strip HTML comments before applying this signature rule, in the Strip Comments drop-down list choose All or Exclude Script Tag.
 - b. To set CSRF Referer Header checking, in the CSRF Referer Header checking radio button array, select either the If Present or Always radio button.
 - c. To manually modify the Rule ID assigned to this local signature rule, modify the number in the Rule ID text box. The ID must be a positive integer between 1000000 and 1999999 that has not already been assigned to a local signature rule.
 - d. To assign a version number to the new signature rule, modify the number in the Version Number text box.
 - e. To assign a Source ID, modify the string in the Source ID text box.
 - f. To specify the source, choose Local or Snort from the Source drop-down list, or click the Add icon to the right of the list and add a new source.
 - g. To assign a harm score to violations of this local signature rule, type a number between 1 and 10 in the Harm Score text box.
 - h. To assign a severity rating to this local signature rule, in the Severity drop-down list choose High, Medium, or Low, or click the Add icon to the right of the list and add a new severity rating.
 - i. To assign a violation type to this local signature rule, in the Type drop-down list choose Vulnerable or Warning, or click the Add icon to the right of the list and add a new violation type.
9. In the **Patterns** list, add or edit a pattern.

- To add a pattern, click Add. In the Create New Signature Rule Pattern dialog box, add one or more patterns for your signature rule, and then click OK.
- To edit a pattern, select the pattern, and then click Open. In the Edit Signature Rule Pattern dialog box, modify the pattern, and then click OK.

For more information about adding or editing patterns, see "[Signature Rule Patterns](#)."

10. Click OK.

To add a signature rule category

Putting signature rules into a category enables you to configure the actions for a group of signatures instead of for each individual signature. You might want to do so for the following reasons:

- **Ease of selection.** For example, assume that all of signature rules in a particular group protect against attacks on a specific type of web server software or technology. If your protected web sites use that software or technology, you want to enable them all. If they do not, you do not want to enable any of them.
- **Ease of initial configuration.** It is easiest to set defaults for a group of signatures as a category, instead of one-by-one. You can then make any changes to individual signatures as needed.
- **Ease of ongoing configuration.** It is easier to configure signatures if you can display only those that meet specific criteria, such as belonging to a specific category.

1. Navigate to Security > Application Firewall > Signatures.
2. In the details pane, select that signatures object that you want to configure, and then click Open.
3. In the Modify Signatures Object dialog box, in the middle of the screen, beneath the Filtered Results window, click Add .
4. In the Add Local Signature Rule dialog box, click the icon to the right of the Category drop-down list.
5. In the Add Signature Rule Category dialog box, New Category text box, type a name for your new signature category. The name can consist of from one to 64 characters.
6. Click **OK**.

Signature Rule Patterns

You can add a new pattern to a signature rule or modify an existing pattern of a signature rule to specify a string or expression that characterizes an aspect of the attack that the signature matches. To determine which patterns an attack exhibits, you can examine the logs on your web server, use a tool to observe connection data in real time, or obtain the string or expression from a third-party report about the attack.

Caution: Any new pattern that you add to a signature rule is in an **AND** relationship with the existing patterns. Do not add a new pattern to an existing signature rule if you do not want a potential attack to have to match all of the patterns in order to match the signature.

Each pattern can consist of a simple string, a PCRE-format regular expression, or the built-in SQL injection or cross-site scripting pattern. Before you attempt to add a pattern that is based on a regular expression, you should make sure that you understand PCRE-format regular expressions. PCRE expressions are complex and powerful; if you do not understand how they work, you can unintentionally create a pattern that matches something that you did not want (a *false positive*) or that fails to match something that you did want (a *false negative*).

If you are not already familiar with PCRE-format regular expressions, you can use the following resources to learn the basics, or for help with some specific issue:

- *"Mastering Regular Expressions"*, Third Edition. Copyright (c) 2006 by Jeffrey Friedl. O'Reilly Media, ISBN: 9780596528126
- *"Regular Expressions Cookbook"*. Copyright (c) 2009 by Jan Goyvaerts and Steven Levithan. O'Reilly Media, ISBN: 9780596520687
- **PCRE Man page/Specification** (text/official): "<http://www.pcre.org/pcre.txt>"
- **PCRE Man Page/Specification** (html/gammon.edu.au): "<http://www.gammon.com.au/pcre/index.html>"
- **Wikipedia PCRE entry**: "<http://en.wikipedia.org/wiki/PCRE>"
- **PCRE Mailing List** (run by exim.org): "<http://lists.exim.org/mailman/listinfo/pcre-dev>"

If you need to encode non-ASCII characters in a PCRE-format regular expression, the NetScaler platform supports encoding of hexadecimal UTF-8 codes. For more information, see "[PCRE Character Encoding Format](#)."

To configure a signature rule pattern

1. Navigate to Security > Application Firewall > Signatures.
2. In the details pane, select that signatures object that you want to configure, and then click **Open**.
3. In the Modify Signatures Object dialog box, in the middle of the screen beneath the Filtered Results window, either click Add to create a signature rule, or select an existing signature rule and click Open.
Note: You can modify only signature rules that you added. You cannot modify the default signature rules. Depending on your action, either the Add Local Signature Rule or the Modify Local Signature Rule dialog box appears. Both dialog boxes have the same contents.
4. Under the Patterns window in the dialog box, either click Add to add a new pattern, or select an existing pattern from the list beneath the Add button and click Open. Depending on your action, either the Create New Signature Rule Pattern or the Edit Signature Rule Pattern dialog box appears. Both dialog boxes have the same contents.
5. In the Pattern Type drop-down list, choose the type of connection that the pattern is intended to match.
 - If the pattern is intended to match request elements or features, such as injected SQL code, attacks on web forms, cross-site scripts, or inappropriate URLs, choose **Request**.
 - If the pattern is intended to match response elements or features, such as credit card numbers or safe objects, choose **Response**.
6. In the Location area, define the elements to examine with this pattern.

The Location area describes what elements of the HTTP request or response to examine for this pattern. The choices that appear in the Location area depend upon the chosen pattern type. If you chose *Request* as the pattern type, items relevant to HTTP requests appear; if you chose *Response*, items relevant to HTTP responses appear.

In addition, as you choose a value from the Area drop-down list, the remaining parts of the Location area change interactively. Following are all configuration items that might appear in this section.

Area

Drop-down list of elements that describe a particular portion of the HTTP connection. The choices are as follows:

- **HTTP_ANY**. All parts of the HTTP connection.
- **HTTP_COOKIE**. All cookies in the HTTP request headers after any cookie transformations are performed.

Note: Does not search HTTP response "Set-Cookie:" headers.

- **HTTP_FORM_FIELD.** Form fields and their contents, after URL decoding, percent decoding, and removal of excess whitespace. You can use the <Location> tag to further restrict the list of form field names to be searched.
- **HTTP_HEADER.** The value portions of the HTTP header after any cross-site scripting or URL decoding transformations.
- **HTTP_METHOD.** The HTTP request method.
- **HTTP_ORIGIN_URL.** The origin URL of a web form.
- **HTTP_POST_BODY.** The HTTP post body and the web form data that it contains.
- **HTTP_RAW_COOKIE.** All HTTP request cookie, including the "Cookie:" name portion.
Note: Does not search HTTP response "Set-Cookie:" headers.
- **HTTP_RAW_HEADER.** The entire HTTP header, with individual headers separated by linefeed characters (\n) or carriage return/line-feed strings (\r\n).
- **HTTP_RAW_RESP_HEADER.** The entire response header, including the name and value parts of the response header after URL transformation has been done, and the complete response status. As with HTTP_RAW_HEADER, individual headers are separated by linefeed characters (\n) or carriage return/line-feed strings (\r\n).
- **HTTP_RAW_SET_COOKIE.** The entire Set-Cookie header after any URL transformations have been performed.
Note: URL transformation can change both the domain and path parts of the Set-Cookie header.
- **HTTP_RAW_URL.** The entire request URL before any URL transformations are performed, including any query or fragment parts.
- **HTTP_RESP_HEADER.** The value part of the complete response headers after any URL transformations have been performed.
- **HTTP_RESP_BODY.** The HTTP response body.
- **HTTP_SET_COOKIE.** All "Set-Cookie" headers in the HTTP response headers.
- **HTTP_STATUS_CODE.** The HTTP status code.
- **HTTP_STATUS_MESSAGE.** The HTTP status message.
- **HTTP_URL.** The value portion of the URL in the HTTP headers, excluding any query or fragment parts, after conversion to the UTF-* character set, URL decoding, stripping of whitespace, and conversion of relative URLs to absolute. Does not include HTML entity decoding.

URL

Examines any URLs found in the elements specified by the Area setting. Select one of the following settings.

- **Any.** Checks all URLs.
- **Literal.** Checks URLs that contain a literal string. After you select Literal, a text box is displayed. Type the literal string that you want in the text box.
- **PCRE.** Checks URLs that match a PCRE-format regular expression. After you select this choice, the regular expression window is displayed. Type the regular expression in the window. You can use the **Regex Tokens** to insert common regular expression elements at the cursor, or you can click Regex Editor to display the Regular Expression Editor dialog box, which provides more assistance in constructing the regular expression that you want.
- **Expression.** Checks URLs that match a NetScaler default expression.

Field Name

Examines any form field names found in the elements specified by the Area selection.

- **Any.** Checks all URLs.
- **Literal.** Checks URLs that contain a literal string. After you select Literal, a text box is displayed. Type the literal string that you want in the text box.
- **PCRE.** Checks URLs that match a PCRE-format regular expression. After you select this choice, the regular expression window is displayed. Type the regular expression in the window. You can use the **Regex Tokens** to insert common regular expression elements at the cursor, or you can click Regex Editor to display the Regular Expression Editor dialog box, which provides more assistance in constructing the regular expression that you want.
- **Expression.** Checks URLs that match a NetScaler default expression.

7. In the Pattern area, define the pattern. A pattern is a literal string or PCRE-format regular expression that defines the pattern that you want to match. The Pattern area contains the following elements:

Match

A drop-down list of search methods that you can use for the signature. This list differs depending on whether the pattern type is Request or Response.

Request Match Types

- **Literal.** A literal string.
- **PCRE.** A PCRE-format regular expression.

NOTE: When you choose PCRE, the regular expression tools beneath the Pattern window are enabled. These tools are not useful for most other types of patterns.

- **Injection.** Directs the application firewall to look for injected SQL in the specified location. The Pattern window disappears, because the application firewall already has the patterns for SQL injection.
- **CrossSiteScripting.** Directs the application firewall to look for cross-site scripts in the specified location. The Pattern window disappears, because the application firewall already has the patterns for cross-site scripts.
- **Expression.** An expression in the NetScaler default expressions language. This is the same expressions language that is used to create application firewall policies and other policies on the NetScaler appliance. Although the NetScaler expressions language was originally developed for policy rules, it is a highly flexible general purpose language that can also be used to define a signature pattern.

When you choose Expression, the NetScaler Expression Editor appears beneath Pattern window. For more information about the Expression Editor and instructions on how to use it, see ["To add a firewall rule \(expression\) by using the Add Expression dialog box."](#) For more information about NetScaler expressions, see ["Policies and Expressions."](#)

Response Match Types

- **Literal.** A literal string.
- **PCRE.** A PCRE-format regular expression.

NOTE: When you choose PCRE, the regular expression tools beneath the Pattern window are enabled. These tools are not useful for most other types of patterns.

- **Credit Card.** A built-in pattern to match one of the six supported types of credit card number.

Note: The Expression match type is not available for Response-side signatures.

Pattern Window (unlabeled)

In this window, type the pattern that you want to match, and fill in any additional data.

- **Literal.** Type the string you want to search for in the text area.
- **PCRE.** Type the regular expression in the text area. Use the **Regex Editor** for more assistance in constructing the regular expression that you want, or the Regex Tokens to insert common regular expression elements at the cursor. To enable UTF-8 characters, click UTF-8.
- **Expression.** Type the NetScaler advanced expression in the text area. Use Prefix to choose the first term in your expression, or Operator to insert common operators at the cursor. Click **Add** to open the Add Expression dialog box for more assistance in constructing the regular expression that you want. Click Evaluate to open the Advanced Expression Evaluator to help determine what effect your expression has.
- **Offset.** The number of characters to skip over before starting to match on this pattern. You use this field to start examining a string at some point other than the first character.
- **Depth.** How many characters from the starting point to examine for matches. You use this field to limit searches of a large string to a specific number of characters.
- **Min-Length.** The string to be searched must be at least the specified number of bytes in length. Shorter strings are not matched.
- **Max-Length.** The string to be searched must be no longer than the specified number of bytes in length. Longer strings are not matched.
- **Search method.** A check box labeled fastmatch. You can enable fastmatch only for a literal pattern, to improve performance.

8. Click OK.
9. Repeat the previous four steps to add or modify additional patterns.
10. When finished adding or modifying patterns, click OK to save your changes and return to the Signatures pane.
Caution: Until you click **OK** in the **Add Local Signature Rule** or **Modify Local Signature Rule** dialog box, your changes are not saved. Do not close either of these dialog boxes without clicking **OK** unless you want to discard your changes.

Overview of Security checks

The application firewall advanced protections (security checks) are a set of filters designed to catch complex or unknown attacks on your protected web sites and web services. The security checks use heuristics, positive security, and other techniques to detect attacks that may not be detected by signatures alone. You configure the security checks by creating and configuring an application firewall profile, which is a collection of user-defined settings that tell the application firewall which security checks to use and how to handle a request or response that fails a security check. A profile is associated with a signatures object and with a policy to create a security configuration.

The application firewall provides twenty security checks, which differ widely in the types of attacks that they target and how complex they are to configure. The security checks are organized into the following categories:

- **Common security checks.** Checks that apply to any aspect of web security that either does not involve content or is equally applicable to all types of content.
- **HTML security checks.** Checks that examine HTML requests and responses. These checks apply to HTML based web sites and to the HTML portions of Web 2.0 sites, which contain mixed HTML and XML content.
- **XML security checks.** Checks that examine XML requests and responses. These checks apply to XML-based web services and to the XML portions of Web 2.0 sites.

The security checks protect against a wide range of types of attack, including attacks on operation system and web server software vulnerabilities, SQL database vulnerabilities, errors in the design and coding of web sites and web services, and failures to secure sites that host or can access sensitive information.

All security checks have a set of configuration options, the check actions, which control how the application firewall handles a connection that matches a check. Three check actions are available for all security checks. They are:

- **Block.** Block connections that match the signature. Disabled by default.
- **Log.** Log connections that match the signature, for later analysis. Enabled by default.
- **Stats.** Maintain statistics, for each signature, that show how many connections it matched and provide certain other information about the types of connections that were blocked. Disabled by default.

A fourth check action, **Learn**, is available for more than half of the check actions. It observes traffic to a protected Web site or web service and uses connections that repeatedly violate the security check to generate recommended exceptions (relaxations) to the check, or new rules for the check. In addition to the check actions, certain security checks have parameters that control the rules that the check uses to determine which connections violate that check, or that configure the application firewall's response to connections that violate the check. These parameters are different for each check, and they are described in the documentation for each check.

To configure security checks, you can use the application firewall wizard, as described in "[The Application Firewall Wizard](#)," or you can configure the security checks manually, as described in "[Manual Configuration By Using the Configuration Utility](#)." Some tasks, such as manually entering relaxations or rules or reviewing learned data, can be done only by using the configuration utility, not the command line. Using the wizard is usually best configuration method, but in some cases manual configuration might be easier if you are thoroughly familiar with it and simply want to adjust the configuration for a single security check.

Regardless of which method you use to configure the security checks, each security check requires that certain tasks be performed. Many checks require that you specify exceptions (relaxations) to prevent blocking of legitimate traffic before you enable blocking for that security check. You can do this manually, by observing the log entries after a certain amount of traffic has been filtered and then creating the necessary exceptions. However, it is usually much easier to enable the learning feature and let it observe the traffic and recommend the necessary exceptions.

Top-Level Protections

Four of the application firewall protections are especially effective against common types of Web attacks, and are therefore more commonly used than any of the others. They are:

- **HTML Cross-Site Scripting.** Examines requests and responses for scripts that attempt to access or modify content on a different Web site than the one on which the script is located. When this check finds such a script, it either renders the script harmless before forwarding the request or response to its destination, or it blocks the connection.
- **HTML SQL Injection.** Examines requests that contain form field data for attempts to inject SQL commands into an SQL database. When this check detects injected SQL code, it either blocks the request or renders the injected SQL code harmless before forwarding the request to the Web server.

Note: If both of the following conditions apply to your configuration, you should make certain that your Application Firewall is correctly configured:

If you enable the HTML Cross-Site Scripting check or the HTML SQL Injection check (or both), and Your protected Web sites accept file uploads or contain Web forms that can contain large POST body data.

For more information about configuring the Application Firewall to handle this case, see "[Configuring the Application Firewall](#)."

- **Buffer Overflow.** Examines requests to detect attempts to cause a buffer overflow on the Web server.
- **Cookie Consistency.** Examines cookies returned with user requests to verify that they match the cookies your Web server set for that user. If a modified cookie is found, it is stripped from the request before the request is forwarded to the Web server.

The Buffer Overflow check is simple; you can usually enable blocking for it immediately. The other three top-level checks are considerably more complex and require configuration before you can safely use them to block traffic. Citrix strongly recommends that, rather than attempting to configure these checks manually, you enable the learning feature and allow it to generate the necessary exceptions.

HTML Cross-Site Scripting Check

HTML Cross-Site Scripting Check

The HTML Cross-Site Scripting check examines both the headers and the POST bodies of user requests for possible cross-site scripting attacks. If it finds a cross-site script, it either modifies (transforms) the request to render the attack harmless, or blocks the request.

To prevent misuse of the scripts on your protected web sites to breach security on your web sites, the HTML Cross-Site Scripting check blocks scripts that violate the same origin rule, which states that scripts should not access or modify content on any server but the server on which they are located. Any script that violates the same origin rule is called a cross-site script, and the practice of using scripts to access or modify content on another server is called cross-site scripting. The reason cross-site scripting is a security issue is that a web server that allows cross-site scripting can be attacked with a script that is not on that web server, but on a different web server, such as one owned and controlled by the attacker.

Unfortunately, many companies have a large installed base of JavaScript-enhanced web content that violates the same origin rule. If you enable the HTML Cross-Site Scripting check on such a site, you have to generate the appropriate exceptions so that the check does not block legitimate activity.

If you use the wizard or the configuration utility, in the Modify HTML Cross-Site Scripting Check dialog box, on the General tab you can enable or disable the Block, Log, Learn, and Statistics actions, and in addition the following parameters:

- **Transform.** If enabled, the application firewall makes the following changes to requests that match the HTML Cross-Site Scripting check:

Left angle bracket (<) to HTML character entity equivalent (<)

Right angle bracket (>) to HTML character entity equivalent (>)

This ensures that browsers do not interpret unsafe html tags, such as `<script>`, and thereby execute malicious code. If you enable both request-header checking and transformation, any special characters found in request headers are also modified as described above. If scripts on your protected web site contain cross-site scripting features, but your web site does not rely upon those scripts to operate correctly, you can safely disable blocking and enable transformation. This configuration ensures that no legitimate web traffic is blocked, while stopping any potential cross-site scripting attacks.

- **Check complete URLs.** If checking of complete URLs is enabled, the application firewall examines entire URLs for HTML cross-site scripting attacks instead of checking just the query portions of URLs.
- **Check Request headers.** If Request header checking is enabled, the application firewall examines the headers of requests for HTML cross-site scripting attacks, instead of just URLs. If you use the configuration utility, you can enable this parameter in the **Settings** tab of the application firewall profile.

If you use the command-line interface, you can enter the following commands to configure the HTML Cross-Site Scripting Check:

- set appfw profile <name> -crossSiteScriptingAction [block] [learn] [log] [stats] [none]
- set appfw profile <name> -crossSiteScriptingTransformUnsafeHTML ([ON] | [OFF])
- set appfw profile <name> -crossSiteScriptingCheckCompleteURLs ([ON] | [OFF])
- set appfw profile <name> -checkRequestHeaders ([ON] | [OFF])

To specify relaxations for the HTML Cross-Site Scripting check, you must use the configuration utility. On the Checks tab of the Modify HTML Cross-Site Scripting Check dialog box, click Add to open the Add HTML Cross-Site Scripting Check Relaxation dialog box, or select an existing relaxation and click Open to open the Modify HTML Cross-Site Scripting Check Relaxation dialog box. Either dialog box provides the same options for configuring a relaxation, as described in "Manual Configuration By Using the Configuration Utility."

Following are examples of HTML Cross-Site Scripting check relaxations:

Web Form Field Expressions

- **Logon Fields.** The following expression exempts all fields beginning with the string `logon_` followed by a string of upper- and lower-case letters or numbers that is at least two characters long and no more than fifteen characters long:

```
^logon_[0-9A-Za-z]{2,15}$
```

- o **Name Fields.** The following expression exempts form fields with names beginning with Name_ followed by a string beginning with a letter or number and consisting of from one to twenty letters, numbers, or the apostrophe or hyphen symbol:

```
^Name_[0-9A-Za-z][0-9A-Za-z'-]{0,20}$
```

- o **Name Fields (Special Characters).** If your web site has Turkish-speaking customers whose first names may contain special characters, you might have a form field that begins with the string Turkish-Name_ on their logon page. In addition, the customers may use the same special characters in their names. The special characters in both of these strings must be represented as encoded UTF-8 strings. The following expression exempts form fields beginning with Turkish-Name_ and containing Turkish special characters:

```
^T\xC3\xBCrk\xC3\xA7e-Name_([0-9A-Za-z]|\\x[0-9A-Fa-f][0-9A-Fa-f])+ $
```

- o **Session-ID Fields.** The following expression exempts all fields beginning with the string sessionid- followed by a ten-digit number:

```
^sessionid-[0-9]{10,10}$
```

URL Expressions

- o **URLs using JavaScript.** You can use a single expression to exempt all URLs that end with a filename that follows a specified pattern. The following expression exempts all URLs that end with the string query_ followed by a string of upper- and lower-case letters or numbers that is at least two characters long and no more than forty characters long, and ending with the string .js:

```
query_[0-9A-Za-z]{2,40}[. ]js$
```

- o **URLs containing a Specified String.** You can use an expression to exempt all URLs that contain a specific string. The following expression exempts all URLs that contain the string prodinfo:

```
^https?:/(( [0-9A-Za-z]|\\x[0-9A-Fa-f][0-9A-Fa-f])(( [0-9A-Za-z_-]|\\x[0-9A-Fa-f][0-9A-Fa-f])+[. ])+[a-z]{2,6}/[^<>?]*\?prodinfo[^\<>?]*$
```

In the above expression, each character class has been grouped with the string `\\x[0-9A-Fa-f][0-9A-Fa-f]`, which matches all properly constructed character encoding strings but does not allow stray backslash characters that are not associated with a UTF-8 character encoding string. The double backslash (`\\`) is an escaped backslash, which tells the application firewall to interpret it as a literal backslash. If you included only one backslash, the application firewall would interpret the following left square bracket (`[`) as a literal character instead of as the opening of a character class, which would break the expression.

Caution: Regular expressions are powerful. Especially if you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions you write. Make sure that they define exactly the URL you want to add as an exception, and nothing else. Careless use of wildcards, and especially of the dot-asterisk (`. *`) metacharacter/wildcard combination, can have results you do not want or expect, such as blocking access to web content that you did not intend to block or allowing an attack that the HTML Cross-Site Scripting check would otherwise have blocked.

HTML SQL Injection Check

The HTML SQL Injection check provides special defenses against injection of unauthorized SQL code that might break security. It examines both the headers and the POST bodies of requests for injected SQL code. If the application firewall detects unauthorized SQL code in a user request, it either transforms the request, to render the SQL code inactive, or blocks the request.

Many web applications have web forms that use SQL to communicate with relational database servers. Often, the scripts that pass web form information to the database do not validate the information provided by the user before sending it to the database. Malicious code or a hacker can use the insecure web form to send SQL commands to the web server.

The application firewall HTML SQL Injection check provides special defenses against injection of unauthorized SQL code that might break security. If the application firewall detects unauthorized SQL code in a user request, it either transforms the request, to render the SQL code inactive, or blocks the request. The application firewall examines the request payload for injected SQL code in three locations: 1) POST body, 2) headers, and 3) cookies.

A default set of keywords and special characters provides known keywords and special characters that are commonly used to launch SQL attacks. You can add new patterns, and you can edit the default set to customize the SQL check inspection. The application firewall offers various action options for implementing SQL Injection protection. In addition to the **Block**, **Log**, **Stats** and **Learn** actions, the application firewall profile also offers the option to **transform SQL special characters** to render an attack harmless.

In addition to actions, there are several parameters that can be configured for SQL injection processing. You can check for **SQL wildcard characters**. You can change the SQL Injection type and select one of the 4 options (**SQLKeyword**, **SQLSplChar**, **SQLSplCharANDKeyword**, **SQLSplCharORKeyword**) to indicate how to evaluate the SQL keywords and SQL special characters when processing the payload. The **SQL Comments Handling** parameter gives you an option to specify the type of comments that need to be inspected or exempted during SQL Injection detection.

You can deploy relaxations to avoid false positives. The application firewall learning engine can provide recommendations for configuring relaxation rules.

Following options are available for configuring an optimized SQL Injection protection for your application:

- **Block**—If you enable block, the block action is triggered only if the input matches the SQL injection type specification. For example, if **SQLSplCharANDKeyword** is configured as the SQL injection type, a request is not blocked if it contains no key words, even if SQL special characters are detected in the input. Such a request is blocked if the SQL injection type is set to either **SQLSplChar**, or **SQLSplCharORKeyword**.
- **Log**—If you enable the log feature, the SQL Injection check generates log messages indicating the actions that it takes. If block is disabled, a separate log message is generated for each input field in which the SQL violation was detected. However, only one message is generated when the request is blocked. Similarly, one log message per request is generated for the transform operation, even when SQL special characters are transformed in multiple fields. You can monitor the logs to determine whether responses to legitimate requests are getting blocked. A large increase in the number of log messages can indicate attempts to launch an attack.
- **Stats**—If enabled, the stats feature gathers statistics about violations and logs. An unexpected surge in the stats counter might indicate that your application is under attack. If legitimate requests are getting blocked, you might have to revisit the configuration to see if you need to configure new relaxation rules or modify the existing ones.
- **Learn**—If you are not sure which SQL relaxation rules might be ideally suited for your application, you can use the learn feature to generate recommendations based on the learned data. The application firewall learning engine monitors the traffic and provides SQL learning recommendations based on the observed values. To get optimal benefit without compromising performance, you might want to enable the learn option for a short time to get a representative sample of the rules, and then deploy the rules and disable learning.
- **Transform SQL special characters**—The application firewall considers three characters, Single straight quote ('), Backslash (\), and Semicolon (;) as special characters for SQL security check processing. The SQL Transformation feature modifies the SQL Injection code in an HTML request to ensure that the request is rendered harmless. The modified HTML request is then sent to the server. All default transformation rules are specified in the /netscaler/default_custom_settings.xml file.

The transform operation renders the SQL code inactive by making the following changes to the request:

Single straight quote (') to double straight quote (").

Backslash (\) to double backslash (\\).

Semicolon (;) is dropped completely.

These three characters (special strings) are necessary to issue commands to an SQL server. Unless an SQL command is prefaced with a special string, most SQL servers ignore that command. Therefore, the changes that the application firewall performs when transformation is enabled prevent an attacker from injecting active SQL. After these changes are made, the request can safely be forwarded to your protected web site. When web forms on your protected web site can legitimately contain SQL special strings, but the web forms do not rely on the special strings to operate correctly, you can disable blocking and enable transformation to prevent blocking of legitimate web form data without reducing the protection that the application firewall provides to your protected web sites.

The transform operation works independently of the SQL Injection Type setting. If transform is enabled and the SQL Injection type is specified as SQL keyword, SQL special characters are transformed even if the request does not contain any keywords.

Note: You normally enable either transformation or blocking, but not both. If the block action is enabled, it takes precedence over the transform action. If you have blocking enabled, enabling transformation is redundant.

In addition to configuring the actions, in the HTML SQL Injection Settings pane, you can also configure the following parameters:

- o **Check for SQL Wildcard Characters**—Wild card characters can be used to broaden the selections of a structured query language (SQL-SELECT) statement. These wild card operators can be used in conjunction with LIKE and NOT LIKE operators to compare a value to similar values. The percent (%), and underscore (_) characters are frequently used as wild cards. The percent sign is analogous to the asterisk (*) wildcard character used with MS-DOS and to match zero, one, or multiple characters in a field. The underscore is similar to the MS-DOS question mark (?) wildcard character. It matches a single number or character in an expression.

For example, you can use the following query to do a string search to find all customers whose names contain the D character.

```
SELECT * from customer WHERE name like "%D%"
```

The following example combines the operators to find any salary values that have 0 in the second and third place.

```
SELECT * from customer WHERE salary like '_00%'
```

Different DBMS vendors have extended the wildcard characters by adding extra operators. The NetScaler application firewall can protect against attacks that are launched by injecting these wildcard characters. The 5 default Wildcard characters are percent (%), underscore (_), caret (^), opening square bracket ([), and closing square bracket (]). This protection applies to both HTML and XML profiles.

The default wildcard chars are a list of literals specified in the ***Default Signatures:**

```
<wildchar type=LITERAL>%</wildchar>
```

```
<wildchar type=LITERAL>_</wildchar>
```

```
<wildchar type=LITERAL>^</wildchar>
```

```
<wildchar type=LITERAL>[</wildchar>
```

```
<wildchar type=LITERAL>]</wildchar>
```

Wildcard characters in an attack can be PCRE, like [^A-F]. The application firewall also supports PCRE wildcards, but the literal wildcard chars above are sufficient to block most attacks.

Note: The SQL wildcard character check is different from the SQL special character check. This option must be used with caution to avoid false positives.

- o **Check Request Containing SQL Injection Type**—The application firewall provides 4 options to implement the desired level of strictness for SQL Injection inspection, based on the individual need of the application. The request is checked against the injection type specification for detecting SQL violations. The 4 SQL injection type options are:

SQL Special Character and Keyword—Both an SQL keyword and an SQL special character must be present in the input to trigger SQL violation. This least restrictive setting is also the default setting.

SQL Special Character—At least one of the special characters must be present in the input to trigger SQL violation.

SQL key word—At least one of the specified SQL keywords must be present in the input to trigger an SQL violation. Do not select this option without due consideration. To avoid false positives, make sure that none of the keywords are expected in the inputs.

SQL Special Character or Keyword—Either the key word or the special character string must be present in the input to trigger the security check violation.

- o **SQL comments handling.** By default, the application firewall checks all SQL comments for injected SQL commands. Many SQL servers ignore anything in a comment, however, even if it is preceded by an SQL special character. For faster processing, if your SQL server ignores comments, you can configure the application firewall to skip comments when examining requests for injected SQL. The SQL comments handling options are:

ANSI. Skip ANSI-format SQL comments, which are normally used by UNIX-based SQL databases.

Nested. Skip nested SQL comments, which are normally used by Microsoft SQL Server.

ANSI/Nested. Skip comments that adhere to both the ANSI and nested SQL comment standards. Comments that match only the ANSI standard, or only the nested standard, are still checked for injected SQL.

Check all Comments. Check the entire request for injected SQL, without skipping anything. The default setting.

Caution: In most cases, you should not choose the Nested or the ANSI/Nested option unless your back-end database runs on Microsoft SQL Server. Most other types of SQL server software do not recognize nested comments. If nested comments appear in a request directed to another type of SQL server, they may indicate an attempt to breach security on that server.

- o **Check Request headers.** Examine the headers of requests for HTML SQL Injection attacks, instead of just URLs. If you use the configuration utility, you can enable this parameter in the Advanced Settings - > Profile Settings pane of the application firewall profile.

Caution: If you enable both request header checking and transformation, any SQL special characters found in headers are also transformed. The Accept, Accept-Charset, Accept-Encoding, Accept-Language, Expect, and User-Agent headers normally contain semicolons (;), so enabling both Request header checking and transformation simultaneously may cause errors.

If you use the command-line interface, you can enter the following commands to configure the HTML SQL Injection Check:

- o **set appfw profile <name> -SQLInjectionAction ([block] [learn] [log] [stats]) | [none])**
- o **set appfw profile <name> -SQLInjectionTransformSpecialChars (ON | OFF)**
- o **set appfw profile <name> -SQLInjectionCheckSQLWildChars (ON | OFF)**
- o **set appfw profile <name> -SQLInjectionType ([SQLKeyword] | [SQLSpiChar] | [SQLSpiCharANDKeyword] | [SQLSpiCharORKeyword])**
- o **set appfw profile <name> -SQLInjectionParseComments ([checkall] | [ansi|nested] | [ansinested])**
- o **set appfw profile <name> -CheckRequestHeaders (ON | OFF)**

To configure or modify the SQL Injection check by using the configuration utility

- o Navigate to **Application Firewall > Profiles**, highlight the target profile, and click **Edit**.
- o In the Advanced Settings pane, click **Security Checks**.

The security check table displays the currently configured action settings for all the security checks. You have 2 options for configuration:

If you just want to enable or disable **Block**, **Log**, **Stats**, and **Learn** actions for **HTML SQL Injection**, you can select or clear check boxes in the table, click **OK**, and then click **Save and Close** to close the Security Check pane.

If you want to configure additional options for this security check, double click **HTML SQL Injection**, or select the row and click Action Settings, to display the following options:

Transform SQL Special character—Transform any SQL Special characters in the request.

Check for SQL Wildcard Charactersâ€”Consider SQL Wildcard characters in the payload to be attack patterns.

Check Request Containingâ€”Type of SQL injection (**SQLKeyword**, **SQLSpIChar**, **SQLSpICharANDKeyword**, or **SQLSpICharORKeyword**) to check.

SQL Comments Handlingâ€”Type of comments (**Check All Comments**, **ANSI**, **Nested**, or **ANSI/Nested**) to check.

After changing any of the above settings, click **OK** to save the changes and return to the Security Checks table. You can proceed to configure other security checks if needed. Click **OK** to save all the changes you have made in the Security Checks section, and then click **Save and Close** to close the Security Check pane.

- o To enable or disable the **Check request Header** setting, in the Advanced Settings pane, click **Profile Settings**. In Common Settings, Select or clear the Check Request Headers check box. Click **OK**. You can either use the X icon at the top right hand side of the Profile Settings pane to close this section or, if you have finished configuring this profile, you can click **Done** to return to the **Application Firewall > Profile**.

To configure an SQL Injection relaxation rule by using the configuration utility

- o Navigate to **Application Firewall > Profiles**, highlight the target profile, and click **Edit**.
- o In the Advanced Settings pane, click **Relaxation Rules**.
- o In the Relaxation Rules table, double-click the **HTML SQL Injection** entry, or select it and click **Edit**.
- o In the HTML SQL Injection Relaxation Rules dialogue box, perform **Add**, **Edit**, **Delete**, **Enable**, or **Disable** operations for relaxation rules.

Following are examples of HTML SQL Injection check relaxations:

Web Form Field Name Expressions

- o **Logon Fields.** The following expression exempts all fields beginning with the string `logon_` followed by a string of letters or numbers that is at least two characters long and no more than fifteen characters long:

```
^logon_[0-9A-Za-z]{2,15}$
```

- o **Name Fields.** The following expression exempts form fields with names beginning with `Name_` followed by a string beginning with a letter or number and consisting of from one to twenty letters, numbers, or the apostrophe or hyphen symbol:

```
^Name_[0-9A-Za-z][0-9A-Za-z' -]{0,20}$
```

- o **Name Fields (Special Characters).** If your web site has Turkish-speaking customers whose first names may contain special characters, you might have a form field that begins with the string `Turkish-Name_` on their logon page. In addition, the customers may use the same special characters in their names. The special characters in both of these strings must be represented as encoded UTF-8 strings. The following expression exempts form fields beginning with `Turkish-Name_` and containing Turkish special characters:

```
^T\xC3\xBCrk\xC3\xA7e-Name_([0-9A-Za-z]|\x[0-9A-Fa-f][0-9A-Fa-f])+ $
```

- o **Session-ID Fields.** The following expression exempts all fields beginning with the string `sessionid-` followed by a ten-digit number:

```
^sessionid-[0-9]{10,10}$
```

Action URL Expressions

- o **URLs using JavaScript.** You can use a single expression to exempt all URLs that end with a filename that follows a specified pattern. The following expression exempts all URLs that end with the string `query_` followed by a string of upper- and lower-case letters or numbers that is at least two characters long and no more than forty characters long, and that end with the string `.js`:

```
query_[0-9A-Za-z]{2,40}[. ]js$
```

- o **URLs containing a Specified String.** You can use an expression to exempt all URLs that contain a specific string. The following expression exempts all URLs that contain the string `prodinfo`:

```
^https?:/(( [0-9A-Za-z]|\x[0-9A-Fa-f][0-9A-Fa-f])(( [0-9A-Za-z_-]|\x[0-9A-Fa-f][0-9A-Fa-f])+[.])+[a-z]{2,6}/[^<>?]*\?prodinfo[^\<>?]*$
```

In the expression above, each character class has been grouped with the string `\x[0-9A-Fa-f][0-9A-Fa-f]`, which matches all properly constructed character encoding strings but does not allow stray backslash characters that are not associated with a UTF-8 character encoding string. The double backslash (`\\`) is an escaped backslash, which tells the application firewall to interpret it as a literal backslash. If you included only one backslash, the application firewall would interpret the following left square bracket (`[`) as a literal character instead of as the opening of a character class, which would break the expression.

Caution: Regular expressions are powerful. Especially if you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions you write. Make sure that they define exactly the URL that you want to add as an exception, and nothing else. Careless use of wildcards, and especially of the dot-asterisk (`. *`) metacharacter/wildcard combination, can have results that you do not want, such as blocking access to web content that you did not intend to block or allowing an attack that the HTML SQL Injection check would otherwise have blocked.

Buffer Overflow Check

The Buffer Overflow check detects attempts to cause a buffer overflow on the web server. If the application firewall detects that the URL, cookies or header are longer than the specified maximum length in a request, it blocks that request because it might be an attempt to cause a buffer overflow.

The Buffer Overflow check prevents attacks against insecure operating-system or web-server software that can crash or behave unpredictably when it receives a data string that is larger than it can handle. Proper programming techniques prevent buffer overflows by checking incoming data and either rejecting or truncating overlong strings. Many programs, however, do not check all incoming data and are therefore vulnerable to buffer overflows. This issue especially affects older versions of web-server software and operating systems, many of which are still in use.

If you use the wizard or the configuration utility, in the Modify Buffer Overflow Check dialog box, on the General tab you can enable or disable the Block, Log, and Statistics actions. On the Checks tab, you can set the following parameters:

- o **Maximum URL Length.** The maximum length the application firewall allows in a requested URL. Requests with longer URLs are blocked. Possible Values: 0-65536. Default: 1024
- o **Maximum Cookie Length.** The maximum length the application firewall allows for all cookies in a request. Excess cookies are stripped from requests before those requests are forwarded to your protected web server. Possible Values: 0-65536. Default: 4096
- o **Maximum Header Length.** The maximum length the application firewall allows for HTTP headers. Requests with longer headers are blocked. Possible Values: 0-65536. Default: 4096

If you use the command-line interface, you can add the following Buffer Overflow Check arguments to the set appfwl profile <profileName> command:

- o -bufferOverflowAction [**block**] [**log**] [**stats**]
- o -bufferOverflowMaxURLLength <positiveInteger>
- o -bufferOverflowMaxCookieLength <positiveInteger>
- o -bufferOverflowMaxHeaderLength <positiveInteger>

Cookie Consistency Check

The Cookie Consistency check examines cookies returned by users, to verify that they match the cookies that your web site set for that user. If a modified cookie is found, it is stripped from the request before the request is forwarded to the web server. You can also configure the Cookie Consistency check to transform all of the server cookies that it processes, by encrypting the cookies, proxying the cookies, or adding flags to the cookies. This check applies to requests and responses.

An attacker would normally modify a cookie to gain access to sensitive private information by posing as a previously authenticated user, or to cause a buffer overflow. The Buffer Overflow check protects against attempts to cause a buffer overflow by using a very long cookie. The Cookie Consistency check focuses on the first scenario.

If you use the wizard or the configuration utility, in the Modify Cookie Consistency Check dialog box, on the General tab you can enable or disable the following actions:

- Block
- Log
- Learn
- Statistics
- Transform. If enabled, the Transform action modifies all cookies as specified in the following settings:
 - Encrypt Server Cookies.** Encrypt cookies set by your web server, except for any listed in the Cookie Consistency check relaxation list, before forwarding the response to the client. Encrypted cookies are decrypted when the client sends a subsequent request, and the decrypted cookies are reinserted into the request before it is forwarded to the protected web server. Specify one of the following types of encryption:
 - **None.** Do not encrypt or decrypt cookies. The default.
 - **Decrypt only.** Decrypt encrypted cookies only. Do not encrypt cookies.
 - **Encrypt session only.** Encrypt session cookies only. Do not encrypt persistent cookies. Decrypt any encrypted cookies.
 - **Encrypt all.** Encrypt both session and persistent cookies. Decrypt any encrypted cookies.

Note: When encrypting cookies, the application firewall adds the **HttpOnly** flag to the cookie. This flag prevents scripts from accessing and parsing the cookie. The flag therefore prevents a script-based virus or trojan from accessing a decrypted cookie and using that information to breach security. This is done regardless of the Flags to Add in Cookies parameter settings, which are handled independently of the Encrypt Server Cookies parameter settings.
 - **Proxy Server Cookies.** Proxy all non-persistent (session) cookies set by your web server, except for any listed in the Cookie Consistency check relaxation list. Cookies are proxied by using the existing application firewall session cookie. The application firewall strips session cookies set by the protected web server and saves them locally before forwarding the response to the client. When the client sends a subsequent request, the application firewall reinserts the session cookies into the request before forwarding it to the protected web server. Specify one of the following settings:
 - **None.** Do not proxy cookies. The default.
 - **Session only.** Proxy session cookies only. Do not proxy persistent cookies.

Note: If you disable cookie proxying after having enabled it (set this value to None after it was set to Session only), cookie proxying is maintained for sessions that were established before you disabled it. You can therefore safely disable this feature while the application firewall is processing user sessions.
 - **Flags to Add in Cookies.** Add flags to cookies during transformation. Specify one of the following settings:
 - **None.** Do not add flags to cookies. The default.
 - **HTTP only.** Add the HttpOnly flag to all cookies. Browsers that support the HttpOnly flag do not allow scripts to access cookies that have this flag set.
 - **Secure.** Add the Secure flag to cookies that are to be sent only over an SSL connection. Browsers that support the Secure flag do not send the flagged cookies over an insecure connection.
 - **All.** Add the HttpOnly flag to all cookies, and the Secure flag to cookies that are to be sent only over an SSL connection.

If you use the command-line interface, you can enter the following commands to configure the Cookie Consistency Check:

- set appfw profile <name> -cookieConsistencyAction [block] [learn] [log] [stats] [none]
- set appfw profile <name> -cookieTransforms ([ON] | [OFF])
- set appfw profile <name> -cookieEncryption ([none] | [decryptOnly] | [encryptSession] | [encryptAll])
- set appfw profile <name> -cookieProxying ([none] | [sessionOnly])
- set appfw profile <name> -addCookieFlags ([none] | [httpOnly] | [secure] | [all])

To specify relaxations for the Cookie Consistency check, you must use the configuration utility. On the Checks tab of the Modify Cookie Consistency Check dialog box, click Add to open the Add Cookie Consistency Check Relaxation dialog box, or select an existing relaxation and click Open to open the Modify Cookie Consistency Check Relaxation dialog box. Either dialog box provides the same options for configuring a relaxation.

Following are examples of Cookie Consistency check relaxations:

- o **Logon Fields.** The following expression exempts all form fields beginning with the string `logon_` followed by a string of letters or numbers that is at least two characters long and no more than fifteen characters long:

```
^logon_[0-9A-Za-z]{2,15}$
```

- o **Logon Fields (special characters).** The following expression exempts all form fields beginning with the string `tÃ¼rkÃ§e-logon_` followed by a string of letters or numbers that is at least two characters long and no more than fifteen characters long:

```
^t\xC3\xBCrk\xC3\xA7e-logon_[0-9A-Za-z]{2,15}$
```

- o **Arbitrary strings.** Allow cookies that contain the string `sc-item_`, followed by the ID of an item that the user has added to his shopping cart (`[0-9A-Za-z]+`), a second underscore (`_`), and finally the number of these items he wants (`[1-9][0-9]?`), to be user-modifiable:

```
^sc-item_[0-9A-Za-z]+_[1-9][0-9]?$
```

Caution: Regular expressions are powerful. Especially if you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions you write. Make sure that they define exactly the URL you want to add as an exception, and nothing else. Careless use of wildcards, and especially of the dot-asterisk (`. *`) metacharacter/wildcard combination, can have results you do not want or expect, such as blocking access to web content that you did not intend to block or allowing an attack that the Cookie Consistency check would otherwise have blocked.

Application Firewall Support for Google Web Toolkit

Note: This feature is available in NetScaler release 10.5.e.

Web servers following Google Web Toolkit (GWT) Remote Procedure Call (RPC) mechanisms can be secured by the NetScaler application firewall without a need for any specific configuration to enable the GWT support.

What is GWT

The GWT is used for building and optimizing complex high-performance web applications by people who do not have expertise in XMLHttpRequest, and JavaScript. This open source, free development toolkit is used extensively for developing small and large scale applications and is quite frequently used for displaying browser based data such as search results for flights, hotels, and so on. The GWT provides a core set of Java APIs and widgets for writing optimized JavaScript scripts that can run on most browsers and mobile devices. The GWT RPC framework makes it easy for the client and server components of the web application to exchange Java objects over HTTP. GWT RPC services are not the same as web services based on SOAP or REST. They are simply a lightweight method for transferring data between the server and the GWT application on the client. GWT handles serialization of the Java objects exchanging the arguments in the method calls and the return value.

For popular websites that use GWT, see

<https://www.quora.com/What-web-applications-use-Google-Web-Toolkit-%28GWT%29>

How a GWT Request Works

The GWT RPC request is pipe delimited and has variable number of arguments. It is carried as a payload of HTTP POST and has the following values:

1. Content-type = text/x-gwt-rpc. Charset can be any value.
2. Method = POST.

The following example shows a valid payload for a GWT request:

```
5|0|8|http://localhost:8080/test/|16878339F02B83818D264AE430C20468| com.test.client.
TestService|testMethod| java.lang.String| java.lang.Integer| myInput1| java.lang.
Integer/3438268394|1|2|3|4|2|5|6|7|8|1|
```

The request can be divided into three parts:

a) Header: 5|0|8|

The first 3 digits "5|0|8|" in the above request, represent "version, subversion, and size of table", respectively. These must be positive integers.

b) String Table:

```
http://localhost:8080/test/|16878339F02B83818D264AE430C20468| com.test.client.
TestService|testMethod| java.lang.String| java.lang.Integer|myInput1| java.lang.
Integer/3438268394|
```

The members of the above pipe delimited string table contain the user-provided inputs. These inputs are parsed for the application firewall checks and are identified as follows:

- 1st : http://localhost:8080/test/

This is the Request URL. It should not contain the query part, because the GET method is not allowed.

- 2nd : 16878339F02B83818D264AE430C20468

Unique HEX identifier. A request is considered malformed if this string has non-hex characters.

- 3rd : com.test.client.TestService

Service Class name

- 4th : testMethod

Service method name

- 5th onwards: java.lang.String|java.lang.Integer|myInput1|java.lang.Integer/3438268394

Data-types and data. Non-primitive data-types are specified as

<container>.<sub-cntr>.name/<integer_identifier>

c) Payload: 1|2|3|4|2|5|6|7|8|1|

The payload consists of references to the elements in the string table. These integer values cannot be larger than the number of elements in the string table.

Application firewall protection for GWT applications

The application firewall understands and interprets GWT RPC requests, inspects the payload for security check violations, and takes specified actions.

The application firewall headers and cookies checks for GWT requests are similar to those for other request formats. After appropriate URL decoding and charset conversion, all the parameters in the string table are inspected. The GWT request body does not contain field names, just the field values. The input values can be validated against the specified format by using the application firewall Field Format check, which can also be used to control the length of the input. The **Cross-site Scripting** and **SQL Injection** attacks in the inputs can be easily detected and thwarted by the application firewall.

Learning and relaxation rules: Learning and deployment of relaxation rules are supported for GWT requests. Application firewall rules are in the form of <actionURL> <fieldName> mapping. The GWT request format does not have the field names and thus requires special handling. The application firewall inserts dummy field names in the learned rules that can be deployed as relaxation rules. The -isRegex flag works as it does for non-GWT rules.

Action URL:

Multiple services responding to an RPC can be configured on the same web server. The HTTP request has the URL of the web server, not of the actual service handling the RPC. Therefore, relaxation is not applied on the basis of the HTTP request URL, because that would relax all the services on that URL for the target field. For GWT requests, the application firewall uses the URL of the actual service found in the GWT payload, in the fourth field in the string table.

Field name:

Since the GWT request body contains only field values, the application firewall inserts dummy field names such as 1, 2, and so on when recommending learned rules.

Example of a GWT learned rule

```
POST /abcd/def/gh HTTP/1.1
Content-type: text/x-gwt-rpc
Host: 10.217.222.75
Content-length: 157
```

```
5|0|8|http://localhost:8080/acdtest/|16878339F02Baf83818D264AE430C20468|
com.test.client.TestService|testMethod|java.lang.String%3b|java.lang.Integer|onblur|
```

```
The learn data will be as follows:
> sh learningdata pr1 crossSiteScripting
Profile: pr1          SecurityCheck: crossSiteScripting
1)      Url:          http://localhost:8080/acdtest/  >> From GWT Payload.
        Field:       10
        Hits:        1
Done
```

Example of a GWT relaxation rule

```
bind appfw profile pr1 -crossSiteScripting 1 abcd -isregex NOTREGEX
```

Log Messages: The application firewall generates log messages for the security check violations that are detected in the GWT requests. A log message generated by a malformed GWT request contains the string "GWT" for easy identification.

Example of a Log message for malformed GWT request

```
Dec 5 21:48:02 <local0.notice> 10.217.31.247 12/05/2014:21:48:02 GMT ns 0-PPE-0 : APPFW Message 696 0 : "GWT
RPC request with malformed payload. <blocked>"
```

Difference in processing of GWT vs non-GWT requests

The same payload can trigger different application firewall security check violations for different Content-types. Consider the following example:

```
5|0|8|http://localhost:8080/acctest/|16878339F02Baf83818D264AE430C20468|com.test.client.  
TestService|testMethod|java.lang.String%3b|java.lang.Integer|select|
```

Content-type: application/x-www-form-urlencoded

A request sent with this content type results in a SQL violation if the SQL Injection Type is configured to use any of the four available options: **SQLSpICharANDKeyword**, **SQLSpICharORKeyword**, **SQLKeyword**, or **SQLSpIChar**. The application firewall considers '&' to be the field separator and '=' to be the name-value separator when processing the above payload. Since neither of these characters appears anywhere in the post body, the entire content is treated as a single field name. The field name in this request contains both an SQL special character (;) and an SQL Keyword (select). Therefore violations are caught for all four SQL Injection type options.

Content-type: text/x-gwt-rpc

A request sent with this content type triggers an SQL violation only if the SQL injection type is set to one of the following three options: **SQLSpICharORKeyword**, **SQLKeyword**, or **SQLSpIChar**. No violation is triggered if the SQL injection type is set to **SQLSpICharANDKeyword**, which is the default option. The application firewall considers the vertical bar '|' to be the field separator for the above payload in the GWT request. Therefore, the post body is divided into various form-field values, and form-field names are added (in accordance with the convention described earlier). Because of this splitting, the SQL special character and SQL keyword become parts of separate form fields.

Form field 8: java.lang.String%3b -> %3b is the (;) char

Form Field 10: select

As a result, when SQL Injection Type is set to **SQLSpIChar**, field 8 indicates the SQL violation. For **SQLKeyword**, field 10 indicates the violation. Either of these two fields can indicate a violation if the SQL Inject type is configured with the **SQLSpICharORKeyword** option, which looks for the presence of either a keyword or a special character. No violation is caught for the default **SQLSpICharANDKeyword** option, because there is no single field that has a value that contains both **SQLSpIChar** and **SQLKeyword** together.

Tips:

- No special application firewall configuration is needed to enable GWT support.
- The Content-type must be text/x-gwt-rpc.
- Learning and deploying of the relaxation rules for all the pertinent application firewall security checks applied to GWT payload works the same as it does for the other supported content-types.
- Only POST requests are considered valid for GWT. All other request methods are blocked if the content-type is text/x-gwt-rpc.
- GWT requests are subject to the configured POST body limit of the profile.
- The sessionless setting for the security checks is not applicable and will be ignored.
- CEF log format is supported for the GWT log messages.

Data Leak Prevention Checks

The data-leak-prevention checks filter responses to prevent leaks of sensitive information, such as credit card numbers and social security numbers, to unauthorized recipients.

Credit Card Check

The Credit Card check provides special handling for credit card numbers. A web application does not usually send a credit card number in a response to a user request, even when the user supplies a credit card number in the request. The application firewall examines web server responses, including headers, for credit card numbers. If it finds a credit card number in the response, and the administrator has not configured it to allow credit card numbers to be sent, it responds in one of two ways:

- It blocks the response.
- It replaces all but the final group of digits in the credit card with x's. For example, a credit card number of 9876-5432-1234-5678 would be rendered xxx-xxx-xxx-5678.

The Credit Card check prevents attackers from exploiting a security flaw in your web server software or on your web site to obtain credit card numbers of your customers. If your web sites do not have access to credit card information, you do not need to configure this check. If you have a shopping cart or other application that can access credit card numbers, or your web sites have access to database servers that contain credit card numbers, you should configure protection for each type of credit card that you accept.

Note: A web site that does not access an SQL database usually does not have access to sensitive private information such as credit card numbers.

If you use the wizard or the configuration utility, in the Credit Card Check dialog box, on the General tab you can enable or disable the Block, Log, and Statistics actions, and the following actions:

- **X-Out.** Mask any credit card number detected in a response by replacing each digit, except the digits in the final group, with the letter X.
- **Maximum credit cards allowed per page.** Allow up to the specified number of credit card numbers per page in responses without masking the credit card numbers or blocking the response. The Maximum is set to zero (0) by default. Web pages do not usually contain unmasked credit card numbers, but occasionally a web page might legitimately contain a credit card number or even a list of credit card numbers. To allow one or more credit card numbers to appear in a web page before masking the numbers or blocking the response change the value in the "Maximum credit cards allowed per page" text box to the number of credit cards that you want to allow.

To configure the types of credit cards to be protected, in the Modify Credit Card Check dialog box, select each credit card type that you want to protect, and then click Protect. If you want to cancel protection for a credit card type, select that credit card type and then click Unprotect. You can hold down your Shift or Ctrl key while choosing credit card types, and then enable or disable several credit card types at once by clicking the Protect or Unprotect button while multiple credit card types are selected.

If you use the command-line interface, you can enter the following commands to configure the Credit Card Check:

- set appfw profile <name> -creditCardAction [block] [log] [stats] [none]
- set appfw profile <name> -creditCard (VISA | MASTERCARD | DISCOVER | AMEX | JCB | DINERSCLUB)
- set appfw profile <name> -creditCardMaxAllowed <integer>
- set appfw profile <name> -creditCardXOut ([ON] | [OFF])

Safe Object Check

The Safe Object check provides user-configurable protection for sensitive business information, such as customer numbers, order numbers, and country-specific or region-specific telephone numbers or postal codes. A user-defined regular expression or custom plug-in tells the application firewall the format of this information and defines the rules to be used to protect it. If a string in a user request matches a safe object definition, the application firewall either blocks the response, masks the protected information, or removes the protected information from the response before sending it to the user, depending on how you configured that particular safe object rule.

The Safe Object check prevents attackers from exploiting a security flaw in your web server software or on your web site to obtain sensitive private information, such as company credit card numbers or social security numbers. If your web sites do not have access to these types of information, you do not need to configure this check. If you have a shopping cart or other application that can access such information, or your web sites have access to database servers that contain such information, you should configure protection for each type of sensitive private information that you handle and store.

Note: A web site that does not access an SQL database usually does not have access to sensitive private information.

The Safe Object Check dialog box is unlike that for any other check. Each safe object expression that you create is the equivalent of a separate security check, similar to the Credit Card check, for that type of information. If you use the wizard or the configuration utility, you add a new expression by clicking Add and configuring the expression in the Add Safe Object dialog box. You modify an existing expression by selecting it, then clicking Open, and then configuring the expression in the Modify Safe Object dialog box.

In the Safe Object dialog box for each safe object expression, you can configure the following:

- **Safe Object Name.** A name for your new safe object. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 255 letters, numbers, and the hyphen (-), period (.) pound (#), space (), at sign (@), equals (=), colon (:), and underscore (_) symbols.
- **Actions.** Enable or disable the Block, Log, and Statistics actions, and the following actions:
 - X-Out.** Mask any information that matches the safe object expression with the letter X.
 - Remove.** Remove any information that matches the safe object expression.
- **Regular Expression.** Enter a PCRE-compatible regular expression that defines the safe object. You can create the regular expression in one of three ways: by typing the regular expression directly into the text box by using the **Regex Tokens** menu to enter regular expression elements and symbols directly into the text box, or by opening the Regular Expressions Editor and using it to construct the expression. The regular expression must consist of ASCII characters only. Do not cut and paste characters that are not part of the basic 128-character ASCII set. If you want to include non-ASCII characters, you must manually type those characters in PCRE hexadecimal character encoding format.

Note: Do not use start anchors (^) at the beginning of Safe Object expressions, or end anchors (\$) at the end of Safe Object expressions. These PCRE entities are not supported in Safe Object expressions, and if used, will cause your expression not to match what it was intended to match.
- **Maximum Match Length.** Enter a positive integer that represents the maximum length of the string that you want to match. For example, if you want to match U.S. social security numbers, enter the number eleven (11) in this field. That allows your regular expression to match a string with nine numerals and two hyphens. If you want to match California driver's license numbers, enter the number eight (8).

Caution: If you do not enter a maximum match length in this field, the application firewall uses a default value of one (1) when filtering for strings that match your safe object expressions. As a result, most safe object expressions fail to match their target strings.

You cannot use the command-line interface to configure the Safe Object check. You must configure it by using either the application firewall wizard or the configuration utility.

Following are examples of Safe Object check regular expressions:

- Look for strings that appear to be U.S. social security numbers, which consist of three numerals (the first of which must not be zero), followed by a hyphen, followed by two more numerals, followed by a second hyphen, and ending with a string of four more numerals:

```
[1-9][0-9]{2,2}-[0-9]{2,2}-[0-9]{4,4}
```

- Look for strings that appear to be California driver's license IDs, which start with a letter and are followed by a string of exactly seven numerals:

```
[A-Za-z][0-9]{7,7}
```

- o Look for strings that appear to be Example Manufacturing customer IDs which, consist of a string of five hexadecimal characters (all the numerals and the letters A through F), followed by a hyphen, followed by a three-letter code, followed by a second hyphen, and ending with a string of ten numerals:

```
[0-9A-Fa-f]{5,5}-[A-Za-z]{3,3}-[0-9]{10,10}
```

Caution: Regular expressions are powerful. Especially if you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions you write to ensure that they define exactly the type of string you want to add as a safe object definition, and nothing else. Careless use of wildcards, and especially of the dot-asterisk (. *) metacharacter/wildcard combination, can have results you did not want or expect, such as blocking access to web content that you did not intend to block.

Advanced Form Protection Checks

The advanced Form Protection checks examine web form data to prevent attackers from compromising your system by modifying the web forms on your web sites or sending unexpected types and quantities of data to your web site in a form.

Field Formats Check

The Field Formats check verifies the data that users send to your web sites in a web form. It examines both the length and type of data to ensure that it is appropriate for the form field in which it appears. If the application firewall detects inappropriate web form data in a user request, it blocks the request. This check applies to HTML requests only. It does not apply to XML requests.

By preventing an attacker from sending inappropriate web form data to your web site, the Field Formats check prevents certain types of attacks on your web site and database servers. For example, if a particular field expects the user to enter a phone number, the Field Formats check examines the user's response to ensure that the data matches the format for a phone number. If a particular field expects a first name, the Field Formats check ensures that the data in that field is of a type and length appropriate for a first name. It does the same thing for each form field that you configure it to protect.

The Field Formats check provides a different type of protection than does the Form Field Consistency check. The Form Field Consistency check verifies that the structure of the web forms returned by users is intact, that data format restrictions configured in the HTML are respected, and that data in hidden fields has not been modified. It can do this without any specific knowledge about your web forms other than what it derives from the web form itself. The Field Formats check verifies that the data in each form field matches the specific formatting restrictions that you configured manually, or that the learning feature generated and you approved. In other words, the Form Field Consistency check enforces general web form security, while the Field Formats check enforces the specific rules that you set for your web forms.

Before it can protect your web forms, the Field Formats check requires that you configure the application firewall to recognize the type and length of data expected in each form field on each web form that you want to protect.

If you use the wizard or the configuration utility, in the Modify Field Formats Check dialog box, on the General tab you can enable or disable the Block, Log, Learn, and Statistics actions and the following parameters:

- **Field Type.** Assign a default field type to form fields in web forms that do not have a field type. This parameter is not set by default. You can assign any field type that is defined on your application firewall as the default field type.
Caution: If you set a restrictive default field type and do not disable blocking until you are certain that the field types assigned to your form fields are correct, users may be unable to use your web forms.
- **Minimum Length.** The default minimum data length assigned to form fields in web forms that do not have an explicit setting. This parameter is set to 0 by default, which allows the user to leave the field blank. Any higher setting forces users to fill in the field.
- **Maximum Length.** The default maximum data length assigned to form fields in web forms that do not have an explicit setting. This parameter is set to 65535 by default.

If you use the command-line interface, you can enter the following commands to configure the Field Formats Check:

- `set appfw profile <name> -fieldFormatAction [block] [learn] [log] [stats] [none]`
- `set appfw profile <name> -defaultFieldFormatType <string>`
- `set appfw profile <name> -defaultFieldFormatMinLength <integer>`
- `set appfw profile <name> -defaultFieldFormatMaxLength <integer>`

To specify relaxations for the Field Formats check, you must use the configuration utility. On the Checks tab of the Modify Field Formats Check dialog box, click Add to open the Add Field Formats Check Relaxation dialog box, or select an existing relaxation and click Open to open the Modify Field Formats Check Relaxation dialog box. Either dialog box provides the same options for configuring a relaxation.

Following are examples of Field Formats check relaxations:

- Choose form fields with the name FirstName:

```
^FirstName$
```

- Choose form fields with names that begin with Name_ and are followed by a string beginning with a letter or number and consisting of from one to twenty letters, numbers, or the apostrophe or hyphen symbol:

```
^Name_[0-9A-Za-z][0-9A-Za-z'-]{0,20}$
```

- Choose form fields with names that begin with Turkish-FirstName_ and are otherwise the same as the previous expression, except that they can contain Turkish special characters throughout:

`^T\xC3\xBCrk\xC3\xA7e-FirstName_([0-9A-Za-z]|\x[0-9A-Fa-f][0-9A-Fa-f])+ $`

- o Choose form field names that begin with a letter or number, consist of a combination of letters and/or numbers only, and that contain the string Num anywhere in the string:

`^[0-9A-Za-z]*Num[0-9A-Za-z]* $`

Form Field Consistency Check

The Form Field Consistency check examines the web forms returned by users of your web site, and verifies that web forms were not modified inappropriately by the client. This check applies only to HTML requests that contain a web form, with or without data. It does not apply to XML requests.

The Form Field Consistency check prevents clients from making unauthorized changes to the structure of the web forms on your web site when they fill out and submit a form. It also ensures that the data a user submits meets the HTML restrictions for length and type, and that data in hidden fields is not modified. This prevents an attacker from tampering with a web form and using the modified form to gain unauthorized access to web site, redirect the output of a contact form that uses an insecure script and thereby send unsolicited bulk email, or exploit a vulnerability in your web server software to gain control of the web server or the underlying operating system. Web forms are a weak link on many web sites and attract a wide range of attacks.

The Form Field Consistency check verifies all of the following:

- If a field is sent to the user, the check ensures that it is returned by the user.
- The check enforces HTML field lengths and types.
Note: The Form Field Consistency check enforces HTML restrictions on data type and length but does not otherwise validate the data in web forms. You can use the Field Formats check to set up rules that validate data returned in specific form fields on your web forms.
- If your web server does not send a field to the user, the check does not allow the user to add that field and return data in it.
- If a field is a read-only or hidden field, the check verifies that the data has not changed.
- If a field is a list box or radio button field, the check verifies that the data in the response corresponds to one of the values in that field.

If a web form returned by a user violates one or more of the Form Field consistency checks, and you have not configured the application firewall to allow that web form to violate the Form Field Consistency checks, the request is blocked.

If you use the wizard or the configuration utility, in the Modify Form Field Consistency Check dialog box, on the General tab you can enable or disable the Block, Log, Learn, and Statistics actions.

You also configure Sessionless Field Consistency in the General tab. If Sessionless Field Consistency is enabled, the application firewall checks only the web form structure, dispensing with those parts of the Form Field Consistency check that depend upon maintaining session information. This can speed the Form Field Consistency check with little security penalty for web sites that use many forms. To use Sessionless Field Consistency on all web forms, select On. To use it only for forms submitted with the HTTP POST method, select postOnly

If you use the command-line interface, you can enter the following command to configure the Form Field Consistency Check:

- `set appfw profile <name> -fieldConsistencyAction [block] [learn] [log] [stats] [none]`

To specify relaxations for the Form Field Consistency check, you must use the configuration utility. On the Checks tab of the Modify Form Field Consistency Check dialog box, click Add to open the Add Form Field Consistency Check Relaxation dialog box, or select an existing relaxation and click Open to open the Modify Form Field Consistency Check Relaxation dialog box. Either dialog box provides the same options for configuring a relaxation, as described in "[Manual Configuration By Using the Configuration Utility](#)."

Following are examples of Form Field Consistency check relaxations:

Form Field Names

- Choose form fields with the name UserType:

```
^UserType$
```

- Choose form fields with names that begin with UserType_ and are followed by a string that begins with a letter or number and consists of from one to twenty-one letters, numbers, or the apostrophe or hyphen symbol:

```
^UserType_[0-9A-Za-z][0-9A-Za-z'-]{0,20}$
```

- Choose form fields with names that begin with Turkish-UserType_ and are otherwise the same as the previous expression, except that they can contain Turkish special characters throughout:

```
^T\xC3\xBCrk\xC3\xA7e-UserType_([0-9A-Za-z]|\\x[0-9A-Fa-f][0-9A-Fa-f])+
```

Note: See "PCRE Character Encoding Format" for a complete description of supported special characters and how to encode them properly.

- Choose form field names that begin with a letter or number, consist of a combination of letters and/or numbers only, and that contain the string Num anywhere in the string:

```
^[0-9A-Za-z]*Num[0-9A-Za-z]*
```

Form Field Action URLs

- Choose URLs beginning with http://www.example.com/search.pl? and containing any string after the query except for a new query:

```
^http://www[.]example[.]com/search[.]pl\?[^\?]*
```

- Choose URLs that begin with http://www.example-español.com and have paths and filenames that consist of upper-case and lower-case letters, numbers, non-ASCII special characters, and selected symbols in the path. The Ñ character and any other special characters are represented as encoded UTF-8 strings containing the hexadecimal code assigned to each special character in the UTF-8 charset:

```
^http://www[.]example-espa\xC3\xB1ol[.]com/((([0-9A-Za-z]|\\x[0-9A-Fa-f][0-9A-Fa-f]([0-9A-Za-z_]|\\x[0-9A-Fa-f][0-9A-Fa-f]))*/)*([0-9A-Za-z]|\\x[0-9A-Fa-f][0-9A-Fa-f]([0-9A-Za-z_]|\\x[0-9A-Fa-f][0-9A-Fa-f]))*[\.](asp|http|php|s?html?))
```

- Choose all URLs that contain the string /search.cgi?:

```
^[^\<>]*/search[.]cgi\?[^\<>]*
```

Caution: Regular expressions are powerful. Especially if you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions you write. Make sure that they define exactly the URL you want to add as an exception, and nothing else. Careless use of wildcards, and especially of the dot-asterisk (. *) metacharacter/wildcard combination, can have results you do not want or expect, such as blocking access to web content that you did not intend to block or allowing an attack that the Cookie Consistency check would otherwise have blocked.

CSRF Form Tagging Check

The Cross Site Request Forgery (CSRF) Form Tagging check tags each web form sent by a protected web site to users with a unique and unpredictable FormID, and then examines the web forms returned by users to ensure that the supplied FormID is correct. This check protects against cross-site request forgery attacks. This check applies only to HTML requests that contain a web form, with or without data. It does not apply to XML requests.

The CSRF Form Tagging check prevents attackers from using their own web forms to send high volume form responses with data to your protected web sites. This check requires relatively little CPU processing capacity compared to certain other security checks that analyze web forms in depth. It is therefore able to handle high volume attacks without seriously degrading the performance of the protected web site or the application firewall itself.

Before you enable the CSRF Form Tagging check, you should be aware of the following:

- You need to enable form tagging. The CSRF check depends on form tagging and does not work without it.
- You should disable the Citrix NetScaler Integrated Caching feature for all web pages containing forms that are protected by that profile. The Integrated Caching feature and CSRF form tagging are not compatible.
- You should consider enabling Referer checking. Referer checking is part of the Start URL check, but it prevents cross-site request forgeries, not Start URL violations. Referer checking also puts less load on the CPU than does the CSRF Form Tagging check. If a request violates Referer checking, it is immediately blocked, so the CSRF Form Tagging check is not invoked.
- The CSRF Form Tagging check does not work with web forms that use different domains in the form-origin URL and form-action URL. For example, CSRF Form Tagging cannot protect a web form with a form-origin URL of `http://www.example.com/` and a form action URL of `http://www.example.org/form.pl`, because `example.com` and `example.org` are different domains.

If you use the wizard or the configuration utility, in the Modify CSRF Form Tagging Check dialog box, on the General tab you can enable or disable the Block, Log, Learn and Statistics actions.

If you use the command-line interface, you can enter the following command to configure the CSRF Form Tagging Check:

- `set appfw profile <name> -fieldConsistencyAction [block] [log] [learn] [stats] [none]`

To specify relaxations for the CSRF Form Tagging check, you must use the configuration utility. On the Checks tab of the Modify CSRF Form Tagging Check dialog box, click Add to open the Add CSRF Form Tagging Check Relaxation dialog box, or select an existing relaxation and click Open to open the Modify CSRF Form Tagging Check Relaxation dialog box. Either dialog box provides the same options for configuring a relaxation.

Following are examples of CSRF Form Tagging check relaxations:

Note: The following expressions are URL expressions that can be used in both the Form Origin URL and Form Action URL roles.

- Choose URLs beginning with `http://www.example.com/search.pl?` and containing any string after the query, except for a new query:

```
^http://www[.]example[.]com/search[.]pl\?[^\?]*$
```

- Choose URLs that begin with `http://www.example-español.com` and have paths and filenames that consist of upper-case and lower-case letters, numbers, non-ASCII special characters, and selected symbols in the path. The `Ã±` character and any other special characters are represented as encoded UTF-8 strings containing the hexadecimal code assigned to each special character in the UTF-8 charset:

```
^http://www[.]example-espa\xC3\xB1ol[.]com/((( [0-9A-Za-z]|\x[0-9A-Fa-f][0-9A-Fa-f]
([0-9A-Za-z_-]|\x[0-9A-Fa-f][0-9A-Fa-f])*)/)*([0-9A-Za-z]|\x[0-9A-Fa-f][0-9A-Fa-f]
([0-9A-Za-z_-]|\x[0-9A-Fa-f][0-9A-Fa-f])*)[.](asp|http|php|s?html?)$
```

- Choose all URLs that contain the string `/search.cgi?`:

```
^[^\?<>]*/search[.]cgi\?[^\?<>]*$
```

Caution: Regular expressions are powerful. Especially if you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions you write. Make sure that they define exactly the URL that you want to add as an exception, and nothing else. Careless use of wildcards, and especially of the dot-asterisk (`. *`)

metacharacter/wildcard combination, can have results you do not want, such as blocking access to web content that you did not intend to block or allowing an attack that the check would otherwise have blocked.

Managing CSRF Form Tagging Check Relaxations

You configure an exception (or relaxation) to the CSRF Form Tagging security check in the Add Cross-Site Request Forgery Tagging Check Relaxation dialog box or the Modify Cross-Site Request Forgery Tagging Check Relaxation dialog box.

To configure a CSRF Form Tagging check relaxation by using the configuration utility

1. Navigate to Security > Application Firewall > Profiles.
2. In the Profiles pane, select the profile you want to configure, and then click Open.
3. In the Configure Application Firewall Profile dialog box, click the Security Checks tab. The Security Checks tab contains the list of application firewall security checks.
4. In the Security Checks window, click CSRF Form Tagging, and then click Open. The Modify Cross-Site Request Forgery Tagging Check dialog box is displayed, with the Checks tab selected. The Checks tab contains a list of existing CSRF relaxations. The list might be empty if you have not either manually added any relaxations or approved any relaxations that were recommended by the learning engine. Beneath the list is a row of buttons that allow you to add, modify, delete, enable, or disable the relaxations on the list.
5. To add or modify a CSRF relaxation, do one of the following:
 - o To add a new relaxation, click Add.
 - o To modify an existing relaxation, select the relaxation that you want to modify, and then click Open.

The Add Cross-Site Request Forgery Tagging Check Relaxation or Modify Cross-Site Request Forgery Tagging Check Relaxation dialog box is displayed. Except for the title, these dialog boxes are identical.

6. Fill in the dialog box as described below.

- o **Enabled check box**—Select to place this relaxation or rule in active use; clear to deactivate it.
- o **Form Origin URL**—In the text area, enter a PCRE-format regular expression that defines the URL that hosts the form.
- o **Form Action URL**—In the text area, enter a PCRE-format regular expression that defines the URL to which data entered into the form is delivered.
- o **Comments**—In the text area, type a comment. Optional.

Note: For any element that requires a regular expression, you can type the regular expression, use the Regex Tokens menu to insert regular expression elements and symbols directly into the text box, or click Regex Editor to open the Add Regular Expression dialog box, and use it to construct the expression.

7. Click OK. The Add Cross-Site Request Forgery Tagging Check Relaxation or Modify Cross-Site Request Forgery Tagging Check Relaxation dialog box closes and you return to the Modify Cross-Site Request Forgery Tagging Check dialog box.
8. To remove a relaxation or rule, select it, and then click Remove.
9. To enable a relaxation or rule, select it, and then click Enable.
10. To disable a relaxation or rule, select it, and then click Disable.
11. To configure the settings and relationships of all existing relaxations in an integrated interactive graphic display, click Visualizer, and use the display tools.
12. To review and configure learned rules for the CSRF check, click Learning and perform the steps in "To configure and use the Learning feature."
13. Click OK.

URL Protection Checks

The URL Protection checks examine request URLs to prevent attackers from aggressively attempting to access multiple URLs (forceful browsing) or using a URL to trigger a known security vulnerability in web server software or web site scripts.

Start URL Check

The Start URL check examines the URLs in incoming requests and blocks the connection attempt if the URL does not meet the specified criteria. To meet the criteria, the URL must match an entry in the Start URL list, unless the Enforce URL Closure parameter is enabled. If you enable this parameter, a user who clicks a link on your Web site is connected to the target of that link.

The primary purpose of the Start URL check is to prevent repeated attempts to access random URLs on a Web site, (forceful browsing). Forceful browsing can be used to trigger a buffer overflow, find content that users were not intended to access directly, or find a back door into secure areas of your Web server.

If you use the wizard or the configuration utility, in the Modify Start URL Check dialog box, on the General tab you can enable or disable Block, Log, Statistics, Learn actions, and the following parameters:

- o **Enforce URL Closure.** Allow users to access any web page on your web site by clicking a hyperlink on any other page on your web site. Users can navigate to any page on your web site that can be reached from the home page or any designated start page by clicking hyperlinks.

Note: The URL closure feature allows any query string to be appended to and sent with the action URL of a web form submitted by using the HTTP GET method. If your protected web sites use forms to access an SQL database, make sure that you have the SQL injection check enabled and properly configured.

- o **Sessionless URL Closure.** From the client's point of view, this type of URL closure functions in exactly the same way as standard, session-aware URL Closure, but uses a token embedded in the URL instead of a cookie to track the user's activity, which consumes considerably fewer resources.

Note: When enabling sessionless (Sessionless URL Closure), you must also enable regular URL closure (Enforce URL Closure) or sessionless URL closure does not work.

- o **Validate Referer Header.** Verify that the Referer header in a request that contains web form data from your protected web site instead of another web site. This action verifies that your web site, not an outside attacker, is the source of the web form. Doing so protects against cross-site request forgeries (CSRF) without requiring form tagging, which is more CPU-intensive than header checks. The application firewall can handle the HTTP Referer header in one of the following three ways, depending on which option you select in the drop-down list:

Off. Do not validate the Referer header.

If-Present. Validate the Referer header if a Referer header exists. If an invalid Referer header is found, the request generates a referer-header violation. If no Referer header exists, the request does not generate a referer-header violation. This option enables the application firewall to perform Referer header validation on requests that contain a Referer header, but not block requests from users whose browsers do not set the Referer header or who use web proxies or filters that remove that header.

Always. Always validate the Referer header. If there is no Referer header, or if the Referer header is invalid, the request generates a referer-header violation.

Note: Although the referer header check and Start URL security check share the same action settings, it is possible to violate the referer header check without violating the Start URL check. The difference is visible in the logs, which log referer header check violations separately from Start URL check violations.

One Start URL setting, Exempt Closure URLs from Security Checks, is not configured in the Modify Start URL Check dialog box, but in the Settings tab of the Configure Application Firewall Profile dialog box. If enabled, this setting directs the application firewall not to run further security checks on URLs that meet the URL Closure criteria.

If you use the command-line interface, you can enter the following commands to configure the Start URL Check:

- o set appfw profile <name> -startURLAction [**block**] [**learn**] [**log**] [**stats**] [**none**]
- o set appfw profile <name> -startURLClosure ([**ON**] | [**OFF**])
- o set appfw profile <name> -sessionlessURLClosure ([**ON**] | [**OFF**])
- o set appfw profile <name> -exemptClosureURLsFromSecurityChecks ([**ON**] | [**OFF**])
- o set appfw profile <name> -RefererHeaderCheck ([**none**] | [**if-present**] | [**always**])

To specify relaxations for the Start URL check, you must use the configuration utility. On the Checks tab of the Modify Start URL Check dialog box, click Add to open the Add Start URL Check Relaxation dialog box, or select an existing relaxation and click Open to open the Modify Start URL Check Relaxation dialog box. Either dialog box provides the same options for configuring a relaxation.

Following are examples of Start URL check relaxations:

- o Allow users to access the home page at `www.example.com`:

```
^http://www[.]example[.]com$
```

- o Allow users to access all static HTML (.htm and .html), server-parsed HTML (.http and .shtml), PHP (.php), and Microsoft ASP (.asp) format web pages at www.example.com:

```
^http://www[.]example[.]com/([0-9A-Za-z][0-9A-Za-z_-]*/)*  
[0-9A-Za-z][0-9A-Za-z_-]*[.](asp|http|php|s?html?)$
```

- o Allow users to access web pages with pathnames or file names that contain non-ASCII characters at www.example-español.com:

```
^http://www[.]example-espa\xC3\xB1ol[.]com/(([0-9A-Za-z]|\\x[0-9A-Fa-f][0-9A-Fa-f]  
[0-9A-Za-z]|\\x[0-9A-Fa-f][0-9A-Fa-f])([0-9A-Za-z_-]|\\x[0-9A-Fa-f][0-9A-Fa-f])
```

Note: In the above expression, each character class has been grouped with the string `\\x[0-9A-Fa-f][0-9A-Fa-f]`, which matches all properly-constructed character encoding strings but does not allow stray backslash characters that are not associated with a UTF-8 character encoding string. The double backslash (`\\`) is an escaped backslash, which tells the application firewall to interpret it as a literal backslash. If you included only one backslash, the application firewall would instead interpret the following left square bracket (`[`) as a literal character instead of the opening of a character class, which would break the expression.

- o Allow users to access all GIF (.gif), JPEG (.jpg and .jpeg), and PNG (.png) format graphics at www.example.com:

```
^http://www[.]example[.]com/([0-9A-Za-z][0-9A-Za-z_-]*/)*  
[0-9A-Za-z][0-9A-Za-z_-]*[.](gif|jpe?g|png)$
```

- o Allow users to access CGI (.cgi) and PERL (.pl) scripts, but only in the CGI-BIN directory:

```
^http://www[.]example[.]com/CGI-BIN/[0-9A-Za-z][0-9A-Za-z_-]*[.](cgi|pl)$
```

- o Allow users to access Microsoft Office and other document files in the docsarchive directory:

```
^http://www[.]example[.]com/docsarchive/[0-9A-Za-z][0-9A-Za-z_-]*[.](doc|xls|pc
```

Caution: Regular expressions are powerful. Especially if you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions that you write. Make sure that they define exactly the URL you want to add as an exception, and nothing else. Careless use of wildcards, and especially of the dot-asterisk (`.*`) metacharacter/wildcard combination, can have results you do not want, such as blocking access to web content that you did not intend to block or allowing an attack that the Start URL check would otherwise have blocked.

Deny URL Check

The Deny URL check examines and blocks connections to URLs that are commonly accessed by hackers and malicious code. This check contains a list of URLs that are common targets of hackers or malicious code and that rarely if ever appear in legitimate requests. You can also add URLs or URL patterns to the list. The Deny URL check prevents attacks against various security weaknesses known to exist in web server software or on many web sites.

The Deny URL check takes priority over the Start URL check, and thus denies malicious connection attempts even when a Start URL relaxation would normally allow a request to proceed.

In the Modify Deny URL Check dialog box, on the General tab you can enable or disable the Block, Log, and Statistics actions.

If you use the command-line interface, you can enter the following command to configure the Deny URL Check:

- o `set appfw profile <name> -denyURLAction [block] [log] [stats] [none]`

To create and configure your own deny URLs, you must use the configuration utility. On the Checks tab of the Modify Deny URL Check dialog box, click Add to open the Add Deny URL dialog box, or select an existing user-defined deny URL and click Open to open the Modify Deny URL dialog box. Either dialog box provides the same options for creating and configuring a deny URL.

Following are examples of Deny URL expressions:

- o Do not allow users to access the image server at `images.example.com` directly:

```
^http://images[.]example[.]com$
```

- o Do not allow users to access CGI (.cgi) or PERL (.pl) scripts directly:

```
^http://www[.]example[.]com/([0-9A-Za-z][0-9A-Za-z_-]*/)*[0-9A-Za-z][0-9A-Za-z_-]*[.](cgi|pl)$
```

- o Here is the same deny URL, modified to support non-ASCII characters:

```
^http://www[.]example[.]com/(([0-9A-Za-z]|\\x[0-9A-Fa-f][0-9A-Fa-f])([0-9A-Za-z_-]|\\x[0-9A-Fa-f][0-9A-Fa-f])*/)*([0-9A-Za-z]|\\x[0-9A-Fa-f][0-9A-Fa-f])([0-9A-Za-z_-]|\\x[0-9A-Fa-f][0-9A-Fa-f])*[.](cgi|pl)$
```

Caution: Regular expressions are powerful. Especially if you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions you write. Make sure that they define exactly the URL or pattern that you want to block, and nothing else. Careless use of wildcards, and especially of the dot-asterisk (. *) metacharacter/wildcard combination, can have results that you do not want, such as blocking access to web content that you did not intend to block.

XML Protection Checks

The XML Protection checks examine requests for XML-based attacks of all types.

Caution: The XML security checks apply only to content that is sent with an HTTP content-type header of text/xml. If the content-type header is missing, or is set to a different value, all XML security checks are bypassed. If you plan to protect XML or Web 2.0 web applications, the webmasters of each web server that hosts those applications should ensure that the proper HTTP content-type header is sent.

XML Format Check

The XML Format check examines the XML format of incoming requests and blocks those requests that are not well formed or that do not meet the criteria in the XML specification for properly-formed XML documents. Some of those criteria are:

- An XML document must contain only properly-encoded Unicode characters that match the Unicode specification.
- No special XML syntax characters such as `<`, `>` and `&` can be included in the document except when used in XML markup.
- All begin, end, and empty-element tags must be correctly nested, with none missing or overlapping.
- XML element tags are case-sensitive. All beginning and end tags must match exactly.
- A single root element must contain all the other elements in the XML document.

A document that does not meet the criteria for well-formed XML does not meet the definition of an XML document. Strictly speaking, it is not XML. However, not all XML applications and web services enforce the XML well-formed standard, and not all handle poorly-formed or invalid XML correctly. Inappropriate handling of a poorly-formed XML document can cause security breaches. The purpose of the XML Format check is to prevent a malicious user from using a poorly-formed XML request to breach security on your XML application or web service.

If you use the wizard or the configuration utility, in the Modify XML Format Check dialog box, on the General tab you can enable or disable the Block, Log, and Statistics actions.

If you use the command-line interface, you can enter the following command to configure the XML Format Check:

- `set appfw profile <name> -xmlFormatAction [block] [log] [stats] [none]`

You cannot configure exceptions to the XML Format check. You can only enable or disable it.

XML Denial-of-Service Check

The XML Denial of Service (XML DoS or XDoS) check examines incoming XML requests to determine whether they match the characteristics of a denial-of-service (DoS) attack, and blocks those requests that do. The purpose of the XML DoS check is to prevent an attacker from using XML requests to launch a denial-of-service attack on your web server or web site.

If you use the wizard or the configuration utility, in the Modify XML Denial-of-Service Check dialog box, on the General tab you can enable or disable the Block, Log, Statistics, and Learn actions:

If you use the command-line interface, you can enter the following command to configure the XML Denial-of-Service check:

- o set appfw profile <name> -xmlDoSAction [block] [log] [learn] [stats] [none]

To configure individual XML Denial-of-Service rules, you must use the configuration utility. On the Checks tab of the Modify XML Denial-of-Service Check dialog box, select a rule and click Open to open the Modify XML Denial-of-Service dialog box for that rule. The individual dialog boxes differ for the different rules but are extremely simple. Some only allow you to enable or disable the rule; others allow you to modify a number by typing a new value in a text box.

The individual XML Denial-of-Service rules are:

Maximum Element Depth

Restrict the maximum number of nested levels in each individual element to 256. If this rule is enabled, and the application firewall detects an XML request with an element that has more than the maximum number of allowed levels, it blocks the request. You can modify the maximum number of levels to any value from one (1) to 65,535.

Maximum Element Name Length

Restrict the maximum length of each element name to 128 characters. This includes the name within the expanded namespace, which includes the XML path and element name in the following format:

```
{http://prefix.example.com/path/}target_page.xml
```

The user can modify the maximum name length to any value between one (1) character and 65,535.

Maximum # Elements

Restrict the maximum number of any one type of element per XML document to 65,535. You can modify the maximum number of elements to any value between one (1) and 65,535.

Maximum # Element Children

Restrict the maximum number of children (including other elements, character information, and comments) each individual element is allowed to have to 65,535. You can modify the maximum number of element children to any value between one (1) and 65,535.

Maximum # Attributes

Restrict the maximum number of attributes each individual element is allowed to have to 256. You can modify the maximum number of attributes to any value between one (1) and 256.

Maximum Attribute Name Length

Restrict the maximum length of each attribute name to 128 characters. You can modify the maximum attribute name length to any value between one (1) and 2,048.

Maximum Attribute Value Length

Restrict the maximum length of each attribute value to 2048 characters. You can modify the maximum attribute name length to any value between one (1) and 2,048.

Maximum Character Data Length

Restrict the maximum character data length for each element to 65,535. You can modify the length to any value between one (1) and 65,535.

Maximum File Size

Restrict the size of each file to 20 MB. You can modify the maximum file size to any value.

Minimum File Size

Require that each file be at least 9 bytes in length. You can modify the minimum file size to any positive integer representing a number of bytes.

Maximum # Entity Expansions

Limit the number of entity expansions allowed to the specified number. Default: 1024.

Maximum Entity Expansion Depth

Restrict the maximum number of nested entity expansions to no more than the specified number. Default: 32.

Maximum # Namespaces

Limit the number of namespace declarations in an XML document to no more than the specified number. Default: 16.

Maximum Namespace URI Length

Limit the URL length of each namespace declaration to no more than the specified number of characters. Default: 256.

Block Processing Instructions

Block any special processing instructions included in the request. This rule has no user-modifiable values.

Block DTD

Block any document type definitions (DTD) included with the request. This rule has no user-modifiable values.

Block External Entities

Block all references to external entities in the request. This rule has no user-modifiable values.

SOAP Array Check

Enable or disable the following SOAP array checks:

- **Maximum SOAP Array Size.** The maximum total size of all SOAP arrays in an XML request before the connection is blocked. You can modify this value. Default: 20000000.
- **Maximum SOAP Array Rank.** The maximum rank or dimensions of any single SOAP array in an XML request before the connection is blocked. You can modify this value. Default: 16.

XML Cross-Site Scripting Check

The XML Cross-Site Scripting check examines the user requests for possible cross-site scripting attacks in the XML payload. If it finds a possible cross-site scripting attack, it blocks the request.

To prevent misuse of the scripts on your protected web services to breach security on your web services, the XML Cross-Site Scripting check blocks scripts that violate the same origin rule, which states that scripts should not access or modify content on any server but the server on which they are located. Any script that violates the same origin rule is called a cross-site script, and the practice of using scripts to access or modify content on another server is called cross-site scripting. The reason cross-site scripting is a security issue is that a web server that allows cross-site scripting can be attacked with a script that is not on that web server, but on a different web server, such as one owned and controlled by the attacker.

Unfortunately, many companies have a large installed base of JavaScript-enhanced web content that violates the same origin rule. If you enable the XML Cross-Site Scripting check on such a site, you have to generate the appropriate exceptions so that the check does not block legitimate activity.

Actions: If you use the wizard or the configuration utility, in the XML Cross-Site Scripting Check dialog box, on the General tab you can enable or disable the Block, Log, and Statistics actions.

If you use the command-line interface, you can enter the following command to configure the actions for the XML Cross-Site Scripting Check:

- o set appfw profile <name> -XMLXSSAction [**block**] [**log**] [**stats**] [**none**]

Relaxations: You can use the configuration utility to specify relaxations for the XML Cross-Site Scripting check. On the Checks tab of the Modify XML Cross-Site Scripting Check dialog box, click Add to open the Add XML Cross-Site Scripting Check Relaxation dialog box, or select an existing relaxation and click Open to open the Modify XML Cross-Site Scripting Check Relaxation dialog box to edit an existing rule. Either dialog box provides the same options for configuring a relaxation, as described in ["Manual Configuration By Using the Configuration Utility."](#)

The XML Cross-Site Scripting check relaxation rules have the following parameters :

- o **Name:** You can use literal strings or Regular Expressions to configure the name. The following expression exempts all elements beginning with the string `name_` followed by a string of upper- and lower-case letters or numbers that is at least two characters long and no more than fifteen characters long:

```
^name_[0-9A-Za-z]{2,15}$
```

Caution: Regular expressions are powerful. Especially if you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions you write. Make sure that they define exactly the name that you want to add as an exception, and nothing else. Careless use of Regular Expressions can have results that you do not want, such as blocking access to web content that you did not intend to block or allowing an attack that the XML Cross-Site Scripting check would otherwise have blocked.

- o **Location** You can specify the Location of the Cross-site Scripting Check exception in your XML payload. The option ELEMENT is selected by Default. You can change it to select ATTRIBUTE.
- o **Comment** This is an optional field. You can use up to a 255 character long string to describe the purpose of this relaxation Rule.

XML SQL Injection Check

The XML SQL Injection check examines both the headers and the bodies of user requests for possible XML SQL Injection attacks. If it finds injected SQL, it blocks the request.

To prevent misusing the scripts on your protected web services to breach security on your web services, the XML SQL Injection check blocks scripts that violate the same origin rule, which states that scripts should not access or modify content on any server but the server on which they are located. Any script that violates the same origin rule is called a cross-site script, and the practice of using scripts to access or modify content on another server is called XML SQL Injection. The reason XML SQL Injection is a security issue is that a web server that allows XML SQL Injection can be attacked with a script that is not on that web server, but on a different web server, such as one owned and controlled by the attacker.

Unfortunately, many companies have a large installed base of JavaScript-enhanced web content that violates the same origin rule. If you enable the XML SQL Injection check on such a site, you have to generate the appropriate exceptions so that the check does not block legitimate activity. In addition, to prevent blocking of legitimate requests, this check ignores cookies that were set by the server, even if they contain elements that the Cookie Consistency check would otherwise block. You should keep this in mind when configuring this check.

Note: To prevent blocking of legitimate requests, this check ignores cookies that were set by the server, even if they contain elements that the Cookie Consistency check would otherwise block.

If you use the wizard or the configuration utility, in the Modify XML SQL Injection Check dialog box, on the General tab you can enable or disable Block, Log, and Statistics actions, and the following parameters:

- **Restrict checks to fields containing SQL special characters.** If you configure the application firewall to check only fields that contain SQL special strings, the application firewall skips web form fields that do not contain special characters. Since most SQL servers do not process SQL commands that are not preceded by a special character, enabling this parameter can significantly reduce the load on the application firewall and speed up processing without placing your protected web sites at risk.
 - **SQL comments handling.** By default, the application firewall checks all SQL comments for injected SQL commands. Many SQL servers ignore anything in a comment, however, even if it is preceded by an SQL special character. For faster processing, if your SQL server ignores comments, you can configure the application firewall to skip comments when examining requests for injected SQL. The SQL comments handling options are:
 - ANSI.** Skip ANSI-format SQL comments, which are normally used by UNIX-based SQL databases.
 - Nested.** Skip nested SQL comments, which are normally used by Microsoft SQL Server.
 - ANSI/Nested.** Skip comments that adhere to both the ANSI and nested SQL comment standards. Comments that match only the ANSI standard, or only the nested standard, are checked for injected SQL.

Caution: In most cases, you should not choose the Nested or the ANSI/Nested option unless your database runs on Microsoft SQL Server. Most other types of SQL server software do not recognize nested comments. If nested comments appear in a request directed to another type of SQL server, they may indicate an attempt to breach security on that server.

 - Check all Comments.** Check the entire request for injected SQL, without skipping anything. The default setting.
- **Check Request headers.** If Request header checking is enabled, the application firewall examines the headers of requests for XML SQL Injection attacks, instead of just URLs.

Caution: If you enable both request header checking and transformation, any SQL special characters found in headers are also transformed. The Accept, Accept-Charset, Accept-Encoding, Accept-Language, Expect, and User-Agent headers normally contain semicolons (;), so enabling both Request header checking and transformation simultaneously may cause errors.

If you use the command-line interface, you can enter the following commands to configure the XML SQL Injection Check:

- set appfw profile <name> -XMLSQLInjectionAction [**block**] [**learn**] [**log**] [**stats**] [**none**]
- set appfw profile <name> -XMLSQLInjectionOnlyCheckFieldsWithSQLChars (**ON** | **OFF**)]
- set appfw profile <name> -XMLSQLInjectionParseComments ([**checkall**] | [**ansi** | **nested**] | [**ansinested**])

You configure the exceptions to the XML SQL Injection check by opening the Modify XML SQL Injection Check dialog box, Checks tab. An exception can consist of either a literal string or a PCRE-format regular expression. For information about adding, modifying, removing, enabling, or disabling exceptions, see ["Manual Configuration By Using the Configuration Utility."](#)

Following are examples of XML SQL Injection check relaxations:

- o **Name element or attribute.** The following expression exempts all elements beginning with the string `name_` followed by a string of upper- and lower-case letters or numbers that is at least two characters long and no more than fifteen characters long:

```
^name_[0-9A-Za-z]{2,15}$
```

- o **URL element or attribute.** The following expression exempts URLs with hostnames of `web.example.com`, with a path up to four levels deep followed by an optional file name and extension, but no HTML or query symbols :

```
^https?://web[.]example[.]com(/[<>?]{1,30}){0,4}(/[<>?]{1,30})*$
```

- o **URL element or attribute (special characters).** The following expression exempts URLs with hostnames of `web.tÃ¼rkÃ§e-example.com`, with the same path and file restrictions as above:

```
^https?://web[.]t\xC3\xBCrk\xC3\xA7e-example[.]com(/[<>?]{1,30}){0,4}(/[<>?]{1,30})*$
```


XML Attachment Check

The XML Attachment check examines incoming requests for malicious attachments, and it blocks those requests that contain attachments that might breach applications security. The purpose of the XML Attachment check is to prevent an attacker from using an XML attachment to breach security on your server.

If you use the wizard or the configuration utility, in the Modify XML Attachment Check dialog box, on the General tab you can enable or disable the Block, Learn, Log, Statistics, and Learn actions:

If you use the command-line interface, you can enter the following command to configure the XML Attachment Check:

- o `set appfw profile <name> -xmlAttachmentAction [block] [learn] [log] [stats] [none]`

You must configure the other XML Attachment check settings in the configuration utility. In the Modify XML Attachment Check dialog box, on the Checks tab, you can configure the following settings:

- o **Maximum Attachment Size.** Allow attachments that are no larger than the maximum attachment size you specify. To enable this option, first select the Enabled check box, and then type the maximum attachment size in bytes in the Size text box.
- o **Attachment Content Type.** Allow attachments of the specified content type. To enable this option, first select the Enabled check box, and then enter a regular expression that matches the Content-Type attribute of the attachments that you want to allow.
You can type the URL expression directly in the text window. If you do so, you can use the Regex Token menu to enter a number of useful regular expressions at the cursor instead of typing them manually. You can click Regex Editor to open the Add Regular Expression dialog box and use it to construct the URL expression.

Web Services Interoperability Check

The Web Services Interoperability (WS-I) check examines both requests and responses for adherence to the WS-I standard, and blocks those requests and responses that do not adhere to this standard. The purpose of the WS-I check is to block requests that might not interact with other XML appropriately. An attacker can use inconsistencies in interoperability to launch an attack on your XML application.

If you use the wizard or the configuration utility, in the Modify Web Services Interoperability Check dialog box, on the General tab you can enable or disable the Block, Log, Statistics, and Learn actions.

If you use the command-line interface, you can enter the following command to configure the Web Services Interoperability check:

- o `set appfw profile <name> -xmlWSIAction [block] [log] [learn] [stats] [none]`

To configure individual Web Services Interoperability rules, you must use the configuration utility. On the Checks tab of the Modify Web Services Interoperability Check dialog box, select a rule and click Enable or Disable to enable or disable the rule. You can also click Open to open the Web Services Interoperability Detail message box for that rule. The message box displays read-only information about the rule. You cannot modify or make other configuration changes to any of these rules.

XML Message Validation Check

The XML Message Validation check examines requests that contain XML messages to ensure that they are valid. If a request contains an invalid XML message, the application firewall blocks the request. The purpose of the XML Validation check is to prevent an attacker from using specially constructed invalid XML messages to breach the security of your application.

If you use the wizard or the configuration utility, in the Modify XML Message Validation Check dialog box, on the General tab you can enable or disable the Block, Log, and Statistics actions.

If you use the command-line interface, you can enter the following command to configure the XML Message Validation Check:

- o set appfw profile <name> -xmlValidationAction [**block**] [**log**] [**stats**] [**none**]

You must use the configuration utility to configure the other XML Validation check settings. In the Modify XML Message Validation Check dialog box, on the Checks tab, you can configure the following settings:

- o **XML Message Validation.** Use one of the following options to validate the XML message:
 - SOAP Envelope.** Validate only the SOAP envelope of XML messages.
 - WSDL.** Validate XML messages by using an XML SOAP WSDL. If you choose WSDL validation, in the WSDL Object drop-down list you must choose a WSDL. If you want to validate against a WSDL that has not already been imported to the application firewall, you can click the Import button to open the Manage WSDL Imports dialog box and import your WSDL. See "[WSDL](#)" for more information.
 - If you want to validate the entire URL, leave the Absolute radio button in the End Point Check button array selected. If you want to validate only the portion of the URL after the host, select the Relative radio button.
 - If you want the application firewall to enforce the WSDL strictly, and not allow any additional XML headers not defined in the WSDL, you must clear the Allow additional headers not defined in the WSDL check box.
Caution: If you uncheck the Allow Additional Headers not defined in the WSDL check box, and your WSDL does not define all XML headers that your protected XML application or Web 2.0 application expects or that a client sends, you may block legitimate access to your protected service.
- **XML Schema.** Validate XML messages by using an XML schema. If you choose XML schema validation, in the XML Schema Object drop-down list you must choose an XML schema. If you want to validate against an XML schema that has not already been imported to the application firewall, you can click the Import button to open the Manage XML Schema Imports dialog box and import your WSDL. See "[WSDL](#)" for more information.
- o **Response Validation.** By default, the application firewall does not attempt to validate responses. If you want to validate responses from your protected application or Web 2.0 site, select the Validate Response check box. When you do, the Reuse the XML Schema specified in request validation check box and the XML Schema Object drop-down list are activated.
 - Check the Reuse XML Schema check box to use the schema you specified for request validation to do response validation as well.
Note: If you check this check box, the XML Schema Object drop-down list is grayed out.
 - If you want to use a different XML schema for response validation, use the XML Schema Object drop-down list to select or upload that XML schema .

XML SOAP Fault Filtering Check

The XML SOAP Fault Filtering check examines responses from your protected web services and filters out XML SOAP faults. This prevents leaking of sensitive information to attackers.

If you use the wizard or the configuration utility, in the Modify XML SOAP Fault Filtering Check dialog box, on the General tab you can enable or disable the Block, Log, and Statistics actions, and the Remove action, which removes SOAP faults before forwarding the response to the user.

If you use the command-line interface, you can enter the following command to configure the XML SOAP Fault Filtering Check:

```
set appfw profile <name> -XMLSOAPFaultAction [block] [log] [stats] [none]
```

You cannot configure exceptions to the XML SOAP Fault Filtering check. You can only enable or disable it.

Managing Content Types

Web servers usually add a Content-Type header that contains a MIME/type definition for the type of content in each file that the web server serves to users. Web servers serve many different types of content. For example, standard HTML is assigned the "text/html" MIME type. JPG images are assigned the "image/jpeg" or "image/jpg" content type. A normal web server can serve dozens or hundreds of different types of content, all defined in the Content Type header by an assigned MIME/type.

Many application firewall filtering rules are designed to filter specific types of content. Because filtering rules that apply to one type of content (such as HTML) are often inappropriate when filtering a different type of content (such as images), the application firewall attempts to determine the content type of requests and responses before it filters them. When a web server or browser does not add a Content-Type header to a request or response, the application firewall applies a default content type to the connection and filters the content accordingly.

The default content type is normally "application/octet-stream", the most generic MIME/type definition. This MIME/type is appropriate for any type of content that a web server is likely to serve, but also does not provide much information to the application firewall to allow it to choose appropriate filtering. If a protected web server on your network is configured to add accurate content type headers to the content it serves, or serves only one type of content, you can create a profile for that web server and assign a different default content type to it to improve both the speed and the accuracy of filtering.

You can also configure a list of allowed response content types for a specific profile. When this feature is configured, if the application firewall filters a response that does not match one of the allowed content types, it blocks the response.

Requests must always be of either the "application/x-www-form-urlencoded" or "multipart/form-data" types. The application firewall bypasses any request that has any other content type designated.

Note: You cannot include the "application/x-www-form-urlencoded" or "multipart/form-data" content types on the allowed response content types list.

To set the default request content type by using the command line interface

At the command prompt, type the following commands:

- o set appfw profile <name> -requestContentType <type>
- o save ns config

Example

The following example sets the "text/html" content type as the default for the specified profile:

```
set appfw profile profile1 -requestContentType "text/html"
save ns config
```

To remove the user-defined default request content type by using the command line interface

At the command prompt, type the following commands:

- o unset appfw profile <name> -requestContentType <type>
- o save ns config

Example

The following example unsets the default content type of "text/html" for the specified profile, allowing the type to revert to "application/octet-stream":

```
unset appfw profile profile1 -requestContentType "text/html"
save ns config
```

To set the default response content type by using the command line interface

At the command prompt, type the following commands:

- o set appfw profile <name> -responseContentType <type>
- o save ns config

Example

The following example sets the "text/html" content type as the default for the specified profile:

```
set appfw profile profile1 -responseContentType "text/html"  
save ns config
```

To remove the user-defined default response content type by using the command line interface

At the command prompt, type the following commands:

- o unset appfw profile <name> -responseContentType <type>
- o save ns config

Example

The following example unsets the default content type of "text/html" for the specified profile, allowing the type to revert to "application/octet-stream":

```
unset appfw profile profile1 -responseContentType "text/html"  
save ns config
```

To add a content type to the allowed content types list by using the command line interface

At the command prompt, type the following commands:

- o bind appfw profile <name> -ContentType <contentTypeName>
- o save ns config

Example

The following example adds the "text/shtml" content type to the allowed content types list for the specified profile:

```
bind appfw profile profile1 -contentType "text/shtml"  
save ns config
```

To remove a content type from the allowed content types list by using the command line interface

At the command prompt, type the following commands:

- o unbind appfw profile <name> -ContentType <contentTypeName>
- o save ns config

Example

The following example removes the "text/shtml" content type from the allowed content types list for the specified profile:

```
unbind appfw profile profile1 -contentType "text/shtml"  
save ns config
```

To manage the default and allowed content types by using the configuration utility

1. Navigate to Security > Application Firewall > Profiles.
2. In the details pane, select the profile that you want to configure, and then click Edit. The Configure Application Firewall Profile dialog box is displayed.
3. In the Configure Application Firewall Profile dialog box, click the Settings tab
4. On the Settings tab, scroll down about halfway to the Content Type area.
5. In the Content Type area, configure the default request or response content type:
 - o To configure the default request content type, type the MIME/type definition of the content type you want to use in the Default Request text box.

- To configure the default response content type, type the MIME/type definition of the content type you want to use in the Default Response text box.
 - To create a new allowed content type, click Add. The Add Allowed Content Type dialog box is displayed.
 - To edit an existing allowed content type, select that content type, and then click Open. The Modify Allowed Content Type dialog box is displayed.
6. To manage the allowed content types, click Manage Allowed Content Types.
 7. To add a new content type or modify an existing content type, click Add or Open, and in the Add Allowed Content Type or Modify Allowed Content Type dialog box, do the following steps.
 - a. Select/clear the Enabled check box to include the content type in, or exclude it from, the list of allowed content types.
 - b. In the Content Type text box, type a regular expression that describes the content type that you want to add, or change the existing content type regular expression.

Content types are formatted exactly as MIME type descriptions are.

Note: You can include any valid MIME type on the allowed contents type list. Since many types of document can contain active content and therefore could potentially contain malicious content, you should exercise caution when adding MIME types to this list.

- c. In the Comments text box, add an optional comment that describes the reason for adding this particular MIME type to the allowed contents type list.
 - d. Click Create or OK to save your changes.
8. Click Close to close the Manage Allowed Content Types dialog box and return to the Settings tab.
9. Click OK to save your changes.

Profiles

A profile is a collection of security settings that are used to protect specific types of web content or specific parts of your web site. In a profile, you determine how the application firewall applies each of its filters (or checks) to requests to your web sites, and responses from them. The application firewall supports two types of profile: four built-in (default) profiles that do not require further configuration, and user-defined profiles that do require further configuration.

Built-In Profiles

The four application firewall built-in profiles provide simple protection for applications and web sites that either do not require protection, or that should not be directly accessed by users at all. These profile types are:

- **APPFW_BYPASS.** Skips all application firewall filtering and sends the unmodified traffic to the protected application or web site, or to the client.
- **APPFW_RESET.** Resets the connection, requiring that the client re-establish his or her session by visiting a designated start page.
- **APPFW_DROP.** Drops all traffic to or from the protected application or web site, and sends no response of any kind to the client.
- **APPFW_BLOCK.** Blocks traffic to or from the protected application or web site.

You use the built-in profiles exactly as you do user-defined profiles, by configuring a policy that selects the traffic to which you want to apply the profile and then associating the profile with your policy. Since you do not have to configure a built-in policy, it provides a quick way to allow or block specified types of traffic or traffic that is sent to specific applications or web sites.

User-Defined Profiles

User-defined profiles are profiles that are build and configured by users. Unlike the default profiles, you must configure a user-defined profile before it will be of use filtering traffic to and from your protected applications.

There are three types of user-defined profile:

- **HTML.** Protects HTML-based web pages.
- **XML.** Protects XML-based web services and web sites.
- **Web 2.0.** Protects Web 2.0 content that combines HTML and XML content, such as ATOM feeds, blogs, and RSS feeds.

The application firewall has a number of security checks, all of which can be enabled or disabled, and configured in a number of ways in each profile. Each profile also has a number of settings that control how it handles different types of content. Finally, rather than manually configuring all of the security checks, you can enable and configure the learning feature. This feature observes normal traffic to your protected web sites for a period of time, and uses those observations to provide you with a tailored list of recommended exceptions (*relaxations*) to some security checks, and additional rules for other security checks.

During initial configuration, whether by using the Application Firewall Wizard or manually, you normally create one general purpose profile to protect all content on your web sites that is not covered by a more specific profile. After that, you can create as many specific profiles as you want to protect more specialized content.

The Profiles pane consists of a table that contains the following elements:

Name. Displays all the application firewall profiles configured in the appliance.

Bound signature. Displays the signatures object that is bound to the profile in the previous column, if any.

Policies. Displays the application firewall policy that invokes the profile in the leftmost column of that row, if any.

Comments. Displays the comment associated with the profile in the leftmost column of that row, if any.

Profile Type. Displays the type of profile. Types are Built-In, HTML, XML, and Web 2.0.

Above the table is a row of buttons and a drop-down list that allow you to create, configure, delete, and view information about your profiles:

- **Add.** Add a new profile to the list.
- **Edit.** Edit the selected profile.
- **Delete.** Delete the selected profile from the list.

- o **Statistics.** View the statistics for the selected profile.
- o **Action.** Drop-down list that contains additional commands. Currently allows you to import a profile that was exported from another application firewall configuration.

Creating Application Firewall Profiles

You can create an application firewall profile in one of two ways: by using the command line, and by using the configuration utility. Creating a profile by using the command line requires that you specify options on the command line. The process is similar to that of [configuring an existing profile](#), and with a few exceptions the two commands take the same parameters.

Creating a profile by using the configuration utility requires that you specify only two options. You specify basic or advanced *defaults*, the default configuration for the various security checks and settings that are part of a profile, and choose the profile *type* to match the type of content that the profile is intended to protect. You can also, optionally, add a comment. After you create the profile, you must then configure it by selecting it in the data pane, and then clicking Edit.

If you plan to use the learning feature or to enable and configure a large number of advanced protections, you should choose advanced defaults. In particular, if you plan to configure either of the SQL injection checks, either of the cross-site scripting checks, any check that provides protection against Web form attacks, or the cookie consistency check, you should plan to use the learning feature. Unless you include the proper exceptions for your protected Web sites when configuring these checks, they can block legitimate traffic. Anticipating all of the necessary exceptions without creating any that are too broad is difficult. The learning feature makes this task much easier. Otherwise, basic defaults are quick and should provide the protection that your web applications need.

There are three profile types:

- **HTML.** Protects standard HTML-based web sites.
- **XML.** Protects XML-based web services and web sites.
- **Web 2.0 (HTML XML).** Protects sites that contain both HTML and XML elements, such as ATOM feeds, blogs, and RSS feeds.

There are also a few restrictions on the name that you can give to a profile. A profile name cannot be the same as the name assigned to any other profile or action in any feature on the NetScaler appliance. Certain action or profile names are assigned to built-in actions or profiles, and can never be used for user profiles. A complete list of disallowed names can be found in the Application Firewall Profile [Supplemental Information](#). If you attempt to create a profile with a name that has already been used for an action or a profile, an error message is displayed and the profile is not created.

To create an application firewall profile by using the command line interface

At the command prompt, type the following commands:

- add appfw profile <name> [-defaults (**basic** | **advanced**)]
- set appfw profile <name> -type (**HTML** | **XML** | **HTML XML**)
- set appfw profile <name> -comment "<comment>"
- save ns config

Example

The following example adds a profile named `pr-basic`, with basic defaults, and assigns a profile type of `HTML`. This is the appropriate initial configuration for a profile to protect an HTML Web site.

```
add appfw profile pr-basic -defaults basic -comment "Simple profile for web sites."
set appfw profile pr-basic -type HTML
save ns config
```

To create an application firewall profile by using the configuration utility

Creating an application firewall profile requires that you specify only a few configuration details.

1. Navigate to Security > Application Firewall > Profiles.
2. In the details pane, click Add.
3. In the Create Application Firewall Profile dialog box, type a name for your profile.

The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 31 letters, numbers, and the hyphen (-), period (.) pound (#), space (), at (@), equals (=), colon (:), and underscore (_) symbols.

4. Choose the profile type from the drop-down list.
5. Click Create, and then click Close.

Configuring Application Firewall Profiles

To configure a user-defined application firewall profile, first configure the security checks, which are called *deep protections* or *advanced protections* in the application firewall wizard. Certain checks require configuration if you are to use them at all. Others have default configurations that are safe but limited in scope; your web sites might need or benefit from a different configuration that takes advantage of additional features of certain security checks.

After you have configured the security checks, you can also configure a number of other settings that control the behavior, not of a single security check, but the application firewall feature. The default configuration is sufficient to protect most web sites, but you should review them to make sure that they are right for your protected web sites.

For more information about the application firewall security checks, see ["Advanced Protections."](#)

To configure an application firewall profile by using the command line

At the command prompt, type the following commands:

- `set appfw profile <name> <arg1> [<arg2> ...]`

where:

- `<arg1>` = a parameter and any associated options.
- `<arg2>` = a second parameter and any associated options.
- `...` = additional parameters and options.

For descriptions of the parameters to use when configuring specific security checks, see ["Advanced Protections."](#)

- `save ns config`

Example

The following example shows how to enable blocking for the HTML SQL Injection and HTML Cross-Site Scripting checks in a profile named `pr-basic`. This command enables blocking for those actions while making no other changes to the profile.

```
set appfw profile pr-basic -crossSiteScriptingAction block
                        -SQLInjectionAction block
```

To configure an application firewall profile by using the configuration utility

1. Navigate to Security > Application Firewall > Profiles.
2. In the details pane, select the profile that you want to configure, and then click Edit.
3. In the Configure Application Firewall Profile dialog box, on the Security Checks tab, configure the security checks.
 - To enable or disable an action for a check, in the list, select or clear the check box for that action.
 - To configure other parameters for those checks that have them, in the list, click the blue chevron to the far right of that check. In the dialog box that appears, configure the parameters. These vary from check to check.

You can also select a check and, at the bottom of the dialog box, click Open to display the Configure Relaxation dialog box or Configure Rule dialog box for that check. These dialog boxes also vary from check to check. Most of them include a Checks tab and a General tab. If the check supports relaxations or user-defined rules, the Checks tab includes an Add button, which opens yet another dialog box, in which you can specify a relaxation or rule for the check. (A relaxation is a rule for exempting specified traffic from the check.) If relaxations have already been configured, you can select one and click Open to modify it.

- To review learned exceptions or rules for a check, select the check, and then click Learned Violations. In the Manage Learned Rules dialog box, select each learned exception or rule in turn.
 - To edit the exception or rule, and then add it to the list, click Edit & Deploy.
 - To accept the exception or rule without modification, click Deploy.
 - To remove the exception or rule from the list, click Skip.
- To refresh the list of exceptions or rules to be reviewed, click Refresh.
- To open the Learning Visualizer and use it to review learned rules, click Visualizer.
- To review the log entries for connections that matched a check, select the check, and then click Logs. You can use this information to determine which checks are matching attacks, so that you can enable blocking for those checks. You can also use this information to determine which checks are matching

legitimate traffic, so that you can configure an appropriate exemption to allow those legitimate connections. For more information about the logs, see "[Logs, Statistics, and Reports](#)."

- To completely disable a check, in the list, clear all of the check boxes to the right of that check.
4. On the Settings tab, configure the profile settings.
 - To associate the profile with the set of signatures that you previously created and configured, under Common Settings, choose that set of signatures in the Signatures drop-down list.
Note: You may need to use the scroll bar on the right of the dialog box to scroll down to display the Common Settings section.
 - To configure an HTML or XML Error Object, select the object from the appropriate drop-down list.
Note: You must first upload the error object that you want to use in the Imports pane. For more information about importing error objects, see "[Imports](#)."
 - To configure the default XML Content Type, type the content type string directly into the Default Request and Default Response text boxes, or click Manage Allowed Content Types to manage the list of allowed content types. ">>More...."
 5. If you want to use the learning feature, click Learning, and configure the learning settings for the profile, as described in "[Configuring and Using the Learning Feature](#)".
 6. Click OK to save your changes and return to the Profiles pane.

Changing an Application Firewall Profile Type

If you chose the wrong profile type for an application firewall profile, or the type of content on the protected web site has changed, you can change the profile type.

Note: When you change the profile type, you lose all configuration settings and learned relaxations or rules for the features that the new profile type does not support. For example, if you change the profile type from Web 2.0 to XML, you lose any configuration options for Start URL, Form Field Consistency Check, and the other HTML-specific security checks. The configuration for any options that is supported by both the old and the new profile types remains unchanged.

To change an application firewall profile type by using the command line interface

At the command prompt, type the following commands:

- o set appfw profile <name> -type (**HTML** | **XML** | **HTML XML**)
- o save ns config

Example

The following example changes the type of a profile named `pr-basic`, from `HTML` to `HTML XML`, which is equivalent to the Web 2.0 type in the configuration utility.

```
set appfw profile pr-basic -type HTML XML
save ns config
```

To change an application firewall profile type by using the configuration utility.

1. Navigate to Security > Application Firewall > Policies.
2. In the details pane, click Action, and then Change Profile Type.
3. In the Change Application Firewall Profile Type dialog box, Profile Type drop-down list, select a new profile type.
4. Click OK to save your changes and return to the Profiles pane.

Exporting and Importing an Application Firewall Profile

You can export application firewall profiles to your local computer as files, and import previously exported profile files. You might want to configure an application firewall in a test bed configuration and then export the profile or profiles so that you can import the profile configuration to your production NetScaler ADCs. You might also want to export a profile to back up your configuration before making changes so that you can easily roll the configuration back to a known state if necessary.

To export an application firewall profile by using the command line interface

At the command prompt, type the following commands:

- `archive appfw profile <name> <archiveName>`

Make the following substitutions:

`name` = Name of the profile to archive.

`archiveName` = Name of the archive file to create.

- `export appfw archive <archiveName> <target>`

Make the following substitutions:

`archiveName` = Name of the archive to export. (The same name as in the previous step.)

`target` = Full path and filename of the exported file on your local computer. Enclose in straight double quotation marks if the path or filename contains spaces.

Example

Assuming that your local computer uses the Windows 7 operating system, and you are logged onto your local computer as "jsmith", the following example exports a profile named `pr-basic` to the home directory.

```
archive appfw profile pr-basic pr-basic.tgz
export appfw pr-basic.tgz "C:\Users\jsmith\Documents\pr-basic.tgz"
```

To export an application firewall profile by using the configuration utility

1. Navigate to Security > Application Firewall > Profiles.
2. In the details pane, select a profile to export, and then click Actions and select Export.
3. Choose the local path and filename for the exported file.
 - You can accept the default choice, which consists of the path to your home directory or folder and a filename of the profile name plus the extension `.tgz`, which indicates a Unix-style `tar` archive that is compressed by `gzip`.
 - You can type a new path and/or file name. The path must exist, and the filename must be a valid filename in your local computer's operating system. If you do not specify the `.tgz` extension, it is added automatically.
 - You can use the Browse dialog to locate the path and save the file under the default filename. (Recommended)
4. Click Export. The profile is exported and saved to your computer under the path and file name that you designated.

To import an application firewall profile by using the command line interface

At the command prompt, type the following command:

- `import appfw archive <source> <archiveName>`

Make the following substitutions:

- `source` = Full path and filename of the archive file to be imported from your local computer. Enclose in straight double quotation marks if the path or filename contains spaces.
- `archiveName` = Name of the archive file on the NetScaler ADC.

- `restore appfw profile <archiveName>`

Example

Assuming that your local computer uses the Windows 7 operating system, and you are logged onto your local computer as "jsmith", the following example imports a profile named `pr-basic.tgz`, located in your home directory, to the application firewall and installs it as a profile named `pr-basic`.

```
import appfw archive "C:\Users\jsmith\Documents\pr-basic.tgz" pr-basic.tgz  
restore appfw profile pr-basic.tgz
```

Configuring and Using the Learning Feature

The learning feature is a repetitive pattern filter that observes activity on a web site or application protected by the application firewall, to determine what constitutes normal activity on that web site or application. It then generates a list of up to 2,000 suggested rules or exceptions (relaxations) for each security checks that includes support for the learning feature. Users normally find it easier to configure relaxations by using the learning feature than by entering the necessary relaxations manually.

The security checks that support the learning feature are:

- o Start URL check
- o Cookie Consistency check
- o Form Field Consistency check
- o Field Formats check
- o CSRF Form Tagging check
- o HTML SQL Injection check
- o HTML Cross-Site Scripting check
- o XML Denial-of-Service check
- o XML Attachment check
- o Web Services Interoperability check

You perform two different types of activities when using the learning feature. First, you enable and configure the feature to use it. You can use learning on all traffic to your protected web applications, or you can configure a list of IPs (called the *Add Trusted Learning Clients* list) from which the learning feature should generate recommendations. Second, after the feature has been enabled and has processed a certain amount of traffic to your protected web sites, you review the list of suggested rules and relaxations (learned rules) and mark each with one of the following designations:

- o **Edit & Deploy.** The rule is pulled into the Edit dialog box so that you can modify it, and the modified form is deployed.
- o **Deploy.** The unmodified learned rule is placed on the list of rules or relaxations for this security check.
- o **Skip.** The learned rule is placed on a list of rules or relaxations that are not deployed, and that should not be learned again.

Although you can use the command line interface for basic configuration of the learning feature, the feature is designed primarily for configuration through the Application Firewall wizard or the configuration utility. You can perform only limited configuration of the learning feature by using the command line.

The wizard integrates configuration of learning features with configuration of the application firewall as a whole, and is therefore the easiest method for configuring this feature on a new NetScaler appliance or when managing a simple application firewall configuration. The configuration utility visualizer and manual interface both provide direct access to all learned rules for all security checks, and are therefore often preferable when you must review learned rules for a large number of security checks.

The learning database is limited to 20 MB in size, which is reached after approximately 2,000 learned rules or relaxations are generated per security check for which learning is enabled. If you do not regularly review and either approve or ignore learned rules and this limit is reached, an error is logged to the NetScaler log and no more learned rules are generated until you review the existing learned rules and relaxations.

If learning stops because the database has reached its size limit, you can restart learning either by reviewing the existing learned rules and relaxations or by resetting the learning data. After learned rules or relaxations are approved or ignored, they are removed from the database. After you reset the learning data, all existing learning data is removed from the database and it is reset to its minimum size. When the database falls below 20 MB in size, learning restarts automatically.

To configure the learning settings by using the command line interface

Specify the application firewall profile to be configured and, for each security check that you want to include in that profile, specify the minimum threshold or the percent threshold. The minimum threshold is an integer representing the minimum number of user sessions that the application firewall must process before it learns a rule or relaxation (default: 1). The percent threshold is an integer representing the percentage of user sessions in which the application firewall must observe a particular pattern (URL, cookie, field, attachment, or rule violation) before it learns a rule or relaxation (default: 0). Use the following commands:

- o `set appfw learningsettings <profileName> [-startURLMinThreshold <positive_integer>] [-startURLPercentThreshold <positive_integer>] [-cookieConsistencyMinThreshold <positive_integer>] [-cookieConsistencyPercentThreshold <positive_integer>] [-CSRFtagMinThreshold <positive_integer>] [-`


```

CSRFTagPercentThreshold <positive_integer>] [-fieldConsistencyMinThreshold <positive_integer>] [-
fieldConsistencyPercentThreshold <positive_integer>] [-crossSiteScriptingMinThreshold <positive_integer>] [-
crossSiteScriptingPercentThreshold <positive_integer>] [-SQLInjectionMinThreshold <positive_integer>] [-
SQLInjectionPercentThreshold <positive_integer>] [-fieldFormatMinThreshold <positive_integer>] [-
fieldFormatPercentThreshold <positive_integer>] [-XMLWSIMinThreshold <positive_integer>] [-
XMLWSIPercentThreshold <positive_integer>] [-XMLAttachmentMinThreshold <positive_integer>] [-
XMLAttachmentPercentThreshold <positive_integer>]

```

- o save ns config

Example

The following example enables and configures the learning settings in the profile `pr-basic` for the HTML SQL Injection security check. This is an appropriate initial test bed learning configuration, where you have complete control over the traffic that is sent to the application firewall.

```

set appfw learningsettings pr-basic -SQLInjectionMinThreshold 10
set appfw learningsettings pr-basic -SQLInjectionPercentThreshold 70
save ns config

```

To reset learning settings to their defaults by using the command line interface

To remove any custom configuration of the learning settings for the specified profile and security check, and return the learning settings to their defaults, at the command prompt type the following commands:

- o unset appfw learningsettings <profileName> [-startURLMinThreshold] [-startURLPercentThreshold] [-cookieConsistencyMinThreshold] [-cookieConsistencyPercentThreshold] [-CSRFTagMinThreshold] [-CSRFTagPercentThreshold] [-fieldConsistencyMinThreshold] [-fieldConsistencyPercentThreshold] [-crossSiteScriptingMinThreshold] [-crossSiteScriptingPercentThreshold] [-SQLInjectionMinThreshold] [-SQLInjectionPercentThreshold] [-fieldFormatMinThreshold] [-fieldFormatPercentThreshold] [-XMLWSIMinThreshold] [-XMLWSIPercentThreshold] [-XMLAttachmentMinThreshold] [-XMLAttachmentPercentThreshold]
- o save ns config

To display the learning settings for a profile by using the command line interface

At the command prompt, type the following command:

```
show appfw learningsettings <profileName>
```

To display unreviewed learned rules or relaxations for a profile by using the command line interface

At the command prompt, type the following command:

```
show appfw learningdata <profileName> <securityCheck>
```

To remove specific unreviewed learned rules or relaxations from the learning database by using the command line interface

At the command prompt, type the following command:

```

rm appfw learningdata <profileName> (-startURL <expression> | -cookieConsistency <string> | (-fieldConsistency <string>
<formActionURL>) | (-crossSiteScripting <string> <formActionURL>) | (-SQLInjection <string> <formActionURL>) | (-
fieldFormat <string><formActionURL>) | (-CSRFTag <expression> <CSRFFormOriginURL>) | -XMLDoSCheck <expression>
| -XMLWSICheck <expression> | -XMLAttachmentCheck <expression>) [-TotalXMLRequests]

```

Example

The following example removes all unreviewed learned relaxations for the `pr-basic` profile, HTML SQL Injection security check, that apply to the `LastName` form field.

```
rm appfw learningdata pr-basic -SQLInjection LastName
```

To remove all unreviewed learned data by using the command line interface

At the command prompt, type the following command:

```
reset appfw learningdata
```

To export learning data by using the command line interface

At the command prompt, type the following command:

```
export appfw learningdata <profileName> <securitycheck>[-target <string>]
```

Example

The following example exports learned relaxations for the `pr-basic` profile and the HTML SQL Injection security check to a comma-separated values (CSV) format file in the `/var/learn_data/` directory under the filename specified in the `-target` parameter.

```
export appfw learningdata pr-basic SQLInjection -target sqli_ld
```

To configure the Learning feature by using the configuration utility

1. Navigate to Security > Application Firewall > Profiles.
2. In the Profiles pane, select the profile, and then click Edit.
3. Click the Learning tab. At the top of the Learning tab is list of the security checks that are available in the current profile and that support the learning feature.
4. To configure the learning thresholds, select a security check, and then type the appropriate values in the following text boxes:
 - o **Minimum number threshold.** Depending on which security check's learning settings you are configuring, the minimum number threshold might refer to the minimum number of total user sessions that must be observed, the minimum number of requests that must be observed, or the minimum number of times a specific form field must be observed, before a learned relaxation is generated. Default: 1
 - o **Percentage of times threshold.** Depending on which security check's learning settings you are configuring, the percentage of times threshold might refer to the percentage of total observed user sessions that violated the security check, the percentage of requests, or the percentage of times a form field matched a particular field type, before a learned relaxation is generated. Default: 0
5. To remove all learned data and reset the learning feature, so that it must start its observations again from the beginning, click Remove All Learned Data.
Note: This button removes only learned recommendations that have not been reviewed and either approved or skipped. It does not remove learned relaxations that have been accepted and deployed.
6. To restrict the learning engine to traffic from a specific set of IPs, click Trusted Learning Clients, and add the IP addresses that you want to use to the list.
 - a. To add an IP address or IP address range to the Trusted Learning Clients list, click Add.
 - b. In the Add Trusted Learning Clients dialog box, Trusted Clients IP list box, type the IP address or an IP address range in CIDR format.
 - c. In the Comments text area, type a comment that describes this IP address or range.
 - d. Click Create to add your new IP address or range to the list.
 - e. To modify an existing IP address or range, click the IP address or range, and then click Open. Except for the name, the dialog box that appears is identical to the Add Trusted Learning Clients dialog box.
 - f. To disable or enable an IP address or range, but leave it on the list, click the IP address or range, and then click Disable or Enable, as appropriate.
 - g. To remove an IP address or range completely, click the IP address or range, and then click Remove.
7. Click Close to return to the Configure Application Firewall Profile dialog box.
8. Click Close to close the Configure Application Firewall Profile dialog box, and return to the Application Firewall Profile screen.

To review learned rules or relaxations by using the configuration utility

1. Navigate to Security > Application Firewall > Profiles.
2. Select the security check for which you want to review learned rules or relaxations, and then click Manage Rules.
3. In the Manage Learned Rules dialog box, choose how you want to review the learned rules.
 - o To review the actual learned patterns as displayed in the window, do nothing and proceed to the next step.
 - o To review the learned data hierarchically as a branching tree, enabling you to choose general patterns that match many of the learned patterns, click Visualizer.

4. If you have chosen to review actual learned patterns, perform the following steps.
 - a. Select the first learned relaxation and choose how to handle it.
 - To modify and then accept the relaxation, click Edit & Deploy, edit the relaxation regular expression, and then click OK.
 - To accept the relaxation without modifications, click Deploy.
 - To remove the relaxation from the list without deploying it, click Skip.
 - b. Repeat the previous step to review each additional learned relaxation.
5. If you have chosen to use the Learning Visualizer, perform the following steps.
 - a. In the branching hierarchical display, select a node that contains a learned pattern, and choose how to handle it.

The screen area beneath the tree structure, under Regex of Selected Node, displays a generalized expression that matches all of the patterns in that node. If you want to display an expression that matches just one of the branches or just one of the leaves, select that branch or leaf.

 - To modify and then accept the learned relaxation, click Edit & Deploy, edit the relaxation regular expression, and then click OK.
 - To accept the relaxation without modifications, click Deploy.
 - To remove the modification from the list without deploying it, click Skip.
 - b. Repeat the previous step to review other portions of the display.
 - c. Click Close to return to the Manage Learned Rules dialog box.
6. Click Close to return to the Configure Application Firewall Profile dialog box.
7. Click Close to close the Configure Application Firewall Profile dialog box, and return to the Application Firewall Profile screen.

Supplemental Information about Profiles

Following is supplemental information about particular aspects of application firewall profiles. This information explains how to include special characters in a security check rule or relaxation, and how to use variables when configuring profiles.

Configuration Variable Support

Instead of using static values, to configure the application firewall's security checks and settings, you can now use standard NetScaler named variables. By creating variables, you can more easily export and then import configurations to new NetScaler appliances, or update existing NetScaler appliances from a single set of configuration files. This simplifies updates when you use a test bed setup to develop a complex application firewall configuration that is tuned for your local network and servers and then transfer that configuration to your production NetScaler appliances.

You create application firewall configuration variables in the same manner as you do any other NetScaler named variables, following standard NetScaler conventions. To create a named expression variable by using the configuration utility, you use the ["Add Expression dialog box."](#) To create a named expression variable by using the NetScaler command line, you use the `add expression` command followed by the appropriate parameter.

The following URLs and expressions can be configured with variables instead of static values:

- o **Start URL** (-starturl)
- o **Deny URL** (-denyurl)
- o **Form Action URL** for *Form Field Consistency Check* (-fieldconsistency)
- o **Action URL** for *XML SQL Injection Check* (-xmlSQLInjection)
- o **Action URL** for *XML Cross-Site Scripting Check* (-xmlXSS)
- o **Form Action URL** for *HTML SQL Injection Check* (-sqlInjection)
- o **Form Action URL** for *Field Format Check* (-fieldFormat)
- o **Form Origin URL** and **Form Action URL** for *Cross-Site Request Forgery (CSRF) Check* (-csrfTag)
- o **Form Action URL** for *HTML Cross-Site Scripting Check* (-crossSiteScripting)
- o **Safe Object** (-safeObject)
- o **Action URL** for *XML Denial-of-Service (XDoS) check* (-XMLDoS)
- o **URL** for *Web Services Interoperability check* (-XMLWSIURL)
- o **URL** for *XML Validation check* (-XMLValidationURL)
- o **URL** for *XML Attachment check* (-XMLAttachmentURL)

For more information, see ["Policies and Expressions."](#)

To use a variable in the configuration, you enclose the variable name between two at (@) symbols and then use it exactly as you would the static value that it replaces. For example, if you are configuring the Deny URL check by using the configuration utility and want to add the named expression variable `myDenyURL` to the configuration, you would type `@myDenyURL@` into the Add Deny URL dialog box, Deny URL text area. To do the same task by using the NetScaler command line, you would type `add appfw profile <name> -denyURLAction @myDenyURL@`.

PCRE Character Encoding Format

The NetScaler operating system supports direct entry of characters in the printable ASCII character set only—characters with hexadecimal codes between HEX 20 (ASCII 32) and HEX 7E (ASCII 127). To include a character with a code outside that range in your application firewall configuration, you must enter its UTF-8 hexadecimal code as a PCRE regular expression.

A number of character types require encoding using a PCRE regular expression if you include them in your application firewall configuration as a URL, form field name, or Safe Object expression. They include:

- o **Upper-ASCII characters.** Characters with encodings from HEX 7F (ASCII 128) to HEX FF (ASCII 255). Depending on the character map used, these encodings can refer to control codes, ASCII characters with accents or other modifications, non-Latin alphabet characters, and symbols not included in the basic ASCII set. These characters can appear in URLs, form field names, and safe object expressions.
- o **Double-Byte characters.** Characters with encodings that use two 8-byte words. Double-byte characters are used primarily for representing Chinese, Japanese, and Korean text in electronic format. These characters can appear in URLs, form field names, and safe object expressions.

- o **ASCII control characters.** Non-printable characters used to send commands to a printer. All ASCII characters with hexadecimal codes less than HEX 20 (ASCII 32) fall into this category. These characters should never appear in a URL or form field name, however, and would rarely if ever appear in a safe object expression.

The NetScaler appliance does not support the entire UTF-8 character set, but only the characters found in the following eight charsets:

- o **English US (ISO-8859-1).** Although the label reads, "English US," the application firewall supports all characters in the ISO-8859-1 character set, also called the Latin-1 character set. This character set fully represents most modern western European languages and represents all but a few uncommon characters in the rest.
- o **Chinese Traditional (Big5).** The application firewall supports all characters in the BIG5 character set, which includes all of the Traditional Chinese characters (ideographs) commonly used in modern Chinese as spoken and written in Hong Kong, Macau, Taiwan, and by many people of Chinese ethnic heritage who live outside of mainland China.
- o **Chinese Simplified (GB2312).** The application firewall supports all characters in the GB2312 character set, which includes all of the Simplified Chinese characters (ideographs) commonly used in modern Chinese as spoken and written in mainland China.
- o **Japanese (SJIS).** The application firewall supports all characters in the Shift-JIS (SJIS) character set, which includes most characters (ideographs) commonly used in modern Japanese.
- o **Japanese (EUC-JP).** The application firewall supports all characters in the EUC-JP character set, which includes all characters (ideographs) commonly used in modern Japanese.
- o **Korean (EUC-KR).** The application firewall supports all characters in the EUC-KR character set, which includes all characters (ideographs) commonly used in modern Korean.
- o **Turkish (ISO-8859-9).** The application firewall supports all characters in the ISO-8859-9 character set, which includes all letters used in modern Turkish.
- o **Unicode (UTF-8).** The application firewall supports certain additional characters in the UTF-8 character set, including those used in modern Russian.

When configuring the application firewall, you enter all non-ASCII characters as PCRE-format regular expressions using the hexadecimal code assigned to that character in the UTF-8 specification. Symbols and characters within the normal ASCII character set, which are assigned single, two-digit codes in that character set, are assigned the same codes in the UTF-8 character set. For example, the exclamation point (!), which is assigned hex code 21 in the ASCII character set, is also hex 21 in the UTF-8 character set. Symbols and characters from another supported character set have a paired set of hexadecimal codes assigned to them in the UTF-8 character set. For example, the letter a with an acute accent (Ã) is assigned UTF-8 code C3 A1.

The syntax you use to represent these UTF-8 codes in the application firewall configuration is `\xNN` for ASCII characters; `\xNN\xNN` for non-ASCII characters used in English, Russian, and Turkish; and `\xNN\xNN\xNN` for characters used in Chinese, Japanese, and Korean. For example, if you want to represent a ! in an application firewall regular expression as a UTF-8 character, you would type `\x21`. If you want to include an Ã, you would type `\xC3\xA1`.

Note: Normally you do not need to represent ASCII characters in UTF-8 format, but when those characters might confuse a web browser or an underlying operating system, you can use the character's UTF-8 representation to avoid this confusion. For example, if a URL contains a space, you might want to encode the space as `\x20` to avoid confusing certain browsers and web server software.

Below are examples of URLs, form field names, and safe object expressions that contain non-ASCII characters that must be entered as PCRE-format regular expressions to be included in the application firewall configuration. Each example shows the actual URL, field name, or expression string first, followed by a PCRE-format regular expression for it.

- o A URL containing extended ASCII characters.

Actual URL: `http://www.josÃ@nuÃez.com`

Encoded URL: `^http://www[.]jos\xC3\xA9nu\xC3\xB1ez[.]com$`

- o Another URL containing extended ASCII characters.

Actual URL: `http://www.example.de/trÃ¶mso.html`

Encoded URL: `^http://www[.]example[.]de/tr\xC3\xB6mso[.]html$`

- o A form field name containing extended ASCII characters.

Actual Name: `nome_do_usuÃrio`

Encoded Name: `^nome_do_usu\xC3\xA1rio$`

- o A safe object expression containing extended ASCII characters.

Unencoded Expression `[A-Z]{3,6}Â¥[1-9][0-9]{6,6}`

Encoded Expression: `[A-Z]{3,6}\xC2\xA5[1-9][0-9]{6,6}`

You can find a number of tables that include the entire Unicode character set and matching UTF-8 encodings on the Internet. A useful web site that contains this information is located at the following URL:

<http://www.utf8-chartable.de/unicode-utf8-table.pl>

For the characters in the table on this web site to display correctly, you must have an appropriate Unicode font installed on your computer. If you do not, the visual display of the character may be in error. Even if you do not have an appropriate font installed to display a character, however, the description and the UTF-8 and UTF-16 codes on this set of web pages will be correct.

Inverted PCRE Expressions

In addition to matching content that contains a pattern, you can match content that does not contain a pattern by using an inverted PCRE expression. To invert an expression, you simply include an exclamation point (!) followed by white space as the first character in the expression.

Note: If an expression consists only of an exclamation point with nothing following, the exclamation point is treated as a literal character, not syntax indicating an inverted expression.

The following application firewall commands support inverted PCRE expressions:

- o Start URL (URL)
- o Deny URL (URL)
- o Form Field Consistency (form action URL)
- o Cookie Consistency (form action URL)
- o Cross Site Request Forgery (CSRF) (form action URL)
- o HTML Cross-site Scripting (form action URL)
- o Field Format (form action URL)
- o Field Type (type)
- o Confidential Field (URL)

Note: If the security check contains an isRegex flag or check box, it must be set to YES or checked to enable regular expressions in the field. Otherwise the contents of that field are treated as literal and no regular expressions (inverted or not) are parsed.

Disallowed Names for Application Firewall Profiles

The following names are assigned to built-in actions and profiles on the NetScaler appliance, and cannot be used as names for a user-created application firewall profile.

- o AGGRESSIVE
- o ALLOW
- o BASIC
- o CLIENTAUTH
- o COMPRESS
- o CSSMINIFY
- o DEFLATE
- o DENY

- o DNS-NOP
- o DROP
- o GZIP
- o HTMLMINIFY
- o IMGOPTIMIZE
- o JSMINIFY
- o MODERATE
- o NOCLIENTAUTH
- o NOCOMPRESS
- o NONE
- o NOOP
- o NOREWRITE
- o RESET
- o SETASLEARNNSLOG_ACT
- o SETNSLOGPARAMS_ACT
- o SETSYSLOGPARAMS_ACT
- o SETTMSESSPARAMS_ACT
- o SETVPNPARAMS_ACT
- o SET_PREAUTHPARAMS_ACT
- o default_DNS64_action
- o dns_default_act_Cachebypass
- o dns_default_act_Drop
- o nshttp_default_profile
- o nshttp_default_strict_validation
- o nstcp_default_Mobile_profile
- o nstcp_default_XA_XD_profile
- o nstcp_default_profile
- o nstcp_default_tcp_interactive_stream
- o nstcp_default_tcp_lan
- o nstcp_default_tcp_lan_thin_stream
- o nstcp_default_tcp_lfp
- o nstcp_default_tcp_lfp_thin_stream
- o nstcp_default_tcp_lnp
- o nstcp_default_tcp_lnp_thin_stream
- o nstcp_internal_apps

Policy Labels

A policy label consists of a set of policies, other policy labels, and virtual server-specific policy banks. The application firewall evaluates each policy bound to the policy label in order of priority. If the policy matches, it filters the connection as specified in the associated profile. Then it does whatever the Goto parameter specifies, which can be to terminate policy evaluation, go to the next policy, or go to the policy with the specified priority. If the Invoke parameter is set, it terminates processing of the current policy label and begins to process the specified policy label or virtual server.

To create an application firewall policy label by using the command line

At the command prompt, type the following commands:

- o add appfw policylabel <labelName> http_req
- o save ns config

Example

The following example creates a policy label named `policylbl1`.

```
add appfw policylabel policylbl1 http_req
save ns config
```

To bind a policy to a policy label by using the command line

At the command prompt, type the following commands:

- o bind appfw policylabel <labelName> <policyName> <priority> [<gotoPriorityExpression>] [-invoke (<labelType> <labelName>)]
- o save ns config

Example

The following example binds the policy `policy1` to the policy label `policylbl1` with a priority of 1.

```
bind appfw policylabel policylbl1 policy1 1
save ns config
```

To configure an application firewall policy label by using the configuration utility

1. Navigate to Security > Application Firewall > Policy Labels.
2. In the details pane, do one of the following:
 - o To add a new policy label, click Add.
 - o To configure an existing policy label, select the policy label and the click Open.The Create Application Firewall Policy Label or the Configure Application Firewall Policy Label dialog box opens. The dialog boxes are nearly identical.
3. If you are creating a new policy label, in the Create Application Firewall Policy Label dialog box, type a name for your new policy label.

The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.), pound (#), space (), at (@), equals (=), colon (:), and underscore (_) symbols.

4. Select Insert Policy to insert a new row and display a drop-down list with all existing application firewall policies.
5. Select the policy you want to bind to the policy label, or select New Policy to create a new policy and follow the instructions in [To create and configure a policy by using the configuration utility](#). The policy that you selected or created is inserted into the list of globally bound application firewall policies.
6. Make any additional adjustments.
 - o To modify the policy priority, click the field to enable it, and then type a new priority. You can also select Regenerate Priorities to renumber the priorities evenly.
 - o To modify the policy expression, double click that field to open the Configure Application Firewall Policy dialog box, where you can edit the policy expression.
 - o To set the Goto Expression, double click field in the Goto Expression column heading to display the drop-down list, where you can choose an expression.

- To set the Invoke option, double click field in the Invoke column heading to display the drop-down list, where you can choose an expression
7. Repeat steps 5 through 7 to bind any additional application firewall policies you want to the policy label.
 8. Click Create or OK, and then click Close. A message appears in the status bar, stating that you have successfully created or modified the policy label.

Policies

The application firewall uses two types of policies: firewall policies and auditing policies. Firewall policies control which traffic is sent to the application firewall. Auditing policies control the log server to which application firewall logs are sent.

Firewall policies can be complex because the policy rule can consist of multiple expressions in the NetScaler expressions language, which is a full-fledged object oriented programming language capable of defining with extreme precision exactly which connections to filter. Because firewall policies operate within the context of the application firewall, they must meet certain criteria that are connected to how the application firewall functions and what traffic is appropriately filtered by it. As long as you keep these criteria in mind, however, firewall policies are similar to policies for other NetScaler features. The instructions here do not attempt to cover all aspects of writing firewall policies, but only provide an introduction to policies and cover those criteria that are unique to the application firewall.

Auditing policies are simple because the policy rule is always `ns_true`. You need only specify the log server that you want to send logs to, the logging levels that you want to use, and a few other criteria that are explained in detail.

Firewall Policies

A firewall policy is a rule associated with a profile. The rule is an expression or group of expressions that defines the types of request/response pairs that the application firewall is to filter by applying the profile. Firewall policy expressions are written in the NetScaler expressions language, an object-oriented programming language with special features to support specific NetScaler functions. The profile is the set of actions that the application firewall is to use to filter request/response pairs that match the rule.

Firewall policies enable you to assign different filtering rules to different types of web content. Not all web content is alike. A simple web site that uses no complex scripting and accesses and handles no private data might require only the level of protection provided by a profile created with basic defaults. Web content that contains JavaScript-enhanced web forms or accesses an SQL database probably requires more tailored protection. You can create a different profile to filter that content, and create a separate firewall policy that can determine which requests are attempting to access that content. You then associate the policy expression with a profile you created and globally bind the policy to put it into effect.

The application firewall processes only HTTP connections, and therefore uses a subset of the overall NetScaler expressions language. The information here is limited to topics and examples that are likely to be useful when configuring the application firewall. Following are links to additional information and procedures for firewall policies:

- For procedures that explain how to create and configure a policy, see ["Creating and Configuring Application Firewall Policies."](#)
- For a procedure that explains in detail how to create a policy rule (expression), see ["To create or configure an Application Firewall rule \(expression\)."](#)
- For a procedure that explains how to use the Add Expression dialog box to create a policy rule, see ["To add a firewall rule \(expression\) by using the Add Expression dialog box."](#)
- For a procedure that explains how to view the current bindings for a policy, see ["Viewing a Firewall Policy's Bindings."](#)
- For procedures that explain how to bind an application firewall policy, see ["Binding Application Firewall Policies."](#)
- For detailed information about the NetScaler expressions language, see ["Policies and Expressions."](#)

Creating and Configuring Application Firewall Policies

A firewall policy consists of two elements: a *rule*, and an associated *profile*. The rule selects the HTTP traffic that matches the criteria that you set, and sends that traffic to the application firewall for filtering. The profile contains the filtering criteria that the application firewall uses.

The policy rule consists of one or more expressions in the NetScaler expressions language. The NetScaler expressions syntax is a powerful, object-oriented programming language that enables you to precisely designate the traffic that you want to process with a specific profile. For users who are not completely familiar with the NetScaler expressions language syntax, or who prefer to configure their NetScaler appliance by using a web-based interface, the configuration utility provides two tools: the Prefix menu and the Add Expression dialog box. Both help you to write expressions that select exactly the traffic that you want to process. Experienced users who are thoroughly familiar with the syntax may prefer to use the NetScaler command line to configure their NetScaler appliances.

Note: In addition to the default expressions syntax, for backward compatibility the NetScaler operating system supports the NetScaler classic expressions syntax on NetScaler Classic and nCore appliances and virtual appliances. Classic expressions are not supported on NetScaler Cluster appliances and virtual appliances. Current NetScaler users who want to migrate existing configurations to the NetScaler Cluster must migrate any policies that contain classic expressions to the default expressions syntax.

For detailed information about the NetScaler expressions languages, see ["Policies and Expressions."](#)

You can create a firewall policy by using the configuration utility or the NetScaler command line.

To create and configure a policy by using the command line interface

At the command prompt, type the following commands:

- add appfw policy <name> <rule> <profileName>
- save ns config

Example

The following example adds a policy named `pl-blog`, with a rule that intercepts all traffic to or from the host `blog.example.com`, and associates that policy with the profile `pr-blog`. This is an appropriate policy to protect a blog hosted on a specific hostname.

```
add appfw policy pl-blog "HTTP.REQ.HOSTNAME.DOMAIN.EQ("blog.example.com")" pr-blog
```

To create and configure a policy by using the configuration utility

1. Navigate to Security > Application Firewall > Policies.
2. In the details pane, do one of the following:
 - To create a new firewall policy, click Add. The Create Application Firewall Policy is displayed.
 - To edit an existing firewall policy, select the policy, and then click Edit.The Create Application Firewall Policy or Configure Application Firewall Policy is displayed.
3. If you are creating a new firewall policy, in the Create Application Firewall Policy dialog box, Policy Name text box, type a name for your new policy.

The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 128 letters, numbers, and the hyphen (-), period (.), pound (#), space (), at (@), equals (=), colon (:), and underscore (_) symbols.

If you are configuring an existing firewall policy, this field is read-only. You cannot modify it.

4. Select the profile that you want to associate with this policy from the Profile drop-down list. You can create a new profile to associate with your policy by clicking New, and you can modify an existing profile by clicking Modify.
5. In the Expression text area, create a rule for your policy.
 - You can type a rule directly into the text area.
 - You can click Prefix to select the first term for your rule, and follow the prompts. See ["To Create an Application Firewall Rule \(Expression\)"](#) for a complete description of this process.
 - You can click Add to open the Add Expression dialog box, and use it to construct the rule. See ["The Add Expression Dialog Box"](#) for a complete description of this process.
6. Click Create or OK, and then click Close.

To create or configure an Application Firewall rule (expression)

The policy rule, also called the *expression*, defines the web traffic that the application firewall filters by using the profile associated with the policy. Like other NetScaler policy rules (or *expressions*), application firewall rules use NetScaler expressions syntax. This syntax is powerful, flexible, and extensible. It is too complex to describe completely in this set of instructions. You can use the following procedure to create a simple firewall policy rule, or you can read it as an overview of the policy creation process.

1. If you have not already done so, navigate to the appropriate location in the Application Firewall wizard or the NetScaler configuration utility to create your policy rule:
 - o If you are configuring a policy in the Application Firewall wizard, in the navigation pane, click Application Firewall, then in the details pane click Application Firewall Wizard, and then navigate to the Specify Rule screen.
 - o If you are configuring a policy manually, in the navigation pane, expand Application Firewall, then Policies, and then Firewall. In the details pane, to create a new policy, click Add. To modify an existing policy, select the policy, and then click Open.
2. On the Specify Rule screen, the Create Application Firewall Profile dialog box, or the Configure Application Firewall Profile dialog box, click Prefix, and then choose the prefix for your expression from the drop-down list. Your choices are:
 - o **HTTP**. The HTTP protocol. Choose this if you want to examine some aspect of the request that pertains to the HTTP protocol.
 - o **SYS**. The protected Web site(s). Choose this if you want to examine some aspect of the request that pertains to the recipient of the request.
 - o **CLIENT**. The computer that sent the request. Choose this if you want to examine some aspect of the sender of the request.
 - o **SERVER**. The computer to which the request was sent. Choose this if you want to examine some aspect of the recipient of the request.

After you choose a prefix, the application firewall displays a two-part prompt window that displays the possible next choices at the top, and a brief explanation of what the selected choice means at the bottom.

3. Choose your next term.

If you chose HTTP as your prefix, your only choice is REQ, which specifies the Request/Response pair. (The application firewall operates on the request and response as a unit instead of on each separately.) If you chose another prefix, your choices are more varied. For help on a specific choice, click that choice once to display information about it in the lower prompt window.

When you have decided which term you want, double-click it to insert it into the Expression window.

4. Type a period after the term you just chose. You are then prompted to choose your next term, as described in the previous step. When a term requires that you type a value, fill in the appropriate value. For example, if you choose `HTTP.REQ.HEADER(" ")`, type the header name between the quotation marks.
5. Continue choosing terms from the prompts and filling in any values that are needed, until your expression is finished.

Following are some examples of expressions for specific purposes.

- o **Specific web host.** To match traffic from a particular web host:

```
HTTP.REQ.HEADER( "Host" ).EQ( "shopping.example.com" )
```

For `shopping.example.com`, substitute the name of the web host that you want to match.

- o **Specific web folder or directory.** To match traffic from a particular folder or directory on a Web host:

```
HTTP.REQ.URL.STARTSWITH( "https://www.example.com/folder" )
```

For `www.example.com`, substitute the name of the web host. For `folder`, substitute the folder or path to the content that you want to match. For example, if your shopping cart is in a folder called `/solutions/orders`, you substitute that string for `folder`.

- o **Specific type of content: GIF images.** To match GIF format images:

```
HTTP.REQ.URL.ENDSWITH( ".gif" )
```

To match other format images, substitute another string in place of `.gif`.

- o **Specific type of content: scripts.** To match all CGI scripts located in the `CGI-BIN` directory:

```
HTTP.REQ.URL.STARTSWITH( "https://www.example.com/CGI-BIN" )
```

To match all JavaScripts with .js extensions:

```
HTTP.REQ.URL.ENDSWITH( ".js" )
```

For more information about creating policy expressions, see ["Policies and Expressions."](#)

Note: If you use the command line to configure a policy, remember to escape any double quotation marks within NetScaler expressions. For example, the following expression is correct if entered in the configuration utility:

```
HTTP.REQ.HEADER( "Host" ).EQ( "shopping.example.com" )
```

If entered at the command line, however, you must type this instead:

```
HTTP.REQ.HEADER( \"Host\" ).EQ( \"shopping.example.com\" )
```

To add a firewall rule (expression) by using the Add Expression dialog box

The Add Expression dialog box (also referred to as the Expression Editor) helps users who are not familiar with the NetScaler expressions language to construct a policy that matches the traffic that they want to filter.

1. If you have not already done so, navigate to the appropriate location in the Application Firewall wizard or the NetScaler configuration utility:
 - o If you are configuring a policy in the Application Firewall wizard, in the navigation pane, click Application Firewall, then in the details pane click Application Firewall Wizard, and then navigate to the Specify Rule screen.
 - o If you are configuring a policy manually, in the navigation pane, expand Application Firewall, then Policies, and then Firewall. In the details pane, to create a new policy, click Add. To modify an existing policy, select the policy, and then click Open.
2. On the Specify Rule screen, in the Create Application Firewall Profile dialog box, or in the Configure Application Firewall Profile dialog box, click Add.
3. In the Add Expression dialog box, in the Construct Expression area, in the first list box, choose one of the following prefixes:
 - o **HTTP.** The HTTP protocol. Choose this if you want to examine some aspect of the request that pertains to the HTTP protocol. The default choice.
 - o **SYS.** The protected Web site(s). Choose this if you want to examine some aspect of the request that pertains to the recipient of the request.
 - o **CLIENT.** The computer that sent the request. Choose this if you want to examine some aspect of the sender of the request.
 - o **SERVER.** The computer to which the request was sent. Choose this if you want to examine some aspect of the recipient of the request.
4. In the second list box, choose your next term. The available terms differ depending on the choice you made in the previous step, because the dialog box automatically adjusts the list to contain only those terms that are valid for the context. For example, if you selected **HTTP** in the previous list box, the only choice is **REQ**, for requests. Because the application firewall treats requests and associated responses as a single unit and filters both, you do not need to specify responses separately. After you choose your second term, a third list box appears to the right of the second. The Help window displays a description of the second term, and the Preview Expression window displays your expression.
5. In the third list box, choose the next term. A new list box appears to the right, and the Help window changes to display a description of the new term. The Preview Expression window updates to display the expression as you have specified it to that point.
6. Continue choosing terms, and when prompted filling in arguments, until your expression is complete. If you make a mistake or want to change your expression after you have already selected a term, you can simply choose another term. The expression is modified, and any arguments or additional terms that you added after the term that you modified are cleared.
7. When you have finished constructing your expression, click OK to close the Add Expression dialog box. Your expression is inserted into the Expression text area.

Binding Application Firewall Policies

After you have configured your application firewall policies, you bind them to Global or a bind point to put them into effect. After binding, any request or response that matches an application firewall policy is transformed by the profile associated with that policy.

When you bind a policy, you assign a priority to it. The priority determines the order in which the policies you define are evaluated. You can set the priority to any positive integer. In the NetScaler OS, policy priorities work in reverse order - the higher the number, the lower the priority.

Because the application firewall feature implements only the first policy that a request matches, not any additional policies that it might also match, policy priority is important for achieving the results that you intend. If you give your first policy a low priority (such as 1000), you configure the application firewall to perform it only if other policies with a higher priority do not match a request. If you give your first policy a high priority (such as 1), you configure the application firewall to perform it first, and skip any other policies that might also match. You can leave yourself plenty of room to add other policies in any order, without having to reassign priorities, by setting priorities with intervals of 50 or 100 between each policy when you bind your policies.

For more information about binding policies on the NetScaler appliance, see ["Policies and Expressions."](#)

To bind an application firewall policy by using the command line interface

At the command prompt, type the following commands:

- `bind appfw global <policyName> <priority>`
- `save ns config`

Example

The following example binds the policy named `pl-blog` and assigns it a priority of 10.

```
bind appfw global pl-blog 10
save ns config
```

To bind an application firewall policy by using the configuration utility

1. Do one of the following:
 - Navigate to Security > Application Firewall, and in the details pane, click Application Firewall policy manager.
 - Navigate to Security > Application Firewall > Policies > Firewall Policies, and in the details pane, click Policy Manager.
2. In the Application Firewall Policy Manager dialog, choose the bind point to which you want to bind the policy from the drop-down list. The choices are:
 - **Override Global.** Policies that are bound to this bind point process all traffic from all interfaces on the NetScaler appliance, and are applied before any other policies.
 - **LB Virtual Server.** Policies that are bound to a load balancing virtual server are applied only to traffic that is processed by that load balancing virtual server, and are applied before any Default Global policies. After selecting LB Virtual Server, you must also select the specific load balancing virtual server to which you want to bind this policy.
 - **CS Virtual Server.** Policies that are bound to a content switching virtual server are applied only to traffic that is processed by that content switching virtual server, and are applied before any Default Global policies. After selecting CS Virtual Server, you must also select the specific content switching virtual server to which you want to bind this policy.
 - **Default Global.** Policies that are bound to this bind point process all traffic from all interfaces on the NetScaler appliance.
 - **Policy Label.** Policies that are bound to a policy label process traffic that the policy label routes to them. The policy label controls the order in which policies are applied to this traffic.
 - **None.** Do not bind the policy to any bind point.
3. Click Continue. A list of existing application firewall policies appears.
4. Select the policy you want to bind by clicking it.
5. Make any additional adjustments to the binding.
 - To modify the policy priority, click the field to enable it, and then type a new priority. You can also select Regenerate Priorities to renumber the priorities evenly.
 - To modify the policy expression, double click that field to open the Configure Application Firewall Policy dialog box, where you can edit the policy expression.

- To set the Goto Expression, double click field in the Goto Expression column heading to display the drop-down list, where you can choose an expression.
 - To set the Invoke option, double click field in the Invoke column heading to display the drop-down list, where you can choose an expression
6. Repeat steps 3 through 6 to add any additional application firewall policies you want to globally bind.
 7. Click OK. A message appears in the status bar, stating that the policy has been successfully bound.

Viewing a Firewall Policy's Bindings

You can quickly check to determine what bindings are in place for any firewall policy by viewing the bindings in the configuration utility.

To view bindings for an application firewall policy

1. Navigate to Security > Application Firewall > Policies > Firewall Policies
2. In the details pane, select the policy that you want to check, and then click Show Bindings. The Binding Details for Policy: Policy message box is displayed, with a list of bindings for the selected policy.
3. Click Close.

Supplemental Information about Application Firewall Policies

Following is supplemental information about particular aspects of application firewall policies that system administrators who manage the application firewall might need to know.

Correct but Unexpected Behavior

Web application security and modern web sites are complex. In a number of scenarios, a NetScaler policy might cause the application firewall to behave differently in certain situations than a user who is familiar with policies would normally expect. Following are a number of cases where the application firewall may behave in an unexpected fashion.

- o **Request with a missing HTTP Host header and an absolute URL.** When a user sends a request, in the majority of cases the request URL is relative. That is, it takes as its starting point the Referer URL, the URL where the user's browser is located when it sends the request. If a request is sent without a Host header, and with a relative URL, the request is normally blocked both because it violates the HTTP specification and because a request that fails to specify the host could under some circumstances constitute an attack. If a request is sent with an absolute URL, however, even if the Host header is missing, the request bypasses the application firewall and is forwarded to the web server. Although such a request violates the HTTP specification, it poses no possible threat because an absolute URL contains the host.

Auditing Policies

Auditing policies determine the messages that are generated and logged during an Application Firewall session. These messages are logged in SYSLOG format to the local NSLOG server or to an external logging server. Different types of messages are logged on the basis of the level of logging selected.

To create an auditing policy, you must first create either an NSLOG server or a SYSLOG server. After specifying the server, you create the policy and specify the type of log and the server to which logs are sent.

To create an auditing server by using the command line interface

You can create two different types of auditing server: an NSLOG server or a SYSLOG server. The command names are different, but the parameters for the commands are the same.

To create an auditing server, at the NetScaler command prompt, type the following commands:

- o add audit syslogAction <name> <serverIP> [-serverPort <port>] -logLevel <logLevel> ... [-dateFormat (**MMDDYYYY** | **DDMMYYYY**)] [-logFacility <logFacility>] [-tcp (**NONE** | **ALL**)] [-acl (**ENABLED** | **DISABLED**)] [-timeZone (**GMT_TIME** | **LOCAL_TIME**)] [-userDefinedAuditlog (**YES** | **NO**)] [-appflowExport (**ENABLED** | **DISABLED**)]
- o save ns config

Example

The following example creates a syslog server named syslog1 at IP 10.124.67.91, with loglevels of emergency, critical, and warning, log facility set to LOCAL1, that logs all TCP connections:

```
add audit syslogAction syslog1 10.124.67.91 -logLevel emergency critical warning -logFacility LOCAL1 -tcp ALL
save ns config
```

To modify or remove an auditing server by using the command line interface

- o To modify an auditing server, type the set audit <type> command, the name of the auditing server, and the parameters to be changed, with their new values.
- o To remove an auditing server, type the rm audit <type> command and the name of the auditing server.

Example

The following example modifies the syslog server named syslog1 to add errors and alerts to the log level:

```
set audit syslogAction syslog1 10.124.67.91 -logLevel emergency critical warning alert error -logFacility LOCAL1 -tcp ALL
save ns config
```

To create or configure an auditing server by using the configuration utility

1. Navigate to Security > Application Firewall > Policies > Auditing.
2. In the details pane, click the Server tab.
3. Do one of the following:
 - o To add a new auditing server, click Add.
 - o To modify an existing auditing server, select the server, and then click Edit.
4. In the Create Auditing Server or Configure Auditing Server dialog box, set the following parameters:
 - o Name
 - o Auditing Type
 - o IP Address
 - o Port
 - o Log Levels
 - o Log Facility
 - o TCP Logging
 - o ACL Logging
 - o User-Configurable Log Messages
 - o AppFlow Logging
 - o Date Format

- o Time Zone
5. Click Create or OK.

To create an auditing policy by using the command line interface

You can create an NSLOG policy or a SYSLOG policy. The type of policy must match the type of server. The command names for the two types of policy are different, but the parameters for the commands are the same.

At the command prompt, type the following commands:

- o add audit syslogPolicy <name> [-rule <expression>] [-action <string>]
- o save ns config

Example

The following example creates a policy named syslogP1 that logs application firewall traffic to a syslog server named syslog1.

```
add audit syslogPolicy syslogP1 -rule "ns_true" -action syslog1
save ns config
```

To configure an auditing policy by using the command line interface

At the command prompt, type the following commands:

- o set audit syslogPolicy <name> [-rule <expression>] [-action <string>]
- o save ns config

Example

The following example modifies the policy named syslogP1 to log application firewall traffic to a syslog server named syslog2.

```
set audit syslogPolicy syslogP1 -rule "ns_true" -action syslog2
save ns config
```

To configure an auditing policy by using the configuration utility

1. Navigate to Security > Application Firewall > Policies > Auditing.
2. In the details pane, do one of the following:
 - o To add a new policy, click Add.
 - o To modify an existing policy, select the policy, and then click Edit.
3. In the Create Auditing Policy or Configure Auditing Policy dialog box, set the following parameters:
 - o Name
 - o Auditing Type
 - o Server
4. Click Create or OK.

Imports

Several application firewall features make use of external files that you upload to the application firewall when configuring it. Using the configuration utility, you manage those files in the Imports pane, which has four tabs corresponding to the four types of files you can import: HTML error objects, XML error objects, XML schemas, and Web Services Description Language (WSDL) files. Using the NetScaler command line, you can import these types of files, but you cannot export them.

HTML Error Object

When a user's connection to an HTML or Web 2.0 page is blocked, or a user asks for a non-existent HTML or Web 2.0 page, the application firewall sends an HTML-based error response to the user's browser. When configuring which error response the application firewall should use, you have two choices:

- You can configure a redirect URL, which can be hosted on any Web server to which users also have access. For example, if you have a custom error page on your Web server, `404.html`, you can configure the application firewall to redirect users to that page when a connection is blocked.
- You can configure an HTML error object, which is an HTML-based Web page that is hosted on the application firewall itself. If you choose this option, you must upload the HTML error object to the application firewall. You do that in the Imports pane, on the HTML Error Object tab.

The error object must be a standard HTML file that contains no non-HTML syntax except for application firewall error object customization variables. It cannot contain any CGI scripts, server-parsed code, or PHP code. The customization variables enable you to embed troubleshooting information in the error object that the user receives when a request is blocked. While most requests that the application firewall blocks are illegitimate, even a properly configured application firewall can occasionally block legitimate requests, especially when you first deploy it or after you make significant changes to your protected Web sites. By embedding information in the error page, you provide the user with the information that he or she needs to give to the technical support person so that any issues can be fixed.

The application firewall error page customization variables are:

- `${NS_TRANSACTION_ID}`. The transaction ID that the application firewall assigned to this transaction.
- `${NS_APPFW_SESSION_ID}`. The application firewall session ID.
- `${NS_APPFW_VIOLATION_CATEGORY}`. The specific application firewall security check or rule that was violated.
- `${NS_APPFW_VIOLATION_LOG}`. The detailed error message associated with the violation.
- `${COOKIE("<CookieName>")}`. The contents of the specified cookie. For `<CookieName>`, substitute the name of the specific cookie that you want to display on the error page. If you have multiple cookies whose contents you want to display for troubleshooting, you can use multiple instances of this customization variable, each with the appropriate cookie name.

Note: If you have blocking enabled for the Cookie Consistency Check, any blocked cookies are not displayed on the error page because the application firewall blocks them.

To use these variables, you embed them in the HTML or XML of the error page object as if they were an ordinary text string. When the error object is displayed to the user, for each customization variable the application firewall substitutes the information to which the variable refers. An example HTML error page that uses custom variables is shown below.

```
<!doctype html public "-//w3c//dtd html 4.0//en"> <html> <head> <title>Page Not Accessibl
```

To use this error page, copy it into a text or HTML editor. Substitute the appropriate local information for the following variables, which are enclosed in square brackets to distinguish them from the NetScaler variables. (Leave those unchanged.):

- **[homePage]**. The URL for your web site's home page.
- **[helpDeskEmailAddress]**. The email address that you want users to use to report blocking incidents.
- **[helpDeskPhoneNumber]**. The phone number that you want users to call to report blocking incidents.
- **[cookieName]**. The name of the cookie whose contents you want to display on the error page.

XML Error Object

When a user's connection to an XML page is blocked, or a user asks for a nonexistent XML application, the application firewall sends an XML-based error response to the user's browser. You configure the error response by uploading an XML-based error page to the application firewall in the Imports Pane, on the XML Error Object tab. All XML error responses are hosted on the application firewall. You cannot configure a redirect URL for XML applications.

Note: You can use the same customization variables in an XML error object as in an HTML error object.

XML Schema

When the application firewall performs a validation check on a user's request for an XML or Web 2.0 application, it can validate the request against the XML schema or design type document (DTD) for that application and reject any request that does not follow the schema or DTD. Both an XML schema and a DTD are standard XML configuration files that describe the structure of a specific type of XML document.

WSDL

When the application firewall performs a validation check on a user's request for an XML SOAP-based web service, it can validate the request against the web services type definition (WSDL) file for that web service. A WSDL file is a standard XML SOAP configuration file that defines the elements of a specific XML SOAP web service.

Importing and Exporting Files

You can import HTML or XML error objects, XML schemas, DTDs, and WSDLs to the application firewall by using the configuration utility or the command line. You can edit any of these files in a web-based text area after importing them, to make small changes directly on the NetScaler ADC instead of having to make them on your computer and then reimport them. Finally, you can export any of these files to your computer, or delete any of these files, by using the configuration utility.

Note: You cannot delete or export an imported file by using the command line.

To import a file by using the command line interface

At the command prompt, type the following commands:

- o `import appfw htmlerrorpage <src> <name>`
- o `save ns config`

Example

The following example imports an HTML error object from a file named `error.html` and assigns it the name `HTMLError`.

```
import htmlerrorpage error.html HTMLError
save ns config
```

To import a file by using the configuration utility

Before you attempt to import an XML schema, DTD, or WSDL file, or an HTML or XML error object from a network location, verify that the NetScaler ADC can connect to the Internet or LAN computer where the file is located. Otherwise, you cannot import the file or object.

1. Navigate to Security > Application Firewall > Imports.
2. Navigate to Application Firewall > Imports.
3. In the Application Firewall Imports pane, select the tab for the type of file you want to import, and then click Add.

The tabs are HTML Error Page, XML Error Page, XML Schema or WSDL. The upload process is identical on all four tabs from the user point of view.

4. Fill in the dialog fields.
 - o **Name**—A name for the imported object.
 - o **Import From**—Choose the location of the HTML file, XML file, XML schema or WSDL that you want to import in the drop-down list:
 - **URL**: A web URL on a website accessible to the ADC.
 - **File**: A file on a local or networked hard disk or other storage device.
 - **Text**: Type or paste the text of the custom response directly into a text field in the configuration utility.

The third text box changes to the appropriate value. The three possible values are provided below.

- o **URL**—Type the URL into the text box.
 - o **File**—Type the path and filename to the HTML file directly, or click Browse and browse to the HTML file.
 - o **Text**—The third field is removed, leaving a blank space.
5. Click Continue. The File Contents dialog is displayed. If you chose URL or File, the File Contents text box contains the HTML file that you specified. If you chose Text, the File Contents text box is empty.
 6. If you chose Text, type or copy and paste the custom response HTML that you want to import.
 7. Click Done.
 8. To delete an object, select the object, and then click Delete.

To export a file by using the configuration utility

Before you attempt to export an XML schema, DTD, or WSDL file, or an HTML or XML error object, verify that the application firewall appliance can access the computer where the file is to be saved. Otherwise, you cannot export the file.

1. Navigate to Security > Application Firewall > Imports.

2. In the Application Firewall Imports pane, select the tab for the type of file you want to export.

The export process is identical on all four tabs from the user point of view.

3. Select the file that you want to export.
4. Expand the Action drop-down list, and select Export.
5. In the dialog box, choose Save File and click OK.
6. In the Browse dialog box, navigate to the local file system and directory where you want to save the exported file, and click Save.

To edit an HTML or XML Error Object in the configuration utility

You edit the text of HTML and XML error objects in the configuration utility without exporting and then reimporting them.

1. Navigate to Security > Application Firewall > Imports, and then select the tab for the type of file that you want to modify.
2. Navigate to Application Firewall > Imports, and then select the tab for the type of file that you want to modify.
3. Select the file that you want to modify, and then click Edit.

The text of the HTML or XML error object is displayed in a browser text area. You can modify the text by using the standard browser-based editing tools and methods for your browser.

Note: The edit window is designed to allow you to make minor changes to your HTML or XML error object. To make extensive changes, you may prefer to export the error object to your local computer and use standard HTML or XML web page editing tools.

4. Click OK, and then click Close.

Global Configuration

The application firewall global configuration affects all profiles and policies. The Global Configuration items are:

- **Engine Settings.** A collection of global settingsâ€”session cookie name, session time-out, maximum session lifetime, logging header name, undefined profile, default profile, and import size limitâ€”that pertain to all connections that the application firewall processes, rather than to a specific subset of connections.
- **Confidential Fields.** A set of form fields in web forms that contain sensitive information that should not be logged to the application firewall logs. Form fields such as password fields on a logon page or credit card information on a shopping cart checkout form are normally designated as confidential fields.
- **Field Types.** The list of web form field types used by the Field Formats security check. Each of these field types is defined by a PCRE-compliant regular expression that defines the type of data and the minimum/maximum length of data that should be allowed in that type of form field.
- **XML Content Types.** The list of content types recognized as XML and subjected to XML-specific security checks. Each of these content types is defined by a PCRE-compliant regular expression that defines the exact MIME type assigned to that content.
- **JSON Content Types.** The list of content types recognized as JSON and subjected to JSON-specific security checks. Each of these content types is defined by a PCRE-compliant regular expression that defines the exact MIME type assigned to that content.

Engine Settings

The engine settings affect all requests and responses that the application firewall processes. They include the following items:

- **Cookie name**—The name of the cookie that stores the NetScaler session ID.
- **Session timeout**—The maximum inactive period allowed. If a user session shows no activity for this length of time, the session is terminated and the user is required to reestablish it by visiting a designated start page.
- **Cookie post-encrypt prefix**—The string that precedes the encrypted portion of any encrypted cookies.
- **Maximum session lifetime**—The maximum amount of time, in seconds, that a session is allowed to remain live. After this period is reached, the session is terminated and the user is required to reestablish it by visiting a designated start page. This setting cannot be less than the session timeout. To disable this setting, so that there is no maximum session lifetime, set the value to zero (0).
- **Logging header name**—The name of the HTTP header that holds the Client IP, for logging.
- **Undefined profile**—The profile applied when the corresponding policy action evaluates as undefined.
- **Default profile**—The profile applied to connections that do not match a policy.
- **Import size limit**—The maximum cumulative total byte count of all files imported to the ADC, including signatures, WSDLs, schemas, HTML and XML error pages. During an import, if the size of the imported object would cause the cumulative total sizes of all imported files to exceed the configured limit, the import operation fails and the ADC displays the following error message: *ERROR: Import failed - exceeding the configured total size limit on the imported objects.*
- **Learn message rate limit**—The maximum number of requests and responses per second that the learning engine is to process. Any additional requests or responses over this limit are not sent to the learning engine.
- **Log malformed request**—Enable logging of malformed HTTP requests.
- **Use configurable secret key**—Use a configurable secret key for application firewall operations.
- **Reset learned data**—Remove all learned data from the application firewall. Restarts the learning process by collecting fresh data.

Two settings, *Reset Learned Data* and *Signatures Auto-Update*, are found in different places depending on whether you use the command line or the configuration utility to configure your application firewall. When using the command line, you configure Reset Learned Data by using the `reset appfw learningdata` command, which takes no parameters and has no other functions. You configure Signatures Auto-Update in the `set appfw settings` command: the `-signatureAutoUpdate` parameter enables or disables auto-updating of the signatures, and `-signatureUrl` configures the URL which hosts the updated signatures file.

When using the configuration utility, you configure Reset Learned Data in Security > Application Firewall > Engine Settings; the Reset Learned Data button is at the bottom of the dialog box. You configure Signatures Auto-Update for each set of signatures in Security > Application Firewall > Signatures, by selecting the signatures file, clicking the right mouse button and selecting Auto Update Settings.

Normally, the default values for the application firewall settings are correct. If the default settings cause a conflict with other servers or cause premature disconnection of your users, however, you might need to modify them.

To configure engine settings by using the command line interface

At the command prompt, type the following commands:

- `set appfw settings [-sessionCookieName <name>] [-sessionTimeout <positiveInteger>] [-sessionLifetime <positiveInteger>] [-clientIPLoggingHeader <headerName>] [-undefaction <profileName>] [-defaultProfile <profileName>] [-importSizeLimit <positiveInteger>] [-logMalformedReq (ON | OFF)] [-signatureAutoUpdate (ON | OFF)] [-signatureUrl <expression>] [-cookiePostEncryptPrefix <string>] [-useConfigurableSecretKey (ON | OFF)] [-learnRateLimit <positiveInteger>]`
- `save ns config`

Example

```
set appfw settings -sessionCookieName citrix-appfw-id -sessionTimeout 3600
-sessionLifetime 14400 -clientIPLoggingHeader NS-AppFW-Client-IP -undefaction APPFW_RESET
-defaultProfile APPFW_RESET -importSizeLimit 4096
save ns config
```

To configure engine settings by using the configuration utility

1. Navigate to Security > Application Firewall
2. In the details pane, click Change Engine Settings.
3. In the Application Firewall Engine Settings dialog box, set the following parameters:
 - Cookie Name
 - Session Timeout
 - Cookie Post Encrypt Prefix
 - Maximum Session Lifetime
 - Logging Header Name
 - Undefined Profile
 - Default Profile
 - Import Size Limit
 - Learn Messages Rate Limit
 - Entity Decoding
 - Log Malformed Request
 - Use Secret Key
 - Learn Message Rate Limit
 - Signatures Auto Update
4. Click OK.

Confidential Fields

You can designate web-form fields as confidential to protect the information users type into them. Normally, any information a user types into a web form on one of your protected web servers is logged in the NetScaler logs. The information typed into a web-form field designated as confidential, however, is not logged. That information is saved only where the web site is configured to save such data, normally in a secure database.

Common types of information that you may want to protect with a confidential field designation include:

- Passwords
- Credit card numbers, validation codes, and expiration dates
- Social security numbers
- Tax ID numbers
- Home addresses
- Private telephone numbers

In addition to being good practice, proper use of confidential field designations may be necessary for PCI-DSS compliance on ecommerce servers, HIPAA compliance on servers that manage medical information in the United States, and compliance with other data protection standards.

Important: In the following two cases, the Confidential Field designation does not function as expected:

- If a Web form has either a confidential field or an action URL longer than 256 characters, the field or action URL is truncated in the NetScaler logs.
- With certain SSL transactions, the logs are truncated if either the confidential field or the action URL is longer than 127 characters.

In either of these cases, the application firewall masks a fifteen-character string with the letter "x," instead of the normal eight character string. To ensure that any confidential information is removed, the user must use form field name and action URL expressions that match the first 256, or (in cases where SSL is used) the first 127 characters.

To configure your application firewall to treat a web-form field on a protected web site as confidential, you add that field to the Confidential Fields list. You can enter the field name as a string, or you can enter a PCRE-compatible regular expression specifying one or more fields. You can enable the confidential-field designation when you add the field, or you can modify the designation later.

To add a confidential field by using the command line interface

At the command prompt, type the following commands:

- `add appfw confidField <fieldName> <url> [-isRegex (REGEX | NOTREGEX)] [-comment "<string>"] [-state (ENABLED | DISABLED)]`
- `save ns config`

Example

The following example adds all web form fields whose names begin with `Password` to the confidential fields list.

```
add appfw confidField Password "https?://www[.]example[.]com/[<>]*[<a-z>]password[0-9a-z._-]"
save ns config
```

To modify a confidential field by using the command line interface

At the command prompt, type the following commands:

- `set appfw confidField <fieldName> <url> [-isRegex (REGEX | NOTREGEX)] [-comment "<string>"] [-state (ENABLED | DISABLED)]`
- `save ns config`

Example

The following example modifies the confidential field designation to add a comment.

```
set appfw confidField Password "https?://www[.]example[.]com/[<>]*[<a-z>]password[0-9a-z._-]"
save ns config
```

To remove a confidential field by using the command line interface

At the command prompt, type the following commands:

- `rm appfw confidField <fieldName> <url>`
- `save ns config`

To configure a confidential field by using the configuration utility

1. Navigate to Security > Application Firewall.
2. In the details pane, under Settings, click Manage Confidential Fields.
3. In the Manage Confidential Fields dialog box, do one of the following:
 - To add a new form field to the list, click Add.
 - To change an existing confidential field designation, select the field, and then click Edit.

The Application Firewall Confidential Fields dialog box appears.

Note: If you select an existing confidential field designation and then click Add, the Create Confidential Form Field dialog box displays the information for that confidential field. You can modify that information to create your new confidential field.

4. In the dialog box, fill out the elements. They are:
 - **Enabled check box.** Select or clear to enable/disable this confidential field designation.
 - **Is form field name a regular expression check box.** Select or clear to enable PCRE-format regular expressions in the form field name.
 - **Field Name.** Enter a literal string or PCRE-format regular expression that either represents a specific field name or that matches multiple fields with names that follow a pattern.
 - **Action URL.** Enter a literal URL or a regular expression that defines one or more URLs of the web page(s) on which the web form(s) that contains the confidential field are located.
 - **Comments.** Enter a comment. Optional.
5. Click Create or OK.
6. To remove a confidential field designation from the confidential fields list, select the confidential field listing you want to remove, then click Remove to remove it, and then click OK to confirm your choice.
7. When you have finished adding, modifying, and removing confidential field designations, click Close.

Examples

Following are some regular expressions that define form field names that you might find useful:

- `^passwd_` (Applies confidential-field status to all field names that begin with the "passwd_" string.)
- `^([0-9a-zA-Z._-]*|\\x[0-9A-Fa-f][0-9A-Fa-f])+)?passwd_` (Applies confidential-field status to all field names that begin with the string `passwd_`, or that contain the string `-passwd_` after another string that might contain non-ASCII special characters.)

Following are some regular expressions that define specific URL types that you might find useful. Substitute your own web host(s) and domain(s) for those in the examples.

- If the web form appears on multiple web pages on the web host `www.example.com`, but all of those web pages are named `login.pl?`, you could use the following regular expression:

```
https?://www[.]example[.]com/([0-9A-Za-z][0-9A-Za-z_-]*)?login[.]pl\?
```

- If the web form appears on multiple web pages on the web host `www.example-español.com`, which contains the n-tilde (Ñ) special character, you could use the following regular expression, which represents the n-tilde special character as an encoded UTF-8 string containing C3 B1, the hexadecimal code assigned to that character in the UTF-8 charset:

```
https?://www[.]example-espa\xC3\xB1ol[.]com/([0-9A-Za-z][0-9A-Za-z_-]*)? login
```

- If the web form containing `query.pl` appears on multiple web pages on different hosts within the `example.com` domain, you could use the following regular expression:

```
https?://([0-9A-Za-z][0-9A-Za-z_-]*[.])*example[.]com/([0-9A-Za-z][0-9A-Za-z_-]
```

- If the web form containing `query.pl` appears on multiple web pages on different hosts in different domains you could use the following regular expression:

`https?://([0-9A-Za-z][0-9A-Za-z_-.]*[.])*[0-9A-Za-z][0-9A-Za-z_-.]+[.][a-z]{2,6}`

- o If the web form appears on multiple web pages on the web host `www.example.com`, but all of those web pages are named `logon.pl?`, you could use the following regular expression:

`https?://www[.]example[.]com/([0-9A-Za-z][0-9A-Za-z_-.]*)logon[.]pl\?`

Field Types

A field type is a PCRE-format regular expression that defines a particular data format and minimum/maximum data lengths for a form field in a web form. Field types are used in the Field Formats check.

The application firewall comes with several default field types, which are:

- `integer`. A string of any length consisting of numbers only, without a decimal point, and with an optional preceding minus sign (-).
- `alpha`. A string of any length consisting of letters only.
- `alphanum`. A string of any length consisting of letters and/or numbers.
- `nohtml`. A string of any length consisting of characters, including punctuation and spaces, that does not contain HTML symbols or queries.
- `any`. Anything at all.
Important: Assigning the `any` field type as the default field type, or to a field, allows active scripts, SQL commands, and other possibly dangerous content to be sent to your protected web sites and applications in that form field. You should use the `any` type sparingly, if you use it at all.

You can also add your own field types to the Field Types list. For example, you might want to add a field type for a social security number, postal code, or phone number in your country. You might also want to add a field type for a customer identification number or store credit card number.

To add a field type to the Field Types list, you enter the field name as a literal string or PCRE-format regular expression.

To add a field type by using the command line interface

At the command prompt, type the following commands:

- `add appfw fieldType <name> <regex> <priority> [-comment "<string>"]`
- `save ns config`

Example

The following example adds a field type named `SSN` that matches US Social Security numbers to the Field Types list, and sets its priority to 1.

```
add appfw fieldType SSN "[1-9][0-9]{2,2}-[0-9]{2,2}-[0-9]{4,4}$" 1
save ns config
```

To modify a field type by using the command line interface

At the command prompt, type the following commands:

- `set appfw fieldType <name> <regex> <priority> [-comment "<string>"]`
- `save ns config`

Example

The following example modifies the field type to add a comment.

```
set appfw fieldType SSN "[1-9][0-9]{2,2}-[0-9]{2,2}-[0-9]{4,4}$" 1 -comment "US Social Secu
save ns config
```

To remove a field type by using the command line interface

At the command prompt, type the following commands:

- `rm appfw fieldType <name>`
- `save ns config`

To configure a field type by using the configuration utility

1. Navigate to Security > Application Firewall.
2. In the details pane, under Settings, click Manage Field Types.

3. In the Manage Field Types dialog box, do one of the following:
 - o To add a new field type to the list, click Add.
 - o To change an existing field type, select the field type, and then click Edit.

The Configure Field Type dialog box appears.

Note: If you select an existing field type designation and then click Add, the dialog box displays the information for that field type. You can modify that information to create your new field type.

4. In the dialog box, fill out the elements. They are:
 - o Name
 - o Regular Expression
 - o Priority
 - o Comment
5. Click Create or OK.
6. To remove a field type from the Field Types list, select the field type listing you want to remove, then click Remove to remove it, and then click OK to confirm your choice.
7. When you have finished adding, modifying, and removing field types, click Close.

Examples

Following are some regular expressions for field types that you might find useful:

- o `^[1-9][0-9]{2,2}-[0-9]{2,2}-[0-9]{4,4}$` U.S. Social Security numbers
- o `^[A-C][0-9]{7,7}$` California driver's license numbers.
- o `^[+][0-9]{1,3} [0-9() -]{1,40}$` International phone numbers with country codes.
- o `^[0-9]{5,5}-[0-9]{4,4}$` U.S. ZIP code numbers.
- o `^[0-9A-Za-z][0-9A-Za-z._+-]{0,25}@([0-9A-Za-z][0-9A-Za-z_-]*[.])\{1,4\}[A-Za-z]{2,6}$` Email addresses.

XML Content Types

By default, the application firewall treats files that follow certain naming conventions as XML. You can configure the application firewall to examine web content for additional strings or patterns that indicate that those files are XML files. This can ensure that the application firewall recognizes all XML content on your site, even if certain XML content does not follow normal XML naming conventions, ensuring that XML content is subjected to XML security checks.

To configure the XML content types, you add the appropriate patterns to the XML Content Types list. You can enter a content type as a string, or you can enter a PCRE-compatible regular expression specifying one or more strings. You can also modify the existing XML content types patterns.

To add an XML content type pattern by using the command line interface

At the command prompt, type the following commands:

- o add appfw XMLContentType <XMLContenttypevalue> [-isRegex (REGEX | NOTREGEX)]
- o save ns config

Example

The following example adds the pattern `.*/*.xml` to the XML Content Types list and designates it as a regular expression.

```
add appfw XMLContentType ".*/*.xml" -isRegex REGEX
```

To remove an XML content type pattern by using the command line interface

At the command prompt, type the following commands:

- o rm appfw XMLContentType <XMLContenttypevalue>
- o save ns config

To configure the XML content type list by using the configuration utility

1. Navigate to Security > Application Firewall.
2. In the details pane, under Settings, click Manage XML Content Types.
3. In the Manage XML Content Types dialog box, do one of the following:
 - o To add a new XML content type, click Add.
 - o To modify an existing XML content type, select that type and then click Edit.The Configure Application Firewall XML Content Type dialog appears.
Note: If you select an existing XML content type pattern and then click Add, the dialog box displays the information for that XML content type pattern. You can modify that information to create your new XML content type pattern.
4. In the dialog box, fill out the elements. They are:
 - o **IsRegex.** Select or clear to enable PCRE-format regular expressions in the form field name.
 - o **XML Content Type** Enter a literal string or PCRE-format regular expression that matches the XML content type pattern that you want to add.
5. Click Create.
6. To remove an XML content type pattern from the list, select it, then click Remove to remove it, and then click OK to confirm your choice.
7. When you have finished adding and removing XML content type patterns, click Close.

JSON Content Types

By default, the application firewall treats files with the content type "application/json" as JSON files. The default setting enables the application firewall to recognize JSON content in requests and responses, and to handle that content appropriately.

You can configure the application firewall to examine web content for additional strings or patterns that indicate that those files are JSON files. This can ensure that the application firewall recognizes all JSON content on your site, even if certain JSON content does not follow normal JSON naming conventions, ensuring that JSON content is subjected to JSON security checks.

To configure the JSON content types, you add the appropriate patterns to the JSON Content Types list. You can enter a content type as a string, or you can enter a PCRE-compatible regular expression specifying one or more strings. You can also modify the existing JSON content types patterns.

To add a JSON content type pattern by using the command line interface

At the command prompt, type the following commands:

- add appfw JSONContentType <JSONContenttypevalue> [-isRegex (REGEX | NOTREGEX)]
- save ns config

Example

The following example adds the pattern `.*\/json` to the JSON Content Types list and designates it as a regular expression.

```
add appfw JSONContentType ".*\/json" -isRegex REGEX
```

To configure the JSON content type list by using the configuration utility

1. Navigate to Security > Application Firewall.
2. In the details pane, under Settings, click Manage JSON Content Types.
3. In the Manage JSON Content Types dialog box, do one of the following:
 - To add a new JSON content type, click Add.
 - To modify an existing JSON content type, select that type and then click Edit.The Configure Application Firewall JSON Content Type dialog appears.
Note: If you select an existing JSON content type pattern and then click Add, the dialog box displays the information for that JSON content type pattern. You can modify that information to create your new JSON content type pattern.
4. In the dialog box, fill out the elements. They are:
 - **IsRegex.** Select or clear to enable PCRE-format regular expressions in the form field name.
 - **JSON Content Type** Enter a literal string or PCRE-format regular expression that matches the JSON content type pattern that you want to add.
5. Click Create or OK.
6. To remove a JSON content type pattern from the list, select it, then click Remove to remove it, and then click OK to confirm your choice.
7. When you have finished adding and removing XML content type patterns, click Close.

Logs, Statistics, and Reports

The information maintained in the logs and statistics, and displayed in the reports, provides important guidance for configuring and maintaining the application firewall.

The Application Firewall Logs

The logs provide information about the requests and responses that the application firewall has observed while protecting your web sites and applications. Most important, it logs each connection that matches a signature or a security check. You can observe the logs to determine which connections are matching a signature or security check. You can then use this information, along with your own knowledge about your protected web sites and applications, to determine whether the connections that each signature or check is matching are valid (false positives). If they are, you can either remove the signature or check from your configuration, or take appropriate measures to mitigate the false positives before you enable blocking for that signature or security check.

NetScaler Format Logs

When configured to use NetScaler format logs, the application firewall produces logs that follow the same format as other NetScaler features. Each log contains the following fields:

- **Timestamp.** The date and time when the connection occurred.
- **Severity.** The severity level of the log.
- **Module.** The NetScaler module that generated the log entry.
- **Event Type.** The type of event, such as signature violation or security check violation.
- **Event ID.** The ID assigned to the event.
- **Client IP.** The IP address of the user whose connection was logged.
- **Transaction ID.** The ID assigned to the transaction that caused the log.
- **Session ID.** The ID assigned to the user session that caused the log.
- **Message.** The log message. Contains information identifying the signature or security check that triggered the log entry.

You can search on any of these fields, or any combination of information from different fields, to select logs to display, limited only by the capabilities of the tools you use to view the logs. You can observe the signatures by using the application firewall wizard to access the NetScaler syslog viewer, or manually by logging onto the NetScaler appliance or NetScaler virtual appliance.

Viewing the Application Firewall Logs

You can view the logs by using the syslog viewer, or by logging onto the NetScaler appliance, opening a Unix shell, and using the Unix text editor of your choice.

- **Viewing by using the syslog viewer.** You invoke the syslog viewer from one of two locations: the Select Signature Actions page or the Select Advanced Actions page in the Application Firewall Wizard. To invoke the syslog viewer for a signature, in the Select Signature Actions pane click the logs link to the right of that signature. To invoke the syslog viewer for a security check, in the Select Advanced Actions page, security checks list, select that security check, and then beneath the list click the Logs button. Either procedure causes the configuration utility to download the current `ns.log` file and then display the entries that are relevant to that signature or security check.

The syslog viewer contains the following elements:

Module list box. The NetScaler module whose logs you want to view. Always set to APPFW for application firewall logs.

Event Type list box. The type of event. For signatures, this is always APPFW_SIGNATURE_MATCH. For security checks, this is the specific security check that you selected.

Severity. It lets you specify only logs of a specific severity level. Leave blank to see all logs.

Find Now button. Search the `nslog.file`, using the current criteria, and display the logs that match.

Clear button. Resets your settings to the defaults.

Logs display window. Displays the logs that meet the current criteria. Log information is displayed in several columns that correspond to the log fields for the log format that the application firewall is currently configured to maintain, with an additional column, Deploy, to the extreme left. You can sort the display by clicking a column heading. You can create and implement a relaxation for a signature or security check that is blocking legitimate use of a protected web site or web service by selecting a log that shows the unwanted blocking, and then clicking Deploy.

Log directory. The directory where the logs are stored. If you have archived logs stored in a different directory and want to view those, you can click Browse and browse to that directory to display those logs in the Log files list.

Log files list. A list of the log files in the Log directory. To download and uncompress an archived log file, select the file, and then click Download. To refresh the display, click Refresh.

Search in list box. Searches in a particular section of logs when selecting logs to display in the Logs display window. To search something other than the log message, select a different choice.

Search string. Search for the specified string or regular expression to choose the logs to display in the Logs display window. This field is filled out by the application firewall wizard for you with the appropriate value to display the logs relevant to the signature or security check that you selected. You can modify the string to choose logs based on different criteria.

Case Sensitive check box. Select if the Search string is case sensitive.

Regular Expression check box. Select if the Search string is a regular expression.

Clear button. Resets the syslog viewer to its default settings.

Go button. Uses the new search criteria to search the `ns.log` file and displays the results in the Logs display window.

For more information about the Application Firewall Wizard, see "[The Application Firewall Wizard](#)."

- o **Viewing from the command line.** Log onto the application firewall appliance, and then type the following command at the NetScaler command prompt:

```
shell
```

After the Unix shell is displayed, type the following command to navigate to the directory where the logs are stored:

```
cd /var/log
```

You can use the vi editor, or any Unix text editor or text search tool of your choice to view and filter the logs for specific entries.

Note: If the text editor or text search tool is not installed by default on the NetScaler appliance, you must first install it before you can use it to view and filter the logs.

The Application Firewall Statistics

When you enable the statistics action for application firewall signatures or security checks, the application firewall maintains information about connections that match that signature or security check. You can view the accumulated statistics information on the Monitoring tab of the main logon page of your application firewall appliance by selecting one of the following choices in the Select Group list box:

- o **Application Firewall.** A summary of all statistics information gathered by your application firewall appliance for all profiles.
- o **Application Firewall (per profile).** The same information, but displayed per-profile rather than summarized.

You can use this information to monitor how your application firewall is operating and determine whether there is any abnormal activity or abnormal amounts of hits on a signature or security check. If you see such a pattern of abnormal activity, you can check the logs for that signature or security check, to diagnose the issue, and then take corrective action.

The Application Firewall Reports

The application firewall reports provide information about your application firewall configuration and how it is handling traffic for your protected web sites.

The PCI DSS Report

The Payment Card Industry (PCI) Data Security Standard (DSS), version 1.2, consists of twelve security criteria that most credit card companies require businesses who accept online payments via credit and debit cards to meet. These criteria are designed to prevent identity theft, hacking, and other types of fraud. If an internet service provider or online merchant does not meet the PCI DSS criteria, that ISP or merchant risks losing authorization to accept credit card payments through its web site.

ISPs and online merchants prove that they are in compliance with PCI DSS by having an audit conducted by a PCI DSS Qualified Security Assessor (QSA) Company. The PCI DSS report is designed to assist them both before and during the audit. Before the audit, it shows which application firewall settings are relevant to PCI DSS, how they should be

configured, and (most important) whether your current application firewall configuration meets the standard. During the audit, the report can be used to demonstrate compliance with relevant PCI DSS criteria.

The PCI DSS report consists of a list of those criteria that are relevant to your application firewall configuration. Under each criterion, it lists your current configuration options, indicates whether your current configuration complies with the PCI DSS criterion, and explains how to configure the application firewall so that your protected web site(s) will be in compliance with that criterion.

The PCI DSS report is located under System > Reports. To generate the report as an Adobe PDF file, click Generate PCI DSS Report. Depending on your browser settings, the report is displayed in the pop-up window or you are prompted to save it to your hard disk.

Note: To view this and other reports, you must have the Adobe Reader program installed on your computer.

The PCI DSS report consists of the following sections:

- **Description.** A description of the PCI DSS Compliance Summary report.
- **Firewall License and Feature Status.** Tells you whether the application firewall is licensed and enabled on your NetScaler appliance.
- **Executive Summary.** A table that lists the PCI DSS criteria and tells you which of those criteria are relevant to the application firewall.
- **Detailed PCI DSS Criteria Information.** For each PCI DSS criterion that is relevant to your application firewall configuration, the PCI DSS report provides a section that contains information about whether your configuration is currently in compliance and, if it is not, how to bring it into compliance.
- **Configuration.** Data for individual profiles, which you access either by clicking Application Firewall Configuration at the top of the report, or directly from the Reports pane. The Application Firewall Configuration report is the same as the PCI DSS report, with the PCI DSS-specific summary omitted, and is described below.

The Application Firewall Configuration Report

The Application Firewall Configuration report is located under System > Reports. To display it, click Generate Application Firewall Configuration Report. Depending on your browser settings, the report is displayed in the pop-up window or you are prompted to save it to your hard disk.

The Application Firewall Configuration report starts with a Summary page, which consists of the following sections:

- **Application Firewall Policies.** A table that lists your current application firewall policies, showing the policy name, the content of the policy, the action (or profile) it is associated with, and global binding information.
- **Application Firewall Profiles.** A table that lists your current application firewall profiles and indicates which policy each profile is associated with. If a profile is not associated with a policy, the table displays INACTIVE in that location.

To download all report pages for all policies, at the top of the Profiles Summary page click Download All Profiles. You display the report page for each individual profile by selecting that profile in the table at the bottom of the screen. The Profile page for an individual profile shows whether each check action is enabled or disabled for each check, and the other configuration settings for the check.

To download a PDF file containing the PCI DSS report page for the current profile, click Download Current Profile at the top of the page. To return to the Profiles Summary page, click Application Firewall Profiles. To go back to the main page, click Home. You can refresh the PCI DSS report at any time by clicking Refresh in the upper right corner of the browser. You should refresh the report if you make changes to your configuration.

Configuring the Application Firewall Logs

You can configure the application firewall logs by using the configuration utility or the NetScaler command line. You can configure the application firewall to produce logs in either native NetScaler format, or in Common Event Format (CEF).

To configure the Application Firewall logs by using the command line interface

At the command prompt, type the following commands:

- set appfw settings -CEFLogging (**ON** | **OFF**)
- save ns config

Example

The following example configures the application firewall to use CEF logs.

```
set appfw settings -CEFLogging ON
save ns config
```

The following example disables CEF logs and returns the application firewall configuration to using native NetScaler format logs.

```
set appfw settings -CEFLogging OFF
save ns config
```

Appendices

The following supplemental material provides additional detail about complex or peripheral application firewall tasks.

PCRE Character Encoding Format

The NetScaler operating system supports direct entry of characters in the printable ASCII character set only. Characters with hexadecimal codes between HEX 20 (ASCII 32) and HEX 7E (ASCII 127). To include a character with a code outside that range in your application firewall configuration, you must enter its UTF-8 hexadecimal code as a PCRE regular expression.

A number of character types require encoding using a PCRE regular expression if you include them in your application firewall configuration as a URL, form field name, or Safe Object expression. They include:

- **Upper-ASCII characters.** Characters with encodings from HEX 7F (ASCII 128) to HEX FF (ASCII 255). Depending on the character map used, these encodings can refer to control codes, ASCII characters with accents or other modifications, non-Latin alphabet characters, and symbols not included in the basic ASCII set. These characters can appear in URLs, form field names, and safe object expressions.
- **Double-Byte characters.** Characters with encodings that use two 8-byte words. Double-byte characters are used primarily for representing Chinese, Japanese, and Korean text in electronic format. These characters can appear in URLs, form field names, and safe object expressions.
- **ASCII control characters.** Non-printable characters used to send commands to a printer. All ASCII characters with hexadecimal codes less than HEX 20 (ASCII 32) fall into this category. These characters should never appear in a URL or form field name, however, and would rarely if ever appear in a safe object expression.

The NetScaler appliance does not support the entire UTF-8 character set, but only the characters found in the following eight charsets:

- **English US (ISO-8859-1).** Although the label reads, "English US," the application firewall supports all characters in the ISO-8859-1 character set, also called the Latin-1 character set. This character set fully represents most modern western European languages and represents all but a few uncommon characters in the rest.
- **Chinese Traditional (Big5).** The application firewall supports all characters in the BIG5 character set, which includes all of the Traditional Chinese characters (ideographs) commonly used in modern Chinese as spoken and written in Hong Kong, Macau, Taiwan, and by many people of Chinese ethnic heritage who live outside of mainland China.
- **Chinese Simplified (GB2312).** The application firewall supports all characters in the GB2312 character set, which includes all of the Simplified Chinese characters (ideographs) commonly used in modern Chinese as spoken and written in mainland China.
- **Japanese (SJIS).** The application firewall supports all characters in the Shift-JIS (SJIS) character set, which includes most characters (ideographs) commonly used in modern Japanese.
- **Japanese (EUC-JP).** The application firewall supports all characters in the EUC-JP character set, which includes all characters (ideographs) commonly used in modern Japanese.
- **Korean (EUC-KR).** The application firewall supports all characters in the EUC-KR character set, which includes all characters (ideographs) commonly used in modern Korean.
- **Turkish (ISO-8859-9).** The application firewall supports all characters in the ISO-8859-9 character set, which includes all letters used in modern Turkish.
- **Unicode (UTF-8).** The application firewall supports certain additional characters in the UTF-8 character set, including those used in modern Russian.

When configuring the application firewall, you enter all non-ASCII characters as PCRE-format regular expressions using the hexadecimal code assigned to that character in the UTF-8 specification. Symbols and characters within the normal ASCII character set, which are assigned single, two-digit codes in that character set, are assigned the same codes in the UTF-8 character set. For example, the exclamation point (!), which is assigned hex code 21 in the ASCII character

set, is also hex 21 in the UTF-8 character set. Symbols and characters from another supported character set have a paired set of hexadecimal codes assigned to them in the UTF-8 character set. For example, the letter a with an acute accent (Ã¡) is assigned UTF-8 code C3 A1.

The syntax you use to represent these UTF-8 codes in the application firewall configuration is `â€œ\xNNâ€•` for ASCII characters; `â€œ\xNN\xNNâ€•` for non-ASCII characters used in English, Russian, and Turkish; and `â€œ\xNN\xNN\xNNâ€•` for characters used in Chinese, Japanese, and Korean. For example, if you want to represent a ! in an application firewall regular expression as a UTF-8 character, you would type `\x21`. If you want to include an Ã¡, you would type `\xC3\xA1`.

Note: Normally you do not need to represent ASCII characters in UTF-8 format, but when those characters might confuse a web browser or an underlying operating system, you can use the characterâ€™s UTF-8 representation to avoid this confusion. For example, if a URL contains a space, you might want to encode the space as `\x20` to avoid confusing certain browsers and web server software.

Below are examples of URLs, form field names, and safe object expressions that contain non-ASCII characters that must be entered as PCRE-format regular expressions to be included in the application firewall configuration. Each example shows the actual URL, field name, or expression string first, followed by a PCRE-format regular expression for it.

- o A URL containing extended ASCII characters.

Actual URL: `http://www.josÃ©nuÃ±ez.com`

Encoded URL: `^http://www[.]jos\xC3\xA9nu\xC3\xB1ez[.]com$`

- o Another URL containing extended ASCII characters.

Actual URL: `http://www.example.de/trÃ¼mso.html`

Encoded URL: `^http://www[.]example[.]de/tr\xC3\xB6mso[.]html$`

- o A form field name containing extended ASCII characters.

Actual Name: `nome_do_usuÃ¡rio`

Encoded Name: `^nome_do_usu\xC3\xA1rio$`

- o A safe object expression containing extended ASCII characters.

Unencoded Expression `[A-Z]{3,6}Ã¢[1-9][0-9]{6,6}`

Encoded Expression: `[A-Z]{3,6}\xC2\xA5[1-9][0-9]{6,6}`

You can find a number of tables that include the entire Unicode character set and matching UTF-8 encodings on the Internet. A useful web site that contains this information is located at the following URL:

<http://www.utf8-chartable.de/unicode-utf8-table.pl>

For the characters in the table on this web site to display correctly, you must have an appropriate Unicode font installed on your computer. If you do not, the visual display of the character may be in error. Even if you do not have an appropriate font installed to display a character, however, the description and the UTF-8 and UTF-16 codes on this set of web pages will be correct.

Whitehat WASC Signature Types for WAF Use

The Citrix NetScaler Application Firewall accepts and generates blocking rules for all vulnerability types that the Whitehat scanners generate. However, certain vulnerabilities are most applicable to a web application firewall. Following are lists of those vulnerabilities, categorized by whether they are addressed by WASC 1.0, WASC 2.0, or best practices signature types.

WASC 1.0 Signature Types

- HTTP Request Smuggling
- HTTP Response Splitting
- HTTP Response Smuggling
- Null Byte Injection
- Remote File Inclusion
- URL Redirector Abuse

WASC 2.0 Signature Types

- Abuse of Functionality
- Brute Force
- Content Spoofing
- Denial of Service
- Directory Indexing
- Information Leakage
- Insufficient Anti-automation
- Insufficient Authentication
- Insufficient Authorization
- Insufficient Session Expiration
- LDAP Injection
- Session Fixation

Best Practices

- Autocomplete Attribute
- Insufficient Cookie Access Control
- Insufficient Password Strength
- Invalid HTTP Method Usage
- Non-HttpOnly Session Cookie
- Persistent Session Cookie
- Personally Identifiable Information
- Secured Cachable HTTP Messages
- Unsecured Session Cookie

Streaming Support for Request Processing

Note: This feature is available in NetScaler release 10.5.e.

The Citrix application firewall now uses request side streaming, which results in a significant performance boost. Instead of buffering the entire request before processing it, the application firewall now looks at the incoming data, field by field, to inspect the input of each field for any configured security check violation (SQL, XSS, Field Consistency, Field Formats, etc.). As soon as the processing of the data for a field is completed, it is forwarded to the backend while the evaluation continues for the remaining fields. This significantly improves the processing time specially when handling large posts where the forms have large number of fields.

Although the streaming process is transparent to the users, minor configuration adjustments are required due to the following changes:

RegEx Pattern Match: RegEx pattern match is now restricted to 4K for contiguous character string match.

Field Name Match: Application firewall learning engine can only distinguish the first 128 bytes of the name for learning. If a form has multiple fields with names that have identical string match for the first 128 bytes, the learning engine may not be able to distinguish between them. Similarly, the deployed relaxation rule might inadvertently relax all such fields.

Removing white spaces, percent decoding, unicode decoding, and charset conversion which is done during canonicalization is carried out prior to security check inspection. The 128 byte limit is applicable to the canonicalized representation of the field name in UTF-8 character format. The ASCII characters are 1 byte but the UTF-8 representation of the characters in some international languages may range from 1-4 bytes. If each character in the name takes 4 bytes when converted to UTF-8 format, only first 32 characters in the name may be distinguished by the learned rule for such a language.

Field Consistency Check: When the field Consistency check is enabled, all the forms in the session are now stored based on the `as_fid` tag inserted by the application firewall without consideration for the `action_url`.

- **Mandatory Form tagging for Form Field consistency:** When the field consistency check is enabled, the form tag must be enabled also. The Field Consistency protection might not work if form tagging is turned off.
- **Sessionless Form Field Consistency:** The application firewall no longer carries out the `GET` to `POST` conversion of forms when sessionless field consistency parameter is enabled. The form tag is required for sessionless field consistency also.
- **Tampering of `as_fid`:** If a form is submitted after tampering `as_fid`, it now triggers field consistency violation even if no other field was tampered. In non-streaming requests, this was allowed because the forms could be validated using the `action_url` stored in the session.

Signatures: The signatures now have the following specifications:

- **Location:** It is now a mandatory requirement that location must be specified for each pattern. All patterns in the rule **MUST** have a `<Location>` tag.
- **Fast Match:** All signature rules must have a fast match pattern. If there is no fast match pattern, an attempt will be made to select one if possible. Fast match must be a literal string but some PCRE™s can be used for fast match if they contain a usable literal string.
- **Deprecated Locations:** Following locations are no longer supported in signature rules.
 - HTTP_ANY
 - HTTP_RAW_COOKIE
 - HTTP_RAW_HEADER
 - HTTP_RAW_RESP_HEADER
 - HTTP_RAW_SET_COOKIE

XSS/SQL Transform: Raw data is used for transformation because the SQL special characters (single quote('), backslash (\), and semicolon (;)), and XSS tags (< and >)) are same in all languages and do not need canonicalization of data. All representations of these characters, such as HTML entity encoding, percent encoding, or ASCII are evaluated for transform operation.

The application firewall no longer inspects both the attribute name and value for the XSS transform operation. Now only XSS attribute names are transformed when streaming is engaged.

RAW POST Body: The security check inspections are always done on RAW POST body.

Form ID: The application firewall inserted `as_fid` tag, which is a computed hash of the form, will no longer be unique for the user session. It will now have an identical value for a specific form irrespective of the user or the session.

Charset: If a request does not have a charset, the default charset specified in the application profile is used when processing the request.

Counters:

Counters with prefix `se_` and `appfwreq_` are added to track the streaming engine and the application firewall streaming engine request counters respectively.

```
nsconsmg -d statswt0 -g se_err_
```

```
nsconsmg -d statswt0 -g se_tot_
```

```
nsconsmg -d statswt0 -g se_cur_
```

```
nsconsmg -d statswt0 -g appfwreq_err_
```

```
nsconsmg -d statswt0 -g appfwreq_tot_
```

```
nsconsmg -d statswt0 -g appfwreq_cur_
```

`_err` counters: indicate the rare event which should have succeeded but failed due to either memory allocation problem or some other resource crunch.

`_tot` counters: ever increasing counters.

`_cur` counters: counters indicating current values that keep changing based on usage from current transactions.

Tips:

- The application firewall security checks should work exactly the same as before.
- There is no set ordering for the processing of the security checks.
- The response side processing is not affected and remains unchanged.
- Streaming is not engaged if CVPN is used.

Trace HTML Requests with Security Logs

Note: This feature is available in NetScaler release 10.5.e.

Troubleshooting a problem, which requires analysis of data received in the client request can be quite challenging specially when there is heavy traffic flowing through the box. Diagnosing issues which may affect the functionality or security of the application require a quick response.

The NetScaler now offers the option to isolate traffic for a specific application firewall profile and collect nstrace for the HTML requests that trigger a log or block action or malformed requests that might be causing reset or aborts. The nstrace collected in "appfw mode" will include details of the entire request including the application firewall generated log messages. You can use "Follow TCP stream" in the trace to view the details of the individual transaction including headers, payload, as well as the corresponding log message, together in the same screen.

This gives you a comprehensive overview regarding your traffic. Having a detailed view of the request, payload, and associated log records can be very useful to analyze security check violation. You can easily identify the pattern that is triggering the violation. If the pattern should be allowed, you can take a decision to modify the configuration and/or add a relaxation rule.

This enhancement helps in troubleshooting the NetScaler ADC and offers the following benefits:

1. **Isolate traffic for specific profile:** This enhancement can be quite useful when you need to isolate traffic for only one profile or specific transactions of a profile for troubleshooting. You no longer have to skim through the entire data collected in the trace or need special filters to isolate requests that are of interest to you which can be tedious especially with heavy traffic. You now have the option to view only the data that you are interested in.
2. **Collect data for specific requests:** The trace can be collected for a specified duration. You can collect trace for only a couple of requests to isolate, analyze, and debug specific transactions if needed.
3. **Identify resets or aborts:** Unexpected closing of connections are not easily visible. The trace collected in "appfw mode" captures a reset or an abort, triggered by the application firewall. This allows a quicker isolation of issue when you do not see a security check violation message. Malformed requests or other non-RFC compliant requests terminated by application firewall will now be easier to identify.
4. **View decrypted SSL traffic:** HTTPS traffic is captured in plain text to allow for easier troubleshooting.
5. **Provides comprehensive view:** Allows you to look at the entire request at the packet level, check the payload, look at the logs to check what security check violation is being triggered and identify the match pattern in the payload. If the payload consists of any unexpected data, junk strings, or non-printable characters (null character, \r or \n etc), they are easy to discover in the trace.
6. **Modify configuration:** The debugging can provide useful information to decide if the observed behavior is the correct behavior or the configuration should be modified.
7. **Expedite response time:** Faster debugging on target traffic can improve the response time to provide explanations and/or root cause analysis by Citrix engineering and support team.

Please see any task topic in eDocs for documenting tasks. <http://support.citrix.com/proddocs/topic/ns-security-10-5-map/appfw-config-manual-cli-tsk.html>

To configure debug tracing for a profile by using the command line interface

Step 1. Enable tracing for the profile. You can use the show command to verify the configured setting.

- set appfw profile <profile> -trace ON

Step 2. Start collecting trace. You can continue to use all the options which are applicable for the nstrace command.

- start nstrace -mode APPFW

Stop collecting the trace

- stop nstrace

Location of the trace: The nstrace is stored in a time-stamped folder which is created in the /var/nstrace directory and can be viewed using wireshark. You can tail the /var/log/ns.log to see the log messages providing details regarding the location of the new trace.

Tips:

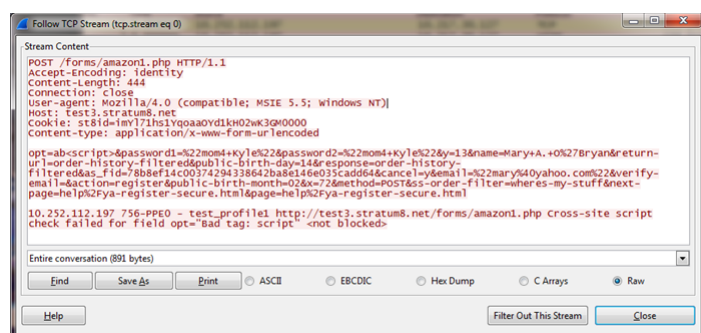
- When "appfw mode" option is used, the nstrace will only collect the data for the profile(s) for which "trace" was enabled.
- Enabling trace on the profile will not automatically start collecting the traces till you explicitly execute the "start nstrace" command to collect the trace.

- Although, enabling trace on a profile may not have any adverse effect on the performance of the application firewall but you may want to enable this feature only for the duration for which you want to collect the data. It is recommended that you turn the “trace flag off after you have collected the trace. This will prevent the risk of inadvertently getting data from profiles for which you had enabled this flag in the past.
- The Block or Log action must be enabled for the security check for the transaction record to be included in the nstrace.
- Resets and aborts will be logged independently of security checks actions when trace is “On” for the profile(s).
- This feature is only applicable for troubleshooting the requests received from the client. The traces in “appfw mode do not include the responses received from the server.
- You can continue to use all the options which are applicable for the nstrace command. For example,

"start nstrace -tcpdump enabled -size 0 -mode appFW"

- If a request triggers multiple violations, the nstrace for that record will include all the corresponding log messages.
- CEF log message format is supported for this functionality.
- Signature violations triggering block and/or log action for request side checks will also be included in the trace.
- Only HTML (non-XML) requests are collected in the trace.

Example of a Log record in the trace:



Content Filtering

Content filtering can do some of the same tasks as the Citrix NetScaler Application Firewall, and is a less CPU-intensive tool. It is limited, however, to examining the header portion of the HTTP request or response and to performing a few simple actions on connections that match. If you have a complex Web site that makes extensive use of scripts and accesses back-end databases, the Application Firewall may be the better tool for protecting that Web site. For more information about the Citrix NetScaler Application Firewall, see the *Citrix Application Firewall Guide* at <http://support.citrix.com/article/CTX132360>.

Content filtering is based on regular expressions that you can apply to either HTTP requests or HTTP responses. To block requests from a particular site, for example, you could use an expression that compares each request's URL to the URL specified in the expression. The expression is part of a policy, which also specifies an action to be performed on requests or responses that match the expression. For example, an action might drop a request or reset the connection.

Following are some examples of things you can do with content filtering policies:

- Prevent users from accessing certain parts of your Web sites unless they are connecting from authorized locations.
- Prevent inappropriate HTTP headers from being sent to your Web server, possibly breaching security.
- Redirect specified requests to a different server or service.

To configure content filtering, once you have made sure that the feature is enabled, you configure filtering actions for your servers to perform on selected connections (unless the predefined actions are adequate for your purposes). Then you can configure policies to apply the actions to selected connections. Your policies can use predefined expressions, or you can create your own. To activate the policies you configured, you bind them either globally or to specific virtual servers.

To configure content filtering, do the following:

1. [Enabling Content Filtering](#)
2. [Configuring a Content Filtering Action](#)
3. [Configuring a Content Filtering Policy](#)
4. [Binding a Content Filtering Policy](#)

Enabling Content Filtering

By default, content filtering is enabled on NetScaler appliances running the NetScaler operating system 8.0 or above. If you are upgrading an existing appliance from an operating system version earlier than 8.0, you must update the licenses before you can use content filtering, and you may need to enable the content filtering feature itself manually.

To enable content filtering by using the command line interface

At the command prompt, type the following commands to enable content filtering and verify the configuration:

- `enable ns feature ContentFiltering`
- `show ns feature`

Example

```
> enable ns feature ContentFiltering
Done
> show ns feature
```

	Feature	Acronym	Status
	-----	-----	-----
1)	Web Logging	WL	ON
2)	Surge Protection	SP	OFF
.			
.			
.			
11)	Http DoS Protection	HDOSP	OFF
12)	Content Filtering	CF	ON
.			
.			
23)	HTML Injection	HTMLInjection	ON
24)	NetScaler Push	push	OFF

Done

To enable content by filtering by using the configuration utility

1. In the navigation pane, expand **System**, and then select **Settings**.
2. In the details pane, click **Configure basic features**.
3. In the Configure Basic Features dialog box, select the **Content Filter** check box, and then click **OK**.

Configuring a Content Filtering Action

After you enable the content filtering feature, you create one or more actions to tell your NetScaler appliance how to handle the connections it receives.

Content filter **Qualifier*** drop-down list allows you to select the **action** to be performed. Following options are available:

Reset

Terminates the connection, sending the appropriate termination notice to the user's browser.

Add

Adds the specified HTTP header before sending the request to the Web server.

Corrupt

Modifies the designated HTTP header in a manner that prevents it from performing the function it was intended to perform, then sends the request to the server.

Forward

Redirects the request to the designated service.

ErrorCode

Returns the designated HTTP error code to the user's browser with an option to configure a **Response page**.

Drop

Silently deletes the request, without sending a response to the user's browser.

To configure a content filtering action by using the command line interface

At the command prompt, type the following commands to configure a Content Filtering action and verify the configuration:

- add filter action <name> <qualifier> [<serviceName>] [<value>] [<respCode>] [<page>]
- show filter action <name>

Example

```
> add filter action act_drop Drop
Done
> show filter action act_drop
1)      Name: act_drop    Filter Type: drop
Done
```

To configure a content filtering action by using the configuration utility

1. Navigate to Security > Protection Features > Filter.
2. In the details pane, do one of the following:
 - To create a new filter action, in the **Filter Actions** Tab, click **Add**.
 - To modify an existing action, select the action, and then click **Edit**.
3. In the Add Filter Action or Configure Filter Action dialog box, specify values for the parameters:
 - Action Name "name"
 - Qualifier "qualifier" (Determines which of the following parameters you can configure)
 - Service Name "servicename"
 - HeaderName:Value "value"
 - Response Code "respcode"
 - Response Page "page"
4. Fill in any other required information. For example, if you are configuring an action to send an HTTP error code, you must choose the appropriate error code from a drop-down list. If necessary, you can then modify the text of the error message, which is displayed beneath the drop-down list.
5. Click Create or OK, and then click Close. The Actions list displays the action you configured, and a message in the status bar indicates that your action has been created.

Configuring a Content Filtering Policy

To implement content filtering, you must configure at least one policy to tell your NetScaler appliance how to distinguish the connections you want to filter. You must first have configured at least one filtering action, because when you configure a policy, you associate it with an action.

Content filtering policies examine a combination of one or more of the following elements to select requests or responses for filtering:

URL

The URL in the HTTP request.

URL query

Only the query portion of the URL, which is the portion after the query (?) symbol.

URL token

Only the tokens in the URL, if any, which are the parts that begin with an ampersand (&) and consist of the token name, followed by an equals sign (=), followed by the token value.

HTTP method

The HTTP method used in the request, which is usually GET or POST, but can be any of the eight defined HTTP methods.

HTTP version

The HTTP version in the request, which is usually HTTP 1.1.

Standard HTTP header

Any of the standard HTTP headers defined in the HTTP 1.1 specification.

Standard HTTP header value

The value portion of the HTTP header, which is the portion after the colon and space (:).

Custom HTTP header

A non-standard HTTP header issued by your Web site or that appears in a user request.

Custom header value

The value portion of the custom HTTP header, which (as with the standard HTTP header) is the portion after the color and space (:).

Client Source IP

The IP from which the client request was sent.

Content filtering policies use the simpler of two NetScaler expressions languages, called classic expressions. For a complete description of classic expressions, how they work, and how to configure them manually, see "[Policies and Expressions](#)."

Note: Users who are not experienced in configuring policies at the NetScaler command line will usually find using the configuration utility considerably easier.

To configure a content filtering policy by using the command line interface

At the command prompt, type the following commands to configure a content filtering policy and verify the configuration:

- add filter policy <name> -rule <expression> (-reqAction <action> | -resAction <string>)
- show filter policy <name>

Example

```
> add filter policy cf-pol -rule "REQ.HTTP.URL CONTAINS http://abc.com" -reqaction DROP
Done
> show filter policy cf-pol
1)      Name: cf-pol      Rule: REQ.HTTP.URL CONTAINS http://abc.com
      Request action: DROP
      Response action:
      Hits: 0
Done
```

To configure a content filtering policy by using the configuration utility

1. Navigate to Security > Protection Features > Filter.

2. Navigate to Protection Features > Filter.
 3. In the details pane, to create a new policy, click Add.
 4. If you are creating a new policy, in the Create Filter Policy dialog box, in the Filter Name text box, type a name for your new policy.
 5. Select either Request Action or Response Action to activate the drop-down list to the right of that item.
 6. Click the down arrow to the right of the drop-down list and select the action to be performed on the request or response. The default choices are RESET and DROP. Any other actions you have created will also appear in this list.
Note: You can also click New to create a new Content Filtering action, or Modify to modify an existing Content Filtering action. You can only modify actions you created; the default actions are read-only.
 7. If you want to use a predefined expression (or named expression) to define your policy, choose one from the Named Expressions list.
 - a. Click the down arrow to the right of the first Named Expressions drop-down list, and choose the category of named expressions that contains the named expression you want to use.
 - b. Click the down arrow to the right of the second Named Expressions drop-down list, and choose the named expression you want. As you choose a named expression, the regular expression definition of that named expression appears in the Preview Expression pane beneath the Named Expression list boxes.
 - c. Click Add Expression to add that named expression to the Expression list.
- Note: You should perform either this step or step 7, but not both.
8. If you want to create a new expression to define your policy, use the Expression Editor.
 - a. Click the Add button. The Add Expression dialog box appears.
 - b. In the Add Expression dialog box, choose the type of connection you want to filter. The Flow Type is set to REQUEST by default, which tells the NetScaler appliance to look at incoming connections, or requests. If you want to filter outgoing connections (responses), you click the right arrow beside the drop-down list and choose RES.
 - c. If the Protocol is not already set to HTTP, click the down arrow to the right of the Protocol drop-down list and choose HTTP.
Note: In the NetScaler classic expressions language, "HTTP" includes HTTPS requests, as well.
 - d. Click the down arrow to the right of the Qualifier drop-down list, and then choose a qualifier for your expression. Your choices are:

METHOD

The HTTP method used in the request.

URL

The contents of the URL header.

URLTOKENS

The URL tokens in the HTTP header.

VERSION

The HTTP version of the connection.

HEADER

The header portion of the HTTP request.

URLLEN

The length of the contents of the URL header.

URLQUERY

The query portion of the contents of the URL header.

URLQUERYLEN

The length of the query portion of the URL header.

The contents of the remaining list boxes change to the choices appropriate to the Qualifier you pick. For example, if you choose HEADER, a text field labeled Header Name* appears below the Flow Type list box.

- e. Click the down arrow to the right of the Operator drop-down list, and choose an operator for your expression. Your choices will vary depending on the Protocol you chose in the preceding step. The following list includes all of the operators:

==

Matches the following text string exactly.

!=

Does not exactly match the following text string.

>

Is greater than the following integer.

CONTAINS

Contains the following text string.

CONTENTS

The contents of the designated header, URL, or URL query.

EXISTS

The specified header or query exists.

NOTCONTAINS

Does not contain the following text string.
NOTEXISTS
The specified header or query does not exist.

- f. If the Value text box is visible, type the appropriate string or number. If you are testing a string in any way, type the string into the Value text box. If you are testing an integer in any way, type the integer into the Value text box.
 - g. If you chose HEADER as the Protocol, type the header you want in the Header Name* text box.
 - h. Click OK to add your expression to the Expressions list.
 - i. Repeat steps B through H to create any additional expressions you want for your profile.
 - j. Click Close to close the Expressions Editor.
9. If you created a new expression, in the Expression frame select an option from the Match Any Expression drop-down list. Your choices are:
- o Match Any Expression. If a request matches any expression in the Expressions list, the request matches this policy.
 - o Match All Expressions If a request matches all expressions in the Expressions list, the request matches this policy. If it does not match all of them, it does not match this policy.
 - o Tabular Expression Switches the Expressions list to a tabular format with three columns. In the first column you can place a BEGIN [() operator. The second column contains the expressions you have selected or created. In the third column, you can place any of the other operators in the following list, to create complex policy groups in which each group can be configured for match any expression or match all expressions.
 - o The AND [&&] operator tells the appliance to require that a request match both the current expression and the following expression.
 - o The OR [||] operator tells the appliance to require that a request match either the current expression or the following expression, or both. Only if the request does not match either expression does it not match the policy.
 - o The END [)] operator tells the appliance that this is the last expression in this expression group or policy.
Note: The Tabular format allows you to create a complex policy that contains both "Match Any Expression" and "Match All Expressions" on a per-expression basis. You are not limited to just one or the other.
 - o Advanced Free-Form Switches off the Expressions Editor entirely and modifies the Expressions list into a text area. In the text area, you can type the PCRE-format regular expression of your choice to define this policy. This is both the most powerful and the most difficult method of creating a policy, and is recommended only for those thoroughly familiar with the NetScaler appliance and PCRE-format regular expressions.
Caution: If you switch to Advanced Free Form expression editing mode, you cannot switch back to any of the other modes. Do not choose this expression editing mode unless you are sure that is what you want.
10. Repeat steps 6 through 8 to add any additional expressions you want to the Expressions list. You can mix named expressions and expressions created in the Expressions Editor. To the NetScaler appliance, they are all the same.
11. Click Create to create your new policy. Your new policy appears in the Policies pane list.
12. Click Close. To create additional Content Filtering policies, repeat the previous procedure. To remove a Content Filtering policy, select the policy in the Policies tab and click Remove.

Binding a Content Filtering Policy

You must bind each content filtering policy to put it into effect. You can bind policies globally or to a particular virtual server. Globally bound policies are evaluated each time traffic directed to any virtual server matches the policy. Policies bound to a specific vserver are evaluated only when that vserver receives traffic that matches the policy.

To bind a policy to a virtual server by using the command line interface

At the command prompt, type the following commands to bind a policy to a virtual server and verify the configuration:

- `bind lb vserver <name>@ -policyName <string> -priority <positive_integer>`
- `show lb vserver <name>`

Example

```
> bind lb vserver vs-loadbal -policyName policyTwo -priority 100
Done
> show lb vserver vs-loadbal
1)      vs-loadbal (10.102.29.20:80) - HTTP      Type: ADDRESS
      State: OUT OF SERVICE
      Last state change was at Wed Aug 19 09:05:47 2009 (+211 ms)
      Time since last state change: 2 days, 00:58:03.260
      Effective State: DOWN
      Client Idle Timeout: 180 sec
      Down state flush: ENABLED
      Disable Primary Vserver On Down : DISABLED
      Port Rewrite : DISABLED
      No. of Bound Services : 0 (Total)          0 (Active)
      Configured Method: LEASTCONNECTION
      Mode: IP
      Persistence: NONE
      Vserver IP and Port insertion: OFF
      Push: DISABLED Push VServer:
      Push Multi Clients: NO
      Push Label Rule: none
```

Done

To globally bind a policy by using the command line interface

At the command prompt, type the following commands to globally bind a policy and verify the configuration:

- `bind filter global (<policyName> [-priority <positive_integer>]) [-state (ENABLED | DISABLED)]`
- `show filter global`

Example

```
bind filter global cf-pol -priority 1
Done show filter global
1)      Policy Name: cf-pol      Priority: 1
Done
```

To bind a policy to a virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server to which you want to bind the content filtering policy from the list, and click Open.
3. In the Configure Virtual Server (Load Balancing) dialog box, select the Policies tab, and then select the check box in the Active column of the filter policy that you want to bind to the virtual server.
4. Click OK. The policies you have bound display a check mark and the word Yes in the Policies Bound column of the Policies tab.

To globally bind a policy by using the configuration utility

1. Navigate to Security > Protection Features > Filter.
2. In the details pane, in the Policies tab, select the policy that you want to bind, and then click Global Bindings.

3. In the Bind/Unbind Filter Policies dialog box, in the Policy Name drop-down list, select a policy, and then click Add. The policy is added to the Configured list.
Note: To select multiple policies from the list, press and hold the Ctrl key, then click each policy you want.
4. Click OK, and then click Close. The policies you have bound display a check mark and the word Yes in the Globally Bound column of the Policies tab.

Configuring Content Filtering for a Commonly Used Deployment Scenario

This example provides instructions for using the configuration utility to implement a content filtering policy in which, if a requested URL contains `root.exe` or `cmd.exe`, the content filtering policy `filter-CF-nimda` is evaluated and the connection is reset.

To configure this content filtering policy, you must do the following:

- Enable content filtering
- Configure content filtering policy
- Bind content filtering policy globally or to a virtual server
- Verify the configuration

Note: Since this example uses a default content filtering action, you do not need to create a separate content filtering action.

To enable content filtering

1. In the navigation pane, expand System, and click Settings.
2. In the details pane, under Modes & Features, click Change Basic Features.
3. In the Configure Basic Features dialog box, select the Content Filtering check box, and then click OK.
4. In the Enable/Disable feature(s) dialog box, click Yes. A message appears in the status bar, stating that the selected feature is enabled.

To configure the content filtering policy filter-CF-nimda

1. Navigate to Security > Protection Features > Filter.
2. In the details pane, click Add. The Create Filter Policy dialog box appears.
3. In the Create Filter Policy dialog box, in the Filter Name text box, type the name `filter-CF-nimda`.
4. Select the Request Action option, and in the drop-down list, select RESET.
5. In the Expression frame, select Match Any Expression from the drop-down list, and then click Add.
6. In the Add Expression dialog box, Expression Type drop-down list, select General.
7. In the Flow Type drop-down list, select REQ.
8. In the Protocol drop-down list, select HTTP.
9. In the Qualifier drop-down list, select URL.
10. In the Operator drop-down list, select CONTAINS.
11. In the Value text box, type `cmd.exe`, and then click OK. The expression is added in the Expression text box.
12. To create another expression, repeat Steps 7 through 11, but in the Value text box, type `root.exe`. Then click OK, and finally click Close.
13. Click Create on the Create Filter Policy dialog box. The filter policy `filter-CF-nimda` appears in the Filter list.
14. Click Close.

To globally bind the content filtering policy

1. Navigate to Security > Protection Features > Filter. The Filter page appears in the right pane.
2. In the details pane, Policies tab, select the policy that you want to bind and click Global Bindings. The Bind/Unbind Filter Policies dialog box appears.
3. In the Bind/Unbind Filter Policies dialog box, in the Policy Name drop-down list, select the policy `filter-CF-nimda`, and click Add. The policy is added to the Configured list.
4. Click OK, and then click Close. The policy you have bound displays a check mark and Yes in the Globally Bound column of the Policies tab.

To bind the content filtering policy to a virtual server

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane virtual servers list, select `vserver-CF-1` to which you want to bind the content filtering policy and click Open.
3. In the Configure Virtual Server (Load Balancing) dialog box, select the Policies tab.
4. In the Active column, select the check box for the policy `filter-CF-nimda`, and then click OK. Your content filtering policy is now active, and should be filtering requests. If it is functioning correctly, the Hits counter is incremented every time there is a request for a URL containing either `root.exe` or `cmd.exe`. This allows you to confirm that your content filtering policy is working. The content filtering policy is bound to the virtual server.

To verify the content filtering configuration by using the command line interface

At the command prompt, type the following command to verify the content filtering configuration:

show filter policy filter-CF-nimda

Example

```
sh filter policy filter-CF-nimda
  Name: filter-CF-nimda    Rule: REQ.HTTP.URL CONTAINS cmd.exe || REQ.HTTP.URL CONTAINS
  Request action: RESET
  Response action:
  Hits: 0
Done
```

Note: The Hits counter displays an integer that denotes the number of times the `filter-CF-nimda` policy is evaluated. In the preceding steps, the Hits counter is set to zero because no requests for a URL containing either `cmd.exe` or `root.exe` have been made yet. If you want to see the counter increment in real time, you can simply request a URL that contains either of these strings.

To verify the content filtering configuration by using the configuration utility

1. Navigate to Security > Protection Features > Filter.
2. In the details pane, select the filter policy `filter-CF-nimda`. The bottom of the pane should display the following:

Request Action:

RESET

Rule:

REQ.HTTP.URL CONTAINS cmd.exe || REQ.HTTP.URL CONTAINS root.exe

Hits:

0

Troubleshooting

If the content filtering feature does not work as expected after you have configured it, you can use some common tools to access NetScaler resources and diagnose the problem.

Resources for Troubleshooting

Updated: 2013-07-22

You can use the following tools and resources to troubleshoot most Content Filtering issues on a NetScaler appliance:

- The Wireshark application customized for the NetScaler trace files
- Trace files recorded when accessing the resource
- The configuration files
- The ns.log file
- The iehttpheaders, or a Fiddler trace or a similar utility

Troubleshooting Content Filtering Issues

Updated: 2013-08-02

To troubleshoot a content filtering issue, proceed as follows:

- Verify that the feature is enabled.
- Verify that the content filtering policy is configured correctly. Pay special attention to the expression that evaluates the incoming requests.
Note: Most content filtering issues are caused by incorrect configuration, and the error is most often in the policy configuration.
- Check the policy's Hits counter to verify that it is incrementing. If it is not, the policy is not getting evaluated.
- If the policy is getting evaluated and the required filtering is still not performed, you need to look into the policy expressions and action.
- If the policy's expression seems valid, test it by assigning a simple NSTRUE value to see if the evaluation of the expression is creating any issue.
- Reevaluate whether the filtering should be based on the request or the response.
- Verify that the action is configured correctly. For example, if a custom action is used to corrupt a header in the request, verify that the header name in the action is correct. If you are not sure about the header name, start a browser with iehttpheaders or a similar utility, and then verify the headers in the request. When this feature is used, you can use nstrace to find out if appropriate action is performed when the packets leave NetScaler appliance.
- An iehttpheaders or Fiddler trace can help you find header options and names, client-side request headers, and response headers recorded on the client.
- To check the modifications made to the request header, record an nstrace on the NetScaler appliance or a Wireshark trace on the server.
- If none of the above measures resolves the issue, verify that the connection has not become untrackable, which can happen in certain circumstances. If a connection becomes untrackable, the appliance does not perform any application-level processing of the requests. In that event, contact Citrix Technical Support.

HTTP Denial-of-Service Protection

Internet hackers can bring down a site by sending a surge of GET requests or other HTTP-level requests. HTTP Denial-of-Service (HTTP DoS) Protection provides an effective way to prevent such attacks from being relayed to your protected Web servers. The HTTP DoS feature also ensures that a NetScaler appliance located between the internet cloud and your Web servers is not brought down by an HTTP DoS attack.

Most attackers on the Internet use applications that discard responses to reduce computation costs, and minimize their size to avoid detection. The attackers focus on speed, devising ways to send attack packets, establish connections or send HTTP requests as rapidly as possible.

Real HTTP clients such as Internet Explorer, Firefox, or NetScape browsers can understand HTML Refresh meta tags, Java scripts, and cookies. In standard HTTP the clients have most of these features enabled. However, the dummy clients used in DoS attacks cannot parse the response from the server. If malicious clients attempt to parse and send requests intelligently, it becomes difficult for them to launch the attack aggressively.

When the NetScaler appliance detects an attack, it responds to a percentage of incoming requests with a Java or HTML script containing a simple refresh and cookie. (You configure that percentage by setting the Client Detect Rate parameter.) Real Web browsers and other Web-based client programs can parse this response and then resend a POST request with the cookie. DoS clients drop the NetScaler appliance's response instead of parsing it, and their requests are therefore dropped as well.

Even when a legitimate client responds correctly to the NetScaler appliance's refresh response, the cookie in the client's POST request may become invalid in the following conditions:

- If the original request was made before the NetScaler appliance detected the DoS attack, but the resent request was made after the appliance had come under attack.
- When the client's think time exceeds four minutes, after which the cookie becomes invalid.

Both of these scenarios are rare, but not impossible. In addition, the HTTP DoS protection feature has the following limitations:

- Under an attack, all POST requests are dropped, and an error page with a cookie is sent.
- Under an attack, all embedded objects without a cookie are dropped, and an error page with a cookie is sent.

The HTTP DoS protection feature may affect other NetScaler features. Using DoS protection for a particular content switching policy, however, creates additional overhead because the policy engine must find the policy to be matched. There is some overhead for SSL requests due to SSL decryption of the encrypted data. Because most attacks are not on a secure network, though, the attack is less aggressive.

If you have implemented priority queuing, while it is under attack a NetScaler appliance places requests without proper cookies in a low-priority queue. Although this creates overhead, it protects your Web servers from false clients. HTTP DoS protection typically has minimal effect on throughput, since the test JavaScript is sent for a small percentage of requests only. The latency of requests is increased, because the client must re-issue the request after it receives the JavaScript. These requests are also queued.

To implement HTTP DoS protection, you enable the feature and define a policy for applying this feature. Then you configure your services with the settings required for HTTP DoS. You also bind a TCP monitor to each service and bind your policy to each service to put it into effect.

Layer 3-4 SYN Denial-of-Service Protection

Any NetScaler appliance with system software version 8.1 or later automatically provides protection against SYN DoS attacks.

To mount such an attack, a hacker initiates a large number of TCP connections but does not respond to the SYN-ACK messages sent by the victimized server. The source IP addresses in the SYN messages received by the server are typically spoofed. Because new SYN messages arrive before the half-open connections initiated by previous SYN messages time out, the number of such connections increases until the server no longer has enough memory available to accept new connections. In extreme cases, the system memory stack can overflow.

A NetScaler appliance defends against SYN flood attacks by using SYN cookies instead of maintaining half-open connections on the system memory stack. The appliance sends a cookie to each client that requests a TCP connection, but it does not maintain the states of half-open connections. Instead, the appliance allocates system memory for a connection only upon receiving the final ACK packet, or, for HTTP traffic, upon receiving an HTTP request. This prevents SYN attacks and allows normal TCP communications with legitimate clients to continue uninterrupted.

SYN DoS protection on the NetScaler appliance ensures the following:

- The memory of the NetScaler is not wasted on false SYN packets. Instead, memory is used to serve legitimate clients.
- Normal TCP communications with legitimate clients continue uninterrupted, even when the Web site is under SYN flood attack.

In addition, because the NetScaler appliance allocates memory for HTTP connection state only after it receives an HTTP request, it protects Web sites from idle connection attacks.

SYN DoS protection on your NetScaler appliance requires no external configuration. It is enabled by default.

Enabling HTTP DoS Protection

To configure HTTP DoS protection, you must first enable the feature.

To enable HTTP DoS protection by using the command line interface

At the command prompt, type the following commands to enable HTTP DoS protection and verify the configuration:

- enable ns feature HttpDoSProtection
- show ns feature

Example

```
> enable ns feature HttpDoSProtection
Done
> show ns feature
```

	Feature -----	Acronym -----	Status -----
1)	Web Logging	WL	ON
2)	Surge Protection	SP	OFF
.			
.			
10)	Global Server Load Balancing	GSLB	ON
11)	Http DoS Protection	HDOSP	ON
12)	Content Filtering	CF	ON
.			
.			
23)	HTML Injection	HTMLInjection	ON
24)	NetScaler Push	push	OFF

```
Done
>
```

To enable HTTP DoS protection by using the configuration utility

1. Navigate to System > Settings.
2. In the details pane, click Configure Advanced Features.
3. In the Configure Advanced Features dialog box, select the HTTP DoS Protection check box.
4. Click OK.

Defining an HTTP DoS Policy

After you enable HTTP DoS protection, you next create a policy.

Note: Before changing the default setting for Client Detect Rate, see ["Tuning the Client Detection/JavaScript Challenge Response Rate."](#)

To configure a HTTP DoS policy by using the command line interface

At the command prompt, type one of the following commands to configure an HTTP DoS policy and verify the configuration:

- add dos policy <name> -qDepth <positive_integer> [-cltDetectRate <positive_integer>]
- set dos policy <name> -qDepth <positive_integer> [-cltDetectRate <positive_integer>]

Example

```
> add dos policy pol-HTTP-DoS -qDepth 30
Done
> set dos policy pol-HTTP-DoS -qDepth 40
Done
> show dos policy
1)      Policy: pol-HTTP-DoS      QDepth: 40
Done
>
```

To configure an HTTP DoS policy by using the configuration utility

1. Navigate to Security > Protection Features > HTTP DoS.
2. In the details pane, do one of the following:
 - To create a new policy, click Add.
 - To modify an existing policy, select the policy, and then click Open.
3. In the Create HTTP DoS Policy or Configure HTTP DoS Policy dialog box, specify values for the parameters:
 - Name*â€™name (You cannot change the name of an existing policy.)
 - QDepth*â€™qdepth
 - Client Detect Rate*â€™cltDetectRate (Before changing the default setting for cltDetectRate, see "[Tuning the Client Detection/JavaScript Challenge Response Rate."](#))
4. Click OK to create your new policy. The policy that you created appears in the details pane, and the status bar displays a message indicating that the DoS policy is successfully configured.

Configuring an HTTP DoS Service

After you configure an HTTP DoS policy, you must configure a service for your policy. The service accepts HTTP traffic that is protected by the HTTP DoS policy.

To configure an HTTP DoS service by using the command line interface

At the command prompt, type one of the following commands to configure an HTTP DoS service and verify the configuration:

- add service <name>@ (<IP>@ | <serverName>@) HTTP <port> [-maxClient <positive_integer>] [-maxReq <positive_integer>] -state ENABLED
- set service <name>@ (<IP>@ | <serverName>@) HTTP <port> [-maxClient <positive_integer>] [-maxReq <positive_integer>] -state ENABLED

Example

```
> add service ser-HTTP-Dos1 10.102.29.40 HTTP 87
Done
> set service ser-HTTP-Dos1 -maxReq 20
Done
> show service
1)      srv-http-10 (10.102.29.30:80) - HTTP
      State: DOWN
      Last state change was at Wed Jul  8 07:49:52 2009
      Time since last state change: 34 days, 00:48:18.700
      Server Name: 10.102.29.30
      Server ID : 0   Monitor Threshold : 0
      Max Conn: 0     Max Req: 0         Max Bandwidth: 0 kbits
      Use Source IP: NO
      Client Keepalive(CKA): NO
      Access Down Service: NO
      TCP Buffering(TCPB): NO
      HTTP Compression(CMP): NO
      Idle timeout: Client: 180 sec   Server: 360 sec
      Client IP: DISABLED
      Cacheable: NO
      SC: OFF
      SP: OFF
      Down state flush: ENABLED
      .
      .
      .

5)      ser-HTTP-Dos1 (10.102.29.40:87) - HTTP
      State: DOWN
      Last state change was at Tue Aug 11 08:23:40 2009
      Time since last state change: 0 days, 00:14:30.300
      Server Name: 10.102.29.40
      Server ID : 0   Monitor Threshold : 0
      Max Conn: 0     Max Req: 20        Max Bandwidth: 0 kbits
      Use Source IP: NO
      Client Keepalive(CKA): NO
      Access Down Service: NO
      TCP Buffering(TCPB): NO
      HTTP Compression(CMP): YES
      Idle timeout: Client: 180 sec   Server: 360 sec
      Client IP: DISABLED
      Cacheable: NO
      SC: OFF
      SP: OFF
      Down state flush: ENABLED

Done
>
```

To configure an HTTP DoS service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services.
2. In the details pane, do one of the following:

- To create a new service, click Add.
 - To modify an existing service, select the service, and then click Open.
3. In the Create Server or Configure Server dialog box, specify values for the following parameters, which correspond to the descriptions in "Parameters for configuring an HTTP DoS service" as follows (asterisk indicates a required parameter):
- Service Name*â€"name (You cannot change the name of an existing service.)
 - Server*â€"IP or serverName (Specify one or the other, not both.)
 - Port*â€"port
4. If the Enable Service check box is not selected, select it.
5. Select the Advanced tab, and select the Override Global check box to enable those choices.
6. Specify values for the following parameters.
- Max Clients*â€"maxClient
 - Max Requests*â€"maxReq
7. Click Create or OK, and then click Close. The service appears in the list of services.

Binding an HTTP DoS Monitor and Policy

To put HTTP DoS protection into effect after you have configured an HTTP DoS service, you must bind the monitor, and then bind the service to the HTTP DoS policy.

To bind the monitor to the service by using the command line interface

At the command prompt, type the following commands to bind the monitor to the service and verify the configuration:

- bind lb monitor <monitorName> <serviceName>
- show lb monitor

Example

```
> bind lb monitor tcp ser-HTTP-DoS
Done
> show lb monitor
1)  Name.....:  ping-default  Type.....:      PING    State....ENABLED
2)  Name.....:  tcp-default   Type.....:      TCP     State....ENABLED
3)  Name.....:  ping          Type.....:      PING    State....ENABLED
4)  Name.....:  tcp           Type.....:      TCP     State....ENABLED
5)  Name.....:  http          Type.....:      HTTP    State....ENABLED
.
.
.
17) Name.....:  ldns-dns      Type.....:  LDNS-DNS  State....ENABLED
Done
```

To bind the policy to the service by using the command line interface

At the command prompt, type the following commands to bind the policy to the service and verify the configuration:

bind service <serviceName> -policyName <policyname>

Example

```
> bind service ser-HTTP-DoS -policyName pol-HTTP-DoS
Done
> show service
1)  srv-http-10 (10.102.29.30:80) - HTTP
    State: DOWN
    Last state change was at Wed Jul  8 07:49:52 2009
    Time since last state change: 34 days, 01:24:58.510
    Server Name: 10.102.29.30
    Server ID : 0   Monitor Threshold : 0
    Max Conn: 0    Max Req: 0         Max Bandwidth: 0 kbits
    Use Source IP: NO
    Client Keepalive(CKA): NO
    Access Down Service: NO
    TCP Buffering(TCPB): NO
    HTTP Compression(CMP): NO
    Idle timeout: Client: 180 sec   Server: 360 sec
    Client IP: DISABLED
    Cacheable: NO
    SC: OFF
    SP: ON
    Down state flush: ENABLED
.
.
.
4)  ser-HTTP-Dos (10.102.29.18:88) - HTTP
    State: DOWN
    Last state change was at Tue Aug 11 08:19:45 2009
    Time since last state change: 0 days, 00:55:05.40
    Server Name: 10.102.29.18
    Server ID : 0   Monitor Threshold : 0
    Max Conn: 0    Max Req: 0         Max Bandwidth: 0 kbits
    Use Source IP: NO
    Client Keepalive(CKA): NO
    Access Down Service: NO
```



```

TCP Buffering(TCPB): NO
HTTP Compression(CMP): YES
Idle timeout: Client: 180 sec    Server: 360 sec
Client IP: DISABLED
Cacheable: NO
SC: OFF
SP: ON
Down state flush: ENABLED
5) ser-HTTP-Dos1 (10.102.29.40:87) - HTTP
State: DOWN
Last state change was at Tue Aug 11 08:23:40 2009
Time since last state change: 0 days, 00:51:10.110
Server Name: 10.102.29.40
Server ID : 0    Monitor Threshold : 0
Max Conn: 0      Max Req: 20      Max Bandwidth: 0 kbits
Use Source IP: NO
Client Keepalive(CKA): NO
Access Down Service: NO
TCP Buffering(TCPB): NO
HTTP Compression(CMP): YES
Idle timeout: Client: 180 sec    Server: 360 sec
Client IP: DISABLED
Cacheable: NO
SC: OFF
SP: OFF
Down state flush: ENABLED
Done
>

```

To bind the monitor and policy to the service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services.
2. In the details pane, select the service that you want to bind, and then click Open.
3. In the Configure Service dialog box, select the Monitor tab, click the name of the monitor you want in the Monitors list, and then click Add. The selected monitor is added to the Configured frame.
4. Select the Policies tab, then select the HTTP DoS tab.
5. Select a policy from the Available Policies list, and then click Add. The policy appears in the Configured Policies list.
6. Click OK, and then click Close. A message appears in the status bar, stating that the service has been configured.

Tuning the Client Detection/JavaScript Challenge Response Rate

After you have enabled and configured HTTP DoS protection, if more than the maximum specified number of clients are waiting in the NetScaler surge queue for the HTTP DoS service, the HTTP DoS protection function is triggered. The default rate of challenged JavaScript responses sent to the client is one percent of the server response rate. The default response rate is inadequate in many real attack scenarios, however, and may need to be tuned.

For example, assume that the Web server is capable of a maximum of 500 responses/sec, but is receiving 10,000 Gets/sec. If 1% of the server responses are sent as JavaScript challenges, responses are reduced to almost none: 5 client (500 * 0.01) JavaScript responses, for 10000 waiting client requests. Only about 0.05% of the real clients receive JavaScript challenge responses. However, if the client detection/JavaScript challenge response rate is very high (for example, 10%, generating 1000 challenge JavaScript responses per second), it may saturate the upstream links or harm the upstream network devices. Exercise care when modifying the default **Client Detect Rate** value.

If the configured triggering surge queue depth is, for example, 200, and the surge queue size is toggling between 199 and 200, the NetScaler toggles between the "attack" and "no-attack" modes, which is not desirable. The HTTP DoS feature includes a window mechanism is provided. When the surge queue size reaches the designated queue depth value, triggering "attack" mode, the surge queue size must fall for the NetScaler to enter "no-attack" mode. In the scenario just described, if the value of WINDOW_SIZE is set to 20, the surge queue size must fall below 180 before the NetScaler enters "no-attack" mode. During configuration, you must specify a value more than the WINDOW_SIZE for the **QDepth** parameter when adding a DoS policy or setting a DoS policy.

The triggering surge queue depth should be configured on the basis of previous observations of traffic characteristics. For more information about setting up a correct configuration, see "[Guidelines for HTTP DoS Protection Deployment](#)."

Guidelines for HTTP DoS Protection Deployment

Citrix recommends you to deploy the HTTP DoS protection feature in a tested and planned manner and closely monitor its performance after the initial deployment. Use the following information to fine-tune the deployment of HTTP DoS Protection.

- The maximum number of concurrent connections supported by your servers.
- The average and normal values of the concurrent connections supported by your servers.
- The maximum output rate (responses/sec) that your server can generate.
- The average traffic that your server handles.
- The typical bandwidth of your network.
- The maximum bandwidth available upstream.
- The limits affecting bandwidth (such as external links, a particular router, or other critical devices on the path that may suffer from a traffic surge).
- Whether allowing a greater number of clients to connect is more important than protecting upstream network devices.

To determine the characteristics of a HTTP DoS attack, you should consider the following issues.

- What is the rate of incoming fake requests that you have experienced in the past?
- What types of requests have you received (complete posts, incomplete gets)?
- Did previous attacks saturate your downstream links? If not, what was the bandwidth?
- What types of source IP addresses and source ports did the HTTP requests have (e.g., IP addresses from one subnet, constant IP, ports increasing by one).
- What types of attacks do you expect in future? What type have you seen in the past?
- Any or all information that can help you tune DoS attack protection.

Priority Queuing

The priority queuing feature lets you filter incoming HTTP traffic on the basis of categories that you create and define, and prioritize those HTTP requests accordingly. Priority queuing directs high-priority requests to the server ahead of low-priority requests, so that users who need resources for important business uses receive expedited access to your protected Web servers.

Note: The priority queuing feature is not supported in NetScaler 9.2 nCore.

To implement priority queuing, you create priority queuing policies that specify a priority, weight, threshold, and implicit action. When an incoming request matches a priority queuing policy, the request is processed as the associated action indicates. For example, you can create a priority queuing policy that places all matching requests above a certain threshold in a surge queue, while giving priority treatment to other requests.

You can bind up to three priority queuing policies to a single load balancing virtual server. The priority levels are:

Level 1

A Level 1 policy processes priority requests.

Level 2

A Level 2 policy processes requests that should receive responses as soon as Level 1 requests have been cleared from the queue.

Level 3

A Level 3 policy processes non-priority requests that receive responses only after requests in the first two queues have been cleared.

You can use weighted queuing to adjust the relative priority of each of these queues. Weights can range from 0 to 101. A weight of 101 tells the NetScaler appliance to clear all requests in that queue before forwarding any requests in the lower-priority queues to the Web server. A weight of 0 tells the appliance to send requests in that queue to the Web server only when there are no requests waiting in any of the other queues.

You must assign a unique name to each priority queuing policy. Policy names can be up to 127 characters. Multiple policies bound to the same load balancing virtual server cannot have the same priority level. No two virtual servers that have one or more common underlying physical services can have priority queuing configured or enabled on both virtual servers simultaneously.

To configure priority queuing the NetScaler, you perform the following steps:

- Enable the load balancing feature
- Define a server and service
- Define a load balancing virtual server
- Bind the service to the load balancing virtual server
- Enable the priority queuing feature
- Create the priority queuing policies
- Bind the priority queuing policies to the load balancing virtual server
- Enable priority queuing on load balancing virtual server

For information about enabling load balancing, creating servers, creating virtual servers and services, and binding these servers and services, see the *Citrix NetScaler Traffic Management Guide* at <http://support.citrix.com/article/CTX132359>. For complete information about policies and expressions, see the *Citrix NetScaler Policy Configuration and Reference Guide* at <http://support.citrix.com/article/CTX132362>.

Enabling Priority Queuing

To use the priority queuing feature the NetScaler appliance, you must first enable it.

To enable priority queuing by using the command line interface

At the command prompt, type the following commands to enable priority queuing and verify the configuration:

- enable ns feature PriorityQueuing
- show ns feature

Example

```
> enable ns feature PriorityQueuing
Done
> show ns feature
```

	Feature	Acronym	Status
	-----	-----	-----
1)	Web Logging	WL	ON
2)	Surge Protection	SP	OFF
3)	Load Balancing	LB	ON
.			
.			
.			
8)	Priority Queuing	PQ	ON
.			
23)	HTML Injection	HTMLInjection	ON
24)	NetScaler Push	push	OFF
Done			

To enable priority queuing by using the configuration utility

1. Navigate to System > Settings.
2. In the details pane, click Configure advanced features.
3. In the Configure Advanced Features dialog box, select the Priority Queuing check box.
4. Click OK.

Configuring a Priority Queuing Policy

To configure a priority queuing policy, you can use either the configuration utility or the command line.

Note: For more information about using the command line, see ["Command Reference."](#)

To configure a priority queuing policy by using the command line interface

At the command prompt, type the following command to configure a priority queuing policy and verify the configuration:

```
add pq policy <policyName> -rule <expression> -priority <positive_integer> [-weight <positive_integer>] [-qDepth <positive_integer>] [-polqDepth <positive_integer>]
```

Example

```
> add pq policy pol_cgibin -rule "URL == '/cgi-bin/'" -priority 1
Done
> show pq policy pol_cgibin
1)      Policy: pol_cgibin      Rule: URL == '/cgi-bin/'      Priority: 1      Weight: 10
      Hits: 0
Done
```

To configure a priority queuing policy by using the configuration utility

1. Navigate to Security > Protection Features > Priority Queuing.
2. In the details pane, do one of the following:
 - o To create a new policy, click Add.
 - o To modify an existing policy, select the policy, and then click Open.
3. If you are creating a new policy, in the Create PQ Policy dialog box, in the Name text box, type a name for your new policy.

The name can consist of from one to 127 letters, numbers and the hyphen and underscore symbol.

If you are modifying an existing policy, skip this step. You cannot change the name of an existing policy.

4. In the Rule text box, either enter the policy expression directly, or click New to create a policy expression. If you click New, perform the following steps:
 - a. In the Create Expression dialog box, click Add.
 - b. In the Add Expression dialog box, leave Expression Type set to General, and in the Flow Type drop-down list select a Flow Type. Your choices are REQ (for requests) and RES (for responses).
 - c. In the Protocol drop-down list, select a protocol. If you selected REQ in the previous step, your choices are HTTP (Web-based connections), SSL (secure Web connections), TCP and IP. If you selected RES in the previous step, your choices are HTTP, TCP and IP.
 - d. In the Qualifier drop-down list, select a qualifier.

Your choices depend upon your selections in the previous step. Common choices are HTTP VERSION (the version of the HTTP connection), HTTP HEADER (the specified HTTP header), TCP SOURCEPORT/ DESTPORT (the source or destination port of a TCP connection), and IP SOURCEIP/DESTIP (the source or destination IP of the connection).

If you choose HTTP HEADER, the Header text box appears beneath the original row of text boxes. You fill in the name of the HTTP header you want.

For a complete description of the available choices, see ["Policies and Expressions."](#)

- e. In the Operator drop-down list, select an operator.

For a complete description of the available choices, see ["Policies and Expressions."](#)

- f. In the Value text box, type the value you want to test for.

This may be a text string or a number, depending upon the context. For a complete description of values appropriate to the specific context, see ["Policies and Expressions."](#)

- g. Click OK. The expression is added in the Expression text box.
 - h. Click Create. The expression appears in the Rule text box.
5. In the Priority and Weight text boxes, type numeric values, for example, 1 and 30. For more information about Priority and Weight, see ["Setting Up Weighted Queuing."](#)
 6. Enter a numeric value for either Queue Depth or Policy Queue Depth, for example 234, and click Create.
 - o Queue Depth Defines the total number of waiting clients or requests on the virtual server to which the policy is bound.
 - o Policy Queue Depth Defines the total number of waiting clients or requests belonging to the policy.

The policy is created and appears in the Priority Queuing page.

Note: To create additional priority queuing policies, repeat the procedure in the preceding section, and click Close after you finish.

Binding a Priority Queuing Policy

After you create a priority queuing policy, you must bind it to the appropriate virtual server to put it into effect.

To bind a priority queuing policy by using the command line interface

At the command prompt, type the following commands to bind a policy and verify the configuration:

- o bind lb vserver <name> -policyName <policyname>
- o show lb vserver <name>

Example

```
> bind lb vserver lbvip -policyname pol_cgibin
Done
> show lb vserver lbvip
    lbvip (8.7.6.6:80) - HTTP          Type: ADDRESS
    State: DOWN
    Last state change was at Wed Jul 15 05:54:24 2009 (+782 ms)
    Time since last state change: 26 days, 05:44:37.370
    Effective State: DOWN
    Client Idle Timeout: 180 sec
    Down state flush: ENABLED
    Disable Primary Vserver On Down : DISABLED
    Port Rewrite : DISABLED
    No. of Bound Services :    0 (Total)          0 (Active)
    Configured Method: LEASTCONNECTION
    Mode: IP
    Persistence: NONE
    Vserver IP and Port insertion: OFF
    Push: DISABLED  Push VServer:
    Push Multi Clients: NO
    Push Label Rule: none

1)      Policy : ns_cmp_msapp Priority:0

1)      Priority Queuing Policy : pol_cgibin
Done
>
```

To bind a priority queuing policy by using the configuration utility

1. In the navigation pane, locate and select the virtual server to which you want to bind the priority queuing policy.
 - o To select a load balancing virtual server, expand Traffic Management > Load Balancing > Virtual Servers, then select the load balancing virtual server that you want..
 - o To select a content switching virtual server, expand Traffic Management > Content Switching > Virtual Servers, then select the content switching virtual server that you want..
2. In the Configure Virtual Server dialog box, select the Policies tab.
3. Click the double right-arrow (») symbol to display the complete list of policy types, and then select Priority Queuing from the drop-down list.
4. Click Insert Policy.
5. In the Policy Name row, select the policy that you want to bind from the drop-down list.
6. Click OK to save your changes.

Setting Up Weighted Queuing

When priority queuing is implemented, lower-priority requests are typically kept on hold while higher-priority requests are served. The lower-priority requests may therefore be delayed if there is a constant flow of higher-priority requests.

To prevent delays for low-priority requests across multiple priority levels, you can configure weighted queuing for serving requests. The default weights for the priorities are:

- o Gold - Priority 1 - Weight 3
- o Silver - Priority 2 - Weight 2
- o Bronze - Priority 3 - Weight 1

You assign the minimum weight, zero (0), to requests that the NetScaler appliance should send to the server only if no requests are stored in any of the other queues. You assign the maximum weight, 101, to requests that the appliance should send to the server immediately, ahead of any requests stored in any of the other queues. Weights between these two set the relative priority of a particular queue in relation to the other queues. Queues with a higher weight are processed first; queues with a lower weight after the others have been processed. To assign the weights, see "[Configuring a Priority Queuing Policy](#)."

Note: The weight assigned to a higher-priority queue must be larger than the weight assigned to a lower-priority queue. For example, the weight assigned to The Gold (Priority 1) queue must be greater than the weight assigned to the Silver (Priority 2) queue.

SureConnect

You can use the SureConnect feature of the Citrix NetScaler appliance to service all incoming connections with either the requested content or a custom Web page that displays information about a delay in the request being serviced.

When servers are overloaded with the requests, the servers might either respond slowly or not at all. The SureConnect feature enables the NetScaler appliance to detect and compensate such conditions by ensuring that every client request gets serviced in some way, such as either a custom Web page or actual content is sent to the client.

SureConnect is activated when the response time or maximum server connections to a client request exceeds a limit that you have set. The SureConnect browser window displays one of the following:

- A progress bar with the amount of time remaining until the requested content will be available.
- Alternate Web content of your choice (alternate page).
- Both a progress bar and alternate page.
- Complete custom content of your choice.

You can configure whether the SureConnect progress bar alone is displayed or both the progress bar and the alternate page are displayed.

When the server becomes responsive again, the original request for content is served. If the user chooses, the alternate content window can remain in focus.

Subsequent requests from the same user within the same session are served immediately. This can be configured using the settings described later in this section.

SureConnect can be activated when a response is delayed, and when the number of user connections to a given URL exceeds a specified threshold.

SureConnect works with all standard browsers, including Microsoft Internet Explorer, Netscape Navigator, and Mozilla Firefox.

SureConnect is advantageous in the following situations:

◦ Full server queue

The server can respond fast, but there are too many users. This results in the server's queue being full and unable to process additional client requests.

SureConnect Solution: In this situation, the SureConnect window is displayed, showing the time left until the content will be available. The alternate page is displayed under the progress bar, if an alternate page has been configured.

◦ Large response delay

The server response is slow. Typically, if a Web server does not respond to a client request quickly, the user will leave the site.

SureConnect Solution: When the predicted delay reaches a configured time threshold, the SureConnect window displays the progress bar and the optional alternate page in the client browser.

◦ Client time-out

When the client requests content from a very slow Web site, a time-out message displays in the client browser, and the content is not delivered. The user may leave the site.

SureConnect Solution: The appliance stores the request until the server is no longer busy and delivers the requested content to the client.

◦ Server experiencing a traffic surge

The server typically responds quickly, but the current load of open connections is greater than the server capacity to serve them. Therefore, the server response is delayed.

SureConnect Solution: A SureConnect window is displayed in the client browser, showing the time left. The alternate page from the server is also displayed if it has been configured.

Installing SureConnect

SureConnect files must be installed on the alternate content server, which can be the same as the primary server.

On a Windows server, extract the sc_xx.exe file (where xx is the build number), or on a UNIX server, extract the sc_xx.tar file (where xx is the build number).

Note: You must install SureConnect in the default Web root directory.

If the alternate content server is the same as the primary server, place the SureConnect and alternate content files in any directory under the Web root directory. Specify this path when you add a policy to configure SureConnect. By default, SureConnect files are installed in the /Citrix NetScaler appliance directory under the default Web root directory.

If the alternate content server is different than the primary server, the SureConnect and alternate content files must be in a unique directory under the Web root directory. By default, this unique directory is the /Citrix NetScaler system directory. Specify this path when you add a policy to configure SureConnect.

The following files are extracted:

- Alternate content files (progressbar.htm, alternatepage.htm, and barandpage.htm)
- System-Logo.gif
- Customer-Logo.gif
- Sample.gif
- README.txt.

Installing on UNIX

This section describes how to install SureConnect alternate content on a UNIX server. The following are the prerequisites:

- The UNIX server is running the Apache server.
- The shell with the # prompt is in use.
- Apache is installed in the default location.
- The sc_xx.tar file is downloaded from the organization's Web site into the /var/ftp/incoming directory.

To install SureConnect

1. At the command prompt, navigate to the htdocs directory:

```
cd /usr/local/apache/htdocs
```

2. Type the following command:

```
tar xvpf/var/ftp/incoming/sc_xx.tar
```

The output from the .tar file is displayed. A /Citrix NetScaler system directory is created under the specified path and the SureConnect files are installed.

Installing on Windows

Updated: 2013-09-03

This section describes how to install SureConnect alternate content on a Windows server. The following are the prerequisites

- The server is running the Microsoft Internet Information Server.
- The DOS prompt is being used.
- The SureConnect zip (self-extracting) file is downloaded from the organization Web site using FTP into the C:\inetpub\wwwroot directory.

To install SureConnect on Windows

Do one of the following:

- At the command prompt, navigate to the wwwroot directory:

```
cd c:\inetpub\wwwroot
```

- Type the name of the executable file:

sc_xx.exe

- o Double-click the sc_xx.exe icon from the Microsoft Windows Explorer Web browser, extract from the compressed file into the default path (for example, the c:\inetpub\wwwroot directory).

Output from the zip file is displayed. A /Citrix NetScaler system directory is created under the specified path, and the SureConnect files are installed.

Configuring SureConnect

The following topics describe how to configure SureConnect for scenarios involving alternate server failure.

- "Configuring the Response for Alternate Server Failure"
- "Configuring the SureConnect Policies"
- "Customizing the Alternate Content File"
- "Configuring SureConnect for Citrix NetScaler Features"

Configuring the Response for Alternate Server Failure

Updated: 2013-09-03

If the alternate server fails, and the primary server cannot immediately deliver the requested content to the client, SureConnect does not display alternate content from the failed alternate server in the client Web browser.

The Citrix NetScaler appliance automatically sends a response to the client browser. You can customize the server response to display information suited to your needs.

The default response is:

Your Request is being processedâ€¦ Estimated Time: ____ Secs

Customizing the Default Response

The NetScaler appliance automatically sends the response to the client if the alternate server fails, or if the appliance is configured to send the default response.

To customize the default response of the appliance, create a vsr.htm file (a sample is provided in this section) as follows:

- The file can contain any valid HTML statements other than embedded objects.
- The file size cannot exceed 800 bytes.
- The file must reside on the NetScaler appliance. If you have a high availability (HA) setup, the file must reside on the primary and secondary nodes. Any changes made to the file on the primary node must also be applied to the file on the secondary node.
- Put vsr.htm file in the /etc directory.

To customize the default response

Change any of the contents between the </HEAD> and </HTML> tags in the vsr.htm file. Following is the sample content from vsr.htm file. The sections that you can edit are in bold text.

```
HTTP/1.1 200 OK
Server: NS_WS3.0
Content-Type: text/html
Cache-control: no-cache
Pragma: no-cache
Set-Cookie: NSC_BPIP=@@SID@@; path=/
<HTML> <HEAD> <META HTTP-EQUIV="Refresh" CONTENT="0">
</HEAD> <font color=blue size=5>Your request is being processed...
<br>Estimated Delay: @@DELAY@@ Sec </font> </HTML>
```

Note: Include @@DELAY@@ to display the predicted delayed response time in seconds.

SureConnect with In-Memory response (NS action)

Updated: 2013-09-03

When defining the SureConnect policy by using the add sc policy command, you can configure the NetScaler Appliance to serve alternative content to the client.

To enable SureConnect and configure the in-memory response, perform the following tasks:

- Enable the SureConnect feature on the appliance by using the enable feature SC command
- Define the services by using the add service <servicename> <IP address> <servicetype> <port> command. This identifies the original server for which the SureConnect is configured and the types of services.

- o Add a SureConnect policy by using the `add sc policy` command. You can configure a URL-based policy or a rule-based policy. The incoming requests are validated against the URL or rule you specify in the policy

Note: You can configure the SureConnect feature on a load balancing virtual server. In that case, perform the following additional actions:

- o Enable Load Balancing by using the `enable feature LB` command.
- o Enable SureConnect feature on the virtual server by using the `set lb vserver <vservname> -sc ON` command.
- o Bind services to the virtual server by using the `bind lb vserver <name> <serviceName>` command.
- o Bind policies to the virtual server by using the `bind lb vserver <name> -policyname <name>` command.

The following example illustrates how to configure SureConnect for the load balancing feature so that SureConnect will display alternative content from the NetScaler appliance.

In this example, two physical servers, with IP addresses, 10.101.3.187 and 10.101.3.188 are load balanced by the NetScaler appliance. The appliance has one configured virtual server, vs-NSact, whose IP address is 10.101.3.201. The file that contains the alternative content is vsr.htm. It is copied from the file system into system memory. Services are loaded until the SureConnect policy triggers, and the appliance supplies the alternate content.

```
enable feature SC LB
add service psvc1 10.101.3.187 http 80
add service psvc2 10.101.3.188 http 80
add lb vserver vs-NSact HTTP 10.101.3.201 80
bind lb vserver vs-NSact psvc1
bind lb vserver vs-NSact psvc2
add sc policy policyNS -url /cgi-bin/*.cgi -delay 400000
-action NS
set sc parameter -vsr /nsconfig/ssl/vsr.htm
bind lb vserver vs-NSact -policyName policyNS
set lb vserver vs-NSact -sc ON
save config
```

Table 1. Parameter values used in this example

Service	Â
Name	psvc1, psvc2
Server	10.101.3.187, 10.101.3.188
Protocol	HTTP
Port	80
Load Balancing Virtual Server	Â
Name	vs-NSact
IP Address	10.101.3.201
Protocol	HTTP
Port	80
SureConnect Policy	Â
Name	policyNS
URL	/cgi-bin/*.cgi
Delay(microseconds)	400000
SC Parameter	Â
VSR File Name	vsr.htm

To configure this example by using the configuration utility

1. In the In the navigation pane, navigate to System > Settings. In the Modes and Features pane, perform the following actions:
 - a. Click Configure Basic Features, select Load Balancing, and Click Go.
 - b. Click Configure Advanced Features, select SureConnect, and Click Go.

2. In the navigation pane, navigate to Security > Protection Features > SureConnect. In the details pane, click Parameters. In the Configure SureConnect Parameters window, browse and select the VSR filename.
3. Navigate to Traffic Management > Load Balancing > Services. In the details pane, click Add. In the **Create Services** window, enter the parameter values as shown in Table 5-1, and click **OK**.
4. Navigate to Traffic Management > Load Balancing > Virtual servers. In the details pane, click Add. In the Create Virtual Server (Load Balancing) dialog box, enter the values shown in Table 5.1 for the Load Balancing Virtual Server parameters and click OK.
5. In the navigation pane, navigate to Traffic Management > Load Balancing > Virtual servers. Select the virtual server vs-NSact and click Open in the details pane. The Configure Virtual system (Load Balancing) dialog box, displays the list of configured services. Select services psvc1 and psvc2 and click OK.
6. In the navigation pane, expand Security > Protection Features > SureConnect. In the details pane, click Add. Create the policy with the values as given in the parameters table.
7. In the navigation pane, navigate to Traffic Management > Load Balancing > Virtual servers. Select the virtual server vs-NSact and click Open in the details pane. In the Configure Virtual system (Load Balancing) dialog box, click the Policies tab. Click >> to expand the features. Select **SureConnect**. When the list of SureConnect policies appear, select policyNS and click OK.
8. In the navigation pane, navigate to Traffic Management > Load Balancing > Virtual servers. Select the virtual server vs-NSact and click Open in the details pane. In the Configure Virtual system (Load Balancing) dialog box, on the Advanced tab, select SC and click OK.

Configuring the SureConnect Policies

You can configure the following SureConnect policies. The NetScaler appliance matches incoming requests in the order the policies are configured:

- o Exact URL-based policies
- o Wildcard rule-based policies

Configuring Exact URL Based Policies

Updated: 2013-09-03

When you configure an exact URL based policy, the NetScaler appliance matches the incoming request against the URL that has been configured in the policy. URL based policies take precedence over rule based policies.

To configure an exact URL based policy by using the command line interface

At the command prompt, type:

```
add sc policy <name> [-url <URL> | -rule <expression>] [-delay <usecs>] [-maxConn <positive_integer>] [-action (ACS
<altContentSvcName> <altContentPath>) | NS | NOACTION)]
```

To configure an exact URL based policy by using the configuration utility

1. Navigate to Security > Protection Features > SureConnect.
2. In the details pane, click Add.
3. In the Create SureConnect Policy dialog box, set the following parameters:
 - o Name*
 - o URL (Make sure that the URL check box is selected)
 - o Value*
 - o Delay (microseconds)*
 - o Maximum Client Connections
 - o Action (Select from the Choose Action list.)
 - o Alternate Service Name (if you select ACS as the Action)
 - o Alternate Content Path (if you select ACS as the Action)
4. Click Create, and click Close. The URL based policy appears in the right pane, and a message displays in the status bar that the policy is successfully configured.

Configuring Wildcard Rule-Based Policies

Updated: 2013-09-03

SureConnect matches the incoming requests to a defined rule, if you configure a rule-based policy.

To configure a SureConnect policy based on a wildcard rule by using the command line interface

1. Create the expression(s).

Use the add expression command to create each expression.

2. Create the rule(s).

Use the add sc policy command with the -rule expression_logic argument to specify the rule(s). In the -rule expression_logic argument, refer to the expression(s) you created in step 1.

Repeat this command to create and name each rule.

The following example creates a rule `rule == /*.cgi`:

```
add vserver vs-lb http 1.1.1.1 80
add expression expr1 url == /cgi-bin/*.cgi
add expression expr2 url == /index.html
add sc policy surecpolicy1 -rule (expr1|expr2) -delay 1000000 -action NS
bind lb vserver vs-lb -policyName surecpolicy1
```

To complete the SureConnect configuration, you will need to enter additional commands, beyond those shown in the example.

To configure a wildcard rule-based policy by using the configuration utility

1. Navigate to Security > Protection Features > SureConnect.
2. In the details pane, click Add.
3. In the Create SureConnect Policy dialog box, in the Name text box, type the name of the policy.
4. Under What to Monitor, click Expression, and then click Configure.
5. In the Create Expression dialog box, click Add.
6. In the Add Expression dialog box, enter an expression. For example, you can select an Expression Type of General, a Flow Type of REQ, a Protocol of HTTP, a Qualifier of URLQUERY, an Operator of CONTAINS, and in the Value text box, type AA. For more information about expressions, see "Policies and Expressions."
7. Click OK, and click Close.
8. In the Create Expression dialog box, click Create.

Examples of wildcard rules:

`/sports/*` matches all URLs under `/sports`

`/sports*` matches all URLs whose prefix matches `/sports`, starting at the beginning of the URL.

`/*.jsp` matches all URLs whose file extension is `.jsp`.

When configuring rule-based policies, first add the more specific rule-based policies, before adding more generic rules (for example, add `/cgi-bin/sports*.cgi` before adding `/cgi-bin/*.cgi`).

Displaying the Configured SureConnect Policy

To view the SureConnect policy that you have configured, at the NetScaler command prompt, enter the `show sc policy` command.

Customizing the Alternate Content File

When SureConnect activates, it can display alternate content from one of the following files that you have configured:

- **progressbar.htm**. Displays the progress information.
- **alternatepage.htm**. Displays an alternate page.
- **barandpage.htm**. Displays both the progress information and an alternate page.

The alternate content files are JavaScript files. During SureConnect installation, these files are copied onto the server that contains the alternate content. These files can contain alternate content (including an alternate page) or references to other files that contain the alternate content.

This section describes the changes you can make to the alternate content file provided by the appliance.

```
//**** DEFINE YOUR VALUES HERE ****
var alt_url = "/Citrix NetScaler system /sample.gif";
var alt_url = "http://www.DomainName.com";
var Citrix NetScaler system _logo = "netscaler_logo.gif";
var our_logo = "netscaler_logo.gif";
var height = 450;
var width = 550;
var top = 200;
```



```
var left = 200;
var popunder = "no"; //specify yes for pop-under & no for pop-up
var shift_focus = "yes" //if you want to send pop-up to background on getting primary content
/***** YOUR DEFINITIONS ENDS HERE *****/
```

You can make these changes:

- o **var alt_url.** Specify the URL for the alternate content if a file provides the alternate content. For example:

```
var alt_url = "/Citrix NetScaler system/sports.htm";
```

Note: The alternate content file must be present in the /Citrix NetScaler system directory under the documents root of the Web server.

- o **var our_logo.** Specify the image file of your organization logo.
- o **var height.** Specify the height of the SureConnect window.
- o **var width.** Specify the width of the SureConnect window.
- o **var top and var left.** Specify the position of the SureConnect window.
- o **var popunder.** Specifies the position of the alternate content window. Specify the value as NO to place the alternate content window above the original window. Specify the value as YES to place the alternate content window beneath the original window.
- o **var shift_focus.** Specify the focus of the alternate content window. YES places the pop-up window in the background when getting the primary content. NO always keeps the pop-up window in focus, even when getting the primary content.

Note: For more information, see the README.txt file provided by the appliance with other alternate content files.

Configuring SureConnect for Citrix NetScaler Features

Updated: 2013-09-03

This section describes how SureConnect works in combination with the load balancing, content switching, cache redirection, and high availability features of the NetScaler appliance.

Configuring SureConnect for Load Balancing

You can use SureConnect in environments where the primary servers use the load balancing feature, with or without alternate servers. If the load balancing virtual server configured for SureConnect fails, the backup virtual server (if there is one) handles the traffic. Backup virtual servers do not support SureConnect policies.

Note: For information about load balancing, see "[Load Balancing](#)."

Configuring SureConnect for Cache Redirection

You can use SureConnect in environments where cache redirection is configured. The primary server is a load balancing virtual server bound to the cache redirection virtual server. Regardless of any rules configured for the cache redirection feature:

- o You can configure any URL for SureConnect.
- o Once SureConnect is activated for a client, requests from the client are always sent to the origin server.

Configuring SureConnect for High Availability

SureConnect is compatible with NetScaler appliances operating in high availability mode.

Note: If the optional vsr.htm file is used, it must be present in both nodes (primary and secondary) and must use the same name and directory.

Activating SureConnect

You can set the Citrix NetScaler appliance to activate SureConnect if either of two criteria match. Both criteria are arguments to the add sc policy command, as described here:

- o -delay <microseconds>

The first time the client requests the URL, the appliance records how long the server takes to respond. The appliance will not activate SureConnect until the second time the URL is requested. The first and second requests may be from the same or different clients.

If you set -delay argument, SureConnect will be activated the second time the delay reaches the threshold you set.

- o -maxConn <positive_integer>

When the appliance receives a request, it checks the number of connections to the server for the configured URL. SureConnect is activated if the number of connections is greater than or equal to the value that you set for the -maxConn argument.

If you will be providing alternate content to be displayed in the client's Web browser, you should configure the -action argument of the add sc policy command. This specifies for the NetScaler appliance whether the alternate content is coming from a dedicated alternate server (-action ACS) or the appliance (-action NS).

When SureConnect is activated by the -maxConn argument, the SureConnect window and progress bar are displayed in the client's browser (with an alternate page, if configured).

SureConnect Environments

The following topics describe SureConnect environments.

- "Primary and Alternate Servers"
- "Configuration Checklist"
- "Example Configurations"

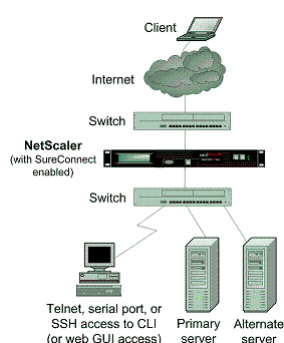
Primary and Alternate Servers

The SureConnect environment uses a dedicated server to provide alternate content when the requested content is not available. The alternate content may include an alternate page, plus optional components such as frame set, organization logo, and so on. The alternate and primary servers can be the same server.

You can configure SureConnect to display a progress bar when the requested content is not available (or the progress bar and an alternate page).

The following figure illustrates the SureConnect environment.

Figure 1. SureConnect - Primary and Alternate Servers



Configuration Checklist

Updated: 2013-09-03

Complete the following checklist before you start configuration:

Table 1. Configuration Checklist

<input type="checkbox"/>	<p>The same builds are running for the appliance and for the SureConnect files as suggested by appliance staff.</p> <p>Appliance Build Number: _____</p> <p>SureConnect (sc_xx.exe) Build Number: _____</p>
<input type="checkbox"/>	<p>The latest SureConnect files (style files) are extracted to:</p> <ul style="list-style-type: none">◦ All primary servers (required for NS action).◦ The alternate content server (required for ACS action).
<input type="checkbox"/>	<p>All customizations to the latest style and vsr.htm files are applied.</p>
<input type="checkbox"/>	<p>The alternate content server is accessible from the Internet (required for ACS action).</p>
<input type="checkbox"/>	<p>If the -redirectURL URL argument of the add vserver CLI command needs to be specified:</p> <ul style="list-style-type: none">◦ The URL is up and running.◦ This URL is not on the configured servers.

	<ul style="list-style-type: none"> o This URL does not match any content in the vserver (that is, do not redirect a missing URL to itself). Redirecting a missing URL to itself can send some browsers into an infinite loop.
□	All URLs to be configured for SureConnect are top-level URLs only. (Only the URLs that occupy the whole window or frame can be configured, not the embedded objects).

Following are the steps to configure SureConnect in a setup with a primary server and a dedicated alternate server:

- o Enable the SureConnect feature
- o Add the SureConnect policy
- o Bind the SureConnect policy

You can optionally configure the following:

- o Redirect the client to another URL if the primary server fails, or send a customized response to the client if the alternate server fails.
- o If the servers do not provide alternate content, send a default or customized response.

To redirect the client to another URL

1. Enable the SureConnect feature.
2. Define the primary server and its service.

You must identify the original server for which SureConnect support is being configured. At the NetScaler command prompt, type the following command:

```
add service <serviceName> <IP> HTTP <port>
```

where <serviceName> assigns a name for the service; <IP> is the server's IP address; and <port> is the port number that the service will use.

Repeat use of the add service CLI command for each service that is to be added.

You can also configure SureConnect on a load balancing virtual server. At the NetScaler command prompt, type the following command:

```
add vserver <name> HTTP <IP> <port>
```

3. Define and bind the SureConnect policy as follows. If you are configuring a rule-based policy, perform this step as described in "Configuring Wildcard Rule-Based Policies." To configure a URL-based policy, at the NetScaler command prompt, type the following command:

```
add sc policy <name> [-url <URL>] [-delay <microsec>] [-maxConn <positiveInteger>]
```

For a detailed description of the add sc policy command, see "Command Reference."

To bind the SureConnect policy, at the NetScaler command prompt, type the following command:

```
bind service <serviceName> -policyname <string>
```

where <serviceName> is the name of the service defined in step 2, and <string> is the name of the SureConnect policy.

Repeat the bind service command for each policy created.

You must include the alternate content page in the altContSvcName argument, and in the altContPath argument of the add sc policy command.

In the following example, the name of the alternate content file is /Citrix NetScaler system /barandpage.htm, and this file resides in svc2.

4. To save the configuration, at the NetScaler command prompt, type the following command:

```
save config
```

Example Configurations

Updated: 2013-09-03

The following examples illustrate various SureConnect configurations.

The examples assume that monitoring of physical services is enabled. If the alternate system is down, SureConnect will deliver the alternate content from the system itself.

Example 1 - SureConnect Progress Bar and Alternate Page

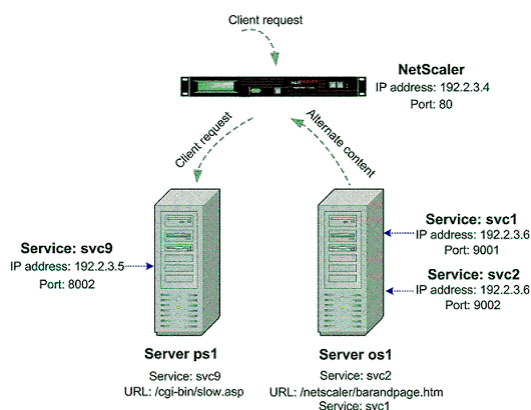
You can configure SureConnect to display both the progress bar and an alternate page to the user.

To bind a SureConnect policy to a load balancing virtual server, at the command prompt, type the following commands:

```
bind lb vserver <virtualServerName> -policyName <string>
```

where <virtualServerName> is the name of the load balancing virtual server defined in step 2 of the configuration process, and <string> is the name of the SureConnect policy defined in step 3.

Figure 2. SureConnect Configuration - Example 1



At the NetScaler command prompt, type the following commands:

```
enable feature SC
show ns info
add service svc2 192.2.3.6 HTTP 9002
show server
show service svc2
add service svc9 192.2.3.5 HTTP 8002
add sc policy policy8 -url /cgi-bin/slow.asp
-delay 3000000 -action ACS svc2 /NetScaler 9000 system barandpage.htm
bind service svc9 -policyname policy8
set service svc9 -sc ON
save config
```

After you configure SureConnect, you can enter commands that show information to verify what you have configured.

Example 2 - SureConnect Progress Bar Only

In this example, SureConnect will display only the progress bar. The server orgsrvr with IP address 10.101.8.187 has service orgsvc. This server is connected to the appliance. The service is bound to the appliance. The progressbar.htm file specifies that only the progress bar will be displayed.

At the NetScaler command prompt, type the following commands:

```
enable feature SC
add service orgsvc 10.101.3.187 HTTP 80
add sc policy policy9 -url /cgi-bin/slow.asp
-delay 4000000 -action ACS orgsvc /NetScaler 9000 system / progressbar.htm
bind service orgsvc -policyname policy9
set service orgsvc -sc ON
save config
```

Example 3 - SureConnect with Load Balancing

This example illustrates how to configure the load balancing feature so that SureConnect will display alternate contents from the primary server. For information about load balancing, see "[Load Balancing](#)."

In this example, two physical servers with IP 10.101.3.187 and 10.101.3.188 are being load balanced by the appliance. The name and location of the alternate page file is specified in the file `alternatepage.htm`, which resides on both servers.

The appliance has one configured virtual server address: 10.101.3.201. At the NetScaler command prompt, type the following commands:

```
enable feature SC LB
add service psvc1 10.101.3.187 HTTP 80
add service psvc2 10.101.3.188 HTTP 80
add vserver vs-SureC HTTP 10.101.3.201 80
bind lb vserver vs-SureC psvc1
bind lb vserver vs-SureC psvc2
add sc policy policy9 -url /cgi-bin/slow.asp -delay 4000000
-action ACS vs-SureC /NetScaler system /alternatepage.htm
bind lb vserver vs-SureC -policyName policy9
set lb vserver vs-SureC -sc ON
save config
```

Example 4 - SureConnect with Load Balancing (ACS Action)

This example illustrates how to configure the NetScaler appliance load balancing feature so that SureConnect will display alternate content from the alternate server. For information about load balancing, see "[Load Balancing](#)."

In this case, there are two physical servers, IP 10.101.3.187 and 10.101.3.188. Both are being load balanced by the appliance.

The name and location of the alternate page file are specified in file `barandpage.htm`, which resides on a third server not being load balanced.

The third server's IP address is 10.101.3.189. Because `barandpage.htm` is specified, the progress bar and alternate page will both be displayed.

The appliance has one configured virtual server whose IP address (Virtual Server) is 10.101.3.200.

At the NetScaler command prompt, type the following commands:

```
enable feature SC LB
add service psvc1 10.101.3.187 HTTP 80
add service psvc2 10.101.3.188 HTTP 80
add service alt-cont-svc 10.101.3.189 HTTP 80
add vserver vsvr HTTP 10.101.3.200 80
bind lb vserver vsvr psvc1
bind lb vserver vsvr psvc2
add sc policy policy10 -url /cgi-bin/slow.asp
-delay 4000000 -action ACS alt-cont-svc
/NetScaler 9000 system /barandpage.htm
bind lb vserver vsvr -policyName policy10
set lb vserver vsvr -sc ON
save config
```

Example 5 - SureConnect with Content Switching

This example illustrates how to configure SureConnect where the NetScaler content switching and load balancing features are being used. SureConnect is configured on a load balancing virtual server bound to a content switching virtual server.

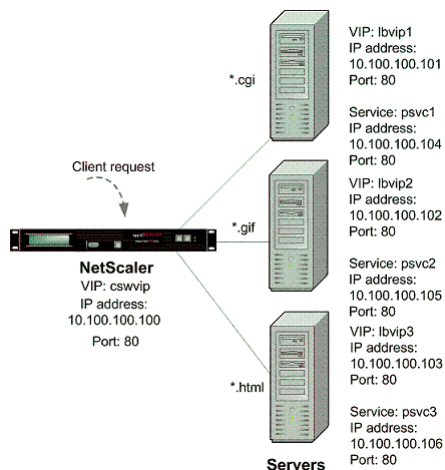
The alternate content is distributed under the content switching virtual server according to the content switching rules. For more information about load balancing and content switching, see "[Load Balancing](#)" and "[Content Switching](#)."

In this case, three physical services with IP addresses 10.100.100.104, 10.100.100.105, and 10.100.100.106 are bound to three load balancing virtual servers with IP addresses 10.100.100.101, 10.100.100.102, and 10.100.100.103. These three load balancing virtual servers are bound to a content switching virtual server with IP address 10.100.100.100.

In this setup, `lbvip1` contains `.cgi` content, `lbvip2` contains `.gif` content, and `lbvip3` contains `.html` content.

The name and location of the alternate page file is specified in the file `alternatepage.htm`, which resides on `lbvip3`. The embedded objects in this file must be distributed according to the content switching rules (any embedded gif will reside on `lbvip2`, any embedded htm will reside on `lbvip3`, and so on).

Figure 3. SureConnect Configuration - Example 5



At the NetScaler command prompt, type the following commands:

```
enable feature CS LB SC
add vservice cswvip HTTP 10.100.100.100 80 -type CONTENT
add vservice lbvip1 HTTP 10.100.100.101 80 -type ADDRESS
add vservice lbvip2 HTTP 10.100.100.102 80 -type ADDRESS
add vservice lbvip3 HTTP 10.100.100.103 80 -type ADDRESS
add service psvc1 10.100.100.104 HTTP 80
add service psvc2 10.100.100.105 HTTP 80
add service psvc3 10.100.100.106 HTTP 80
bind lb vservice lbvip1 psvc1
bind lb vservice lbvip2 psvc2
bind lb vservice lbvip3 psvc3
add cs policy CSWpolicy1 -url /*.cgi
bind cs vservice cswvip lbvip1 -policyName CSWpolicy1
add cs policy CSWpolicy2 -url /*.gif
bind cs vservice cswvip lbvip2 -policyName CSWpolicy2
add cs policy CSWpolicy3 -url /*.htm
bind cs vservice cswvip lbvip3 -policyName CSWpolicy3
add sc policy SCpol -url /cgi-bin/delay.cgi -delay 4000000 -action ACS cswvip /alternatpage
bind lb vservice lbvip1 -policyName SCpol
set lb vservice lbvip1 -sc ON
save config
```

Surge Protection

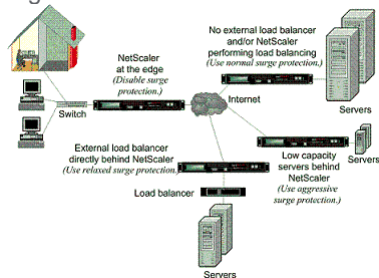
When a surge in client requests overloads a server, server response becomes slow, and the server is unable to respond to new requests. The Surge Protection feature ensures that connections to the server occur at a rate that the server can handle. The response rate depends on how surge protection is configured. The NetScaler appliance also tracks the number of connections to the server, and uses that information to adjust the rate at which it opens new server connections.

Surge protection is enabled by default. If you do not want to use surge protection, as will be the case with some special configurations, you must disable it.

The default surge protection settings are sufficient for most uses, but you can configure surge protection to tune it for your needs. First, you can set the throttle value to tell it how aggressively to manage connection attempts. Second you can set the base threshold value to control the maximum number of concurrent connections that the NetScaler appliance will allow before triggering surge protection. (The default base threshold value is set by the throttle value, but after setting the throttle value you can change it to any number you want.)

The following figure illustrates how surge protection is configured to handle traffic to a Web site.

Figure 1. A Functional Illustration of NetScaler Surge Protection



Note: If the NetScaler appliance is installed at the edge of the network, where it interacts with network devices on the client side of the Internet, the surge protection feature must be disabled. Surge protection must also be disabled if you enable USIP (Using Source IP) mode on your appliance.

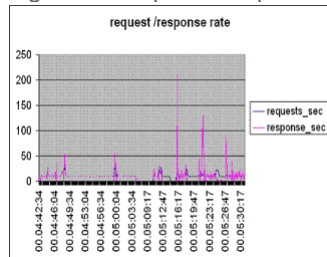
The following example and illustration show the request and response rates for two cases. In one case, surge protection is disabled, and in the other it is enabled.

When surge protection is disabled and a surge in requests occurs, the server accepts as many requests as it can process concurrently, and then begins to drop requests. As the server becomes more overloaded, it goes down and the response rate is reduced to zero. When the server recovers from the crash, usually several minutes later, it sends resets for all pending requests, which is abnormal behavior, and also responds to new requests with resets. The process repeats for each surge in requests. Therefore, a server that is under DDoS attack and receives multiple surges of requests can become unavailable to legitimate users.

When surge protection is enabled and a surge in requests occurs, surge protection manages the rate of requests to the server, sending requests to the server only as fast as the server can handle those requests. This enables the server to respond to each request correctly in the order it was received. When the surge is over, the backlogged requests are cleared as fast as the server can handle them, until the request rate matches the response rate.

The following figure compares the request and response scenarios when surge protection is enabled to that when it is disabled.

Figure 2. Request/Response Rate with and without Surge Protection



Disabling and Reenabling Surge Protection

The surge protection feature is enabled by default. When surge protection is enabled, it is active for any service that you add.

To disable or reenable surge protection by using the command line interface

At the command prompt, type one of the following sets of commands to disable or reenable surge protection and verify the configuration:

- disable ns feature SurgeProtection
- show ns feature
- enable ns feature SurgeProtection
- show ns feature

Example

```
disable ns feature SurgeProtection
Done show ns feature
```

	Feature	Acronym	Status
	-----	-----	-----
1)	Web Logging	WL	ON
2)	Surge Protection	SP	OFF
.			
.			
.			
23)	HTML Injection	HTMLInjection	ON
24)	NetScaler Push	push	OFF

Done

```
enable ns feature SurgeProtection
Done
> show ns feature
```

	Feature	Acronym	Status
	-----	-----	-----
1)	Web Logging	WL	ON
2)	Surge Protection	SP	ON
.			
.			
.			
23)	HTML Injection	HTMLInjection	ON
24)	NetScaler Push	push	OFF

Done

>

To disable or reenable surge protection by using the configuration utility

1. In the navigation pane, expand System, and then select Settings.
2. In the details pane, click Change Advanced Features.
3. In the Configure Advanced Features dialog box, clear the selection from the Surge Protection check box to disable the surge protection feature, or select the check box to enable the feature.
4. Click OK.
5. In the Enable/Disable Feature(s) dialog box, click Yes. A message appears in the status bar, stating that the feature has been enabled or disabled.

To disable or reenable surge protection for a particular service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services. The list of configured services is displayed in the details pane.
2. In the details pane, select the service for which you want to disable or reenable the surge protection feature, and then click Open.
3. In the Configure Service dialog box, click the Advanced tab and scroll down.

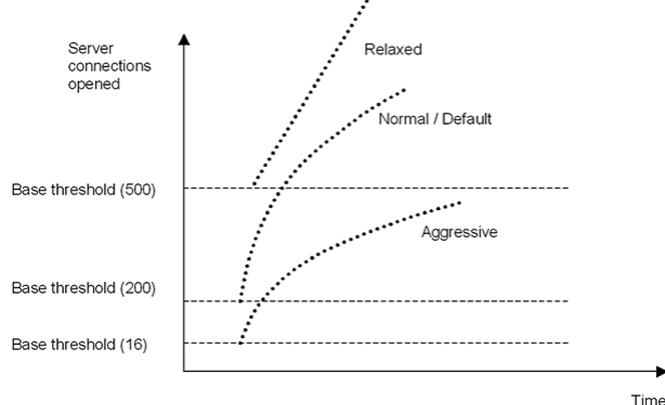
4. In the Others frame, clear the selection from the Surge Protection check box to disable the surge protection feature, or select the check box to enable the feature.
5. Click OK. A message appears in the status bar, stating that the feature has been enabled or disabled.
Note: Surge protection works only when both the feature and the service setting are enabled.

Setting Thresholds for Surge Protection

To set the rate at which the NetScaler appliance opens connections to the server, you must configure the threshold and throttle values for surge protection.

The following figure shows the surge protection curves that result from setting the throttle rate to relaxed, normal, or aggressive. Depending on the configuration of the server capacity, you can set base threshold values to generate appropriate surge protection curves.

Figure 1. Surge Protection Curves

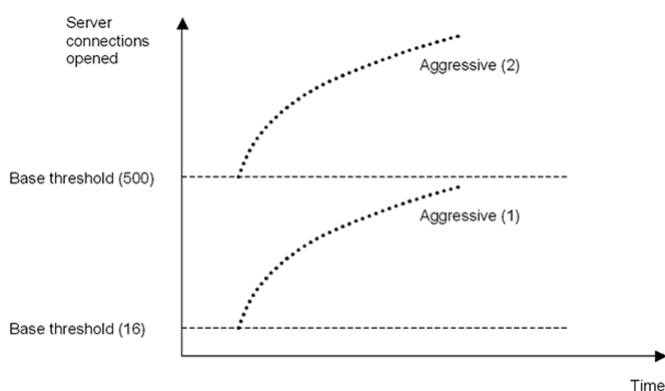


Your configuration settings affect the behavior of surge protection in the following manner:

- If you do not specify a throttle rate, it is set to normal (the default value), and the base threshold is set to 200, as shown in the preceding figure.
- If you specify a throttle rate (aggressive, normal, or relaxed) without specifying a base threshold, the curve reflects the default values of the base threshold for that throttle rate. For example, if you set the throttle rate to relaxed, the resulting curve will have the base threshold value of 500.
- If you specify only the base threshold, the entire surge protection curve shifts up or down, depending on the value you specify, as shown in the figure that follows.
- If you specify both a base threshold and a throttle rate, the resulting surge protection curve is based on the set throttle rate and adjusted according to the value set for the base threshold.

In the following figure, the lower curve (Aggressive 1) results when the throttle rate is set to aggressive but the base threshold is not set. The upper curve (Aggressive 2) results when the base threshold is set to 500, but the throttle rate is not set. The second upper curve (Aggressive 2) also results when the base threshold is set to 500, and the throttle rate is set to aggressive.

Figure 2. Aggressive Rate with the Default or a Set Base Threshold



To set the threshold for surge protection by using the configuration utility

1. In the navigation pane, expand System, and then select Settings.
2. In the details pane, click Global System Settings.
3. If you want to set a base threshold different from the default for the throttle rate, in the Configure Global Settings dialog box, Base Threshold text box, enter the maximum number of concurrent server connections allowed before surge protection is triggered. The base threshold is the maximum number of server connections that can be open before surge protection is activated. The maximum value for this setting is 32,767 server connections. The default setting for this value is controlled by the throttle rate you choose in the next step.
Note: If you do not set an explicit value here, the default value will be used.

4. In the Throttle drop-down list, select a throttle rate. The throttle is the rate at which the NetScaler appliance allows connections to the server to be opened. The throttle can be set to the following values:

Aggressive

Choose this option when the connection-handling and surge-handling capacity of the server is low and the connection needs to be managed carefully. When you set the throttle to aggressive, the base threshold is set to a default value of 16, which means that surge protection is triggered whenever there are 17 or more concurrent connections to the server.

Normal

Choose this option when there is no external load balancer behind the NetScaler appliance or downstream. The base threshold is set to a value of 200, which means that surge protection is triggered whenever there are 201 or more concurrent connections to the server. Normal is the default throttle option.

Relaxed

Choose this option when the NetScaler appliance is performing load balancing between a large number of Web servers, and can therefore handle a high number of concurrent connections. The base threshold is set to a value of 500, which means that surge protection is triggered only when there are 501 or more concurrent connections to the server.

5. Click OK. A message appears in the status bar, stating that the global settings are configured.

Flushing the Surge Queue

When a physical server receives a surge of requests, it becomes slow to respond to the clients that are currently connected to it, which leaves users dissatisfied and disgruntled. Often, the overload also causes clients to receive error pages. To avoid such overloads, the NetScaler appliance provides features such as surge protection, which controls the rate at which new connections to a service can be established.

The appliance does connection multiplexing between clients and physical servers. When it receives a client request to access a service on a server, the appliance looks for an already established connection to the server that is free. If it finds a free connection, it uses that connection to establish a virtual link between the client and the server. If it does not find an existing free connection, the appliance establishes a new connection with the server, and establishes a virtual link between client and the server. However, if the appliance cannot establish a new connection with the server, it sends the client request to a surge queue. If all the physical servers bound to the load balancing or content switching virtual server reach the upper limit on client connections (max client value, surge protection threshold or maximum capacity of the service), the appliance cannot establish a connection with any server. The surge protection feature uses the surge queue to regulate the speed at which connections are opened with the physical servers. The appliance maintains a different surge queue for each service bound to the virtual server.

The length of a surge queue increases whenever a request comes for which the appliance cannot establish a connection, and the length decreases whenever a request in the queue gets sent to the server or a request gets timed out and is removed from the queue.

If the surge queue for a service or service group becomes too long, you may want to flush it. You can flush the surge queue of a specific service or service group, or of all the services and service groups bound to a load balancing virtual server. Flushing a surge queue does not affect the existing connections. Only the requests present in the surge queue get deleted. For those requests, the client has to make a fresh request.

You can also flush the surge queue of a content switching virtual server. If a content switching virtual server forwards some requests to a particular load balancing virtual server, and the load balancing virtual server also receives some other requests, when you flush the surge queue of the content switching virtual server, only the requests received from this content switching virtual server are flushed; the other requests in the surge queue of the load balancing virtual server are not flushed.

Note: You cannot flush the surge queues of cache redirection, authentication, VPN or GSLB virtual servers or GSLB services
Note: Do not use the Surge Protection feature if Use Source IP (USIP) is enabled.

To flush a surge queue by using the command line interface

The `flush ns surgeQ` command works in the following manner:

- You can specify the name of a service, service group, or virtual server whose surge queue has to be flushed.
- If you specify a name while executing the command, surge queue of the specified entity will be flushed. If more than one entity has the same name, the appliance flushes surge queues of all those entities.
- If you specify the name of a service group, and a server name and port while executing the command, the appliance flushes the surge queue of only the specified service group member.
- You cannot directly specify a service group member (`<serverName>` and `<port>`) without specifying the name of the service group (`<name>`) and you cannot specify `<port>` without a `<serverName>`. Specify the `<serverName>` and `<port>` if you want to flush the surge queue for a specific service group member.
- If you execute the command without specifying any names, the appliance flushes the surge queues of all the entities present on the appliance.
- If a service group member is identified with a server name, you must specify the server name in this command; you cannot specify its IP address.

At the command prompt, type:

```
flush ns surgeQ [-name <name>] [-serverName <serverName> <port>]
```

Examples

1.

```
flush ns surgeQ -name SVC1ANZGB -serverName 10.10.10.1 80
```

The above command flushes the surge queue of the service or virtual server that is named SVC
2.

```
flush ns surgeQ
```

The above command flushes all the surge queues on the appliance.

To flush a surge queue by using the configuration utility

Navigate to Traffic Management > Content Switching > Virtual Servers, select a virtual server and, in the Action list, select Flush Surge Queue.

