Implementing Client Certificate Authentication for ADFS Proxy on NetScaler

This guide describes the implementation of client certificate based authentication for the ADFS Proxy solution on NetScaler.
Citrix NetScaler is a world-class product with the proven ability to load balance, accelerate, optimize, and secure enterprise applications.

The ADFS Proxy with client certificate authentication solution allows enterprises to utilize hardware and software based tokens with client certificates for authentication of users. This solution can facilitate secure, multi-factor authentication.

Solution Benefits
- Secure appliance to mitigate malicious attacks
- NetScaler does not need to be domain joined
- Rewrite/URL Transformation for secure redirection
- SSL Offload and Acceleration
- FIPS Benefits (Secure Keys)
- Multi-Factor Authentication
- Centralized user and traffic visibility

Introduction

One of the biggest concerns of any IT organization today is security. Customers are looking to securely provide network resources to their user community, regardless of the physical location of the user (on-premises, off-premises). As cloud computing offerings such as SaaS become more and more prevalent, organizations across all verticals including retail, healthcare, manufacturing, public sector & government and others are looking at these solutions to provide an economies of scale to grow their business functions and capabilities while adhering to their enterprise security policies.

Office 365 (O365) is becoming one of the fastest growing customer adoption cloud based solutions. As organizations look to move to O365 for mail, collaboration, voice & video capabilities they want to extend their security posture without negatively affecting the user experience. The NetScaler platform is the ideal solution for customers to protect their investment on premises and in the cloud. The features and capabilities of the NetScaler to provide federated identity with SAML and ADFS Proxy, enable Single Sign-On (SSO) with our multi-factor authentication element N-Factor allows it to seamlessly integrate into a customer’s identity management and authentication solution. This enables IT organizations to provide the same user experience across all entry points. The SSL Offload capability allows the solution to decrypt encrypted traffic and provide additional security inspection (then re-encrypt if desired). It can also give customers the flexibility to mask and obfuscate their clients URL/URI with URL Rewrite and Transformation capabilities. The NetScaler solution altogether will provide customers a secure and seamless user experience to get access to both their on-premises and cloud resources.

The ADFS Proxy capabilities of NetScaler make it an effective part of the enterprise network for securing the internal ADFS environment by preventing it from being externally exposed. However, it is also necessary that users are still able to access authenticated applications transparently, without being made aware of the presence of the proxy. Preventing unauthorized access to the ADFS server (for users without valid credentials) and securing the proxy server with pre-authentication and other security-oriented capabilities is also an important use case. This solution guide addresses such use cases for enterprises requiring secure ADFS proxy deployments.

Configuration

The table below lists the minimum required software versions for this integration to work successfully. The integration process should also work with higher versions of the same.

<table>
<thead>
<tr>
<th>Product</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>NetScaler</td>
<td>11.1 (Enterprise/Platinum License)</td>
</tr>
</tbody>
</table>
Solution Description

The following instructions assume that ADFS server side configuration has been completed. Please note that on the NetScaler, SNI bindings should be disabled for now on the service side, as we are currently testing the SNI binding functionality with ADFS 3.0.

Part 1: Configure NetScaler

Setup default NetScaler settings, and enable at least the following features — Load Balancing, Content Switching, AAA and Rewrite:

```
enable ns feature WL SP LB CS CMP SSL CF IC SSIVPN AAA REWRITE
enable ns mode FR L3 CKA Edge USNIP SRADV PMTUD
set system parameter -natPcbForceFlushLimit 4294967295 -doppler ENABLED
set system user nsroot 1fffa69faed1dn34k1j1k3739648d125df6b5e609 -encrypted -timeout 36000
set rsskeytype -rsstype ASYMMETRIC
set interface 1/1 -throughput 0 -bandwidthHigh 0 -bandwidthNormal 0 -intftype "Xen Virtual" -ifnum 1/1
set interface 1/2 -throughput 0 -bandwidthHigh 0 -bandwidthNormal 0 -intftype "Xen Virtual" -ifnum 1/2
set interface LO/1 -haMonitor OFF -haHeartbeat OFF -throughput 0 -bandwidthHigh 0 -bandwidthNormal 0 -intftype Loopback -ifnum LO/1
add ns trafficDomain 4094
add ns ip6 fe80::3c7e:7dff:fe40:d2d2/64 -scope link-local -type NSIP -vlan 1 -vServer DISABLED -mgmtAccess ENABLED -dynamicRouting ENABLED
add ns ip 10.105.157.181 255.255.255.0 -vServer DISABLED
add ns ip 192.168.1.13 255.255.255.0 -vServer DISABLED -mgmtAccess ENABLED
```

(This configuration uses sample IP addresses, replace the same with the relevant ones for your environment)
Add the KCD account used for backend Kerberos:

```
add aaa kcdAccount kcd_adfs_proxy -realmStr TESTREALM.COM -delegatedUser administrator -kcdPassword fa0fc1ef5209621785252dajd1237cfcf4564b50d3546999f-52f66385a94 -encrypted -encryptmethod ENCMTHD_3
```

Add the ADFS server and service:

```
add server adfs_xenapp 10.105.157.131
add service adfs_https_xenapp adfs_xenapp SSL 443 -gslb NONE -maxClient 0 -maxReq 0 -cip ENABLED X-MS-Forwarded-Client-IP -usip NO -useproxyport YES -sp ON -cltTimeout 180 -svrTimeout 360 -CKA NO -TCPB NO -CMP YES
```

Add the required SSL certificates (in this setup, all certificates are created using an internal CA):

```
add ssl certKey ns-server-certificate -cert ns-server.cert -key ns-server.key
add ssl certKey adfs_test_cac -cert adfs_test_cac.cer -key adfs_test_cac.key
add ssl certKey wildcard_TESTREALM -cert wildcard_TESTREALM.cer -key wildcard_TESTREALM.key
add ssl certKey test_clientcert -cert test_clientcert.cer -key test_clientcert.key
add ssl certKey nsroot -cert ns-root.cert -key ns-root.key
add ssl certKey startcom -cert startcom.cer -inform DER
add ssl certKey ns-root-ca -cert ns-root-ca.cer
add ssl certKey ns-client-cert -cert ns-client-cert.cer -key ns-client-cert.key
add ssl certKey adfscac -cert adfscac.cer -key adfscac.key
```

Create the authentication and traffic policy/actions required (LDAP, Client certificate and Negotiate (for Kerberos) defined here:

```
add authentication ldapAction ldap_1 -serverIP 10.105.157.116 -ldapBase "CN=Users,DC=TESTREALM,DC=COM" -ldapBindDn "CN=Administrator,CN=Users,DC=TESTREALM,DC=COM" -ldapBindDnPassword 4fcb95f983ceaaabf6d1867280484f5de8ffe-12197a3f5fbbff766bec645acdf0 -encrypted -encryptmethod ENCMTHD_3 -ldapLoginName sAMAccountName
add authentication negotiateAction adfs_proxy_neg -domain TESTREALM.com -domainUser administrator -domainUserPasswd 4606206b6e7ccac0be0425494f506d228c3f3a934e5e90c653eb55208b4943 -encrypted -encryptmethod ENCMTHD_3 -NTLMPath "http://10.105.157.131/integrated.html"
add authentication certAction cert_profile_test -twoFactor ON -userNameField Subject:CN
```

```
add tm trafficAction prf_sso_to_401-adfs -SSO ON -persistentCookie OFF -InitiateLogout OFF -kcdAccount kcd_adfs_proxy
```

(continued on the next page)
Solution Guide
Implementing Client Certificate Authentication for ADFS Proxy on NetScaler

Add CS server and policies, and LB vservers

Add required rewrite actions and the URL Transformation policy

(continued on the next page)
add rewrite policy pol_rw_adfs_mexrequest "http.REQ.URL.CONTAINS("/adfs/services/trust/mex")" act_rw_adfs_mexrequest
add rewrite policy rw_pol_adfs_ProxyHeader_xenapp "http.REQ.URL.TO_LOWER.STARTSWITH("/adfs")" rewrite_adfs_ProxyHeader_xenapp
add rewrite policy rw_pol_adfs_Mex_xenapp "http.REQ.URL.TO_LOWER.STARTSWITH("/adfs/services/trust/mex")" rewrite_adfs_Mex_xenapp
add rewrite policy Replace_URL_Test "http.REQ.HOSTNAME.CONTAINS("adfscac")" Request_Replace_Test
add rewrite policylabel test_http_response http_res
add rewrite policylabel response_rewrite-url
add transform profile Rewrite_Test
add transform action Rewrite_Action Rewrite_Test 100
set transform action Rewrite_Action -priority 100 -reqUrlFrom adfsfake.TESTREALM.com -reqUrlInto adfscac.TESTREALM.com -resUrlFrom adfscac.TESTREALM.com -resUrlInto adfsfake.TESTREALM.com
add transform policy rewrite_response true Rewrite_Test

Bind policies and services to LB vservers

bind lb vserver proxy_adfs_server adfs_https_xenapp
bind lb vserver vip_adfs_http_adfs_https_xenapp
bind lb vserver vip_adfs_https_xenapp adfs_https_xenapp
bind lb vserver vip_adfs_proxy_adfs_proxy_pol -priority 100 -gotoPriorityExpression END -type REQUEST
bind lb vserver vs_https_adfsproxy_noauth -policyName pol_rw_adfs_mexrequest -priority 100 -gotoPriorityExpression END -type REQUEST
bind lb vserver vip_adfs_https_xenapp -policyName rw_pol_adfs_ProxyHeader_xenapp -priority 100 -gotoPriorityExpression NEXT -type REQUEST
bind lb vserver vip_adfs_https_xenapp -policyName rw_pol_adfs_Mex_xenapp -priority 100 -gotoPriorityExpression END -type REQUEST
bind lb vserver vip_adfs_http -policyName rw_pol_adfs_ProxyHeader_xenapp -priority 100 -gotoPriorityExpression END -type REQUEST
bind lb vserver vip_adfs_http -policyName rw_pol_adfs_Mex_xenapp -priority 100 -gotoPriorityExpression END -type REQUEST
bind lb vserver vs_https_adfsproxy -policyName pol_sso_to_401-adfs -priority 100 -gotoPriorityExpression END -type REQUEST
bind cs vserver vip_adfs_xenapp -policyName rewrite_response -priority 100 -gotoPriorityExpression END -type REQUEST
bind cs vserver vip_adfs_xenapp -policyName adfs_xenapp -targetLBVserver vip_adfs_https_xenapp -priority 70
bind cs vserver vip_adfs_xenapp -lbvserver vip_adfs_http
Add SSL certificates and set SSL parameters, add DNS nameservers and LB monitor for the ADFS server (accesses federation metadata)

add dns nameServer 192.168.1.15
add dns nameServer 10.105.157.14
add dns nameServer 10.105.157.116 -dnsProfileName default-dns-profile

add lb monitor mon_adfs_https_xenapp HTTP-ECV -customHeaders "host: adfs-cac.TESTREALM.com\r\n" -send "GET /federationmetadata/2007-06/federationmetadata.xml" -recv "adfstest.TESTREALM.com/adfs/services/trust" -LRTM ENABLED -secure YES

add route 0.0.0.0 0.0.0.0 10.105.157.1
set ssl service adfs_https_xenapp -tls11 DISABLED -tls12 DISABLED
set ssl vserver proxy_adfs_server -cipherRedirect ENABLED -tls11 DISABLED -tls12 DISABLED
set ssl vserver vip_adfs_https_xenapp -clientAuth ENABLED -clientCert Mandatory -tls11 DISABLED -tls12 DISABLED
set ssl vserver ldap_auth_vs -tls11 DISABLED -tls12 DISABLED
set ssl vserver vip_adfs_xenapp -clientAuth ENABLED -clientCert Mandatory -tls11 DISABLED -tls12 DISABLED

add tm sessionPolicy adfs_proxy_sess ns_true adfs_sess_prof

Binding of Certificates and SSL Settings

bind authentication vserver ldap_auth_vs -policy cert_test -priority 100
bind ssl vserver proxy_adfs_server -cipherName ALL
bind ssl vserver proxy_adfs_server -cipherName DEFAULT
bind ssl vserver vip_adfs_xenapp -cipherName DEFAULT
bind ssl vserver vip_adfs_xenapp -cipherName ALL
bind ssl vserver adfs_proxy_activeserver -certkeyName exch2016
bind ssl vserver proxy_adfs_server -certkeyName wildcard_TESTREALM
bind ssl vserver vs_https_adfsproxy -certkeyName exch2016
bind ssl vserver vs_https_adfsproxy_noauth -certkeyName exch2016
bind ssl vserver ADFSProxy Passive -certkeyName exch2016
bind ssl vserver vip_adfs_https_xenapp -certkeyName adfscac
bind ssl vserver vip_adfs_https_xenapp -certkeyName ns-root-ca -CA -ocsp-
Check Optional
bind ssl vserver ldap_auth_vs -certkeyName exch2016
bind ssl vserver ldap_auth_vs -certkeyName nsroot -CA -ocspCheck Optional
bind ssl vserver adfs_proxy_aaa -certkeyName exch2016
bind ssl vserver adfs_proxy_passive_aaa -certkeyName exch2016
bind ssl vserver vs_aaa_adfsproxy -certkeyName exch2016
bind ssl vserver cs_exch2016 -certkeyName exch2016
bind ssl vserver cs_adfs_proxy -certkeyName adfs_test_cac
bind ssl vserver vs_https_adfs -certkeyName exch2016
bind ssl vserver vip_adfs_xenapp -certkeyName adfscac
bind ssl vserver vip_adfs_xenapp -certkeyName ns-root-ca -CA -skipCAName
-ocspCheck Optional
bind ssl vserver adfs_proxy_activeserver -eccCurveName P_256
bind ssl vserver adfs_proxy_activeserver -eccCurveName P_384
bind ssl vserver adfs_proxy_activeserver -eccCurveName P_224
bind ssl vserver adfs_proxy_activeserver -eccCurveName P_521
bind ssl vserver proxy_adfs_server -eccCurveName P_256
bind ssl vserver proxy_adfs_server -eccCurveName P_384
bind ssl vserver proxy_adfs_server -eccCurveName P_224
bind ssl vserver proxy_adfs_server -eccCurveName P_521
bind ssl vserver vs_https_adfsproxy -eccCurveName P_256
bind ssl vserver vs_https_adfsproxy -eccCurveName P_384
bind ssl vserver vs_https_adfsproxy -eccCurveName P_224
bind ssl vserver vs_https_adfsproxy -eccCurveName P_521
bind ssl vserver vs_https_adfsproxy_noauth -eccCurveName P_256
bind ssl vserver vs_https_adfsproxy_noauth -eccCurveName P_384
bind ssl vserver vs_https_adfsproxy_noauth -eccCurveName P_224
bind ssl vserver vs_https_adfsproxy_noauth -eccCurveName P_521
bind ssl vserver ADFSProxy Passive -eccCurveName P_256
bind ssl vserver ADFSProxy Passive -eccCurveName P_384
bind ssl vserver ADFSProxy Passive -eccCurveName P_224
bind ssl vserver ADFSProxy Passive -eccCurveName P_521
bind ssl vserver vip_adfs_https_xenapp -eccCurveName P_256
bind ssl vserver vip_adfs_https_xenapp -eccCurveName P_384
bind ssl vserver vip_adfs_https_xenapp -eccCurveName P_224
bind ssl vserver vip_adfs_https_xenapp -eccCurveName P_521
bind ssl vserver ldap_auth_vs -eccCurveName P_256
bind ssl vserver ldap_auth_vs -eccCurveName P_384
bind ssl vserver ldap_auth_vs -eccCurveName P_224
bind ssl vserver ldap_auth_vs -eccCurveName P_521
bind ssl vserver adfs_proxy_aaa -eccCurveName P_256
bind ssl vserver adfs_proxy_aaa -eccCurveName P_384
bind ssl vserver adfs_proxy_aaa -eccCurveName P_224
bind ssl vserver adfs_proxy_aaa -eccCurveName P_521
bind ssl vserver adfs_proxy_passive_aaa -eccCurveName P_256
bind ssl vserver adfs_proxy_passive_aaa -eccCurveName P_384
bind ssl vserver adfs_proxy_passive_aaa -eccCurveName P_224
bind ssl vserver adfs_proxy_passive_aaa -eccCurveName P_521
bind ssl vserver vs_aaa_adfsproxy -eccCurveName P_256
bind ssl vserver vs_aaa_adfsproxy -eccCurveName P_384
bind ssl vserver vs_aaa_adfsproxy -eccCurveName P_224
bind ssl vserver vs_aaa_adfsproxy -eccCurveName P_521
bind ssl vserver cs_adfs_proxy -eccCurveName P_256
bind ssl vserver cs_adfs_proxy -eccCurveName P_384
bind ssl vserver cs_adfs_proxy -eccCurveName P_224
bind ssl vserver cs_adfs_proxy -eccCurveName P_521
bind ssl vserver vs_https_adfs -eccCurveName P_256
bind ssl vserver vs_https_adfs -eccCurveName P_384
bind ssl vserver vs_https_adfs -eccCurveName P_224
bind ssl vserver vs_https_adfs -eccCurveName P_521
bind ssl vserver vip_adfs_xenapp -eccCurveName P_256
bind ssl vserver vip_adfs_xenapp -eccCurveName P_384
bind ssl vserver vip_adfs_xenapp -eccCurveName P_224
bind ssl vserver vip_adfs_xenapp -eccCurveName P_521

This completes the configuration for the ADFS Client Certificate solution on NetScaler.
Conclusion
NetScaler enables custom, secure ADFS deployments with its authentication and authorization capabilities. With NetScaler, enterprises can enable a host of additional capabilities including but not limited to authentication offload, end point analysis checks, selective server access, URL rewrites, compression, caching, front end optimizations and much more.

To learn more about how NetScaler can bring these benefits to address this and other enterprise requirements, please visit http://www.citrix.com.
Add the KCD account used for backend Kerberos:

```
add aaa kcdAccount kcd_adfs_proxy -realmStr TESTREALM.COM -delegatedUser administrator -kcdPassword fa0fc1ef5209621785252dajd1237cfe54564b50d3546999f-52f66385a94 -encrypted -encryptmethod ENCMTHD_3
```

Add the ADFS server and service:

```
add server adfs_xenapp 10.105.157.131
add service adfs_https_xenapp adfs_xenapp SSL 443 -gslb NONE -maxClient 0 -maxReq 0 -cip ENABLED X-MS-Forwarded-Client-IP -usip NO -useproxyport YES -sp ON -cltTimeout 180 -svrTimeout 360 -CKA NO -TCPB NO -CMP YES
```

Add the required SSL certificates (in this setup, all certificates are created using an internal CA):

```
add ssl certKey ns-server-certificate -cert ns-server.cert -key ns-server.key
add ssl certKey adfs_test_cac -cert adfs_test_cac.cer -key adfs_test_cac.key
add ssl certKey wildcard_TESTREALM -cert wildcard_TESTREALM.cer -key wildcard_TESTREALM.key
add ssl certKey test_clientcert -cert test_clientcert.cer -key test_clientcert.key
add ssl certKey nsroot -cert ns-root.cert -key ns-root.key
add ssl certKey startcom -cert startcom.cer -inform DER
add ssl certKey ns-root-ca -cert ns-root-ca.cer
add ssl certKey ns-client-cert -cert ns-client-cert.cer -key ns-client-cert.key
add ssl certKey adfscac -cert adfscac.cer -key adfscac.key
```

Create the authentication and traffic policy/actions required (LDAP, Client certificate and Negotiate (for Kerberos) defined here:

```
add authentication ldapAction ldap_1 -serverIP 10.105.157.116 -ldapBase "CN=Users,DC=TESTREALM,DC=COM" -ldapBindDn "CN=Administrator,CN=Users,DC=TESTREALM,DC=COM" -ldapBindDnPassword efe52e8fe-12197a3f5fbbbf766bec645acdf0 -encrypted -encryptmethod ENCMTHD_3 -ldapLoginName sAMAccountName
add authentication negotiateAction adfs_proxy_neg -domain TESTREALM.com -domainUser administrator -domainUserPasswd 4fcb95f983ceaaabfd1867280484f5de8ffe-15697a3f5fbbbf766bec645acdf0 -encrypted -encryptmethod ENCMTHD_3 -NTLMPath "http://10.105.157.131\integrated.html"
add authentication certAction cert_profile_test -twoFactor ON -userNameField Subject:CN
```

```
add tm trafficAction prf_sso_to_401-adfs -SSO ON -persistentCookie OFF -InitiateLogout OFF -kcdAccount kcd_adfs_proxy
```

(continued on the next page)