Microsoft TMG Replacement with NetScaler

Replacing Microsoft Forefront TMG with NetScaler for Optimization

This deployment guide focuses on replacing Microsoft Forefront Threat Management Gateway (TMG) with NetScaler for application optimization.
Table of Contents

Introduction .................................................................................. 3
Configuration details ...................................................................... 3
NetScaler features to be enabled ..................................................... 3
Solution description ...................................................................... 5
  Part 1: Configuring optimization on TMG ................................... 5
    HTTP compression .................................................................. 5
    URL filtering ......................................................................... 6
    Web caching .......................................................................... 8
  Part 2: Configuring optimization on NetScaler ......................... 10
    HTTP compression .................................................................. 10
    Integrated caching .................................................................. 12
    Front-end optimization ......................................................... 17
    Content filtering on NetScaler with Responder and Rewrite .... 19
Conclusion .................................................................................... 23
NetScaler is a world-class application delivery controller (ADC) with the proven ability to load balance, accelerate, optimize and secure enterprise applications. Forefront TMG can help improve network performance with web traffic compression and caching of frequently accessed web resources, which allows them to be accessed faster.

In this guide, we explain how to replicate TMG optimization functionality with NetScaler by comparing the configuration process on TMG with the configuration process on NetScaler. The guide then highlights the additional possibilities that NetScaler enables, such as advanced URL filtering and rewriting using custom policies and selective, policy driven caching capabilities.

The following TMG features are replicated with NetScaler in this guide:

<table>
<thead>
<tr>
<th>Networking and performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caching</td>
</tr>
<tr>
<td>HTTP compression</td>
</tr>
<tr>
<td>URL filtering</td>
</tr>
</tbody>
</table>

Configuration Details

<table>
<thead>
<tr>
<th>Product</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forefront TMG</td>
<td>2010 SP1</td>
</tr>
<tr>
<td>NetScaler VPX</td>
<td>10.5 (Enterprise License)</td>
</tr>
</tbody>
</table>

**NetScaler features to be enabled**

The essential NetScaler features that need to be enabled are described below. Please ensure they are enabled in the NetScaler system.

Content switching, load balancing, Rewrite and HTTP compression can be found under **System>Settings>Configure Basic Features**.

Front-end optimization (FEO) and Responder can be found under **System>Settings>Configure Advanced Features**

Here is a quick explanation of how these features work.
Content switching
The content switching module directs incoming traffic to a matching load balancing virtual server. This logical switching of incoming traffic based on content type allows configuration of specific optimization policies.

Load balancing
NetScaler load balancing evenly distributes requests to backend servers. Multiple algorithms (such as LEASTCONNECTION, ROUNDROBIN, etc.) are supported to provide efficient load balancing logic for every application server.

HTTP compression
Compression of HTTP traffic is performed using standard GZIP/DEFLATE methods.

Rewrite and Responder
Rewrite and Responder are advanced policy driven features that enable NetScaler to effectively process user-entered URLs by rewriting them for accuracy or redirecting them to the appropriate page, posting an appropriate error when required and other URL filtered responses.

Front-end optimization
An advanced optimization feature, FEO enables NetScaler to significantly accelerate web content with methods such as image compression.

Integrated caching
Content caching allows NetScaler to serve frequently used content without requiring round trips to the source web server.
Solution description

Part 1: Configuring optimization on TMG

Open the Forefront TMG console, then right-click the Web Access Policy node in the left pane of the main window to gain access to TMG optimization settings, as shown below.

Here, the following optimization options are available:

• HTTP compression
• URL filtering
• Web caching

To enable or disable each of these features, click on the hyperlinked status prompt next to the feature name, which gives a pop-up that allows configuration of the feature.

HTTP compression
This is the prompt provided for HTTP compression. Here, the only available option is to enable or disable HTTP compression. The two additional tabs, as described in the prompt, allow definition of the network clients from which compressed HTTP responses are to be sent/returned to (Return Compressed Data) or requested from (Request Compressed Data).

URL filtering
The URL filtering feature enables traffic to certain websites to be allowed or blocked based on access rules linked to URL categories. This feature helps to optimize bandwidth consumption by restricting use of unnecessary websites.
The URL filtering prompt allows you to enable or disable the feature; URL categorization is determined using the Microsoft Reputation Service as described in the prompt. However, it is possible to override the category of a particular URL using the URL Category Override tab (shown below), thus controlling access to websites directly. The Category Query tab allows verification of the listed category for a particular URL. License Details allows you to update licenses for access to Web Protection Services; expired licenses can be extended only up to December 31, 2015, using information provided at https://support.microsoft.com/en-us/kb/2793998?wa=wsignin1.0
Web caching
TMG’s web caching feature is disabled by default. To enable it, you should first allocate some disk space for a cache drive that TMG then uses to handle web caching.

To enable caching, create a new cache drive by going to the Cache Drives tab and adding cache capacity.
As shown above, click on **Configure** to set up cache capacity. This will give you the following window where cache capacity can be defined:

Additionally, TMG allows creation of cache rules that define how caching for a particular network should be handled. The **Advanced** settings tab provides additional configuration options, as shown below:
Part 2: Configuring optimization on NetScaler

NetScaler provides a flexible, comprehensive suite of optimization capabilities that can be categorized as follows:

- **HTTP compression** (equivalent to TMG’s similarly named feature)
- **Integrated caching** (equivalent to web caching on TMG)
- **Front-end optimization** (additional optimization capabilities)
- **Responder and Rewrite** (provide URL and other filtering capabilities)

To configure HTTP compression, integrated caching and FEO, expand the **Optimization** tab in the NetScaler GUI’s left-hand navigation panel.

**HTTP compression**

The NetScaler optimization suite is, like other NetScaler features, driven by a policy-action architecture.

To enable HTTP compression for a particular service, you should:

- Define the HTTP Compression Policy and Action relevant to the service
- Bind the service to the relevant virtual server

To define the Compression Policy and Action, click on the **Policies** option under **HTTP Compression**, shown above. This gives you the following screen:
To add a new compression policy, click the **Add** button. This will give you the following screen:

Here, you can define a name for the policy, and an expression that defines when this policy is triggered (for example, when a particular URL is encountered.) To make the policy apply to all content, use `ns_true` in the **Expression** window. For assistance here, click on the **Frequently Used Expressions** drop down and the Response Action that should be taken. Here, the actions available are COMPRESS (GZIP or DEFLATE compression, with GZIP given priority), GZIP (GZIP standard compression), DEFLATE (DEFLATE compression) and NOCOMPRESS.

You can either add a new action or reconfigure the existing ones. You can add using the + button, or edit/configure using the pencil-shaped button. Either option gives you a screen similar to the one shown below:
**Vary Header Insertion** is an option that is relevant for caching; the value of the Vary header allows different cache results to be returned for similar requests. For now, we are not changing the options presented here. You can add a new action that uses a compression type of your choice.

**Integrated caching**
After you enable the integrated cache feature and perform basic setup (for example, determining the amount of NetScaler appliance memory the cache is permitted to use), the integrated cache uses built-in policies to store and serve specific types of static content, including simple web pages and image files. You can also configure the integrated cache to store and serve dynamic content that is usually marked as non-cacheable by web and application servers (for example, database records and stock quotes).

To configure caching, you can use the integrated wizard that makes configuration very straightforward. To initiate the wizard, navigate to **Optimization > Integrated Caching** as shown below:
You can initiate the **Caching Wizard** under **Getting Started**.

The first step requires you to specify the content type. This can be either **static** (examples given) or **dynamic**. Helpful hints are provided as shown above to help determine which type of content is relevant for you.

The next step involves defining which content should be cached. The **Frequently Used Expressions** dropdown helps you define the correct expression; however, if you want the caching policy to run for all content, you can use `ns_true` as the expression, as shown below:
The next screen allows you to define when content expires. This can be custom (a defined interval) or heuristic (NetScaler makes the determination based on a guideline percentage of the time since the object was last modified, with a deadline to be set that is used in case the heuristic measurement cannot be made).

The next step involves definition of the caching space to be used on NetScaler and the minimum size of objects to be cached.
Finally, the cache policy should be bound to a relevant LB or CS vserver.

These definitions can be made under **Cache Policies** as shown in the screenshot above.

With respect to dynamic caching, NetScaler allows configuration of selectors and content groups.
Selectors and content groups
A content group is a container for cached objects that can be served in a response. When you first enable the integrated cache, cacheable objects are stored in a content group named Default. You can create new content groups that have unique properties. For example, you can define separate content groups for image data, bug reports and stock quotes, and you can configure the stock quote content group to be refreshed more often than the other groups.

You can configure expiration of an entire content group or selected entries in a content group.

The data in a content group can be static or dynamic:

- **Static content groups** – Find an exact match between the URL stem and host name on the request and the URL stem and host name of the response
- **Dynamic content groups** – Look for objects that contain particular parameter-value pairs, arbitrary strings or string patterns. Dynamic content groups are useful when caching data that is updated frequently (for example, a bug report or a stock quote)

When you add a selector to one or more content groups, you specify whether the selector is to be used for identifying cache hits or identifying cached objects to be invalidated (expired). Selectors are optional. Alternatively, you can configure content groups to use hit parameters and invalidation parameters. However, Citrix recommends that you configure selectors.

Both of these features can be accessed using the navigation menu in the NetScaler GUI.

For more-detailed information on integrated caching and the features described above, please refer to [http://support.citrix.com/proddocs/topic/ns-optimization-10-5-map/ns-IC-gen-wrapper-10-con.html](http://support.citrix.com/proddocs/topic/ns-optimization-10-5-map/ns-IC-gen-wrapper-10-con.html)
Front-end optimization
The FEO feature set makes NetScaler an extremely capable optimization device by implementing enhanced optimization routines for specific front-end entities such as images, JavaScript etc. These provide far better optimization performance than that achieved by compression and caching.

FEO capabilities can be activated by navigating to Optimization>Front End Optimization. As with all NetScaler features, FEO capabilities are implemented using a policy-action mechanism.

To add a new policy, click on Policies then click Add in the section displayed to the right of the navigation menu.

This will give you the following screen for definition of a new FEO policy.
The Expression here works much on the same lines as expressed for the earlier features; the Frequently Used Expressions drop down can be used for assistance. There are certain predefined actions that can be assigned here, all of which have different configurations for the same settings; you can also either edit or create a custom action, which can be done using the plus or pencil buttons next to the Action name.

Upon clicking either of these buttons, the following screen (or a similar one) is shown:

This screen presents all the various FEO options available; NetScaler can help to optimize web traffic with JavaScript, image, cascading style sheets (CSS), HTML and miscellaneous optimization. This last section also allows for domain sharding, which splits resources across subdomains to improve optimization and page load times.
For FEO implementation, it is recommended that you try out different built-in profiles/create custom profiles to see which of the available optimization options can be used without affecting performance. Begin with basic, evaluate performance and then move on to moderate/aggressive to find the profile that gives you best performance improvements.

Content filtering on NetScaler with Responder and Rewrite

The URL filtering capability with NetScaler can be found under AppExpert>Responder on the left-hand navigation menu in the NetScaler GUI. The Responder feature provides several options over and above the standard URL filtering capabilities of TMG.

NetScaler advantage

FEO allows NetScaler to significantly accelerate content across the network by optimizing common sources of inefficiencies in page load times. This provides a big advantage over the standard cache and compression capabilities of TMG, which are also supported by NetScaler.
With the Responder feature, responses can be based on who sends the request, where it is sent from and other criteria with security and system management implications. The feature is simple and quick to use. By avoiding the invocation of more-complex features, it reduces CPU cycles and time spent in handling requests that do not require complex processing.

For example, for handling sensitive data such as financial information, if you want to ensure that the client uses a secure connection to browse a site, you can redirect the request to secure connection by using https:// instead of http://.

To use the Responder feature, first enable it on NetScaler from System>Settings>Configure Advanced Features.

Then, navigate to AppExpert>Responder and click Policies to add a new policy.

After clicking Add, the next screen allows you to configure the policy as well as the action bound to it.
The policy determines the requests (traffic) for which an action has to be taken. This determination is made using the expression provided by the administrator. Multiple frequently used expressions are available for use, as shown below.

To configure the action bound to this policy, click the pencil-shaped icon next to the action name indicated. Note that you will not be able to edit the default policies, only custom ones that you have created. To add a new policy, click the plus-shaped icon.

The action can be to generate a custom response or redirect a request to a different, custom web page.

The policy must be bound to a bind point put it into effect. A bind point refers to an entity where NetScaler examines the traffic to see if it matches a policy. For example, a bind point can be a load balancing virtual server.
You can specify a default action for requests that do not match any policy, and you can bypass the safety check for actions that would otherwise generate error messages.

The Rewrite feature helps in rewriting some information in the requests or responses handled by NetScaler. For example, the following Rewrite action adds the “/owa” string to any user URL when it contains the organization's email domain.

![Rewrite action configuration](image)

The policy that enables this action to be performed when the URL contains the organization's mail domain is defined as follows (again, these are configurable in the AppExpert>Rewrite section under Policies and Actions):
Here, **KCD_Rewrite** is the policy name, and **TMG_Add_OWA** is the action we defined earlier.

There are differences between the Rewrite and Responder features.

Responder cannot be used for response- or server-based expressions. Responder can be used only for the following scenarios, depending on client parameters:

- Redirecting an HTTP request to new websites or web pages
- Responding with a custom response
- Dropping or resetting a connection at the request level

For a Responder policy, the NetScaler device examines the request from the client, takes action according to the applicable policies, sends the response to the client and closes the connection with the client.

For a Rewrite policy, the NetScaler device examines the request from the client or response from the server, takes action according to the applicable policies and forwards the traffic to the client or the server.

In general, it is recommended to use Responder if you want NetScaler to reset or drop a connection based on a client- or request-based parameter. Use Responder to redirect traffic or respond with custom messages. Use Rewrite for manipulating data on HTTP requests and responses.

**Conclusion**

NetScaler provides a complete and enhanced replacement for Microsoft Forefront TMG for enterprise optimization of web content. As indicated by the NetScaler Advantage callout, the Citrix solution presents benefits over TMG for enterprises and provides a wide range of optimization options, not only improving on TMG’s own capabilities but also adding the significant benefit of FEO.