## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP Denial-of-Service Protection</td>
<td>3</td>
</tr>
<tr>
<td>DoS Protection</td>
<td>5</td>
</tr>
<tr>
<td>Layer 3-4 SYN Denial-of-Service Protection</td>
<td>6</td>
</tr>
<tr>
<td>Enabling HTTP DoS Protection</td>
<td>7</td>
</tr>
<tr>
<td>Defining an HTTP DoS Policy</td>
<td>9</td>
</tr>
<tr>
<td>Configuring an HTTP DoS Service</td>
<td>11</td>
</tr>
<tr>
<td>Binding an HTTP DoS Monitor and Policy</td>
<td>14</td>
</tr>
<tr>
<td>Tuning the Client Detection/JavaScript Challenge Response Rate</td>
<td>17</td>
</tr>
<tr>
<td>Guidelines for HTTP DoS Protection Deployment</td>
<td>18</td>
</tr>
</tbody>
</table>
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.
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.
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 NetScaler command line

At the NetScaler 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

<table>
<thead>
<tr>
<th>Feature</th>
<th>Acronym</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Web Logging</td>
<td>WL</td>
<td>ON</td>
</tr>
<tr>
<td>2) Surge Protection</td>
<td>SP</td>
<td>OFF</td>
</tr>
<tr>
<td>11) Http DoS Protection</td>
<td>HDOSP</td>
<td>ON</td>
</tr>
<tr>
<td>12) Content Filtering</td>
<td>CF</td>
<td>ON</td>
</tr>
<tr>
<td>23) HTML Injection</td>
<td>HTMLInjection</td>
<td>ON</td>
</tr>
<tr>
<td>24) NetScaler Push</td>
<td>push</td>
<td>OFF</td>
</tr>
</tbody>
</table>
```

Done
"
To enable HTTP DoS protection by using the configuration utility

1. In the navigation pane, expand System, and then click Settings.

2. In the details pane, click Change Advanced Features.

3. In the Configure Advanced Features dialog box, select HTTP DoS Protection check box.

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.
Defining an HTTP DoS Policy

After you enable HTTP DoS protection, you next create a policy.

Note: Before changing the default setting for cltDetectRate, see Tuning the Client Detection/JavaScript Challenge Response Rate.

To configure a HTTP DoS policy by using the NetScaler command line

At the NetScaler 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
>```

Parameters for defining an HTTP DoS policy

name

A name for your HTTP DoS 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 sign (@), equals (=), colon (:), and underscore (_) symbols. You should choose a name that helps identify the type of action.

qdepth

An integer that represents the maximum number of connections that can be placed in the queue at one time.
cltDetectRate

An integer that represents the percentage of traffic to which the HTTP DoS policy should be applied.

To configure an HTTP DoS policy by using the configuration utility

1. In the navigation pane, expand Protection Features, and then click 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. The contents of the dialog box correspond to the parameters described in "Parameters for defining an HTTP DoS policy" as follows (asterisk indicates a required parameter):
   - Name*—name (You cannot change the name of an existing policy.)
   - QDepth*—qdepth
   - Client Detect Rate—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 NetScaler command line

At the NetScaler 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
  .
  .
```
Parameters for configuring an HTTP DoS service

name

A name for your service. 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 sign (@), equals (=), colon (:), and underscore (_) symbols. You should chose a name that helps identify the traffic this service will handle.

IP

The IP of the server that the service represents.

serverName

The FQDN of the server that the service represents.

port

The port on which your service will listen. This is normally port 80 (for HTTP) or port 443 (for HTTPS).

maxClient

The maximum number of clients.

maxReq

The maximum number of requests that can be sent on a persistent connection to the service.
To configure an HTTP DoS service by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **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*—name (You cannot change the name of an existing service.)
   - Server*—IP or serverName (Specify one or the other, not both.)
   - Port*—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*—maxClient
   - Max Requests*—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 NetScaler command line

At the NetScaler 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

```bash
> 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 NetScaler command line

At the NetScaler 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
To bind the monitor and policy to the service by using the configuration utility

1. In the navigation pane, expand Load Balancing, and then click 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, select a policy from the Available Policies list, and then click Add. The policy appears in the Configured Policies list.

5. 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. 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.