

# **BIG-IP<sup>®</sup> Local Traffic Management: Getting Started with Policies**

Version 12.1





# Table of Contents

<b>Introducing Local Traffic Policies.....</b>	<b>5</b>
About Local Traffic Policies.....	5
About local traffic policy matching.....	5
About strategies for local traffic policy matching.....	5
About rules for local traffic policy matching.....	6
About logical operators for conditions and rules.....	7
About conditions for local traffic policy matching.....	7
About actions for a local traffic policy rule .....	10
About Tcl command substitutions.....	12
About options for conditions and actions.....	12
Common tmsh commands for local traffic policies.....	15
Task summary for local traffic policies.....	15
Creating a draft local traffic policy.....	16
Publishing a local traffic policy.....	16
Modifying a published local traffic policy.....	17
Reordering local traffic draft policy rules.....	17
Associating a published local traffic policy with a virtual server.....	18
Cloning a local traffic policy.....	18
Creating a user-defined local traffic policy matching strategy.....	19
Deleting a local traffic policy.....	19
<b>Local Traffic Policies Examples.....</b>	<b>21</b>
Example: Preventing a Nimda worm attack .....	21
Creating a policy to prevent a Nimda worm attack: video example.....	21
Preventing a Nimda worm attack: tmsh example.....	21
Preventing a Nimda worm attack: iRules example.....	22
Example: Using selective compression .....	22
Creating a policy to support selective compression: video example.....	23
Selective compression: tmsh example.....	23
Selective compression: iRules example.....	24
Example: Preventing a spoof of an x-forwarded-for request .....	24
Creating a policy to prevent a spoof of an x-forwarded-for request: video example.....	24
Preventing a spoof of an x-forwarded-for request: tmsh example.....	25
Preventing a spoof of an x-forwarded-for request: iRules example.....	26
Example: Mitigating shellshock .....	26
Creating a policy to mitigate a shellshock attack: video example.....	26
Mitigating shellshock: tmsh example.....	27
Mitigating shellshock: iRules example.....	28

<b>Legal Notices.....</b>	<b>29</b>
Legal notices.....	29

# Introducing Local Traffic Policies

---

## About Local Traffic Policies

---

The BIG-IP® system provides Local Traffic Policies that simplify the way in which you can manage traffic associated with a virtual server. Using policies involves three basic steps: you create a draft policy, publish the policy, and then associate the published policy with a virtual server. Each policy includes a matching strategy for the specified rules, as well as conditions and actions configured within each rule, to manage traffic.

**Note:** Local Traffic Policies that have been upgraded from BIG-IP software version 12.0, or earlier, appear in the Published Policies list.

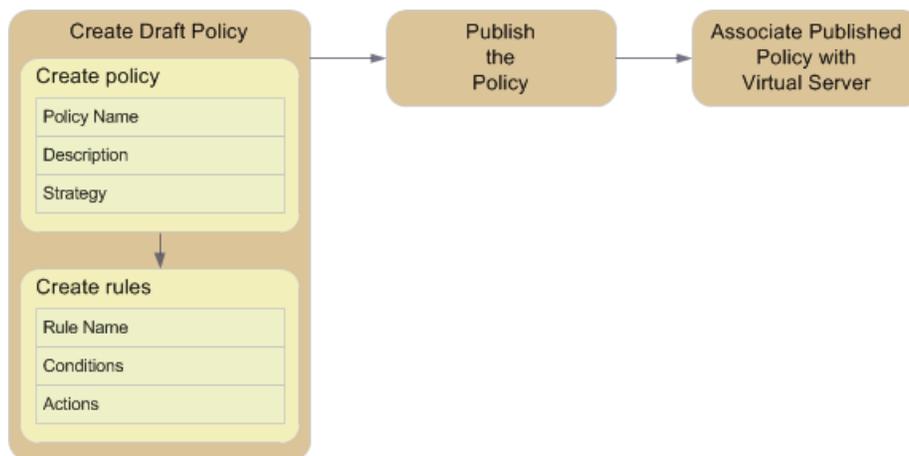


Figure 1: Basic steps for creating and using policies

## About local traffic policy matching

---

BIG-IP® local traffic policies comprise a prioritized list of rules that match defined conditions and run specific actions, which you can associate with a virtual server that directs traffic accordingly. For example, you might create a policy that determines whether a client is using a mobile device, and then redirects its requests to the applicable mobile web site's URL.

## About strategies for local traffic policy matching

---

Each BIG-IP® local traffic policy requires a matching strategy to determine which rule applies if more than one rule matches.

The BIG-IP local traffic policies provide three predefined policy matching strategies: a first-match, best-match, and all-match strategy. Each policy matching strategy prioritizes rules according to the rule's position within the Rules list.

As needed, you can create a user-defined best-match strategy to customize the precedence (order of preference) of added operands and selectors. For example, to meet your preferred operand and selector combinations, you might create a user-defined best-match strategy that changes the precedence of added operands and selectors, compared to the predefined best-match strategy.

---

*Note:* In a best-match or first-match strategy, a rule without conditions becomes the default rule, when the rule is the last entry in the Rules list.

---

**Table 1: Policy matching strategies**

Matching strategy	Description
all-match strategy	<p>An <i>all-match strategy</i> starts the actions for all rules in the Rules list that match.</p> <hr/> <p><i>Note:</i> In an <i>all-match strategy</i>, when multiple rules match, but specify conflicting actions, only the action of the best-match rule is implemented. A best-match rule can be the lowest ordinal, the highest priority, or the first rule that matches in the Rules list.</p>
best-match strategy	<p>A <i>best-match strategy</i> selects and starts the actions of the rule in the Rules list with the best match, as determined by the following factors.</p> <ol style="list-style-type: none"> <li>1. A best-match strategy selects the rule with the most conditions, ignoring details about the conditions.</li> <li>2. If a rule with the most conditions is not determined, then the best-match strategy selects the rule with the highest priority condition types. The best-match strategy sorts the condition types, highest priority first, comparing one at a time until a higher priority is found. For example, a priority sequence of 0,1,3,4,6 wins over 0,1,3,5,7 because 4 is a higher priority than 5.</li> <li>3. If a rule with the highest priority condition types is not determined, then the best-match strategy selects the rule with equal match types over other match types, such as starts-with, ends-with, or contains, and processes according to condition type priority.</li> <li>4. If a rule of equal match types is not determined, then the best-match strategy uses an ordinal (the precedence of the operand).</li> </ol> <hr/> <p><i>Note:</i> In a best-match strategy, when multiple rules match and specify an action, conflicting or otherwise, only the action of the best-match rule is implemented. A best-match rule can be the lowest ordinal, the highest priority, or the first rule that matches in the Rules list.</p>
first-match strategy	<p>A <i>first-match strategy</i> starts the actions for the first rule in the Rules list that matches.</p>

## About rules for local traffic policy matching

---

BIG-IP® local traffic policy *rules* match defined conditions and start specific actions. You can create a policy with rules that are as simple or complex as necessary, based on the passing traffic. For example, a rule might simply determine that a client's browser is a Chrome browser that is not on an administrator

network, and restrict access to certain administrative tools. Or a rule might determine that a request URL starts with `/video`, that the client is a mobile device, and that the client's subnet does not match `172.27.56.0/24`, and then that the request to a video file is designed for mobile devices on a Content Delivery Network (CDN).

## About logical operators for conditions and rules

Local traffic policy rules provide you with different types of logical operators for matching conditions, which are determined by the order and configuration of the conditions within and between the rules. The different types of logical operators that you can configure are AND logical operators for conditions within a rule, and OR logical operators for values within a condition and for conditions between rules. When AND logical operators apply, then all logical operators must match the matching strategy. When OR logical operators apply, then any logical operator must match the matching strategy.

### AND logical operators for conditions within a rule

When you create a rule, you can configure two or more conditions that use AND logic within that rule. For example, you can create Rule1 with two conditions, a and b, which use AND logic when Rule1 is used by the matching strategy. This means that all conditions within a rule must succeed in order to be used by a matching strategy.

### OR logical operators for values within a condition and for conditions between rules

When you configure multiple values within a condition, OR logic determines if any matching value within the condition succeeds. For example, you can create a condition configured with two or more values. The matching strategy uses OR logic to determine if any configured value matches.

Similarly, when you create two or more rules, you can configure each rule with applicable conditions that use OR logic between the rules. For example, you can create Rule1 with a set of conditions, and Rule 2 with another set of conditions. The matching strategy uses OR logic to determine if a rule matches.

### Examples

These examples show the logical operation of three conditions (a, b, and c) and two rules (Rule1 and Rule2).

In this first example, consider the following scenario, where you want to match condition a or b, and c ((a | b) & c). You can configure this logic by creating Rule1 to use conditions a and c (a & c), and Rule 2 to use conditions b and c (b & c). The result is when Rule1 matches the strategy, conditions a and c (a & c) are used, or when Rule 2 matches, conditions b and c (b & c) are used.

In this second example, consider the scenario where you want to match conditions a and b, or c ((a & b) | c). You can configure this logic by creating Rule1 to use conditions a and b, and Rule2 to use condition c. The result is when Rule1 matches the strategy, both conditions a and b are used, or when Rule2 matches the strategy, condition c is used.

## About conditions for local traffic policy matching

The *conditions* for a local traffic policy rule define the necessary criteria that must be met in order for the rule's actions to be applied. For example, a policy might include the following condition type and settings, which, when met by a request, would allow the rule's specified actions to be applied.

Option	Setting
Condition Type	HTTP Host

Option	Setting
Selector	host
Condition	is
Values	www.siterequest.com

You can apply one or more conditions to a rule, as needed.

**Table 2: Conditions for local traffic policy matching**

Condition Type	Description
<b>Client SSL</b>	<p>Inspects the properties of the SSL connection on the client side of the device.</p> <ul style="list-style-type: none"> <li>• <b>cipher</b>. Specifies the cipher name.</li> <li>• <b>cipher bits</b>. Specifies the cipher strength by means of the number of bits.</li> <li>• <b>protocol</b>. Specifies the SSL protocol name.</li> </ul>
<b>CPU Usage</b>	<p>Specifies a condition that is determined by CPU usage during 15-second, 1-minute, or 5-minute intervals.</p>
<b>Geo. IP</b>	<p>Specifies a condition that is based on the properties of the geographical location of the IP address.</p> <ul style="list-style-type: none"> <li>• <b>continent</b>. Matches a two-character continent code, for example, AF (Africa), AN (Antarctica), AS (Asia), OC (Oceania), NA (North America), or SA (South America).</li> <li>• <b>country code</b>. Matches a two-character country code, as defined in ISO-3166-2.</li> <li>• <b>country name</b>. Matches the full name of a country.</li> <li>• <b>isp</b>. Matches the Internet Service Provider associated with the address.</li> <li>• <b>organization</b>. Matches the organization associated with the address.</li> <li>• <b>region code</b>. Matches the abbreviation of a state, province, or country-specific region.</li> <li>• <b>region name</b>. Matches the full name of a state, province, or country-specific region.</li> </ul>
<b>HTTP Basic Auth.</b>	<p>Inspects the username and password specified for basic authentication for the HTTP request.</p> <ul style="list-style-type: none"> <li>• <b>password</b>. Matches the basic authentication password.</li> <li>• <b>username</b>. Matches the basic authentication username.</li> </ul>
<b>HTTP Cookie</b>	<p>Inspects the Cookie header of an HTTP request.</p>
<b>HTTP Header</b>	<p>Matches any HTTP header.</p>
<b>HTTP Host</b>	<p>Matches the host of an HTTP request.</p> <ul style="list-style-type: none"> <li>• <b>host</b>. Matches the host name.</li> <li>• <b>port</b>. Matches the port number.</li> <li>• <b>full string</b>. Matches the full host header string.</li> </ul>
<b>HTTP Method</b>	<p>Inspects the HTTP method for the request, for example, GET, POST, or HEAD.</p>
<b>HTTP Referer</b>	<p>Inspects the HTTP Referer header or parts of the URI.</p> <ul style="list-style-type: none"> <li>• <b>extension</b>. Matches the document extension, for example, <code>.cgi</code>.</li> <li>• <b>host</b>. Matches the DNS host name or IP address.</li> <li>• <b>path</b>. Matches the URI path, for example, <code>/path</code>.</li> </ul>

Condition Type	Description
	<ul style="list-style-type: none"> <li>• <b>path segment.</b> Matches the path segment by numerical index.</li> <li>• <b>port.</b> Matches the numeric port number, for example, 80.</li> <li>• <b>query parameter.</b> Matches the value of the query parameter by name.</li> <li>• <b>query string.</b> Matches the full query string, for example, a=b&amp;c=d.</li> <li>• <b>scheme.</b> Matches the scheme, for example, http, https, or ftp.</li> <li>• <b>unnamed query parameter.</b> Matches the value of the query parameter by numerical index.</li> <li>• <b>full string.</b> Matches the full URI, for example, <code>http://example.com/path/to/page.cgi?a=b&amp;c=d</code>.</li> </ul>
<b>HTTP Set Cookie</b>	<p>Inspects the Set-Cookie header of an HTTP response.</p> <ul style="list-style-type: none"> <li>• <b>domain.</b> Matches the value of the domain specified by the named cookie.</li> <li>• <b>expiry.</b> Matches the time when validity of the named cookie expires in RFC 6265 format (<i>wdy, DD Mon YYYY HH:MM:SS GMT</i>).</li> <li>• <b>path.</b> Matches the path of the named cookie.</li> <li>• <b>value.</b> Matches the value of the named cookie specified by the parameter.</li> <li>• <b>version.</b> Matches the version of the named cookie.</li> </ul>
<b>HTTP Status</b>	<p>Inspects the status of the HTTP response.</p> <ul style="list-style-type: none"> <li>• <b>code.</b> Matches the numeric HTTP response status code.</li> <li>• <b>text.</b> Matches the HTTP response status string, for example, <i>Authentication Required</i>.</li> <li>• <b>full string.</b> Matches the full HTTP status response, including the code and text.</li> </ul>
<b>HTTP URI</b>	<p>Inspects the URI on a request and matches parts of or the entire URI.</p> <ul style="list-style-type: none"> <li>• <b>extension.</b> Matches the file extension in the URI, for example, <i>html</i> or <i>cgi</i>.</li> <li>• <b>host.</b> Matches the host name in the URI.</li> <li>• <b>path.</b> Matches the URI path.</li> <li>• <b>path segment.</b> Matches a part of the URI path by a numeric index.</li> <li>• <b>port.</b> Matches the port number in the URI.</li> <li>• <b>query parameter.</b> Matches the value of the named query parameter from the query string.</li> <li>• <b>query string.</b> Matches the text specified in the query string.</li> <li>• <b>scheme.</b> Matches the scheme, for example, <i>http</i>, <i>https</i>, <i>ftp</i>, or <i>file</i>.</li> <li>• <b>unnamed query parameter.</b> Matches the value of a query parameter by a numeric index instead of by a name index.</li> <li>• <b>full string.</b> Matches a specified full text string.</li> </ul>
<b>HTTP User Agent</b>	<p>Specifies a condition that is based upon the User Agent header.</p> <ul style="list-style-type: none"> <li>• <b>browser type.</b> Matches the browser name or type.</li> <li>• <b>browser version.</b> Matches the browser version string.</li> <li>• <b>device make.</b> Matches the make of the device.</li> <li>• <b>device model.</b> Matches the model of the device.</li> <li>• <b>token.</b> Matches a string associated with a specified parameter.</li> </ul>
<b>HTTP Version</b>	<p>Inspects the version of an HTTP request or response.</p> <ul style="list-style-type: none"> <li>• <b>major.</b> Matches the numeric major part of the HTTP version.</li> <li>• <b>minor.</b> Matches the numeric minor part of the HTTP version.</li> <li>• <b>protocol.</b> Matches the HTTP protocol.</li> </ul>

Condition Type	Description
	<ul style="list-style-type: none"> <li>• <b>full string.</b> Matches the full version string.</li> </ul>
<b>SSL Certificate</b>	Inspects the properties of an SSL certificate.
<b>SSL Extension</b>	Inspects the SSL extensions that are negotiated during the <code>HELLO</code> phase. <ul style="list-style-type: none"> <li>• <b>alpn.</b> Matches the application layer protocol negotiation.</li> <li>• <b>npn.</b> Matches the next protocol negotiation.</li> <li>• <b>server name.</b> Matches the server name indication.</li> </ul>
<b>TCP</b>	Inspects and matches the parameters associated with TCP connections. <ul style="list-style-type: none"> <li>• <b>address.</b> Matches the specified IP address. The IP address is the address associated with the external interface remote end of the connection.</li> <li>• <b>mss.</b> Compares the TCP maximum segment size on the external network interface.</li> <li>• <b>port.</b> Matches the port number. The specified port is associated with the external interface, remote end of the connection.</li> <li>• <b>route domain.</b> Compares traffic with the specified route domain number on the external network interface.</li> <li>• <b>rtt.</b> Inspects the round-trip time on the external network interface.</li> <li>• <b>vlan.</b> Compares traffic with the specified vlan on the external network interface.</li> <li>• <b>vlan id.</b> Compares traffic with the specified vlan ID number on the external interface.</li> </ul>
<b>WebSocket</b>	Specifies a condition based upon the properties of a websocket's connection. <ul style="list-style-type: none"> <li>• <b>extension.</b> Matches the value of the <code>Sec-WebSocket-Extensions</code> header.</li> <li>• <b>key.</b> Matches the value of the <code>masking-key</code>.</li> <li>• <b>protocol.</b> Matches the value of the <code>Sec-WebSocket-Protocol</code> header.</li> <li>• <b>version.</b> Matches the value of the <code>Sec-WebSocket-Version</code> header.</li> </ul>

## About actions for a local traffic policy rule

The *actions* for a local traffic policy rule determine how traffic is handled. For example, actions for a rule could include the following ways of handling traffic.

- Blocking traffic
- Rewriting a URL
- Logging traffic
- Adding a specific header
- Redirecting traffic to a different pool member
- Selecting a specific Web Application policy

**Table 3: Actions for local traffic policy matching**

Action Type	Description
<b>Enable</b>	Enables the following actions. <ul style="list-style-type: none"> <li>• <b>cache.</b> Enables caching for a connection.</li> <li>• <b>compression.</b> Enables compression for a connection.</li> </ul>

Action Type	Description
	<ul style="list-style-type: none"> <li>• <b>decompression.</b> Enables decompression for a connection.</li> <li>• <b>http.</b> Enables HTTP filter processing.</li> <li>• <b>request adapt.</b> Enables request adaptation, optionally sending traffic to a specified internal virtual server.</li> <li>• <b>response adapt.</b> Enables response adaptation, optionally sending traffic to a specified internal virtual server.</li> <li>• <b>server ssl.</b> Enables encrypted connections to backend servers.</li> <li>• <b>websocket.</b> Enables WebSocket processing.</li> </ul>
<b>Disable</b>	<p>Disables the following actions.</p> <ul style="list-style-type: none"> <li>• <b>cache.</b> Disables caching for a connection.</li> <li>• <b>compression.</b> Disables compression for a connection.</li> <li>• <b>decompression.</b> Disables decompression for a connection.</li> <li>• <b>http.</b> Disables HTTP filter processing.</li> <li>• <b>LTM policy.</b> Disables LTM® Policy processing on a request-by-request basis.</li> <li>• <b>persistence.</b> Disables persistence.</li> <li>• <b>request adapt.</b> Disables request adaptation, optionally sending traffic to a specified internal virtual server.</li> <li>• <b>response adapt.</b> Disables response adaptation, optionally sending traffic to a specified internal virtual server.</li> <li>• <b>server ssl.</b> Disables encrypted connections to backend servers.</li> <li>• <b>websocket.</b> Disables WebSocket processing.</li> </ul>
<b>Forward traffic</b>	<p>Controls where a connection is forwarded.</p> <ul style="list-style-type: none"> <li>• <b>node.</b> Forwards the connection to the specified node.</li> <li>• <b>pool.</b> Forwards the connection to the specified pool.</li> <li>• <b>virtual server.</b> Forwards the connection to the specified virtual server.</li> </ul>
<b>Insert</b>	<p>Inserts an HTTP header into the request or response.</p> <ul style="list-style-type: none"> <li>• <b>http cookie.</b> Inserts an HTTP Cookie header into the response.</li> <li>• <b>http header.</b> Inserts an HTTP header into the request or response.</li> <li>• <b>http referer.</b> Inserts an HTTP Referer header into the request.</li> <li>• <b>http set cookie.</b> Inserts an HTTP Set-Cookie header into the response.</li> </ul>
<b>Remove</b>	<p>Removes an HTTP header from the request or response.</p> <ul style="list-style-type: none"> <li>• <b>http cookie.</b> Removes an HTTP Cookie header from the response.</li> <li>• <b>http header.</b> Removes an HTTP header from the request or response.</li> <li>• <b>http referer.</b> Removes an HTTP Referer header from the request.</li> <li>• <b>http set cookie.</b> Removes an HTTP Set-Cookie header from the response.</li> </ul>
<b>Replace</b>	<ul style="list-style-type: none"> <li>• <b>http header.</b> Replaces the HTTP header in the request or response.</li> <li>• <b>http host.</b> Replaces the HTTP Host header in the request.</li> <li>• <b>http referer.</b> Replaces the HTTP Referer header in the request.</li> <li>• <b>http uri.</b> Replaces the HTTP URI, path, or string in the request.</li> </ul>
<b>Redirect</b>	<p>Redirects traffic to a different URL.</p>
<b>Reset traffic</b>	<p>Resets the connection.</p>
<b>Log</b>	<p>Writes messages to the local or remote system log.</p>

Action Type	Description
<b>Persist session</b>	<p>Controls how a connection is persisted.</p> <ul style="list-style-type: none"> <li>• <b>carp</b>. Persists the connection by using a Cache Array Routing Protocol algorithm.</li> <li>• <b>cookie hash</b>. Persists the connection by using a cookie hash method.</li> <li>• <b>cookie insert</b>. Persists the connection by using a cookie insert method.</li> <li>• <b>cookie passive</b>. Persists the connection by using a cookie passive method.</li> <li>• <b>cookie rewrite</b>. Persists the connection by using a cookie rewrite method.</li> <li>• <b>destination address</b>. Persists the connection based on the destination IP address.</li> <li>• <b>hash</b>. Persists the connection by using a cookie hash method.</li> <li>• <b>source address</b>. Persists the connection based on the source IP address.</li> <li>• <b>universal</b>. Persists the connection based on a user-defined key.</li> </ul>
<b>Set variable</b>	Sets a Tcl variable in the runtime environment.

### About Tcl command substitutions

Certain BIG-IP® local traffic policy actions support Tcl command substitutions, giving you significant flexibility in configuring policies. Tcl command substitutions provide quick, read-only access to immediately available runtime data, such as information about a current request's URI, or a header or cookie in the request or response.

---

**Important:** Any Tcl command that requires a delay, for example, the `after` command, or that requires waiting for results from a request outside of the Traffic Management Microkernel (TMM), is not supported and might not succeed.

---

#### Considerations when using Tcl command substitutions

When using Tcl command substitutions, the following guidelines apply.

- Memory and CPU capacity determine a maximum number of rules for active policies; however, excessive Tcl command substitutions can degrade performance.
- Tcl command substitutions are primarily intended for reading and returning data.

#### Tcl command substitution example

The following Tcl command is an example of a Tcl command substitution that can be used within a policy.

```
"tcl:[HTTP::uri]"
```

---

**Note:** Each Tcl command must include a prefix of `tcl:`. If the `tcl:` prefix is omitted, the command is interpreted as a plain string.

---

### About options for conditions and actions

You can apply options to conditions and actions, as determined by the selected condition or action type. For example, you might want to constrain a condition based on the case-sensitivity of a string, which is easily applied by selecting the applicable option for that condition.

Table 4: Options for conditions

Condition Type	Options
Client SSL	<ul style="list-style-type: none"> <li>• Skip this condition if it is missing from the request.</li> <li>• Use case sensitive string comparison.</li> </ul>
CPU Usage	Not applicable.
Geo. IP	<ul style="list-style-type: none"> <li>• Apply to traffic on remote/local side of external/internal interface.</li> <li>• Skip this condition if it is missing from the request.</li> <li>• Use case sensitive string comparison.</li> </ul>
HTTP Basic Auth.	<ul style="list-style-type: none"> <li>• Skip this condition if it is missing from the request.</li> <li>• Use case sensitive string comparison.</li> </ul>
HTTP Cookie	<ul style="list-style-type: none"> <li>• Skip this condition if it is missing from the request.</li> <li>• Use case sensitive string comparison.</li> </ul>
HTTP Header	<ul style="list-style-type: none"> <li>• Skip this condition if it is missing from the request.</li> <li>• Use case sensitive string comparison.</li> </ul>
HTTP Host	<ul style="list-style-type: none"> <li>• Skip this condition if it is missing from the request.</li> <li>• Use case sensitive string comparison.</li> </ul>
HTTP Method	<ul style="list-style-type: none"> <li>• Use case sensitive string comparison.</li> </ul>
HTTP Referer	<ul style="list-style-type: none"> <li>• Use <b>normalized URI</b>. The normalization applied includes a lower case scheme and URI, including host names, removing a trailing period (.) character, and percent encoding. For percent encoding, bytes not allowed in a URI are normalized to their percent encoded representation.</li> <li>• Skip this condition if it is missing from the request.</li> <li>• Use case sensitive string comparison.</li> </ul>
HTTP Set Cookie	<ul style="list-style-type: none"> <li>• Skip this condition if it is missing from the request.</li> <li>• Use case sensitive string comparison.</li> </ul>
HTTP Status	Not applicable.
HTTP URI	<ul style="list-style-type: none"> <li>• Use <b>normalized URI</b>. The normalization applied includes a lower case scheme and URI, including host names, removing a trailing period (.) character, and percent encoding. For percent encoding, bytes not allowed in a URI are normalized to their percent encoded representation.</li> <li>• Skip this condition if it is missing from the request.</li> <li>• Use case sensitive string comparison.</li> </ul>
HTTP User Agent	<ul style="list-style-type: none"> <li>• Skip this condition if it is missing from the request.</li> <li>• Use case sensitive string comparison.</li> </ul>
HTTP Version	<ul style="list-style-type: none"> <li>• Use case sensitive string comparison.</li> </ul> <hr/> <p><i>Note: Applies only to protocol and full string settings.</i></p> <hr/>
SSL Certificate	<ul style="list-style-type: none"> <li>• Skip this condition if it is missing from the request.</li> <li>• Use case sensitive string comparison.</li> </ul>
SSL Extension	<ul style="list-style-type: none"> <li>• Skip this condition if it is missing from the request.</li> <li>• Use case sensitive string comparison.</li> </ul>

Condition Type	Options
TCP	<ul style="list-style-type: none"> <li>• Apply to traffic on remote/local side of external/internal interface.</li> <li>• Use case sensitive string comparison.</li> </ul>
WebSocket	<ul style="list-style-type: none"> <li>• Skip this condition if it is missing from the request.</li> <li>• Use case sensitive string comparison.</li> </ul>

Table 5: Options for actions

Action Type	Options
Enable	Not applicable.
Disable	Not applicable.
Forward traffic	Override default forward action using: <ul style="list-style-type: none"> <li>• SNAT and <b>disable/automap</b>.</li> <li>• SNAT <b>Pool</b> and pool name.</li> <li>• VLAN: <b>(none)</b></li> <li>• VLAN: VLAN Name</li> </ul>
Insert	Not applicable.
Remove	Not applicable.
Replace	Not applicable.
Redirect	Not applicable.
Reset traffic	Not applicable.
Log	Facility: <ul style="list-style-type: none"> <li>• <b>local0</b> (default)</li> <li>• <b>local1</b></li> <li>• <b>local2</b></li> <li>• <b>local3</b></li> <li>• <b>local4</b></li> <li>• <b>local7</b></li> <li>• <b>syslog</b></li> <li>• <b>security</b></li> <li>• <b>autopriv</b></li> <li>• <b>daemon</b></li> <li>• <b>kern</b></li> <li>• <b>cron</b></li> <li>• <b>ftp</b></li> <li>• <b>lpr</b></li> <li>• <b>mail</b></li> <li>• <b>news</b></li> <li>• <b>uucp</b></li> </ul> Priority: <ul style="list-style-type: none"> <li>• <b>debug</b></li> <li>• <b>info</b> (default)</li> <li>• <b>notice</b></li> <li>• <b>warning</b></li> </ul>

Action Type	Options
	<ul style="list-style-type: none"> <li>error</li> </ul> <p>Send log message to remote server:</p> <ul style="list-style-type: none"> <li>Host</li> <li>Port</li> </ul>
Persist session	Not applicable.
Set variable	Not applicable.

## Common tmsh commands for local traffic policies

You can use tmsh commands with policies, as necessary. Common commands include those in the table.

**Table 6: Common tmsh commands for local traffic policies**

Description	tmsh Command
Create a draft policy.	<code>(tmsh)# create ltm policy /Common/Drafts/policy_name strategy first-match</code>
Publish a draft policy.	<code>(tmsh)# publish ltm policy /Common/Drafts/policy_name</code>
List all draft policies.	<code>(tmsh)# show ltm policy /Common/Drafts/*</code>
List all published policies.	<code>(tmsh)# show ltm policy</code>
List configuration details for a draft policy.	<code>(tmsh)# list ltm policy /Common/Drafts/policy_name</code>
List configuration details for a published policy.	<code>(tmsh)# list ltm policy policy_name</code>

## Task summary for local traffic policies

Perform these tasks to create, publish, and manage local traffic policies.

*Creating a draft local traffic policy*

*Publishing a local traffic policy*

*Modifying a published local traffic policy*

*Reordering local traffic draft policy rules*

*Associating a published local traffic policy with a virtual server*

*Cloning a local traffic policy*

*Creating a user-defined local traffic policy matching strategy*

### *Deleting a local traffic policy*

## Creating a draft local traffic policy

You can use BIG-IP® local traffic policy matching to direct traffic in accordance with rules, which are applied as determined by the specified strategy, conditions, and actions.

---

***Note:** Local traffic policies that have been upgraded from BIG-IP software version 12.0, or earlier, appear in the Published Policies list.*

---

1. On the Main tab, click **Local Traffic > Policies > Policy List**.  
The Policy List Page screen opens.
2. Click **Create**.  
The New Policy screen opens.
3. In the **Policy Name** field, type a unique name for the policy.
4. In the **Description** field, type a description for the policy.
5. From the **Strategy** list, select a matching strategy.
6. Click **Create Policy**.  
The policy is created and the Rules area appears.
7. In the Rules area, click **Create**.
8. In the Match all of the following conditions area, click +.
9. From the **Client SSL** list, select a condition type, and configure the applicable settings and available options.
10. In the Match all of the following conditions area, click + to add an additional condition, as necessary, and configure the applicable settings and available options.
11. In the Do the following when the traffic is matched area, click +.
12. From the **Enable** list, select an action type, and configure the applicable settings and available options.
13. In the Do the following when the traffic is matched area, click + to add an additional action, as necessary, and configure the applicable settings and available options.
14. Click **Save**.

The policy appears in the Draft Policies list.

## Publishing a local traffic policy

Before you can publish a local traffic policy, a draft policy must be available.

After you create a draft local traffic policy, you need to publish the policy, and then associate the published policy with a virtual server.

---

***Note:** Local traffic policies that have been upgraded from BIG-IP software version 12.0, or earlier, appear in the Published Policies list.*

---

1. On the Main tab, click **Local Traffic > Policies > Policy List**.  
The Policy List Page screen opens.
2. Select the check box of the draft policy to publish.
3. Click **Publish**.

The draft policy is removed from the Draft Policies list, and the modified published policy appears in the Published Policies list.

The draft local traffic policy is published and available to assign to a virtual server.

## Modifying a published local traffic policy

You must have a published local traffic policy available, before you can modify its settings.

You can modify a published local traffic policy, by creating a draft policy from the published policy, making any necessary changes, and then publishing the modified draft policy. You cannot modify a published policy directly; you can only modify a draft policy. If the published local traffic policy is associated with a virtual server, the modified policy settings are updated in the associated virtual server.

1. On the Main tab, click **Local Traffic > Policies > Policy List**.

The Policy List Page screen opens.

2. Click the name of a published policy.

3. Click **Create Draft**.

A draft policy of the same name appears in the Draft Policies list.

---

*Note:* When you publish a policy, the draft policy is removed from the Draft Policies list.

---

4. Click the name of the draft policy.

5. Modify the applicable settings.

6. Click **Save**.

7. Click **Save Draft**.

The Policy List Page screen opens.

8. Select the check box of the draft policy to publish.

9. Click **Publish**.

The draft policy is removed from the Draft Policies list, and the modified published policy appears in the Published Policies list.

The published local traffic policy is updated.

## Reordering local traffic draft policy rules

Before you can reorder local traffic policy rules, there must be a draft policy with multiple rules available.

---

*Note:* You cannot reorder rules in a published policy.

---

You can reorder rules within a draft policy, as needed.

1. On the Main tab, click **Local Traffic > Policies > Policy List**.

The Policy List Page screen opens.

2. Click the name of a draft policy.

3. Click and drag the rule or rules that you want to reorder into the preferred sequence.

4. Click **Save Draft**.

The Policy List Page screen opens.

Rules in the draft local traffic policy appear in the sequence order that you configured.

### Associating a published local traffic policy with a virtual server

After you publish a local traffic policy, you associate that published policy with the virtual server created to handle application traffic.

1. On the Main tab, click **Local Traffic > Virtual Servers**.  
The Virtual Server List screen opens.
2. Click the name of the virtual server you want to modify.
3. On the menu bar, click **Resources**.
4. In the Policies area, click the **Manage** button.
5. For the **Policies** setting, from the **Available** list, select the local traffic policy you previously created, and move it to the **Enabled** list.
6. Click **Finished**.

The published policy is associated with the virtual server.

### Cloning a local traffic policy

You can clone (copy) either a draft or published BIG-IP® local traffic policy to create a different draft policy with the same settings of the original policy. After you clone the local traffic policy, you can modify it as necessary, publish it, and associate it with a virtual server.

1. On the Main tab, click **Local Traffic > Policies > Policy List**.  
The Policy List Page screen opens.
2. Click the name of a policy.
3. Click **Clone**.  
The **Policy Name** field becomes cleared.
4. In the **Policy Name** field, type a unique name for the policy.
5. Click **Create Policy**.  
The Draft Policy screen opens.
6. In the **Description** field, type a description for the policy.
7. From the **Strategy** list, select a matching strategy.
8. In the Rules area, click **Create**.
9. In the Match all of the following conditions area, click +.
10. From the **Client SSL** list, select a condition type, and configure the applicable settings and available options.
11. In the Match all of the following conditions area, click + to add an additional condition, as necessary, and configure the applicable settings and available options.
12. In the Do the following when the traffic is matched area, click +.
13. From the **Enable** list, select an action type, and configure the applicable settings and available options.
14. In the Do the following when the traffic is matched area, click + to add an additional action, as necessary, and configure the applicable settings and available options.

The policy appears in the Draft Policies list.

## Creating a user-defined local traffic policy matching strategy

You can create a new local traffic policy matching strategy, based on a best-match policy matching strategy type. A user-defined best-match strategy can customize the precedence (order of preference) of added operands and selectors, compared to the predefined best-match policy.

1. On the Main tab, click **Local Traffic > Policies > Strategy List**.  
The Strategy List screen opens.
2. Click **Create**.  
The New Strategy screen opens.
3. In the **Name** field, type a unique name for the strategy.
4. From the **Operands** list, select an operand, configure the applicable settings, and click **Add**.
5. Click **Finished**.

The new user-defined best-match policy matching strategy appears in the Strategy List screen.

## Deleting a local traffic policy

You can delete BIG-IP® local traffic policies when they become obsolete or are no longer used.

1. On the Main tab, click **Local Traffic > Policies > Policy List**.  
The Policy List Page screen opens.
2. Select the check box for each policy that you want to delete.
3. Click **Delete**.  
The Confirm delete? popup screen opens.
4. Click **OK**.

The system deletes the policies that you selected.



# Local Traffic Policies Examples

---

## Example: Preventing a Nimda worm attack

---

You can create a local traffic policy that prevents the Nimda worm attack. If the URL contains certain strings that are known to be associated with the Nimda worm, then the local traffic policy can use a forwarding action that resets the connection.

### Examples

*Creating a policy to prevent a Nimda worm attack: video example*

*Preventing a Nimda worm attack: tmsh example*

*Preventing a Nimda worm attack: iRules example*

## Creating a policy to prevent a Nimda worm attack: video example

You can associate a BIG-IP® local traffic policy to prevent a Nimda worm attack. The policy forwards a URL containing strings associated with the Nimda worm, and resets the connection. Watch the following video for an example of creating a local traffic policy and associating it with a virtual server.



*Watch how to create a policy to prevent a Nimda worm attack*

You can also visit our DevCentral™ YouTube™ channel to see this video. Use any of these ways:

- Click this URL: <https://www.youtube.com/watch?v=0qe-iyt4Bwg>.
- Copy and paste the above URL into your browser window.
- Use your browser to search for this video using the title *F5: Creating a local traffic policy to prevent a Nimda worm attack*.

## Preventing a Nimda worm attack: tmsh example

This topic provides a `tmsh` command to list the configured settings for a Nimda policy. The policy directs that if certain strings are known to be associated with the Nimda worm, the local traffic policy uses a forwarding action that resets the connection. This topic also provides a `tmsh` command to list the configured virtual server settings.

```
(tmsh)# list ltm policy Stop_Nimda
ltm policy StopNimda{
  controls { forwarding }
  description "This policy blocks the Nimda worm."
  last-modified 2016-03-02:11:46:00
  requires { http }
  rules {
    ClobberNimda {
      actions {
        0 {
          forward
          reset
        }
      }
    }
  }
}
```

```
        }
      }
      conditions {
        0 {
          http-uri
          query-string
          values { root.exe admin.dll cmd.exe }
        }
      }
    }
    status published
    strategy first-match
  }

(tmos.ltm.virtual)# list HTTP-VS1
ltm.virtual.HTTP-VS1{
  destination 10.10.0.21:http
  ip-protocol tcp
  mask 255.255.255.255
  policies {
    StopNimda { }
  }
  profiles {
    http { }
    tcp { }
  }
  source 0.0.0.0/0
  translate-address enabled
  translate-port enabled
  vs-index 2
}
```

### Preventing a Nimda worm attack: iRules example

This topic provides an example of iRules<sup>®</sup> code that is equivalent to a policy that protects against a Nimda worm attack. The iRule directs that if certain strings are known to be associated with the Nimda worm, the local traffic policy uses a forwarding action that resets the connection.

```
when HTTP_REQUEST {
  set uri [string tolower [HTTP::uri]]
  if { ($uri contains "cmd.exe") or ($uri contains "root.exe") or ($uri contains
    "admin.dll") } {
    discard
  }
}
```

### Example: Using selective compression

---

You can create a local traffic policy to support selective compression for types of content that can benefit from compression. For example, common text types (HTML, XML, and CSS style sheets) can realize performance improvements, especially across slow connections, if you compress them.

#### Examples

*Creating a policy to support selective compression: video example*

*Selective compression: tmsh example*

*Selective compression: iRules example*

## Creating a policy to support selective compression: video example

You can associate a BIG-IP® local traffic policy with a virtual server to support selective compression for types of content that can benefit from compression. For example, common text types (HTML, XML, and CSS style sheets) can realize performance improvements, especially across slow connections, if you compress them. Watch the following video for an example of creating a local traffic policy and associating it with a virtual server.



*Watch how to create a policy to support selective compression*

You can also visit our DevCentral™ YouTube™ channel to see this video. Use any of these ways:

- Click this URL: <https://youtu.be/d85swKvXS1w>.
- Copy and paste the above URL into your browser window.
- Use your browser to search for this video using the title *F5: Using Selective Compression*.

## Selective compression: tmsh example

This topic provides a `tmsh` command to list the configured settings for a Selective Compression policy, for types of content that can benefit from compression. For example, common text types (HTML, XML, and CSS style sheets) can realize performance improvements, especially across slow connections. This topic also provides a `tmsh` command to list the configured virtual server settings.

```
(tmsh)# list ltm policy SelectiveCompression
ltm policy SelectiveCompression{
  controls { compression }
  description "This policy compresses specified file types."
  last-modified 2016-03-02:11:46:00
  requires { http }
  rules {
    CompressFiles {
      actions {
        0 {
          compress
          response
          enable
        }
      }
      conditions {
        0 {
          http-header
          name Content-Type
          starts-with
          values { text/ }
        }
        1 {
          cpu-usage
          last-1min
          less-or-equal
          values {5}
        }
      }
    }
  }
  status published
  strategy first-match
}
```

```
(tmos.ltm.virtual)# list ltm virtual HTTP-VS2
ltm.virtual.HTTP-VS2{
  destination 10.10.0.31:http
  ip-protocol tcp
  mask 255.255.255.255
  policies {
    SelectCompression { }
  }
  profiles {
    http { }
    httpcompression { }
    tcp { }
  }
  source 0.0.0.0/0
  translate-address enabled
  translate-port enabled
  vs-index 3
}
```

### Selective compression: iRules example

This topic provides an example of iRules<sup>®</sup> code that is equivalent to a policy to support selective compression for types of content that can benefit from compression.

```
when HTTP_REQUEST {
  COMPRESS::disable
  if { [HTTP::header Content-Type] contains "text" } {
    log "Enabling compression for this request"
    COMPRESS::enable
  }
}
```

### Example: Preventing a spoof of an x-forwarded-for request

---

You can create a local traffic policy to prevent a spoof of an x-forwarded-for request. This is a security issue where attackers might attempt to thwart security by falsifying the IP address in a header, and pass it through the BIG-IP<sup>®</sup> system.

#### Examples

*Creating a policy to prevent a spoof of an x-forwarded-for request: video example*

*Preventing a spoof of an x-forwarded-for request: tmsh example*

*Preventing a spoof of an x-forwarded-for request: iRules example*

### Creating a policy to prevent a spoof of an x-forwarded-for request: video example

You can associate a BIG-IP<sup>®</sup> local traffic policy with a virtual server to prevent a spoof of an x-forwarded-for request. This is a security issue where attackers might attempt to thwart security by falsifying the IP address in a header, and pass it through the BIG-IP system. Watch the following video for an example of creating a local traffic policy and associating it with a virtual server.



*Watch how to create a policy to prevent a spoof of an x-forwarded-for request*

You can also visit our DevCentral™ YouTube™ channel to see this video. Use any of these ways:

- Click this URL: <https://youtu.be/QRQxjt4-e4k>.
- Copy and paste the above URL into your browser window.
- Use your browser to search for this video using the title *F5: Creating a local traffic policy to prevent a spoof of an x-forwarded-for request*.

When you have completed the task shown in the video, the policy is associated with a virtual server.

## Preventing a spoof of an x-forwarded-for request: tmsh example

This topic provides a tmsh command to list the configured settings for a policy to prevent a spoof of an x-forwarded-for request. This is a request where attackers might attempt to thwart security by falsifying the IP address in a header, and pass it through the BIG-IP® system. This topic also provides a tmsh command to list the configured virtual server settings.

```
(tmos)# list ltm policy PreventSpoofOfXFF
ltm policy SelectiveCompression{
  controls { compression }
  description "This policy prevents a spoof of an x-forwarded-for request."

  last-modified 2016-03-02:11:46:00
  requires { http }
  rules {
    StopSpoof {
      actions {
        0 {
          http-header
          replace
          name X-forWARDED-for
          value tcl:[IP::client_addr]
        }
      }
    }
  }
  status published
  strategy first-match
}

(tmos.ltm.virtual)# list ltm virtual HTTP-VS3
ltm.virtual.HTTP-VS3{
  destination 10.10.0.41:http
  ip-protocol tcp
  mask 255.255.255.255
  policies {
    PreventSpoofOfXFF { }
  }
  profiles {
    http { }
    tcp { }
  }
  source 0.0.0.0/0
  translate-address enabled
  translate-port enabled
  vs-index 4
}
```

## Preventing a spoof of an x-forwarded-for request: iRules example

This topic provides an example of `iRules` code that is equivalent to a policy that prevents a spoof of an `x-forwarded-for` request. This is a situation where attackers might attempt to thwart security by falsifying the IP address in a header, and pass it through the BIG-IP® system. This example replaces a request that includes an `x-forwarded-for` header with the actual client IP address.

```
when HTTP_REQUEST {
  set xff 0
  foreach x [HTTP::header names] {
    if { [string tolower $x] equals "x-forwarded-for" } {
      set xff 1
      HTTP::header remove $x
      HTTP::header insert X-FORWARDED-FOR [IP::client_addr]
    }
  }
  if { $xff == 0 } {
    HTTP::header insert X-FORWARDED-FOR [IP::client_addr]
  }
}
```

## Example: Mitigating shellshock

---

You can create a local traffic policy to mitigate shellshock. In *shellshock*, an Internet service misuses `bash` shell functionality to process requests that execute arbitrary commands, potentially giving an attacker unauthorized access. This example policy examines requests for an uncommon pattern of " () {" in the URI, to minimize the possibility of false-positive matches.

### Examples

*Creating a policy to mitigate a shellshock attack: video example*

*Mitigating shellshock: tmsh example*

*Mitigating shellshock: iRules example*

## Creating a policy to mitigate a shellshock attack: video example

A shellshock attack refers to a class of exploits that misuse the `bash` shell through a specifically crafted URL. You can associate a BIG-IP® local traffic policy with a virtual server to mitigate a shellshock attack, where the policy examines requests for a pattern of " () {" in the URI. Watch the following video for an example of creating a local traffic policy and associating it with a virtual server.



*Watch how to create a policy to mitigate a shellshock attack*

You can also visit our DevCentral™ YouTube™ channel to see this video. Use any of these ways:

- Click this URL: <https://youtu.be/qL98Xn1zB5U>.
- Copy and paste the above URL into your browser window.
- Use your browser to search for this video using the title *F5: Creating a local traffic policy to mitigate a shellshock attack*.

## Mitigating shellshock: tmsh example

This topic provides a `tmsh` command to list the configured settings for a Mitigating Shellshock policy. During this type of attack, a class of exploits misuse the `bash` shell through a specifically crafted URL. This topic also provides a `tmsh` command to list the configured virtual server settings.

```
(tmsh)# list ltm policy MitigatingShellshock
ltm policy MitigatingShellshock{
  controls { forwarding }
  description "This policy mitigates shellshock."
  last-modified 2016-03-02:11:46:00
  requires { http }
  rules {
    StopShellshock {
      actions {
        0 {
          log
          write
          facility local0
          message "tcl:Shellshock detected from [IP::client_addr],
blocked"
          priority info
        }
        1 {
          forward
          reset
        }
      }
      conditions {
        0 {
          http-uri
          query string
          contains
          values { "() {" }
        }
      }
    }
  }
  status published
  strategy first-match
}

(tmsh)# list ltm virtual HTTP-VS4
ltm.virtual.HTTP-VS4{
  destination 10.10.0.51:http
  ip-protocol tcp
  mask 255.255.255.255
  policies {
    MitigatingShellshock { }
  }
  profiles {
    http { }
    tcp { }
  }
  source 0.0.0.0/0
  translate-address enabled
  translate-port enabled
  vs-index 5
}
```

## Mitigating shellshock: iRules example

This topic provides an example of iRules<sup>®</sup> code that is equivalent to a policy to mitigate shellshock, where the policy examines requests for a pattern of " () {" in the URI.

```

when HTTP_REQUEST {
  set pattern "*() \{*";
  if { [string match $pattern [HTTP::uri]] } {
    log local0. "Detected CVE-2014-6271 attack from '[IP::client_addr]' in URI '[HTTP::uri]';";
    reject;
  } else {
    foreach header_name [HTTP::header names] {
      foreach header_value [HTTP::header values $header_name] {
        if { [string match $pattern $header_value] } {
          log local0. "Detected CVE-2014-6271 attack from '[IP::client_addr]' in HTTP Header $header_";
          reject;
          break;
        }
      }
    }
  }
}

when HTTP_REQUEST {
  if { [string match "*() \{*" [HTTP::request]] } {
    log local0. "Detected CVE-2014-6271 attack from '[IP::client_addr]'; URI = '[HTTP::uri]';";
    reject;
  }
}

```

# Legal Notices

---

## Legal notices

---

### **Publication Date**

This document was published on December 28, 2017.

### **Publication Number**

MAN-0612-00

### **Copyright**

Copyright © 2017, F5 Networks, Inc. All rights reserved.

F5 Networks, Inc. (F5) believes the information it furnishes to be accurate and reliable. However, F5 assumes no responsibility for the use of this information, nor any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent, copyright, or other intellectual property right of F5 except as specifically described by applicable user licenses. F5 reserves the right to change specifications at any time without notice.

### **Trademarks**

For a current list of F5 trademarks and service marks, see <http://www.f5.com/about/guidelines-policies/trademarks/>.

All other product and company names herein may be trademarks of their respective owners.

### **Patents**

This product may be protected by one or more patents indicated at: <https://f5.com/about-us/policies/patents>

### **Link Controller Availability**

This product is not currently available in the U.S.

### **Export Regulation Notice**

This product may include cryptographic software. Under the Export Administration Act, the United States government may consider it a criminal offense to export this product from the United States.

### **RF Interference Warning**

This is a Class A product. In a domestic environment this product may cause radio interference, in which case the user may be required to take adequate measures.

### **FCC Compliance**

This equipment has been tested and found to comply with the limits for a Class A digital device pursuant to Part 15 of FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This unit generates, uses, and

can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case the user, at his own expense, will be required to take whatever measures may be required to correct the interference.

Any modifications to this device, unless expressly approved by the manufacturer, can void the user's authority to operate this equipment under part 15 of the FCC rules.

### **Canadian Regulatory Compliance**

This Class A digital apparatus complies with Canadian ICES-003.

### **Standards Compliance**

This product conforms to the IEC, European Union, ANSI/UL and Canadian CSA standards applicable to Information Technology products at the time of manufacture.

# Index

## A

all-match strategy  
about 5

## B

best-match strategy  
about 5

## C

Centralized Policy Matching  
introduction 5

## D

draft local traffic policy  
reordering rules 17

## F

first-match strategy  
about 5

## I

iRules example  
for mitigating shellshock attack 28  
for preventing a spoof of an x-forwarded-for request 26  
for preventing Nimda worm attack 22  
for selective compression 24

## L

local traffic policies  
cloning 18  
creating 16  
example, mitigating shellshock attack 27–28  
example, mitigating shellshock video 26  
example, preventing a spoof of an x-forwarded-for request 26  
example, preventing Nimda worm attack 21–22  
example, preventing spoof of x-forwarded-for request 25  
example, selective compression 23–24  
example, using selective compression 23–24  
list of tmsh commands 15  
publishing 16  
task summary 15  
video, mitigating shellshock example 26  
video, preventing Nimda worm example 21  
video, preventing spoof of x-forwarded-for request example 24  
video, selective compression example 23

Local Traffic Policies  
about 5

local traffic policy  
associating with virtual servers 18  
deleting 19  
modifying a published policy 17  
reordering rules 17

local traffic policy actions  
about options 12

local traffic policy conditions  
about options for 12

local traffic policy matching  
about 5  
about actions 10  
about all-match strategy 5  
about best-match strategy 5  
about conditions 7  
about first-match strategy 5  
about matching strategies 5  
about rules 6  
about rules and conditions logic 7  
about Tcl command substitutions 12  
condition types 7

local traffic policy matching strategy  
creating 19

local traffic policy rules  
reordering 17

## N

Nimda worm attack  
example of preventing 21  
preventing; iRules example 22  
preventing; tmsh example 21  
preventing; video example 21

## P

policies  
cloning local traffic 18  
creating for local traffic 16  
example of preventing Nimda worm attack 21  
example of selective compression 22  
mitigating shellshock attack, example 26  
preventing spoof of x-forwarded-for worm example 24  
publishing for local traffic 16

## S

selective compression  
example of using 22  
iRules example 24  
tmsh example 23  
using; video example 23

shellshock  
defined 26  
example of mitigating attack 26  
mitigating attack with tmsh; example 27

## Index

### shellshock attack

- mitigating; iRules example [28](#)
- mitigating; tmsh example [27](#)
- mitigating; video example [26](#)

## T

### Tcl command substitutions

- and local traffic policy matching [12](#)

### tmsh command

- example, mitigating shellshock attack [27](#)
- example, preventing Nimda worm attack [21](#)
- example, preventing spoof of x-forwarded-for request [25](#)
- example, selective compression [23](#)

## U

- user-defined matching strategy
  - creating [19](#)

## V

### virtual servers

- associating local traffic policy [18](#)

## X

### x-forwarded-for request

- example of preventing spoof of [24](#)
- preventing; video example [24](#)
- preventing a spoof of : iRules example [26](#)
- preventing a spoof of an [25](#)
- preventing a spoof with tmsh, example [25](#)