

iRule Interference: Custom Closes and Responses



Deb Allen, 2008-07-05

iRules provide a number of customizations to the already full feature set offered by LTM's highly configurable protocol profiles. In this article I'll give an overview of several useful commands which allow LTM to "interfere" with normal basic request / response operations by closing, rejecting, or silently ignoring connections (on either the client or server side), or by completely replacing the server's response with a different one.

Drop Silently

To silently drop an active connection, use the [drop](#) or [discard](#) command. `drop` and `discard` are synonymous. When one of these commands is invoked within an iRule, the connection is silently discarded. The lack of response may cause the peer to retry, but in most cases will eventually cause the peer to time out and close the connection. The connection is removed from the LTM connection table.

For UDP, to drop the current UDP packet *without* removing the flow from the connection table, use [UDP::drop](#) instead.

Reject Actively

To actively reject an active connection, use the [reject](#) command. When the `reject` command is invoked within an iRule processing a TCP connection, it is rejected by sending a packet with TCP Reset (RST) bit set. This response notifies the peer system that the connection is no longer valid, and should cause the peer system to close the connection immediately and cease using it. UDP connections will instead be rejected with an ICMP Unreachable message. The connection is removed from the LTM connection table.

Initiate TCP connection close

To initiate a normal 4-way TCP close handshake on an active connection, use the [TCP::close](#) command. When the `TCP::close` command is invoked within an iRule, the LTM system sends a packet with the FIN bit set, meaning it no longer intends to send data on the connection and the peer system is free to close the connection once it is finished sending data. Only the flow in the current context (client or server) will be affected. The connection is still maintained in the LTM connection table until both sides close the connection or an idle timeout removes it from the table.

Initiate HTTP connection close

To initiate a normal HTTP close on an active connection, use the [HTTP::close](#) command. When the `HTTP::close` command is invoked within an iRule, the LTM system inserts a Connection header with a value of `Close`, indicating that the connection should not be re-used for additional requests. Insertion of this header should trigger the normal 4-way FIN handshake close as each side completes its last transmission. The connection is still maintained in the LTM connection table until both sides close the connection or an idle timeout removes it from the table.

Respond with redirect

To replace a server response to an HTTP request with a 302 Temporary Redirect response (a pointer to an alternate URL), use the [HTTP::redirect](#) command. When the `HTTP::redirect` command is invoked within an iRule, the LTM system discards any server response and instead sends the specified redirect response. Neither the client or server side connection is closed by the `HTTP::redirect` command, but may be by using the `HTTP::close` or `TCP::close` command (see above). The redirect response is sent immediately and no other HTTP iRule processing may be performed on this request. (For a 301 Permanent Redirect response, use the `HTTP::respond` command instead.)

Replace server response with alternate response

To replace a server response on an active connection with an alternate response, use the [TCP::respond](#), [UDP::respond](#), or [HTTP::respond](#) command. When the `*::respond` command is invoked within an iRule, the LTM system discards any server response and instead sends the specified response data. Neither the client or server side connection is closed by the `*::respond` command, but for HTTP may be by using the [HTTP::close](#) command, or for TCP (including HTTP) by using the [TCP::close](#) command (see above). The response should be in keeping with any requirements dictated by the protocol or peer system.

Wiki Pages

[discard](#) - Causes the current packet or connection to be discarded. Same as the drop command.

[drop](#) - Causes the current packet or connection to be discarded. Same as the discard command.

[reject](#) - Causes the connection to be rejected.

[UDP::drop](#) - Drops the current UDP packet without removing the flow from the connection table

[TCP::close](#) - Closes the TCP connection.

[HTTP::close](#) - Closes the HTTP connection.

[HTTP::redirect](#) - Redirects an HTTP request or response to the specified URL.

[TCP::respond](#) - Sends the specified data directly to the peer.

[UDP::respond](#) - Sends data directly to a peer.

[HTTP::respond](#) - Generates a response to the client as if it came from the server.

[Get the Flash Player](#) to see this player.

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | [f5.com](#)

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com

©2016 F5 Networks, Inc. All rights reserved. F5, F5 Networks, and the F5 logo are trademarks of F5 Networks, Inc. in the U.S. and in certain other countries. Other F5 trademarks are identified at [f5.com](#). Any other products, services, or company names referenced herein may be trademarks of their respective owners with no endorsement or affiliation, express or implied, claimed by F5. CS04-00015 0113