Link Tracking With iRules - Part 1

Joe Pruitt, 2011-16-03

One common request that application developers ask of their network admins is to report on the amount of times links in their applications are requested. Various analytics packages including page trackers can be added to the application pages to store those values but in this article, I'll show you how to create a very simple link tracker with a transparent iRule applied to your virtual server of choice.

Storing The Data

The first step is to take the incoming URI's and store a count of the number of times they have been requested. In this application, I will be using the table command to create and store the data in a subtable (for more information on the table command, check out this excellent series of articles on it's various uses and features).

I've created a dynamic name for the table that consists of the prefix "LINK_TRACKING_" and is followed by the name of the virtual server that the iRules is applied to. This is beneficial in that it allows you to have a separate data store for different applications running on different virtual servers. The value of the table name is stored in the **TABLE_LINKDATA** variable.

Next, an attempt is made to increment the link count of the current URI (in the HTTP::uri value) with the "**table incr**" command. If the entry doesn't exist yet (the first time in the system), the if condition will fail and a new entry will be inserted with the "**table set**" command with a link count of 1. That's it! Now all your links are stored with their reference counts.

```
1: set TABLE_LINKDATA "LINK_TRACKING_[virtual name]"
2: if { [table incr -subtable $TABLE_LINKDATA -mustexist [HTTP::uri]] eq ""} {
3:   table set -subtable $TABLE_LINKDATA [HTTP::uri] 1 indefinite indefinite;
4: }
```

Reporting The Metrics

Storing the data is only half of the solution, there needs to be some way to view the data once it is in the system table. This is achieved with the following iRule code where I create a HTML page response containing a HTML table with a row for each entry in the link tracking data table. The first column contains the URI and the second is it's view count. The generated page is then returned to the browser via the HTTP::respond command.

```
1: set content {<html><head><title>Link Tracking</title></head><body><center>
2:   <table border='1'><tr><th colspan='2'><a href='/linkcleardata'>Clear Data</a></th></tr>
3:   <tr><th width='100%'>URI</th><th>Views</th></tr>}
4: foreach key [table keys -subtable $TABLE_LINKDATA] {
5:   append content "<tr><td>$key</td><td>[table lookup -subtable $TABLE_LINKDATA $key]</td></tr>";
6: }
7: append content "</table></center></body></html>";
8: HTTP::respond 200 Content $content;
```

Cleaning Up

You may notice the "Clear Data" hyperlink I created in the reporting page. When storing the data in the system table, I specified a time of "indefinite" for the lifetime of the data. This means that it lives in the system table for the lifetime of the TMM process. It would only be prudent to provide a way to clean up this table from time to time and that is where the "Clear Data" link comes in. It will link back to the iRule and issue the "**table delete**" command to clear out the contents of the link data table.

```
1: table delete -subtable $TABLE_LINKDATA -all;
```

Putting It All Together

Below is the full iRule to implement basic link tracking. All normal requests will fall through the "default" case in the switch statement and add the view counts to the link table. I've created two special URI's to handle the reporting and data management

- "/linkadmin" - This will generate the HTML response page and return it to the browser with all the links and their view counts.
- "/linkcleardata" - This will erase the contents of the link data table.

```
 1: when HTTP_REQUEST {
 2:   set TABLE_LINKDATA "LINK_TRACKING_[virtual name]"
 3:   switch [string tolower [HTTP::uri]] {
 4:     "/linkadmin" {
 5:       set content {<html><head><title>Link Tracking</title></head><body><center>
 6:         <table border='1'><tr><th colspan='2'><a href='/linkcleardata'>Clear Data</a></th></tr>
 7:         <tr><th width='100%'>URI</th><th>Views</th></tr>}
 8:       foreach key [table keys -subtable $TABLE_LINKDATA] {
 9:         append content "<tr><td>$key</td><td>[table lookup -subtable $TABLE_LINKDATA $key]</td></tr>";
10:       }
11:       append content "</table></center></body></html>";
12:       HTTP::respond 200 Content $content;
13:     }
14:     "/linkcleardata" {
15:       table delete -subtable $TABLE_LINKDATA -all;
16:       HTTP::redirect "http://[HTTP::host]/linkadmin"
17:     }
18:     default {
19:       if { [table incr -subtable $TABLE_LINKDATA -mustexist [HTTP::uri]] eq ""} {
20:         table set -subtable $TABLE_LINKDATA [HTTP::uri] 1 indefinite indefinite;
21:       }
22:     }
23:   }
24: }
```

Video Walkthrough

<div align="center">Link Tracking With iRules - Part 1</div>

More To Come!

There are several ways to enhance this iRule that I will be illustrating in future articles over the next few weeks. I'll look into building URI filters to allow you to only track certain URIs as well as adding filtering features in the generated link count report. Be on the lookout!

Get The Source

The source code for this application can be downloaded the LinkTracking topic in the iRules CodeShare.

Related Articles on DevCentral

- F5 DevCentral > Hot Topics > iRules
- Friendly URL Redirection Scaling via iRules > DevCentral > F5 ...
- F5 DevCentral > Community > Group Details - Management, Reporting ...
- DevCentral Wiki: iRules Reference
- An iRules Challenge, Resolved > DevCentral > F5 DevCentral > Tech Tips

- redirect and sorry server, new to irules - DevCentral - F5 ...
- 2x5-Minute iRules - Timing > DevCentral > F5 DevCentral > Tech Tips
- Documenting iRules with Comment Headers > DevCentral > F5 ...
- v10.1 - The table Command - The Basics > DevCentral > F5 ...
- Table Replace command with CMP - Connections reset - DevCentral ...

Technorati Tags: iRules, Link, Tracking, table, reporting

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc.	F5 Networks	F5 Networks Ltd.	F5 Networks
Corporate Headquarters	Asia-Pacific	Europe/Middle-East/Africa	Japan K.K.
info@f5.com	apacinfo@f5.com	emeainfo@f5.com	f5j-info@f5.com