iRules 101 - #13 - TCL String Commands Part 1

Jason Rahm, 2009-17-02

Introduction

An iRule is a powerful and flexible feature of BIG-IP devices based on F5's exclusive TMOS architecture. iRules provide you with unprecedented control to directly manipulate and manage any IP application traffic. iRules utilizes an easy to learn scripting syntax and enables you to customize how you intercept, inspect, transform, and direct inbound or outbound application traffic. In this series of tech tips, we'll talk about the TCL language, it's usage and control structures, as well as iRule extensions to the TCL language. Other articles in the series:

TCL: The String Command

There's a philosophical discussion as to whether or not everything in TCL is a string (and whether that is a good thing or not), but whether or not you subscribe to the debate, strings play a major role in TCL, and so it is with iRules as well. TCL affords plenty of options on the string command, several of which will be covered in this article:

- string length
- string tolower
- string toupper
- string first
- string last

We'll look at concrete examples for these commands, mostly because I find the examples they use in the TCL documentation somewhat lacking.

string length *string*

Returns a decimal string giving the number of characters in ***string***. In this example, it is used as a conditional. If the host is set, the conditional is true and the redirect utilizes the hostname and if false, uses the local address:

```
when HTTP_REQUEST {
    # Check if some condition for the request is true?
```

```
    if {[HTTP::query] contains "Ntt="}{
       # Check if Host header value has a length
       if {[string length [HTTP::host]]}{
   # Redirect client using Host header value
          HTTP::redirect "http://[HTTP::host]/search.do?Ntt=[URI::query [HTTP::uri] Ntt]"
       } else {
# Redirect client using VIP address
          HTTP::redirect "http://[IP::local_addr]/search.do?Ntt=[URI::query [HTTP::uri] Ntt]"
       }
    }
}
```

In this example, the rule is looking for any requests with a URI of "/" and redirects to a login page:

```
when HTTP_REQUEST {
  if {[string length [HTTP::uri]] < 2} {
    HTTP::redirect [HTTP::host]/somepath/Login.aspx
  }
```

}

string tolower *string ?first? ?last?*

Returns a value equal to *string* only that all upper case values are converted to lower case. If specified, *first* and *last* indicate the character index delineating *string* modification. It is useful to convert anything not case sensitive to increase your match and decrease your administration. For example, you can eliminate case on your host comparisons so it matches every request where case might have been misapplied. This avoids a costly regular expression, and the thought necessary to implement the expression.

```
when HTTP_REQUEST {
  if { [string tolower [HTTP::host]] equals "my.domain.com" } {
    forward
  } else { discard }
}
```

string toupper works the same way, but I always use tolower, I guess lower case readability is more appealing to me.

string first *needleString haystackString ?startIndex?*

Search *haystackString* for a sequence of characters that exactly match the characters in *needleString*. If found, return the index of the first character in the first such match within *haystackString*. If not found, return -1. If *startIndex* is specified, then the search is constrained to start with the character in *haystackString* specified by the index. Joe highlighted an example a while back that searches the string returned from the User Agent header for "bot", and if present, the request gets dumped to a pool for spider traffic:

```
   when HTTP_REQUEST {
      if { [HTTP::header User-Agent] == "" } {
         if { [matchclass [IP::remote_addr] equals $::blacklisted_clients] } {
            pool spider_[HTTP::header Host]
         }
      }
      elseif { [matchclass [HTTP::header User-Agent] contains $::blacklisted_useragents] } {
            pool spider_[HTTP::header Host]
      }
      elseif { [string first -nocase "bot" [HTTP::header User-Agent]] >= 0 } {
```

```
            pool spider_[HTTP::header Host]
      }
      else {
            pool pool_[HTTP::header Host]
      }
   }
```

string last *needleString haystackString ?lastIndex?*

Similar to **string first**, only returns the index of the first character in the last such match within *haystackString*. If there is no match, then return -1. If *lastIndex* is specified, then only the characters in *haystackString* at or before the specified *lastIndex* will be considered by the search. In this example, (combined with **string range**, more on that in a forthcoming tip) the conditional returns true if the string beginning with the last "/" in the URI, and ending with the last character of the URI, contains a "."

```
when HTTP_REQUEST {
  if {[matchclass [HTTP::uri] ends_with $::unmc_extends] or
  [matchclass [HTTP::method] equals $::unmc_methods] or
  [matchclass [HTTP::uri] contains $::unmc_sql] or
  [matchclass [IP::client_addr] equals $::unmc_restrict_ips]}{
    discard
  } elseif {
  ([HTTP::uri] ends_with "/") or
([string range [string last / [HTTP::uri]] end] contains ".") or
  ([HTTP::uri] contains "unmcintranet")}{
    pool unmc-intranet-proxy
  } elseif {[HTTP::uri] contains "google"}{
    HTTP::redirect "http://[HTTP::host][HTTP::uri]&restrict=unmcintranet"
  } else {
    HTTP::redirect "http://[HTTP::host][HTTP::uri]/"
  }
}
```

Stay tuned, we'll cover more string commands next time.

Get the Flash Player to see this player.
20090217-iRulesFundamentals101_13_TCLStringCommandPart1.mp3

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