20 Lines or Less #12

Colin Walker, 2008-24-07

What could you do with your code in 20 Lines or Less? That's the question I ask every week, and every week I go looking to find cool new examples that show just how flexible and powerful iRules can be without getting in over your head.

Here we go again, three more examples of the powerful and interesting things you can do with iRules in less than 21 lines. Dipping again into the forums, with a few tweaks here and there (don't worry, I stayed honest to the rule, just took out comments and extra case comparisons, that kind of thing), we've got an action packed 20LoL this week. Here we go:

SSL iRule on a non-SSL VIP

http://devcentral.f5.com/Default.aspx?tabid=53&forumid=5&postid=26299&view=topic

This is a great example of using a single iRule for both HTTP and HTTPS traffic. In the forum post Deb shows a cool trick to allow us to sneak SSL commands past the iRule interpreter so that they are there when we need them, if a cert is found, but aren't used when the connection turns out to be straight HTTP. Pretty cool stuff.

```
when HTTP_REQUEST {
  HTTP::header replace ClientIP [IP::remote_addr]
  if {[PROFILE::exists clientssl] == 1} {
    set cname "SSL::cipher name"
    set cbits "SSL::cipher bits"
    set cver "SSL::cipher version"
    HTTP::header replace SSLCipher [eval $cname]:[eval $cbits]-[eval
$cver]
    if { [SSL::cert count] > 0} {
      HTTP::header replace SSLSubject [b64encode [X509::subject
[SSL::cert 0]]]
      HTTP::header replace SSLClientCert [b64encode [SSL::cert 0]]
      HTTP::header replace WebProtocol "HTTPS-auth"
    } else {
      HTTP::header replace WebProtocol "HTTPS"
    }
  } else {
    HTTP::header replace WebProtocol "HTTP"
  }
}
```

Extracting DHCP Info

http://devcentral.f5.com/Default.aspx?tabid=53&forumid=5&postid=25727&view=topic

This example for extracting DHCP info is very specific. It's looking for option 82 (Support for Routed Bridge Encapsulation) which may not be particularly useful to everyone out there, but the example stands as a great display of how iRules can help you tear into almost any kind of data, even DHCP data, and make intelligent decisions or actions based on that. Sure, it might take some re-working for your purposes, but what a cool example to get started with!

```
when CLIENT_DATA {
  binary scan [UDP::payload] x240H* dhcp_option_payload
  set option 0
  set option_length [expr {([UDP::payload length] -240) * 2 }]
  for {set i 0} {$option != 52 && $i < $option_length} {incr i [expr
{ $length * 2 +2 }]} {
    binary scan $dhcp_option_payload x[expr $i]a2 option
    incr i 2
    binary scan $dhcp_option_payload x[expr $i]a2 length_hex
    set length [expr 0x$length_hex]
  }
  if { $i < $option_length } {
    incr i -[expr { $length * 2 -2 }]
    binary scan $dhcp_option_payload x[expr $i]a2 length_hex
    set length [expr 0x$length_hex]
    incr i 2
    binary scan $dhcp_option_payload x[expr $i]a[expr { $length * 2
}] circuit_id
  } else {
    drop
  }
}
```

URI re-writing based on Load Balancing decision

http://devcentral.f5.com/Default.aspx?tabid=53&forumid=5&postid=25791&view=topic

Talk about a chicken and egg demonstration. Hearing the title, you might think I have it backwards. When making this kind of decision in an iRule, the URI is often used to help make the load balancing decision. In this case, it's just the opposite. In this example we're letting the BIG-IP make a load balancing decision, then going back and updating the URI based on that decision, before the request is sent to the servers. Very cool stuff!

```
when HTTP_REQUEST_SEND {
  set uri [string tolower [clientside {HTTP::uri}]]
  log local0. "[IP::client_addr]:[TCP::client_port]: selected server
details: [LB::server] - \$uri: $uri"
  if {[IP::addr [LB::server addr] equals 10.207.225.101] or [IP::addr
[LB::server addr] equals 10.207.225.102] or [IP::addr [LB::server
addr] equals 10.207.225.103] }{
    log local0. "[IP::client_addr]:[TCP::client_port]: matched server
check for .3 or .4"
    switch -glob [HTTP::uri] {
      "*/gsfo/gsfopub*" {
        clientside {HTTP::uri "/Async/CMReceive.ashx"}
        log local0. "[IP::client_addr]:[TCP::client_port]: updated
URI to /Async/CMReceive.ashx"
      }
      "*/era/erapub*" {
        clientside {HTTP::uri "/Async/ERAReceive.ashx"}
        log local0. "[IP::client_addr]:[TCP::client_port]: updated
URI to /Async/ERAReceive.ashx"
      }
      default {
        log local0. "[IP::client_addr]:[TCP::client_port]: didn't
match URI checks"
      }
    }
  }
}
```

There you have it, the forums deliver yet again. I have to say I love checking out all these cool, new, compact examples of iRules goodness. Many thanks to the awesome DevCentral community for their continued contributions. I'll see you next week for another 20 Lines or Less.

#Colin

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