Client Persistence based on Response

Use Case:

Find out a unique string in HTTP response content from backend to attach it to Client side persistence logic.

F5 iRules:

```
when HTTP_RESPONSE {


   # Use persistence based on the same token that was added in a prior respons
e

   # Need to parse $jsessionid from some portion of the request headers or pay
load

   # See this Codeshare entry for some examples of using cookie or path comman
ds to do so

   # http://devcentral.f5.com/wiki/default.aspx/iRules/Weblogic_JSessionID_Per
sistence.html

   persist $my_jsessionid_parsed_from_the_request 3600


}

when HTTP_RESPONSE {


   # Clear the jsessionid if it exists already on this TCP connection

   if {[info exists jsessionid]}{

      unset jsessionid

   }


   # Only look for the jsessionid in text responses

   if {[HTTP::header value "Content-Type"] starts_with "text"}{


      log local0. "[IP::client_addr]:[TCP::client_port]: Matched text, enablin
g stream filter"
```

```
      # Because TCL doesn't support lookaheads, match the jsessionid string an
d value

      # We'll parse out the value in STREAM_MATCHED

      # Assume the jsessionid is 1 to 100 characters (terminated by a non-alph
anumeric character).

      STREAM::expression ;jsessionid=[A-Za-z0-9]{1,100}

      STREAM::enable


      # Enable the STREAM_MATCHED event as it could have been disabled if ther
e was a prior

      # response on this TCP connection

      event STREAM_MATCHED enable


   } else {


      # Disable the stream filter as this wasn't a text response

      log local0. "[IP::client_addr]:[TCP::client_port]: No Content-Type match
, disabling stream filter"

      STREAM::disable

   }

}

when STREAM_MATCHED {


   # Save the matched value (example: ;jsessionid=ABCDEF)

   set jsessionid [STREAM::match]

   log local0. "[IP::client_addr]:[TCP::client_port]: Matched: $jsessionid"


   # STREAM::match shouldn't match a null string with the defined regex, but c
heck anyhow

   if {[string length $jsessionid]}{


      # Get the jsessionid value (split ;jsessionid=ABCDEF on the equals sign)

      set jsessionid [getfield $jsessionid "=" 2]


      # The iRule parser doesn't allow the persist command in STREAM_MATCHED.

      # It works though, so hide the command from the parser
```

```
      # Add a persistence record with the jsessionid token and a one hour time
out

      set persist_cmd "persist add uie $jsessionid 3600"

      log local0. "[IP::client_addr]:[TCP::client_port]: Parsed: $jsessionid \
$persist_cmd: $persist_cmd"


      eval $persist_cmd

      #persist add uie $jsessionid


      # Assume the first match is the same as any other jsessionids, so stop c
hecking for them

      log local0. "[IP::client_addr]:[TCP::client_port]: Added persistence rec
ord. Exiting event for this response."

      event STREAM_MATCHED disable

   }

}
```

URL: https://devcentral.f5.com/codeshare/persist-client-on-response-content-with-stream

NetScaler Solution:

```
set lb vserver vip1 -persistencetype RULE -rule
'HTTP.REQ.COOKIE.VALUE("jsessionid")' -resRule
'HTTP.RES.SET_COOKIE.COOKIE("jsessionid")'
```

NetScaler provides easy way to set the persistency based on Rule which has combination of both Request side check and Response side parsing. In the configuration above, we are figuring out the “jsessionid” value from response and setting the Persistence based on matching value in the request header.