

Log HTTP Headers

Use Case:

HTTP header logging is typically done for troubleshooting and offline processing purposes.

F5 iRules:

```
when HTTP_REQUEST {
    set LogString "Client [IP::client_addr]:[TCP::client_port] -> [HTTP::host][
HTTP::uri]"
    log local0. "======"
    log local0. "$LogString (request)"
    foreach aHeader [HTTP::header names] {
        log local0. "$aHeader: [HTTP::header value $aHeader]"
    }
    log local0. "======"
}
when HTTP_RESPONSE {
    log local0. "======"
    log local0. "$LogString (response) - status: [HTTP::status]"
    foreach aHeader [HTTP::header names] {
        log local0. "$aHeader: [HTTP::header value $aHeader]"
    }
    log local0. "======"
}

# Sample output:

Rule log_http_headers_rule <HTTP_REQUEST>: =====
=====
Rule log_http_headers_rule <HTTP_REQUEST>: Client 192.168.99.32:2950 -> webmai
l.example.com/exchange/Aaron/Inbox/?Cmd=contents (request)
Rule log_http_headers_rule <HTTP_REQUEST>: Host: webmail
```

```
Rule log_http_headers_rule <HTTP_REQUEST>: User-Agent: Mozilla/5.0 (Windows; U
; Windows NT 5.1; en-US; rv:1.8.1.9)

Rule log_http_headers_rule <HTTP_REQUEST>: Accept: text/xml,application/xml,ap
plication/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,im

Rule log_http_headers_rule <HTTP_REQUEST>: Accept-Language: en-us,en;q=0.5

Rule log_http_headers_rule <HTTP_REQUEST>: Accept-Encoding: gzip,deflate

Rule log_http_headers_rule <HTTP_REQUEST>: Accept-Charset: ISO-8859-1,utf-8;q=
0.7,*;q=0.7

Rule log_http_headers_rule <HTTP_REQUEST>: Keep-Alive: 300

Rule log_http_headers_rule <HTTP_REQUEST>: Connection: keep-alive

Rule log_http_headers_rule <HTTP_REQUEST>: Referer: https://webmail.example.co
m/exchange/

Rule log_http_headers_rule <HTTP_REQUEST>: X-Forwarded-For: 192.168.99.32

Rule log_http_headers_rule <HTTP_REQUEST>: Front-End-Https: On

Rule log_http_headers_rule <HTTP_REQUEST>: =====
=====

Rule log_http_headers_rule <HTTP_RESPONSE>: =====
=====

Rule log_http_headers_rule <HTTP_RESPONSE>: Client 192.168.99.32:2950 -> webma
il.example.com/exchange/Aaron/Inbox/?Cmd=contents (response) - status: 200

Rule log_http_headers_rule <HTTP_RESPONSE>: Date: Tue, 06 Nov 2007 16<!--:30:3
2 GMT-->

Rule log_http_headers_rule <HTTP_RESPONSE>: Server: Microsoft-IIS/6.0

Rule log_http_headers_rule <HTTP_RESPONSE>: X-Powered-By: ASP.NET

Rule log_http_headers_rule <HTTP_RESPONSE>: Content-Type: text/html

Rule log_http_headers_rule <HTTP_RESPONSE>: Content-Length: 55446

Rule log_http_headers_rule <HTTP_RESPONSE>: MS-WebStorage: 6.5.7638

Rule log_http_headers_rule <HTTP_RESPONSE>: Cache-Control: no-cache

Rule log_http_headers_rule <HTTP_RESPONSE>: =====
=====
```

URL: <https://devcentral.f5.com/codeshare/log-http-headers>

NetScaler Solution:

```
set sysLogParams -UserDefinedAuditlog Yes
```

```
add auditmessageaction log_request_headers INFORMATIONAL
HTTP.REQ.FULL_HEADER.AFTER_STR("\r\n") -BypassSafetyCheck Yes
```

Use the above to log Request Side Headers

```
set syslogParams -UserDefinedAuditlog Yes
```

```
add auditmessageaction log_response_headers INFORMATIONAL
HTTP.RES.FULL_HEADER.AFTER_STR("\r\n") -BypassSafetyCheck Yes
```

NetScaler provides pretty efficient way to log the runtime traffic details using auditmessageaction. For capturing the entire headers, we have single expression HTTP.REQ/S.FULL_HEADER which helps capturing entire set of headers on request or response side.