EnableResponseStatePatterns

Introduction

EnableResponseStatePatterns is a hidden feature in WebInspect which can be used to handle a specific type of state management scheme, examples of which have been encountered in web applications and RESTful web services.

The original example that led to this feature being added was a case where an application had a state value that shows up on a redirect in the query part of a URL: example: f?p=7700:LOGIN:999999999:::::, the 999999999 is some kind of identifier that appears to be some kind of state.

That state value is used later in a post parameter called p_instance.

Configuration

The first thing is to set the EnableResponseStatePatterns to true (default is false). You will find that setting in the scan settings XML files (normally in *C:\ProgramData\HP\HP WebInspect\Settings*).

Then add a ReponseStateElement for each variable being tracked.

Each response state element (under ResponseStatePatterns) represents a single state value. Search regexes allow us to determine where we collect state from in HTTP Responses. The Named Group is important as it tells us which part of the match in the regex is the actual state value. Replace regexes tell WebInspect where to apply that state on HTTP Requests. Again, the Named Group (name must match the name of the response state element, in this case “Oracle”) is required to tell us where in the match the actual state value goes.

```
<EnableResponseStatePatterns>true</EnableResponseStatePatterns>
<ResponseStatePatterns>
    <ResponseStateElement>
        <name>Oracle</name>
        <SearchRegexes>
            <string> Location:.*f\?p=[0-9]+:[A-Za-z]+:(?<Oracle>[0-
9]+):::</string>
        </SearchRegexes>
        <ReplaceRegexes>
            <string>>&p_instance=(?<Oracle>[0-9]+)&</string>
        </ReplaceRegexes>
    </ResponseStateElement>
</ResponseStatePatterns>
```

You will find a self-closed element that looks like <ResponseStatePatterns />.

Expand this element and fill it in as needed to look as follows:

```
<EnableResponseStatePatterns>true</EnableResponseStatePatterns>
```

```
<ResponseStatePatterns>
    <ResponseStateElement>
        <name>CaptureGroupName</name>
        <ReplaceRegexes>
            <string>RegexWithCaptureGroupToApplyStateToRequest</string>
        </ReplaceRegexes>
        <SearchRegexes>
            <string>RegexWithCaptureGroupToCollectStateFromResponse</string>
        </SearchRegexes>
    </ResponseStateElement>
</ResponseStatePatterns>
```

The **SearchRegexes** are applied on the entire HTTP Response and the **ReplaceRegexes** are applied on the entire request. The capture group would represent the exact token that needs to be transferred from the HTTP Response to the following HTTP Request. So, there is no need to worry about a parameter name in this technique. All that matters is the exact value that represents the state.

Example 1

Given that the HTTP Response Body returns something like this:

... {"Token":"2bfcf7e0-5b31-46e8-aabf-007e26a36177","Overlayreqd":1,"Disabled":0,"Downloadpriv ...

And that subsequent HTTP Requests must contain this header:

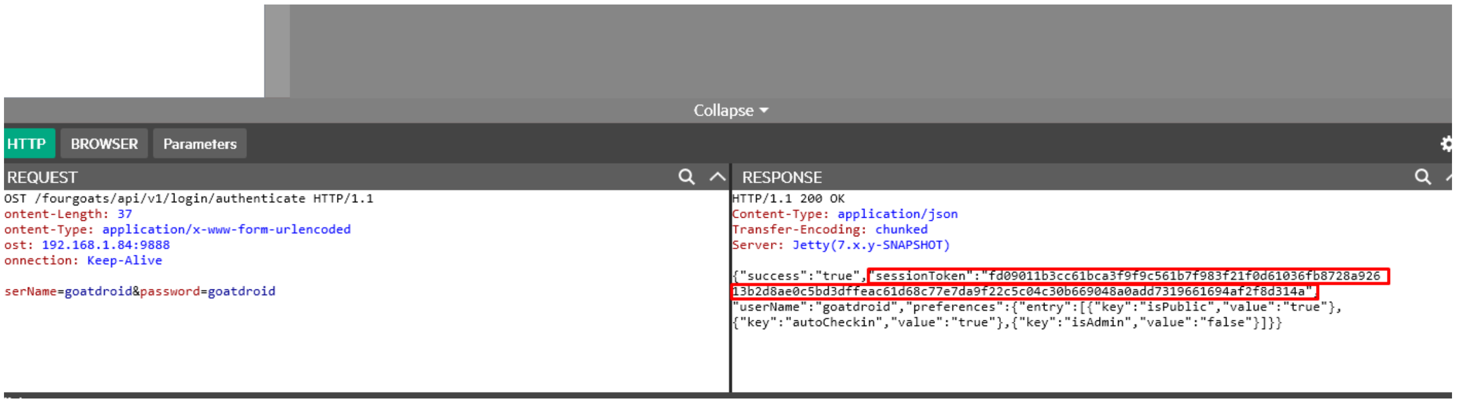
Authorization-Token: 2bfcf7e0-5b31-46e8-aabf-007e26a36177

Then a response state pattern like this would be required:

```
<EnableResponseStatePatterns>true</EnableResponseStatePatterns>
<ResponseStatePatterns>
    <ResponseStateElement>
        <name>Token</name>
        <ReplaceRegexes>
            <string> Authorization-Token:\s(?'Token'[a-fA-F0-9]{8}-[a-fA-F0-
9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{12})</string>
        </ReplaceRegexes>
        <SearchRegexes>
            <string>"Token":"(?'Token'[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-
9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{12})"</string>
        </SearchRegexes>
    </ResponseStateElement>
</ResponseStatePatterns>
```

Example 2

This example is from the native Android web application “Fourgoats”. After authenticating (POSTing the credentials), the backend web service returns a response like this:

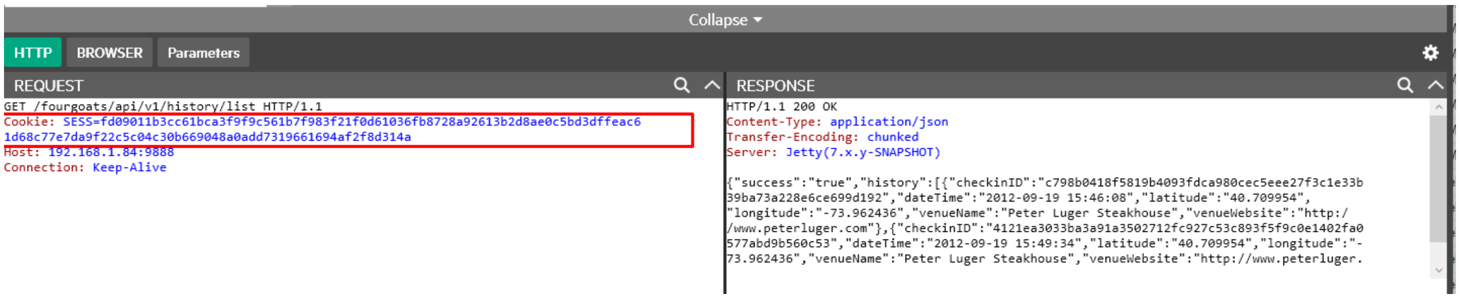


```
OST /fourgoats/api/v1/login/authenticate HTTP/1.1
ontent-Length: 37
ontent-Type: application/x-www-form-urlencoded
ost: 192.168.1.84:9888
onnection: Keep-Alive

serName=goatdroid&password=goatdroid
```

```
HTTP/1.1 200 OK
Content-Type: application/json
Transfer-Encoding: chunked
Server: Jetty(7.x.y-SNAPSHOT)

{"success":"true","sessionToken":"fd09011b3cc61bca3f9f9c561b7f983f21f0d61036fb8728a926
13b2d8ae0c5bd3dffeac61d68c77e7da9f22c5c04c30b669048a0add7319661694af2f8d314a",
"userName":"goatdroid","preferences":{"entry":[{"key":"isPublic","value":"true"},
{"key":"autoCheckin","value":"true"},{"key":"isAdmin","value":"false"}]}}
```

Notice the sessionToken included in the json response body.

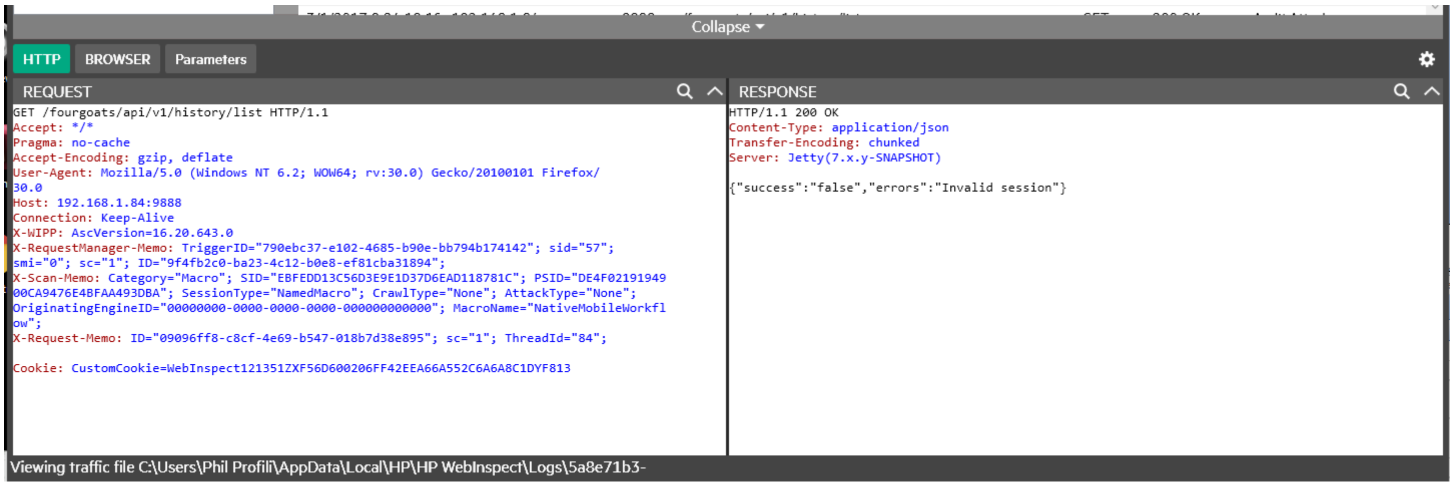
The client application contains logic that understands that each subsequent request should set the cookie “SESS” to that value, like this:



```
GET /fourgoats/api/v1/history/list HTTP/1.1
Cookie: SESS=fd09011b3cc61bca3f9f9c561b7f983f21f0d61036fb8728a92613b2d8ae0c5bd3dffeac6
1d68c77e7da9f22c5c04c30b669048a0add7319661694af2f8d314a
Host: 192.168.1.84:9888
Connection: Keep-Alive
```

```
HTTP/1.1 200 OK
Content-Type: application/json
Transfer-Encoding: chunked
Server: Jetty(7.x.y-SNAPSHOT)

{"success":"true","history":[{"checkinID":"c798b0418f5819b4093fdca980cec5eee27f3c1e33b
39ba73a228e6ce699d192","dateTime":"2012-09-19 15:46:08","latitude":"40.709954",
"longitude":"-73.962436","venueName":"Peter Luger Steakhouse","venueWebsite":"http:/
/www.peterluger.com"},{"checkinID":"4121ea3033ba3a91a3502712fc927c53c893f5f9c0e1402fa0
577abd9b560c53","dateTime":"2012-09-19 15:49:34","latitude":"40.709954","longitude":"-
73.962436","venueName":"Peter Luger Steakhouse","venueWebsite":"http://www.peterluger.
```

But at scan time WebInspect doesn’t recognize that need so sessions are mostly invalid. This is from a scan. Notice there is no SESS cookie because WebInspect does not by default have any knowledge of the logic that is needed to maintain state with this particular application:

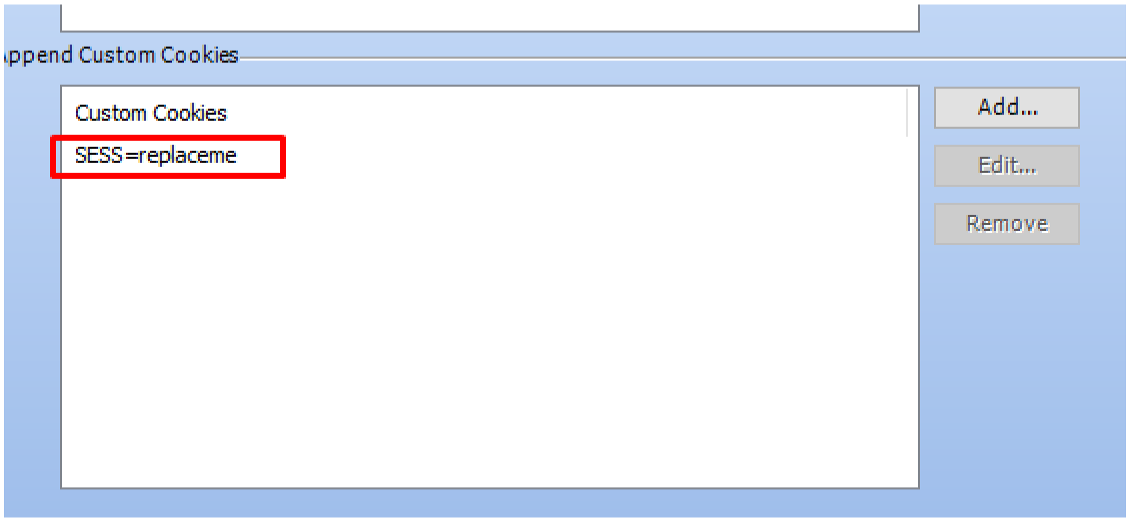


```
GET /fourgoats/api/v1/history/list HTTP/1.1
Accept: */*
Pragma: no-cache
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/5.0 (Windows NT 6.2; WOW64; rv:30.0) Gecko/20100101 Firefox/
30.0
Host: 192.168.1.84:9888
Connection: Keep-Alive
X-WIPP: AscVersion=16.20.643.0
X-RequestManager-Memo: TriggerID="790ebc37-e102-4685-b90e-bb794b174142"; sid="57";
smi="0"; sc="1"; ID="9f4fb2c0-ba23-4c12-b0e8-ef81cba31894";
X-Scan-Memo: Category="Macro"; SID="EBFEDD13C56D3E9E1D37D6EAD118781C"; PSID="DE4F02191949
00CA9476E4BFAA493DBA"; SessionType="NamedMacro"; CrawlType="None"; AttackType="None";
OriginatingEngineID="00000000-0000-0000-0000-000000000000"; MacroName="NativeMobileWorkfl
ow";
X-Request-Memo: ID="09096ff8-c8cf-4e69-b547-018b7d38e895"; sc="1"; ThreadId="84";

Cookie: CustomCookie=WebInspect121351ZXF56D600206FF42EEA66A552C6A6A8C1DYF813
```

```
HTTP/1.1 200 OK
Content-Type: application/json
Transfer-Encoding: chunked
Server: Jetty(7.x.y-SNAPSHOT)

{"success":"false","errors":"Invalid session"}
```

Viewing traffic file C:\Users\Phil Profili\AppData\Local\HP\HP WebInspect\Logs\5a8e71b3-

To address this, add a custom cookie into the scan settings like this:



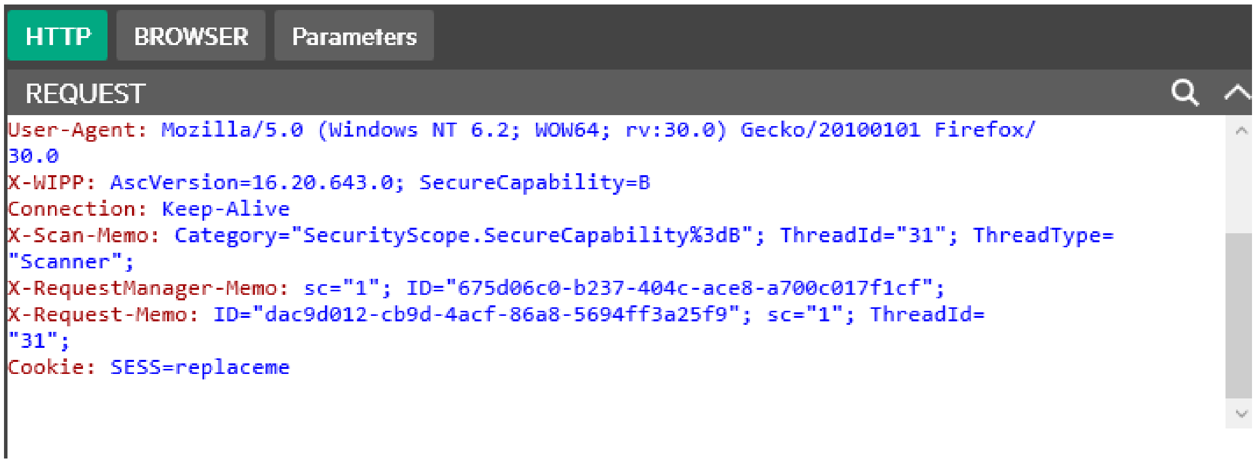
Custom Cookies
SESS=replaceme

And enable response state patterns in the scan settings file, configured in such a way to assign to the capture group “Token” the value of sessToken seen in the initial authentication response, and then use that to replace the “replaceme” portion of the cookie header with that value.

The settings look like this:

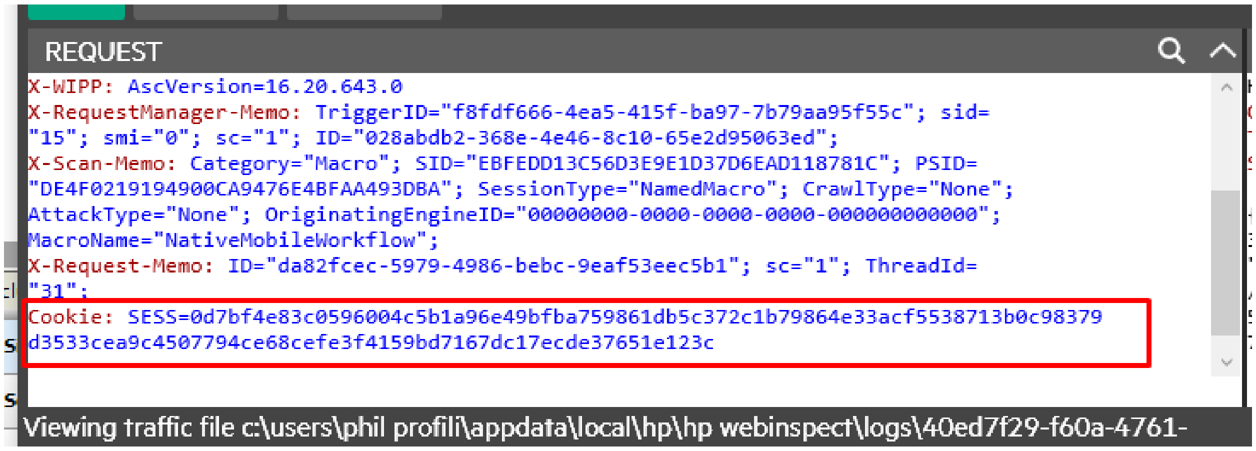
```
    <EnableResponseStatePatterns>true</EnableResponseStatePatterns>
    <ResponseStatePatterns>
      <ResponseStateElement>
        <name>Token</name>
        <ReplaceRegexes>
          <string>Cookie:\sSESS=(?'Token'replaceme)</string>
        </ReplaceRegexes>
        <SearchRegexes>
          <string>"sessionToken":"(?'Token'[a-f0-9]{1,128})"</string>
        </SearchRegexes>
      </ResponseStateElement>
    </ResponseStatePatterns>
```

So what you see now in the initial requests is that the headers have the raw (un-replaced) cookie header:



```
User-Agent: Mozilla/5.0 (Windows NT 6.2; WOW64; rv:30.0) Gecko/20100101 Firefox/
30.0
X-WIPP: AscVersion=16.20.643.0; SecureCapability=B
Connection: Keep-Alive
X-Scan-Memo: Category="SecurityScope.SecureCapability%3dB"; ThreadId="31"; ThreadType=
"Scanner";
X-RequestManager-Memo: sc="1"; ID="675d06c0-b237-404c-ace8-a700c017f1cf";
X-Request-Memo: ID="dac9d012-cb9d-4acf-86a8-5694ff3a25f9"; sc="1"; ThreadId=
"31";
Cookie: SESS=replaceme
```

But once a match is made in the response containing the token, subsequent requests look like this:

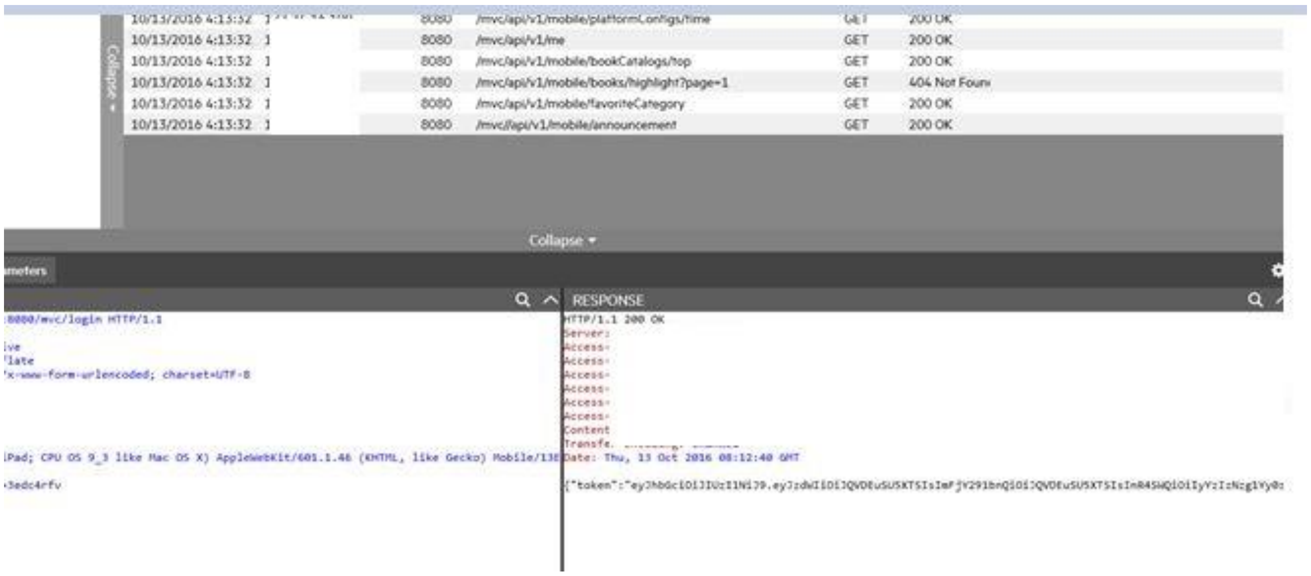


```
X-WIPP: AscVersion=16.20.643.0
X-RequestManager-Memo: TriggerID="f8fdf666-4ea5-415f-ba97-7b79aa95f55c"; sid=
"15"; smi="0"; sc="1"; ID="028abdb2-368e-4e46-8c10-65e2d95063ed";
X-Scan-Memo: Category="Macro"; SID="EBFEDD13C56D3E9E1D37D6EAD118781C"; PSID=
"DE4F0219194900CA9476E4BFAA493DBA"; SessionType="NamedMacro"; CrawlType="None";
AttackType="None"; OriginatingEngineID="00000000-0000-0000-0000-000000000000";
MacroName="NativeMobileWorkflow";
X-Request-Memo: ID="da82fcec-5979-4986-bebc-9eaf53eec5b1"; sc="1"; ThreadId=
"31";
Cookie: SESS=0d7bf4e83c0596004c5b1a96e49bfba759861db5c372c1b79864e33acf5538713b0c98379
d3533cea9c4507794ce68cefe3f4159bd7167dc17ecde37651e123c
```

Viewing traffic file c:\users\phil profili\appdata\local\hp\hp webinspect\logs\40ed7f29-f60a-4761-

Example 3

A token is generated by POST request:



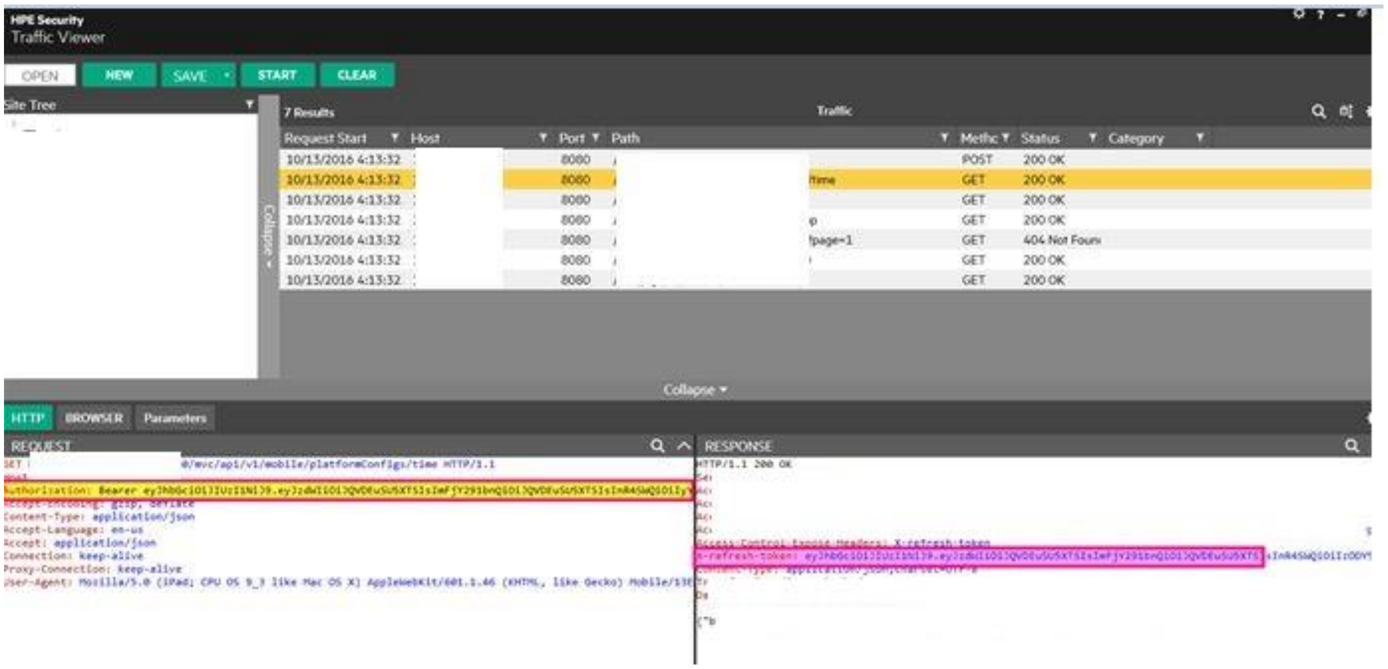
```
10/13/2016 4:13:32  1    8080  /mvc/api/v1/mobile/platformConfigs/time          GET  200 OK
10/13/2016 4:13:32  1    8080  /mvc/api/v1/me                                   GET  200 OK
10/13/2016 4:13:32  1    8080  /mvc/api/v1/mobile/bookCatalogs/top              GET  200 OK
10/13/2016 4:13:32  1    8080  /mvc/api/v1/mobile/books/highlight?page=1        GET  404 Not Foun
10/13/2016 4:13:32  1    8080  /mvc/api/v1/mobile/favoriteCategory              GET  200 OK
10/13/2016 4:13:32  1    8080  /mvc//api/v1/mobile/announcement                 GET  200 OK
```

```
:8080/mvc/login HTTP/1.1

ive
'late
'x-www-form-urlencoded; charset=UTF-8



iPad; CPU OS 9_3 like Mac OS X) AppleWebKit/601.1.46 (KHTML, like Gecko) Mobile/13E

-3edc4rfv
```

RESPONSE

```
HTTP/1.1 200 OK
Server:
Access-
Access-
Access-
Access-
Access-
Access-
Content
Transfe.
Date: Thu, 13 Oct 2016 08:12:40 GMT

{"token":"eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJQVDEuSU5XTSIsImFjY291bnQiOiJQVDEuSU5XTSIsInR4SWQiOiIyYzIzNzg1Yy0z
```

Then the token from the above step is sent and a new token (X-refresh token) is generated.



7 Results — Traffic

Request Start	Host	Port	Path	Methc	Status	Category
10/13/2016 4:13:32		8080		POST	200 OK	
10/13/2016 4:13:32		8080	time	GET	200 OK	
10/13/2016 4:13:32		8080		GET	200 OK	
10/13/2016 4:13:32		8080	p	GET	200 OK	
10/13/2016 4:13:32		8080	?page=1	GET	404 Not Foun	
10/13/2016 4:13:32		8080		GET	200 OK	
10/13/2016 4:13:32		8080		GET	200 OK	

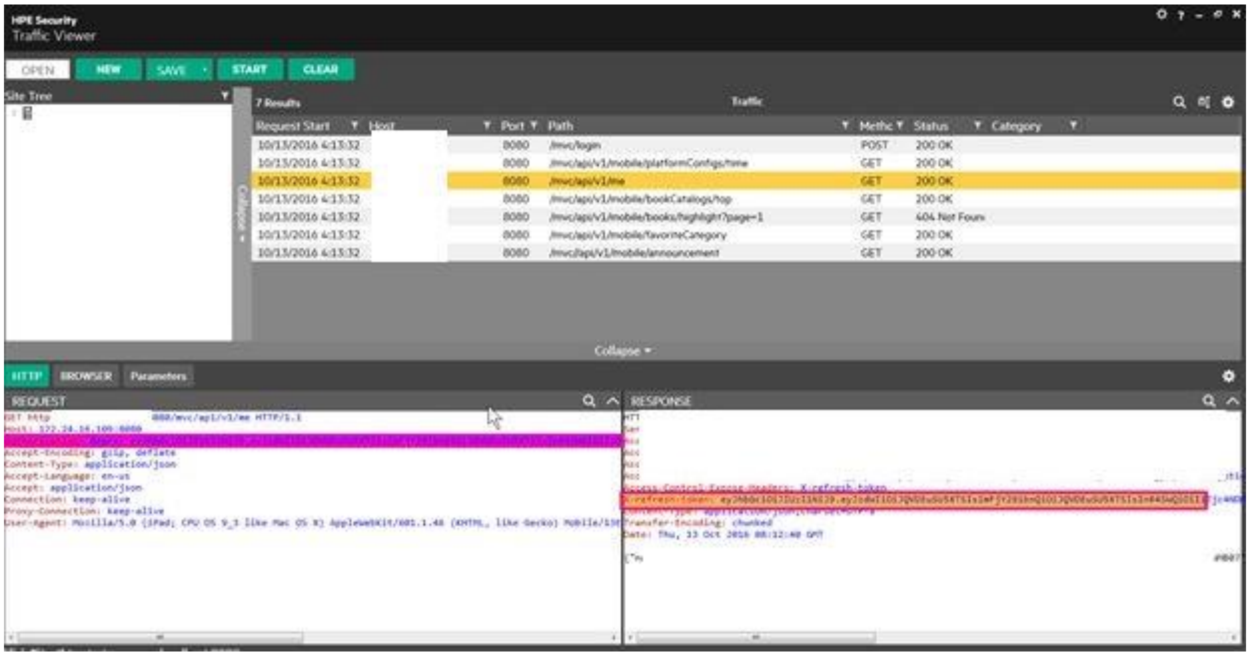
REQUEST

```
GET          0/mvc/api/v1/mobile/platformConfigs/time HTTP/1.1
Host
Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJQVDEuSU5XTSIsImFjY291bnQiOiJQVDEuSU5XTSIsInR4SWQiOiIy
Accept-Encoding: gzip, deflate
Content-Type: application/json
Accept-Language: en-us
Accept: application/json
Connection: keep-alive
Proxy-Connection: keep-alive
User-Agent: Mozilla/5.0 (iPad; CPU OS 9_3 like Mac OS X) AppleWebKit/601.1.46 (KHTML, like Gecko) Mobile/13E
```

RESPONSE

```
HTTP/1.1 200 OK
Se
Ac
Ac
Ac
Ac
Access-Control-Expose-Headers: X-refresh-token
X-refresh-token: eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJQVDEuSU5XTSIsImFjY291bnQiOiJQVDEuSU5XTSIsInR4SWQiOiIzODY
Content-Type: application/json;charset=UTF-8
Tr
Da

{"b
```

The token from the previous response is used as "**Authorization: Bearer**" header in the next request. Another refresh token is generated. And so on.



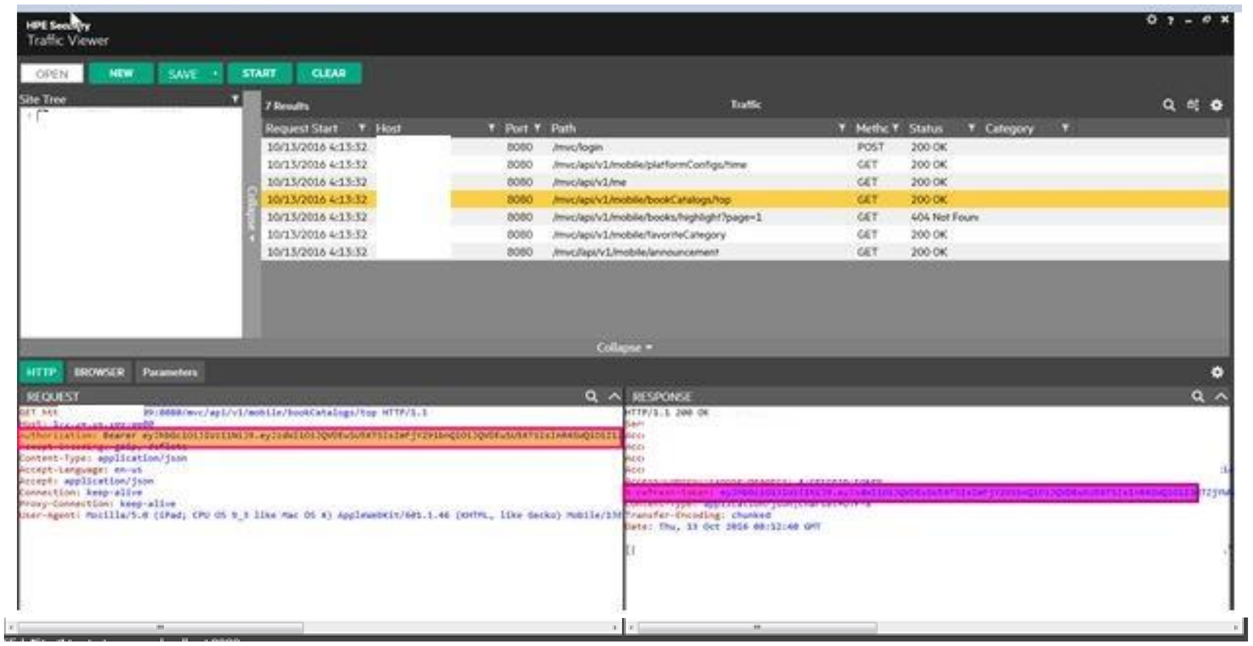
Request Start	Host	Port	Path	Method	Status	Category
10/13/2016 4:13:32	[redacted]	8080	/mvc/login	POST	200 OK	
10/13/2016 4:13:32	[redacted]	8080	/mvc/api/v1/mobile/platformConfigs/time	GET	200 OK	
10/13/2016 4:13:32	[redacted]	8080	/mvc/api/v1/me	GET	200 OK	
10/13/2016 4:13:32	[redacted]	8080	/mvc/api/v1/mobile/bookCatalogs/top	GET	200 OK	
10/13/2016 4:13:32	[redacted]	8080	/mvc/api/v1/mobile/books/highlight?page=1	GET	404 Not Foun	
10/13/2016 4:13:32	[redacted]	8080	/mvc/api/v1/mobile/favoriteCategory	GET	200 OK	
10/13/2016 4:13:32	[redacted]	8080	/mvc/api/v1/mobile/announcement	GET	200 OK	

REQUEST

```
GET http [redacted] 080/mvc/api/v1/me HTTP/1.1
Host: 172.24.14.109:8080
[illegible]
Accept-Encoding: gzip, deflate
Content-Type: application/json
Accept-Language: en-us
Accept: application/json
Connection: keep-alive
Proxy-Connection: keep-alive
User-Agent: Mozilla/5.0 (iPad; CPU OS 9_3 like Mac OS X) AppleWebKit/601.1.46 (KHTML, like Gecko) Mobile/13E
```

RESPONSE

```
HTT [redacted]
Ser
Acc
Acc
Acc
Acc
Access-Control-Expose-Headers: X-refresh-token
X-refresh-token: eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJ[illegible]
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
Date: Thu, 13 Oct 2016 08:12:40 GMT

{"n
```



Request Start	Host	Port	Path	Method	Status	Category
10/13/2016 4:13:32	[redacted]	8080	/mvc/login	POST	200 OK	
10/13/2016 4:13:32	[redacted]	8080	/mvc/api/v1/mobile/platformConfigs/time	GET	200 OK	
10/13/2016 4:13:32	[redacted]	8080	/mvc/api/v1/me	GET	200 OK	
10/13/2016 4:13:32	[redacted]	8080	/mvc/api/v1/mobile/bookCatalogs/top	GET	200 OK	
10/13/2016 4:13:32	[redacted]	8080	/mvc/api/v1/mobile/books/highlight?page=1	GET	404 Not Foun	
10/13/2016 4:13:32	[redacted]	8080	/mvc/api/v1/mobile/favoriteCategory	GET	200 OK	
10/13/2016 4:13:32	[redacted]	8080	/mvc/api/v1/mobile/announcement	GET	200 OK	

REQUEST

```
GET htt [redacted] 09:8080/mvc/api/v1/mobile/bookCatalogs/top HTTP/1.1
Host: [illegible]
Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJ[illegible]
Accept-Encoding: gzip, deflate
Content-Type: application/json
Accept-Language: en-us
Accept: application/json
Connection: keep-alive
Proxy-Connection: keep-alive
User-Agent: Mozilla/5.0 (iPad; CPU OS 9_3 like Mac OS X) AppleWebKit/601.1.46 (KHTML, like Gecko) Mobile/13E
```

RESPONSE

```
HTTP/1.1 200 OK
Ser [redacted]
Acc
Acc
Acc
Acc
Access-Control-Expose-Headers: X-refresh-token
X-refresh-token: [illegible]
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
Date: Thu, 13 Oct 2016 08:12:40 GMT

[]
```

To handle this state management scheme the following response state pattern is used:

```
<ResponseStatePatterns>
      <ResponseStateElement>
        <name>myToken</name>
        <ReplaceRegexes>
          <string>Authorization:\sBearer\s(?'myToken'eyJhbGciOiJIUzI1NiJ9\.([a-
zA-Z0-9]{1,140})\.([a-zA-Z0-9_-]{1,45}))</string>
        </ReplaceRegexes>
        <SearchRegexes>
          <string>token":"(?'myToken'eyJhbGciOiJIUzI1NiJ9\.([a-zA-Z0-
9]{1,140})\.([a-zA-Z0-9_-]{1,45}))</string>
          <string>X-refresh-token:\s(?'myToken'eyJhbGciOiJIUzI1NiJ9\.([a-zA-Z0-
9]{1,140})\.([a-zA-Z0-9_-]{1,45}))</string>
        </SearchRegexes>
      </ResponseStateElement>
</ResponseStatePatterns>
```

An explanation of the above is as follows:

There are 2 patterns we need to parse in the HTTP Response, a “token” value in a json response body:

```
{"token":"eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJQVDEuSU5XTSIsImFjY291bnQiOiJQVDEuSU5XTSIsInR4SWQiOiIyYzI
zNzg1Yy0zZWRkLTQzYTMtYjg2NC1iMzEzMmRkNjgxYzkiLCJpYXQiOjE0NzYzNDYzNjB9.p_U-
EavAS8unQewF5Q7i0Bb5iBfzJLvGlKVJWJrplIA","passwordChangedDays":null,"shouldNoticeChangePassword":null,
"racfLoginSuccess":false,"racfLoginCode":null,"racfLoginMessage":null}
```

And an “X-Refresh-Token” in a response header:

```
X-refresh-token:
eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJQVDEuSU5XTSIsImFjY291bnQiOiJQVDEuSU5XTSIsInR4SWQiOiIzODY5MTYy
ZC1jZDgzLTRhOTgtYTY5ZS04M2E5ZjFhODk4MDYiLCJpYXQiOjE0NzYzNDYzNjB9.zn-tDPMrrp0ep-
7qFpYYqGQleMrk7hjhBgxUIszQirs
```

The following two Search Regexes will look for these patterns in every response and assign the matched string to the “myToken” capture group.

```
          <SearchRegexes>
                    <string>token":"(?'myToken'eyJhbGciOiJIUzI1NiJ9\.([a-zA-Z0-9]{1,140})\.([a-zA-Z0-9_-]{1,45}))</string>
                    <string>X-refresh-token:\s(?'myToken'eyJhbGciOiJIUzI1NiJ9\.([a-zA-Z0-9]{1,140})\.([a-zA-Z0-9_-
]{1,45}))</string>
          </SearchRegexes>
```

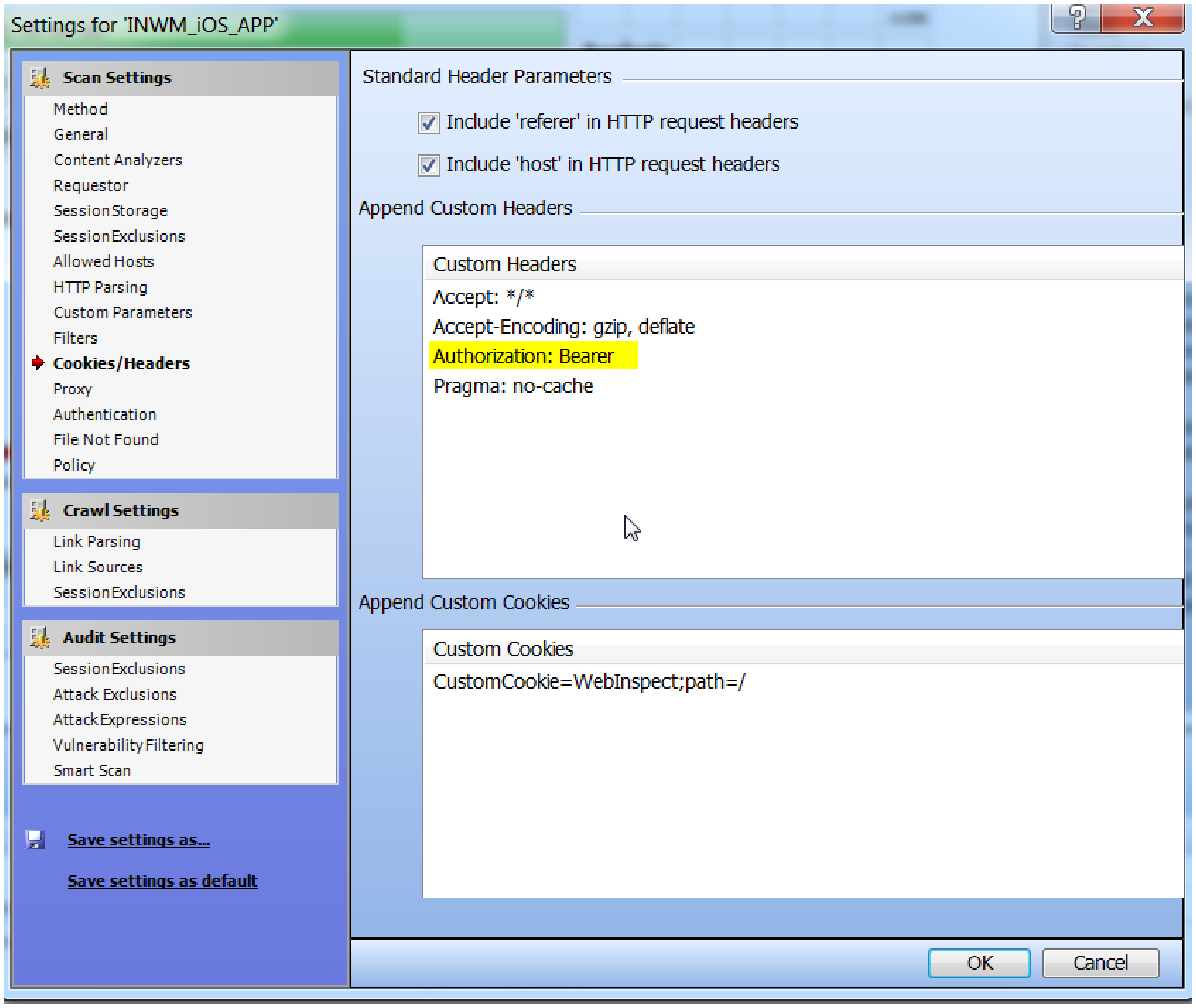
In each request, we want to replace the following header, for example, with one containing the string we matched in the responses:

```
Authorization: Bearer
eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJQVDEuSU5XTSIsImFjY291bnQiOiJQVDEuSU5XTSIsInR4SWQiOiIyYzIzNzg1Yy
0zZWRkLTQzYTMtYjg2NC1iMzEzMmRkNjgxYzkiLCJpYXQiOjE0NzYzNDYzNjB9.p_U-
EavAS8unQewF5Q7i0Bb5iBfzJLvGlKVJWJrplIA
```

To accomplish that we use the following Replace Regex:

```
<ReplaceRegexes>
        <string>Authorization:\sBearer\s(?'myToken'eyJhbGciOiJIUzI1NiJ9\.([a-zA-Z0-9]{1,140})\.([a-zA-Z0-9_-
]{1,45}))</string>
</ReplaceRegexes>
```

And since it's non-standard, we also need to add a custom header “Authorization: Bearer” to every request:


Settings for 'INWM_iOS_APP'

Standard Header Parameters

- [✓] Include 'referer' in HTTP request headers
- [✓] Include 'host' in HTTP request headers

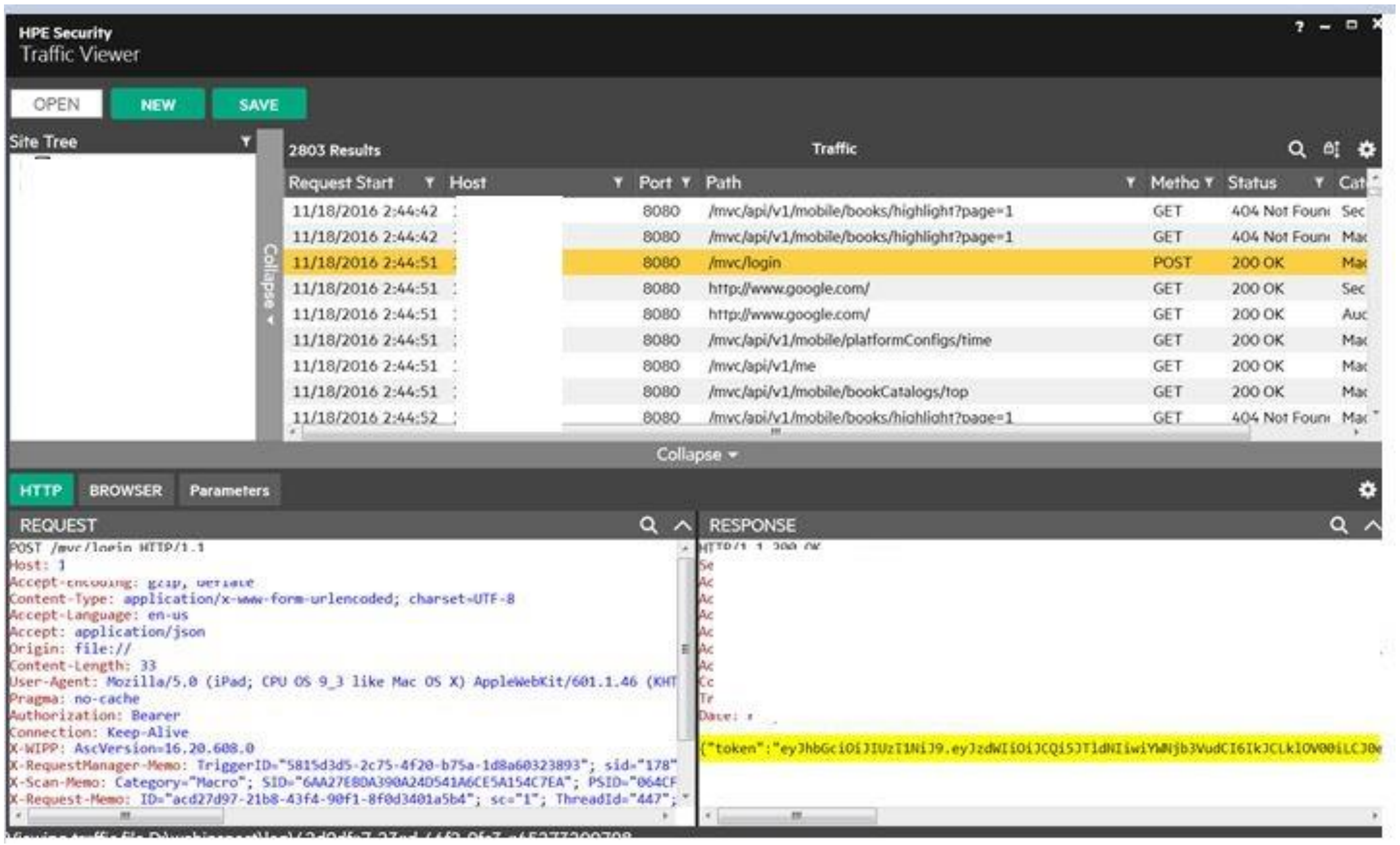
Append Custom Headers

Custom Headers
Accept: */*
Accept-Encoding: gzip, deflate
Authorization: Bearer
Pragma: no-cache

Append Custom Cookies

Custom Cookies
CustomCookie=WebInspect;path=/

The resulting traffic looks like this:



2803 Results

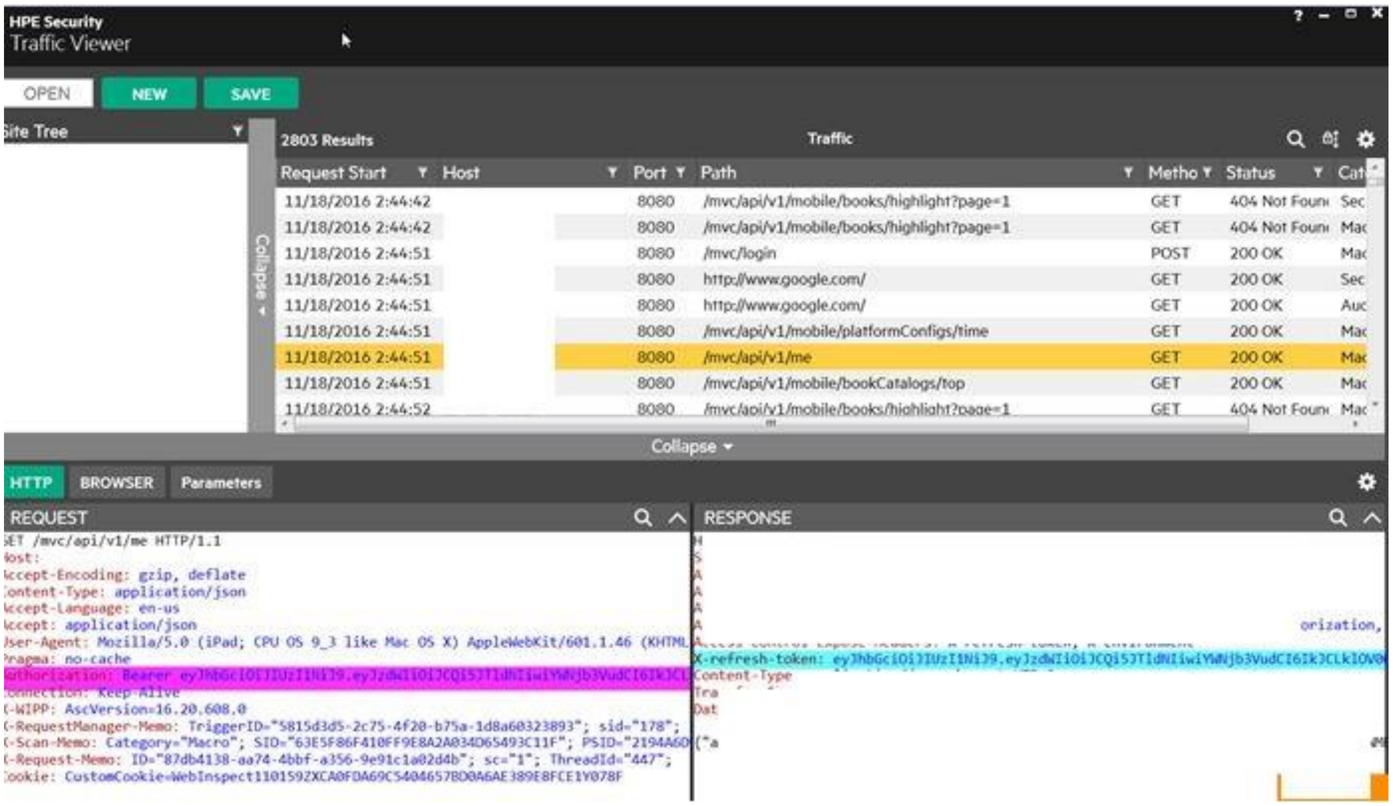
Request Start	Host	Port	Path	Metho	Status	Cat
11/18/2016 2:44:42	[redacted]	8080	/mvc/api/v1/mobile/books/highlight?page=1	GET	404 Not Foun	Sec
11/18/2016 2:44:42	[redacted]	8080	/mvc/api/v1/mobile/books/highlight?page=1	GET	404 Not Foun	Mac
11/18/2016 2:44:51	[redacted]	8080	/mvc/login	POST	200 OK	Mac
11/18/2016 2:44:51	[redacted]	8080	http://www.google.com/	GET	200 OK	Sec
11/18/2016 2:44:51	[redacted]	8080	http://www.google.com/	GET	200 OK	Auc
11/18/2016 2:44:51	[redacted]	8080	/mvc/api/v1/mobile/platformConfigs/time	GET	200 OK	Mac
11/18/2016 2:44:51	[redacted]	8080	/mvc/api/v1/me	GET	200 OK	Mac
11/18/2016 2:44:51	[redacted]	8080	/mvc/api/v1/mobile/bookCatalogs/top	GET	200 OK	Mac
11/18/2016 2:44:52	[redacted]	8080	/mvc/api/v1/mobile/books/highlight?page=1	GET	404 Not Foun	Mac

REQUEST

```
POST /mvc/login HTTP/1.1
Host: 1
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Accept-Language: en-us
Accept: application/json
Origin: file://
Content-Length: 33
User-Agent: Mozilla/5.0 (iPad; CPU OS 9_3 like Mac OS X) AppleWebKit/601.1.46 (KHT
Pragma: no-cache
Authorization: Bearer
Connection: Keep-Alive
X-WIPP: AscVersion=16.20.608.0
X-RequestManager-Memo: TriggerID="5815d3d5-2c75-4f20-b75a-1d8a60323893"; sid="178"
X-Scan-Memo: Category="Macro"; SID="6AA27E8DA390A24D541A6CE5A154C7EA"; PSID="064CF
X-Request-Memo: ID="acd27d97-21b8-43f4-90f1-8f0d3401a5b4"; sc="1"; ThreadId="447";
```

RESPONSE

```
HTTP/1.1 200 OK
Se
Ac
Ac
Ac
Ac
Ac
Ac
Cc
Tr
Date:

{"token":"eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJCQi5JT1dNIiwiYWNjb3VudCI6IkJCLkIOV00iLCJ0e
```



2803 Results

Request Start	Host	Port	Path	Metho	Status	Cat
11/18/2016 2:44:42		8080	/mvc/api/v1/mobile/books/highlight?page=1	GET	404 Not Foun	Sec
11/18/2016 2:44:42		8080	/mvc/api/v1/mobile/books/highlight?page=1	GET	404 Not Foun	Mac
11/18/2016 2:44:51		8080	/mvc/login	POST	200 OK	Mac
11/18/2016 2:44:51		8080	http://www.google.com/	GET	200 OK	Sec
11/18/2016 2:44:51		8080	http://www.google.com/	GET	200 OK	Auc
11/18/2016 2:44:51		8080	/mvc/api/v1/mobile/platformConfigs/time	GET	200 OK	Mac
11/18/2016 2:44:51		8080	/mvc/api/v1/me	GET	200 OK	Mac
11/18/2016 2:44:51		8080	/mvc/api/v1/mobile/bookCatalogs/top	GET	200 OK	Mac
11/18/2016 2:44:52		8080	/mvc/api/v1/mobile/books/highlight?page=1	GET	404 Not Foun	Mac

```
GET /mvc/api/v1/me HTTP/1.1
Host:
Accept-Encoding: gzip, deflate
Content-Type: application/json
Accept-Language: en-us
Accept: application/json
User-Agent: Mozilla/5.0 (iPad; CPU OS 9_3 like Mac OS X) AppleWebKit/601.1.46 (KHTML
Pragma: no-cache
Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJCQi5JTldNIiwiYWNjb3VudCI6IkJCL
Connection: Keep-Alive
X-WIPP: AscVersion=16.20.608.0
X-RequestManager-Memo: TriggerID="5815d3d5-2c75-4f20-b75a-1d8a60323893"; sid="178";
X-Scan-Memo: Category="Macro"; SID="63E5F86F410FF9E8A2A034D65493C11F"; PSID="2194A6D
X-Request-Memo: ID="87db4138-aa74-4bbf-a356-9e91c1a02d4b"; sc="1"; ThreadId="447";
Cookie: CustomCookie=WebInspect110159ZXCA0FDA69C5404657BD0A6AE389E8FCE1Y078F
```

```
H
S
A
A
A
A                                                              orization,
A
X-refresh-token: eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJCQi5JTldNIiwiYWNjb3VudCI6IkJCLkIOV0
Content-Type
Tra
Dat

("a
```