Using Burp to Test Session Token Generation

Session management mechanisms can be vulnerable to attack if tokens are generated in an unsafe manner that enables an attacker to predict values of that have been issued to other users. A password recovery token, sent to the user's registered email address is an example where an application's secu depends on the unpredictability of tokens it generates.

You can use Burp Suite to analyze tokens generated by a web application. This article demonstrates how to analyze and test token generation using the Intruder, Sequencer and Decoder tools.

In this example we are using three pages from the "Attacking session management" section of the "MDSec Training Labs".

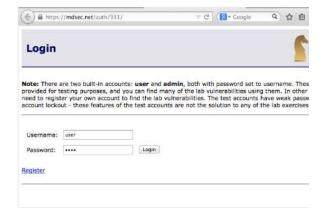
Using Burp Decoder to Test Session Tokens

First, ensure that Burp is correctly configured with your browser.

Ensure "Intercept is off" in the Proxy "Intercept" tab.



Locate the page you wish to test and ensure that any required details are entered in order to produce an appropriate response that contains a session token.



Return to Burp and ensure "Intercept is on" in the Proxy "Intercept" tab.

Submit a request, in this example by clicking the "Login" button.

The request will be captured by Burp. Use the "Forward" button to view the HTTP response containing the session token.





The HTTP response will now be displayed in the Proxy "Intercept" tab.

The cookie "SessionId_331" is the token used to track the session.

Raw Params Headers Hex

GET /auth/331/Home.ashx HTTP/1.1

Host: mdsec.net
User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 5_1 lik
Safari/7534.48.3

Accept: text/html,application/xhtml+xml,application/xm
Accept-Language: en-GB,en;q=0.5

Accept-Encoding: gzip, deflate
Referer: httms://mdsec.net/auth/331/Default.ashx2usern
Cookie: SessionId 331=757365726E616D653D757365727C756:
Connection: keep-alive

Select and highlight the full token.

Right click anywhere on the request to bring up the context menu.

Click "Send to Decoder".

Sarari//534.40.3
Accept: text/html,application/xhtml+xml,application/xm
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://mdsec.net/auth/331/Default.ashx
Cookie: SessionId__331=757365726E616D653D757365727C756
Connection: keep-alive
Cache-Control: max-age=0

Send to Spider
Do an active scan
Send to Intruder
Send to Repeater
Send to Repeater
Send to Comparer
Send to Comparer
Send to Decoder
Request in browser
Engagement tools
Change request method

Go to the "Decoder" tab. The token from the request will be displayed in the Decoder form.

The token may initially appear to be a long random string. However, on closer inspection, you can see that it contains only hexadecimal characters.



Guessing that the string may actually be a hex encoding of a string of ASCII characters, you can run it through the Decoder.

Use the drop down menu and select the appropriate encoding string to reveal the results.





The results will be displayed below in a second form box.

In this example we can see how the token has been created using a transformation of the user's username, UID and timestamp.



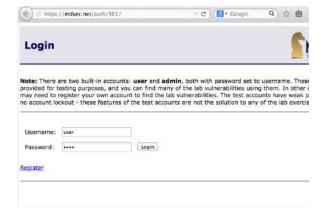
Using Burp Sequencer to Test Session Tokens

First, ensure that Burp is correctly configured with your browser.

Ensure "Intercept is off" in the Proxy "Intercept" tab.



Locate the page you wish to test and ensure that any required details are entered in order to produce an appropriate response that contains a session token.



Return to Burp and ensure "Intercept is on" in the Proxy "Intercept" tab.

Submit a request, in this example by clicking the "Login" button.

The request will be captured by Burp. Use the "Forward" button to view the HTTP response containing the session token.



The HTTP response will now be displayed in the Proxy "Intercept" tab.

In this example, the cookie "SessionId_361" is the token used to track the session.

Raw Headers Hex HTML Render

HTTP/1.1 302 Found

Date: Thu, 09 Apr 2015 13:00:40 GMT

Server: Microsoft-IIS/6.0

MicrosoftOfficeWebServer: 5.0 Pub

X-Powered-By: ASP.NET

X-AspNet-Version: 2.0.50727

Location: /auth/361/Home.ashx

Set-Cookie: SessionId 361=3512088CA196DC60; secure; H'
Cache-Control: no-cache

Pragma: no-cache

Expires: -1

Content-Type: text/html; charset=utf-8

Content-Length: 142

<html><head><title>Object moved</title></head><body>
<h2>Object moved to Aref="%2fauth%2561%2fHome.ashx">Aref="%2fauth%2536

Right click anywhere on the request to bring up the context menu.

Click "Send to Sequencer".

Raw Params Headers /auth/361/Hor .ashx HTTP/1. Host: mdsec.net User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 5_1 like Safari/7534.48.3 Accept: text/html,application/xhtml+xml,application/xm Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, defl
Referer: https://mdsec.net/
Cookie: SessionId_361=3512
Connection: keep-alive
Cache-Control: max-age=0 Send to Spider Do an active scan Send to Intruder Send to Repeater Send to Sequencer Send to Comparer Send to Decoder Request in browser Engagement tools

Ensure that you have selected the correct request from the "Select Live Capture Request" table and click the "Start live capture" button.

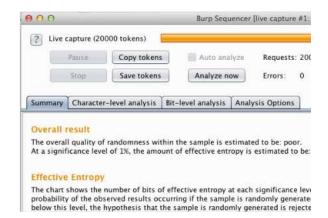


The "Burp Sequencer [live capture]" window will pop up.

Burp Sequencer will repeatedly issue the request and extract the relevant token from the application's responses.

The window shows the progress of the capture, and the number of tokens that have been obtained.

You can find out more about how the randomness test works, analyzing the results and the various analysis options in the full documentation for Burp Sequencer.

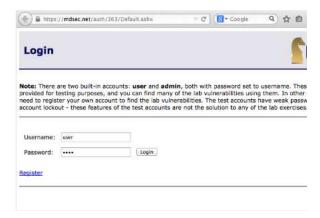


First, ensure that Burp is correctly configured with your browser.

Ensure "Intercept is off" in the Proxy "Intercept" tab.



Locate the page you wish to test and ensure that any required details are entered in order to produce an appropriate response that contains a session token.



Return to Burp and ensure "Intercept is on" in the Proxy "Intercept" tab.

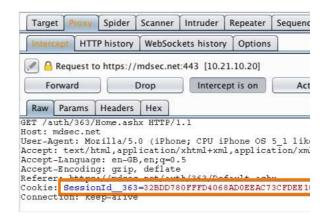
Submit a request, in this example by clicking the "Login" button.

The request will be captured by Burp. Use the "Forward" button to view the HTTP request containing the session token.



The HTTP request will now be displayed in the Proxy "Intercept" tab.

In this example, the cookie "SessionId_336" is the token used to enable the session.



Right click anywhere on the request to bring up the context menu.

Click "Send to Intruder".

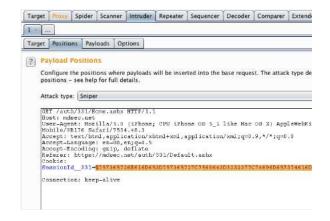
Raw Params Headers Hex

GET /auth/363/Home.ashx HTTP/1.1
Host: mdsec.net
User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 5_1 like
Accept: text/html,application/xhtml+xml,application/xm
Accept-Encoding: gzip, deflate
Referer: https://mdsec.net/aut
Cookie: SessionId_363=32BDD78
Connection: keep-alive

Send to Spider
Do an active scan
Send to Repeater
Send to Sequencer
Send to Comparer
Send to Comparer
Send to Decoder
Request in browser
Engagement tools

Go to the "Intruder" tab, then the "Positions" tab.

Ensure that the token you wish to test is the only position selected in the HTTP response.



Go to the "Payloads" tab.

Under the "Payload Sets" header, use the drop down menu to select either the "Character frobber" or "Bit flipper" payload type.

In this example we will continue with the "Character frobber".

You can find more about these payload types in the full documentation.

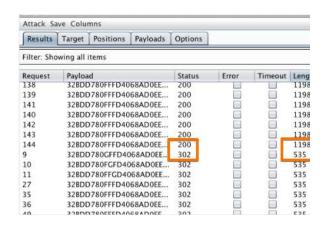
With the appropriate payload type selected, click the "Start Attack" button on the right of the Burp console.

Target Positions Payloads Options ? Payload Sets You can define one or more payload sets. The number of paylo available for each payload set, and each payload type can be c Payload set: Payload count: Payload type: Simple list Request count: Dates Brute forcer Payload Op Null payloads Character frobber This payload aple list of strings th Bit flipper Paste Username generator

The "Character frobber" payload type operates on a string input and modifies the value of each character position in turn.

We can use the results of this attack to assess which characters affect the validity of the token.

In this example, by sorting the results by length and/or status, we can clearly see how useful the "Character frobber" can be when testing which parts of a complex session token are actually being used to track session state.



Burp Decoder documentation

Getting started with Burp Proxy

Using Burp Intruder

Burp Sequencer documentation

Using Burp to attack session management

Burp Suite

Web vulnerability scanner Burp Suite Editions Release Notes

Vulnerabilities

Cross-site scripting (XSS) SQL injection Cross-site request forgery XML external entity injection Directory traversal Server-side request forgery

Customers

Organizations Testers Developers

Company

About
PortSwigger News
Careers
Contact
Legal
Privacy Notice

Insights

Web Security Academy Blog Research The Daily Swig

PortSwig



© 2020 PortSwigger Ltc