Introducing Burp Extractor

The Problem With Tokens and Scanning

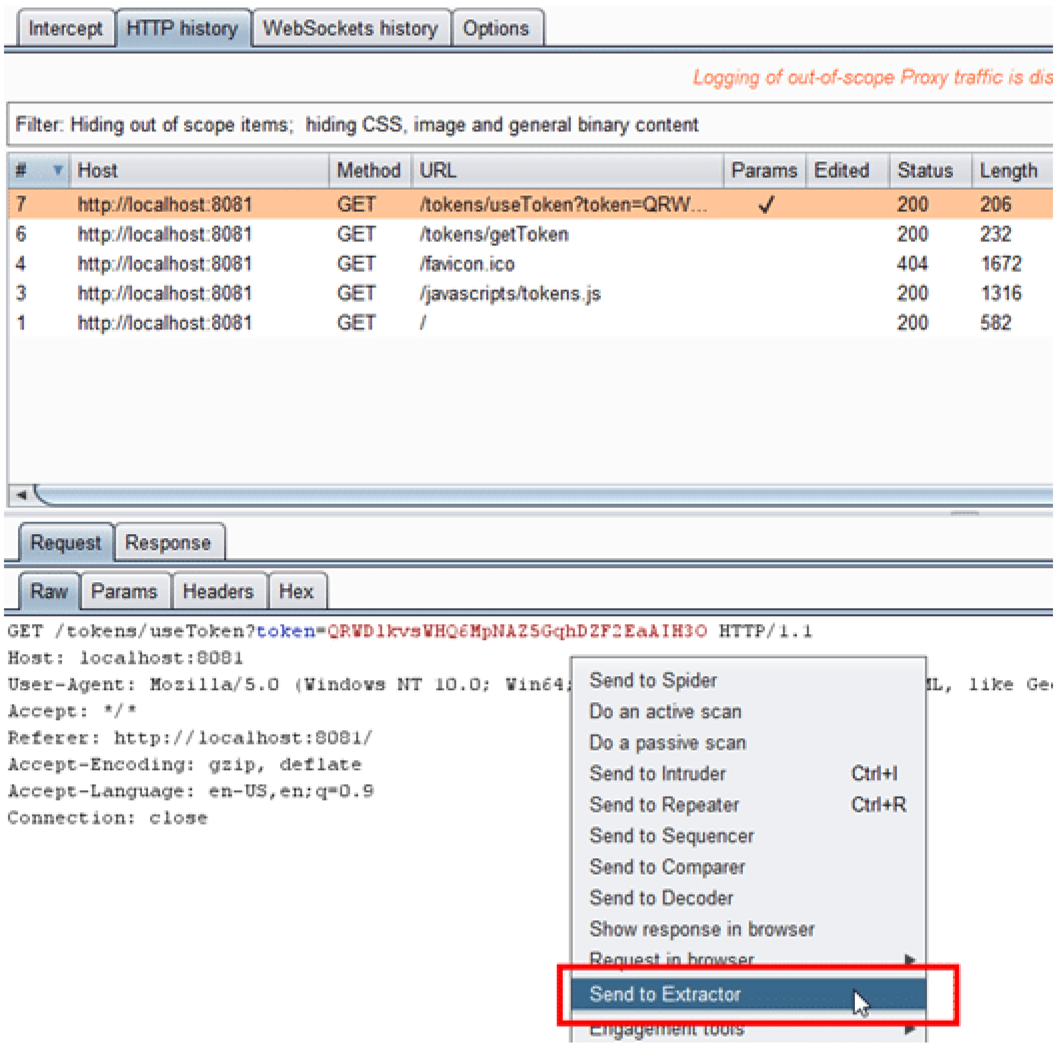
Burp Suite’s cookie jar is an incredibly handy tool that makes a penetration tester’s life much easier in many situations. It makes for a no hassle way to reissue requests in Repeater as a different user, scan requests which were originally issued in a previous session, and other fun things. But what happens when you need to change more than just cookies? Sometimes an application requires an anti-CSRF token, an updated expiration time, or maybe a session is tracked in an Authorization header instead of cookies. Normally with Burp, this could cause a major headache and a match/replace might be used, or it could be worth struggling through setting up a macro. Both methods requiring a good amount of manual intervention.

Burp Extractor as an Easy-to-Use Solution

Burp Extractor is intended to be a pain killer for these headaches. This extension allows penetration testers to select a string to extract from responses using a regular expression-based matching mechanism, and insert that string into any request with a similar matching mechanism, effectively making a cookie jar for whatever dynamic information the tester needs.

Using Burp Extractor

If a request requires a value from a response, right click on that request and select “Send to Extractor”. Then find a response where the client receives this value from, and select “Send to Extractor”.



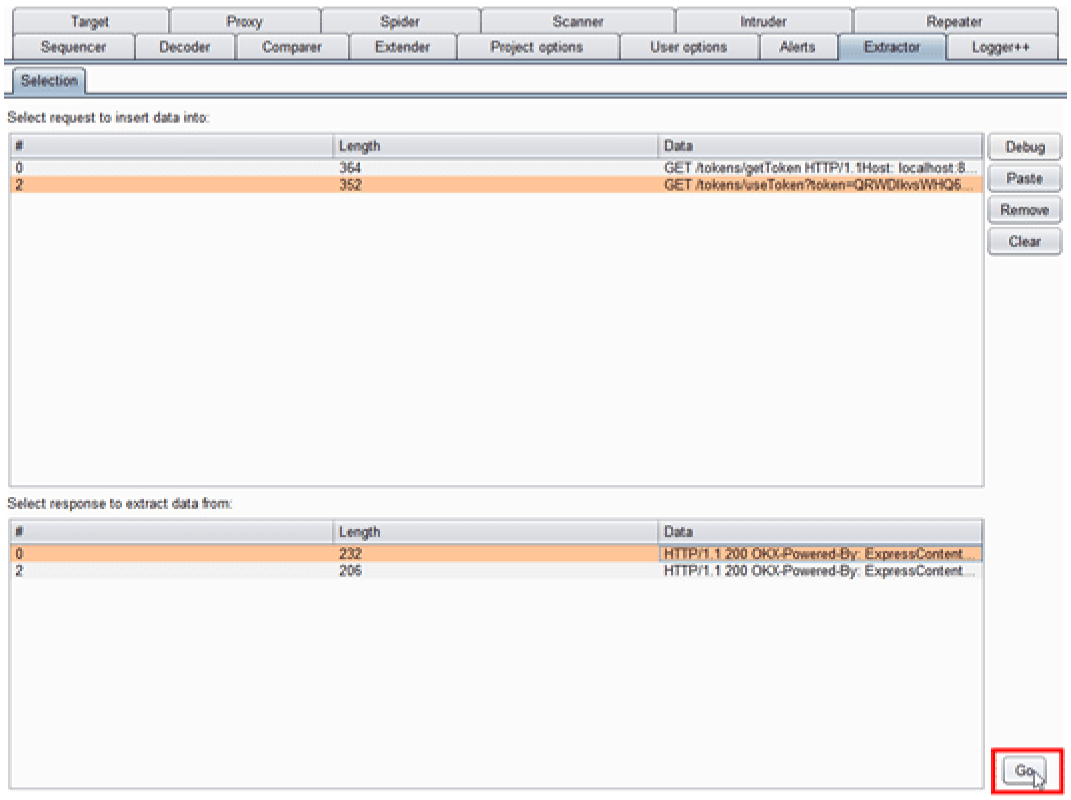
Logging of out-of-scope Proxy traffic is dis

Filter: Hiding out of scope items; hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Status	Length
7	http://localhost:8081	GET	/tokens/useToken?token=QRW...	✓		200	206
6	http://localhost:8081	GET	/tokens/getToken			200	232
4	http://localhost:8081	GET	/favicon.ico			404	1672
3	http://localhost:8081	GET	/javascripts/tokens.js			200	1316
1	http://localhost:8081	GET	/			200	582

```
GET /tokens/useToken?token=QRWDlkvsWHQ6MpNAZ5GqhDZF2EaAIH3O HTTP/1.1
Host: localhost:8081
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64;                    ML, like Ge
Accept: */*
Referer: http://localhost:8081/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close
```

Go to the Extractor tab to view a Comparer-like interface, and select the request and response needed, then click “Go”.



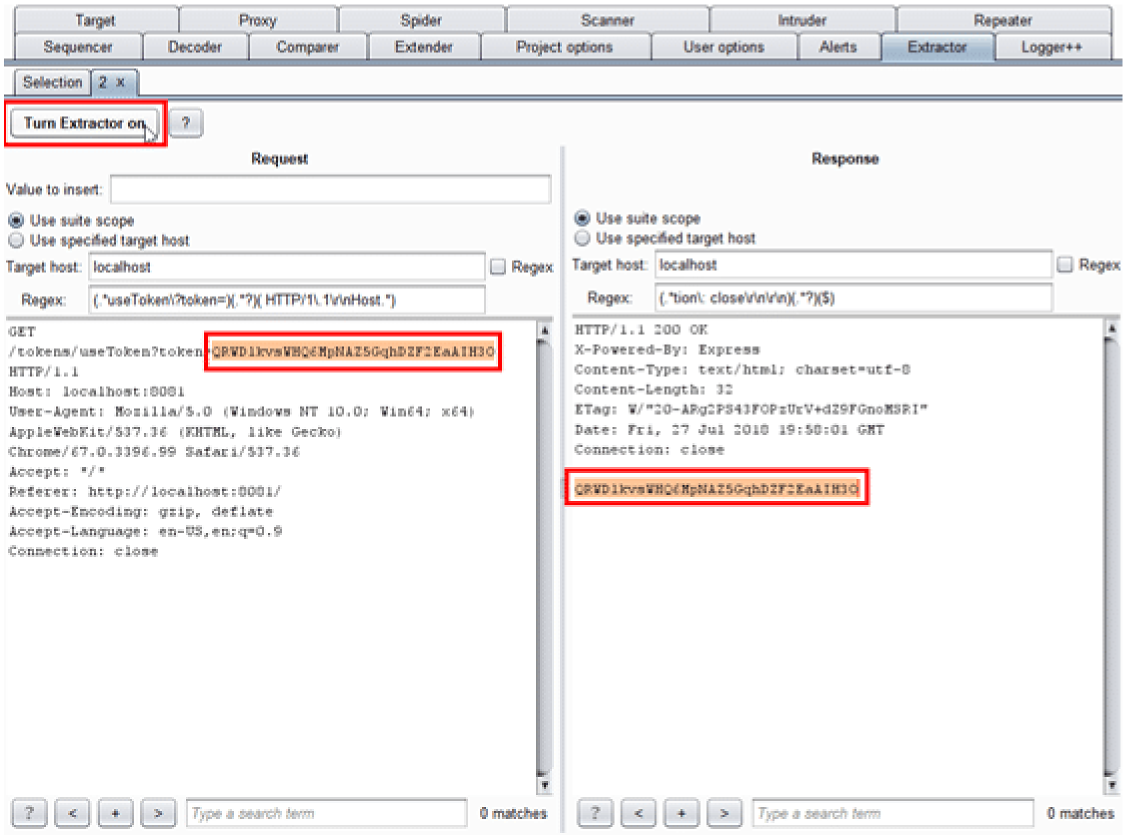
Select request to insert data into:

#	Length	Data
0	364	GET /tokens/getToken HTTP/1.1Host: localhost:8...
2	352	GET /tokens/useToken?token=QRWDlkvsWHQ6...

Select response to extract data from:

#	Length	Data
0	232	HTTP/1.1 200 OKX-Powered-By: ExpressContent...
2	206	HTTP/1.1 200 OKX-Powered-By: ExpressContent...

Within the newly created tab, highlight the content of the request which needs to be replaced, and the content of the response which contains the value to be inserted. Adjust the scope as necessary, and click “Turn Extractor on”.



Request

Value to insert:

(● Use suite scope) Use specified target host

Target host: localhost ☐ Regex

Regex: (.*useToken\?token=)(.*?)(HTTP/1\.1\r\nHost.*)

```
GET
/tokens/useToken?token=QRWDlkvsWHQ6MpNAZ5GqhDZF2EaAIH3O
HTTP/1.1
Host: localhost:8081
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/67.0.3396.99 Safari/537.36
Accept: */*
Referer: http://localhost:8081/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close
```

Response

(● Use suite scope) Use specified target host

Target host: localhost ☐ Regex

Regex: (.*tion\: close\r\n\r\n)(.*?)($)

```
HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: text/html; charset=utf-8
Content-Length: 32
ETag: W/"20-ARg2PS43FOPzUrV+dZ9FGnoMSRI"
Date: Fri, 27 Jul 2018 19:58:01 GMT
Connection: close

QRWDlkvsWHQ6MpNAZ5GqhDZF2EaAIH3O
```

Once turned on, Extractor will look for occurrences of the regex listed in the request and response panels, and extract or insert data appropriately. It will also update the “Value to insert” field with the newest value extracted.

Check out the demonstration below to see an example usage of the extension, and grab the extension off GitHub! Hopefully this tool will help ease some pains when dealing with web apps that use tokens or authentication that Burp is not always well-equipped to cope with.

Example usage of Burp Extractor