



# Burp Suite



Presented By:  
Joe McCray

[joe@strategicsec.com](mailto:joe@strategicsec.com)

<http://www.linkedin.com/in/joemccray>

<http://twitter.com/j0emccray>



# Burp Suite

- Burp is a set of tools, all tightly integrated
  - Proxy
  - Spider
  - Scanner
  - Intruder
  - Repeater
  - Sequencer
- API
- Save, search, compare, decode, filter

There is an **API** for that!



# Burp Suite

- Burp is a set of tools, all tightly integrated
  - Proxy
  - Spider
  - ~~Scanner~~
  - ~~Intruder~~
  - Repeater
  - Sequencer
- ~~API~~
- ~~Save, search,~~ compare, decode, filter



# Burp Suite

- Burp is a set of tools, all tightly integrated:
  - Proxy
  - Spider
  - ~~Scanner~~
  - ~~Intruder~~
  - Repeater
  - Sequencer
- ~~API~~
- ~~Save, search,~~ compare, decode, filter



# Burp Suite

The screenshot displays the Burp Suite Free Edition v1.5 interface. The top menu bar includes Burp, Intruder, Repeater, Window, and Help. Below it is a toolbar with buttons for Target, Proxy, Spider, Scanner, Intruder, Repeater, Sequencer, Decoder, Comparer, Options, and Alerts. The main interface is divided into two panes. The left pane shows a site map with a tree view of the website structure. The right pane shows a list of HTTP requests and responses.

Filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders

Host	Method	URL	Params	Status	Length	MIME type	Title	Comment	Time requested
http://strategicsec.com	GET	/comments/feed/				XML			

Request Response

Raw Headers Hex

```
GET /comments/feed/ HTTP/1.1
Host: strategicsec.com
Accept: */*
Accept-Language: en
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)
Connection: close
```

0 matches

# Burp for FREE

- Use the free version

<https://portswigger.net/burp/download.html>

- For this class we will be using the FREE version of Burp





# Proxy

- Always use a proxy with your browser
  - Use a separate browser to hack
  - Have it sent all traffic through the Burp proxy

- Easily done with Firefox



- You can set up multiple profiles
- Proxy is not system wide
- There are lots of plugins available

## ✓ The Top 5 Burp Suite Extensions - Bug Hunt

[bughunting.guide/the-top-5-burp-suite-extensions/](http://bughunting.guide/the-top-5-burp-suite-extensions/) ▼ Vertaal deze pagina

24 apr. 2015 - Both the free and paid versions of Burp support extensions that add extra

... (and free) scanner, an IP randomizer, or a plugin for validating XSS vulnerabilities. ...

Burp Notes adds an additional tab to your Burp Suite interface, ...



# Proxy

- Send “all” traffic to Burp Suite



	▲ Pattern Name	URL pattern	Whitelist (Inclusive) or
✓	all	*	Whitelist
✓	audio	\.(ogg mp3 wav)\$	Blacklist
✓	evsecure-ocsp.thawte.com	*evsecure-ocsp.thawte.com/*	Blacklist
✓	facebook	*.facebook.com/*	Blacklist
✓	imagens	\.(jpg png ico gif cur svg)(\?.+ \$)	Blacklist
✓	lastpass	*lastpass.com/*	Blacklist
✓	localhost	*://localhost/*	Blacklist
✓	localhost numérico	*://127.0.0.1/*	Blacklist
✓	mozilla.com	^https?://.*\.mozilla\.com/.*\$	Blacklist
✓	mozilla.org	^https?://.*\.mozilla\.org/.*\$	Blacklist
✓	OCSP thawte	http://ocsp.thawte.com/*	Blacklist
✓	ocsp.digicert.com	*ocsp.digicert.com/*	Blacklist
✓	ocsp.multicert.com	*ocsp.multicert.com/*	Blacklist
✓	ocsp.verisign.com	*ocsp.verisign.com/*	Blacklist
✓	safebrowsing	^https?://safebrowsing.*\.google\.com/.*\$	Blacklist
✓	safebrowsing SSL	^https?://sb-ssl\.google\.com/.*\$	Blacklist
✓	twitter	^https?://.*\.twitter\.com/.*\$	Blacklist



# Proxy

- Filtering further

Filter: Hiding image content; hiding 4xx responses; hiding specific extensions

<p> <b>Filter by request type</b></p> <p></p> <p><input type="checkbox"/> Show only in-scope items</p> <p><input type="checkbox"/> Hide items without responses</p> <p><input type="checkbox"/> Show only parameterized requests</p>	<p><b>Filter by MIME type</b></p> <table><tr><td><input checked="" type="checkbox"/> HTML</td><td><input checked="" type="checkbox"/> Other text</td></tr><tr><td><input checked="" type="checkbox"/> Script</td><td><input type="checkbox"/> Images</td></tr><tr><td><input checked="" type="checkbox"/> XML</td><td><input checked="" type="checkbox"/> Flash</td></tr><tr><td><input checked="" type="checkbox"/> CSS</td><td><input checked="" type="checkbox"/> Other binary</td></tr></table>	<input checked="" type="checkbox"/> HTML	<input checked="" type="checkbox"/> Other text	<input checked="" type="checkbox"/> Script	<input type="checkbox"/> Images	<input checked="" type="checkbox"/> XML	<input checked="" type="checkbox"/> Flash	<input checked="" type="checkbox"/> CSS	<input checked="" type="checkbox"/> Other binary	<p><b>Filter by status code</b></p> <p><input checked="" type="checkbox"/> 2xx [success]</p> <p><input checked="" type="checkbox"/> 3xx [redirection]</p> <p><input type="checkbox"/> 4xx [request error]</p> <p><input checked="" type="checkbox"/> 5xx [server error]</p>
<input checked="" type="checkbox"/> HTML	<input checked="" type="checkbox"/> Other text									
<input checked="" type="checkbox"/> Script	<input type="checkbox"/> Images									
<input checked="" type="checkbox"/> XML	<input checked="" type="checkbox"/> Flash									
<input checked="" type="checkbox"/> CSS	<input checked="" type="checkbox"/> Other binary									
<p><b>Filter by search term</b></p> <p><input type="text"/></p> <p><input type="checkbox"/> Regex</p> <p><input type="checkbox"/> Case sensitive <input type="checkbox"/> Negative search</p>	<p><b>Filter by file extension</b></p> <p><input type="checkbox"/> Show only: <input type="text" value="asp,aspx,jsp,php"/></p> <p><input checked="" type="checkbox"/> Hide: <input type="text" value="gif,jpg,png,css,woff,ico"/></p>	<p><b>Filter by annotation</b></p> <p><input type="checkbox"/> Show only commented items</p> <p><input type="checkbox"/> Show only highlighted items</p>								



# Proxy

- Auto-scroll
- just sort by # desc

Filter: Hiding CSS, image and general binary content

#	Host	Method	URL	Par
3	http://strategicsec.com	GET	/strategic-security-difference-2/	
13	http://strategicsec.com	GET	/wp-includes/js/jquery/jquery.js?ver=1....	
14	http://strategicsec.com	GET	/wp-includes/js/jquery/jquery-migrate.m...	
15	http://strategicsec.com	GET	/wp-content/plugins/login-with-ajax/wid...	
16	http://strategicsec.com	GET	/wp-content/plugins/toggle-box/js/toggl...	
17	http://strategicsec.com	GET	/wp-includes/js/json2.min.js?ver=2011-...	
18	http://strategicsec.com	GET	/wp-content/themes/strategicsecurity/j...	
19	http://strategicsec.com	GET	/wp-content/themes/strategicsecurity/j...	
20	http://strategicsec.com	GET	/wp-content/themes/strategicsecurity/j...	
21	http://strategicsec.com	GET	/wp-content/plugins/all-in-one-event-ca...	
22	http://strategicsec.com	GET	/wp-content/themes/strategicsecurity/j...	
24	http://strategicsec.com	GET	/wp-content/plugins/fancier-author-box...	
25	http://strategicsec.com	GET	/wp-includes/js/comment-reply.min.js?u...	

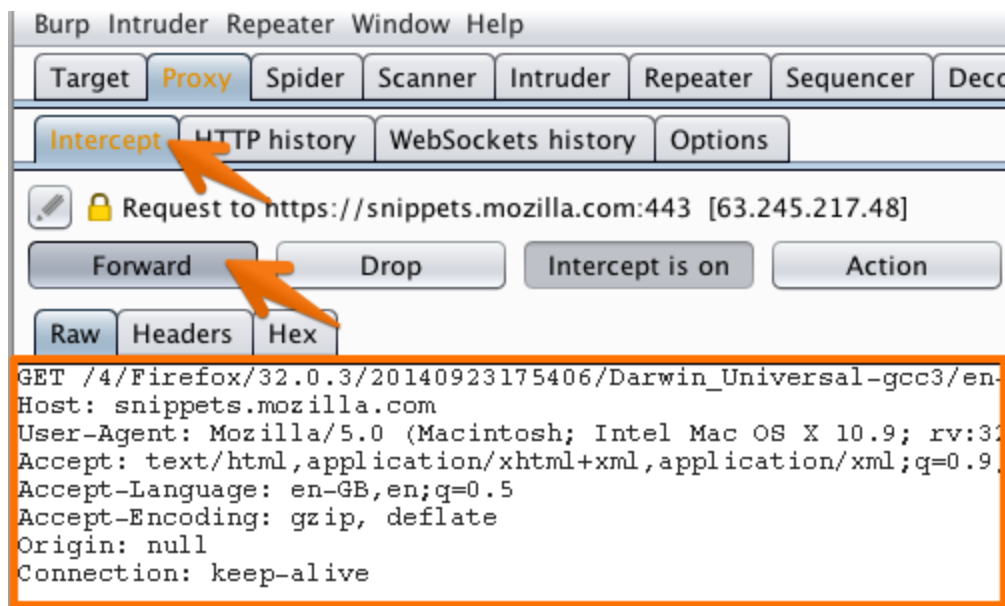


# Proxy

- What to look for when using the proxy feature?
  - Failing requests
  - Error and Debug messages
  - Sensitive information
  - Missing headers
- If want to get active:
  - input: URL parameters, POST-data, headers, cookies

# Proxy

- You can do simple, yet powerful, tests in two ways:
  - Intercepting requests
  - Repeating requests



# Proxy

Request to http://strategicsec.com:80 [204.244.123.113]

Forward

Drop

Intercept is on

Action

[Comment this item](#)

Raw

Params

Headers

Hex

POST /wp-login.php HTTP/1.1

Host: strategicsec.com

User-Agent: Mozilla/5.0 (Windows NT 6.2; WOW64; rv:7.0.1) Gecko/20100101 Firefox/7.0.1

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8

Accept-Language: en-us,en;q=0.5

Accept-Encoding: gzip, deflate

Accept-Charset: ISO-8859-1,utf-8;q=0.7,\*;q=0.7

Proxy-Connection: keep-alive

Referer: http://strategicsec.com/strategic-security-difference-2/

Cookie: PHPSESSID=h0ukhvb18ubdfha9rhcvf5tue7; \_\_utma=13636341.470710462.1388140835.1388140835.1388140835.1; \_\_utmb=13636341.3.10.1388140835; \_\_utmc=13636341; \_\_utmz=13636341.1388140835.1.1.utmcsr=(direct)|utmccn=(direct)|utmcmd=(none); \_\_qca=PO-1326908180-1388140841085; orbtr\_session=1388140841886; orbtr\_uid=1141189

Content-Type: application/x-www-form-urlencoded

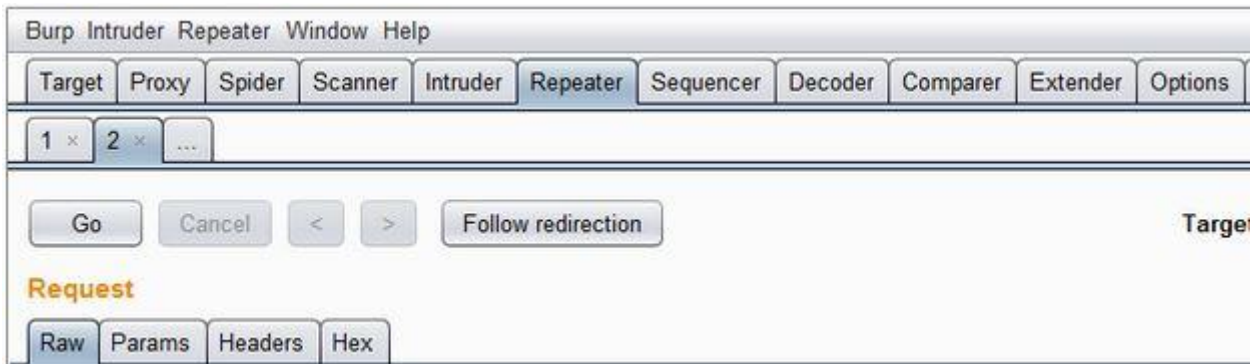
Content-Length: 28

log=demo&pwd=demo&wp-submit=



# Repeater

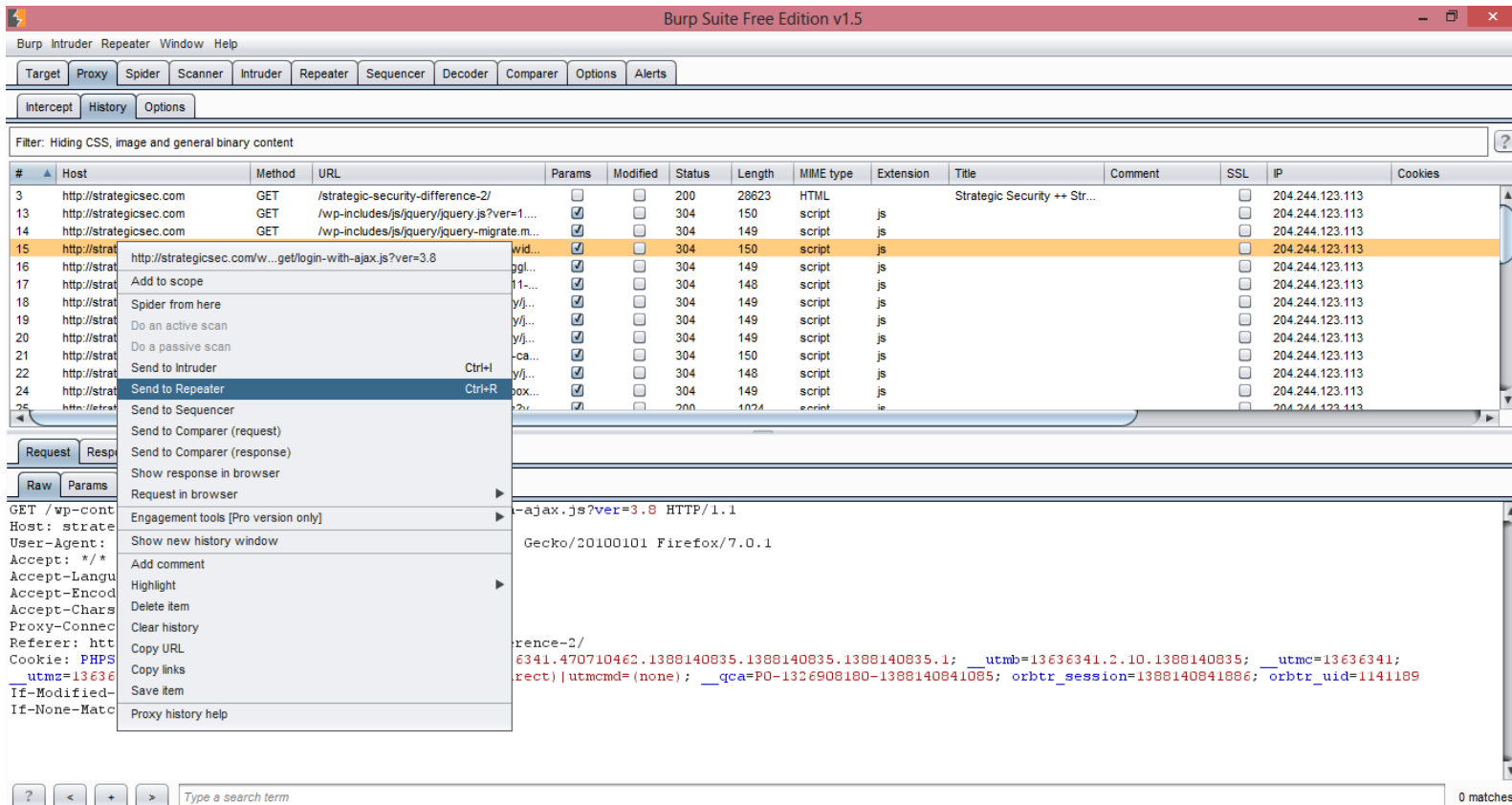
- Intercepting requests with the proxy is good for single tests
  - Or when you have a single shot
- For deeper testing use the Repeater
  - Allows arbitrary replay and modification of requests





# Repeater

- From Proxy to Repeater



# Repeater

## Request

Raw

Params

Headers

Hex

```
POST /wp-login.php HTTP/1.1
Host: strategicsec.com
User-Agent: Mozilla/5.0 (Windows NT 6.2; WOW64; rv:7.0.1) Gecko/20100101
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip, deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Proxy-Connection: keep-alive
Referer: http://strategicsec.com/strategic-security-difference-2/
Cookie: PHPSESSID=h0ukhvb18ubdfha9rhcvf5tue7; __utma=13636341.47071046
```

?

<

+

>

Type a search term

## Response

Raw

Headers

Hex

HTML

Render

```
HTTP/1.1 200 OK
Date: Fri, 27 Dec 2013 10:52:24 GMT
Server: Apache/2.2.22 (EL)
X-Powered-By: PHP/5.2.17
X-CF-Powered-By: WP 1.3.10
Expires: Wed, 11 Jan 1984 05:00:00 GMT
Cache-Control: no-cache, must-revalidate, max-age=0
Pragma: no-cache
Set-Cookie: wordpress_test_cookie=WP+Cookie+check; path=/
X-Frame-Options: SAMEORIGIN
Vary: Accept-Encoding
Content-Length: 13621
Connection: close
Content-Type: text/html; charset=UTF-8
```



# Repeater

- With the Repeater you can just play with the requests, whatever your objective is:
  - Debugging
  - Functional Testing
  - Security Testing

# Repeater

- XSS - a simple payload to get 80/20
  - "><img src=a onerror=alert(1)>
- Using the repeater avoids browser defensive measures:

- auto URL encoding
- XSS filters





# Repeater

## Request

Raw Params Headers Hex

Content-Length: 1778

-----41184676334

Content-Disposition: form-data; name="input\_2.3"

<script>alert(1);</script>

-----41184676334

Content-Disposition: form-data; name="input\_2.6"

fsdfs

? < + > Type a search term

## Response

Raw Headers Hex HTML Render

<ul id='gform\_fields\_2' class='gform\_fields top\_label description\_below'><li id='field\_2\_2' class='gfield gfield\_contains\_required'><label class='gfield\_label' for='input\_2\_2\_3'>Name<span class='gfield\_required'>\*</span></label><div class='ginput\_complex ginput\_container' id='input\_2\_2'><span id='input\_2\_2\_3\_container' class='ginput\_left'><input type='text' name='input\_2.3' id='input\_2\_2\_3' value='&lt;&script&gt;alert(1);&lt;/script&gt;' tabindex='1' /><label for='input\_2\_2\_3'>First</label></span><span id='input\_2\_2\_6\_container' class='ginput\_right'><input type='text' name='input\_2.6' id='input\_2\_2\_6' value='fsdfs' tabindex='2' /><label for='input\_2\_2\_6'>Last</label></span></div>



# Repeater

- SQLi - you don't have to test for it, because you can use prepared statements





# Repeater

- SQL-injection – No need to manually test for it, because you can use the prepared statements from Burp Suite!
- ’ (the mother of all SQLi) ☺
- Just in case
- and benchmark(10000000,md5(md5(1))) --%20

# Repeater

```
__uom2=13636341.1366140633.1.1.uomcsi-(direct) | uomccn-(direct) | uomcmd-(none); __qca=PU-1326906160-13661
wordpress_test_cookie=WP+Cookie+check
Content-Type: multipart/form-data; boundary=-----41184676334
Content-Length: 1794
```

```
-----41184676334
Content-Disposition: form-data; name="input_2.3"
```

```
and benchmark(100000000, md5(md5(1))) --%20
```

? < + >

## Response

Raw Headers Hex HTML Render

```
media='all' />
```

```
<div class='gf_browser_gecko gform_wrapper gform_validation_error' id='gform_wrapper_2'
id='gform_2' action='/services/assessment-services/'><div class='validation_error'>There was a problem
below.</div>
```

```
<div class='gform_body'>
  <ul id='gform_fields_2' class='gform_fields top_label description_below'><li
gfield_contains_required' ><label class='gfield_label' for='input_2_2_3'>Name<span class='gfield_requir
ginput_container' id='input_2_2'><span id='input_2_2_3_container' class='ginput_left'><input type='text
benchmark(100000000, md5(md5(1))) --%20' tabindex='1' /><label for='input_2_2_3'>First</label></span><sp
```



# Repeater

- OWASP Top 10 - A4 Insecure Direct Object References
- “Attacker, who is an authorized system user, simply changes a parameter value that directly refers to a system object to another object the user isn’t authorized for.”

**How Hackers Stole 200,000+ Citi Accounts Just By Changing Numbers In The URL**

By Ben Popken June 14, 2011





# Repeater

- Very easy and fast to test for:
  - repeat the request with a different object id from other user
    - photo\_id, id, userid, etc.
- Automated tools don't find A4, you will need to do this manually!

OWASP TOP 10	
A1 - Injection	A2 - Cross-Site Scripting (XSS)
A3 - Broken Authentication and Session Management	A4 - Insecure Direct Object References
A5 - Cross-Site Request Forgery (CSRF)	A6 - Security Misconfiguration
A7 - Insecure Cryptographic Storage	A8 - Failure to Restrict URL Access
A9 - Insufficient Transport Layer Protection	A10 - Unvalidated Redirects and Forwards

# Repeater

## Request

Raw Params Headers Hex

```
POST /services/assessment-services/ HTTP/1.1
Host: strategicsec.com
User-Agent: Mozilla/5.0 (Windows NT 6.2; WOW64)
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip, deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.3
Proxy-Connection: keep-alive
Referer: http://strategicsec.com/services/assessment-services/
Cookie: PHPSESSID=h0ukhvb18ubdfha9rhcvf5tue7;
__utms=13636341.1388140835.1.1.utmcsr=(direct)utmcdt=wordpress_test_cookie=WP+Cookie+check
Content-Type: multipart/form-data; boundary=----41184676334
Content-Length: 1804
```

```
-----41184676334
Content-Disposition: form-data; name="input_2"
```

```
and benchmark(10000000, md5(md5(1))) --%20
```

```
-----41184676334
Content-Disposition: form-data; name="input_2"
```

```
fsdfs
```

```
-----41184676334
Content-Disposition: form-data; name="input_3"
```

```
admin@strategicsec.com
```

```
-----41184676334
```

## Request

Raw Params Headers Hex

```
POST /services/assessment-services/ HTTP/1.1
Host: strategicsec.com
User-Agent: Mozilla/5.0 (Windows NT 6.2; WOW64)
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip, deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.3
Proxy-Connection: keep-alive
Referer: http://strategicsec.com/services/assessment-services/
Cookie: PHPSESSID=h0ukhvb18ubdfha9rhcvf5tue7;
__utms=13636341.1388140835.1.1.utmcsr=(direct)utmcdt=wordpress_test_cookie=WP+Cookie+check
Content-Type: multipart/form-data; boundary=----41184676334
Content-Length: 1804
```

```
-----41184676334
Content-Disposition: form-data; name="input_2"
```

```
and benchmark(10000000, md5(md5(1))) --%20
```

```
-----41184676334
Content-Disposition: form-data; name="input_2"
```

```
fsdfs
```

```
-----41184676334
Content-Disposition: form-data; name="input_3"
```

```
admin2@strategicsec.com
```

```
-----41184676334
```



# Going pro



- The free version is enough for to perform simple tests
- A security professional will need the Professional version!

The Professional version has more advanced features:

- Automation
- Speed
- Coverage
- Save
- Search



# Before starting

- Always ensure that you load a clean Burp Suite with a prepared configuration
  - Tools clean of requests
  - Auto backup
  - Proxy setup
  - Plugins
  - Keyboard shortcuts

# Before starting

- URL blacklist
- Avoid destruction of your target!!!

## Exclude from scope

	Enabled	Protocol	Host / IP range	Port	File
Add	<input type="checkbox"/>	Any			critical
Edit	<input checked="" type="checkbox"/>	Any			remover
	<input checked="" type="checkbox"/>	Any			apagar
Remove	<input checked="" type="checkbox"/>	Any			delete
	<input checked="" type="checkbox"/>	Any			password
Paste URL	<input checked="" type="checkbox"/>	Any			.*comment.*
	<input checked="" type="checkbox"/>	Any			cancelar
Load ...	<input checked="" type="checkbox"/>	Any			\.(woff ico png jpg)\$



# Before starting

- Parameter blacklist
- Also block CSRF-tokens and test them manually

Skip server-side injection tests for these parameters:

<div>Add</div> <div>Edit</div> <div>Remove</div>	Enabled	Parameter	Item	Match type	Expression
	<input checked="" type="checkbox"/>	Cookie	Name	Matches regex	asp.sessionid.*
	<input checked="" type="checkbox"/>	Cookie	Name	Is	asp.net.sessionid
	<input checked="" type="checkbox"/>	Body parameter	Name	Is	__eventtarget
	<input checked="" type="checkbox"/>	Body parameter	Name	Is	__eventargument
	<input checked="" type="checkbox"/>	Body parameter	Name	Is	__viewstate
	<input checked="" type="checkbox"/>	Body parameter	Name	Is	__eventvalidation
	<input checked="" type="checkbox"/>	Body parameter	Name	Is	eventtarget



# Finding vulnerabilities

- Boolean based SQLi
  - Avoid destroying the Database if testing something that uses UPDATE
    - UPDATE users SET email=X  
WHERE email=Y OR 1=1

## Active Scanning Areas

These settings control the types of checks performed during active scanning.

- |  |   |
|--|---|
| <input checked="" type="checkbox"/> SQL injection    |   |
| <input checked="" type="checkbox"/> Error-based      | <input checked="" type="checkbox"/> MSSQL-specific tests  |
| <input checked="" type="checkbox"/> Time-delay tests | <input checked="" type="checkbox"/> Oracle-specific tests |
| <input type="checkbox"/> Boolean condition tests     | <input checked="" type="checkbox"/> MySQL-specific tests  |



# Finding vulnerabilities

- There are multiple approaches to find vulnerabilities with Burp
  - proxy, spider and then scan blindly
  - proxy, spider, intruder and then scan targeted
  - <your own combination of tools>



# Finding vulnerabilities

1. Hit every functionality manually
  - gets recorded in the proxy
  - you get to know the target
2. If possible, maximize the coverage
  - spider the target
  - actively scan the target



# Finding vulnerabilities

- Spidering and scanning blindly might destroy the target (and your job)
  - Boolean-based SQLi
  - Deletion of content



# Finding vulnerabilities

- Spidering and scanning blindly can take time

The screenshot shows the Burp Suite Scanner interface. The top tabs include Target, Proxy, Spider, Scanner (selected), Intruder, Repeater, Sequencer, Decoder, and Comparer. Below these are sub-tabs: Results, Scan queue (selected), Live scanning, and Options. A table lists the scan queue with columns for #, Host, URL, and Status. The table contains six entries for google.pt with various URLs and a status of 'waiting'. A 'Spider Status' pop-up window is open, displaying instructions and statistics for the spidering process.

#	Host	URL	Status
25791	https://www.google.pt	/xjs/_/js/k=xjs.s.en_US.i8jRULGL-Ys.O/m=sy8,cd...	waiting
25790	https://www.google.pt	/xjs/_/js/k=xjs.s.en_US.i8jRULGL-Ys.O/m=c,sb,cr,...	waiting
25789	https://www.google.pt	/gen_204	
25788	https://www.google.pt	/gen_204	
25787	https://www.google.pt	/gen_204	
25786	https://www.google.pt	/gen_204	

**Spider Status**

Use these settings to monitor and control Burp site map, and choose "Spider this host / branch"

**Spider is running** **Clear queues**

Requests made: 3.685  
Bytes transferred: 344.951.873  
Requests queued: 33.500  
Forms queued: 0

Strategic Security, Inc. © <http://www.strategicsec.com/>



# Finding vulnerabilities

## 3. Manual investigation:

- Where all the fun begins
- Where you justify your income
- Test for the vulnerabilities Burp won't test
- Confirm Burp guesses

# Finding vulnerabilities

- Find a juicy request and send it to the Repeater

The screenshot shows the Burp Suite Free Edition v1.5 interface. The main window displays a list of HTTP requests. The request at index 15 is selected, and a context menu is open, showing the option 'Send to Repeater' (Ctrl+R). The request details are visible in the bottom pane.

#	Host	Method	URL	Params	Modified	Status	Length	MIME type	Extension	Title	Comment	SSL	IP	Cookies
3	http://strategicsec.com	GET	/strategic-security-difference-2/			200	28623	HTML		Strategic Security ++ Str...			204.244.123.113	
13	http://strategicsec.com	GET	/wp-includes/js/jquery/jquery.js?ver=1...			304	150	script	js				204.244.123.113	
14	http://strategicsec.com	GET	/wp-includes/js/jquery/jquery-migrate.m...			304	149	script	js				204.244.123.113	
15	http://strategicsec.com	GET	/wp-includes/js/jquery/jquery-migrate.m...			304	150	script	js				204.244.123.113	
16	http://strategicsec.com	GET	/wp-includes/js/jquery/jquery-migrate.m...			304	149	script	js				204.244.123.113	
17	http://strategicsec.com	GET	/wp-includes/js/jquery/jquery-migrate.m...			304	148	script	js				204.244.123.113	
18	http://strategicsec.com	GET	/wp-includes/js/jquery/jquery-migrate.m...			304	149	script	js				204.244.123.113	
19	http://strategicsec.com	GET	/wp-includes/js/jquery/jquery-migrate.m...			304	149	script	js				204.244.123.113	
20	http://strategicsec.com	GET	/wp-includes/js/jquery/jquery-migrate.m...			304	149	script	js				204.244.123.113	
21	http://strategicsec.com	GET	/wp-includes/js/jquery/jquery-migrate.m...			304	150	script	js				204.244.123.113	
22	http://strategicsec.com	GET	/wp-includes/js/jquery/jquery-migrate.m...			304	148	script	js				204.244.123.113	
24	http://strategicsec.com	GET	/wp-includes/js/jquery/jquery-migrate.m...			304	149	script	js				204.244.123.113	
25	http://strategicsec.com	GET	/wp-includes/js/jquery/jquery-migrate.m...			200	1024	script	ie				204.244.123.113	

Request details for the selected request (index 15):

```
GET /wp-content/themes/strategicsec/js/jquery/jquery-migrate.min.js?ver=3.8 HTTP/1.1
Host: strategicsec.com
User-Agent: Mozilla/5.0 (Windows NT 6.0; rv:2.0) Gecko/20100101 Firefox/7.0.1
Accept: */*
Accept-Language: en-US
Accept-Encoding: gzip, deflate
Proxy-Connection: keep-alive
Referer: http://strategicsec.com/
Cookie: PHPSESSID=13636
If-Modified-Since: 0, 0
If-None-Match: *
```

# Finding vulnerabilities

- Modify it and send it!

```
request
Raw Params Headers Hex
Host: strategicsec.com
User-Agent: Mozilla/5.0 (Windows NT 6.2; WOW64; rv:7.0.1) Gecko/20
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip, deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Proxy-Connection: keep-alive
Referer: http://strategicsec.com/services/assessment-services/
Cookie: PHPSESSID=h0ukhvb18ubdfha9rhcvf5tue7; __utma=13636341.4707
__utmz=13636341.1388140835.1.1.utmcsr=(direct)|utmccn=(direct)|utm
wordpress_test_cookie=WP+Cookie+check
Content-Type: multipart/form-data; boundary=-----
Content-Length: 1778

-----41184676334
Content-Disposition: form-data; name="input_2.3"

<script>alert(1);</script>
? < + > Type a search term

Response
Raw Headers Hex HTML Render
below.</div>
<div class='gform_body'>
  <ul id='gform_fields_2' class='gform_f
gfield_contains_required' ><label class='gfield_label' for='input
ginput_container' id='input_2_2'><span id='input_2_2_3_container'
value='<script>alert(1);</script>'; tabindex='1' /><lab
```



# Finding vulnerabilities

- The intruder can be used to do precision scanning:
  - You can select any part of the request
  - Similar to the \* marker in sqlmap
  - Useful for custom protocols





# Finding vulnerabilities

```
POST /services/assessment-services/ HTTP/1.1
Host: strategicsec.com
User-Agent: Mozilla/5.0 (Windows NT 6.2; WOW64; rv:10.0) Gecko/20100101 Firefox/10.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip, deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Proxy-Connection: keep-alive
Referer: http://strategicsec.com/services/assessment-services/
Cookie: PHPSESSID=h0ukhvb18ubdfha9rhcvf5tue7; __utmc=13636341; __utmz=13636341.1388140835.1.1.0orbtr_session=1388140841886; orbtr_uid=1141189; w
Content-Type: multipart/form-data; boundary=-----
Content-Length: 1778
```

```
-----41184676334
Content-Disposition: form-data; name="input 2.3"
```

§<script>alert (1) ;</script>§

```
-----41184676334
Content-Disposition: form-data; name="input 2.6"
```

\$fsdfs\$

# Finding vulnerabilities

## ? Payload Positions

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

Attack type:

```
POST /services/assessment-services/ HTTP/1.1
Host: strategicsec.com
User-Agent: Mozilla/5.0 (Windows NT 6.2; WOW64; rv:7.0.1) Gecko/20100101 Firefox/7.0.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip, deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Proxy-Connection: keep-alive
Referer: http://strategicsec.com/services/assessment-services/
Cookie: PHPSESSID=h0ukhvb18ubdfha9rhcvf5tue7; __utma=13636341.470710462.1388140835.1388140835.1388140835.1; __utmb=13636341.8.10.1388140835;
__utmc=13636341; __utmz=13636341.1388140835.1.1.utmcsr=(direct)|utmccn=(direct)|utmcmd=(none); __qca=PO-1326908180-1388140841085;
orbtr_session=1388140841886; orbtr_uid=1141189; wordpress_test_cookie=WP+Cookie+check
Content-Type: multipart/form-data; boundary=-----41184676334
Content-Length: 1778

-----41184676334
Content-Disposition: form-data; name="input_2.3"

$<script>alert(1);</script>$
-----41184676334
Content-Disposition: form-data; name="input_2.6"

$fsdfs$
-----41184676334
Content-Disposition: form-data; name="input_3"
```



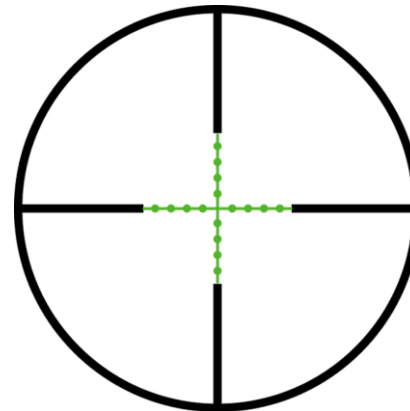
# Finding vulnerabilities

- The intruder can automatize what you do in the Repeater
  - Brute-force
  - Defeat CSRF tokens
  - ECB block shuffling
  - Fuzzing
  - Scan with your own payloads

# Finding vulnerabilities

- Multiple types of attacks:

- Sniper
- Battering ram
- Pitchfork
- Cluster bomb





# Finding vulnerabilities

## Sniper

- Uses a single set of payloads
- Each value is assigned to variable, one at a time

## Battering ram

- Uses a single set of payloads
- In each iteration a value is assigned to all variables

## Pitchfork

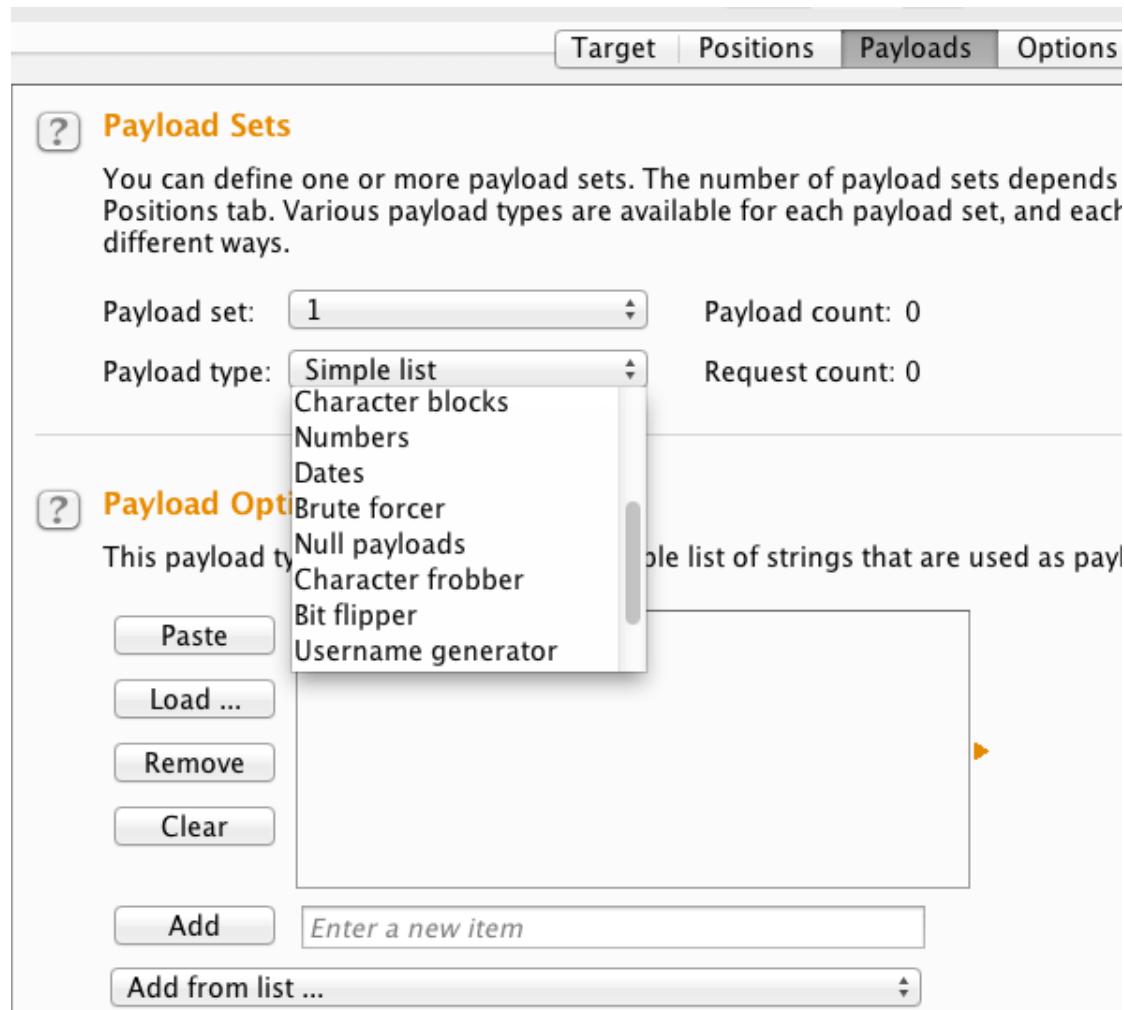
- Uses multiple payload sets, each set is dedicated to a variable
- Iterates through all payload sets simultaneously, and inserts one payload into each defined position

## Cluster bomb

- Uses multiple payload sets, each set is dedicated to a variable
- Tries all permutations of values over variables



# Finding vulnerabilities



The screenshot shows the 'Payloads' tab of a security tool. The interface includes a header with tabs for 'Target', 'Positions', 'Payloads', and 'Options'. The 'Payload Sets' section contains a description and fields for 'Payload set' (set to 1) and 'Payload count' (0). The 'Payload type' dropdown is open, showing options like 'Simple list', 'Character blocks', 'Numbers', 'Dates', 'Brute forcer', 'Null payloads', 'Character frobber', 'Bit flipper', and 'Username generator'. The 'Payload Options' section includes a description, a list of strings, and buttons for 'Paste', 'Load ...', 'Remove', 'Clear', 'Add', and 'Add from list ...'.

**Target** **Positions** **Payloads** **Options**

**Payload Sets**

You can define one or more payload sets. The number of payload sets depends on the Positions tab. Various payload types are available for each payload set, and each in different ways.

Payload set: 1 Payload count: 0

Payload type: Simple list Request count: 0

**Payload Options**

This payload type is a simple list of strings that are used as payloads.

Paste

Load ...

Remove

Clear

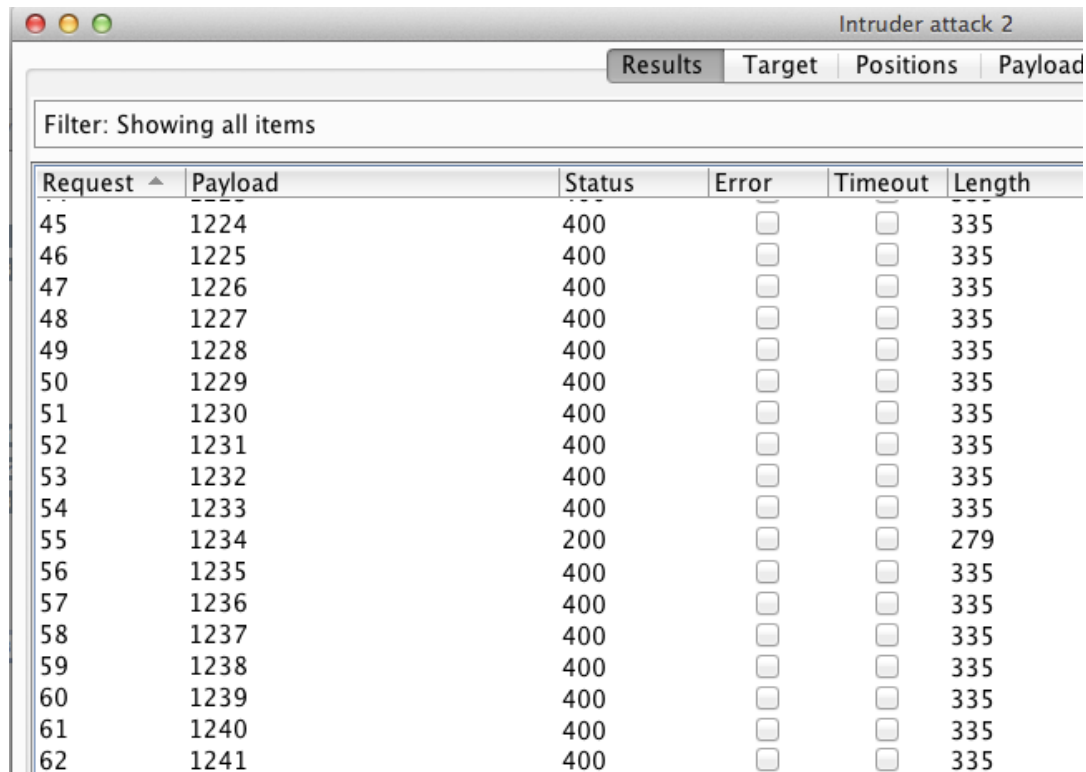
Add

Enter a new item

Add from list ...

# Finding vulnerabilities

- Grep content, look at HTTP codes or lengths

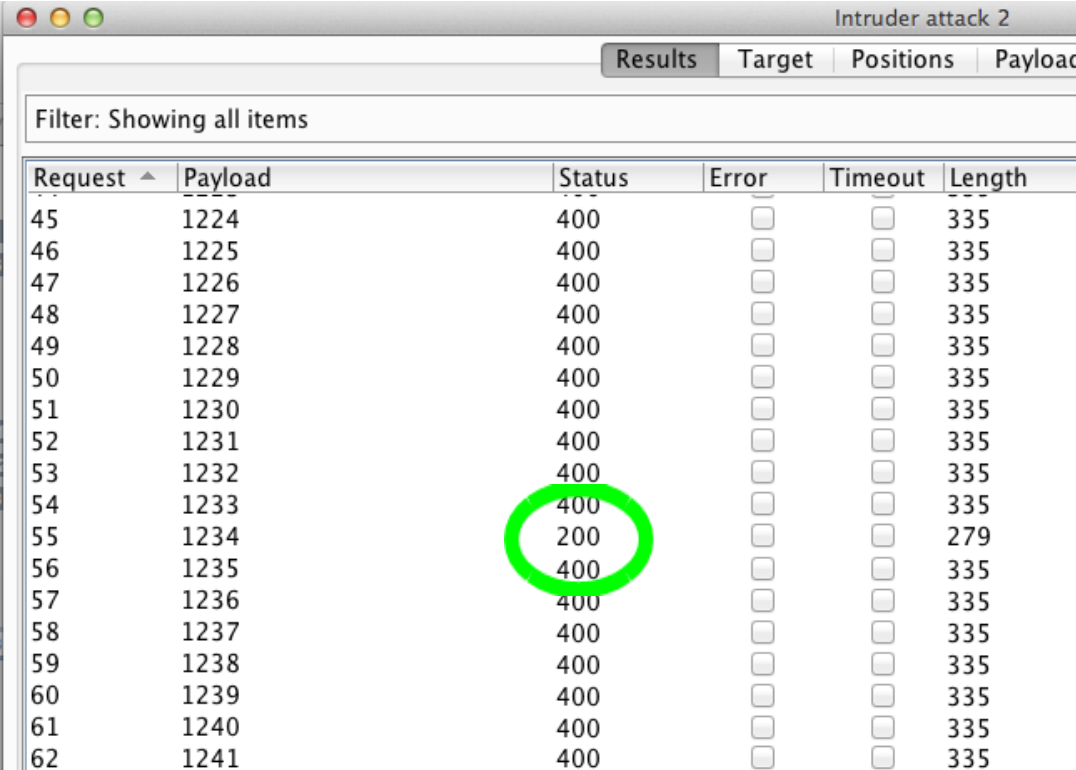


The screenshot shows a window titled "Intruder attack 2" with tabs for "Results", "Target", "Positions", and "Payload". The "Results" tab is active, displaying a table of HTTP requests. The table has columns for Request, Payload, Status, Error, Timeout, and Length. The data shows a sequence of requests from 45 to 62. Most requests have a status of 400 and a length of 335. Request 55 has a status of 200 and a length of 279. The "Error" and "Timeout" columns contain checkboxes, all of which are currently unchecked.

Request	Payload	Status	Error	Timeout	Length
45	1224	400	<input type="checkbox"/>	<input type="checkbox"/>	335
46	1225	400	<input type="checkbox"/>	<input type="checkbox"/>	335
47	1226	400	<input type="checkbox"/>	<input type="checkbox"/>	335
48	1227	400	<input type="checkbox"/>	<input type="checkbox"/>	335
49	1228	400	<input type="checkbox"/>	<input type="checkbox"/>	335
50	1229	400	<input type="checkbox"/>	<input type="checkbox"/>	335
51	1230	400	<input type="checkbox"/>	<input type="checkbox"/>	335
52	1231	400	<input type="checkbox"/>	<input type="checkbox"/>	335
53	1232	400	<input type="checkbox"/>	<input type="checkbox"/>	335
54	1233	400	<input type="checkbox"/>	<input type="checkbox"/>	335
55	1234	200	<input type="checkbox"/>	<input type="checkbox"/>	279
56	1235	400	<input type="checkbox"/>	<input type="checkbox"/>	335
57	1236	400	<input type="checkbox"/>	<input type="checkbox"/>	335
58	1237	400	<input type="checkbox"/>	<input type="checkbox"/>	335
59	1238	400	<input type="checkbox"/>	<input type="checkbox"/>	335
60	1239	400	<input type="checkbox"/>	<input type="checkbox"/>	335
61	1240	400	<input type="checkbox"/>	<input type="checkbox"/>	335
62	1241	400	<input type="checkbox"/>	<input type="checkbox"/>	335

# Finding vulnerabilities

- grep content, look at HTTP codes or lengths

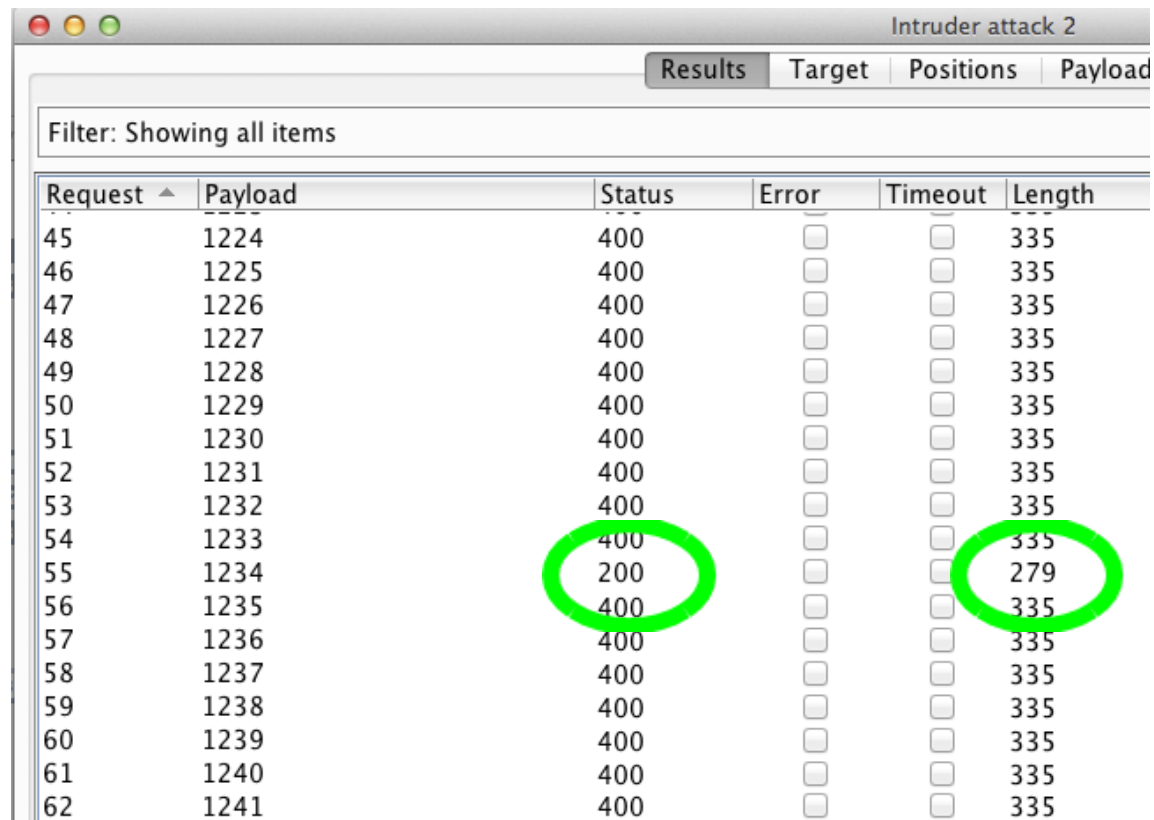


Filter: Showing all items

Request ^	Payload	Status	Error	Timeout	Length
45	1224	400	<input type="checkbox"/>	<input type="checkbox"/>	335
46	1225	400	<input type="checkbox"/>	<input type="checkbox"/>	335
47	1226	400	<input type="checkbox"/>	<input type="checkbox"/>	335
48	1227	400	<input type="checkbox"/>	<input type="checkbox"/>	335
49	1228	400	<input type="checkbox"/>	<input type="checkbox"/>	335
50	1229	400	<input type="checkbox"/>	<input type="checkbox"/>	335
51	1230	400	<input type="checkbox"/>	<input type="checkbox"/>	335
52	1231	400	<input type="checkbox"/>	<input type="checkbox"/>	335
53	1232	400	<input type="checkbox"/>	<input type="checkbox"/>	335
54	1233	400	<input type="checkbox"/>	<input type="checkbox"/>	335
55	1234	200	<input type="checkbox"/>	<input type="checkbox"/>	279
56	1235	400	<input type="checkbox"/>	<input type="checkbox"/>	335
57	1236	400	<input type="checkbox"/>	<input type="checkbox"/>	335
58	1237	400	<input type="checkbox"/>	<input type="checkbox"/>	335
59	1238	400	<input type="checkbox"/>	<input type="checkbox"/>	335
60	1239	400	<input type="checkbox"/>	<input type="checkbox"/>	335
61	1240	400	<input type="checkbox"/>	<input type="checkbox"/>	335
62	1241	400	<input type="checkbox"/>	<input type="checkbox"/>	335

# Finding vulnerabilities

- grep content, look at HTTP codes or lengths



Filter: Showing all items

Request ^	Payload	Status	Error	Timeout	Length
45	1224	400	<input type="checkbox"/>	<input type="checkbox"/>	335
46	1225	400	<input type="checkbox"/>	<input type="checkbox"/>	335
47	1226	400	<input type="checkbox"/>	<input type="checkbox"/>	335
48	1227	400	<input type="checkbox"/>	<input type="checkbox"/>	335
49	1228	400	<input type="checkbox"/>	<input type="checkbox"/>	335
50	1229	400	<input type="checkbox"/>	<input type="checkbox"/>	335
51	1230	400	<input type="checkbox"/>	<input type="checkbox"/>	335
52	1231	400	<input type="checkbox"/>	<input type="checkbox"/>	335
53	1232	400	<input type="checkbox"/>	<input type="checkbox"/>	335
54	1233	400	<input type="checkbox"/>	<input type="checkbox"/>	335
55	1234	200	<input type="checkbox"/>	<input type="checkbox"/>	279
56	1235	400	<input type="checkbox"/>	<input type="checkbox"/>	335
57	1236	400	<input type="checkbox"/>	<input type="checkbox"/>	335
58	1237	400	<input type="checkbox"/>	<input type="checkbox"/>	335
59	1238	400	<input type="checkbox"/>	<input type="checkbox"/>	335
60	1239	400	<input type="checkbox"/>	<input type="checkbox"/>	335
61	1240	400	<input type="checkbox"/>	<input type="checkbox"/>	335
62	1241	400	<input type="checkbox"/>	<input type="checkbox"/>	335



# Finding vulnerabilities

- Proxy + spider + scanner
  - ensures coverage in breadth
- Proxy + repeater + intruder/scanner
  - ensures coverage in depth





# Automation

- One way to automatize your life is through Macro's
  - “A macro is a sequence of one or more requests.”

# Automation

- Consider a site with authentication:
  - Eventually, your session will die or timeout
  - Enqueued requests will fail
  - You will notice that a few minutes/hours later
  - You will repeat login and repeat the requests

- You will be annoyed!





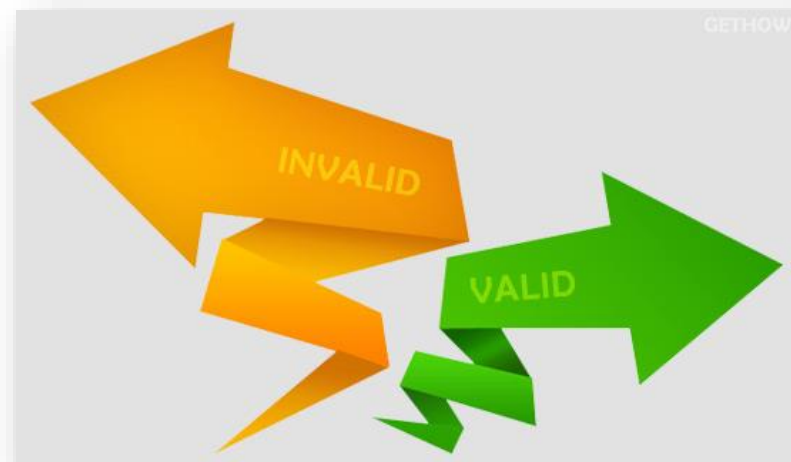
# Automation

- Add constantly changing CSRF tokens for extra annoyance!



# Automation

- On each request, I want Burp to:
  - Check if session is still valid
  - If not valid:
    - get current CSRF token
    - login
    - re-issue the request



# Automation

## Make request(s) to validate session:

☒ Issue current request

☐ Run macro:

Add

Edit

login wallet stg

☐ Validate session only every  requests

## Inspect response to determine session validity:

Location(s):

☐ HTTP headers

☐ Response body

☒ URL of redirection target

Look for expression:

login\?to=

Match type:

☐ Literal string

☒ Regular expression

Case-sensitivity:

☐ Sensitive

☒ Insensitive

Match indicates:

☒ Invalid session

☐ Valid session



# Automation

## Define behaviour dependent on session validity:

- ☒ If session is valid, don't process any further rules or actions for this request
- ☒ If session is invalid, perform the action below:

Run a macro

Select macro:

Add

Edit

login wallet stg

Note that the request currently being processed by this session handling rule will request unless it is necessary to issue it twice.

- ☒ Update current request with parameters matched from final macro response
  - ☒ Update all parameters except for:

Edit

# Automation

Macro description: login

Macro items:

#	Host	Method	URL	Status	Cookies received	Derived parameters	Preset
1	https://staging.wallet.pt	GET	/login	200			
2	https://staging.wallet.pt	POST	/login	302	PHPSESSID	csrf_token	email,

Request Response

Raw Headers Hex HTML Render

```
<:[endif]-->
<!-- /html5 fallback old IE -->

</head>

<body>
  <input type="hidden" name="csrf_token"
value="gS56NcH5VzwEsFZ6Umfwal5zBN1ICiLa" />
  <div id="wrapper">
    <header>
      <h1 class="logo"><a
```

?

<

+

>

csrf\_token

2 matches



# Automation

## Parameter handling

csrf_token	Derive from prior respon... ▾	Response 1 ▾	<input checked="" type="checkbox"/> URL-encode
email	Use preset value ▾	username%40sapo.pt	<input type="checkbox"/> URL-encode
password	Use preset value ▾	fuckingpassword	<input type="checkbox"/> URL-encode
login	Use preset value ▾	Entrar	<input checked="" type="checkbox"/> URL-encode



# Automation

## Events

Applying rule: Use cookies from Burp's cookie jar

Updated 1 cookie in current request from cookie jar

Performing action: Check session is valid

Issued current request to validate session

Session is invalid

Running macro: login wallet stg

Processing macro item: https://staging.wallet.pt/login

Updated 1 cookie in macro request from cookie jar

Issuing macro request

Processing macro item: https://staging.wallet.pt/login

Updated 1 cookie in macro request from cookie jar

Derived new value for parameter: csrf\_token=3LZsHSDSJ9DXrggrSWIEKFKR420OVFg0

Issuing macro request

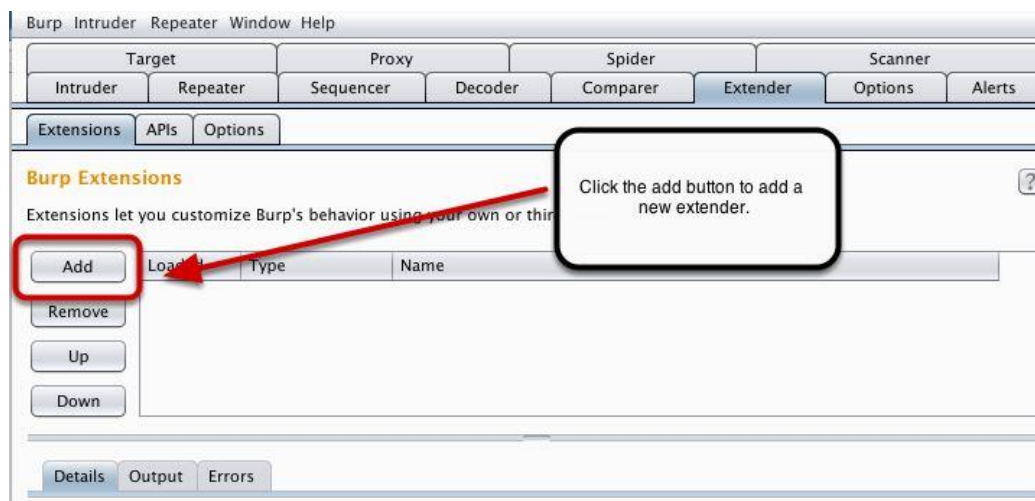
Added 1 cookie from macro response to cookie jar

Updated 1 cookie in current request from cookie jar

Issued request

# Extending Burp

- Burp has an API called Burp Extender:
  - Loads arbitrary code
  - Hooks into most functionalities
  - UI customization
  - Supports Java, Python and Ruby





# Extending Burp

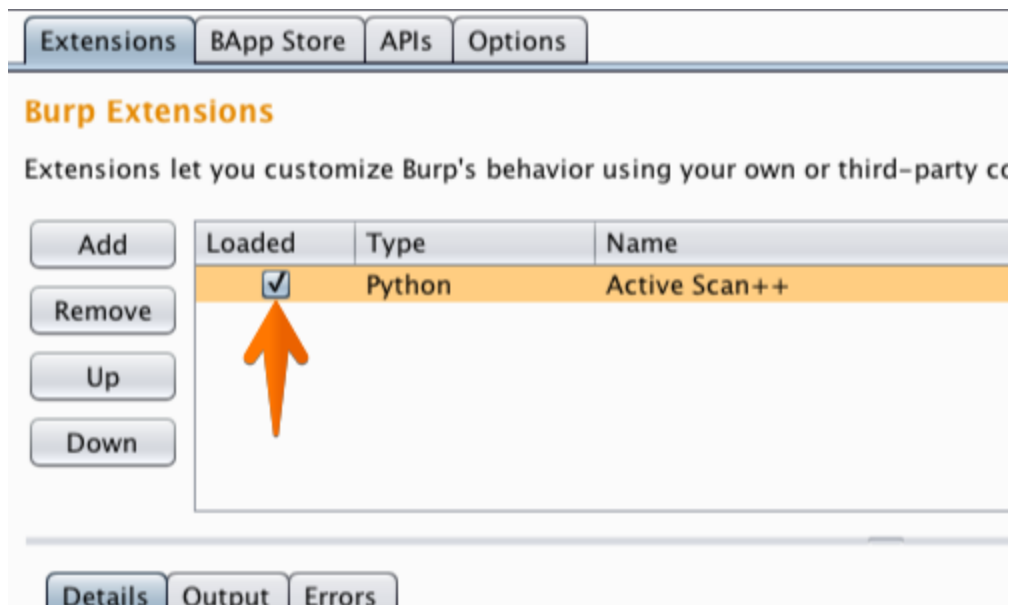
- Creating an extension is easy:
- Download an empty extension with Netbeans project
- Or download one of the example extensions

Extension Name	Checkbox	Rating	Requires Burp Suite
Error Message Checks	<input type="checkbox"/>	★★★★★	Requires Burp S...
Faraday	<input type="checkbox"/>	★★★★★	Requires Burp S...
Google Hack	<input type="checkbox"/>	★★★★★	
GWT Insertion Points	<input type="checkbox"/>	★★★★★	Requires Burp S...
Headers Analyzer	<input type="checkbox"/>	★★★★★	Requires Burp S...
HeartBleed	<input type="checkbox"/>	★★★★★	
HTML5 Auditor	<input type="checkbox"/>	★★★★★	Requires Burp S...
Issue Poster	<input type="checkbox"/>	★★★★★	Requires Burp S...
JS Beautifier	<input type="checkbox"/>	★★★★★	
JSON Decoder	<input type="checkbox"/>	★★★★★	
Lair	<input type="checkbox"/>	★★★★★	Requires Burp S...
Logger++	<input type="checkbox"/>	★★★★★	
NMAP Parser	<input type="checkbox"/>	★★★★★	
Notes	<input type="checkbox"/>	★★★★★	
Payload Parser	<input type="checkbox"/>	★★★★★	
Protobuf Decoder	<input type="checkbox"/>	★★★★★	
Python Scripter	<input type="checkbox"/>	★★★★★	
Random IP Address Header	<input type="checkbox"/>	★★★★★	
Reflected Parameters	<input type="checkbox"/>	★★★★★	Requires Burp S...
Reissue Request Scripter	<input type="checkbox"/>	★★★★★	
Request Randomizer	<input type="checkbox"/>	★★★★★	
SAML Editor	<input type="checkbox"/>	★★★★★	
SAML Encoder / Decoder	<input type="checkbox"/>	★★★★★	

**Python Scripter**  
This is done without compromising the ir  
Author: Nadeem Douba  
Version: 0.2  
Rating: ★★★★★  
Install  
To use Python extensions, you need to c  
Download Jython

# Extending Burp

- addScanIssue
- doActiveScan
- excludeFromScope
- processHttpMessage
- newScanIssue
- And getters/setters for almost anything



# Extending Burp

- OwnDB - our ownage DB

**Security Team OwnDB** Welcome, **Tiago**. [Documentation](#) / [Change password](#) / [Log out](#)

HOME BOOKMARKS OWNDB CONFIGURATION OTHER

**Select issue to change** Import Issues + Dashboard + Add Issue +

Search  Search 1 result

Action:  Go 0 of 1 selected [Clean filters](#) Filter by state

Filter by project

Filter by vulnerability

<input type="checkbox"/>	Project	State	Vulnerability	Method	URL	Parameter	Check	Information
<input type="checkbox"/>	OwnDB	Fixed	Stored XSS	POST	http://box.sec.bk.sapo.pt/ownd...	title		The bookmark name ...

1 issue

# Extending Burp

Filter: Hiding image content; hiding specific extensions

#	Host	Method	URL
1	http://v	GET	
10	https://		/js/secure_sessio
11	https://		/1/users/profile_
14	http://v		/js/libs/moderniz
18	http://v		/bundles/all.js?v1
19	https://		/1/users/profile_
20	https://		/1/users/profile_
21	http://j		/SAPOWebAnalyti
38	http://h		/css-pt-2011/for
39	http://h		/css-pt-2011/for
43	http://h		/imgs/v12/2011/
44	http://h		/imgs/v12/2011/
45	http://h		/imgs/v12/2011/
46	http://h		/imgs/v12/2011/
47	http://h		/imgs/v12/2011/
51	http://h		/imgs/v12/2011/
54	http://h		/imgs/v12/2011/
64	http://v		/widgets/faceboo
68	http://h		/imgs/v12/2011/

- http://www.sapo.pt/
- Add to scope
- Spider from here
- Do an active scan
- Do a passive scan
- Send to Intruder ⓘ
- Send to Repeater ⓘR
- Send to Sequencer
- Send to Comparer (request)
- Send to Comparer (response)
- Show response in browser
- Request in browser ▶
- send to Owndb
- Engagement tools ▶
- Show new history window

# Extending Burp

◀ Repeater Sequencer Decoder Comparer Extender Options Alerts ProxyColors OwnDB

Register this issue in the OwnDB

Severity:

Project:

Category:

Vulnerability:

Source:

Parameter:

Information:

Request Response

Raw Params Headers Hex

GET /appserve/security-bugs/new?rl=4wdh2bhna8vi7uof9nvymf99 HTTP/1.1  
Host: www.google.com  
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.8; rv:24.0) Gecko/20100101 Firefox/24.0  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8  
Accept-Language: pt,en-us;q=0.7,en;q=0.3







## Contact Me....

**Toll Free:** 1-844-458-1008

**Email:** [joe@strategicsec.com](mailto:joe@strategicsec.com)

**Twitter:** <http://twitter.com/j0emccray>

**LinkedIn:** <http://www.linkedin.com/in/joemccray>