Quick Recap of Module Admin Matters

- **Project 1**:
  - **Team formation**: The Discussion thread on Canvas
  - **Brief**: See its PDF file uploaded to Canvas
  - **Topic allocation**: *In the class today!*

IFS4103:
Penetration Testing Practice

Lecture 3:
Web App Pen-Testing Review

Outline

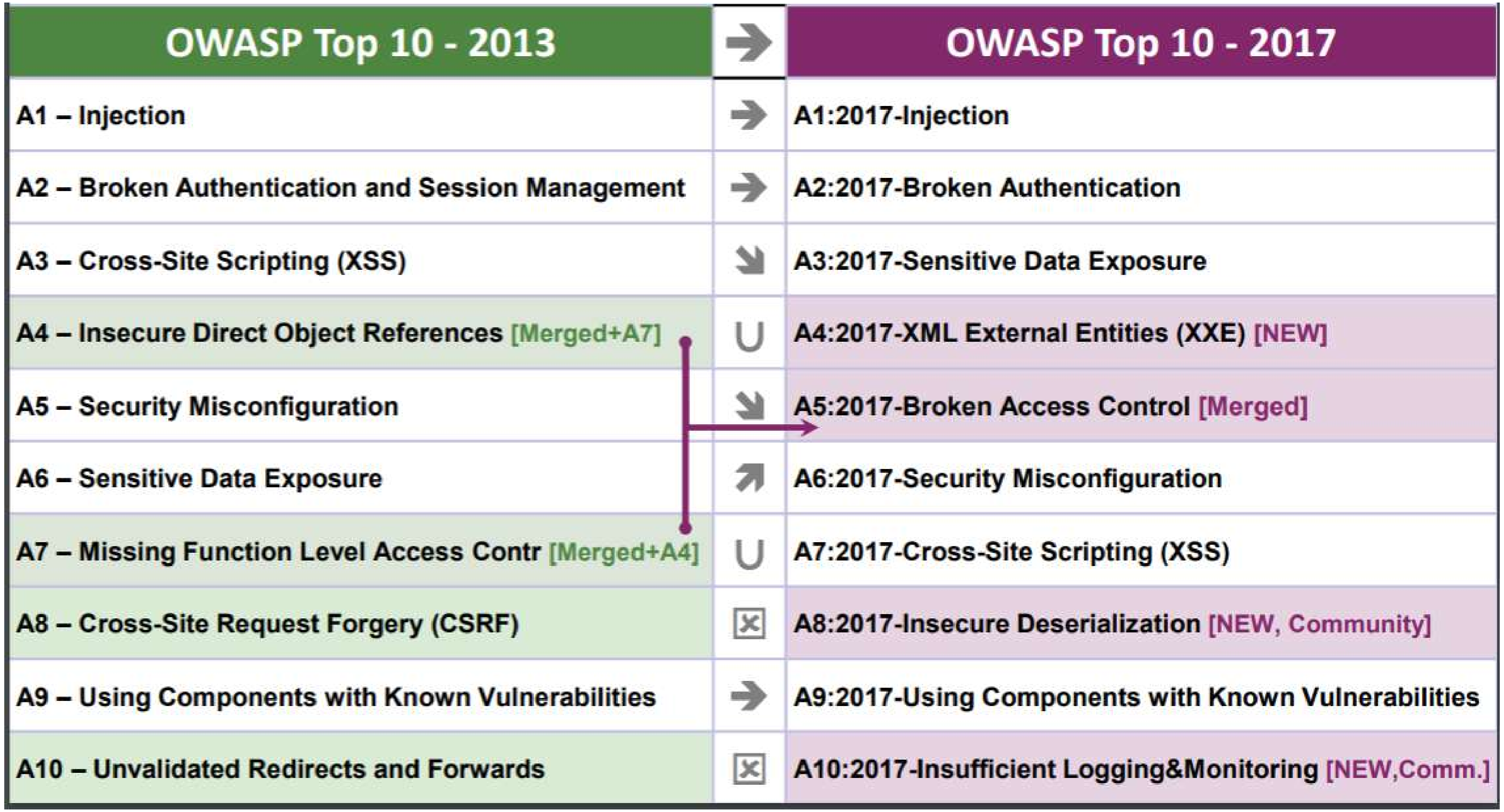
- Common web vulnerabilities: OWASP & HackerOne Top 10
- Web pen-testing & OWASP WSTG
- Web app vulnerabilities
- Secure web development & OWASP ASVS
- Useful resources

Common Web Vulnerabilities: OWASP & HackerOne Top 10

OWASP

- **OWASP**: **Open Web Application Security Project**
  - Community focused on Web Application Security
  - From Wikipedia:
    “The **Open Web Application Security Project** (**OWASP**)
    is an online community that produces freely-available articles,
    methodologies, documentation, tools, and technologies
    in the field of web application security.
    The OWASP provides free and open resources.”
  - https://owasp.org/

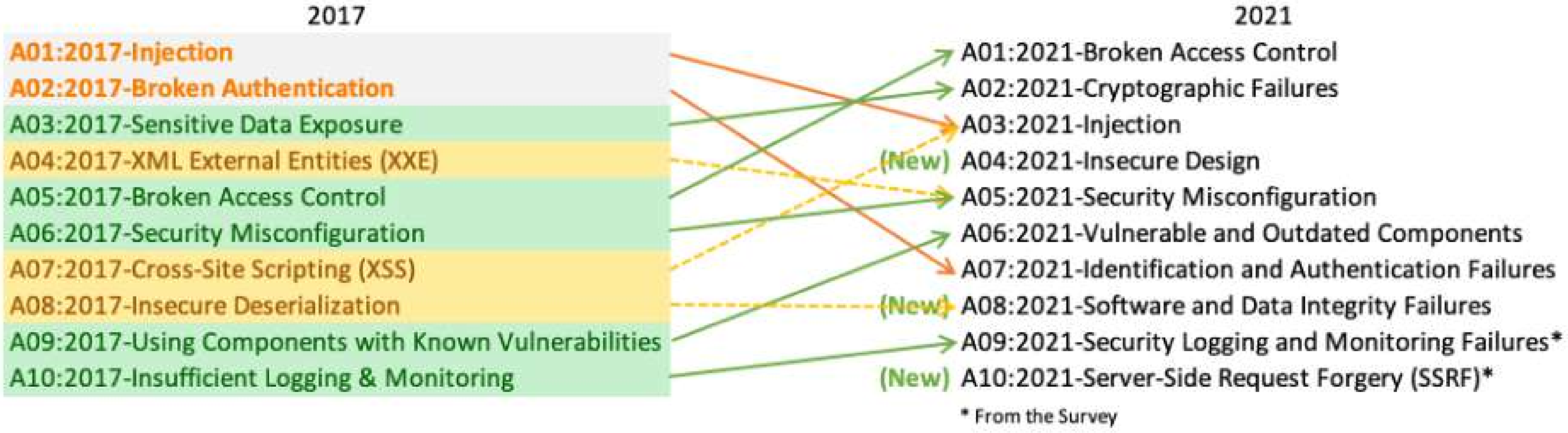
OWASP Top 10 - 2017



OWASP Top 10 - 2013	→	OWASP Top 10 - 2017
A1 – Injection	→	A1:2017-Injection
A2 – Broken Authentication and Session Management	→	A2:2017-Broken Authentication
A3 – Cross-Site Scripting (XSS)	↘	A3:2017-Sensitive Data Exposure
A4 – Insecure Direct Object References [Merged+A7]	∪	A4:2017-XML External Entities (XXE) [NEW]
A5 – Security Misconfiguration	↘	A5:2017-Broken Access Control [Merged]
A6 – Sensitive Data Exposure	↗	A6:2017-Security Misconfiguration
A7 – Missing Function Level Access Contr [Merged+A4]	∪	A7:2017-Cross-Site Scripting (XSS)
A8 – Cross-Site Request Forgery (CSRF)	☒	A8:2017-Insecure Deserialization [NEW, Community]
A9 – Using Components with Known Vulnerabilities	→	A9:2017-Using Components with Known Vulnerabilities
A10 – Unvalidated Redirects and Forwards	☒	A10:2017-Insufficient Logging&Monitoring [NEW,Comm.]

Source: OWASP Top 10 - 2017

OWASP Top 10 - 2021



2017

A01:2017-Injection

A02:2017-Broken Authentication

A03:2017-Sensitive Data Exposure

A04:2017-XML External Entities (XXE)

A05:2017-Broken Access Control

A06:2017-Security Misconfiguration

A07:2017-Cross-Site Scripting (XSS)

A08:2017-Insecure Deserialization

A09:2017-Using Components with Known Vulnerabilities

A10:2017-Insufficient Logging & Monitoring

2021

A01:2021-Broken Access Control

A02:2021-Cryptographic Failures

A03:2021-Injection

(New) A04:2021-Insecure Design

A05:2021-Security Misconfiguration

A06:2021-Vulnerable and Outdated Components

A07:2021-Identification and Authentication Failures

(New) A08:2021-Software and Data Integrity Failures

A09:2021-Security Logging and Monitoring Failures*

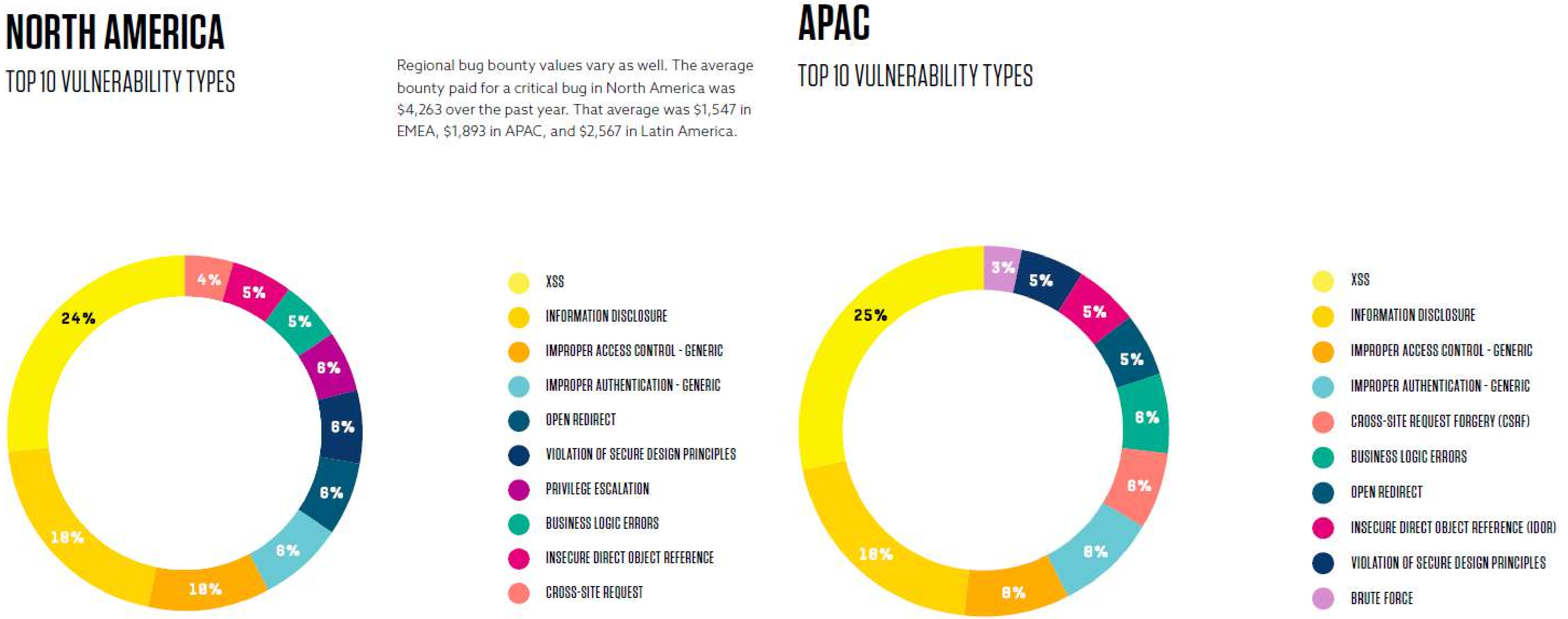
(New) A10:2021-Server-Side Request Forgery (SSRF)*

* From the Survey

Reference: https://owasp.org/www-project-top-ten/

The HackerOne Top 10 - 2020 Edition

- "***The 4th Annual Hacker Powered Security Report***" report (https://www.hackerone.com/top-ten-vulnerabilities):



NORTH AMERICA

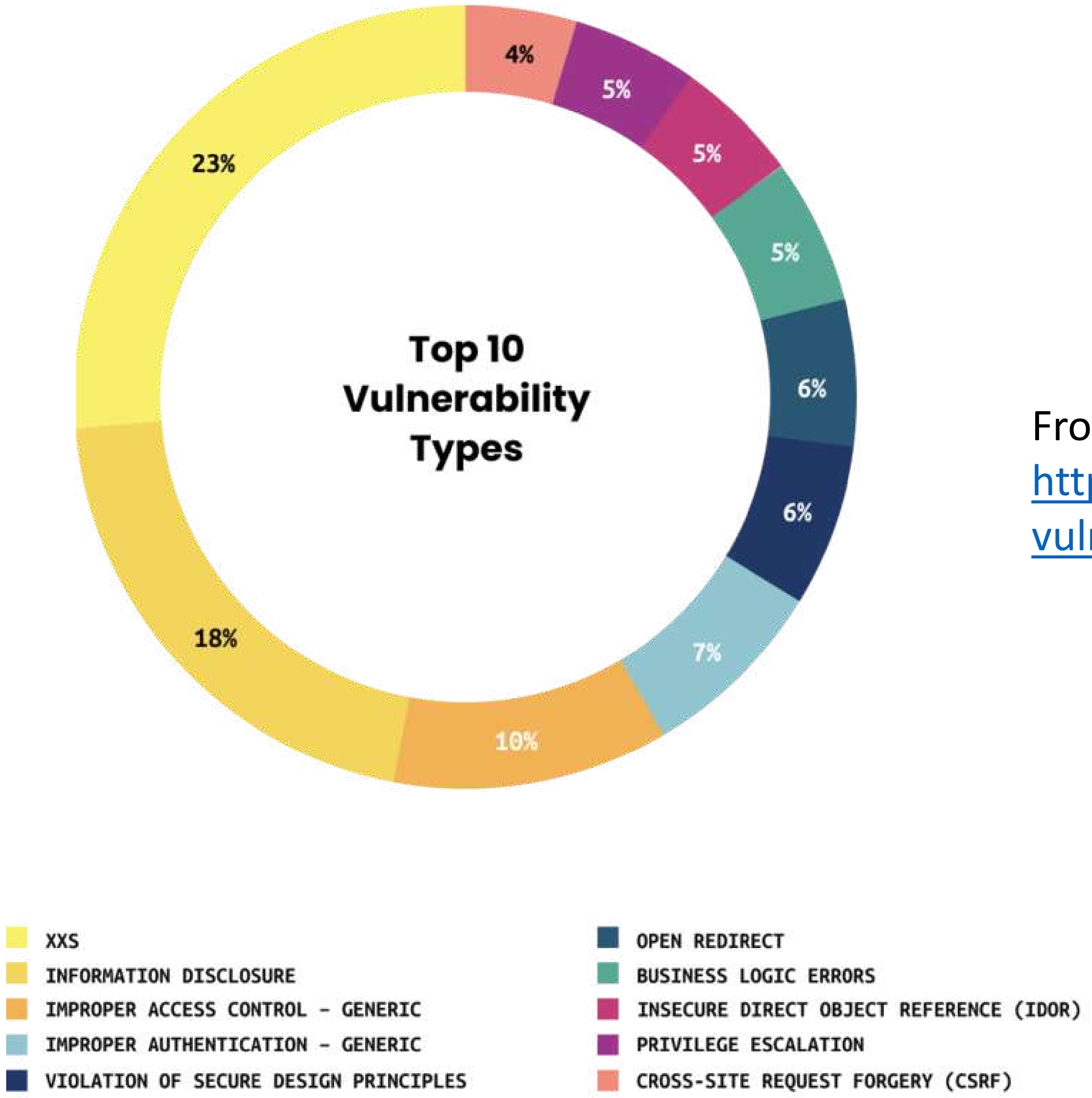
TOP 10 VULNERABILITY TYPES

Regional bug bounty values vary as well. The average bounty paid for a critical bug in North America was $4,263 over the past year. That average was $1,547 in EMEA, $1,893 in APAC, and $2,567 in Latin America.

APAC

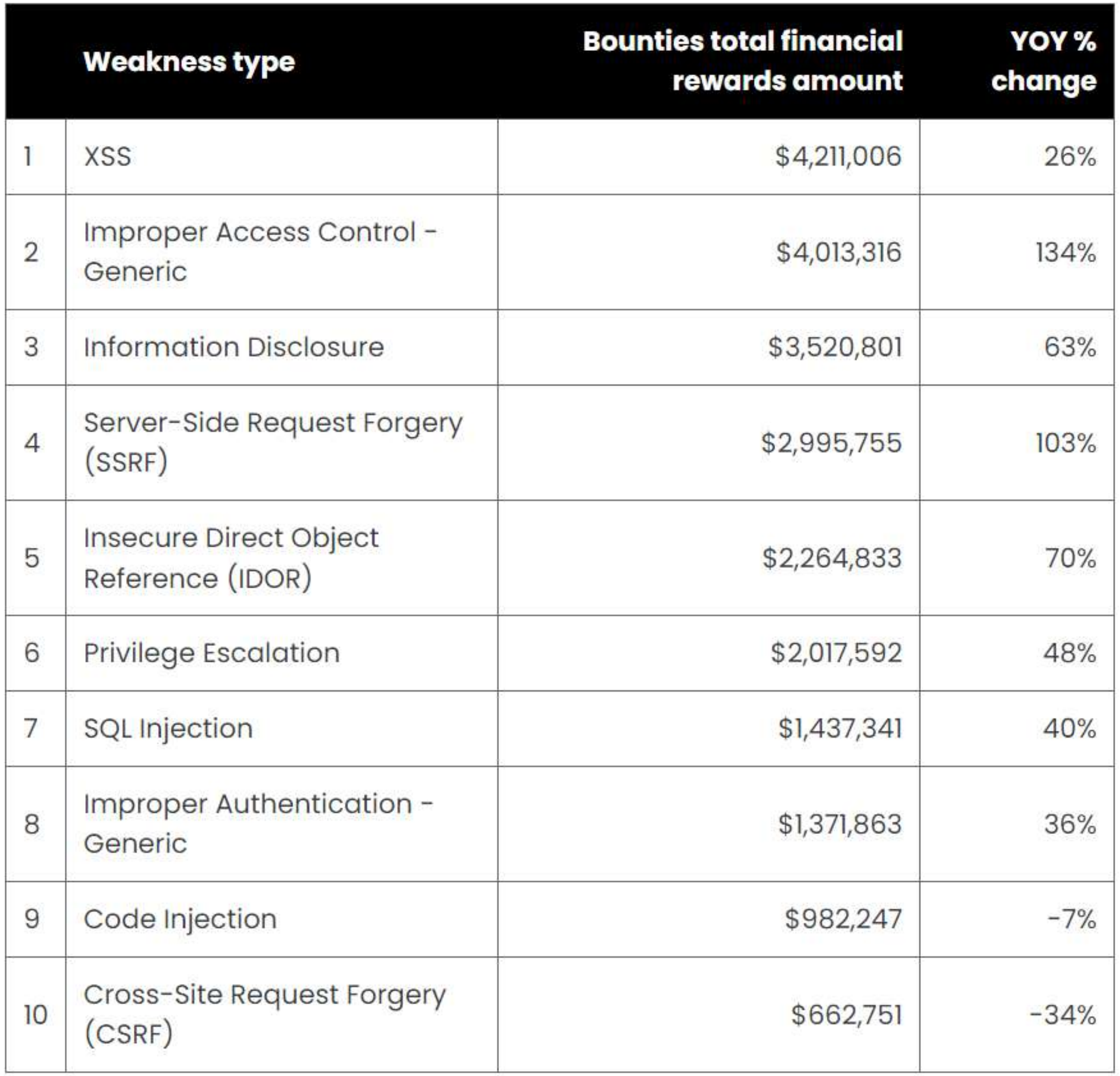
TOP 10 VULNERABILITY TYPES

The HackerOne Top 10 - 2020 Edition



From: https://www.hackerone.com/top-ten-vulnerabilities

The HackerOne Top 10 - 2020 Edition



	Weakness type	Bounties total financial rewards amount	YOY % change
1	XSS	$4,211,006	26%
2	Improper Access Control - Generic	$4,013,316	134%
3	Information Disclosure	$3,520,801	63%
4	Server-Side Request Forgery (SSRF)	$2,995,755	103%
5	Insecure Direct Object Reference (IDOR)	$2,264,833	70%
6	Privilege Escalation	$2,017,592	48%
7	SQL Injection	$1,437,341	40%
8	Improper Authentication - Generic	$1,371,863	36%
9	Code Injection	$982,247	-7%
10	Cross-Site Request Forgery (CSRF)	$662,751	-34%

From: https://www.hackerone.com/top-ten-vulnerabilities

Web Pen-Testing (WPT) & OWASP Web Security Testing Guide (WSTG)

WPT Methodologies & Guidelines

- **OWASP Web Security Testing Guide (WSTG):**
  - A web app penetration testing guide that describes **how to find certain issues**
  - OWASP Web Security Testing Guide v 4.2, https://owasp.org/www-project-web-security-testing-guide/v42/
- **Web Application Hacker's Methodology**:
  - Chapter 21 of Stuttard & Pinto, "*The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws*", 2nd ed, 2011
  - Freely accessible list
- **The Burp Methodology**:
  - It is tool specific, but is useful if you use Burp Suite
  - https://portswigger.net/support/burp-testing-methodologies

WPT Methodologies and Guidelines

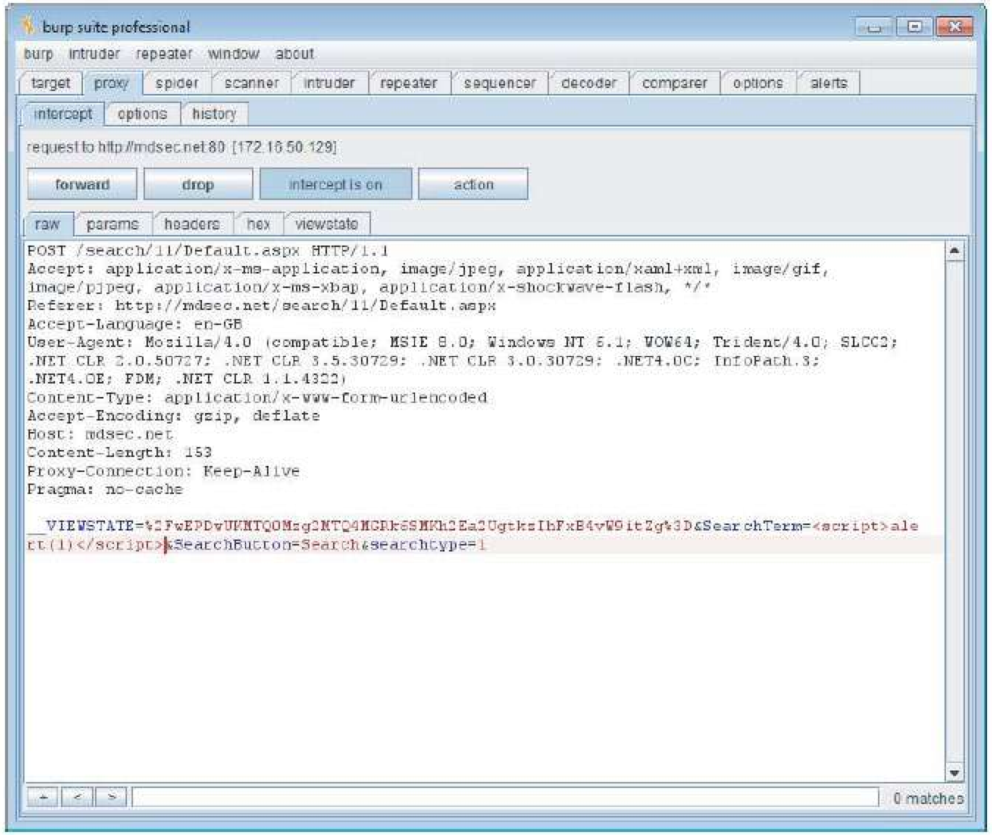
- How about **OWASP** **Application Security Verification Standard (ASVS)**?
  - It provides web developers with a list of requirements for **secure development**
  - *How is it useful for a pen-testing?*
- You can use and refer to OWASP ASVS when suggesting **remediation steps**
- It nicely matches OWASP Testing Guide
- OWASP ASVS Version - 4.0.3 is available

Useful Web Pen-Testing Tools

- **Client (web browser) side tools**:
  - Browser
  - Browser’s developer tools
  - Browser extensions:
    - IE: HttpWatch, IEWatch, …
    - Firefox: HttpWatch, FoxyProxy, LiveHTTPHeaders, PrefBar, Wappalyzer, …
    - Chrome: XSS Rays, Cookie editor, Wappalyzer, …
- Integrated **suites**:
  - Burp Suite, Zed Attack Proxy (ZAP), WebScarab, Paros, Andiparos, Fiddler, Charles, …

Web Pen-Testing Tools: Common Features

- **Proxy**: intercept & manipulate requests



burp suite professional
burp intruder repeater window about
target proxy spider scanner intruder repeater sequencer decoder comparer options alerts
intercept options history
request to http://mdsec.net:80 [172.16.50.129]
forward
drop
intercept is on
action
raw params headers hex viewstate
0 matches

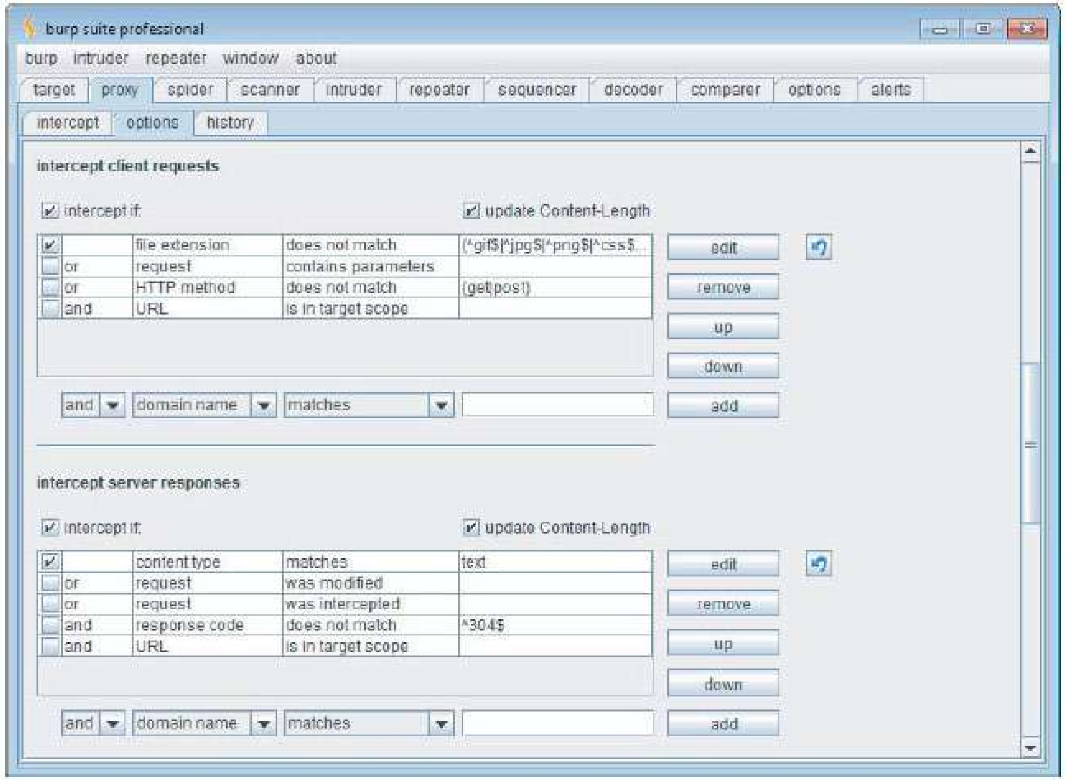
```
POST /search/11/Default.aspx HTTP/1.1
Accept: application/x-ms-application, image/jpeg, application/xaml+xml, image/gif,
image/pjpeg, application/x-ms-xbap, application/x-shockwave-flash, */*
Referer: http://mdsec.net/search/11/Default.aspx
Accept-Language: en-GB
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; Trident/4.0; SLCC2;
.NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; .NET4.0C; InfoPath.3;
.NET4.0E; FDM; .NET CLR 1.1.4322)
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate
Host: mdsec.net
Content-Length: 153
Proxy-Connection: Keep-Alive
Pragma: no-cache

__VIEWSTATE=%2FwEPDwUKMTQOMzg2NTQ4MGRk6SMKh2Ea2UgtkzIhFxB4vW9itZg%3D&SearchTerm=<script>ale
rt(1)</script>&SearchButton=Search&searchtype=1
```

Figure 20-2: Editing an HTTP request on-the-fly using an intercepting proxy

From: Stuttard and Pinto, “The Web Application Hacker's Handbook”

Web Pen-Testing Tools: Common Features

- **Proxy**: intercept & manipulate requests


burp suite professional
burp intruder repeater window about
target proxy spider scanner intruder repeater sequencer decoder comparer options alerts
intercept options history
edit
remove
up
down
add

intercept client requests

☑ intercept if: ☑ update Content-Length

☑		file extension	does not match	(^gif$\|^jpg$|^png$\|^css$...
□	or	request	contains parameters	
□	or	HTTP method	does not match	(get|post)
□	and	URL	is in target scope	

and ▾ | domain name ▾ | matches ▾

intercept server responses

☑ intercept if: ☑ update Content-Length

☑		content type	matches	text
□	or	request	was modified	
□	or	request	was intercepted	
□	and	response code	does not match	^304$
□	and	URL	is in target scope	

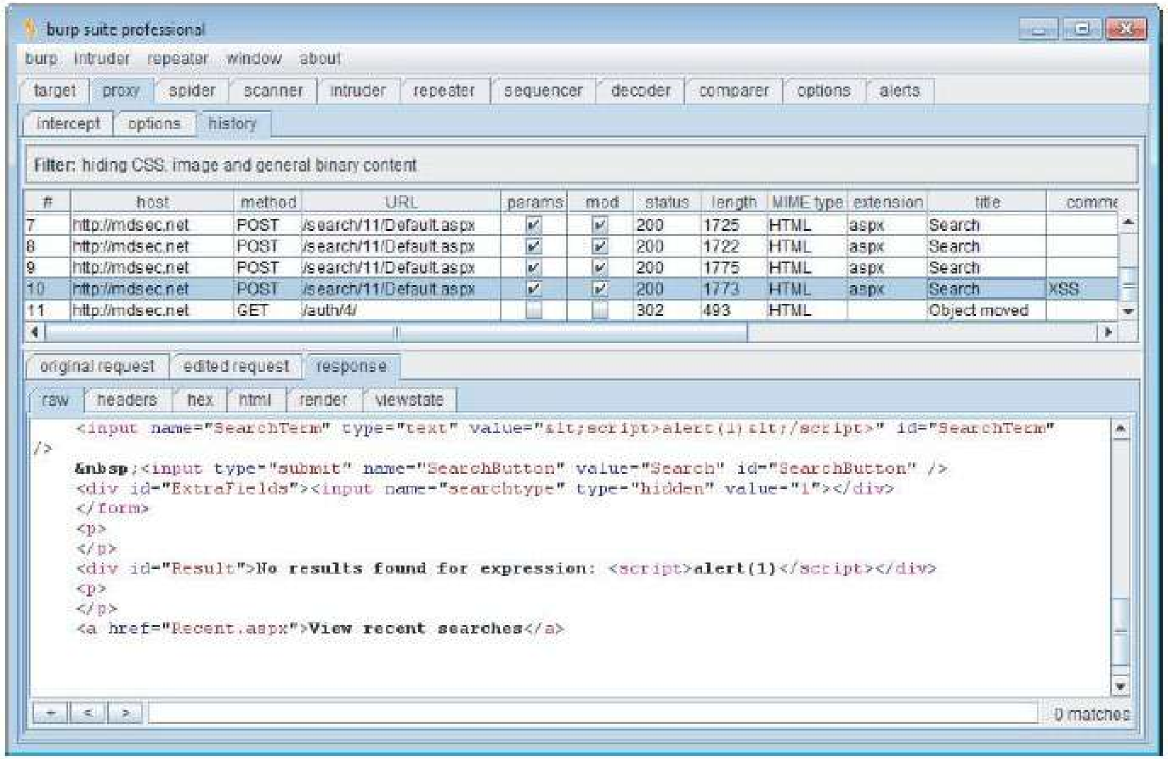
and ▾ | domain name ▾ | matches ▾

Figure 20-5: Burp proxy supports configuration of fine-grained rules for intercepting requests and responses

From: Stuttard and Pinto, “The Web Application Hacker's Handbook”

Web Pen-Testing Tools: Common Features

- **Proxy**: intercept & manipulate requests



burp suite professional
burp intruder repeater window about
target proxy spider scanner intruder repeater sequencer decoder comparer options alerts
intercept options history
original request edited request response
raw headers hex html render viewstate
+ < >
0 matches

Filter: hiding CSS, image and general binary content

#	host	method	URL	params	mod	status	length	MIME type	extension	title	comme
7	http://mdsec.net	POST	/search/11/Default.aspx	☑	☑	200	1725	HTML	aspx	Search	
8	http://mdsec.net	POST	/search/11/Default.aspx	☑	☑	200	1722	HTML	aspx	Search	
9	http://mdsec.net	POST	/search/11/Default.aspx	☑	☑	200	1775	HTML	aspx	Search	
10	http://mdsec.net	POST	/search/11/Default.aspx	☑	☑	200	1773	HTML	aspx	Search	XSS
11	http://mdsec.net	GET	/auth/4/	□	□	302	493	HTML		Object moved	

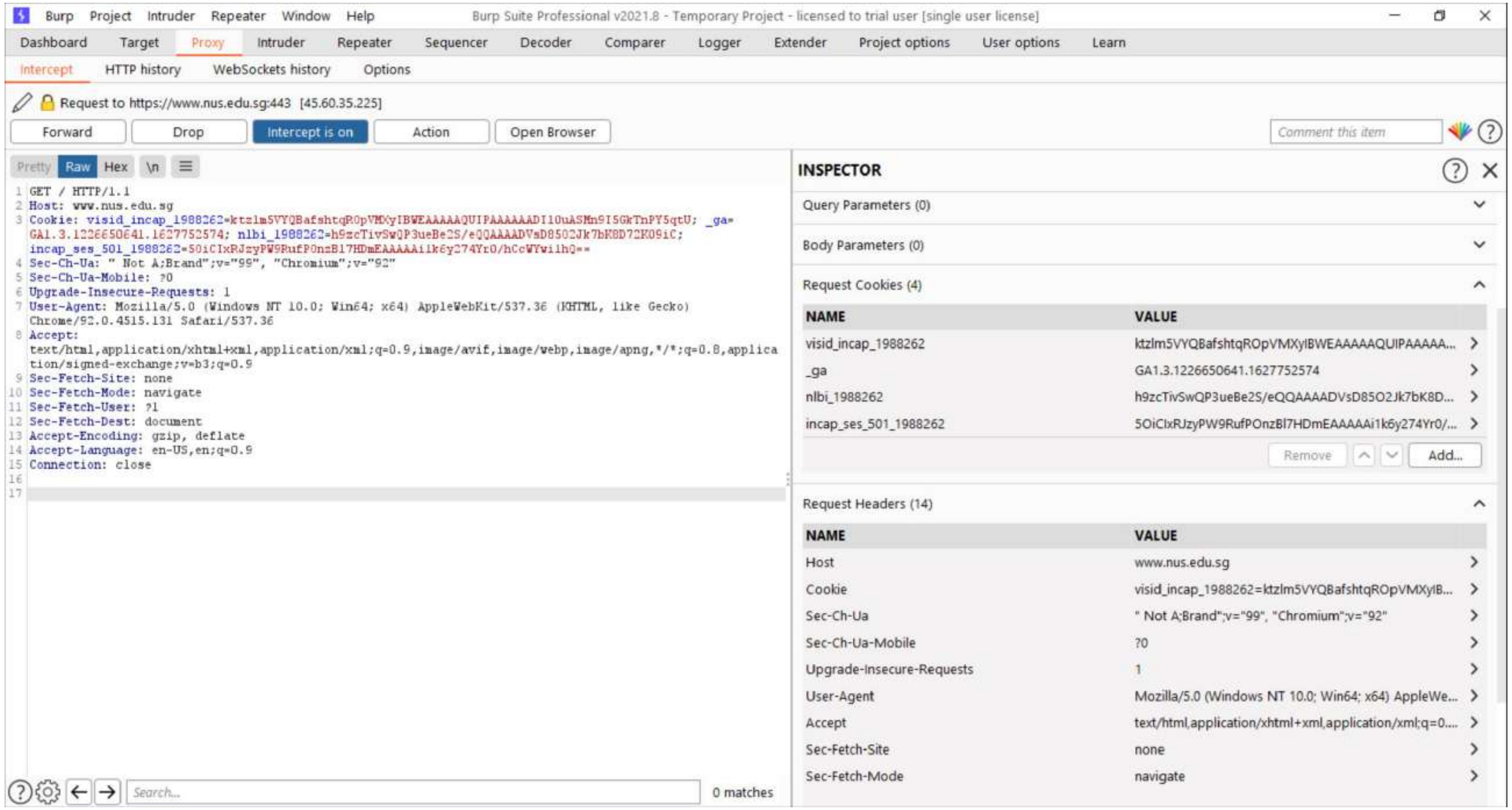
```
    <input name="SearchTerm" type="text" value="&lt;script>alert(1)&lt;/script>" id="SearchTerm"
/>
    &nbsp;<input type="submit" name="SearchButton" value="Search" id="SearchButton" />
    <div id="ExtraFields"><input name="searchtype" type="hidden" value="1"></div>
    </form>
    <p>
    </p>
    <div id="Result">No results found for expression: <script>alert(1)</script></div>
    <p>
    </p>
    <a href="Recent.aspx">View recent searches</a>
```

Figure 20-6: The proxy history, allowing you to view, filter, search, and annotate requests and responses made via the proxy

From: Stuttard and Pinto, “The Web Application Hacker's Handbook”

Web Pen-Testing Tools: Common Features

- **Proxy** in Burp 2



Burp Project Intruder Repeater Window Help
Burp Suite Professional v2021.8 - Temporary Project - licensed to trial user [single user license]
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn
Intercept HTTP history WebSockets history Options
Forward
Drop
Intercept is on
Action
Open Browser
Comment this item
Pretty Raw Hex \n
INSPECTOR
Search...
0 matches

Request to https://www.nus.edu.sg:443 [45.60.35.225]

```
GET / HTTP/1.1
Host: www.nus.edu.sg
Cookie: visid_incap_1988262=ktzlm5VYQBafshtqR0pVMXyIBWEAAAAAQUIPAAAAAADI10uASMn9I5GkTnPY5qtU; _ga=
GA1.3.1226650641.1627752574; nlbi_1988262=h9zcTivSwQP3ueBe2S/eQQAAAADVsD8502Jk7bK8D72K09iC;
incap_ses_501_1988262=50iCIxRJzyPW9RufP0nzBl7HDmEAAAAAi1k6y274Yr0/hCcWYwi1hQ==
Sec-Ch-Ua: " Not A;Brand";v="99", "Chromium";v="92"
Sec-Ch-Ua-Mobile: ?0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/92.0.4515.131 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applica
tion/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close


```

Query Parameters (0)

Body Parameters (0)

Request Cookies (4)

NAME	VALUE
visid_incap_1988262	ktzlm5VYQBafshtqROpVMXyIBWEAAAAAQUIPAAAAA...
_ga	GA1.3.1226650641.1627752574
nlbi_1988262	h9zcTivSwQP3ueBe2S/eQQAAAADVsD85O2Jk7bK8D...
incap_ses_501_1988262	5OiClxRJzyPW9RufPOnzBl7HDmEAAAAAi1k6y274Yr0/...

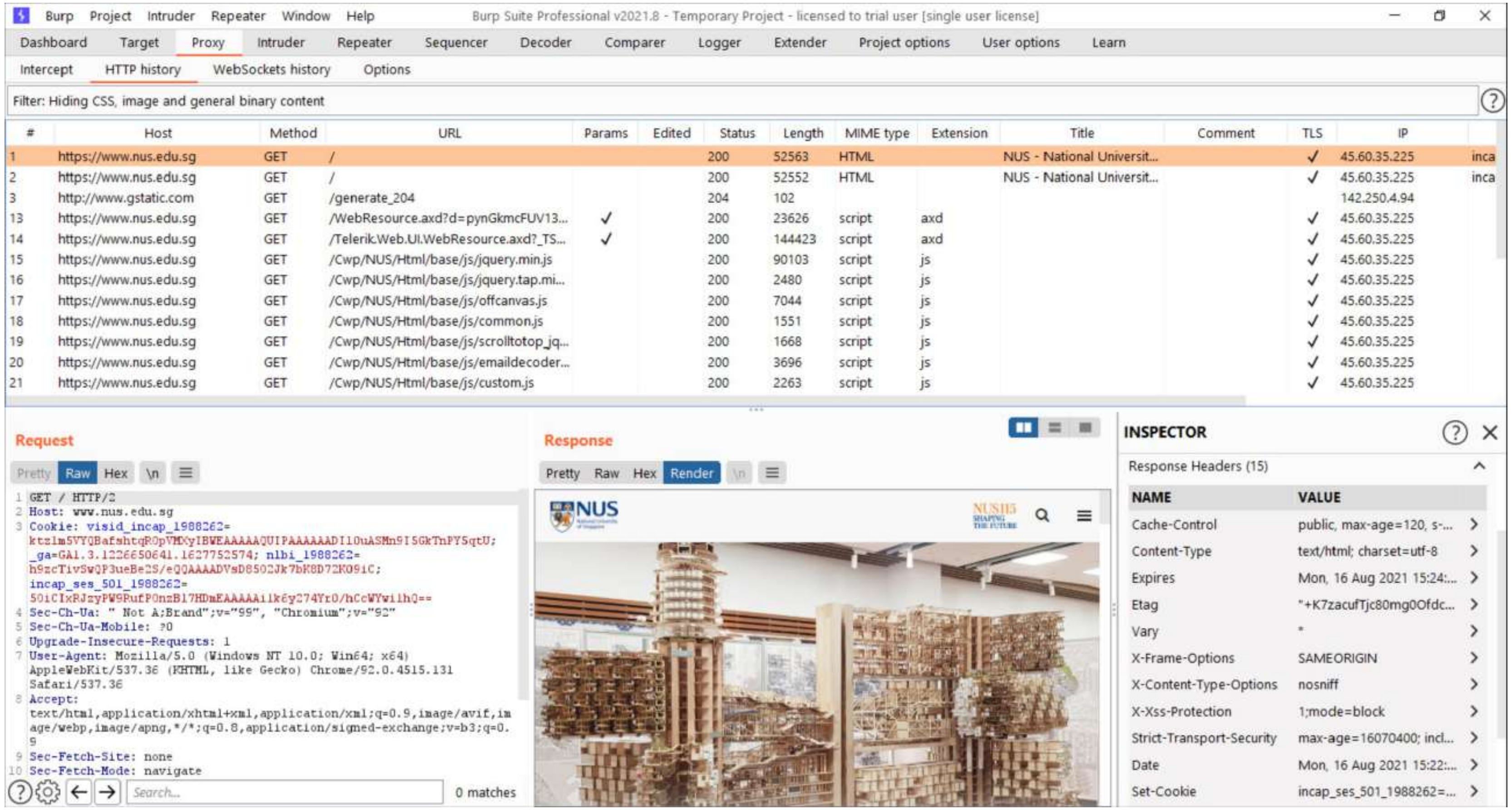
Remove Add...

Request Headers (14)

NAME	VALUE
Host	www.nus.edu.sg
Cookie	visid_incap_1988262=ktzlm5VYQBafshtqROpVMXyIB...
Sec-Ch-Ua	" Not A;Brand";v="99", "Chromium";v="92"
Sec-Ch-Ua-Mobile	?0
Upgrade-Insecure-Requests	1
User-Agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWe...
Accept	text/html,application/xhtml+xml,application/xml;q=0....
Sec-Fetch-Site	none
Sec-Fetch-Mode	navigate

Web Pen-Testing Tools: Common Features

- **Proxy** in Burp 2



Burp Project Intruder Repeater Window Help
Burp Suite Professional v2021.8 - Temporary Project - licensed to trial user [single user license]
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn
Intercept HTTP history WebSockets history Options

Filter: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title	Comment	TLS	IP	
1	https://www.nus.edu.sg	GET	/			200	52563	HTML		NUS - National Universit...		✓	45.60.35.225	inca
2	https://www.nus.edu.sg	GET	/			200	52552	HTML		NUS - National Universit...		✓	45.60.35.225	inca
3	http://www.gstatic.com	GET	/generate_204			204	102						142.250.4.94	
13	https://www.nus.edu.sg	GET	/WebResource.axd?d=pynGkmcFUV13...	✓		200	23626	script	axd			✓	45.60.35.225	
14	https://www.nus.edu.sg	GET	/Telerik.Web.UI.WebResource.axd?_TS...	✓		200	144423	script	axd			✓	45.60.35.225	
15	https://www.nus.edu.sg	GET	/Cwp/NUS/Html/base/js/jquery.min.js			200	90103	script	js			✓	45.60.35.225	
16	https://www.nus.edu.sg	GET	/Cwp/NUS/Html/base/js/jquery.tap.mi...			200	2480	script	js			✓	45.60.35.225	
17	https://www.nus.edu.sg	GET	/Cwp/NUS/Html/base/js/offcanvas.js			200	7044	script	js			✓	45.60.35.225	
18	https://www.nus.edu.sg	GET	/Cwp/NUS/Html/base/js/common.js			200	1551	script	js			✓	45.60.35.225	
19	https://www.nus.edu.sg	GET	/Cwp/NUS/Html/base/js/scrolltotop_jq...			200	1668	script	js			✓	45.60.35.225	
20	https://www.nus.edu.sg	GET	/Cwp/NUS/Html/base/js/emaildecoder...			200	3696	script	js			✓	45.60.35.225	
21	https://www.nus.edu.sg	GET	/Cwp/NUS/Html/base/js/custom.js			200	2263	script	js			✓	45.60.35.225	

Request

```
GET / HTTP/2
Host: www.nus.edu.sg
Cookie: visid_incap_1988262=
ktzlm5VYQBafshtqR0pVMXyIBWEAAAAAQUIPAAAAAADI10uASMn9I5GkTnPY5qtU;
_ga=GA1.3.1226650641.1627752574; nlbi_1988262=
h9zcTivSwQP3ueBe2S/eQQAAAADVsD8502Jk7bK8D72K09iC;
incap_ses_501_1988262=
50iCIxRJzyPW9RufP0nzB17HDmEAAAAAi1k6y274Yr0/hCcWYwi1hQ==
Sec-Ch-Ua: " Not A;Brand";v="99", "Chromium";v="92"
Sec-Ch-Ua-Mobile: ?0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.131
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,im
age/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.
9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
```

Response

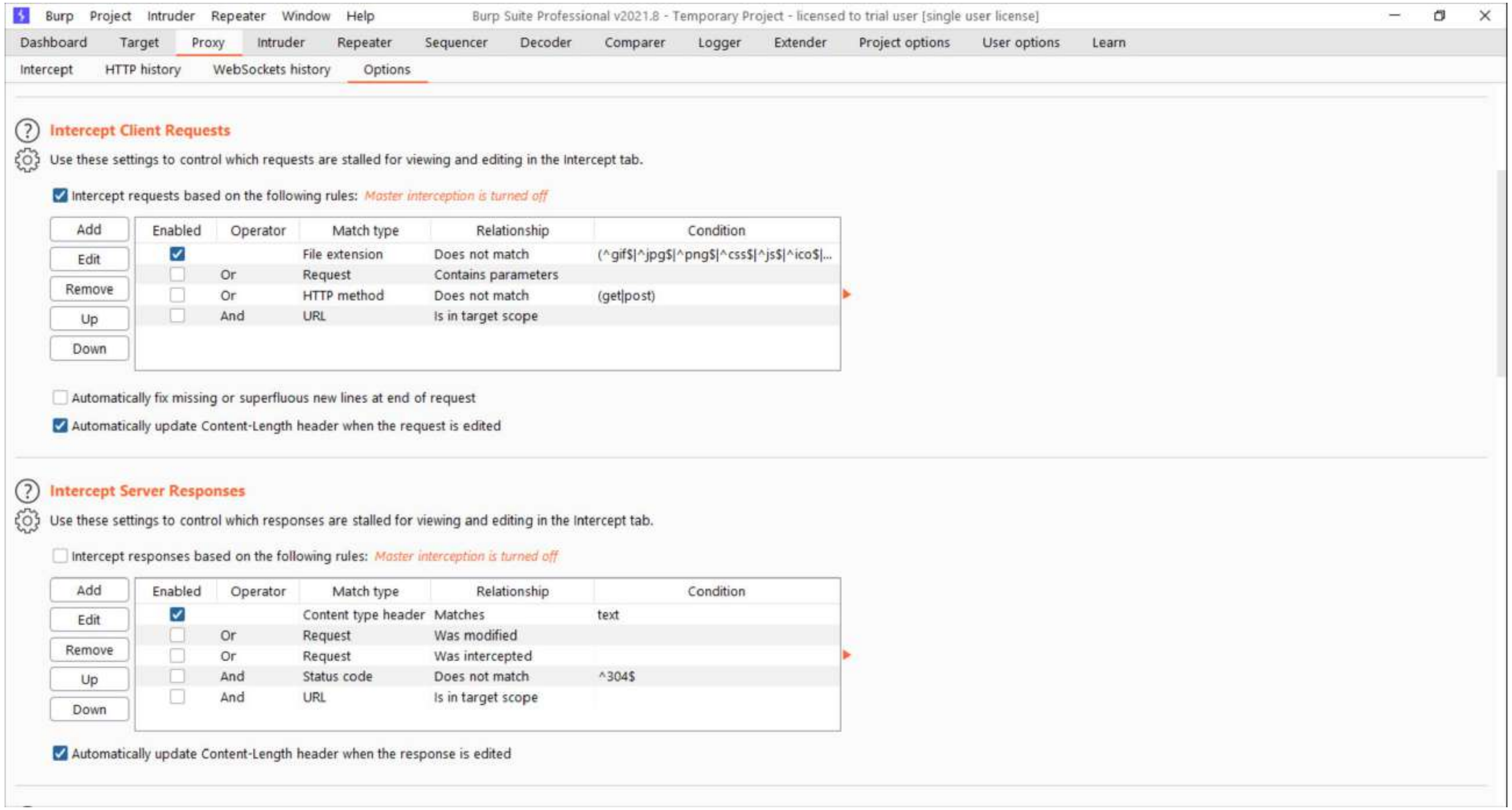
INSPECTOR

Response Headers (15)

NAME	VALUE
Cache-Control	public, max-age=120, s-...
Content-Type	text/html; charset=utf-8
Expires	Mon, 16 Aug 2021 15:24:...
Etag	"+K7zacufTjc80mg0Ofdc...
Vary	*
X-Frame-Options	SAMEORIGIN
X-Content-Type-Options	nosniff
X-Xss-Protection	1;mode=block
Strict-Transport-Security	max-age=16070400; incl...
Date	Mon, 16 Aug 2021 15:22:...
Set-Cookie	incap_ses_501_1988262=...

Web Pen-Testing Tools: Common Features

- **Proxy** in Burp 2



Burp Project Intruder Repeater Window Help
Burp Suite Professional v2021.8 - Temporary Project - licensed to trial user [single user license]
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn
Intercept HTTP history WebSockets history Options
Add
Edit
Remove
Up
Down

Intercept Client Requests

Use these settings to control which requests are stalled for viewing and editing in the Intercept tab.

☑ Intercept requests based on the following rules: *Master interception is turned off*

Enabled	Operator	Match type	Relationship	Condition
☑		File extension	Does not match	(^gif$\|^jpg$|^png$\|^css$|^js$\|^ico$|...
☐	Or	Request	Contains parameters	
☐	Or	HTTP method	Does not match	(get|post)
☐	And	URL	Is in target scope	

☐ Automatically fix missing or superfluous new lines at end of request

☑ Automatically update Content-Length header when the request is edited

Intercept Server Responses

Use these settings to control which responses are stalled for viewing and editing in the Intercept tab.

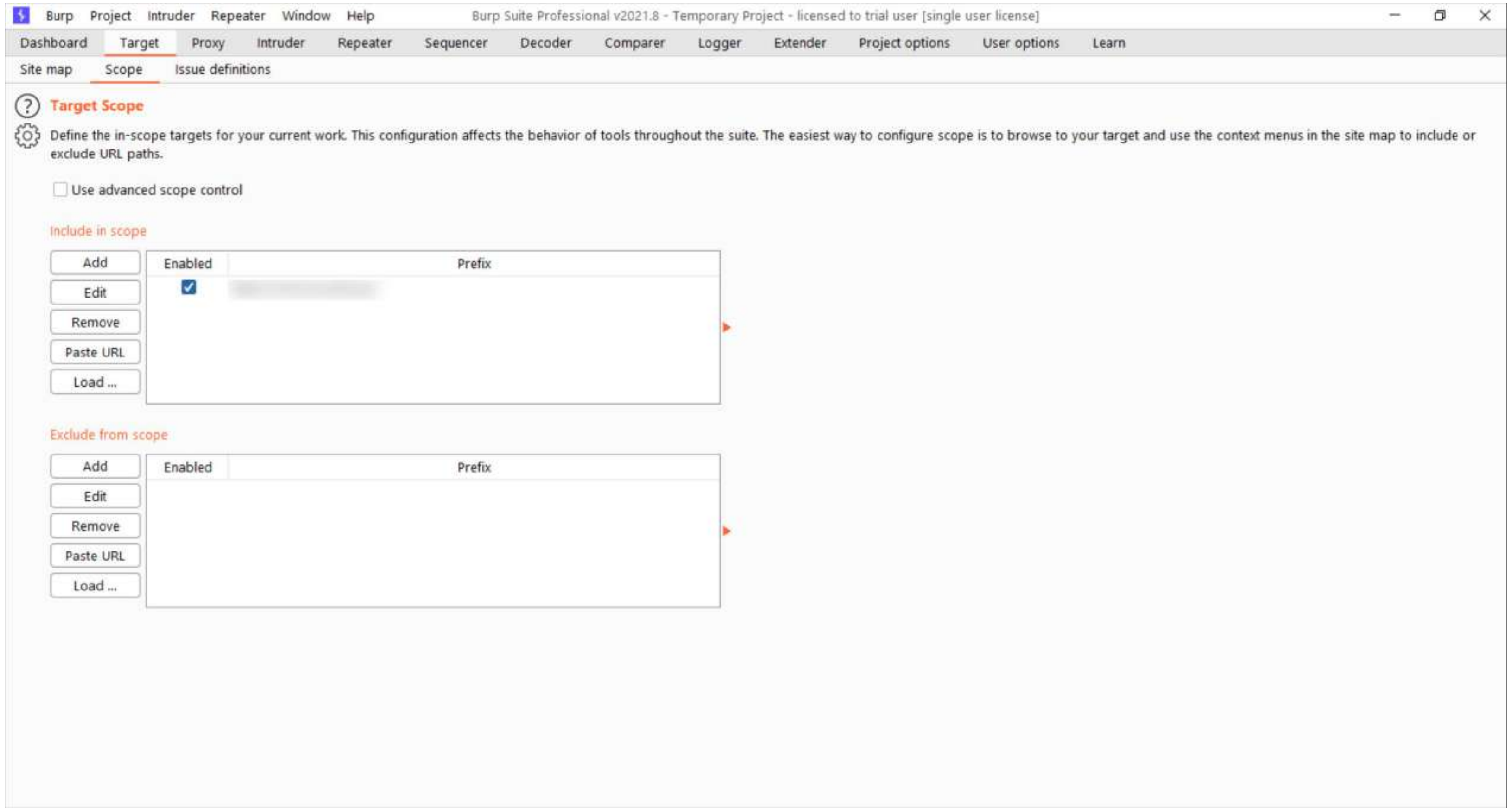
☐ Intercept responses based on the following rules: *Master interception is turned off*

Enabled	Operator	Match type	Relationship	Condition
☑		Content type header	Matches	text
☐	Or	Request	Was modified	
☐	Or	Request	Was intercepted	
☐	And	Status code	Does not match	^304$
☐	And	URL	Is in target scope	

☑ Automatically update Content-Length header when the response is edited

Web Pen-Testing Tools: Common Features

- Web application **target**: for specifying **in-scope target**



Target Scope

Define the in-scope targets for your current work. This configuration affects the behavior of tools throughout the suite. The easiest way to configure scope is to browse to your target and use the context menus in the site map to include or exclude URL paths.

□ Use advanced scope control

Include in scope

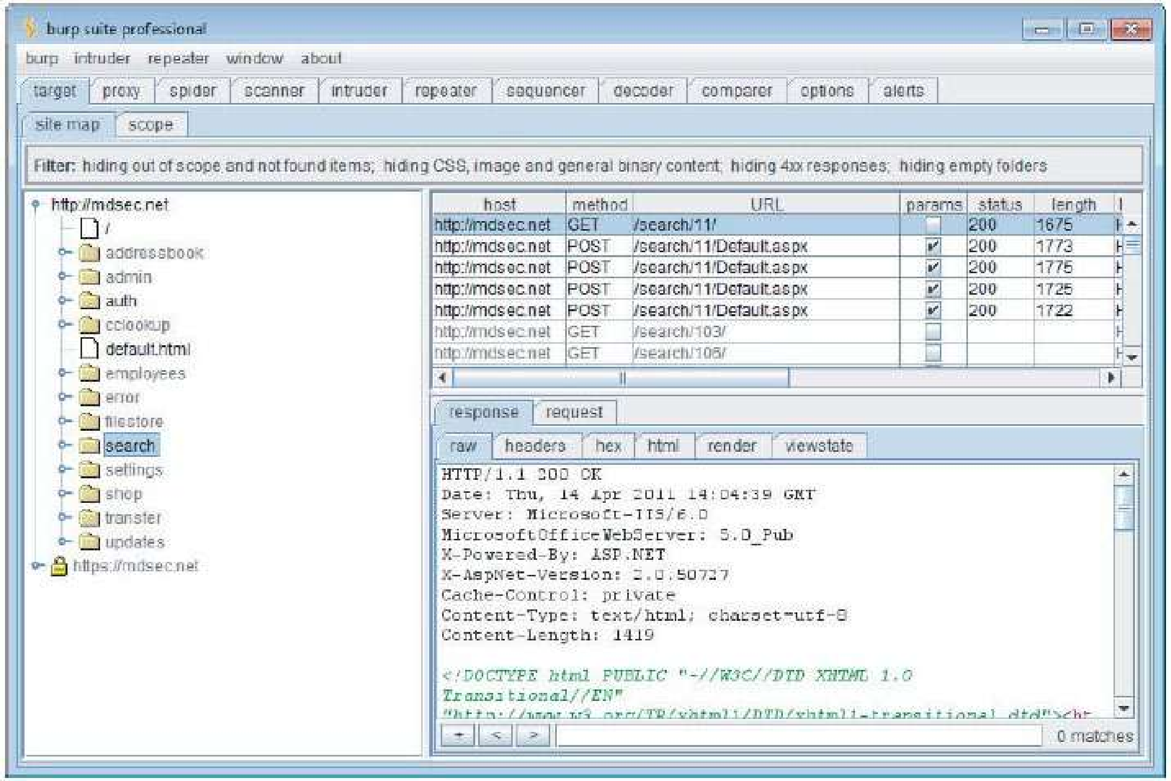
Enabled	Prefix
☑	[redacted]

Exclude from scope

Enabled	Prefix

Web Pen-Testing Tools: Common Features

- Web application **spidering/*crawling***



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

host	method	URL	params	status	length
http://mdsec.net	GET	/search/11/	☐	200	1675
http://mdsec.net	POST	/search/11/Default.aspx	☑	200	1773
http://mdsec.net	POST	/search/11/Default.aspx	☑	200	1775
http://mdsec.net	POST	/search/11/Default.aspx	☑	200	1725
http://mdsec.net	POST	/search/11/Default.aspx	☑	200	1722
http://mdsec.net	GET	/search/103/	☐		
http://mdsec.net	GET	/search/108/	☐		

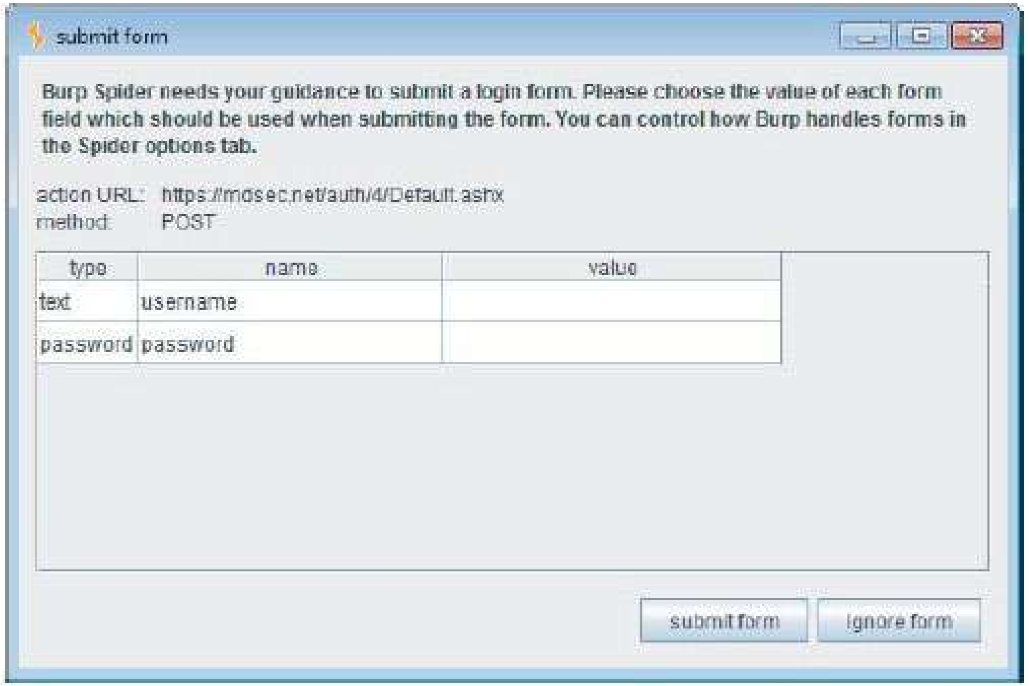
```
HTTP/1.1 200 OK
Date: Thu, 14 Apr 2011 14:04:39 GMT
Server: Microsoft-IIS/6.0
MicrosoftOfficeWebServer: 5.0_Pub
X-Powered-By: ASP.NET
X-AspNet-Version: 2.0.50727
Cache-Control: private
Content-Type: text/html; charset=utf-8
Content-Length: 1419

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"><ht
```

Figure 20-7: The results of passive application spidering, where items in gray have been identified passively but not yet requested

From: Stuttard and Pinto, “The Web Application Hacker's Handbook”

Web Pen-Testing Tools: Common Features

- Web application **spidering/*crawling***: form submission



Burp Spider needs your guidance to submit a login form. Please choose the value of each form field which should be used when submitting the form. You can control how Burp handles forms in the Spider options tab.

action URL: https://mdsec.net/auth/4/Default.ashx
method: POST

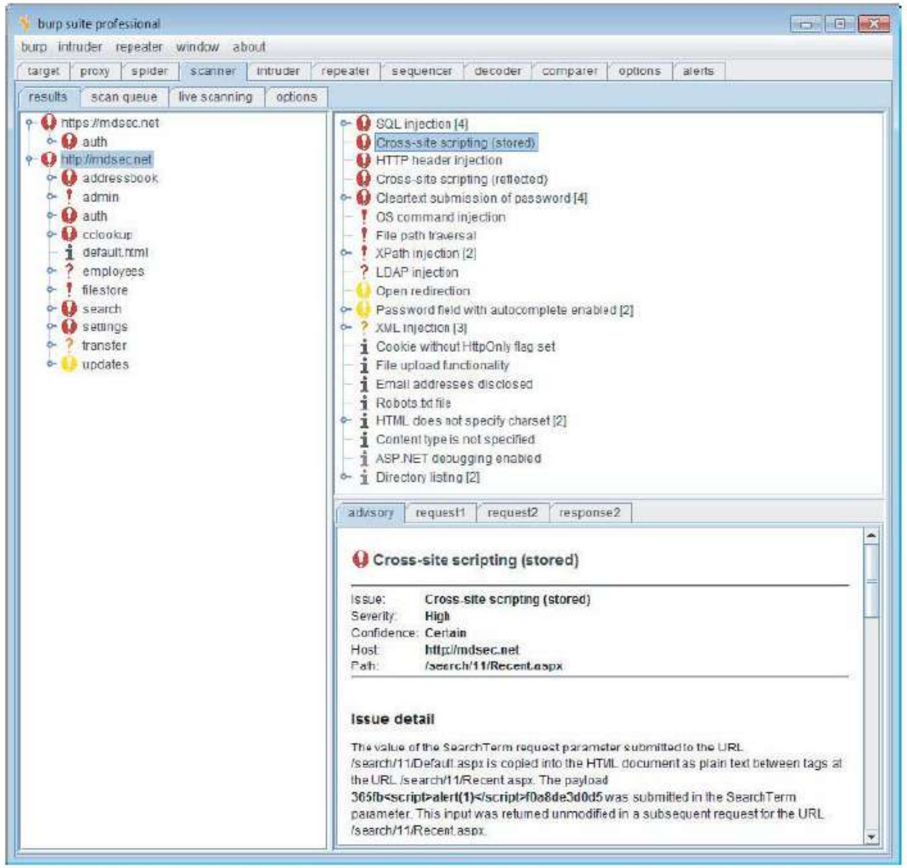
type	name	value
text	username	
password	password	

Figure 20-8: Burp Spider prompting for user guidance when submitting forms

From: Stuttard and Pinto, “The Web Application Hacker's Handbook”

Web Pen-Testing Tools: Common Features

- Vulnerability **scanner/*auditing***


burp suite professional
burp intruder repeater window about
target proxy spider scanner intruder repeater sequencer decoder comparer options alerts
results scan queue live scanning options
advisory request1 request2 response2

https://mdsec.net
auth
http://mdsec.net
addressbook
admin
auth
cclookup
default.html
employees
filestore
search
settings
transfer
updates

SQL injection [4]
Cross-site scripting (stored)
HTTP header injection
Cross-site scripting (reflected)
Cleartext submission of password [4]
OS command injection
File path traversal
XPath injection [2]
LDAP injection
Open redirection
Password field with autocomplete enabled [2]
XML injection [3]
Cookie without HttpOnly flag set
File upload functionality
Email addresses disclosed
Robots.txt file
HTML does not specify charset [2]
Content type is not specified
ASP.NET debugging enabled
Directory listing [2]

Cross-site scripting (stored)

Issue:	**Cross-site scripting (stored)**
Severity:	**High**
Confidence:	**Certain**
Host:	**http://mdsec.net**
Path:	**/search/11/Recent.aspx**

Issue detail

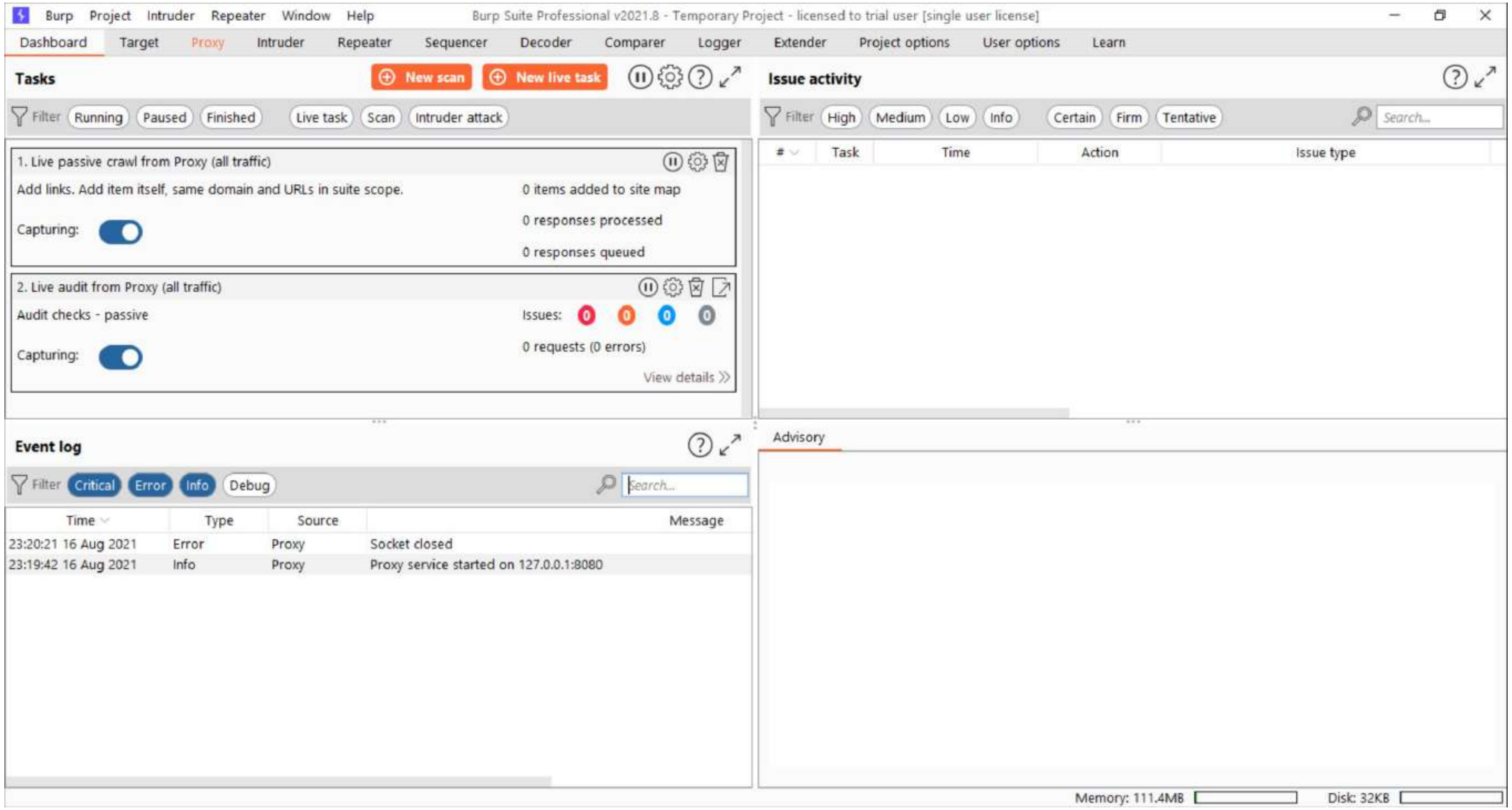
The value of the SearchTerm request parameter submitted to the URL /search/11/Default.aspx is copied into the HTML document as plain text between tags at the URL /search/11/Recent.aspx. The payload **365fb<script>alert(1)</script>f0a8de3d0d5** was submitted in the SearchTerm parameter. This input was returned unmodified in a subsequent request for the URL /search/11/Recent.aspx.

Figure 20-10: The results of live scanning as you browse with Burp Scanner

From: Stuttard and Pinto, “The Web Application Hacker's Handbook”

Web Pen-Testing Tools: Common Features

- **Scanner** launchable from the **Dashboard** in Burp 2



Burp Project Intruder Repeater Window Help
Burp Suite Professional v2021.8 - Temporary Project - licensed to trial user [single user license]
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn
Tasks
New scan
New live task
Filter Running Paused Finished Live task Scan Intruder attack
Issue activity
Filter High Medium Low Info Certain Firm Tentative
Search...
# Task Time Action Issue type
Event log
Filter Critical Error Info Debug
Search...
Advisory
View details
Memory: 111.4MB
Disk: 32KB

1. Live passive crawl from Proxy (all traffic)

Add links. Add item itself, same domain and URLs in suite scope.

Capturing:

0 items added to site map

0 responses processed

0 responses queued

2. Live audit from Proxy (all traffic)

Audit checks - passive

Capturing:

Issues: 0 0 0 0

0 requests (0 errors)

Time	Type	Source	Message
23:20:21 16 Aug 2021	Error	Proxy	Socket closed
23:19:42 16 Aug 2021	Info	Proxy	Proxy service started on 127.0.0.1:8080

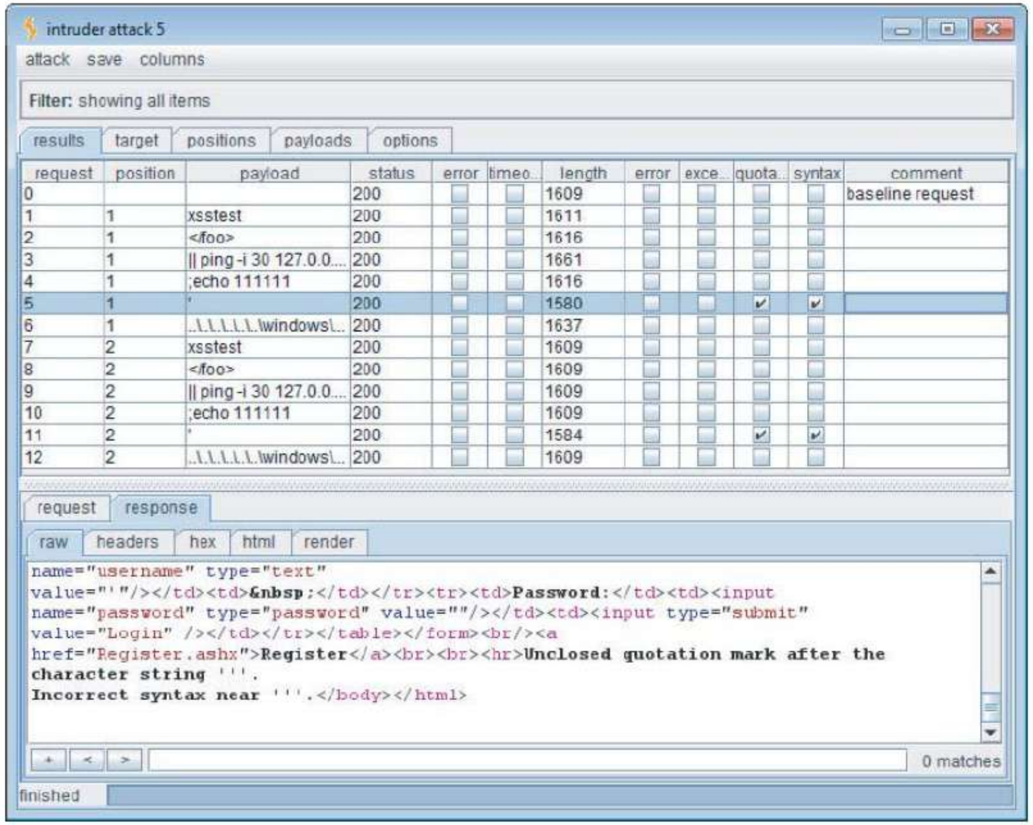
Web Pen-Testing Tools: Common Features

- Some other **web vulnerability scanners**:
  - Acunetix
  - AppScan
  - Hailstorm
  - NetSparker
  - N-Stalker
  - NTOSpider
  - Skipfish
  - WebInspect

- Evaluation and analysis:
  Doupe et al., *Why Johnny Can't Pentest: An Analysis of Black-box Web Vulnerability Scanners*, DIMVA, 2010

Web Pen-Testing Tools: Common Features

- Customizable web-app login **fuzzer** (**Intruder**)



intruder attack 5
attack save columns
Filter: showing all items
results target positions payloads options
request response
raw headers hex html render
+ < >
0 matches
finished

request	position	payload	status	error	timeo...	length	error	exce...	quota...	syntax	comment
0			200	□	□	1609	□	□	□	□	baseline request
1	1	xsstest	200	□	□	1611	□	□	□	□	
2	1	</foo>	200	□	□	1616	□	□	□	□	
3	1	|| ping -i 30 127.0.0....	200	□	□	1661	□	□	□	□	
4	1	;echo 111111	200	□	□	1616	□	□	□	□	
5	1	'	200	□	□	1580	□	□	☑	☑	
6	1	..\..\..\..\..\..\windows\...	200	□	□	1637	□	□	□	□	
7	2	xsstest	200	□	□	1609	□	□	□	□	
8	2	</foo>	200	□	□	1609	□	□	□	□	
9	2	|| ping -i 30 127.0.0....	200	□	□	1609	□	□	□	□	
10	2	;echo 111111	200	□	□	1609	□	□	□	□	
11	2	'	200	□	□	1584	□	□	☑	☑	
12	2	..\..\..\..\..\..\windows\...	200	□	□	1609	□	□	□	□	

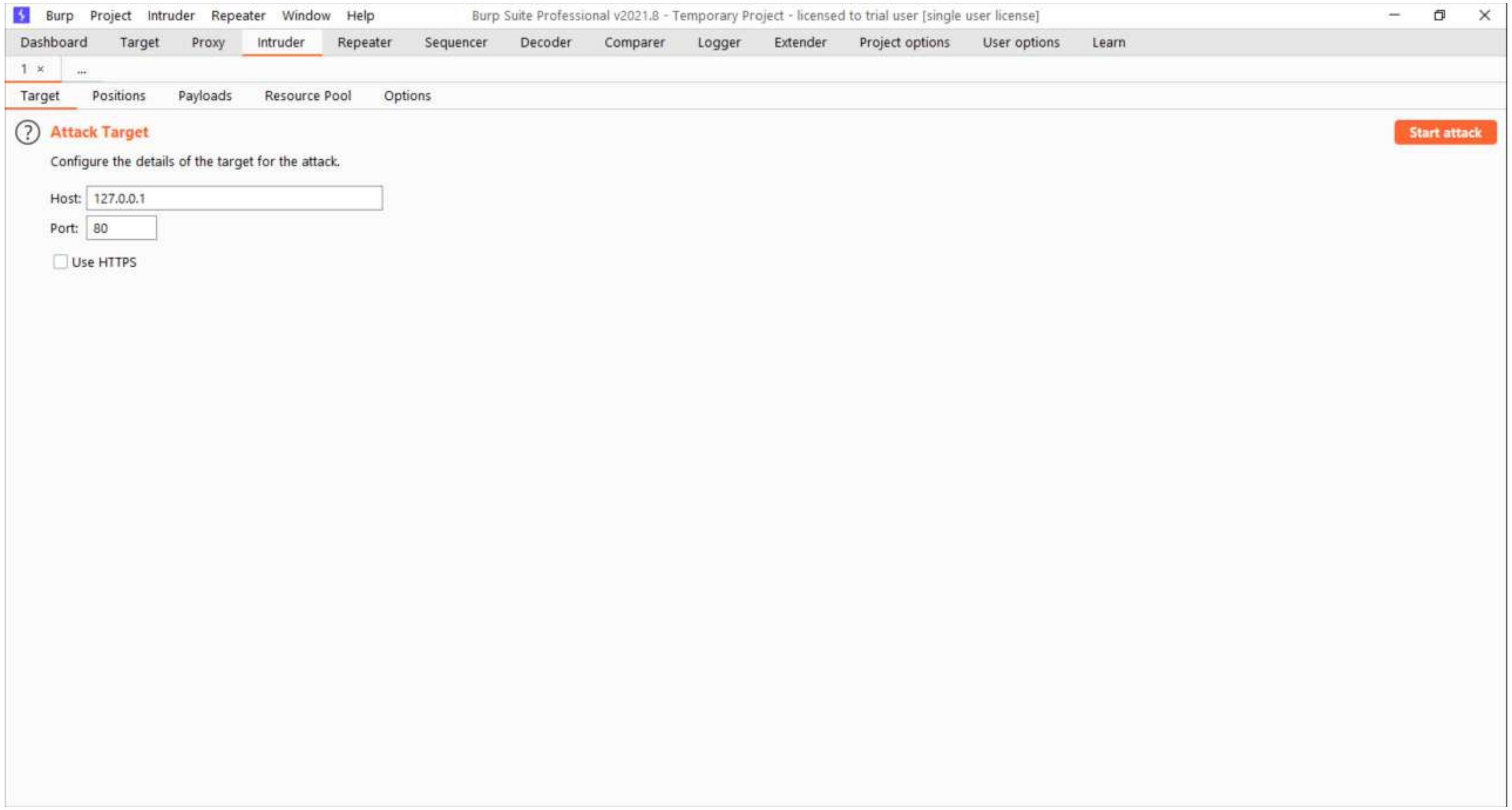
```
name="username" type="text"
value="'"/></td><td>&nbsp;</td></tr><tr><td>Password:</td><td><input
name="password" type="password" value=""/></td><td><input type="submit"
value="Login" /></td></tr></table></form><br/><a
href="Register.ashx">Register</a><br><br><hr>Unclosed quotation mark after the
character string '''.
Incorrect syntax near '''.</body></html>
```

Figure 20-9: The results of a fuzzing exercise using Burp Intruder

From: Stuttard and Pinto, “The Web Application Hacker's Handbook”

Web Pen-Testing Tools: Common Features

- **Intruder** in Burp 2: target site



Burp Project Intruder Repeater Window Help
Burp Suite Professional v2021.8 - Temporary Project - licensed to trial user [single user license]
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn
1 × ...
Target Positions Payloads Resource Pool Options
Start attack

Attack Target

Configure the details of the target for the attack.

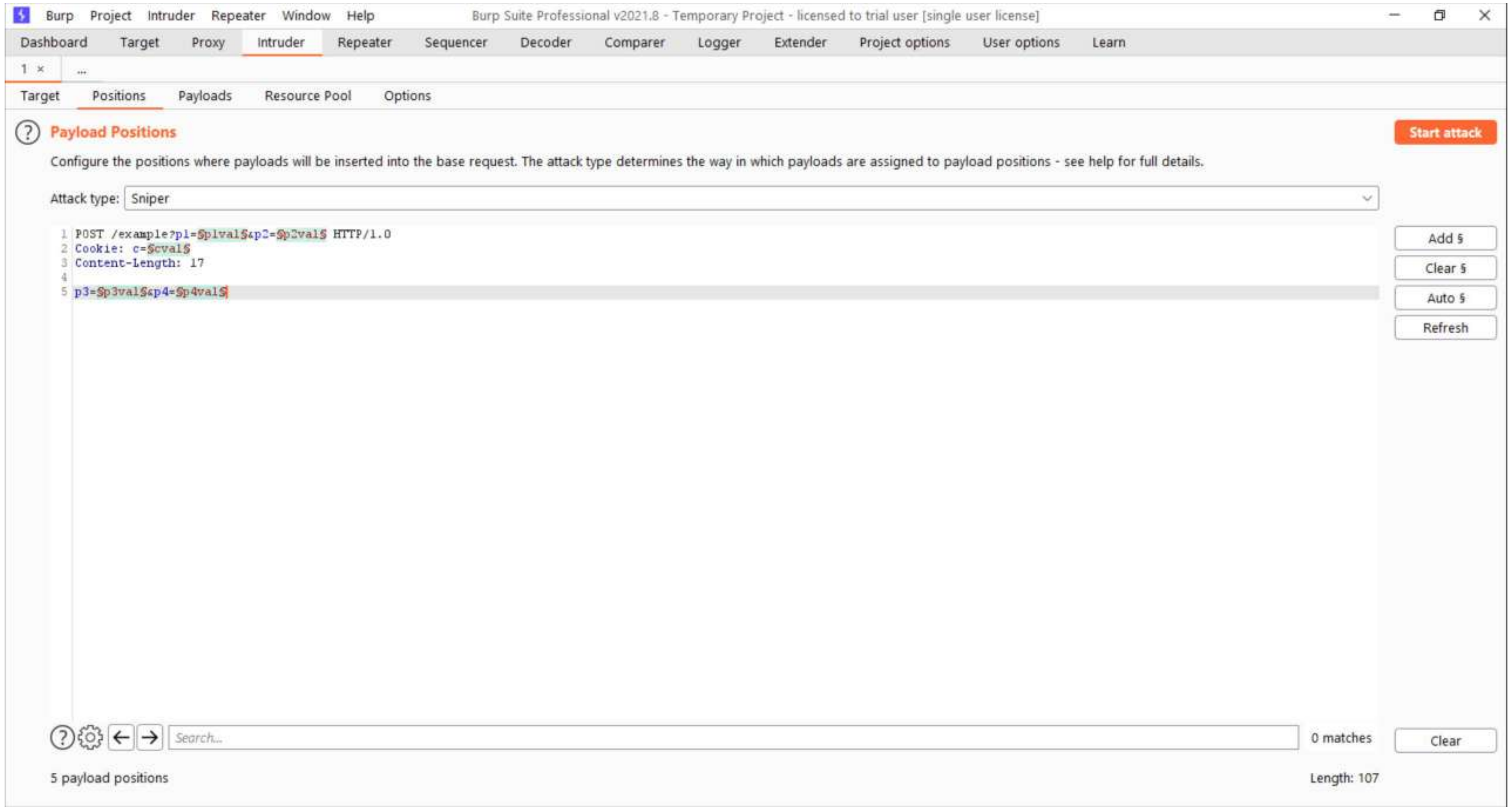
Host: 127.0.0.1

Port: 80

□ Use HTTPS

Web Pen-Testing Tools: Common Features

- **Intruder** in Burp 2: payload positions


Burp Project Intruder Repeater Window Help
Burp Suite Professional v2021.8 - Temporary Project - licensed to trial user [single user license]
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn
1 × ...
Target Positions Payloads Resource Pool Options
Start attack
Add §
Clear §
Auto §
Refresh
Search...
0 matches
Clear

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: Sniper

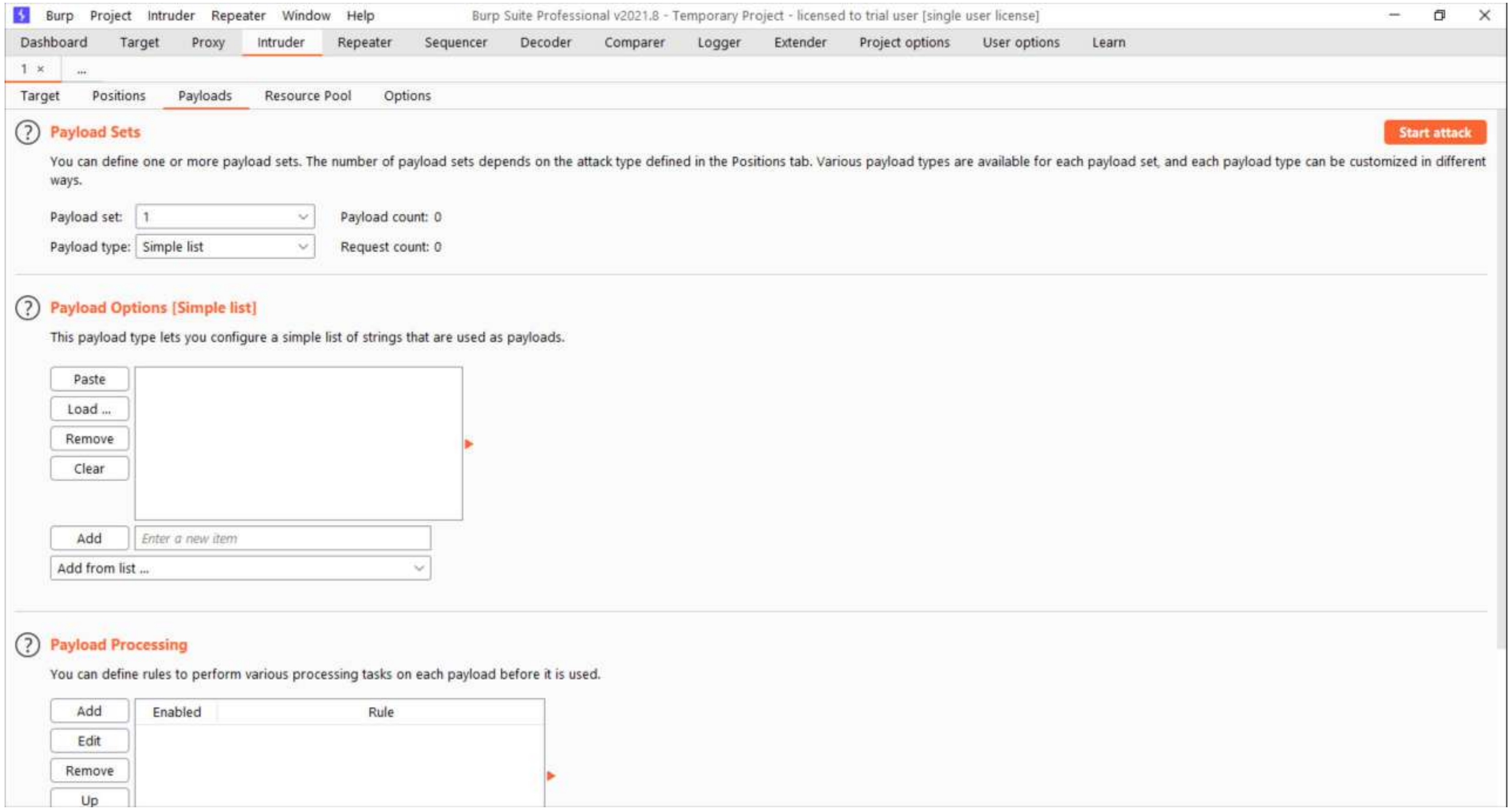
```
POST /example?p1=§p1val§&p2=§p2val§ HTTP/1.0
Cookie: c=§cval§
Content-Length: 17

p3=§p3val§&p4=§p4val§
```

5 payload positions

Length: 107

Web Pen-Testing Tools: Common Features

- **Intruder** in Burp 2: payload options



Burp Project Intruder Repeater Window Help
Burp Suite Professional v2021.8 - Temporary Project - licensed to trial user [single user license]
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn
1 × ...
Target Positions Payloads Resource Pool Options
Start attack
Paste
Load ...
Remove
Clear
Add
Enter a new item
Add from list ...
Add
Edit
Remove
Up
Enabled
Rule

Payload Sets

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

Payload set: 1 Payload count: 0

Payload type: Simple list Request count: 0

Payload Options [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

Payload Processing

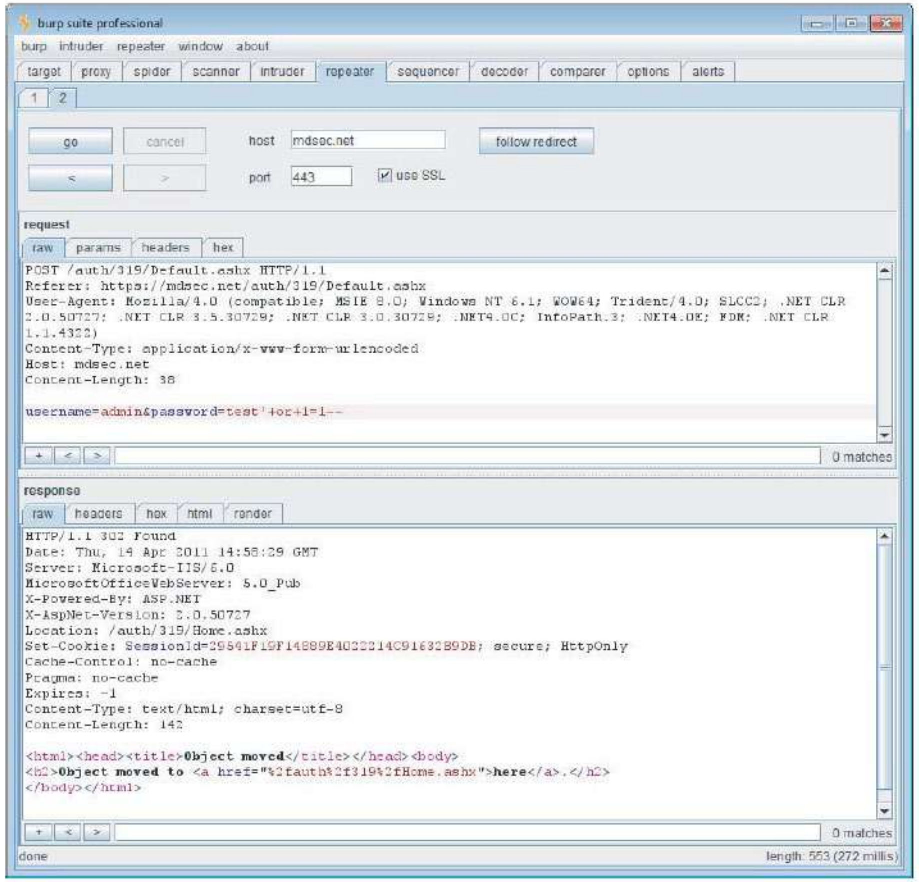
You can define rules to perform various processing tasks on each payload before it is used.

An Alternative Tool for Fuzzing

- **FFUF**, or “Fuzz Faster U Fool”
- Usage:
  - Directory brute forcing
  - Virtual host discovery
  - GET parameter fuzzing
  - POST data fuzzing
- Resources:
  - Github page: https://github.com/ffuf/ffuf
  - Documentation page: https://codingo.io/tools/ffuf/bounty/2020/09/17/everything-you-need-to-know-about-ffuf.html
  - See its video tutorial as well!
  - Also see a comparison with **Wfuzz**: …..

Web Pen-Testing Tools: Common Features

- Manual request tool (e.g. **Repeater**)


burp suite professional
burp intruder repeater window about
target proxy spider scanner intruder repeater sequencer decoder comparer options alerts
1 2
go
cancel
<
>
follow redirect
request
raw params headers hex
+ < >
0 matches
response
raw headers hex html render
+ < >
0 matches
done
length: 553 (272 millis)

host mdsec.net

port 443 ☑ use SSL

```
POST /auth/319/Default.ashx HTTP/1.1
Referer: https://mdsec.net/auth/319/Default.ashx
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; Trident/4.0; SLCC2; .NET CLR
2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; .NET4.0C; InfoPath.3; .NET4.0E; FDM; .NET CLR
1.1.4322)
Content-Type: application/x-www-form-urlencoded
Host: mdsec.net
Content-Length: 38

username=admin&password=test'+or+1=1--
```

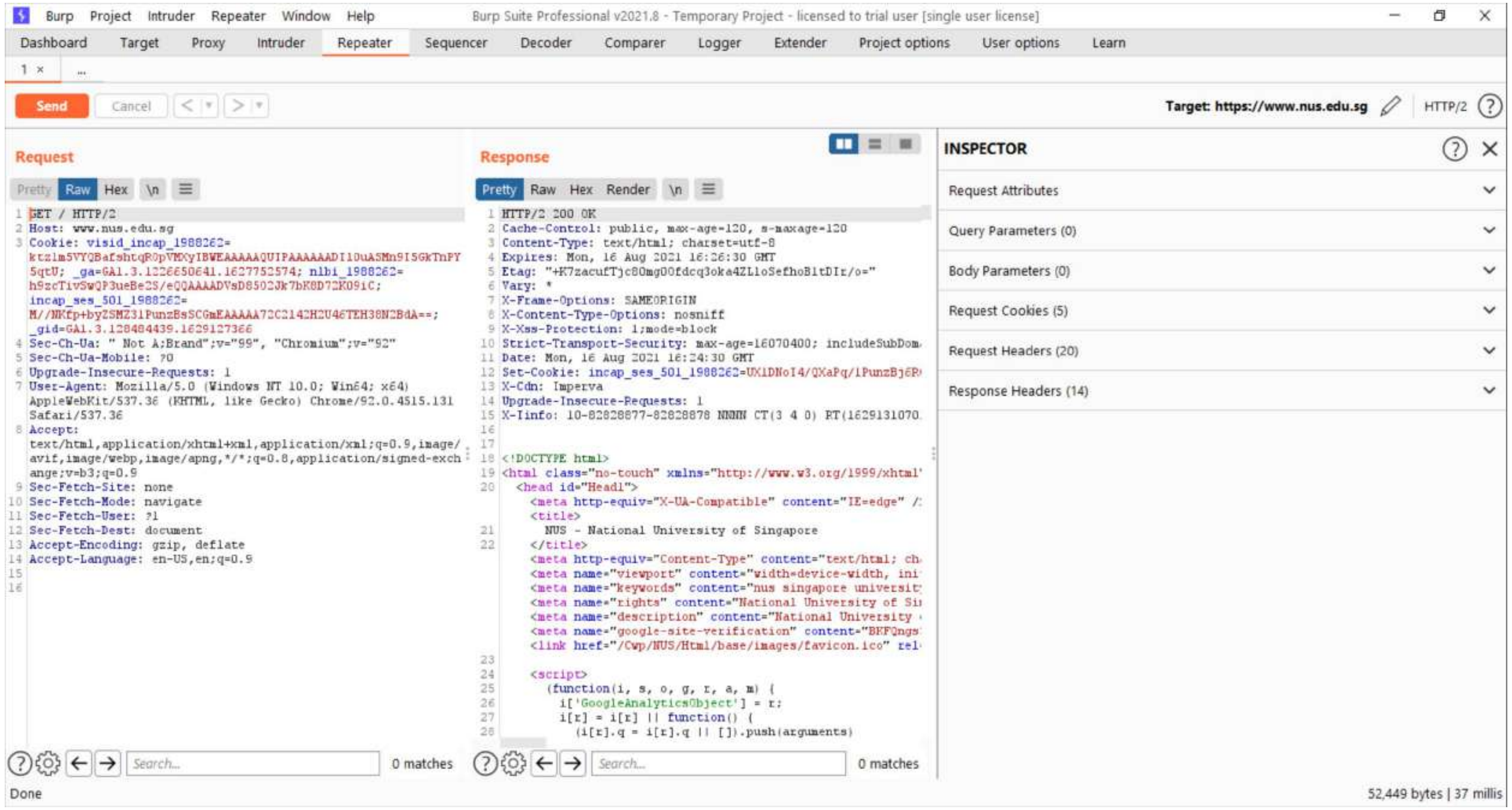
```
HTTP/1.1 302 Found
Date: Thu, 14 Apr 2011 14:58:29 GMT
Server: Microsoft-IIS/6.0
MicrosoftOfficeWebServer: 5.0_Pub
X-Powered-By: ASP.NET
X-AspNet-Version: 2.0.50727
Location: /auth/319/Home.ashx
Set-Cookie: SessionId=29541F19F14889E4022214C91632B9DB; secure; HttpOnly
Cache-Control: no-cache
Pragma: 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 <a href="%2fauth%2f319%2fHome.ashx">here</a>.</h2>
</body></html>
```

Figure 20-11: A request being reissued manually using Burp Repeater

From: Stuttard and Pinto, “The Web Application Hacker's Handbook”

Web Pen-Testing Tools: Common Features

- **Repeater** in Burp 2



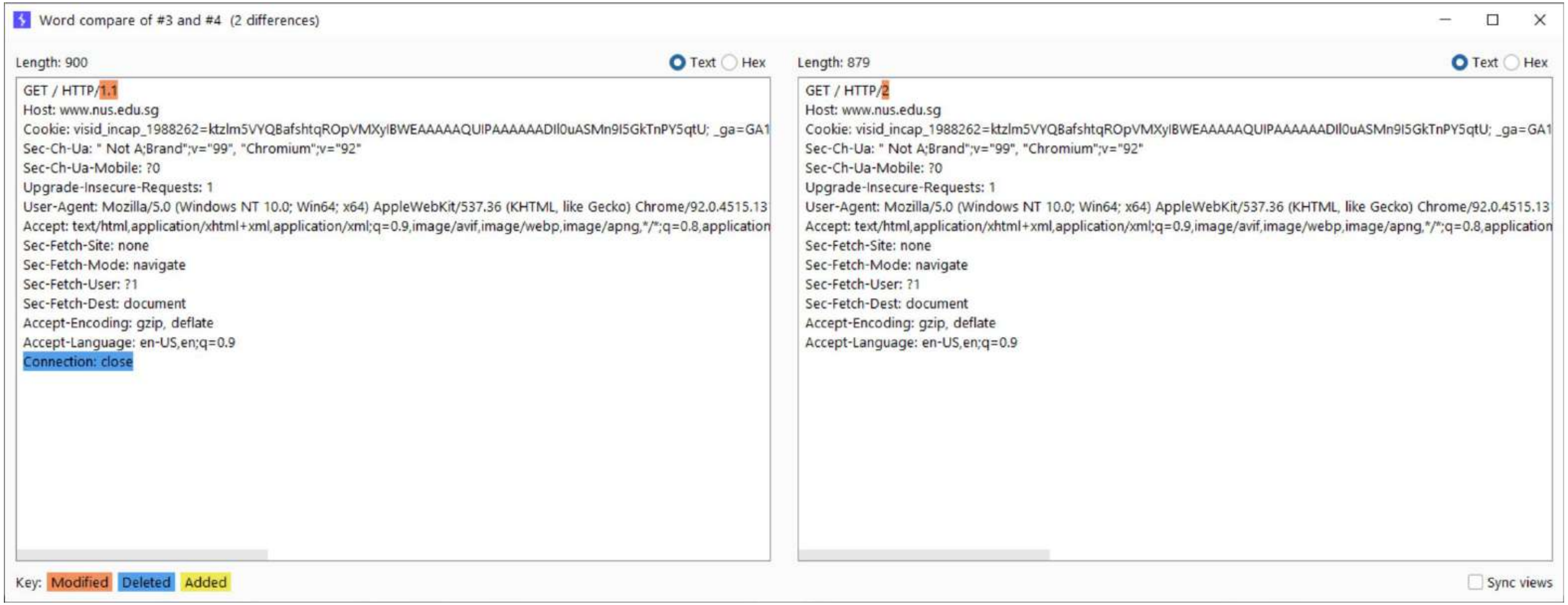
Burp Project Intruder Repeater Window Help
Burp Suite Professional v2021.8 - Temporary Project - licensed to trial user [single user license]
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn
1 ×
...
Send
Cancel
Target: https://www.nus.edu.sg
HTTP/2
Request
Pretty Raw Hex \n
Response
Pretty Raw Hex Render \n
INSPECTOR
Request Attributes
Query Parameters (0)
Body Parameters (0)
Request Cookies (5)
Request Headers (20)
Response Headers (14)
Search...
0 matches
Search...
0 matches
Done
52,449 bytes | 37 millis

```
GET / HTTP/2
Host: www.nus.edu.sg
Cookie: visid_incap_1988262=
ktzlm5VYQBafshtqR0pVMXyIBWEAAAAAQUIPAAAAAADI10uASMn9I5GkTnPY
5qtU; _ga=GA1.3.1226650641.1627752574; nlbi_1988262=
h9zcTivSwQP3ueBe2S/eQQAAAADVsD8502Jk7bK8D72K09iC;
incap_ses_501_1988262=
M//NKfp+byZSMZ31PunzBsSCGmEAAAAA72C2142H2U46TEH38N2BdA==;
_gid=GA1.3.128484439.1629127366
Sec-Ch-Ua: " Not A;Brand";v="99", "Chromium";v="92"
Sec-Ch-Ua-Mobile: ?0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.131
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/
avif,image/webp,image/apng,*/*;q=0.8,application/signed-exch
ange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9


```

```
HTTP/2 200 OK
Cache-Control: public, max-age=120, s-maxage=120
Content-Type: text/html; charset=utf-8
Expires: Mon, 16 Aug 2021 16:26:30 GMT
Etag: "+K7zacufTjc80mg00fdcq3oka4ZLloSefhoBltDIr/o="
Vary: *
X-Frame-Options: SAMEORIGIN
X-Content-Type-Options: nosniff
X-Xss-Protection: 1;mode=block
Strict-Transport-Security: max-age=16070400; includeSubDom
Date: Mon, 16 Aug 2021 16:24:30 GMT
Set-Cookie: incap_ses_501_1988262=UX1DNoI4/QXaPq/1PunzBj6R
X-Cdn: Imperva
Upgrade-Insecure-Requests: 1
X-Iinfo: 10-82828877-82828878 NNNN CT(3 4 0) RT(1629131070


<!DOCTYPE html>
<html class="no-touch" xmlns="http://www.w3.org/1999/xhtml'
  <head id="Head1">
    <meta http-equiv="X-UA-Compatible" content="IE=edge" /
    <title>
      NUS - National University of Singapore
    </title>
    <meta http-equiv="Content-Type" content="text/html; ch
    <meta name="viewport" content="width=device-width, ini
    <meta name="keywords" content="nus singapore universit
    <meta name="rights" content="National University of Si
    <meta name="description" content="National University
    <meta name="google-site-verification" content="BKFQngs
    <link href="/Cwp/NUS/Html/base/images/favicon.ico" rel

    <script>
      (function(i, s, o, g, r, a, m) {
        i['GoogleAnalyticsObject'] = r;
        i[r] = i[r] || function() {
          (i[r].q = i[r].q || []).push(arguments)
```

Web Pen-Testing Tools: Common Features

- **Comparer** in Burp 2



Word compare of #3 and #4 (2 differences)

Length: 900

```
GET / HTTP/1.1
Host: www.nus.edu.sg
Cookie: visid_incap_1988262=ktzlm5VYQBafshtqROpVMXyIBWEAAAAAQUIPAAAAAADIl0uASMn9I5GkTnPY5qtU; _ga=GA1
Sec-Ch-Ua: " Not A;Brand";v="99", "Chromium";v="92"
Sec-Ch-Ua-Mobile: ?0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.13
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close
```

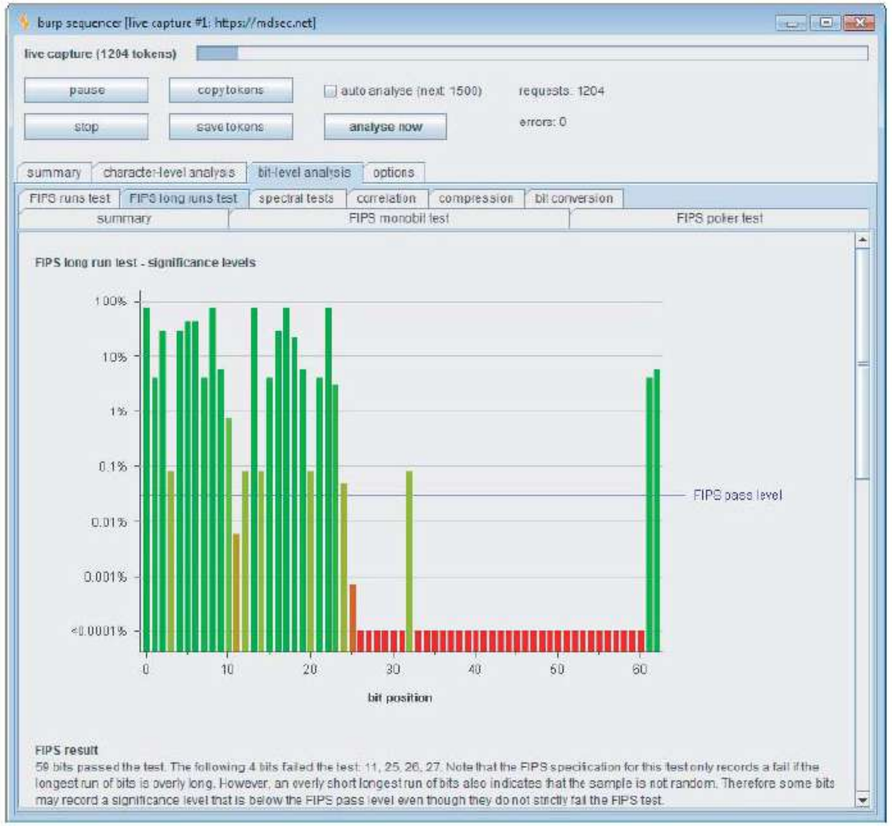
Length: 879

```
GET / HTTP/2
Host: www.nus.edu.sg
Cookie: visid_incap_1988262=ktzlm5VYQBafshtqROpVMXyIBWEAAAAAQUIPAAAAAADIl0uASMn9I5GkTnPY5qtU; _ga=GA1
Sec-Ch-Ua: " Not A;Brand";v="99", "Chromium";v="92"
Sec-Ch-Ua-Mobile: ?0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.13
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
```

Key: Modified Deleted Added

Web Pen-Testing Tools: Common Features

- Session cookie & other token **analyzer** (e.g. Burp's **Sequencer**)



live capture (1204 tokens)

□ auto analyse (next: 1500)

requests: 1204

errors: 0

FIPS result

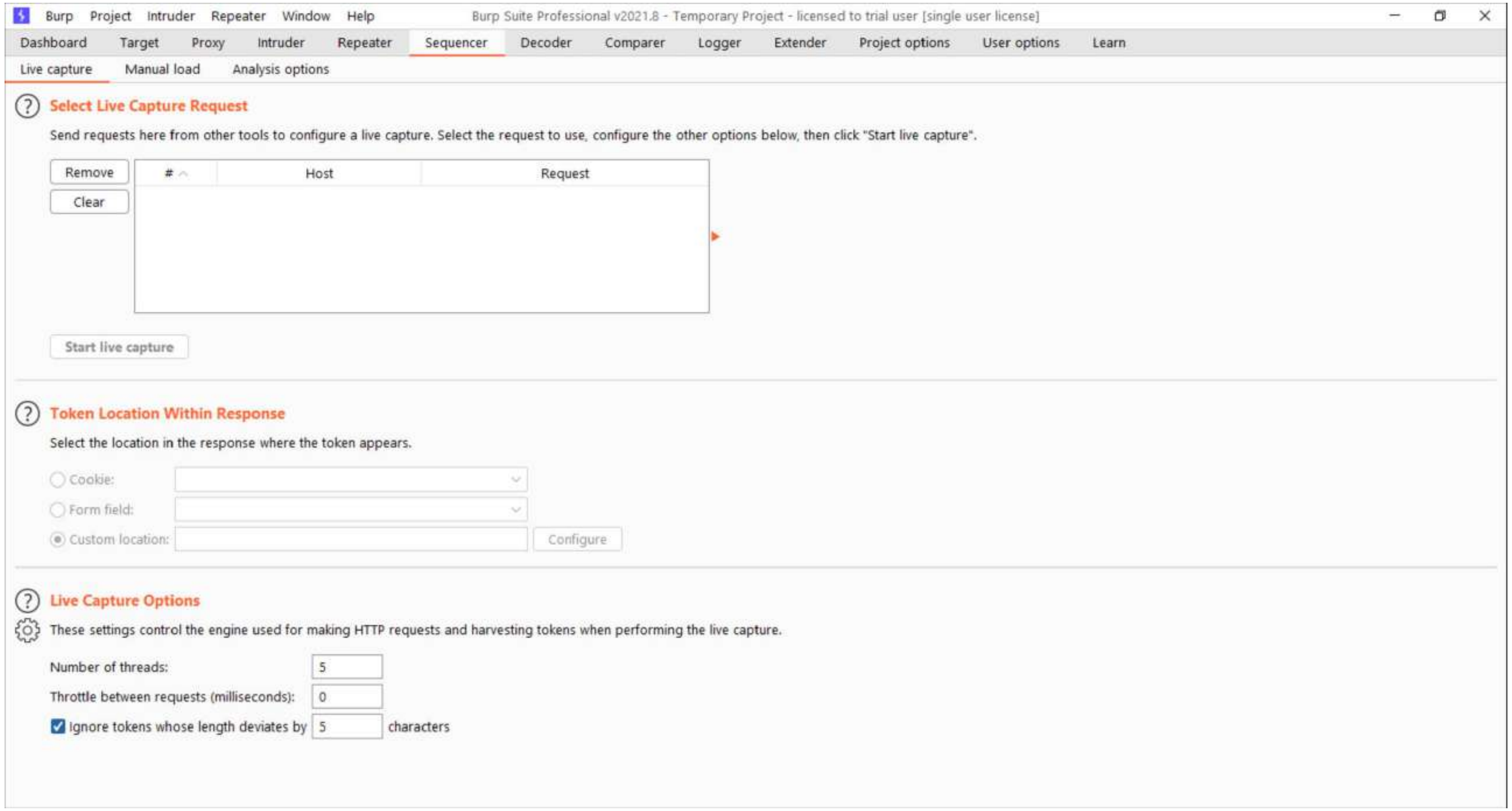
59 bits passed the test. The following 4 bits failed the test: 11, 25, 26, 27. Note that the FIPS specification for this test only records a fail if the longest run of bits is overly long. However, an overly short longest run of bits also indicates that the sample is not random. Therefore some bits may record a significance level that is below the FIPS pass level even though they do not strictly fail the FIPS test.

Figure 20-12: Using Burp Sequencer to test the randomness properties of an application's session token

From: Stuttard and Pinto, "The Web Application Hacker's Handbook"

Web Pen-Testing Tools: Common Features

- **Sequencer** in Burp 2



Select Live Capture Request

Send requests here from other tools to configure a live capture. Select the request to use, configure the other options below, then click "Start live capture".

Token Location Within Response

Select the location in the response where the token appears.

○ Cookie:

○ Form field:

● Custom location:

Live Capture Options

These settings control the engine used for making HTTP requests and harvesting tokens when performing the live capture.

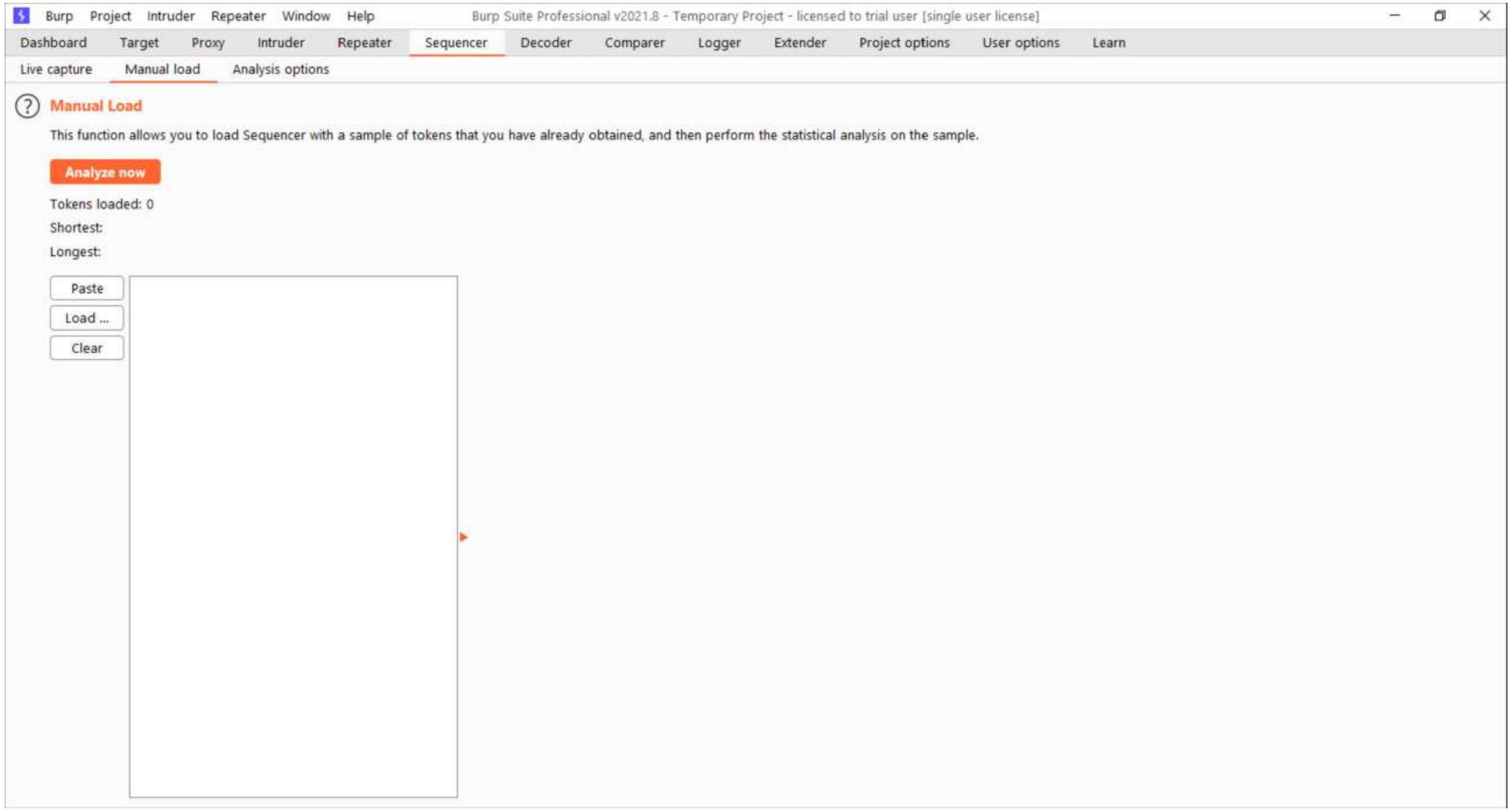
Number of threads: 5

Throttle between requests (milliseconds): 0

☑ Ignore tokens whose length deviates by 5 characters

Web Pen-Testing Tools: Common Features

- **Sequencer** in Burp 2


Burp Project Intruder Repeater Window Help
Burp Suite Professional v2021.8 - Temporary Project - licensed to trial user [single user license]
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn
Live capture Manual load Analysis options

Manual Load

This function allows you to load Sequencer with a sample of tokens that you have already obtained, and then perform the statistical analysis on the sample.

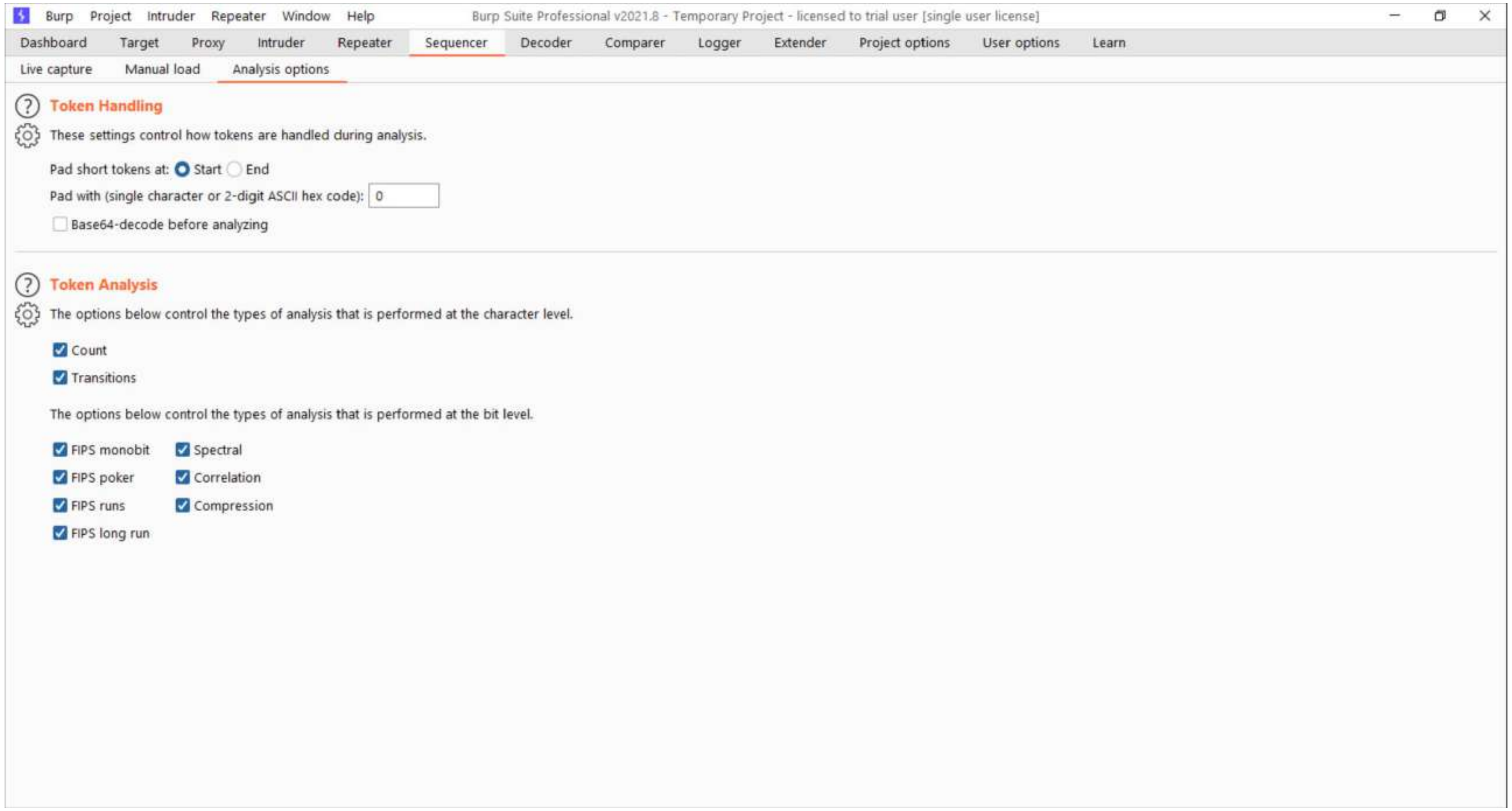
Tokens loaded: 0

Shortest:

Longest:

Web Pen-Testing Tools: Common Features

- **Sequencer** in Burp 2



Burp Project Intruder Repeater Window Help
Burp Suite Professional v2021.8 - Temporary Project - licensed to trial user [single user license]
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn
Live capture Manual load Analysis options

Token Handling

These settings control how tokens are handled during analysis.

Pad short tokens at: ● Start ○ End

Pad with (single character or 2-digit ASCII hex code): 0

□ Base64-decode before analyzing

Token Analysis

The options below control the types of analysis that is performed at the character level.

☑ Count

☑ Transitions

The options below control the types of analysis that is performed at the bit level.

☑ FIPS monobit ☑ Spectral

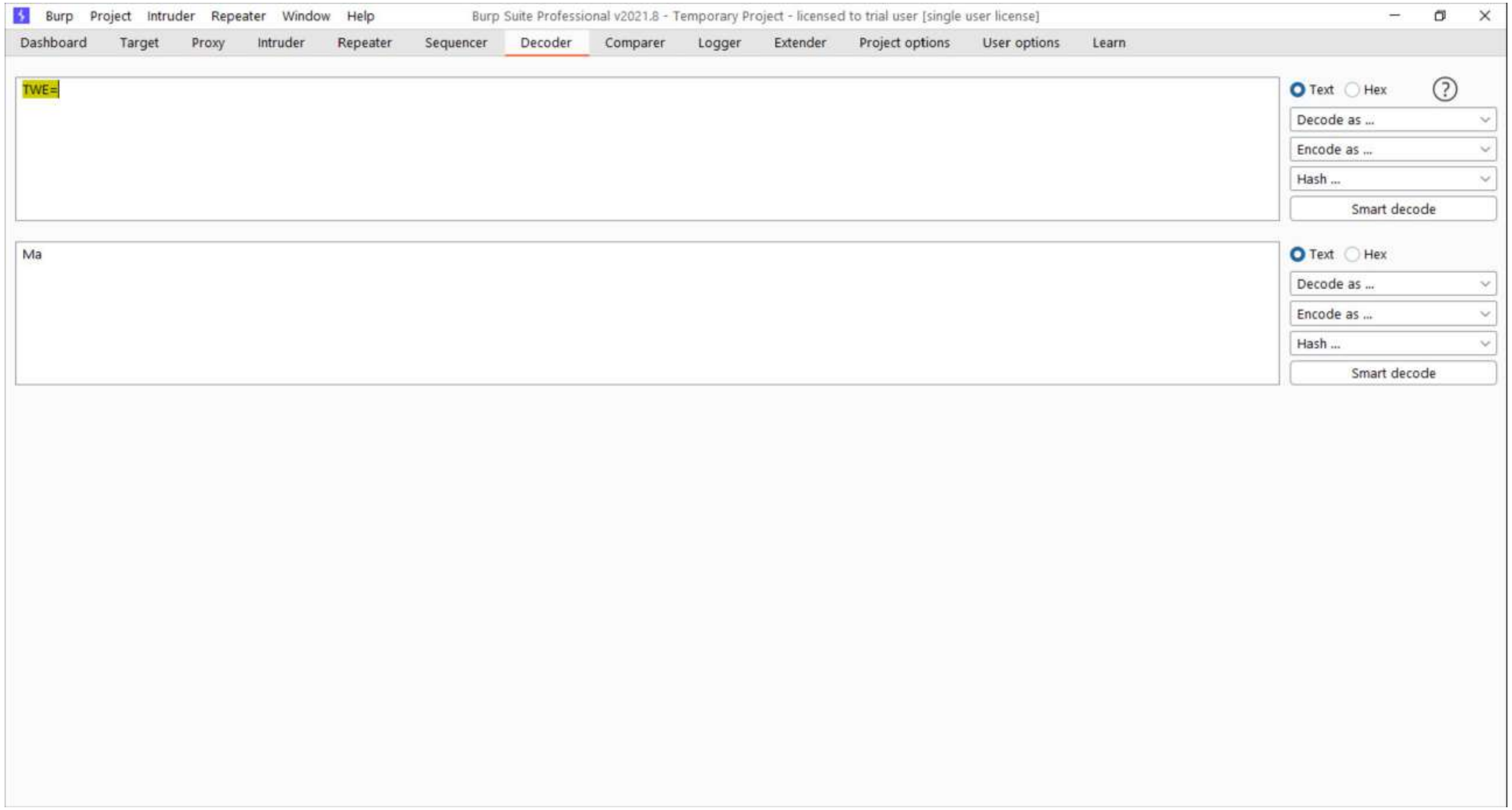
☑ FIPS poker ☑ Correlation

☑ FIPS runs ☑ Compression

☑ FIPS long run

Web Pen-Testing Tools: Common Features

- **Decoder** in Burp 2



Burp Project Intruder Repeater Window Help
Burp Suite Professional v2021.8 - Temporary Project - licensed to trial user [single user license]
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn

TWE=

● Text ○ Hex

Decode as ...

Encode as ...

Hash ...

Smart decode

Ma

● Text ○ Hex

Decode as ...

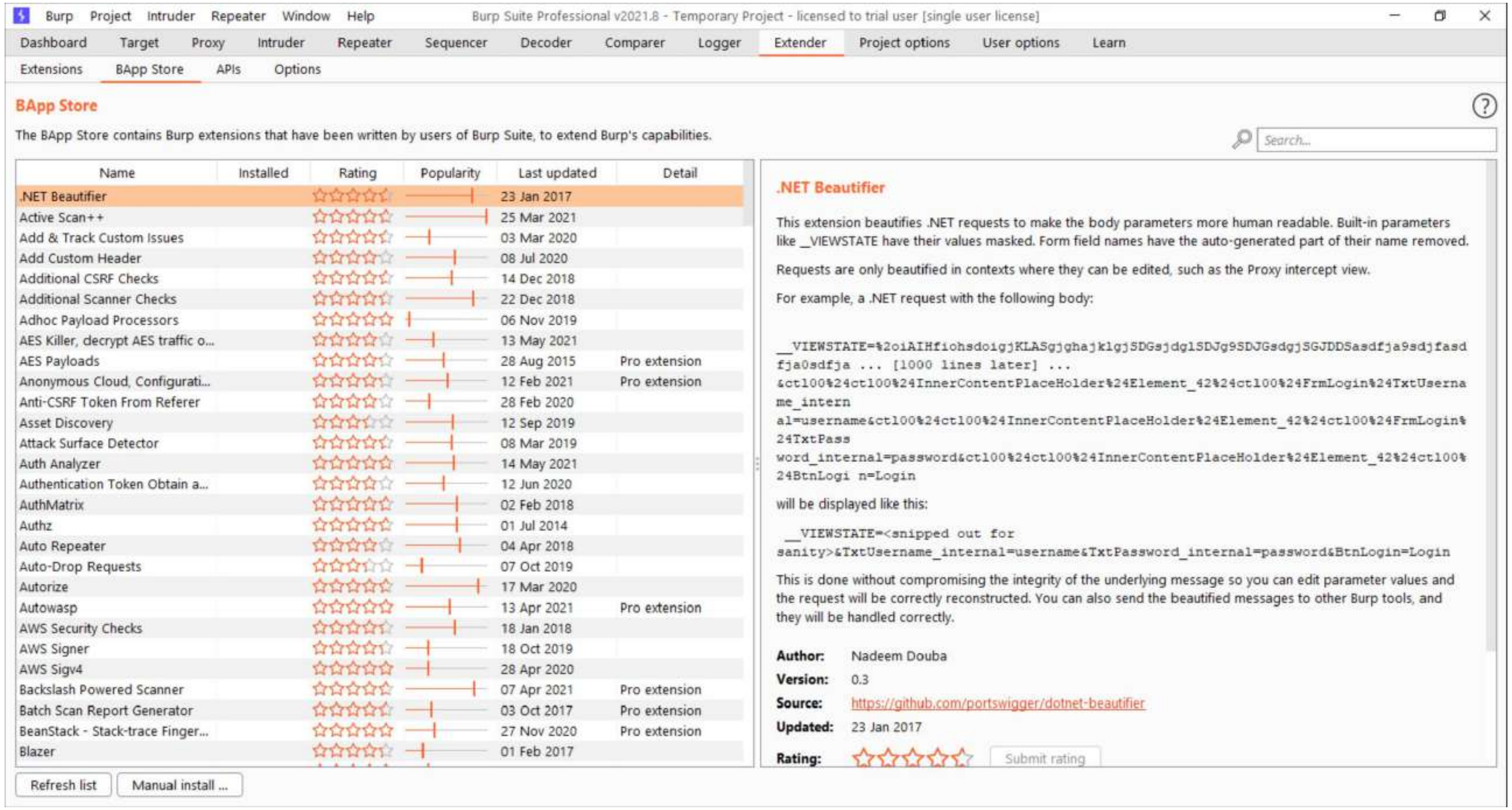
Encode as ...

Hash ...

Smart decode

Web Pen-Testing Tools: Common Features

- **Extender** in Burp 2



Burp Project Intruder Repeater Window Help
Burp Suite Professional v2021.8 - Temporary Project - licensed to trial user [single user license]
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn
Extensions BApp Store APIs Options
Search...
Refresh list
Manual install ...
Submit rating

BApp Store

The BApp Store contains Burp extensions that have been written by users of Burp Suite, to extend Burp's capabilities.

Name	Installed	Rating	Popularity	Last updated	Detail
.NET Beautifier				23 Jan 2017	
Active Scan++				25 Mar 2021	
Add & Track Custom Issues				03 Mar 2020	
Add Custom Header				08 Jul 2020	
Additional CSRF Checks				14 Dec 2018	
Additional Scanner Checks				22 Dec 2018	
Adhoc Payload Processors				06 Nov 2019	
AES Killer, decrypt AES traffic o...				13 May 2021	
AES Payloads				28 Aug 2015	Pro extension
Anonymous Cloud, Configurati...				12 Feb 2021	Pro extension
Anti-CSRF Token From Referer				28 Feb 2020	
Asset Discovery				12 Sep 2019	
Attack Surface Detector				08 Mar 2019	
Auth Analyzer				14 May 2021	
Authentication Token Obtain a...				12 Jun 2020	
AuthMatrix				02 Feb 2018	
Authz				01 Jul 2014	
Auto Repeater				04 Apr 2018	
Auto-Drop Requests				07 Oct 2019	
Autorize				17 Mar 2020	
Autowasp				13 Apr 2021	Pro extension
AWS Security Checks				18 Jan 2018	
AWS Signer				18 Oct 2019	
AWS Sigv4				28 Apr 2020	
Backslash Powered Scanner				07 Apr 2021	Pro extension
Batch Scan Report Generator				03 Oct 2017	Pro extension
BeanStack - Stack-trace Finger...				27 Nov 2020	Pro extension
Blazer				01 Feb 2017	

.NET Beautifier

This extension beautifies .NET requests to make the body parameters more human readable. Built-in parameters like __VIEWSTATE have their values masked. Form field names have the auto-generated part of their name removed.

Requests are only beautified in contexts where they can be edited, such as the Proxy intercept view.

For example, a .NET request with the following body:

```
__VIEWSTATE=%2oiAIHfiohsdoigjKLASgjghajklgjSDGsjdglSDJg9SDJGsdgjSGJDDSasdfja9sdjfasd
fja0sdfja ... [1000 lines later] ...
&ctl00%24ctl00%24InnerContentPlaceHolder%24Element_42%24ctl00%24FrmLogin%24TxtUserna
me_intern
al=username&ctl00%24ctl00%24InnerContentPlaceHolder%24Element_42%24ctl00%24FrmLogin%
24TxtPass
word_internal=password&ctl00%24ctl00%24InnerContentPlaceHolder%24Element_42%24ctl00%
24BtnLogi n=Login
```

will be displayed like this:

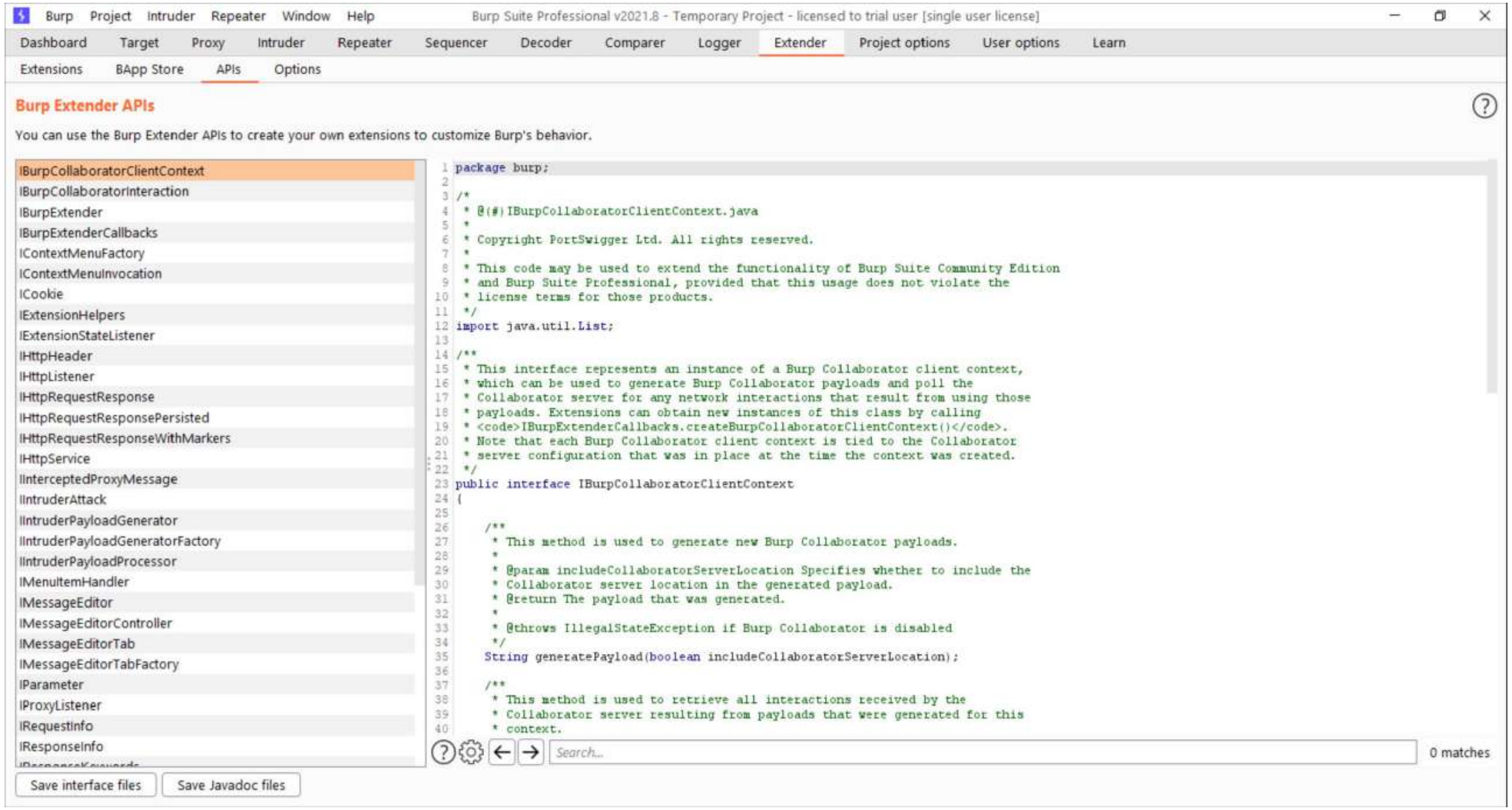
```
 __VIEWSTATE=<snipped out for
sanity>&TxtUsername_internal=username&TxtPassword_internal=password&BtnLogin=Login
```

This is done without compromising the integrity of the underlying message so you can edit parameter values and the request will be correctly reconstructed. You can also send the beautified messages to other Burp tools, and they will be handled correctly.

Author: Nadeem Douba
Version: 0.3
Source: https://github.com/portswigger/dotnet-beautifier
Updated: 23 Jan 2017
Rating:

Web Pen-Testing Tools: Common Features

- **Extender** in Burp 2



Burp Extender APIs

You can use the Burp Extender APIs to create your own extensions to customize Burp's behavior.

- IBurpCollaboratorClientContext
- IBurpCollaboratorInteraction
- IBurpExtender
- IBurpExtenderCallbacks
- IContextMenuFactory
- IContextMenuInvocation
- ICookie
- IExtensionHelpers
- IExtensionStateListener
- IHttpHeader
- IHttpListener
- IHttpRequestResponse
- IHttpRequestResponsePersisted
- IHttpRequestResponseWithMarkers
- IHttpService
- IInterceptedProxyMessage
- IIntruderAttack
- IIntruderPayloadGenerator
- IIntruderPayloadGeneratorFactory
- IIntruderPayloadProcessor
- IMenuItemHandler
- IMessageEditor
- IMessageEditorController
- IMessageEditorTab
- IMessageEditorTabFactory
- IParameter
- IProxyListener
- IRequestInfo
- IResponseInfo

```
package burp;

/*
 * @(#)IBurpCollaboratorClientContext.java
 *
 * Copyright PortSwigger Ltd. All rights reserved.
 *
 * This code may be used to extend the functionality of Burp Suite Community Edition
 * and Burp Suite Professional, provided that this usage does not violate the
 * license terms for those products.
 */
import java.util.List;

/**
 * This interface represents an instance of a Burp Collaborator client context,
 * which can be used to generate Burp Collaborator payloads and poll the
 * Collaborator server for any network interactions that result from using those
 * payloads. Extensions can obtain new instances of this class by calling
 * <code>IBurpExtenderCallbacks.createBurpCollaboratorClientContext()</code>.
 * Note that each Burp Collaborator client context is tied to the Collaborator
 * server configuration that was in place at the time the context was created.
 */
public interface IBurpCollaboratorClientContext
{

    /**
     * This method is used to generate new Burp Collaborator payloads.
     *
     * @param includeCollaboratorServerLocation Specifies whether to include the
     * Collaborator server location in the generated payload.
     * @return The payload that was generated.
     *
     * @throws IllegalStateException if Burp Collaborator is disabled
     */
    String generatePayload(boolean includeCollaboratorServerLocation);

    /**
     * This method is used to retrieve all interactions received by the
     * Collaborator server resulting from payloads that were generated for this
     * context.
```

Others Useful Web Pen-Testing Tools

- Other useful tools:
  - Nikto / Wikto
  - w3af
  - **Hydra: online password cracker**
  - **sqlmap: for SQL injection**
  - wget
  - curl
  - **nmap**
  - Numerous OWASP tools: (do check them!)
  - **Kali Linux**: a Linux distro
  - Samurai WTF: web penetration testing VM
  - (And don't forget:) browsers' developer tools
  - As well as sites like SecLists & PayloadsAllTheThings (for multiple types of lists), Webhook.site (to test any incoming HTTP request)
- List of web hacking tools: http://sectools.org/tag/web-scanners/

Web App Vulnerabilities

Web Vulnerability Classification

- Various older references on a **classification (e.g. taxonomy) of web vulnerabilities**
- An example:
  - “Web Application Security Frame”, Microsoft Corporation, 2005
  - See: https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff649461(v=pandp.10)
- **OWASP Web Security Testing Guide**:
  - Web oriented: unlike e.g. CWE (Common Weakness Enumeration), which lists (more general) software weaknesses types
  - Regularly **updated**
  - Much more **complete** than OWASP Top 10, of course
  - **Very detailed classification** of web vulnerabilities
  - Useful for your **naming & grouping** your found entries in your report!

OWASP WSTG for Web Pen-Testing

- Testing **philosophy/methodology**:
  - Pen-testing: the “art” of testing a running application **remotely** to find security vulnerabilities, without knowing the inner workings of the application itself
  - Typically, the penetration test team would have access to an application **as if they were users**
  - A gray box testing: it is assumed that the tester has **some partial knowledge** about the session management of the application
- Why **OWASP**?
  - Specific for **web-app** penetration testing
  - Unlike the **general penetration testing**: OS & network security

General Steps

- Information gathering
- Configuration & deployment management testing
- Identity management testing
- Authentication testing
- Authorization testing
- Session management testing
- Input validation testing
- Testing for error handling
- Testing for weak cryptography
- Business logic testing
- Client-side testing

General Steps

- **Information gathering**
- Configuration & deployment management testing
- Identity management testing
- Authentication testing
- Authorization testing
- Session management testing
- Input validation testing
- Testing for error handling
- Testing for weak cryptography
- Business logic testing
- Client-side testing

Testing for Information Gathering

- Conduct **search engine discovery** & **reconnaissance** for information leakage (WSTG-INFO-01)
- Fingerprint **web server** (WSTG-INFO-02)
- Review webserver **metafiles** for information leakage (WSTG-INFO-03)
- Enumerate **applications** on webserver (WSTG-INFO-04)
- Review **webpage content** for information leakage (WSTG-INFO-05)
- Identify application **entry points** (WSTG-INFO-06)
- Map **execution paths** through application (WSTG-INFO-07)
- Fingerprint **web application framework** (WSTG-INFO-08)
- Fingerprint **web application** (WSTG-INFO-09)
- Map **application architecture** (WSTG-INFO-10)

Conduct Search Engine Discovery/Reconnaissance for Information Leakage (WSTG-INFO-01)

- **Objective**: To understand what **sensitive design & configuration information** of the application/system/organization is exposed both directly (on the organization's website) or indirectly (on a third-party website)
- **How to test**: Use a **search engine** to search for:
  - Network diagrams & configurations
  - Archived posts & emails by administrators and other key staff
  - Log on procedures & username formats
  - Usernames & passwords
  - Error message content
  - Development, test, UAT & staging versions of the website

Conduct Search Engine Discovery/Reconnaissance for Information Leakage (WSTG-INFO-01)

- **Techniques**: Google hacking
  - Search operators
    - “site:”
    - “cache:”
  - Google Hacking Database
- **Tools**:
  - Google hacking, FoundStone SiteDigger, PunkSPIDER, …

Fingerprint Web Server (WSTG-INFO-02)

- **Objective**: To find the **version & type of a running web server** to determine known vulnerabilities & the appropriate exploits to use during testing
- **How to test** (black box testing):
  - Fingerprinting by observing server behaviour:
    - Issue a valid request & check HTTP "Server" header, HTTP header field ordering
    - Issue a malformed requests test & check the response
- **Tools**:
  - Manual: browser’s developer tool, nc, curl, wget, proxy tools
  - Automated: httprint - http://net-square.com/httprint.html
  - Online: Netcraft - http://www.netcraft.com
  - AutoRecon: https://github.com/Tib3rius/AutoRecon (for the server!)

Enumerate Applications on Webserver (WSTG-INFO-04)

- **Objective**: To enumerate the **applications** within scope that exist on a web server
- **How to test** (black box testing):
  - Different base URL
  - Non-standard ports:
    ```
    nmap –PN –sT -sV –p0-65535 192.168.1.100
    ```
  - Virtual hosts: DNS zone transfers, DNS inverse queries, Web-based DNS searches, Reverse-IP services

Identify Application Entry Points (WSTG-INFO-06)

- **Objective**: To understand how **requests are formed & typical responses** from the application
- **Analyse requests**:
  - Identify where **GETs** are used & where **POSTs** are used
  - Identify all **parameters used in a POST** request (these are in the body of the request)
  - Within the POST request, pay special attention to any **hidden parameters**
  - Identify **all parameters used in a GET** request (i.e., URL), in particular the query string (usually after a ? mark)
  - Identify **all the parameters of the query string**
  - …

Fingerprint Web Application Framework (WSTG-INFO-08)

- **Objective**: To define type of used **web framework** so as to have a better understanding of the security testing methodology
- **How to test** (Black Box testing):
  - Several most **common locations** to look in in order to define the current framework:
    - **HTTP headers**: X-Powered-By header
    - **Cookies**: CAKEPHP
    - **HTML source code**: for specific markers
    - **File extensions**: .php (PHP), .aspx (Microsoft ASP.NET), .jsp (Java Server pages)
    - **Specific files & folders**
    - **Error message**

General Steps

- Information gathering
- **Configuration & deployment management testing**
- Identity management testing
- Authentication testing
- Authorization testing
- Session management testing
- Input validation testing
- Testing for error handling
- Testing for weak cryptography
- Business logic testing
- Client-side testing

Testing for Configuration & Deployment Management

- Test network infrastructure configuration (WSTG-CONF-01)
- Test application platform configuration (WSTG-CONF-02)
- Test file extensions handling for sensitive information (WSTG-CONF-03)
- Review old backup and unreferenced files for sensitive information (WSTG-CONF-04)
- Enumerate infrastructure & application admin interfaces (WSTG-CONF-05)
- **Test HTTP methods (WSTG-CONF-06)**
- **Test HTTP Strict Transport Security (WSTG-CONF-07)**
- Test RIA cross domain policy (WSTG-CONF-08)
- **Test file permission (WSTG-CONF-09)**
- Test for subdomain takeover (WSTG-CONF-10)
- Test cloud storage (WSTG-CONF-11)

General Steps

- Information gathering
- Configuration & Deployment management testing
- Identity management testing
- Authentication testing
- Authorization testing
- Session management testing
- Input validation testing
- **Testing for error handling**
- Testing for weak cryptography
- Business logic testing
- Client-side testing

Testing for Error Handling

- Testing for improper error handling (WSTG-ERRH-01)
- Testing for stack traces (WSTG-ERRH-02)

General Steps

- Information gathering
- Configuration & Deployment management testing
- Identity management testing
- Authentication testing
- Authorization testing
- Session management testing
- Input validation testing
- Testing for error handling
- **Testing for weak cryptography**
- Business logic testing
- Client-side testing

Testing for Weak Cryptography

- Testing for **weak transport layer security** (WSTG-CRYP-01)
  - **Server configuration**
  - **Digital certificates**
  - **Implementation** vulnerabilities
  - **Application** vulnerabilities: **mixed active content**, …
- Testing for **padding oracle** (WSTG-CRYP-02)
- Testing for **sensitive information sent via unencrypted channels** (WSTG-CRYPST-03)
- Testing for weak encryption (WSTG-CRYP-04)

General Steps

- Information gathering
- Configuration & Deployment management testing
- **Identity management testing**
- Authentication testing
- Authorization testing
- Session management testing
- Input validation testing
- Testing for error handling
- Testing for weak cryptography
- Business logic testing
- Client-side testing

Testing Identity Management

- Test **role** definitions (WSTG-IDNT-01)
- Test **user registration** process (WSTG-IDNT-02)
- Test **account provisioning** process (WSTG-IDNT-03)
- Testing for **account enumeration** & **guessable user account** (WSTG-IDNT-04)
- Testing for **weak or unenforced username policy** (WSTG-IDNT-05)

General Steps

- Information gathering
- Configuration & Deployment management testing
- Identity management testing
- **Authentication testing**
- Authorization testing
- Session management testing
- Input validation testing
- Testing for error handling
- Testing for weak cryptography
- Business logic testing
- Client-side testing

Testing for Authentication

- Testing for **credentials transported over an encrypted channel** (WSTG-ATHN-01)
- Testing for **default credentials** (WSTG-ATHN-02)
- Testing for **weak lock out mechanism** (WSTG-ATHN-03)
- Testing for **bypassing authentication schema** (WSTG-ATHN-04):
  - Direct page request (**forced browsing**)
  - **Parameter modification**
  - **Session ID** prediction
  - **SQL injection**
- Testing for **vulnerable remember password** (WSTG-ATHN-05)

Testing for Authentication

- Testing for browser cache weakness (WSTG-ATHN-06)
- Testing for **weak password policy** (WSTG-ATHN-07)
- Testing for **weak security question answer** (WSTG-ATHN-08)
- Testing for **weak password change or reset functionalities** (WSTG-ATHN-09)
- Testing for weaker authentication **in alternative channel** (WSTG-ATHN-10)

General Steps

- Information gathering
- Configuration & Deployment management testing
- Identity management testing
- Authentication testing
- **Authorization testing**
- Session management testing
- Input validation testing
- Testing for error handling
- Testing for weak cryptography
- Business logic testing
- Client-side testing

Testing for Authorization

- Testing **directory traversal/file include** (WSTG-ATHZ-01)
  - http://example.com/getUserProfile.jsp?item=**../../../../etc/passwd**
  - Cookie: USER=1826cc8f:PSTYLE=**../../../../etc/passwd**
  - Possible character encoding mechanisms:
    - URL encoding and double URL encoding: **%2e%2e%2f** represents **../**
    - Unicode/UTF-8 Encoding (it only works in systems that are able to accept overlong UTF-8 sequences): **..%c0%af** represents **../**
- Testing **for bypassing authorization schema** (WSTG-ATHZ-02)
- Testing for **privilege escalation** (WSTG-ATHZ-03):
  - Manipulation of user group, user profile, condition value, IP Address

Testing for Authorization

- Testing for **Insecure Direct Object References** (WSTG-ATHZ-04):
  - **IDORs** occur when an application provides direct access to objects based on **user-supplied input**
  - Hence, attackers can bypass authorization & access resources in the system directly, for example database records or files
  - Some variations: The **value of a parameter** is used directly to
    - **Retrieve a database record**: http://foo.bar/somepage?invoice=**12345**
    - **Perform an operation** in the system: http://foo.bar/changepassword?user=**someuser**
    - **Retrieve a file system resource**: http://foo.bar/showImage?img=**img00011**
    - **Access application functionality**: http://foo.bar/accessPage?menuitem=**12**

General Steps

- Information gathering
- Configuration & Deployment management testing
- Identity management testing
- Authentication testing
- Authorization testing
- **Session management testing**
- Input validation testing
- Testing for error handling
- Testing for weak cryptography
- Business logic testing
- Client-side testing

Testing for Session Management

- Testing for **session management schema** (WSTG-SESS-01):
  - ***Cookies, cookies, cookies!***
  - Cookie **collection**
  - Session **analysis**: including **Session ID predictability & randomness**
  - Cookie **reverse engineering**
  - All **interaction** between the client & application should be tested at least against **the following criteria**:
    - Are all Set-Cookie directives tagged as **Secure**?
    - Do any Cookie operations take place over unencrypted transport?
    - Are any Cookies persistent?
    - What Expires= times are used on persistent cookies, and are they reasonable?
    - …

Testing for Session Management

- Testing for **cookies attributes** (WSTG-SESS-02):
  - Secure, HttpOnly, Domain, Path, Expires
- Testing for **session fixation** (WSTG-SESS-03)
- Testing for exposed session variables (WSTG-SESS-04)
- Testing for **CSRF** (WSTG-SESS-05)
- Testing for **logout functionality** (WSTG-SESS-06)
- Testing **session timeout** (WSTG-SESS-07)

Testing for Session Management

- Testing for **session puzzling** (WSTG-SESS-08)
  - ***Session puzzling*** = session variable overloading
  - Occurs when an app uses **the same session variable** for ***more than* one purpose**
  - An attacker can potentially access pages in an order **unanticipated** by the developers, so that the session variable is set in **one context & then used in another**
- Testing for **session hijacking** (WSTG-SESS-09)

General Steps

- Information gathering
- Configuration & Deployment management testing
- Identity management testing
- Authentication testing
- Authorization testing
- Session management testing
- **Input validation testing**
- Testing for error handling
- Testing for weak cryptography
- Business logic testing
- Client-side testing

Input Validation Testing

- An **important class** of web vulnerability!
- Due to unchecked/unsanitized **user-controlled inputs**
- Various data items **get contaminated** by the inputs
- The data items become **parts of code** executed by client's browser, web server, database server:

Contaminated Item	Target System/Component	Web Attack
Web page	Web browser	XSS
SQL query	Database server	SQL Injection
OS command	OS (shell)	Command Injection
XML elements	XML	XML/XPATH, XXE Injections
LDAP query	LDAP directory	LDAP Injection

Input Validation Testing

- Testing for **reflected Cross Site Scripting** (WSTG-INPV-01)
- Testing for **stored Cross Site Scripting** (WSTG-INPV-02)
- Testing for **HTTP verb tampering** (WSTG-INPV-03)
- Testing for **HTTP Parameter Pollution** (WSTG-INPV-04)
- Testing for **SQL Injection** (WSTG-INPV-05):
  Oracle, MySQL, SQL Server, PostgreSQL, MS Access,
  NoSQL injection, ORM injection, client-side
- Testing for **LDAP injection** (WSTG-INPV-06)
- Testing for **XML injection** (WSTG-INPV-07)

Input Validation Testing

- Testing for SSI injection (WSTG-INPV-08)
- Testing for **XPath injection** (WSTG-INPV-09)
- Testing for IMAP SMTP injection (WSTG-INPV-10)
- Testing for **code injection** (WSTG-INPV-11):
  Local File Inclusion (LFI), Remote File Inclusion (RFI)
- Testing for **command injection** (WSTG-INPV-12)
- Testing for format string injection (WSTG-INPV-13)
- Testing for incubated vulnerabilities (WSTG-INPV-14)

Input Validation Testing

- Testing for **HTTP splitting smuggling** (WSTG-INPV-15)
- Testing for **HTTP incoming requests** (WSTG-INPV-16)
- Testing for **host header injection** (WSTG-INPV-17)
- Testing for **server-side template injection** (WSTG-INPV-18)
- Testing for **Server-Side Request Forgery** (WSTG-INPV-19)

General Steps

- Information gathering
- Configuration & Deployment management testing
- Identity management testing
- Authentication testing
- Authorization testing
- Session management testing
- Input validation testing
- Testing for error handling
- Testing for weak cryptography
- **Business logic testing**
- Client-side testing

Business Logic Testing

- Test business **logic data validation** (WSTG-BUSL-01)
- Test ability to **forge requests** (WSTG-BUSL-02)
- Test **integrity checks** (WSTG-BUSL-03)
- Test for process timing (WSTG-BUSL-04)
- Test number of times a function can be used limits (WSTG-BUSL-05)
- Testing for the circumvention of work flows (WSTG-BUSL-06)
- Test defenses against application misuse (WSTG-BUSL-07)
- Test upload of **unexpected file types** (WSTG-BUSL-08)
- Test upload of **malicious files** (WSTG-BUSL-09)

General Steps

- Information gathering
- Configuration & Deployment management testing
- Identity management testing
- Authentication testing
- Authorization testing
- Session management testing
- Input validation testing
- Testing for error handling
- Testing for weak cryptography
- Business logic testing
- **Client-side testing**

Client-Side Testing

- Testing for **DOM based Cross Site Scripting** (WSTG-CLNT-01)
- Testing for **JavaScript execution** (WSTG-CLNT-02)
- Testing for HTML injection (WSTG-CLNT-03)
- Testing for **client side URL redirect** (WSTG-CLNT-04)
- Testing for **CSS injection** (WSTG-CLNT-05)
- Testing for client-side resource manipulation (WSTG-CLNT-06)
- Testing **Cross Origin Resource Sharing** (WSTG-CLNT-07)
- Testing for Cross Site Flashing (WSTG-CLNT-08)

Client-Side Testing

- Testing for **clickjacking** (WSTG-CLNT-09)
- Testing WebSockets (WSTG-CLNT-10)
- Testing web messaging (WSTG-CLNT-11)
- Testing **browser storage** (WSTG-CLNT-12)
- Testing for Cross Site Script Inclusion (WSTG-CLNT-13)

Secure Web Development & OWASP ASVS

Secure Web Development Guideline

- From **web developer's viewpoint**:
  - A large attack surface to defend
  - Various web attacks to prevent
- Any requirement/verification standard for web applications?
- **OWASP [Application Security Verification Standard (ASVS)]**?
  - It provides web developers with a *list of requirements* for secure development
  - [OWAS ASVS Version - 4.0.3] is available

OWASP ASVS

- *What is OWASP ASVS in short?*
- Two main **goals**:
  - To help organizations **develop** & **maintain** secure applications
  - To allow security service vendors, security tools vendors & consumers to align their **requirements** & **offerings**
- It catalogs **security requirements** & **verification criteria**:
  a source of detailed security requirements for development teams

Other Uses for the ASVS

- Aside from being used to **assess the security** of an application, ***other potential uses*** for the ASVS:
  - As detailed **security architecture guidance**
  - As a replacement for off-the-shelf **secure coding checklists**
  - As a guide for automated unit & integration **tests**
  - For secure development **training**
  - As a driver for **agile** application security
  - As a framework for guiding the **procurement** of secure software

Application Security Verification Levels

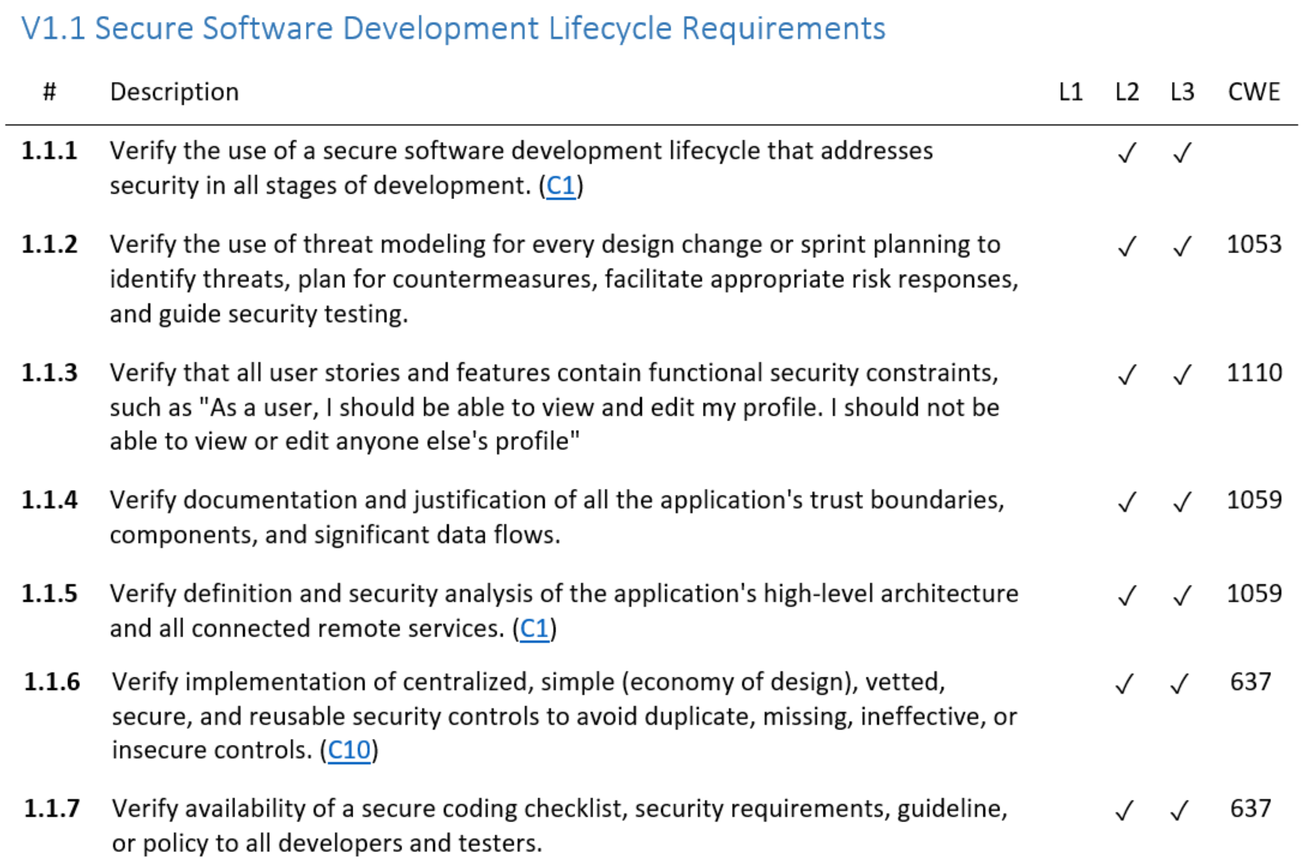
- ASVS defines 3 ***security verification levels***, with each level increasing in depth:
  - ASVS **Level 1 (L1)**: For **low** assurance levels, and is completely penetration testable
  - ASVS **Level 2 (L2)**: For apps that contain **sensitive data**, which requires protection, and is the *recommended* level for most apps
  - ASVS **Level 3 (L3)**: For the most **critical** apps (apps that perform high value transactions, contain sensitive medical data), or any application that requires the *highest level* of trust

List of Requirements

- V1: Architecture, design and threat modeling requirements
- V2: Authentication verification requirements
- V3: Session management verification requirements
- V4: Access control verification requirements
- V5: Validation, sanitization and encoding verification requirements
- V6: Stored cryptography verification requirements
- V7: Error handling and logging verification requirements
- V8: Data protection verification requirements
- V9: Communications verification requirements
- V10: Malicious code verification requirements
- V11: Business logic verification requirements
- V12: File and resources verification requirements
- V13: API and web service verification requirements
- V14: Configuration verification requirements

Sample Requirements

V1: Architecture, Design and Threat Modeling Requirements



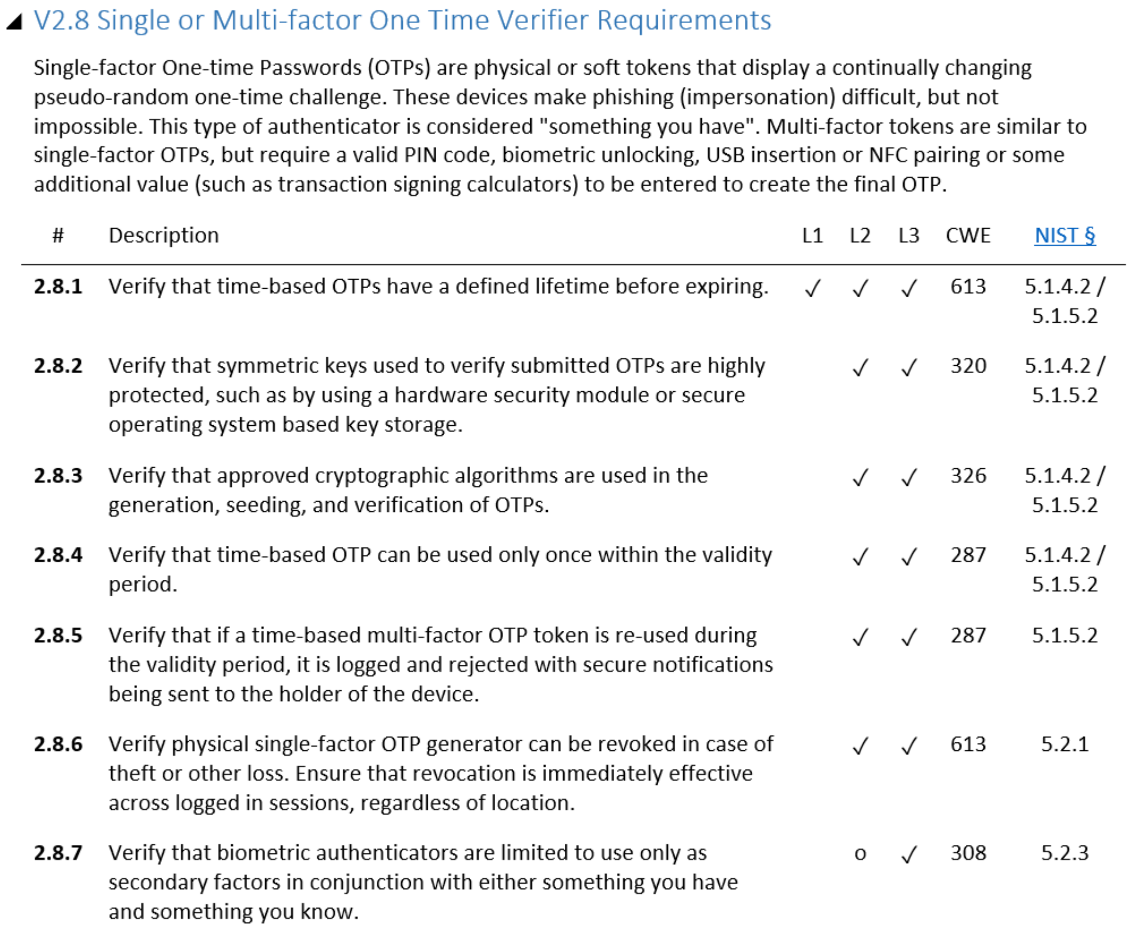
V1.1 Secure Software Development Lifecycle Requirements

#	Description	L1	L2	L3	CWE
1.1.1	Verify the use of a secure software development lifecycle that addresses security in all stages of development. (C1)		✓	✓	
1.1.2	Verify the use of threat modeling for every design change or sprint planning to identify threats, plan for countermeasures, facilitate appropriate risk responses, and guide security testing.		✓	✓	1053
1.1.3	Verify that all user stories and features contain functional security constraints, such as "As a user, I should be able to view and edit my profile. I should not be able to view or edit anyone else's profile"		✓	✓	1110
1.1.4	Verify documentation and justification of all the application's trust boundaries, components, and significant data flows.		✓	✓	1059
1.1.5	Verify definition and security analysis of the application's high-level architecture and all connected remote services. (C1)		✓	✓	1059
1.1.6	Verify implementation of centralized, simple (economy of design), vetted, secure, and reusable security controls to avoid duplicate, missing, ineffective, or insecure controls. (C10)		✓	✓	637
1.1.7	Verify availability of a secure coding checklist, security requirements, guideline, or policy to all developers and testers.		✓	✓	637

Source: OWASP
ASVS 4.0.2

Sample Requirements

V2: Authentication Verification Requirements



V2.8 Single or Multi-factor One Time Verifier Requirements

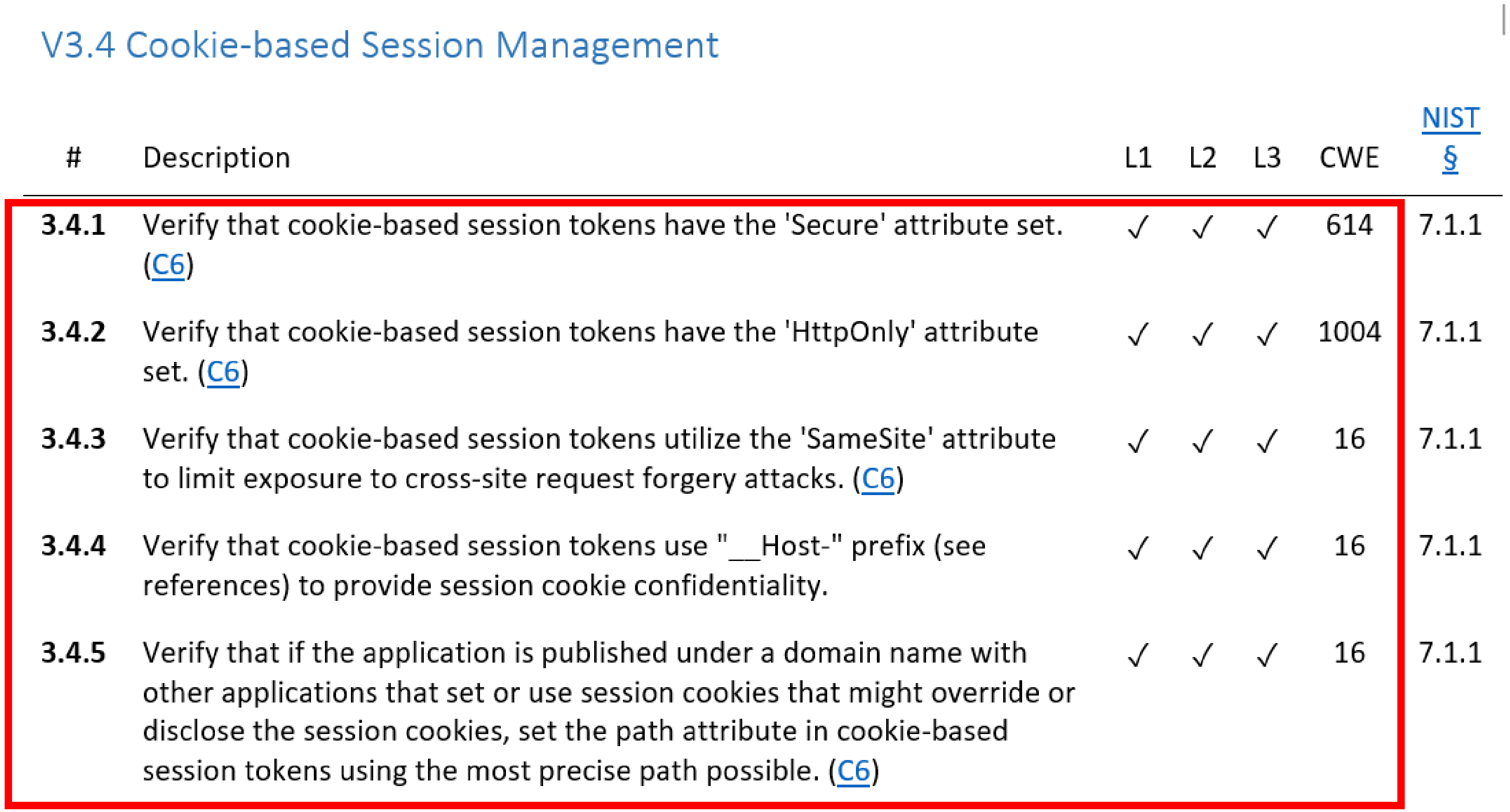
Single-factor One-time Passwords (OTPs) are physical or soft tokens that display a continually changing pseudo-random one-time challenge. These devices make phishing (impersonation) difficult, but not impossible. This type of authenticator is considered "something you have". Multi-factor tokens are similar to single-factor OTPs, but require a valid PIN code, biometric unlocking, USB insertion or NFC pairing or some additional value (such as transaction signing calculators) to be entered to create the final OTP.

#	Description	L1	L2	L3	CWE	NIST §
2.8.1	Verify that time-based OTPs have a defined lifetime before expiring.	✓	✓	✓	613	5.1.4.2 / 5.1.5.2
2.8.2	Verify that symmetric keys used to verify submitted OTPs are highly protected, such as by using a hardware security module or secure operating system based key storage.		✓	✓	320	5.1.4.2 / 5.1.5.2
2.8.3	Verify that approved cryptographic algorithms are used in the generation, seeding, and verification of OTPs.		✓	✓	326	5.1.4.2 / 5.1.5.2
2.8.4	Verify that time-based OTP can be used only once within the validity period.		✓	✓	287	5.1.4.2 / 5.1.5.2
2.8.5	Verify that if a time-based multi-factor OTP token is re-used during the validity period, it is logged and rejected with secure notifications being sent to the holder of the device.		✓	✓	287	5.1.5.2
2.8.6	Verify physical single-factor OTP generator can be revoked in case of theft or other loss. Ensure that revocation is immediately effective across logged in sessions, regardless of location.		✓	✓	613	5.2.1
2.8.7	Verify that biometric authenticators are limited to use only as secondary factors in conjunction with either something you have and something you know.		o	✓	308	5.2.3

Source: OWASP ASVS 4.0.2

Sample Requirements

V3: Session Management Verification Requirements



V3.4 Cookie-based Session Management

#	Description	L1	L2	L3	CWE	NIST §
3.4.1	Verify that cookie-based session tokens have the 'Secure' attribute set. (C6)	✓	✓	✓	614	7.1.1
3.4.2	Verify that cookie-based session tokens have the 'HttpOnly' attribute set. (C6)	✓	✓	✓	1004	7.1.1
3.4.3	Verify that cookie-based session tokens utilize the 'SameSite' attribute to limit exposure to cross-site request forgery attacks. (C6)	✓	✓	✓	16	7.1.1
3.4.4	Verify that cookie-based session tokens use "__Host-" prefix (see references) to provide session cookie confidentiality.	✓	✓	✓	16	7.1.1
3.4.5	Verify that if the application is published under a domain name with other applications that set or use session cookies that might override or disclose the session cookies, set the path attribute in cookie-based session tokens using the most precise path possible. (C6)	✓	✓	✓	16	7.1.1

Source: OWASP ASVS 4.0.2

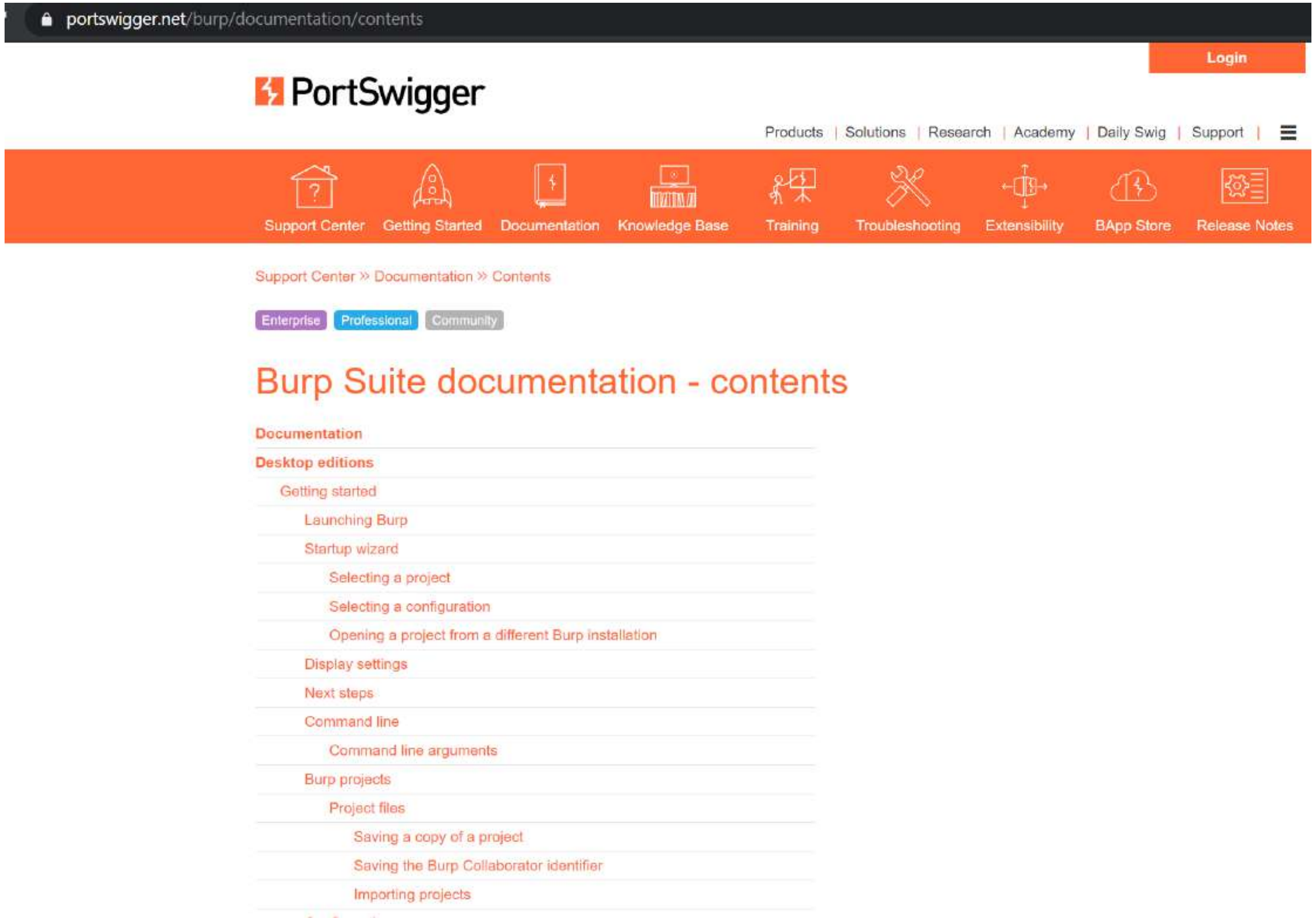
Useful Resources

Shared Past Lecture Materials

- **TIC4304 slides**:
  - **Web basics review**
  - **Implementation & deployment** issues & defenses: Heartbleed, Shellshock, DoS attacks, **Web Application Firewall (WAF)**
- **CS5331 slides** & the cited **resources**: for **web attacks**
- **Dinner of Web Security**: check the **referred links** if needed

Burp Suite Resources

- **Documentation**: https://portswigger.net/burp/documentation/contents



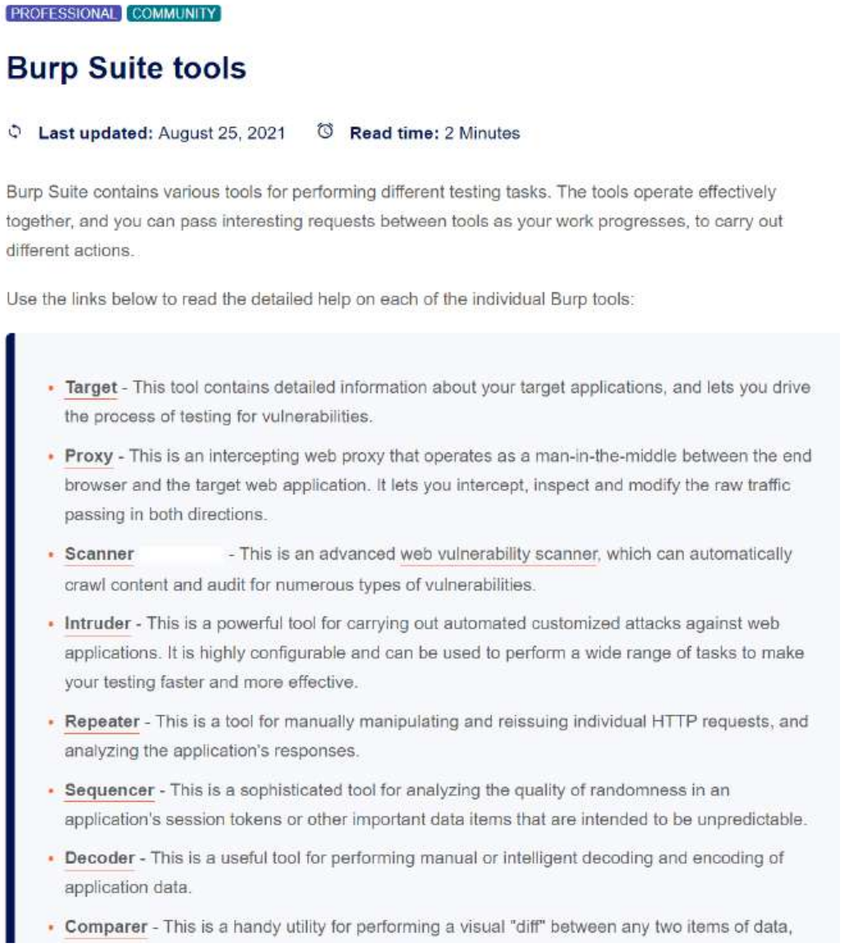
portswigger.net/burp/documentation/contents

Support Center » Documentation » Contents

Burp Suite documentation - contents

Burp Suite Resources

- **Documentation** on **tools/components**: https://portswigger.net/burp/documentation/desktop/tools



PROFESSIONAL COMMUNITY

Burp Suite tools

Last updated: August 25, 2021 **Read time:** 2 Minutes

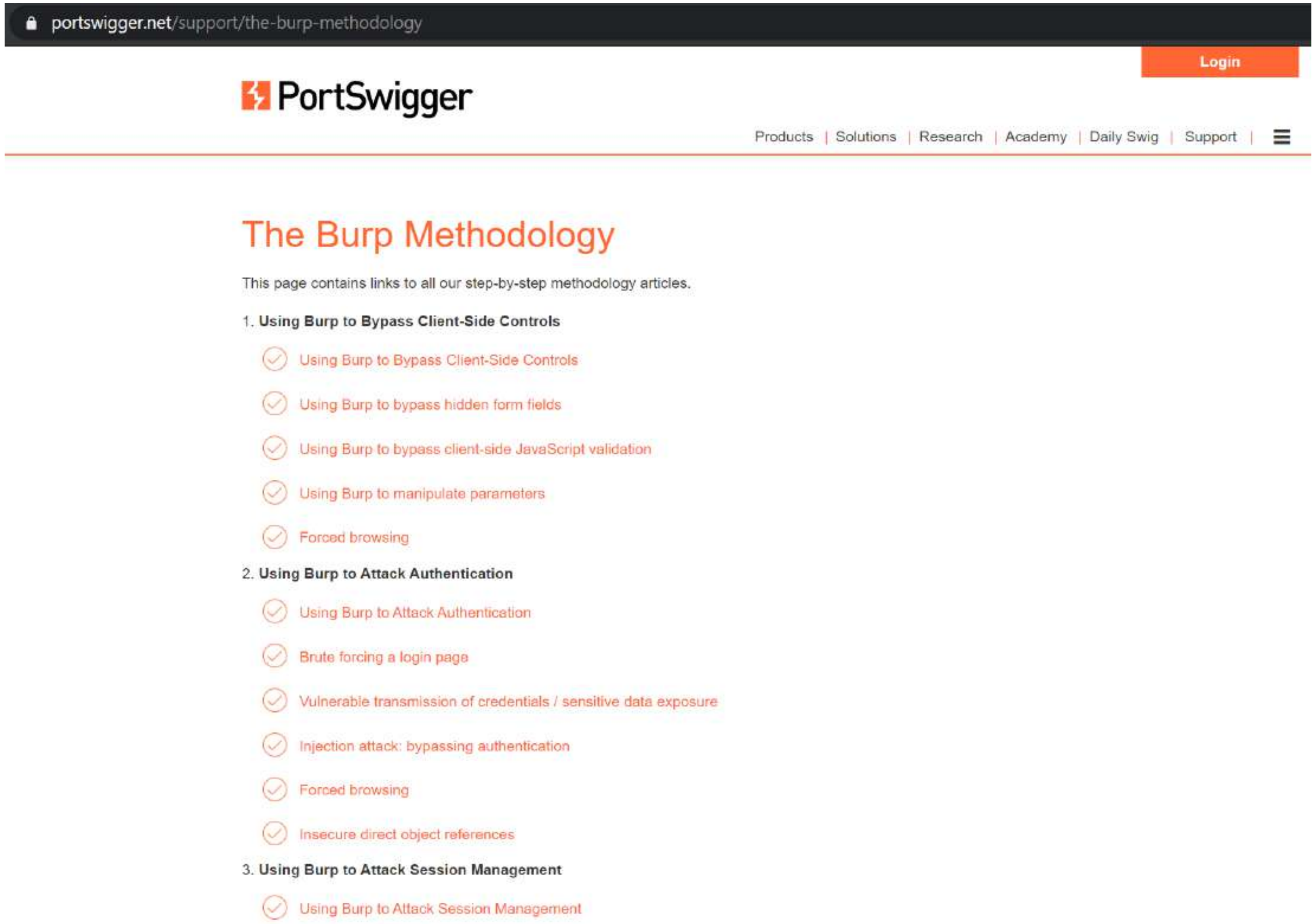
Burp Suite contains various tools for performing different testing tasks. The tools operate effectively together, and you can pass interesting requests between tools as your work progresses, to carry out different actions.

Use the links below to read the detailed help on each of the individual Burp tools:

- **Target** - This tool contains detailed information about your target applications, and lets you drive the process of testing for vulnerabilities.
- **Proxy** - This is an intercepting web proxy that operates as a man-in-the-middle between the end browser and the target web application. It lets you intercept, inspect and modify the raw traffic passing in both directions.
- **Scanner** - This is an advanced web vulnerability scanner, which can automatically crawl content and audit for numerous types of vulnerabilities.
- **Intruder** - This is a powerful tool for carrying out automated customized attacks against web applications. It is highly configurable and can be used to perform a wide range of tasks to make your testing faster and more effective.
- **Repeater** - This is a tool for manually manipulating and reissuing individual HTTP requests, and analyzing the application's responses.
- **Sequencer** - This is a sophisticated tool for analyzing the quality of randomness in an application's session tokens or other important data items that are intended to be unpredictable.
- **Decoder** - This is a useful tool for performing manual or intelligent decoding and encoding of application data.
- **Comparer** - This is a handy utility for performing a visual "diff" between any two items of data,

Burp Suite Resources

- **Methodology**: https://portswigger.net/support/the-burp-methodology


portswigger.net/support/the-burp-methodology
Login
PortSwigger
Products | Solutions | Research | Academy | Daily Swig | Support |

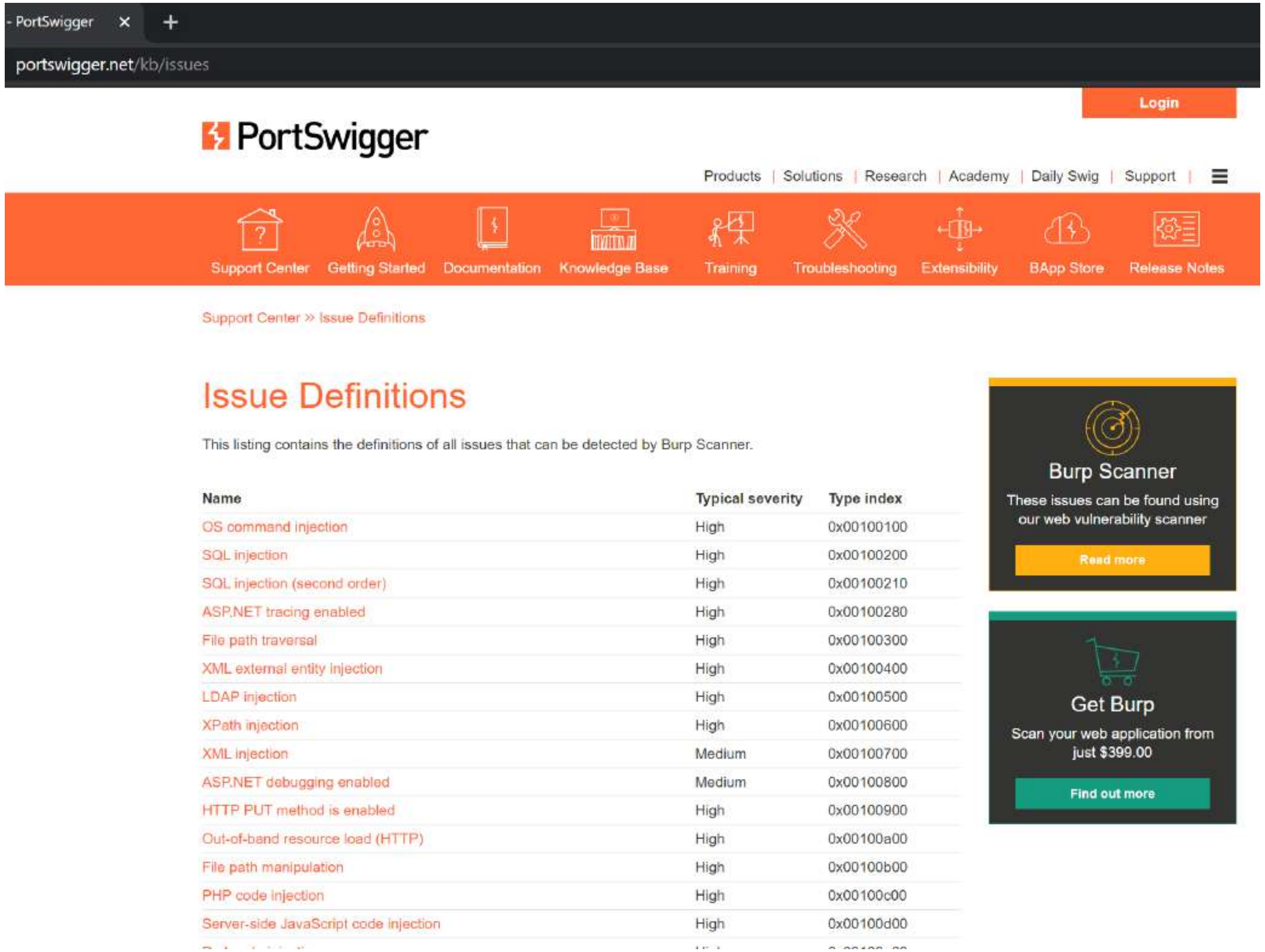
The Burp Methodology

This page contains links to all our step-by-step methodology articles.

1. **Using Burp to Bypass Client-Side Controls**
   - Using Burp to Bypass Client-Side Controls
   - Using Burp to bypass hidden form fields
   - Using Burp to bypass client-side JavaScript validation
   - Using Burp to manipulate parameters
   - Forced browsing
2. **Using Burp to Attack Authentication**
   - Using Burp to Attack Authentication
   - Brute forcing a login page
   - Vulnerable transmission of credentials / sensitive data exposure
   - Injection attack: bypassing authentication
   - Forced browsing
   - Insecure direct object references
3. **Using Burp to Attack Session Management**
   - Using Burp to Attack Session Management

Burp Suite Resources

- **Issue Definitions**: https://portswigger.net/kb/issues



Issue Definitions

This listing contains the definitions of all issues that can be detected by Burp Scanner.

Name	Typical severity	Type index
OS command injection	High	0x00100100
SQL injection	High	0x00100200
SQL injection (second order)	High	0x00100210
ASP.NET tracing enabled	High	0x00100280
File path traversal	High	0x00100300
XML external entity injection	High	0x00100400
LDAP injection	High	0x00100500
XPath injection	High	0x00100600
XML injection	Medium	0x00100700
ASP.NET debugging enabled	Medium	0x00100800
HTTP PUT method is enabled	High	0x00100900
Out-of-band resource load (HTTP)	High	0x00100a00
File path manipulation	High	0x00100b00
PHP code injection	High	0x00100c00
Server-side JavaScript code injection	High	0x00100d00

Burp Scanner

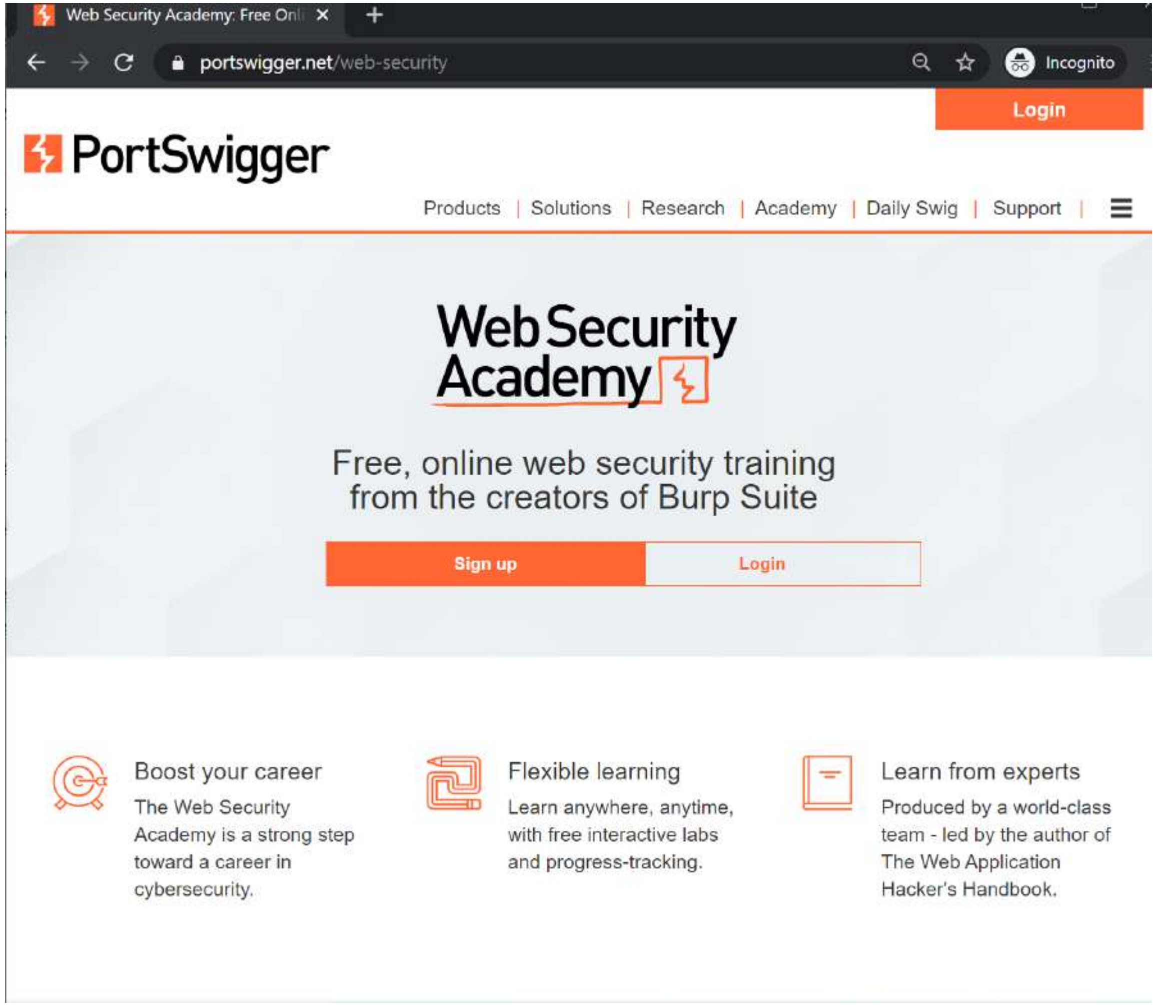
These issues can be found using our web vulnerability scanner

Get Burp

Scan your web application from just $399.00

Burp Suite Resources

- **Web Security Academy**: https://portswigger.net/web-security



Web Security Academy: Free Onli
portswigger.net/web-security
Incognito
Login
PortSwigger
Products | Solutions | Research | Academy | Daily Swig | Support
Web Security Academy
Sign up
Login

Free, online web security training from the creators of Burp Suite

Boost your career

The Web Security Academy is a strong step toward a career in cybersecurity.

Flexible learning

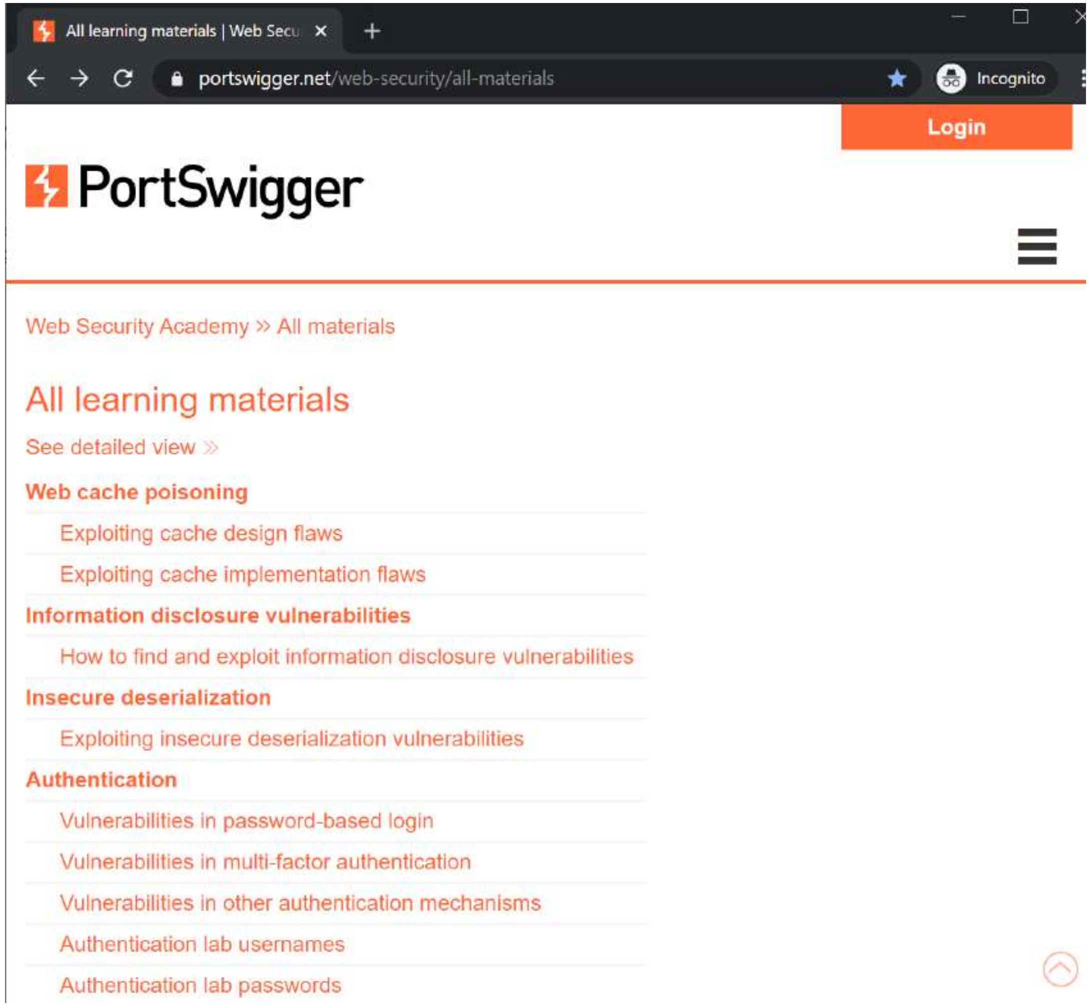
Learn anywhere, anytime, with free interactive labs and progress-tracking.

Learn from experts

Produced by a world-class team - led by the author of The Web Application Hacker's Handbook.

Burp Suite Resources

- **Web Security Academy**: https://portswigger.net/web-security/all-materials



All learning materials | Web Secu
portswigger.net/web-security/all-materials
Incognito
Login
PortSwigger

Web Security Academy » All materials

All learning materials

See detailed view »

Web cache poisoning

- Exploiting cache design flaws
- Exploiting cache implementation flaws

Information disclosure vulnerabilities

- How to find and exploit information disclosure vulnerabilities

Insecure deserialization

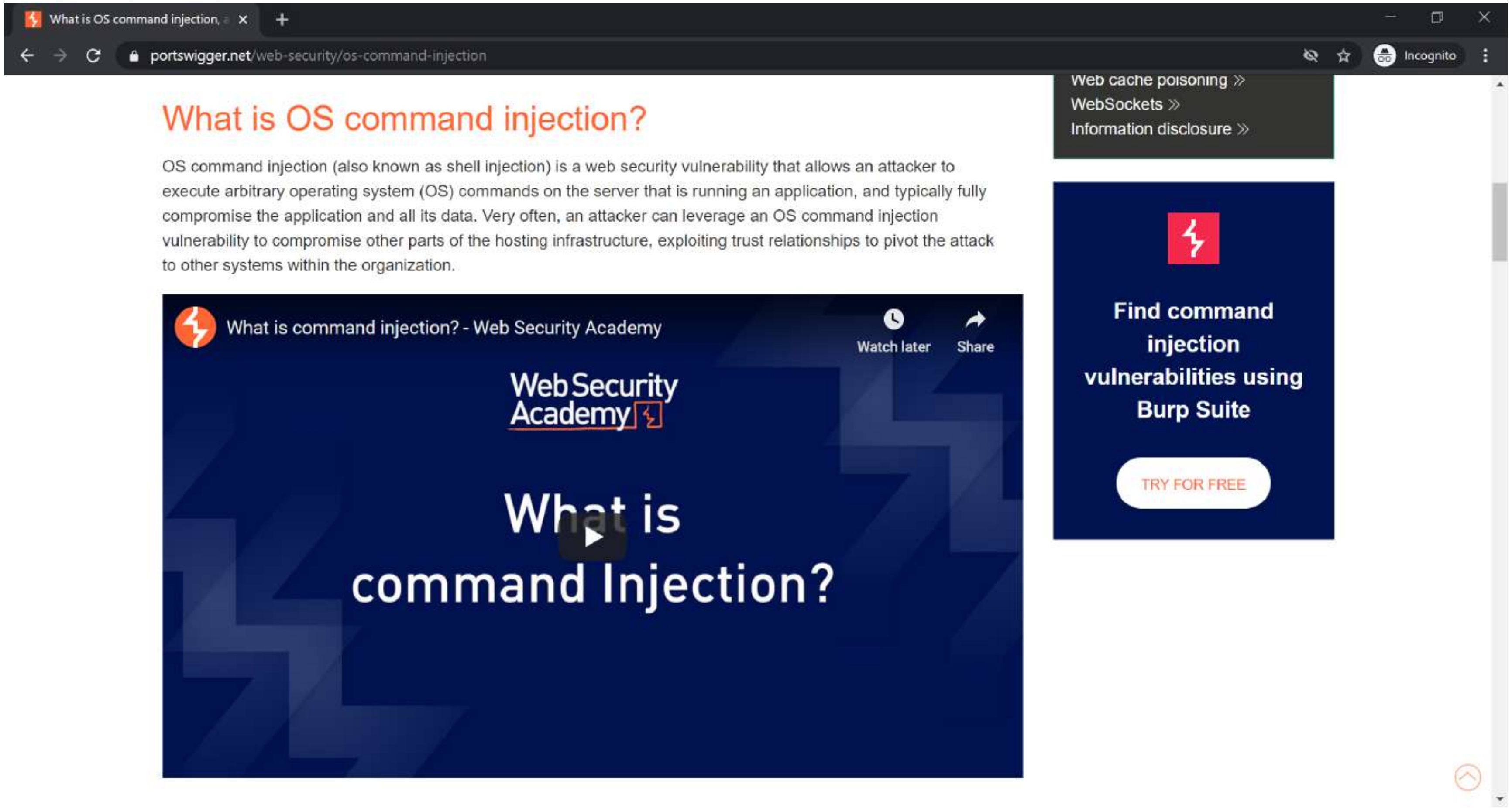
- Exploiting insecure deserialization vulnerabilities

Authentication

- Vulnerabilities in password-based login
- Vulnerabilities in multi-factor authentication
- Vulnerabilities in other authentication mechanisms
- Authentication lab usernames
- Authentication lab passwords

Burp Suite Resources

- Explanation, videos, labs



What is OS command injection, a ×
portswigger.net/web-security/os-command-injection
Incognito
Web cache poisoning »
WebSockets »
Information disclosure »

What is OS command injection?

OS command injection (also known as shell injection) is a web security vulnerability that allows an attacker to execute arbitrary operating system (OS) commands on the server that is running an application, and typically fully compromise the application and all its data. Very often, an attacker can leverage an OS command injection vulnerability to compromise other parts of the hosting infrastructure, exploiting trust relationships to pivot the attack to other systems within the organization.

What is command injection? - Web Security Academy
Watch later
Share
Web Security Academy
What is command Injection?
Find command injection vulnerabilities using Burp Suite
TRY FOR FREE

Web Pen-Testing References & Resources

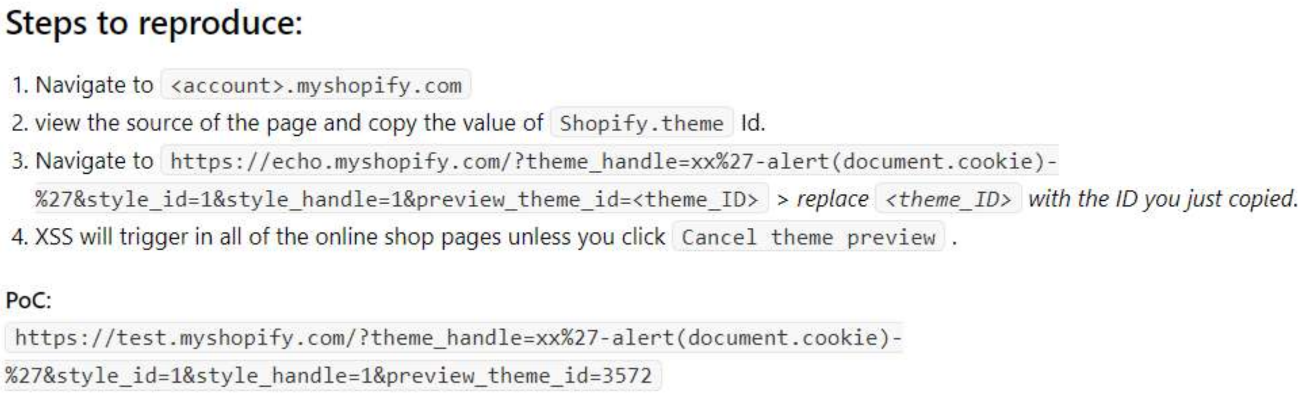
- Most common **web vulnerabilities**:
  - OWASP Top 10
  - The HackerOne Top 10 Most Impactful & Rewarded Vulnerability Types
- **Web attacks & pen-testing**:
  - OWASP Web Security Testing Guide v 4.2
  - Manh Pham Tien's "*Web Application Penetration Testing*"
  - OWASP Cheat Sheet Series
  - SecLists, PayloadsAllTheThings: many lists (fuzzing payloads, web shells, ...)
  - Hacker101.com: Videos on web attacks (https://www.hacker101.com/videos)
- **Burp Suite**:
  - Sunny Wear, "*Burp Suite Cookbook: Practical recipes to help you master web penetration testing with Burp Suite*", Packt Publishing, 2018
  - Hacker101.com: Video lessons on using Burp Suite (https://www.hacker101.com/playlists/burp_suite)

References for Your Practice Cases

- Web attack **case studies**:
  - **Broken web app**:
    Björn Kimminich, "*Pwning OWASP Juice Shop*",
    https://leanpub.com/juice-shop
  - **Real web apps**:
    - Exploitdb: https://www.exploit-db.com/
      under "WebApps" vulnerability type
    - CVE: https://cve.mitre.org/cve/
    - **HackerOne Bug Reports**: *for your sharing presentation later!*
    - …

HackerOne's Bug Reports: An Example

- **Reflected XSS** in <any>.myshopify.com through theme preview:
  - Disclosed on 30 May, 2017
  - https://hackerone.com/reports/226428

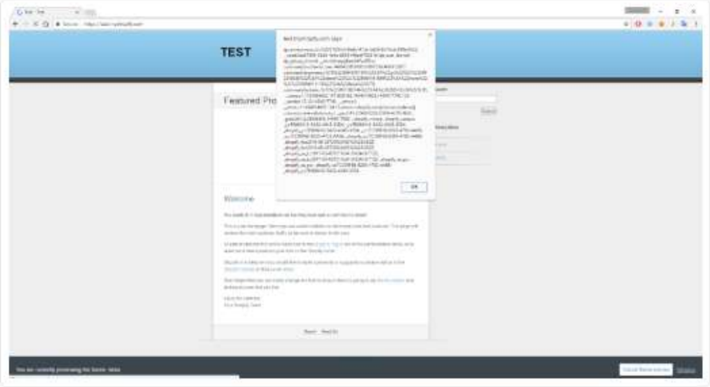


Steps to reproduce:

1. Navigate to `<account>.myshopify.com`
2. view the source of the page and copy the value of `Shopify.theme` Id.
3. Navigate to `https://echo.myshopify.com/?theme_handle=xx%27-alert(document.cookie)-%27&style_id=1&style_handle=1&preview_theme_id=<theme_ID>` > *replace* `<theme_ID>` *with the ID you just copied.*
4. XSS will trigger in all of the online shop pages unless you click `Cancel theme preview` .

PoC:

```
https://test.myshopify.com/?theme_handle=xx%27-alert(document.cookie)-%27&style_id=1&style_handle=1&preview_theme_id=3572
```



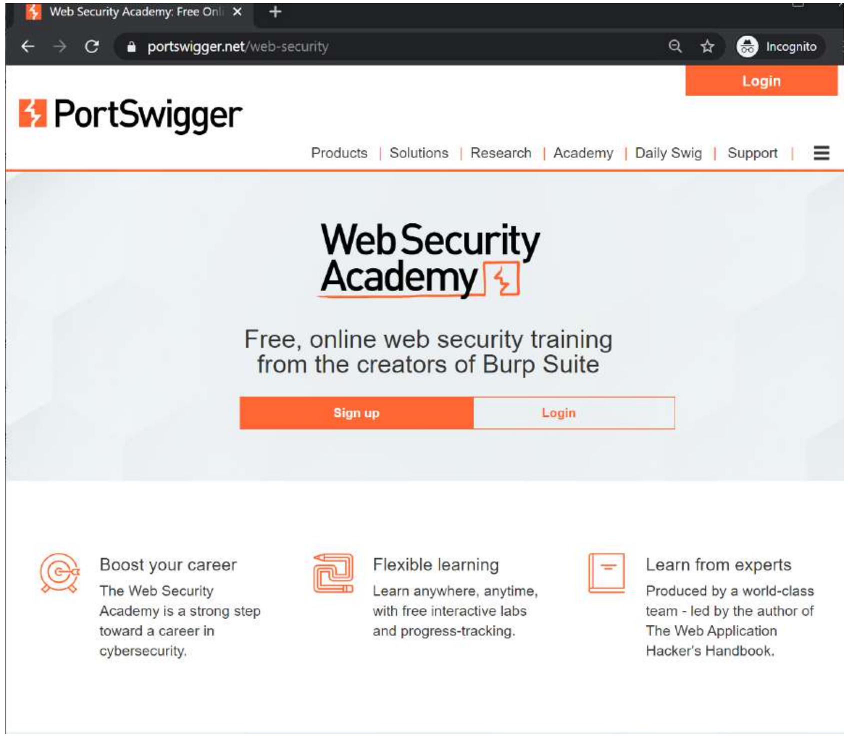
Lab 3

Uploaded Lab 3 (on Using Burp Suite)

- The tasks on Burp Suite's **components**:
  - To resend (possibly modified) individual requests using **Burp Repeater**: https://www.youtube.com/watch?v=_Wifm2g9ugg
  - To specify your **target scope** in Burp Suite: https://www.youtube.com/watch?v=0mTg2BsYVmg
  - To scan a website for vulnerabilities using **Burp Scanner**: https://www.youtube.com/watch?v=VP9eQhUASYQ

For Week 4 with Ensign

- Hands-on Burp Suite & web pen-testing review: ***please create an account with Portswigger's Web Security Academy!***
- **Project 1** discussion
- **Q&A** with Ensign:
  Via Slack (*to be emailed*)
- Burp Suite Pro training license
  (*to be emailed*)


Web Security Academy: Free Onl
portswigger.net/web-security
Incognito
Login
PortSwigger
Products | Solutions | Research | Academy | Daily Swig | Support
Web Security Academy
Sign up
Login

Free, online web security training from the creators of Burp Suite

Boost your career
The Web Security Academy is a strong step toward a career in cybersecurity.

Flexible learning
Learn anywhere, anytime, with free interactive labs and progress-tracking.

Learn from experts
Produced by a world-class team - led by the author of The Web Application Hacker's Handbook.

Thanks!
See you next week
(*Together with Ensign Team Again*)!