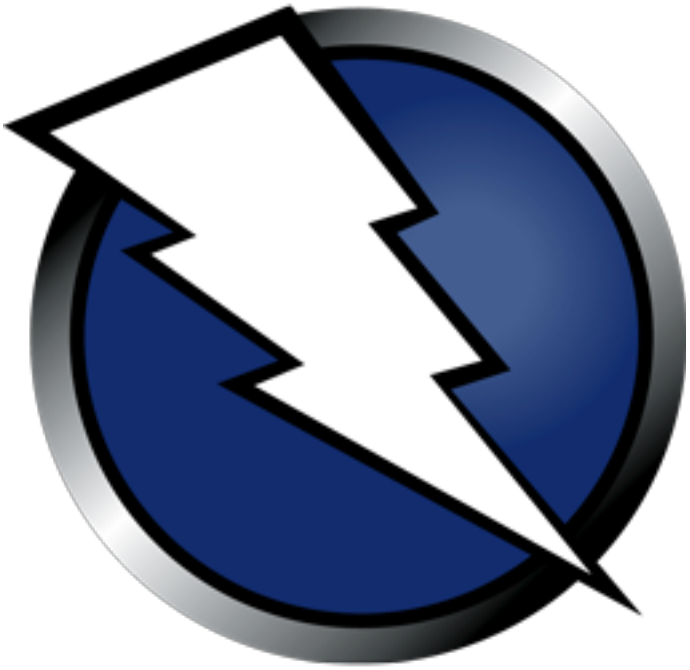
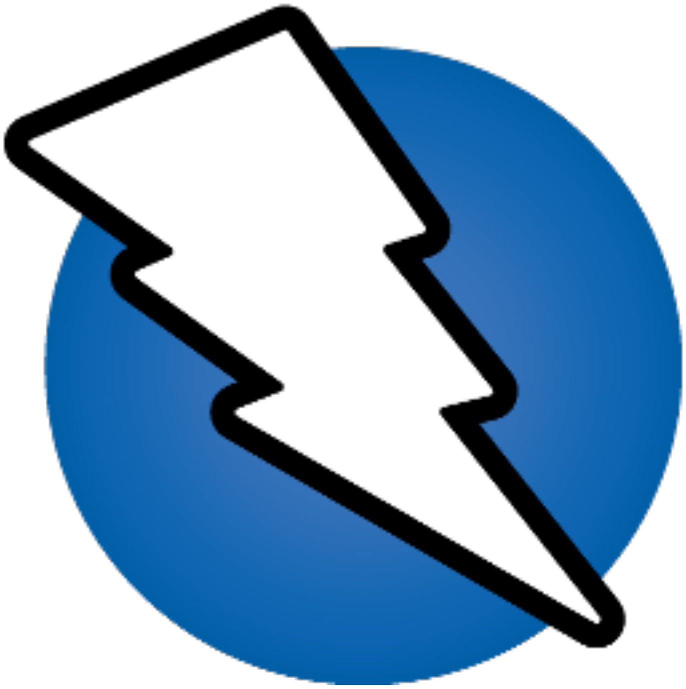
Introduction to OWASP ZAP



Introduction to OWASP ZAP

Learn how to use OWASP ZAP from the ground up. An alternative to BurpSuite.

[Task 1] Intro to ZAP



OWASP Zap is a security testing framework much like Burp Suite. It acts as a very robust enumeration tool. It’s used to test web applications.

Why wouldn’t I use Burp Suite? That’s a GOOD question! Most people in the Info-sec community DO just use Burp Suite.

But OWASP ZAP has a few benefits and features that the Burp Suite does not and it’s my preferred program of the two.

What are the benefits to OWASP ZAP? It’s completely open source and free. There is no premium version, no features are locked behind a paywall, and there is no proprietary code.
There’s a couple of feature benefits too with using OWASP ZAP over Burp Suite:

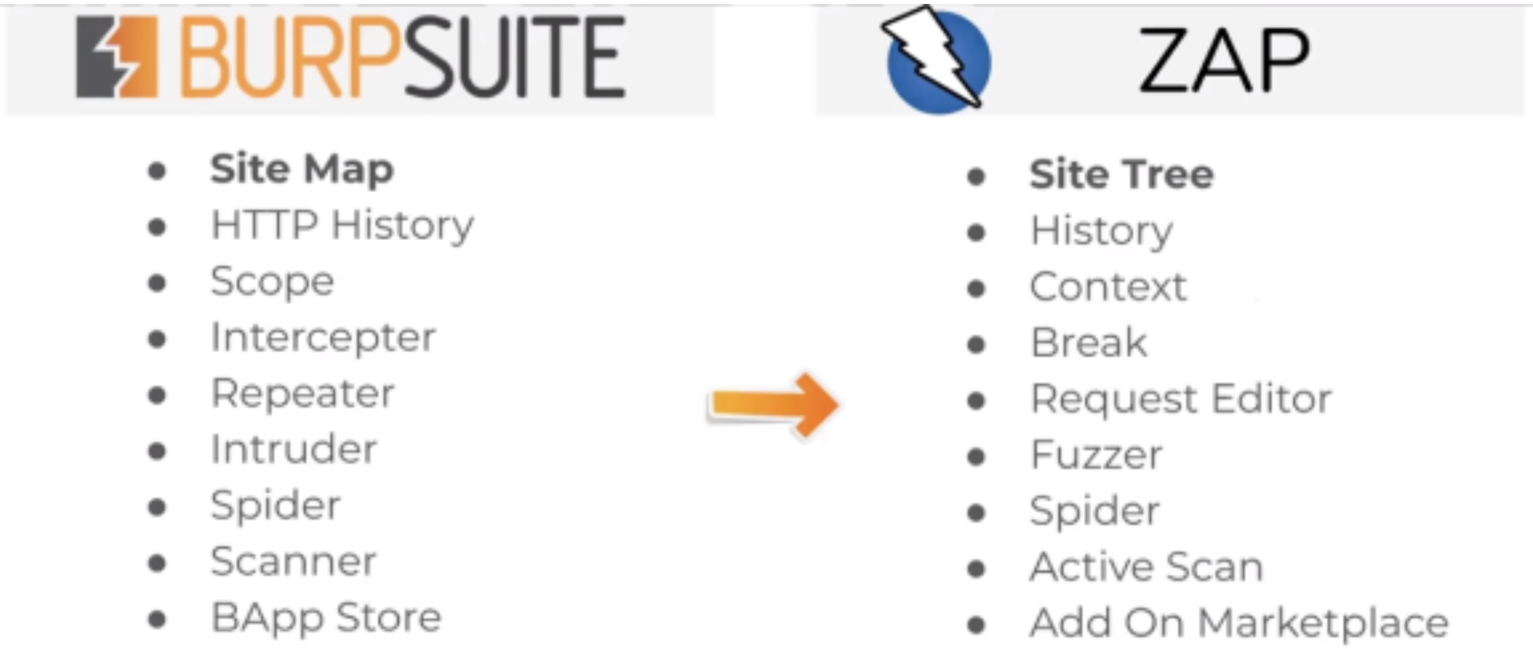
• Automated Web Application Scan: This will automatically passively and actively scan a web application, build a sitemap, and discover vulnerabilities. This is a paid feature in Burp.

• Web Spidering: You can passively build a website map with Spidering. This is a paid feature in Burp.

• Unthrottled Intruder: You can bruteforce login pages within OWASP as fast as your machine and the web-server can handle. This is a paid feature in Burp.

• No need to forward individual requests through Burp: When doing manual attacks, having to change windows to send a request through the browser, and then forward in burp, can be tedious. OWASP handles both and you can just browse the site and OWASP will intercept automatically. This is NOT a feature in Burp.

If you’re already familiar with Burp the keywords translate over like so:



BURPSUITE	ZAP
Site Map	**Site Tree**
HTTP History	History
Scope	Context
Intercepter	Break
Repeater	Request Editor
Intruder	Fuzzer
Spider	Spider
Scanner	Active Scan
BApp Store	Add On Marketplace

This guide will teach you how to do the following in ZAP:

• Automated Scan

• Directory Bruteforce

• Authenticated Scan

• Login Page Bruteforce

• Install ZAP Extensions

This room will be using OWASP Zap against the DVWA machine, feel free to deploy your own instance and follow along.

#1

What does ZAP stand for?

Zed Attack Proxy

#2

Connect to the TryHackMe network and deploy the machine. Once deployed, wait a few minutes and visit the web application: http://10.10.153.246

No answer needed

[Task 2] Disclaimer

ZAP is a great tool that’s totally slept on, and I personally prefer it over Burp, but the documentation and support for the tool is microscopic compared to the titan that is Burp.

Burp has some extensions and features that ZAP does not have, as an example ZAP is unable to perform Login timing attacks. Burp can. If you wish to learn more about login timing attacks you can check out the TryHackMe room Hackernote.
ZAP can be used as your go-to tool to start Web Application testing but it should not be your *only* tool. ZAP is just one of many tools to put under your hacker utility belt.

#1
I've read the task.

No answer needed

[Task 3] Installation

OWASP ZAP has a handy installer for Windows, Mac OS, and Linux systems.
Download and install it from the official website: https://www.zaproxy.org/download/

#1
Install ZAP on an operating system of your choice!

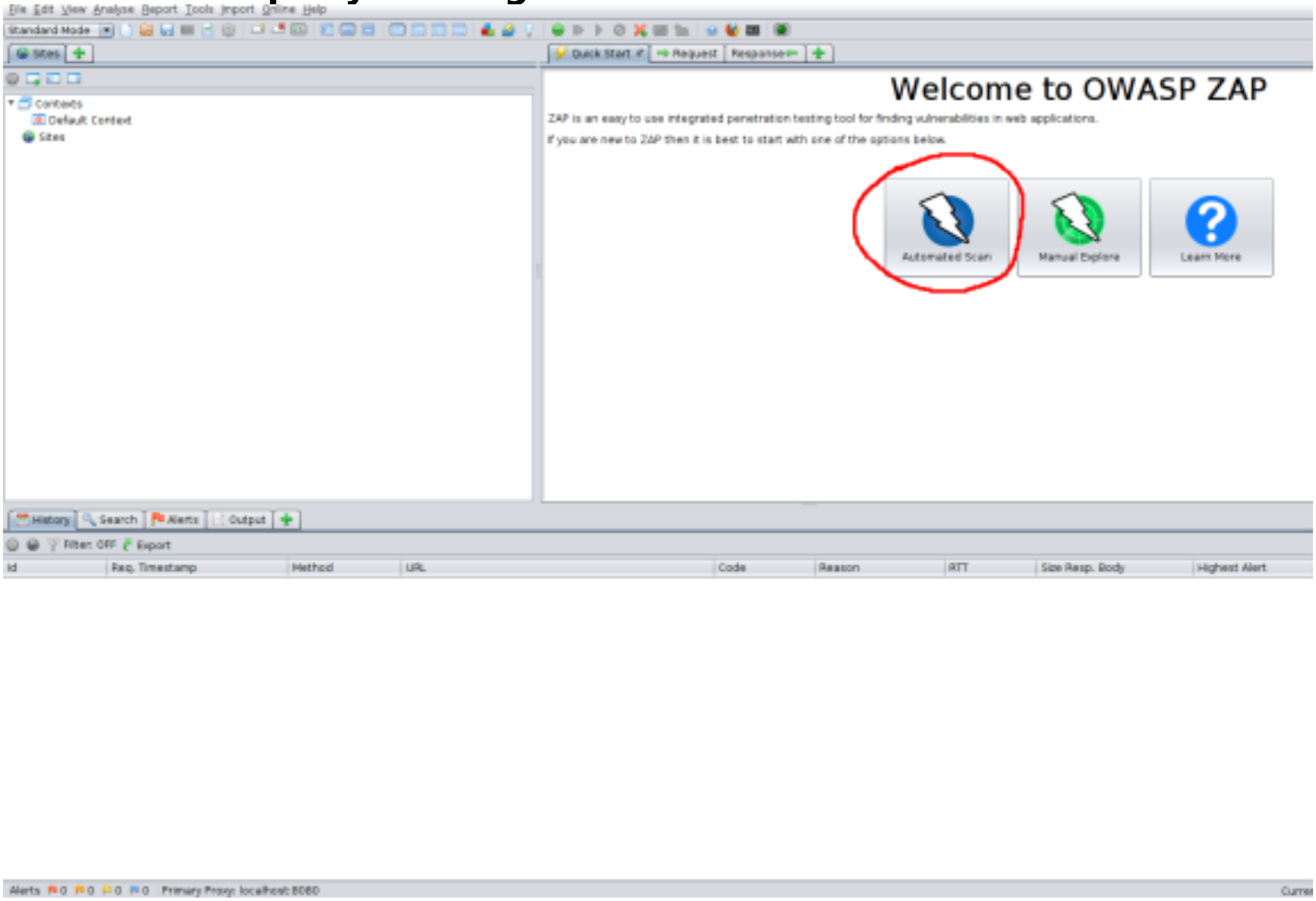
No answer needed

#2
Open OWASP ZAP, ready to follow along with this room.

No answer needed

[Task 4] How to perform an automated scan

Lets perform an automated scan. Click the big Automated Scan button and input your target.



Welcome to OWASP ZAP

ZAP is an easy to use integrated penetration testing tool for finding vulnerabilities in web applications.

If you are new to ZAP then it is best to start with one of the options below.

The automated scan performs both passive and automated scans to build a sitemap and detect vulnerabilities. On the next page you may see the options to select either to use “traditional spider” or “Ajax spider”.

A traditional spider scan is a passive scan that enumerates links and directories of the website. It builds a website index without brute-forcing. This is much quieter than a brute-force attack and can still net a login page or other juicy details, but is not as comprehensive as a bruteforce.

The Ajax Spider is an add-on that integrates in ZAP a crawler of AJAX rich sites called Crawljax. You can use it in conjunction with the traditional spider for better results. It uses your web browser and proxy.

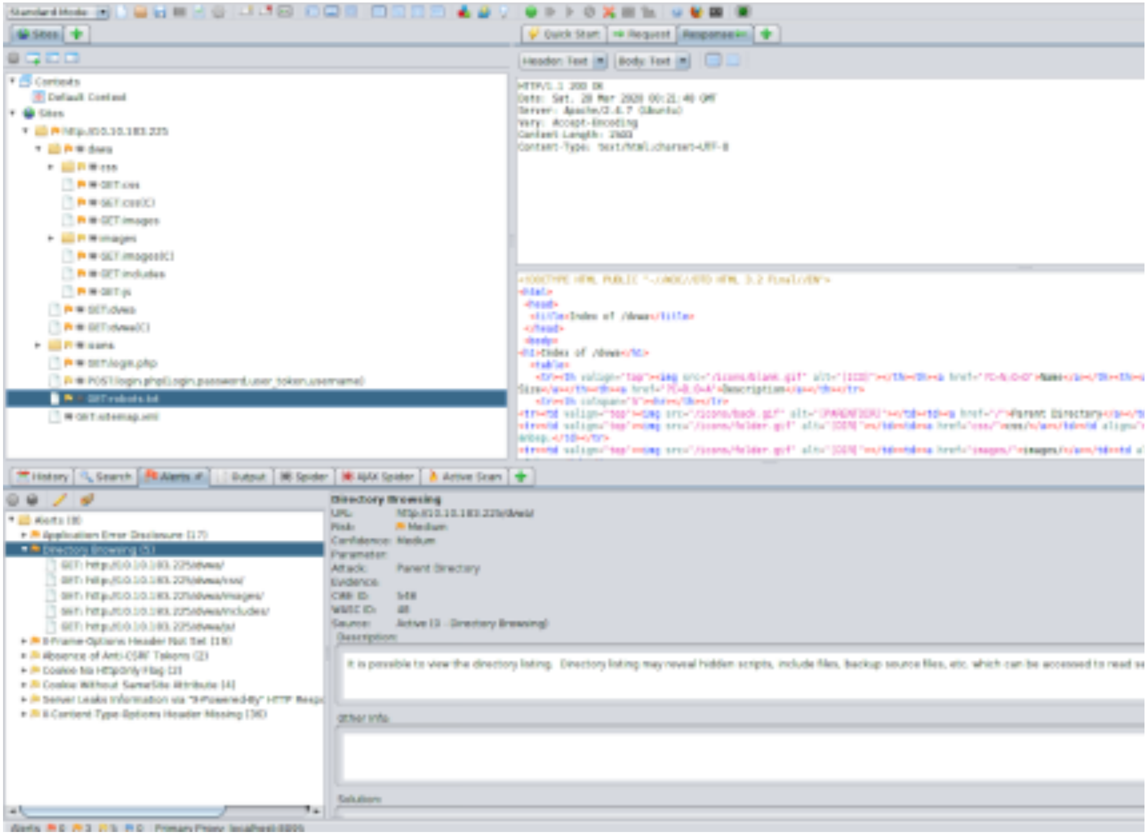
The easiest way to use the Ajax Spider is with HTMLUnit.

To install HTML Unit use the command

```
sudo apt install libjenkins-htmlunit-core-js-java
```

And then select HtmlUnity from the Ajax Spider Dropdown.
Both utilities can further be configured in the options menu (Ctrl+Alt+O)

Example Automated Scan Output:



```
HTTP/1.1 200 OK
Date: Sat, 28 Mar 2020 00:21:40 GMT
Server: Apache/2.4.7 (Ubuntu)
Vary: Accept-Encoding
Content-Length: [illegible]
Content-Type: text/html;charset=UTF-8
```

```
Directory Browsing
URL:        http://10.10.183.225/dvwa/
Risk:       Medium
Confidence: Medium
Parameter:
Attack:     Parent Directory
Evidence:
CWE ID:     548
WASC ID:    48
Source:     Active (0 - Directory Browsing)
Description:
It is possible to view the directory listing. Directory listing may reveal hidden scripts, include files, backup source files, etc. which can be accessed to read se
```

With very minimal setup we were able to do an automated scan that gave us a sitemap and a handful of vulnerabilities.

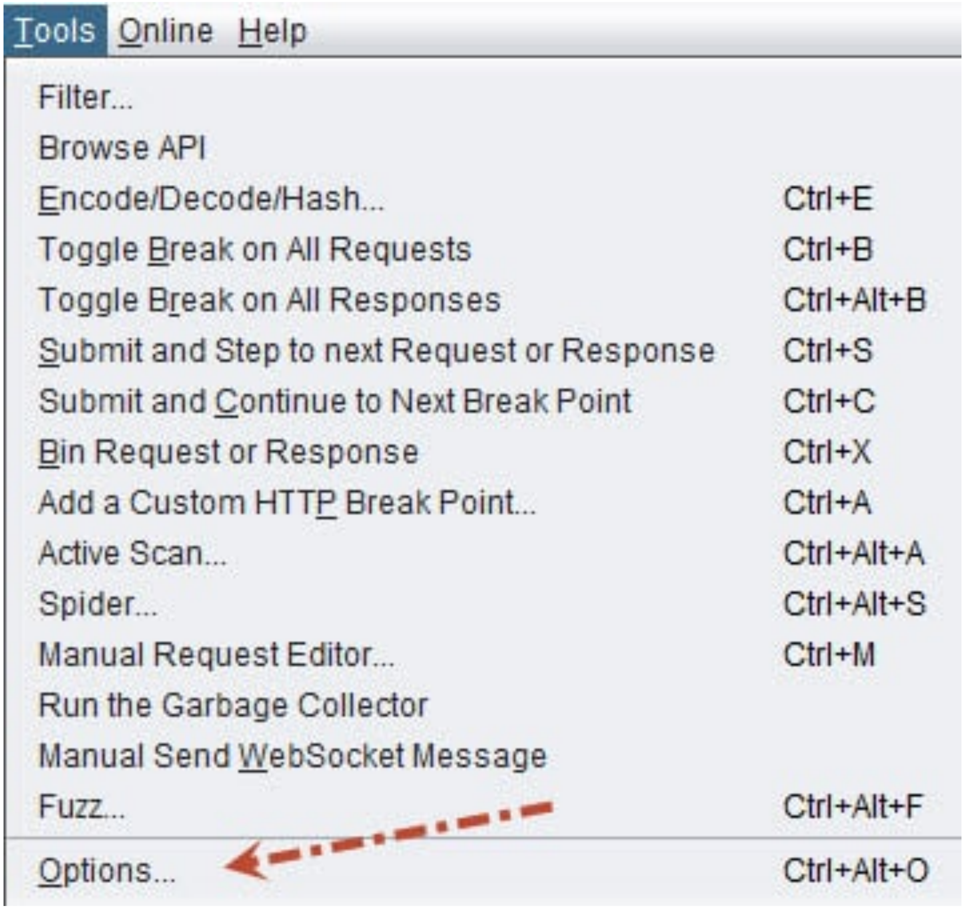
#1
Set up Ajax Spider

No answer needed

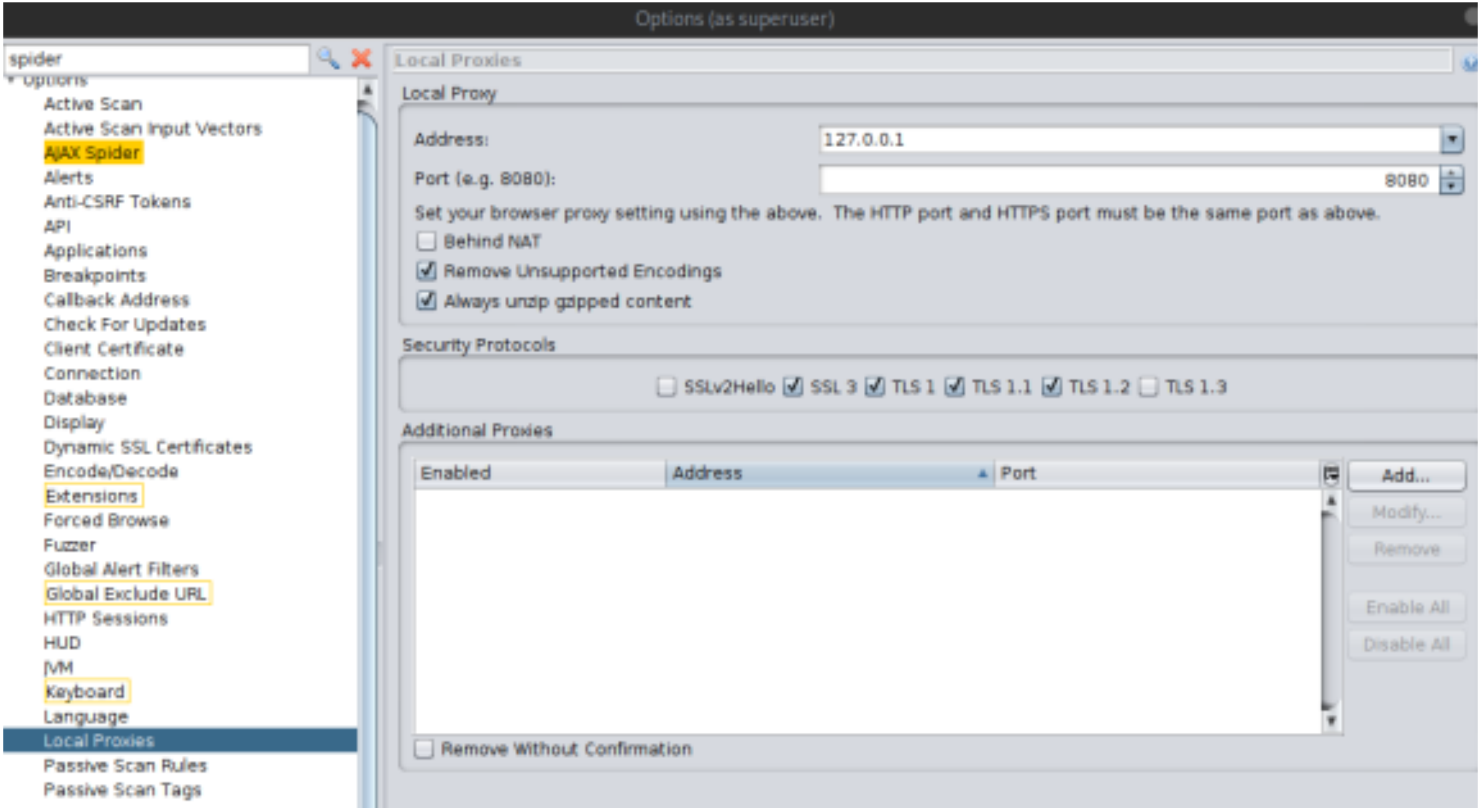
[Task 5] Manual Scanning

Lets perform a manual scan against the DVWA machine.
Like Burp, you should set-up your proxy between OWASP ZAP and your Browser. We'll be using Firefox.
===

OWASP Proxy Setup:


Tools
Online
Help
Filter...
Browse API
Encode/Decode/Hash... Ctrl+E
Toggle Break on All Requests Ctrl+B
Toggle Break on All Responses Ctrl+Alt+B
Submit and Step to next Request or Response Ctrl+S
Submit and Continue to Next Break Point Ctrl+C
Bin Request or Response Ctrl+X
Add a Custom HTTP Break Point... Ctrl+A
Active Scan... Ctrl+Alt+A
Spider... Ctrl+Alt+S
Manual Request Editor... Ctrl+M
Run the Garbage Collector
Manual Send WebSocket Message
Fuzz... Ctrl+Alt+F
Options... Ctrl+Alt+O

Open Options



Local Proxy

Address: 127.0.0.1

Port (e.g. 8080): 8080

Set your browser proxy setting using the above. The HTTP port and HTTPS port must be the same port as above.

☐ Behind NAT

☑ Remove Unsupported Encodings

☑ Always unzip gzipped content

Security Protocols

☐ SSLv2Hello ☑ SSL 3 ☑ TLS 1 ☑ TLS 1.1 ☑ TLS 1.2 ☐ TLS 1.3

Additional Proxies

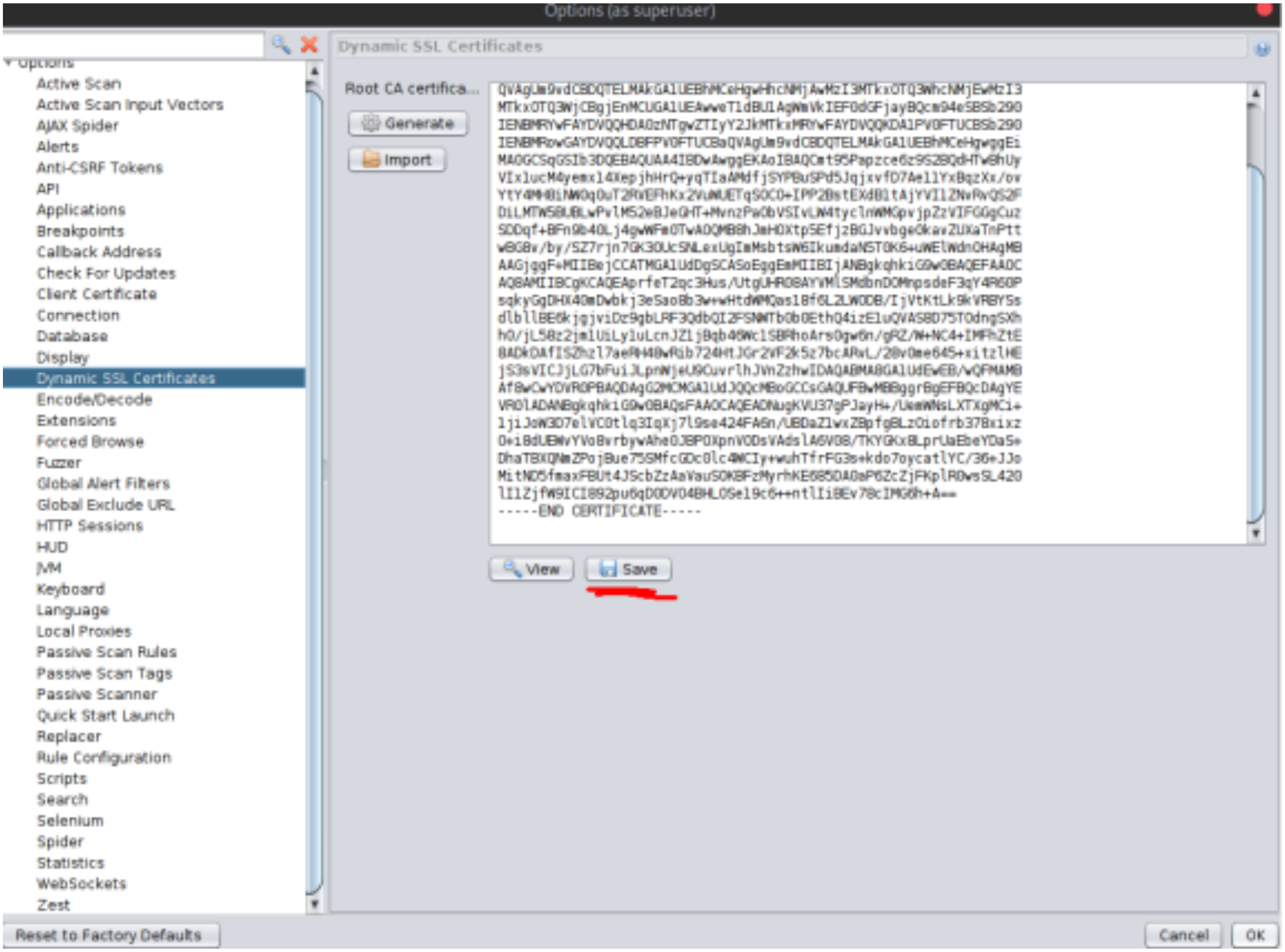
Enabled	Address	Port

☐ Remove Without Confirmation

Change Local Proxy settings to the above.
==

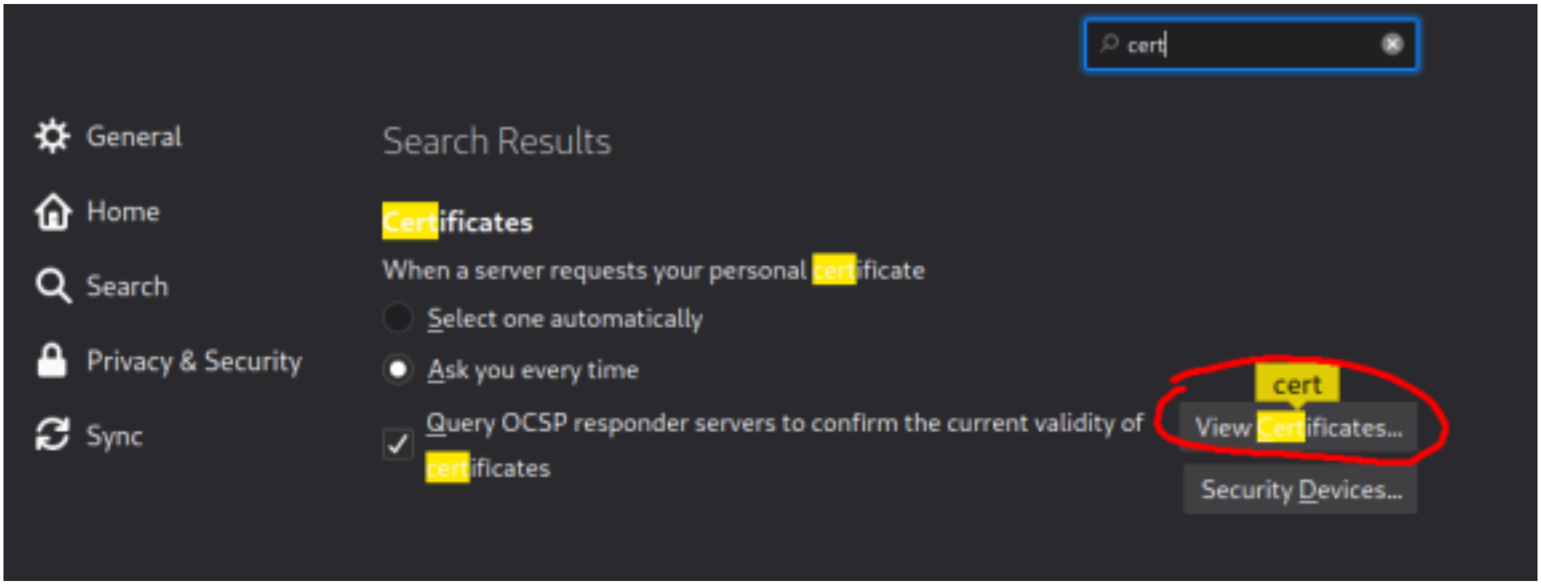
Add ZAP Certificates:

Without importing ZAP Certificates, ZAP is unable to handle simultaneous Web request forwarding and intercepting. Do not skip this step.



```
QVAgUm9vdCBDQTELMAkGA1UEBhMCeHgwHhcNMjAwMzI3MTkxOTQ3WhcNMjEwMzI3
MTkxOTQ3WjCBgjEnMCUGA1UEAwweT1dBU1AgWmVkIEF0dGFjayBQcm94eSBSb290
IENBMRYwFAYDVQQHDA0zNTgwZTIyY2JkMTkxMRYwFAYDVQQKDA1PV0FTUCBSb290
IENBMRowGAYDVQQLDBFPV0FTUCBaQVAgUm9vdCBDQTELMAkGA1UEBhMCeHgwggEi
MA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCmt95Papzce6z9S2BQdHTw8hUy
VIxlucM4yemxl4XepjhHrQ+yqTIaAMdfjSYPBuSPd5JqjxvfD7AellYxBqzXx/ow
YtY4MH0iNWOqOuT2RVEFhKx2VuWUETqSOC0+IPP20stEXdB1tAjYVIlZNwRvQS2F
DiLMTW5BUBLwPvlM52eBJeGHT+MvnzPaObVSIvLW4tyclnWMGpvjpZzVIFGGgCuz
SDDqf+BFn9b40Lj4gwWFm0TwAOQMB8hJmHOXtp5EfjzBGJvvbge0kavZUXaTnPtt
wBG8v/by/SZ7rjn7GK30UcSNLexUgImMsbtsW6IkumdaNST0K6+uWElWdnOHAgMB
AAGjggF+MIIBejCCATMGA1UdDgSCASoEggEmMIIBIjANBgkqhkiG9w0BAQEFAAOC
AQ8AMIIBCgKCAQEAprfeT2qc3Hus/UtgUHR08AYWMlSMdbnDOMnpsdeF3qY4R60P
sqkyGgDHX40mDwbkj3eSaoBb3w+wHtdWMQas18f6L2LW0DB/IjVtKtLk9kVRBYSs
dlbllBE6kjgjviDz9gbLRF3QdbQI2FSNWTbObOEthQ4izE1uQVAS8D75TOdngSXh
hO/jL58z2jmlUiLyluLcnJZljBqb40Wc1SBRhoArsOgw6n/gRZ/W+NC4+IMFhZtE
0ADkOAfISZhzl7aeRH40wRib724HtJGr2VF2k5z7bcARwL/20vOme645+xitzlHE
jS3sVICJjLG7bFuiJLpnWjeU9CuvrlhJVnZzhwIDAQABMA8GA1UdEwEB/wQFMAMB
Af8wCwYDVR0PBAQDAgG2MCMGA1UdJQQcMBoGCCsGAQUFBwMBBggrBgEFBQcDAgYE
VR0lADANBgkqhkiG9w0BAQsFAAOCAQEADNugKVU37gPJayH+/UemWNsLXTXgMCi+
1jiJoW3D7elVC0tlq3IqXj7l9se424FA6n/UBDaZlwxZBpfgBLzOiofrb37Bxixz
O+iBdUEWvYVoBvrbywAhe0JBPOXpnVODsVAdslA6V08/TKYGKxBLprUaEbeYDaS+
DhaTBXQNmZPojBue75SMfcGDc0lc4WCIy+wuhTfrFG3s+kdo7oycatlYC/36+JJo
MitND5fmaxFBUt4JScbZzAaVauSOKBFzMyrhKE685DA0aP6ZcZjFKplROwsSL420
lIlZjfW9ICI892pu6qDODVO4BHLOSel9c6++ntlIiBEv78cIMG6h+A==
-----END CERTIFICATE-----
```

In the same options menu, navigate to Dynamic SSL Certificates and save the certificate somewhere you’ll remember and not delete.



Certificates

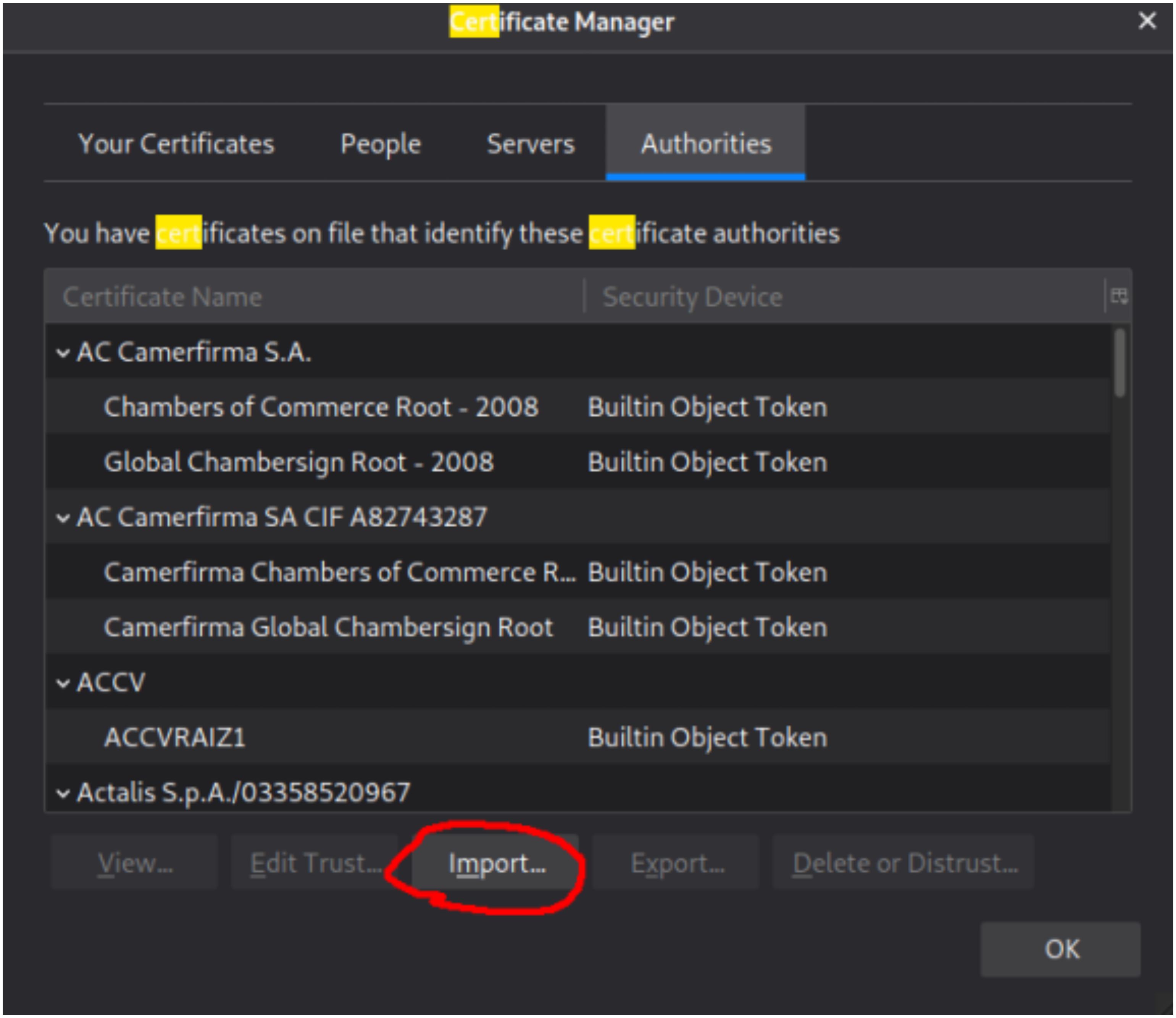
When a server requests your personal certificate

○ Select one automatically

● Ask you every time

☑ Query OCSP responder servers to confirm the current validity of certificates

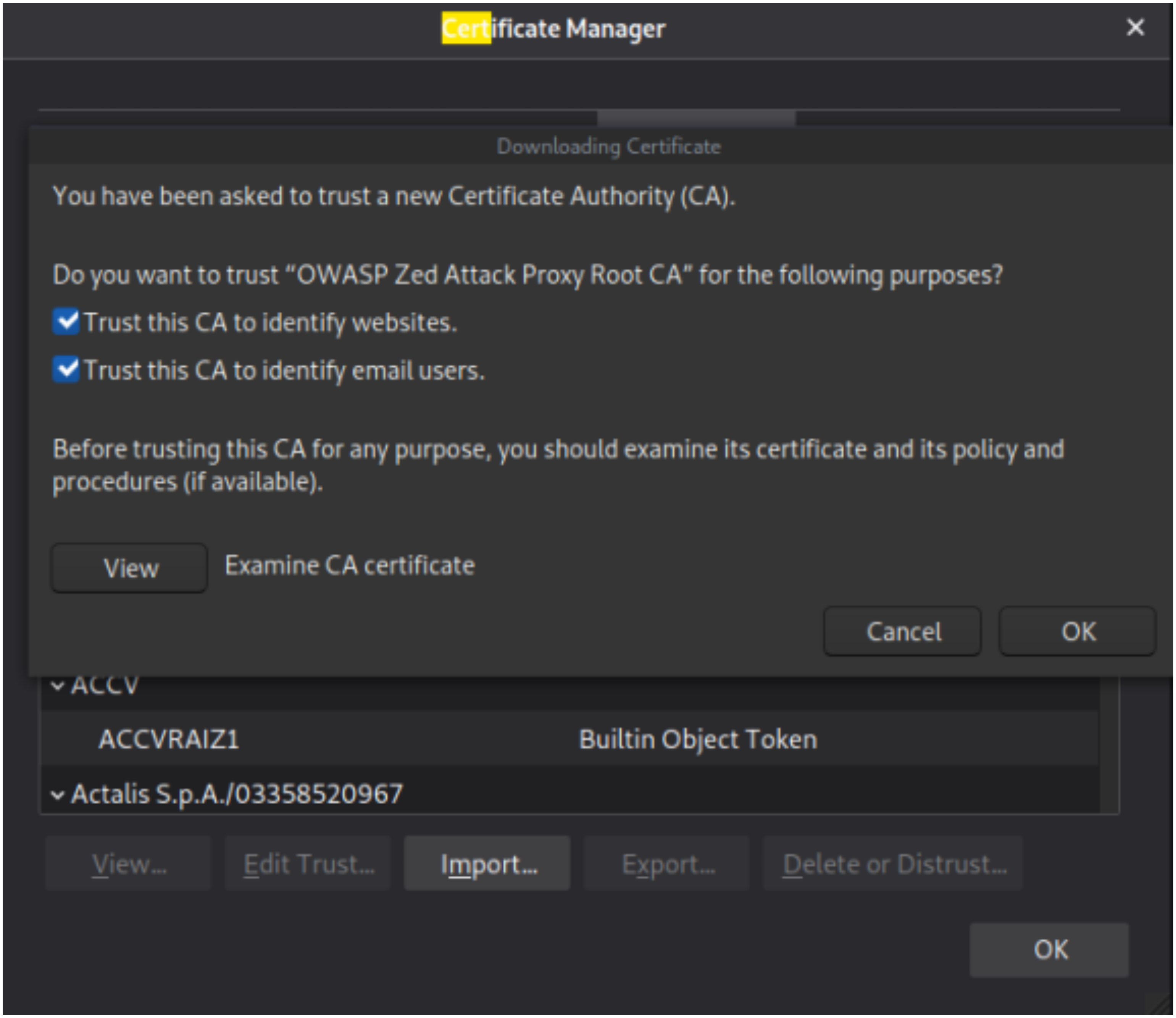
Then, open Firefox, navigate to your preferences, and search for certificates and click “View Certificates”



You have certificates on file that identify these certificate authorities

Certificate Name	Security Device
AC Camerfirma S.A.	
Chambers of Commerce Root - 2008	Builtin Object Token
Global Chambersign Root - 2008	Builtin Object Token
AC Camerfirma SA CIF A82743287	
Camerfirma Chambers of Commerce R...	Builtin Object Token
Camerfirma Global Chambersign Root	Builtin Object Token
ACCV	
ACCVRAIZ1	Builtin Object Token
Actalis S.p.A./03358520967	

Then click “Import” and then navigate to the earlier downloaded certificate and open it.



You have been asked to trust a new Certificate Authority (CA).

Do you want to trust “OWASP Zed Attack Proxy Root CA” for the following purposes?

☑ Trust this CA to identify websites.

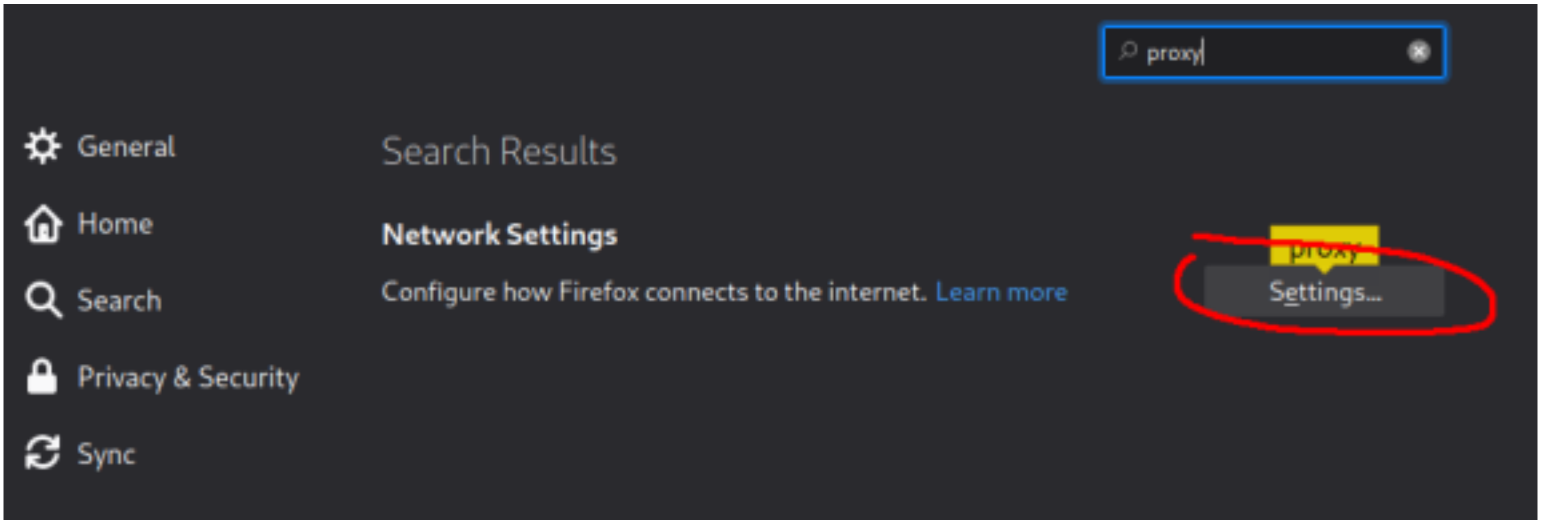
☑ Trust this CA to identify email users.

Before trusting this CA for any purpose, you should examine its certificate and its policy and procedures (if available).

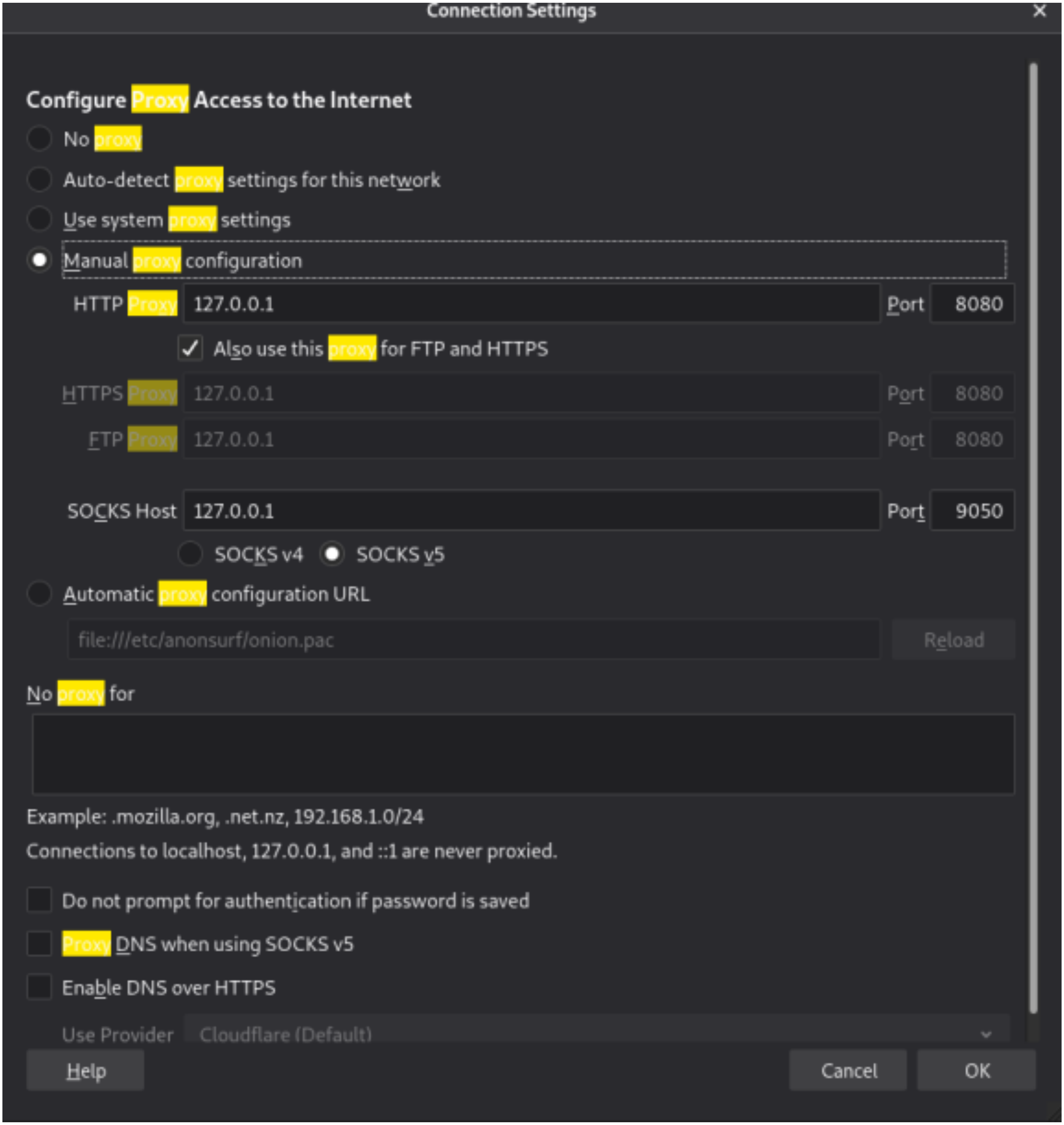
⌄ ACCV	
ACCVRAIZ1	Builtin Object Token
⌄ Actalis S.p.A./03358520967	

Select both and then hit OK.
===

Firefox Proxy Setup:



Go back to your Firefox preferences and search for “proxy”. Click Settings.



Connection Settings

Configure Proxy Access to the Internet

○ No proxy

○ Auto-detect proxy settings for this network

○ Use system proxy settings

● Manual proxy configuration

HTTP Proxy 127.0.0.1 Port 8080

☑ Also use this proxy for FTP and HTTPS

HTTPS Proxy 127.0.0.1 Port 8080

FTP Proxy 127.0.0.1 Port 8080

SOCKS Host 127.0.0.1 Port 9050

○ SOCKS v4 ● SOCKS v5

○ Automatic proxy configuration URL

file:///etc/anonsurf/onion.pac Reload

No proxy for

Example: .mozilla.org, .net.nz, 192.168.1.0/24

Connections to localhost, 127.0.0.1, and ::1 are never proxied.

□ Do not prompt for authentication if password is saved

□ Proxy DNS when using SOCKS v5

□ Enable DNS over HTTPS

Use Provider Cloudflare (Default)

Adjust your Manual Proxy Configuration to match and then click OK.

Now you’re set-up! Time to get into the fun stuff :)

#1
What IP do we use for the proxy?

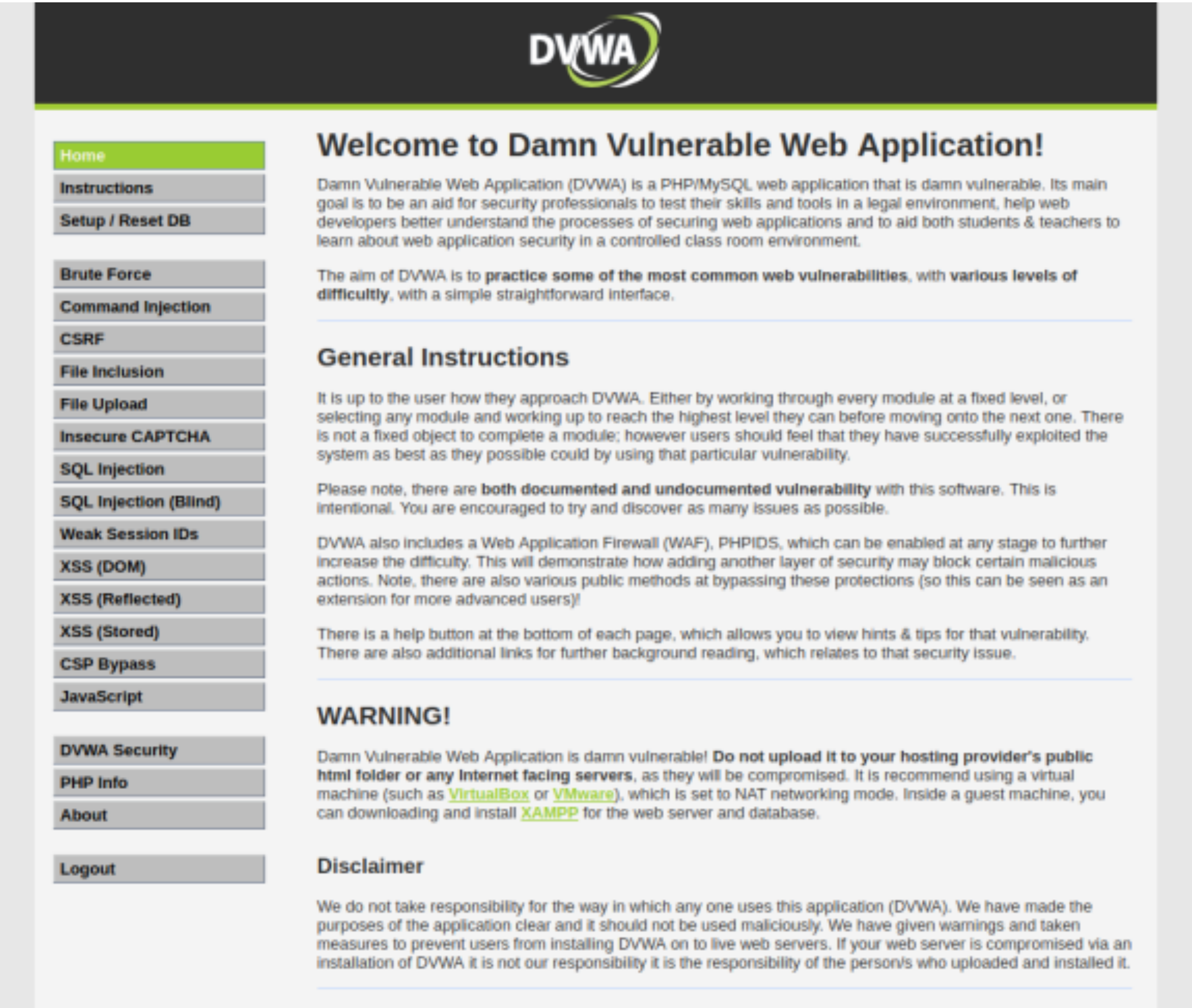
127.0.0.1

[Task 6] Scanning an Authenticated Web Application

Without your Zap application being authenticated, it can't scan pages that are only accessible when you've logged in. Lets set up the OWASP ZAP application to scan these pages, using your logged in session.

Lets go to the DVWA machine (http://10.10.153.246), and login using the following credentials:

Username: admin**Password**: password



Welcome to Damn Vulnerable Web Application!

Damn Vulnerable Web Application (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goal is to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and to aid both students & teachers to learn about web application security in a controlled class room environment.

The aim of DVWA is to **practice some of the most common web vulnerabilities**, with **various levels of difficultly**, with a simple straightforward interface.

General Instructions

It is up to the user how they approach DVWA. Either by working through every module at a fixed level, or selecting any module and working up to reach the highest level they can before moving onto the next one. There is not a fixed object to complete a module; however users should feel that they have successfully exploited the system as best as they possible could by using that particular vulnerability.

Please note, there are **both documented and undocumented vulnerability** with this software. This is intentional. You are encouraged to try and discover as many issues as possible.

DVWA also includes a Web Application Firewall (WAF), PHPIDS, which can be enabled at any stage to further increase the difficulty. This will demonstrate how adding another layer of security may block certain malicious actions. Note, there are also various public methods at bypassing these protections (so this can be seen as an extension for more advanced users)!

There is a help button at the bottom of each page, which allows you to view hints & tips for that vulnerability. There are also additional links for further background reading, which relates to that security issue.

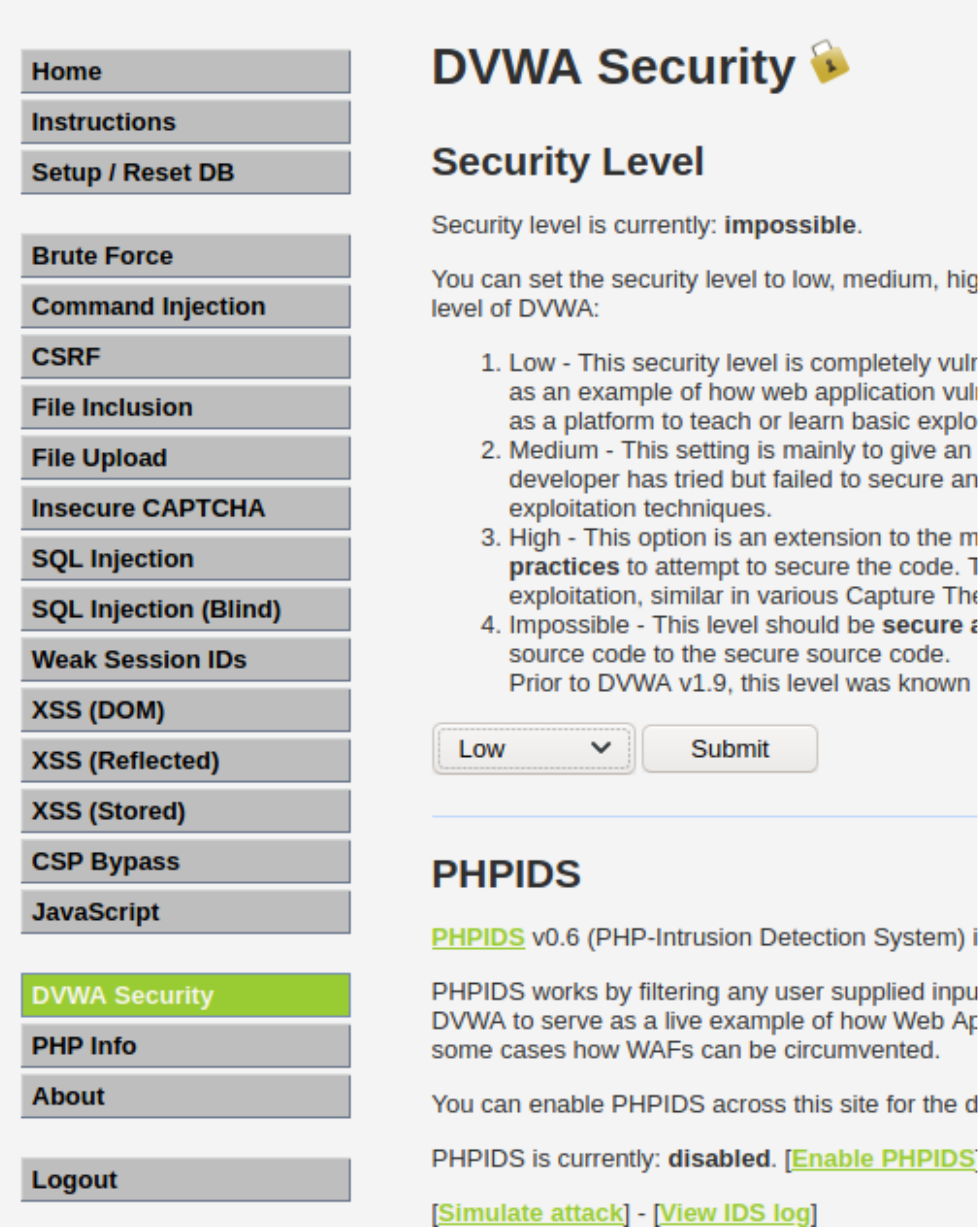
WARNING!

Damn Vulnerable Web Application is damn vulnerable! **Do not upload it to your hosting provider's public html folder or any Internet facing servers**, as they will be compromised. It is recommend using a virtual machine (such as VirtualBox or VMware), which is set to NAT networking mode. Inside a guest machine, you can downloading and install XAMPP for the web server and database.

Disclaimer

We do not take responsibility for the way in which any one uses this application (DVWA). We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the person/s who uploaded and installed it.

Afte



DVWA Security

Security Level

Security level is currently: **impossible**.

You can set the security level to low, medium, hig
level of DVWA:

1. Low - This security level is completely vuln
   as an example of how web application vul
   as a platform to teach or learn basic explo
2. Medium - This setting is mainly to give an
   developer has tried but failed to secure an
   exploitation techniques.
3. High - This option is an extension to the m
   practices to attempt to secure the code. T
   exploitation, similar in various Capture The
4. Impossible - This level should be **secure a**
   source code to the secure source code.
   Prior to DVWA v1.9, this level was known

PHPIDS

PHPIDS v0.6 (PHP-Intrusion Detection System) i

PHPIDS works by filtering any user supplied inpu
DVWA to serve as a live example of how Web Ap
some cases how WAFs can be circumvented.

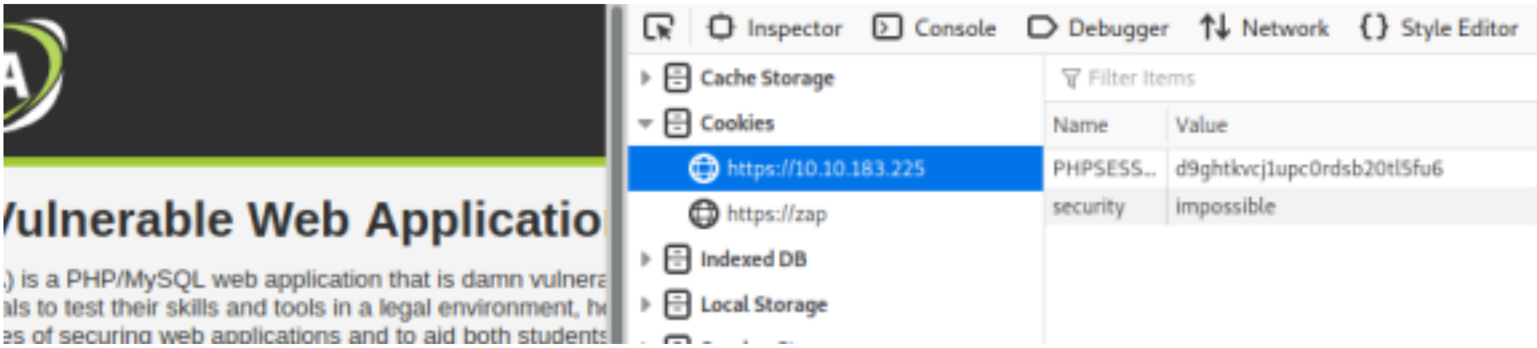
You can enable PHPIDS across this site for the d

PHPIDS is currently: **disabled**. [Enable PHPIDS

[Simulate attack] - [View IDS log]

For the purpose of this exercise, once you've logged in, navigate to the DVWA Security tab and set the Security level to Low and then hit submit.

We're going to pass our authentication token into ZAP so that we can use the tool to scan authenticated webpages.

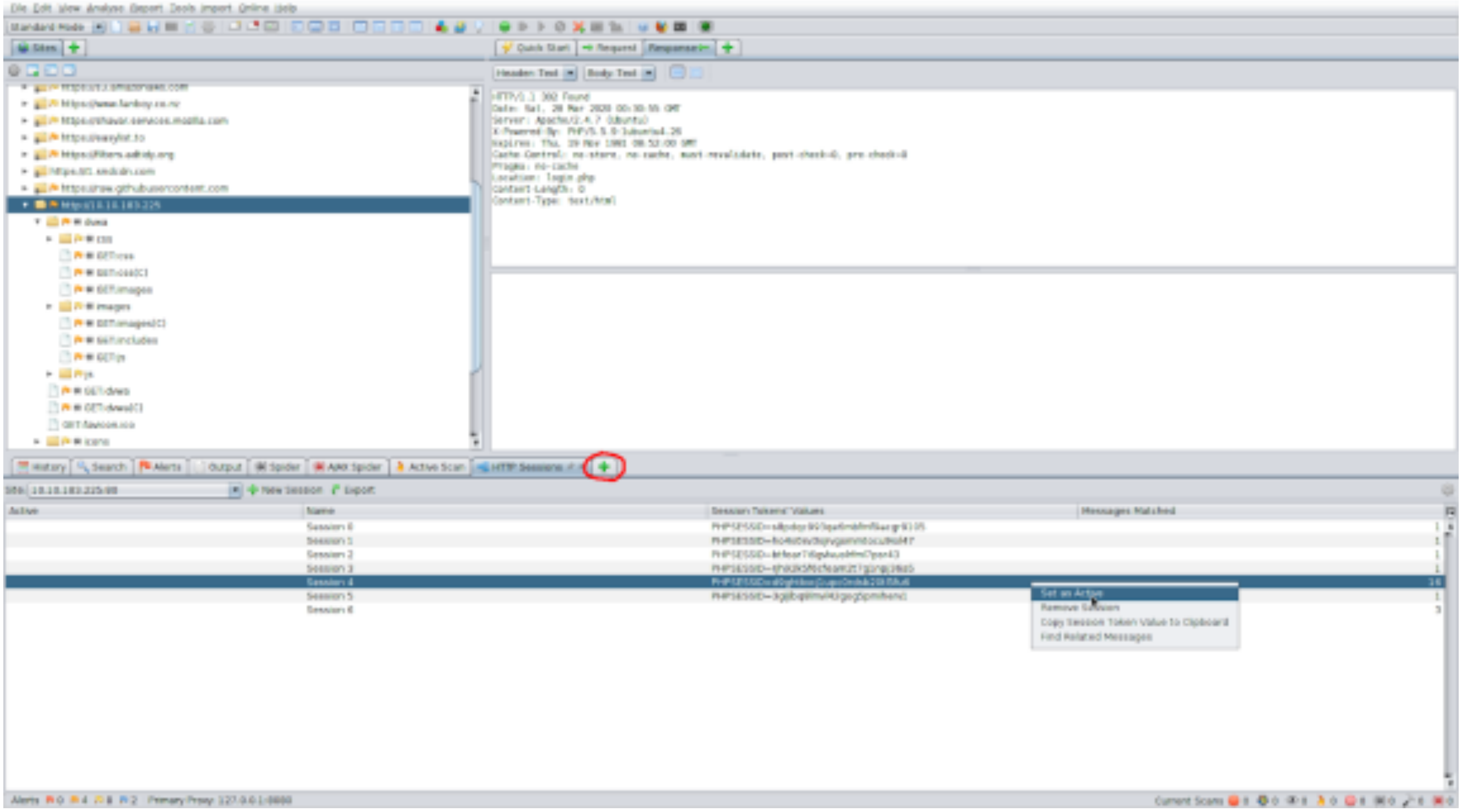


Vulnerable Web Applicatio

) is a PHP/MySQL web application that is damn vulnera
als to test their skills and tools in a legal environment, he
es of securing web applications and to aid both students

Name	Value
PHPSESS...	d9ghtkvcj1upc0rdsb20tl5fu6
security	impossible

Enter inspect element and take note of your PHPSESSION cookie.



```
HTTP/1.1 302 Found
Date: Sat, 28 Mar 2020 00:30:55 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.26
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Location: login.php
Content-Length: 0
Content-Type: text/html
```

Site: 10.10.183.225:80

Active	Name	Session Tokens' Values	Messages Matched
	Session 0	PHPSESSID=[illegible]	1
	Session 1	PHPSESSID=[illegible]	1
	Session 2	PHPSESSID=[illegible]	1
	Session 3	PHPSESSID=[illegible]	1
	Session 4	PHPSESSID=[illegible]	16
	Session 5	PHPSESSID=[illegible]	1
	Session 6		3

In ZAP open the HTTP Sessions tab with the new tab button, and set the authenticated session as active.

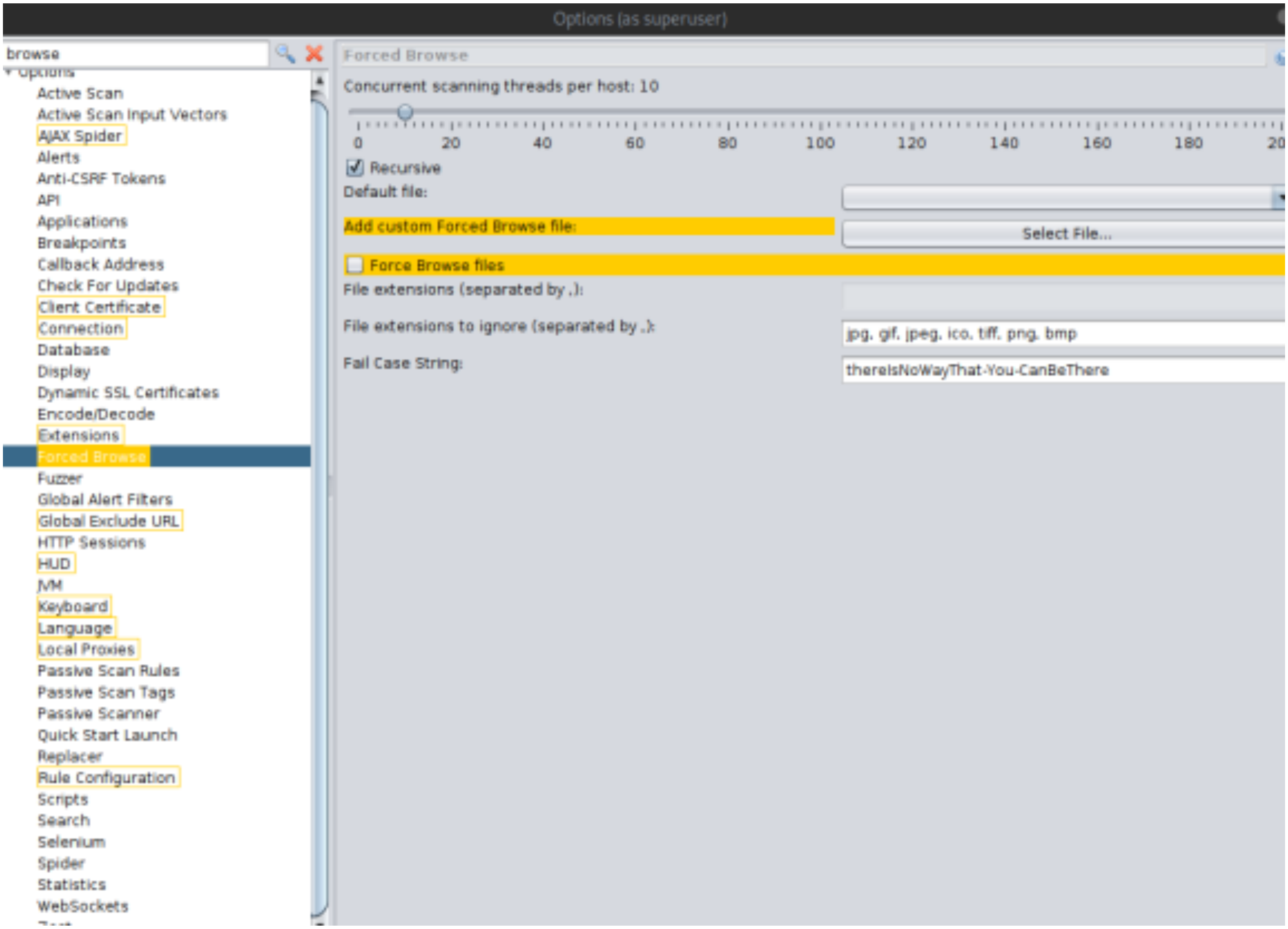
Now re-scan the application. You'll see it's able to pick up *a lot* more. This is because its able to see all of the sections of DVWA that was previously behind the login page.

#1
Try scanning the DVWA web application as an authenticated user.

No answer needed

[Task 7] Brute-force Directories

If the passive scans are not enough, you can use a wordlist attack and directory bruteforce through ZAP just as you would with gobuster. This would pick up pages that are not indexed.



Concurrent scanning threads per host: 10

☑ Recursive

Default file:

Add custom Forced Browse file:

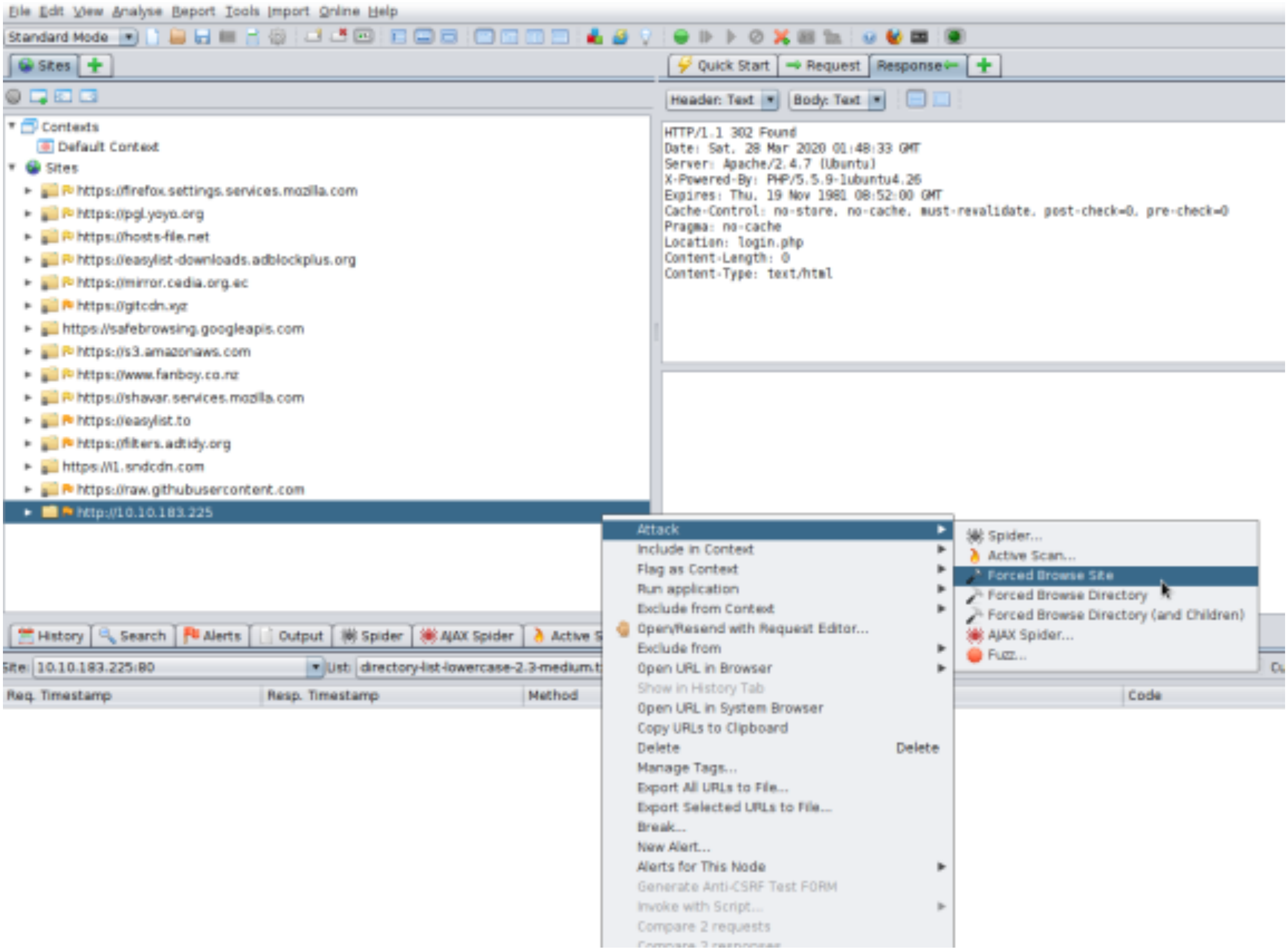
□ Force Browse files

File extensions (separated by ,):

File extensions to ignore (separated by ,): jpg, gif, jpeg, ico, tiff, png, bmp

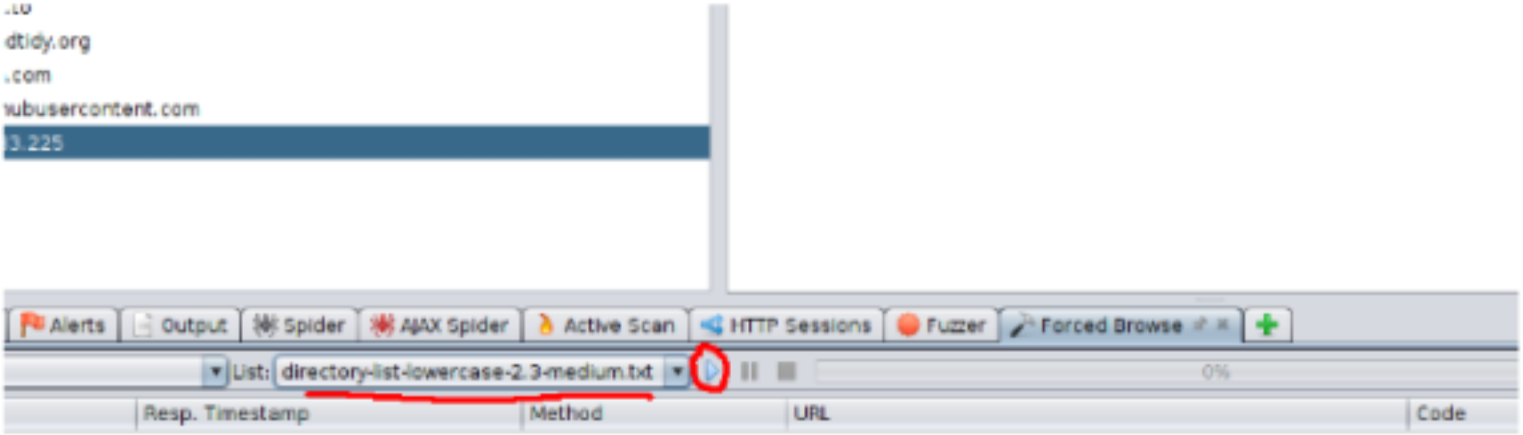
Fail Case String: thereIsNoWayThat-You-CanBeThere

First. Go into your ZAP Options (at the bottom navigation panel, with the screen plus button), navigate to Forced Browse, and add the Custom Wordlist. You can also add more threads and turn off recursive brute-forcing.



```
HTTP/1.1 302 Found
Date: Sat, 28 Mar 2020 01:48:33 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.26
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Location: login.php
Content-Length: 0
Content-Type: text/html
```

Then, right click the site->attack->forced browse site



Select your imported wordlist from the list menu, and then hit the play button! We recommend using this wordlist for this exercise.
ZAP will now bruteforce the entire website with your wordlist.

#1

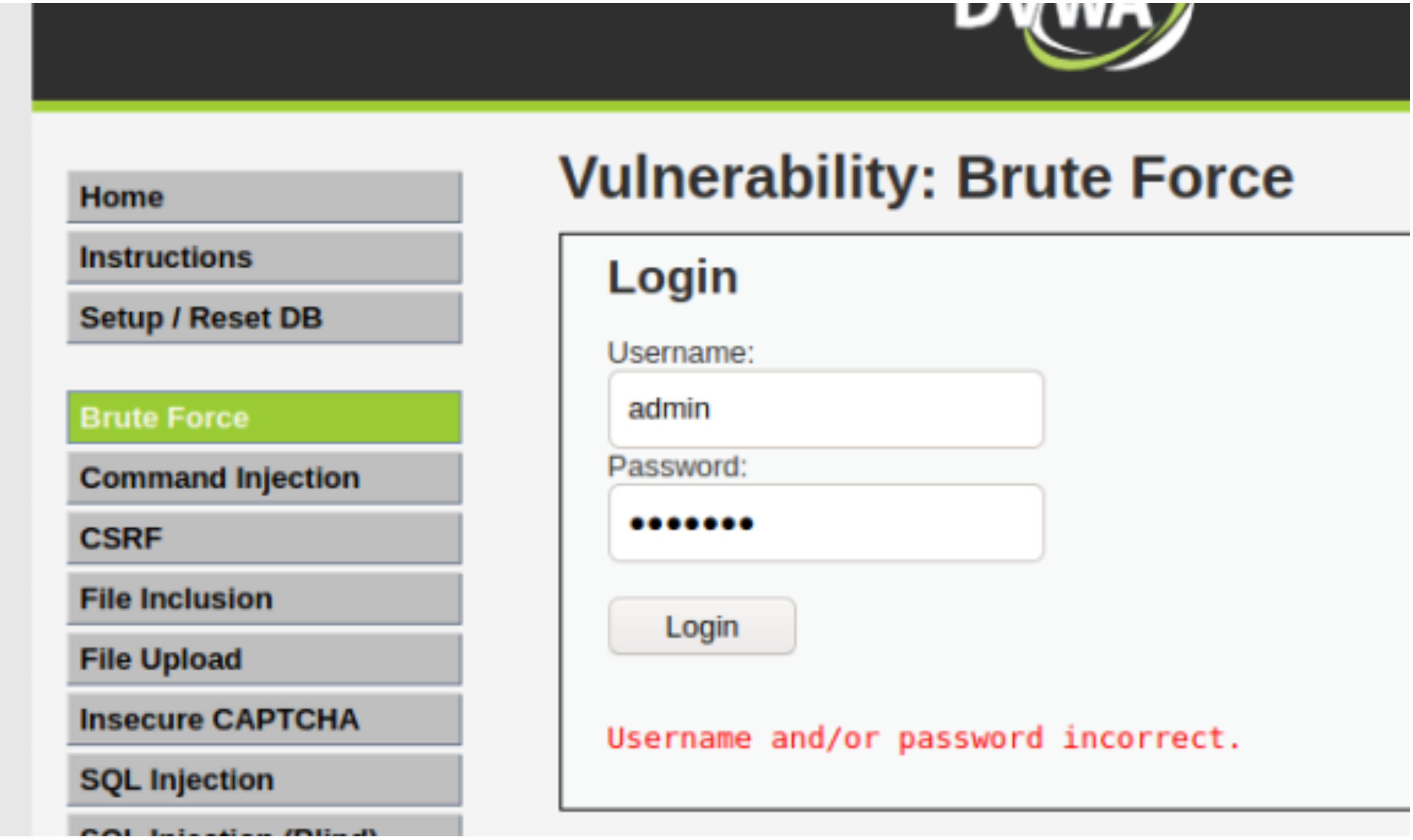
Try brute-forcing the DVWA web application.

No answer needed

[Task 8] Bruteforce Web Login

Lets brute-force a form to get credentials. Although we already know the credentials, lets see if we can use Zap to obtain credentials through a Brute-Force attack.

If you wanted to do this with BurpSuite, you'd need to intercept the request, and then pass it to Hydra. However, this process is much easier with ZAP!



Vulnerability: Brute Force

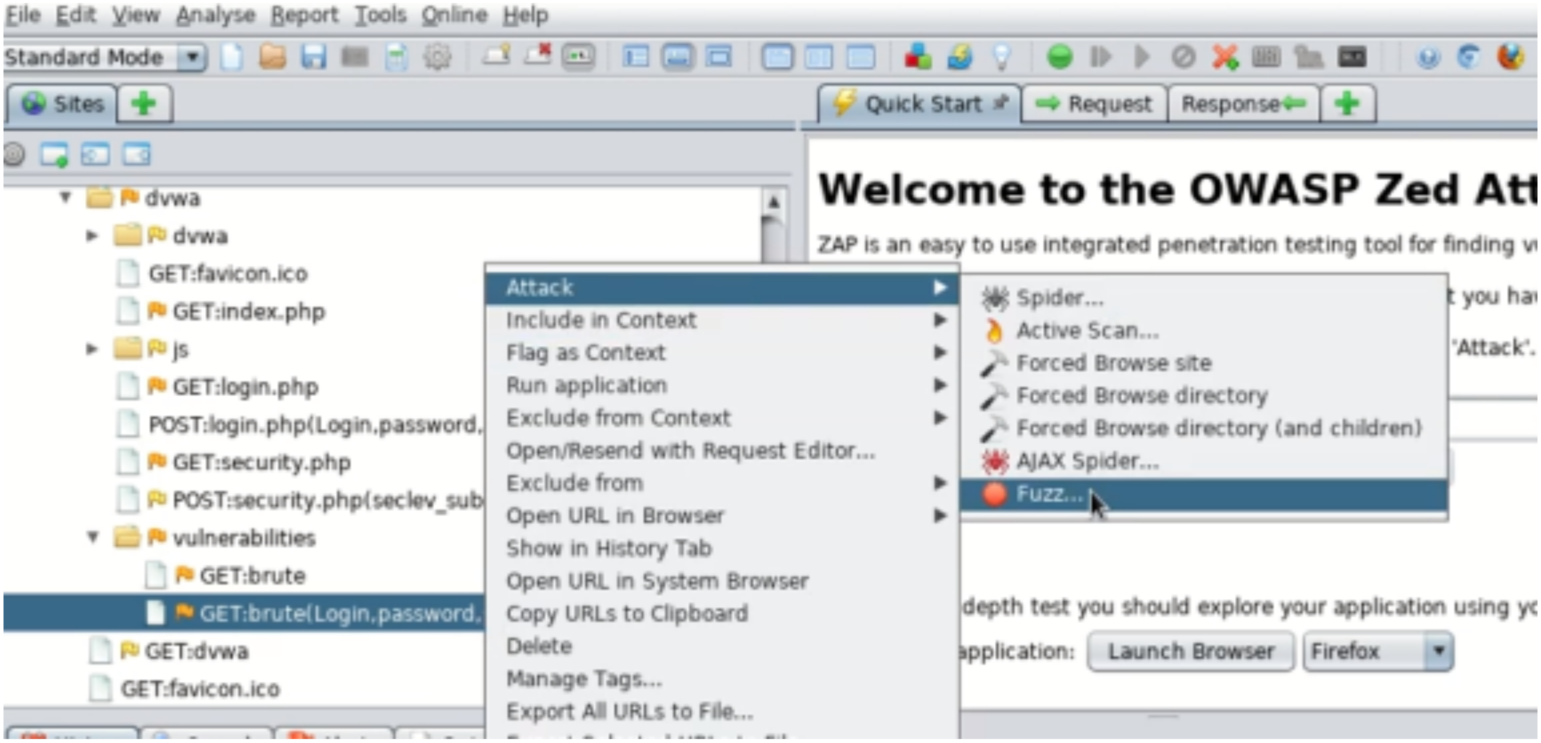
Login

Username: admin

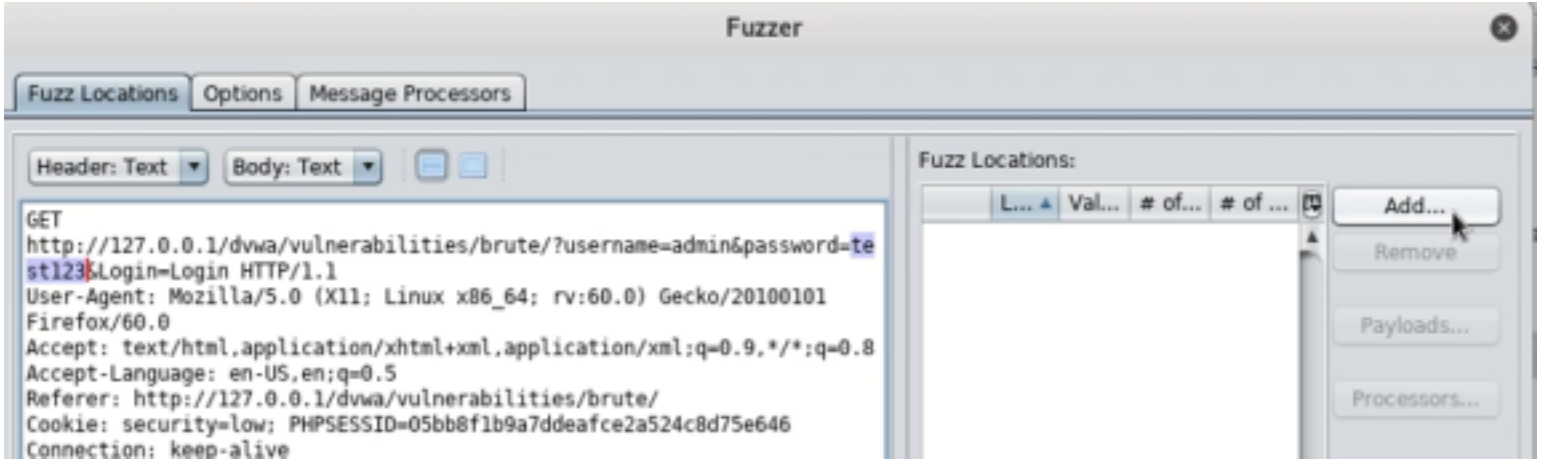
Password: •••••••

Username and/or password incorrect.

Navigate to the Brute Force page on DVWA and attempt login as “admin” with the password “test123”

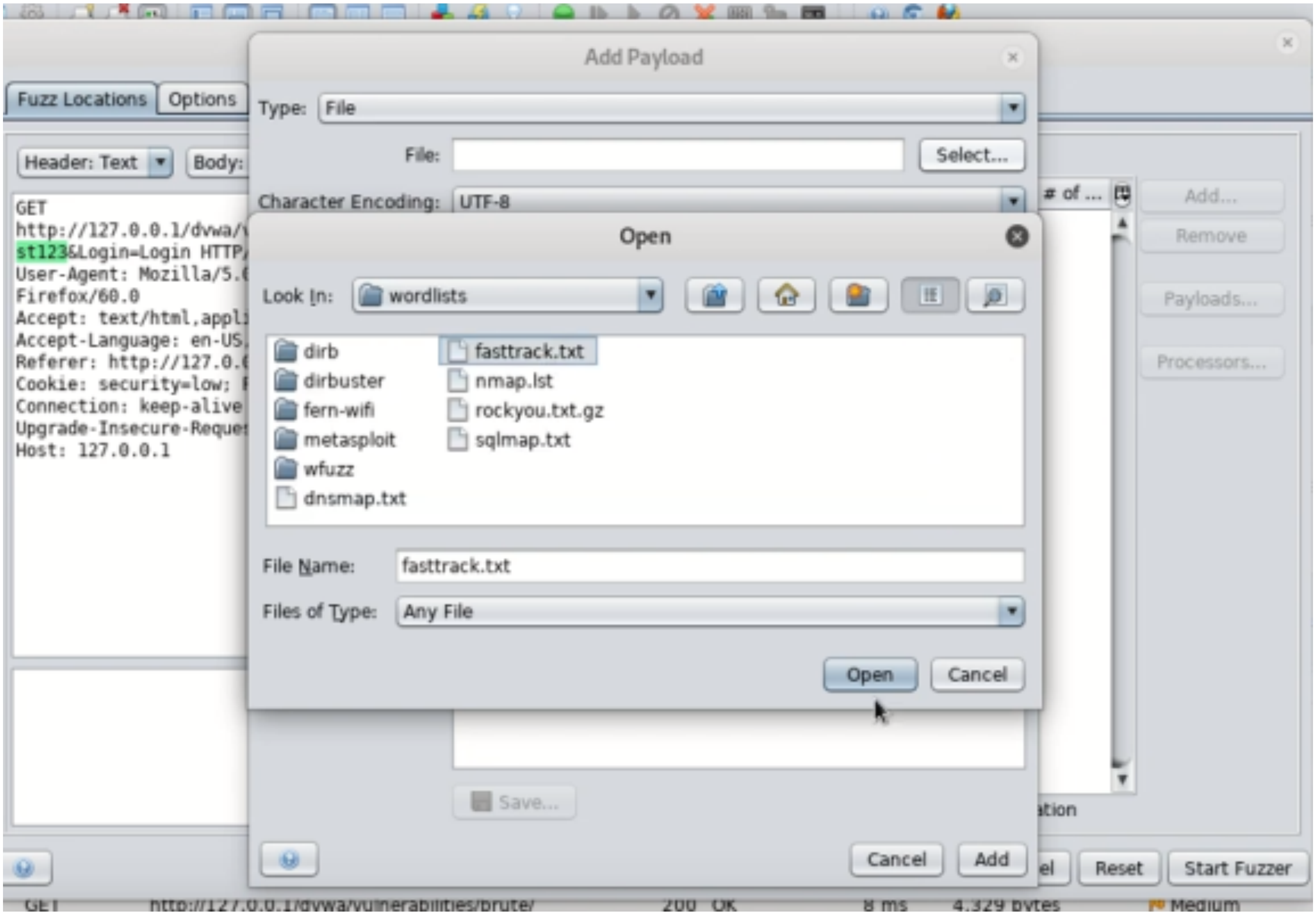


Then, find the GET request and open the Fuzz menu.



```
GET
http://127.0.0.1/dvwa/vulnerabilities/brute/?username=admin&password=te
st123&Login=Login HTTP/1.1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101
Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Referer: http://127.0.0.1/dvwa/vulnerabilities/brute/
Cookie: security=low; PHPSESSID=05bb8f1b9a7ddeafce2a524c8d75e646
Connection: keep-alive
```

Then highlight the password you attempted and add a wordlist. This selects the area of the request you wish to replace with other data.



```
GET
http://127.0.0.1/dvwa/\
st123&Login=Login HTTP/
User-Agent: Mozilla/5.(
Firefox/60.0
Accept: text/html,appl:
Accept-Language: en-US,
Referer: http://127.0.(
Cookie: security=low; F
Connection: keep-alive
Upgrade-Insecure-Reque:
Host: 127.0.0.1
```

Type: File

File:

Character Encoding: UTF-8

Look In: wordlists

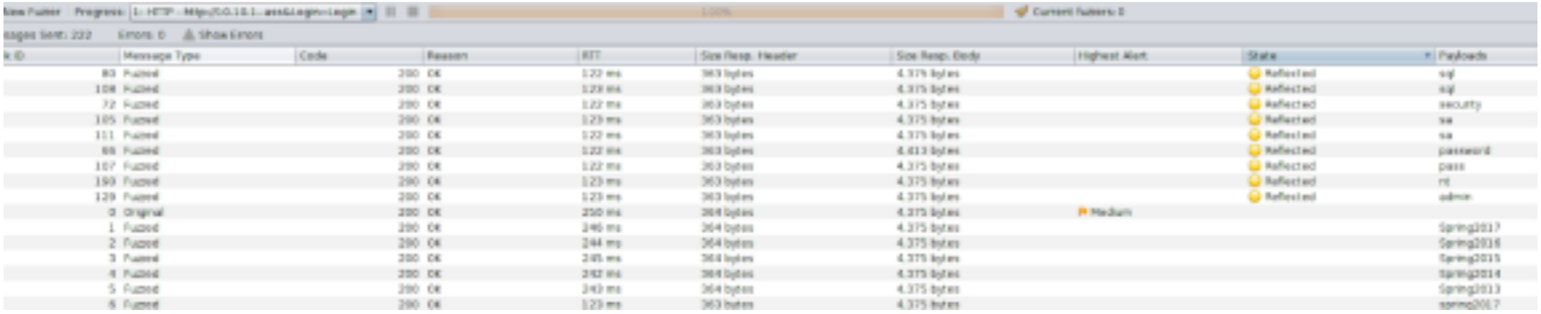
dirb
dirbuster
fern-wifi
metasploit
wfuzz
dnsmap.txt
fasttrack.txt
nmap.lst
rockyou.txt.gz
sqlmap.txt

File Name: fasttrack.txt

Files of Type: Any File

```
GET    http://127.0.0.1/dvwa/vulnerabilities/brute/    200  OK    8 ms    4,329 bytes    Medium
```

For speed we can use fasttrack.txt which is located in your /usr/share/wordlists if you’re using Kali Linux.


New Fuzzer Progress: 1: HTTP - http://10.10.1...ass&Login=Login 100% Current fuzzers: 0

Messages Sent: 222 Errors: 0 Show Errors

k ID	Message Type	Code	Reason	RTT	Size Resp. Header	Size Resp. Body	Highest Alert	State	Payloads
83	Fuzzed	200	OK	122 ms	363 bytes	4,375 bytes		Reflected	sql
108	Fuzzed	200	OK	123 ms	363 bytes	4,375 bytes		Reflected	sql
72	Fuzzed	200	OK	122 ms	363 bytes	4,375 bytes		Reflected	security
105	Fuzzed	200	OK	123 ms	363 bytes	4,375 bytes		Reflected	sa
111	Fuzzed	200	OK	122 ms	363 bytes	4,375 bytes		Reflected	sa
66	Fuzzed	200	OK	122 ms	363 bytes	4,413 bytes		Reflected	password
107	Fuzzed	200	OK	122 ms	363 bytes	4,375 bytes		Reflected	pass
190	Fuzzed	200	OK	123 ms	363 bytes	4,375 bytes		Reflected	rt
129	Fuzzed	200	OK	123 ms	363 bytes	4,375 bytes		Reflected	admin
0	Original	200	OK	250 ms	364 bytes	4,375 bytes	Medium		
1	Fuzzed	200	OK	246 ms	364 bytes	4,375 bytes			Spring2017
2	Fuzzed	200	OK	244 ms	364 bytes	4,375 bytes			Spring2016
3	Fuzzed	200	OK	245 ms	364 bytes	4,375 bytes			Spring2015
4	Fuzzed	200	OK	242 ms	364 bytes	4,375 bytes			Spring2014
5	Fuzzed	200	OK	243 ms	364 bytes	4,375 bytes			Spring2013
6	Fuzzed	200	OK	123 ms	363 bytes	4,375 bytes			spring2017

After running the fuzzer, sort the state tab to show Reflected results first. Sometimes you will get false-positives, but you can ignore the passwords that are less than 8 characters in length.

#1
Use ZAP to bruteforce the DVWA 'brute-force' page. What's the password?

password

[Task 9] ZAP Extensions

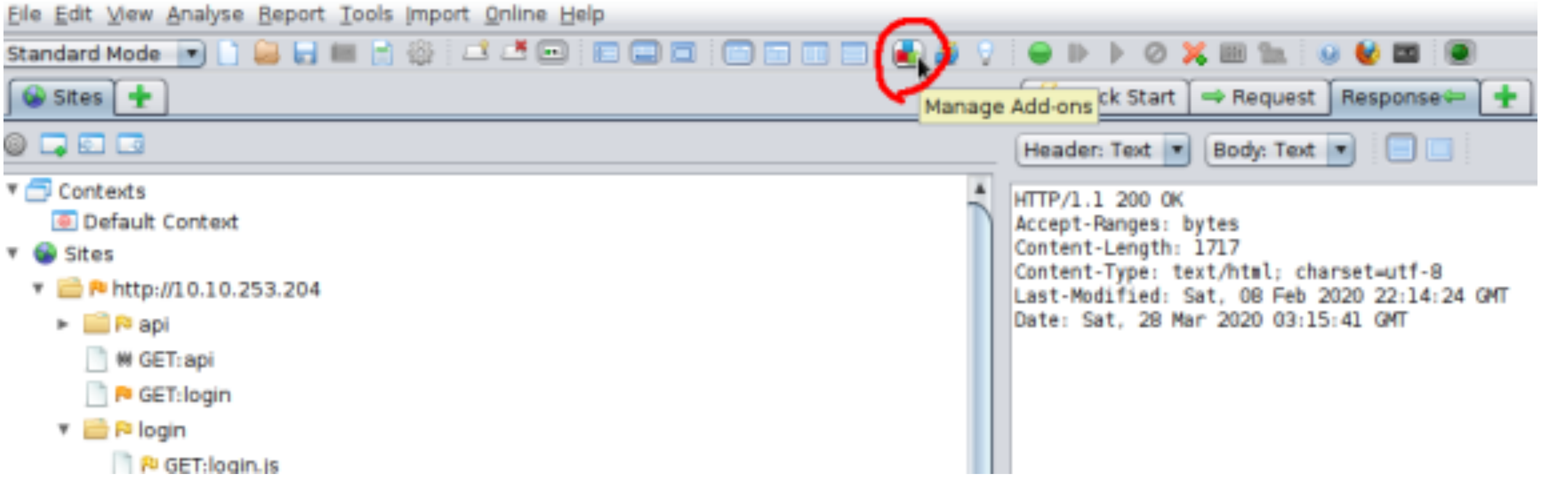
Want to further enhance ZAPs capabilities? Look at some of it’s downloadable extensions!

https://github.com/zaproxy/zap-extensions https://github.com/bugcrowd/HUNT

Let’s install the bugcrowd HUNT extensions for OWASP ZAP. This will passively scan for known vulnerabilities in web applications.

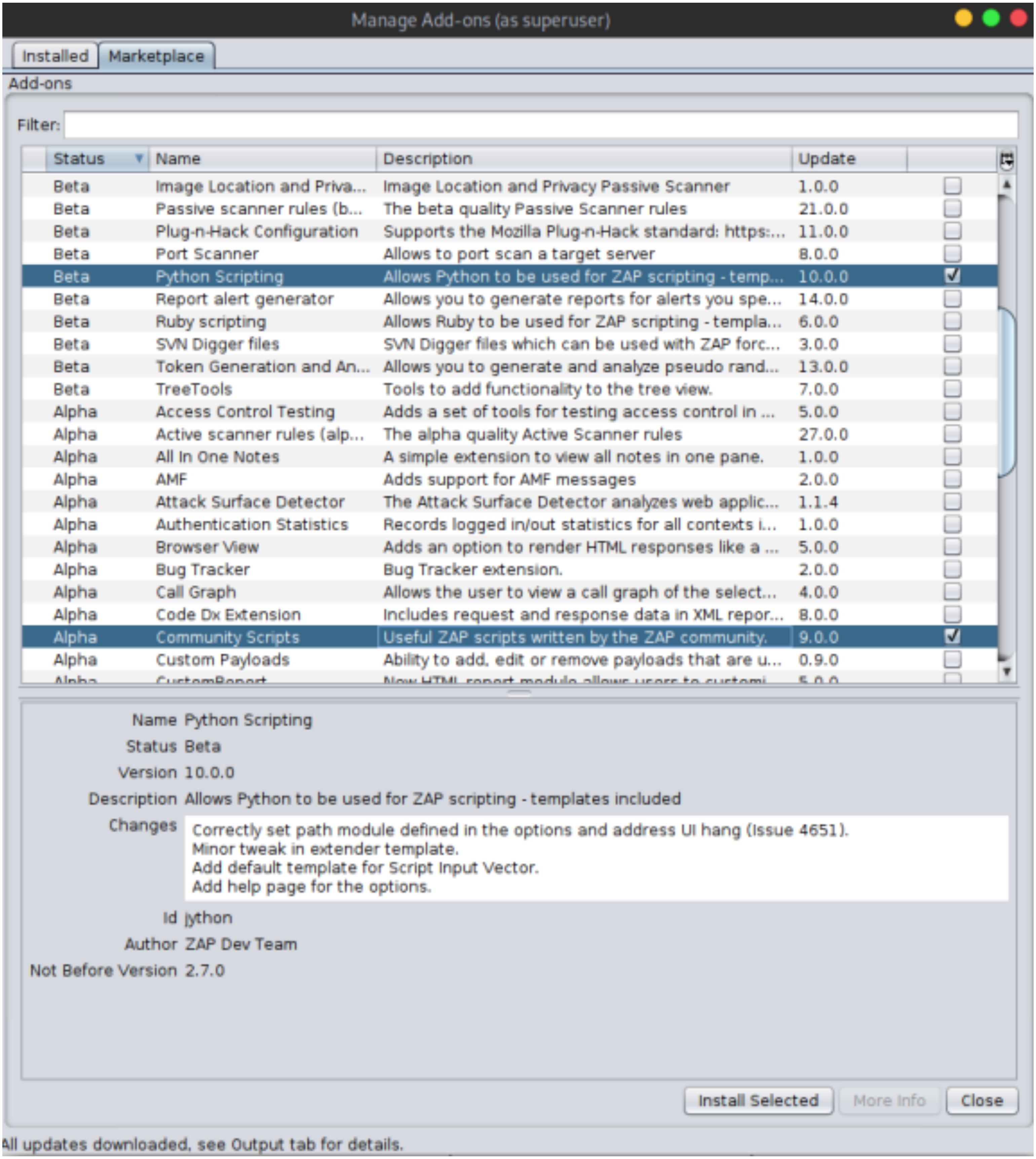
DISOBEY
25
DEF CON
HUNT
Data Driven Web Hacking & Manual Testing
@jhaddix @swagnetow @FatihEgbatan @digitalwoot @_Sha128
@bugcrowd

First navigate in your terminal somewhere you’d like to store the scripts
` git clone https://github.com/bugcrowd/HUNT `



```
HTTP/1.1 200 OK
Accept-Ranges: bytes
Content-Length: 1717
Content-Type: text/html; charset=utf-8
Last-Modified: Sat, 08 Feb 2020 22:14:24 GMT
Date: Sat, 28 Mar 2020 03:15:41 GMT
```

Then in ZAP click the “Manage Add-Ons” icon

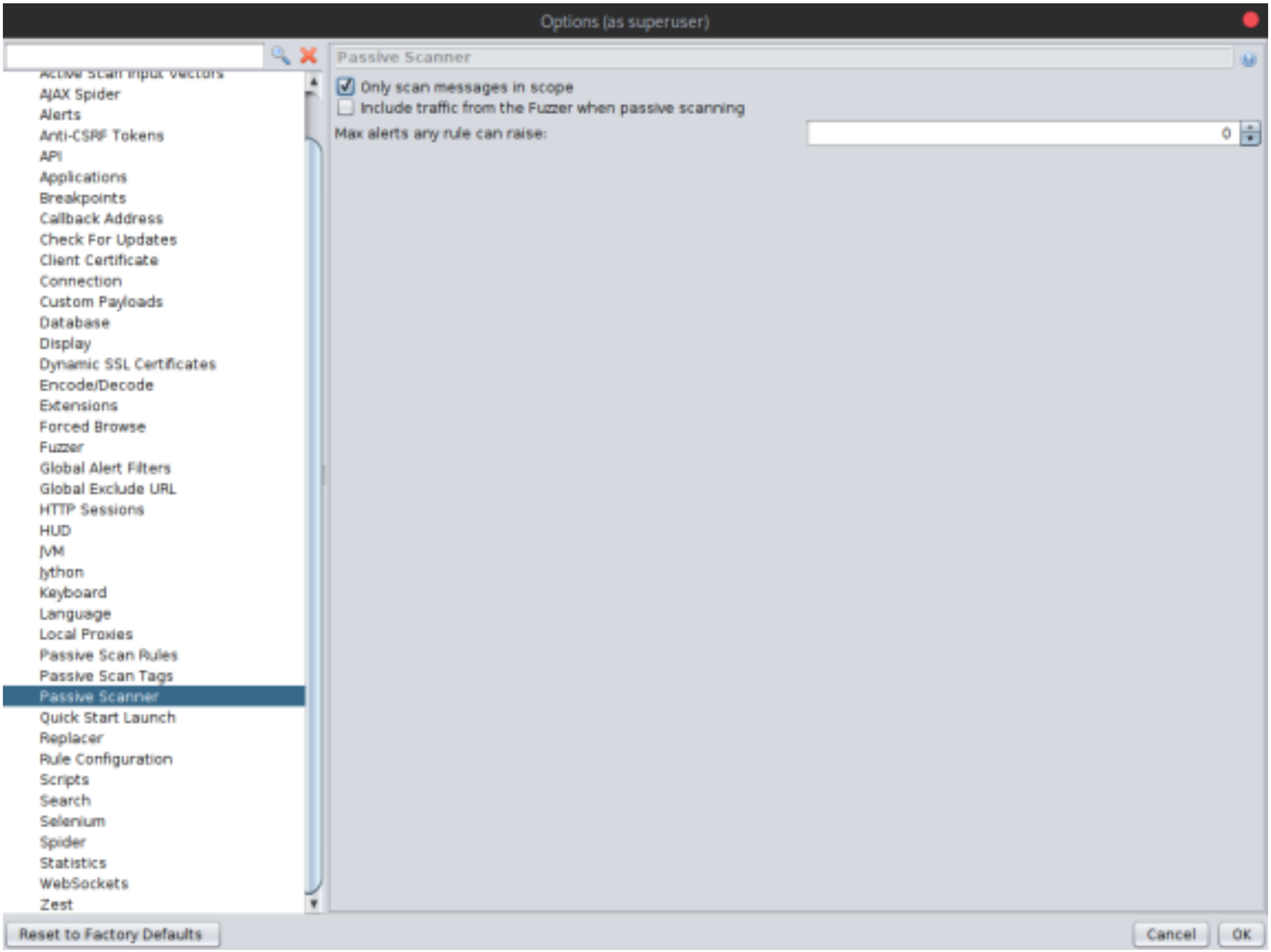


Status	Name	Description	Update	
Beta	Image Location and Priva...	Image Location and Privacy Passive Scanner	1.0.0	□
Beta	Passive scanner rules (b...	The beta quality Passive Scanner rules	21.0.0	□
Beta	Plug-n-Hack Configuration	Supports the Mozilla Plug-n-Hack standard: https:...	11.0.0	□
Beta	Port Scanner	Allows to port scan a target server	8.0.0	□
Beta	Python Scripting	Allows Python to be used for ZAP scripting - temp...	10.0.0	☑
Beta	Report alert generator	Allows you to generate reports for alerts you spe...	14.0.0	□
Beta	Ruby scripting	Allows Ruby to be used for ZAP scripting - templa...	6.0.0	□
Beta	SVN Digger files	SVN Digger files which can be used with ZAP forc...	3.0.0	□
Beta	Token Generation and An...	Allows you to generate and analyze pseudo rand...	13.0.0	□
Beta	TreeTools	Tools to add functionality to the tree view.	7.0.0	□
Alpha	Access Control Testing	Adds a set of tools for testing access control in ...	5.0.0	□
Alpha	Active scanner rules (alp...	The alpha quality Active Scanner rules	27.0.0	□
Alpha	All In One Notes	A simple extension to view all notes in one pane.	1.0.0	□
Alpha	AMF	Adds support for AMF messages	2.0.0	□
Alpha	Attack Surface Detector	The Attack Surface Detector analyzes web applic...	1.1.4	□
Alpha	Authentication Statistics	Records logged in/out statistics for all contexts i...	1.0.0	□
Alpha	Browser View	Adds an option to render HTML responses like a ...	5.0.0	□
Alpha	Bug Tracker	Bug Tracker extension.	2.0.0	□
Alpha	Call Graph	Allows the user to view a call graph of the select...	4.0.0	□
Alpha	Code Dx Extension	Includes request and response data in XML repor...	8.0.0	□
Alpha	Community Scripts	Useful ZAP scripts written by the ZAP community.	9.0.0	☑
Alpha	Custom Payloads	Ability to add, edit or remove payloads that are u...	0.9.0	□
Alpha	CustomReport	New HTML report module allows users to customi...	5.0.0	□

Name Python Scripting
Status Beta
Version 10.0.0
Description Allows Python to be used for ZAP scripting - templates included
Changes Correctly set path module defined in the options and address UI hang (Issue 4651).
Minor tweak in extender template.
Add default template for Script Input Vector.
Add help page for the options.
Id jython
Author ZAP Dev Team
Not Before Version 2.7.0

All updates downloaded, see Output tab for details.

From the Marketplace install “Python Scripting” and “Community Scripts”



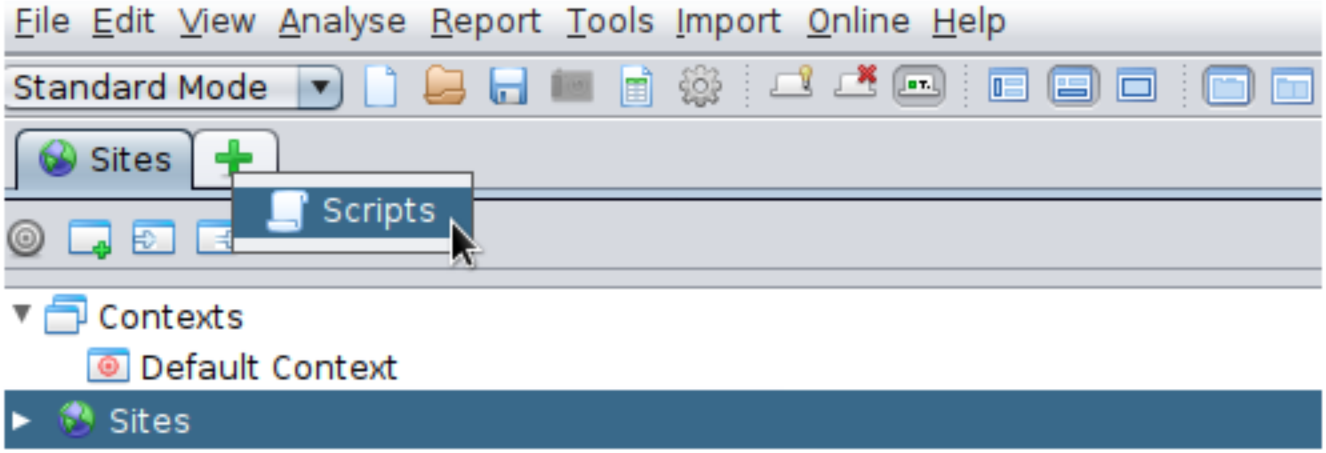
Passive Scanner

☑ Only scan messages in scope

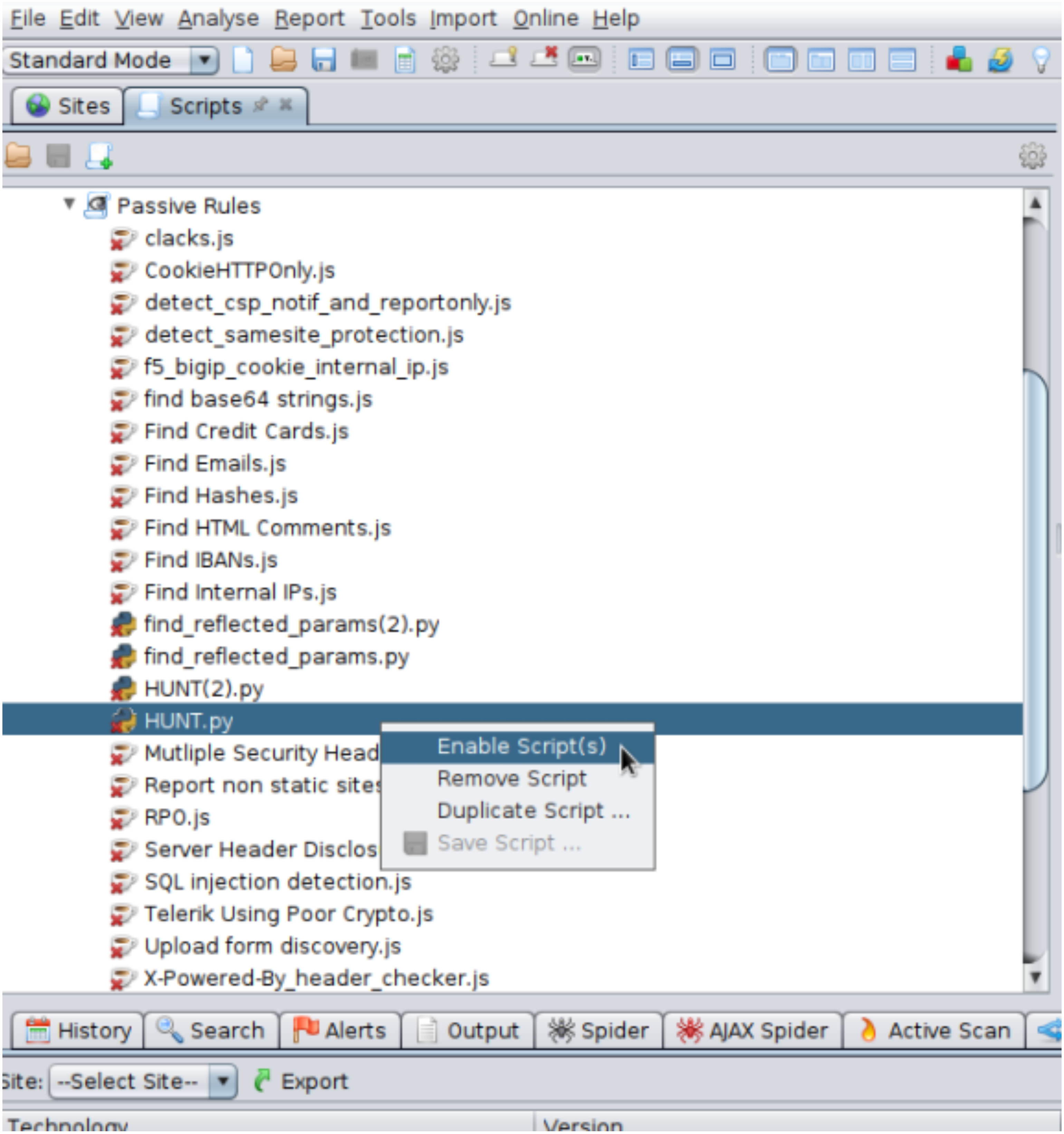
□ Include traffic from the Fuzzer when passive scanning

Max alerts any rule can raise: 0

In ZAP Options, under Passive Scanner, make sure “Only scan messages in scope” is enabled. Then hit OK.



In ZAP open the Scripts tab.



And under Passive Rules, find and enable the HUNT.py script

Now when you browse sites and HUNT will passively scan for SQLi, LFI, RFI, SSRF, and others. Exciting!

#1

Set up HUNT on your Zap application to automatically perform passive scans on sites you visit!

No answer needed

[Task 10] Further Reading

Wow! You reached the end! Good job! Try your new ZAP skills on some Web Application CTFs.

TryHackMe has quite the variety. My personal favorite is HackPark.

Desktop eManuel: https://www.zaproxy.org/docs/desktop/ui/

OWASP ZAP Forums: https://groups.google.com/forum/#!forum/zaproxy-users

Yeah that's pretty much all there is. I wasn't kidding when I said "microscopic" in comparison to Burp suite.

That's the one major con of ZAP is the pitiful amount of documentation there is. The project is still active and contributed to though. Just no one's really writing guides.

#1
Check out the additional reading material.