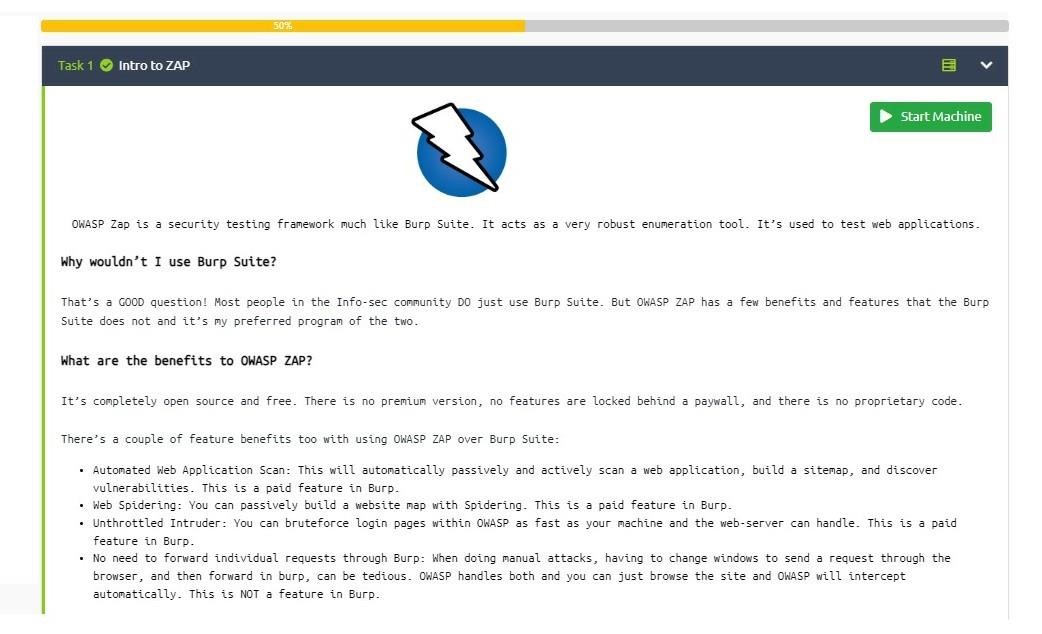
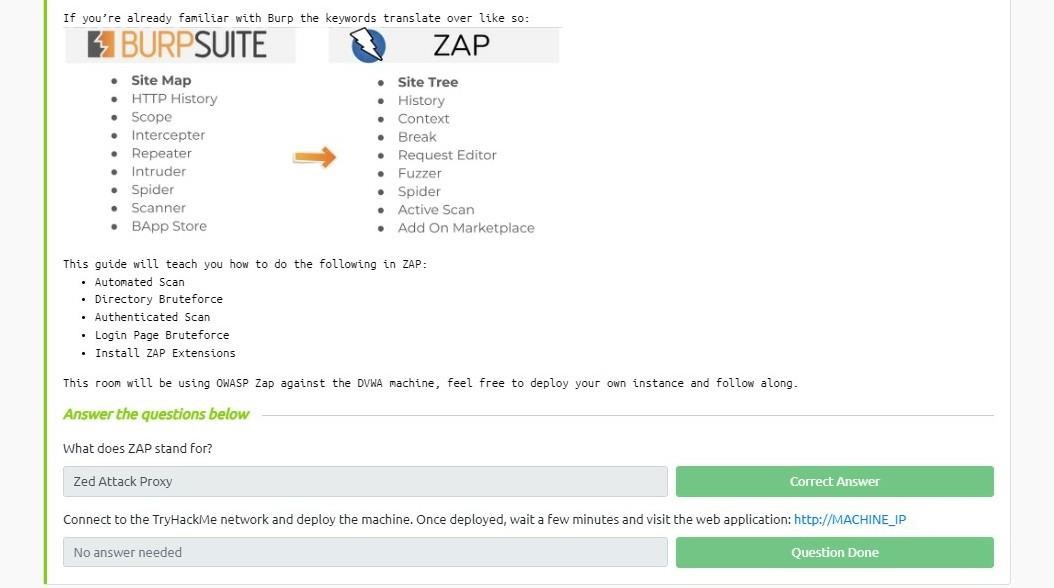
**Introduction to OWASP ZAP**

This report is a summary of my participation in the TryHackMe room "Introduction to OWASP ZAP." The purpose of the session was to familiarize attendees with the fundamentals of OWASP ZAP (Zed Attack Proxy) and its capabilities for web application security testing. I followed the directions to practice automatic scans and get hands-on experience with ZAP around the entire area.

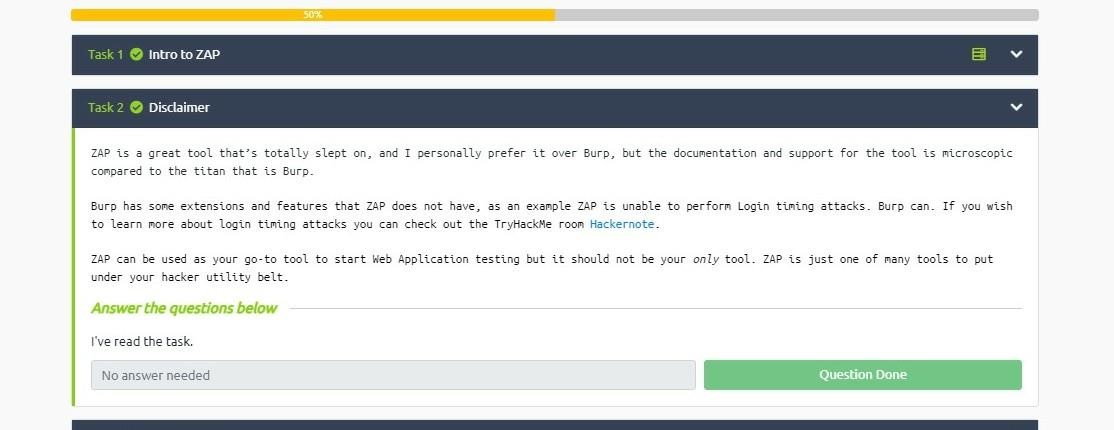
# **Task 1:** ZAP introduction





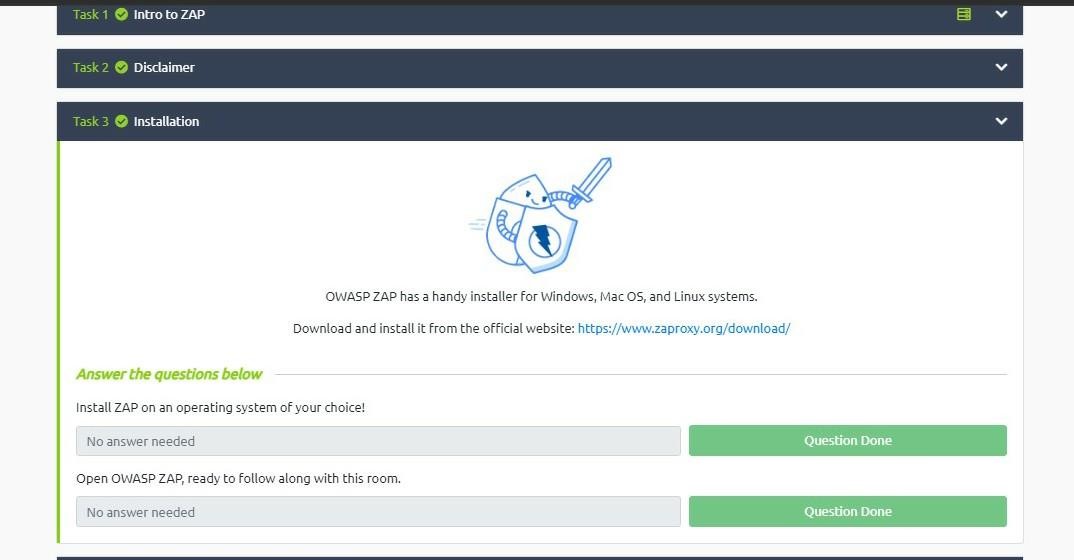
This portion taught me about the underlying ideas of OWASP ZAP and how important it is for finding flaws in web applications. The details addressed different scanning methods, proxy capabilities, and how ZAP may be a crucial weapon in a penetration tester's toolbox.

# **Task 2:** Acknowledgment



I realized the significance of doing security testing only on authorized systems and with the appropriate authorization before moving on. I acknowledged the ethical obligations related to employing ZAP and consented to do so under the TryHackMe setting.

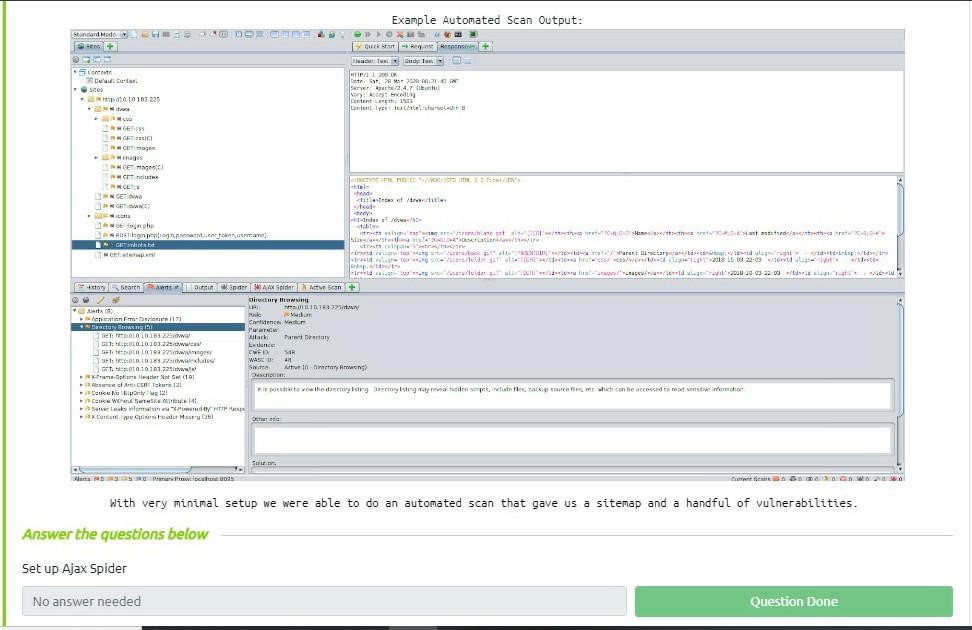
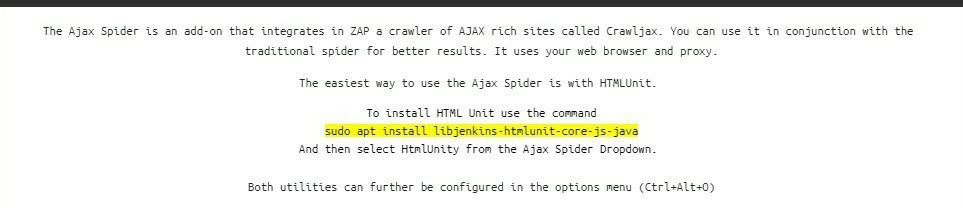
# **Task 3:** Installation



I followed the instructions to install OWASP ZAP on my choice operating system. The installation process went well and without any problems.

After the installation went smoothly, I started OWASP ZAP and made sure it was prepared to be utilized in the upcoming tasks. A variety of tools and options for testing web apps were provided, and the user interface looked to be easy.

# **Task 4:** How to Run an Automatic Scan



I set up the Ajax Spider capability in OWASP ZAP according to the instructions in order to launch an automated scan. This technology made it possible to automatically crawl and scan web applications.

**Observations**

I identified the following crucial details during the Ajax Spider setup:

* Identifying and scanning dynamically produced material on web sites is made easier by the Ajax Spider function.
* I set the target URL or web application's IP address as the scope of the scan's configuration.
* As the Ajax Spider began to crawl the given destination, it found several pages and endpoints.
* Sending queries and examining answers were part of the scanning procedure to find possible vulnerabilities.
* I noticed that the amount of requests made and vulnerabilities discovered were indicated in the OWASP ZAP interface's scan progress.

**Conclusion**

It was a rewarding experience to take part in the "Introduction to OWASP ZAP" TryHackMe room. I obtained a thorough grasp of OWASP ZAP's features and how automated web application security testing can be done using it. I was able to properly discover possible vulnerabilities thanks to the Ajax Spider feature's insights into the scanning procedure.

Both newcomers and seasoned pros in the realm of web application security can benefit from OWASP ZAP's user-friendly interface and powerful scanning features. I'm eager to learn more about OWASP ZAP's capabilities and extend my expertise in upcoming TryHackMe rooms.

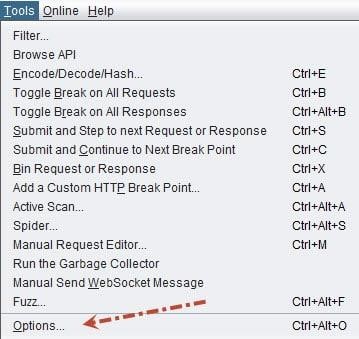
Note: Prior to doing security testing on any real-world online apps, it is crucial to adhere to ethical standards and secure the appropriate authorization.

Task 5:

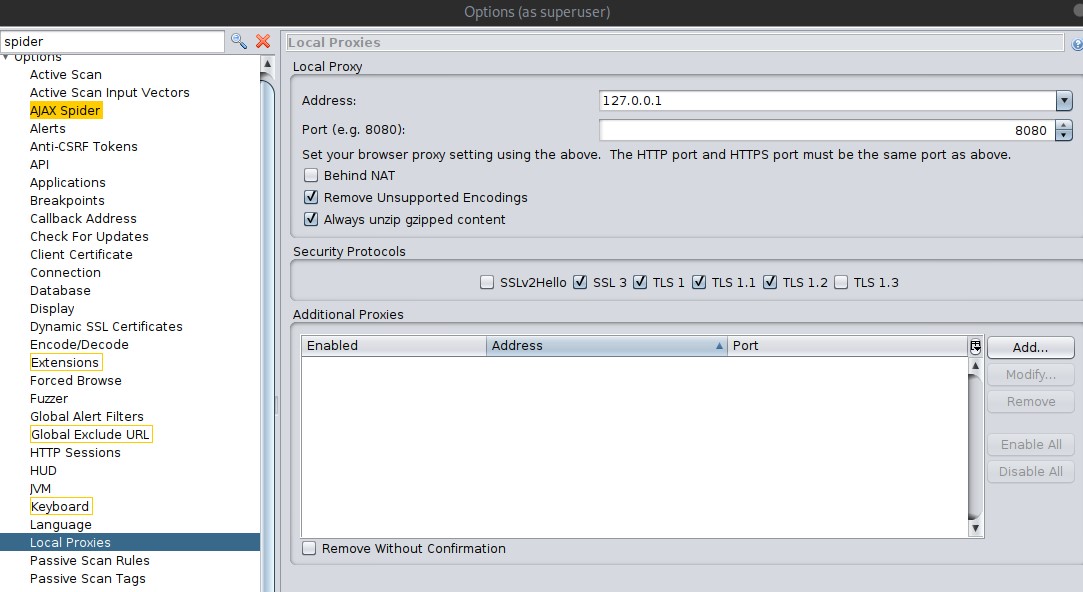
Manual Scanning

In this task we will perform Manual Scanning on DVWA machine.

**OWASP Proxy Setup:**



Open Options

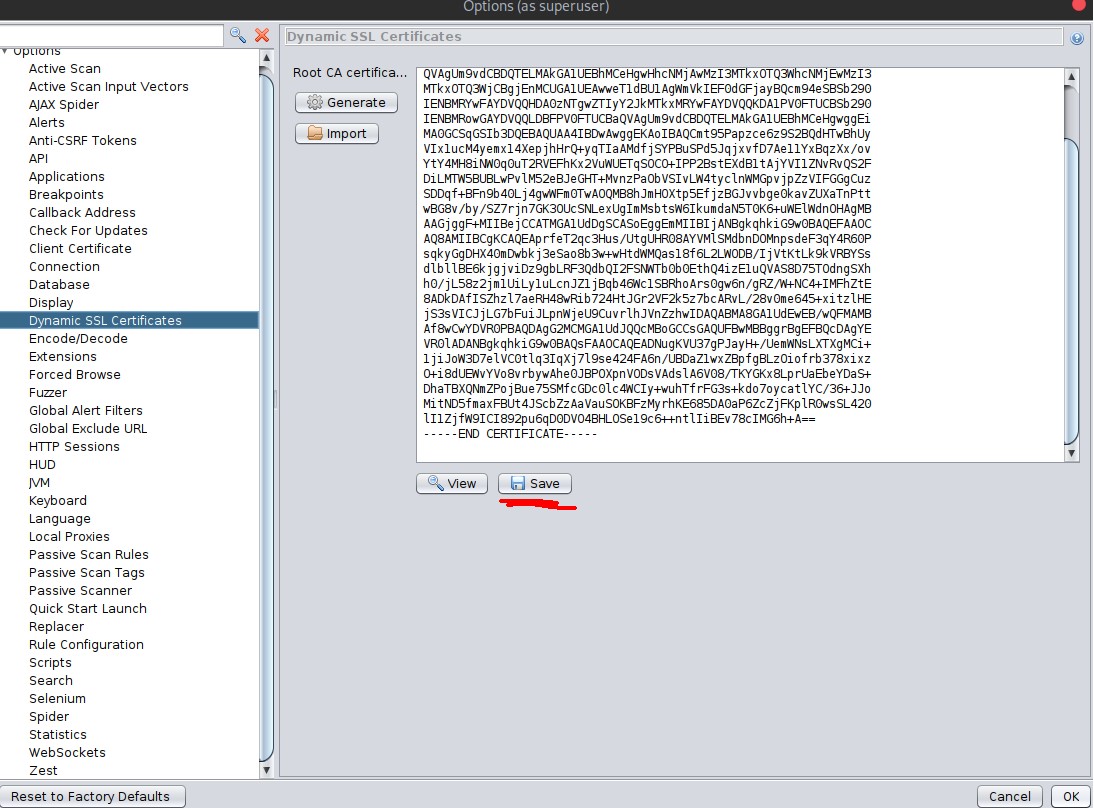


Change Local Proxy settings to the above.

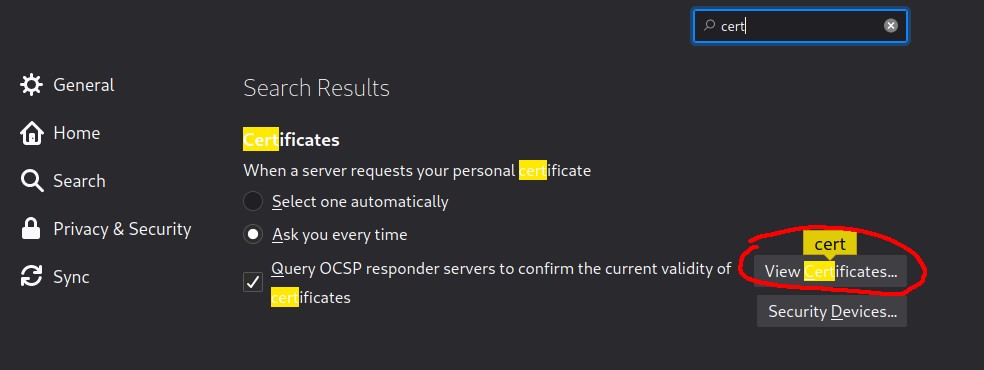
=============================================================

**Add ZAP Certificates:**

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

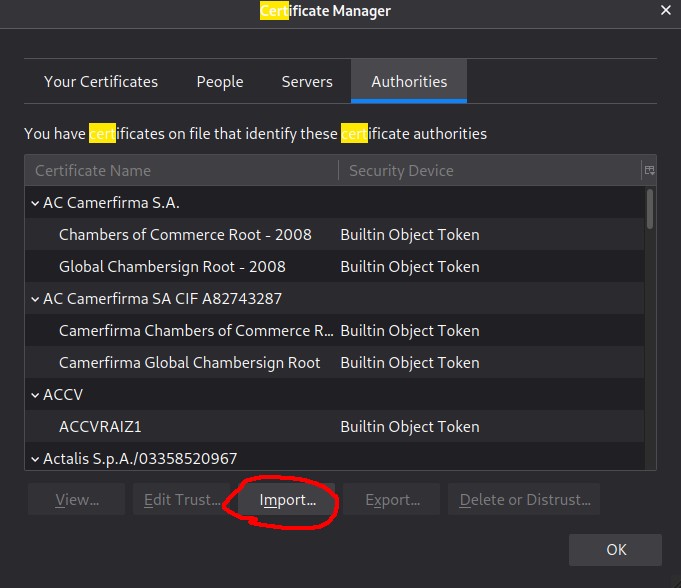


In the same options menu, navigate to Dynamic SSL Certificates and save the certificate somewhere you’ll remember and not delete.

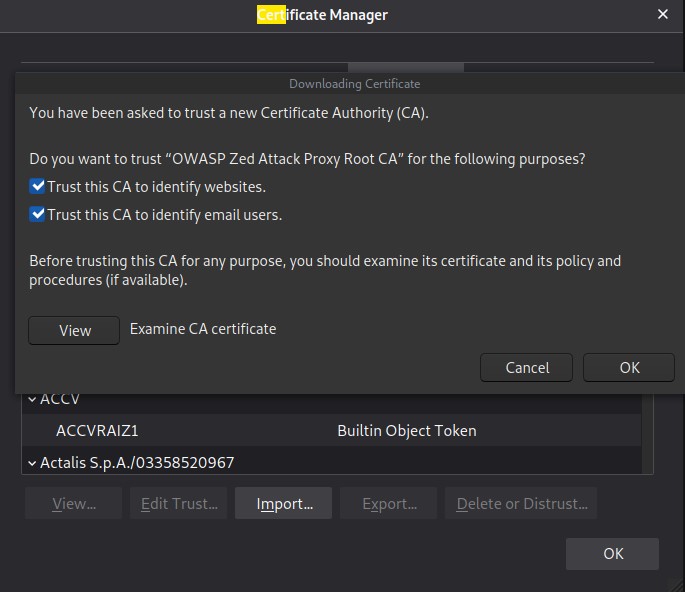


Then, open Firefox, navigate to your preferences, and search for certificates and click “View

Certificates”



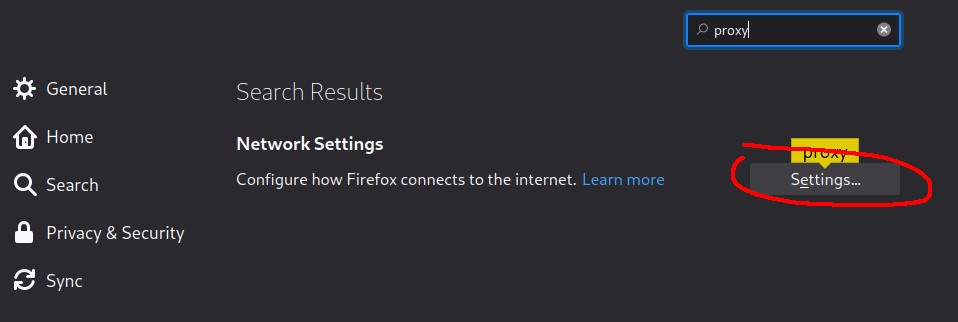
Then click “Import” and then navigate to the earlier downloaded certificate and open it.



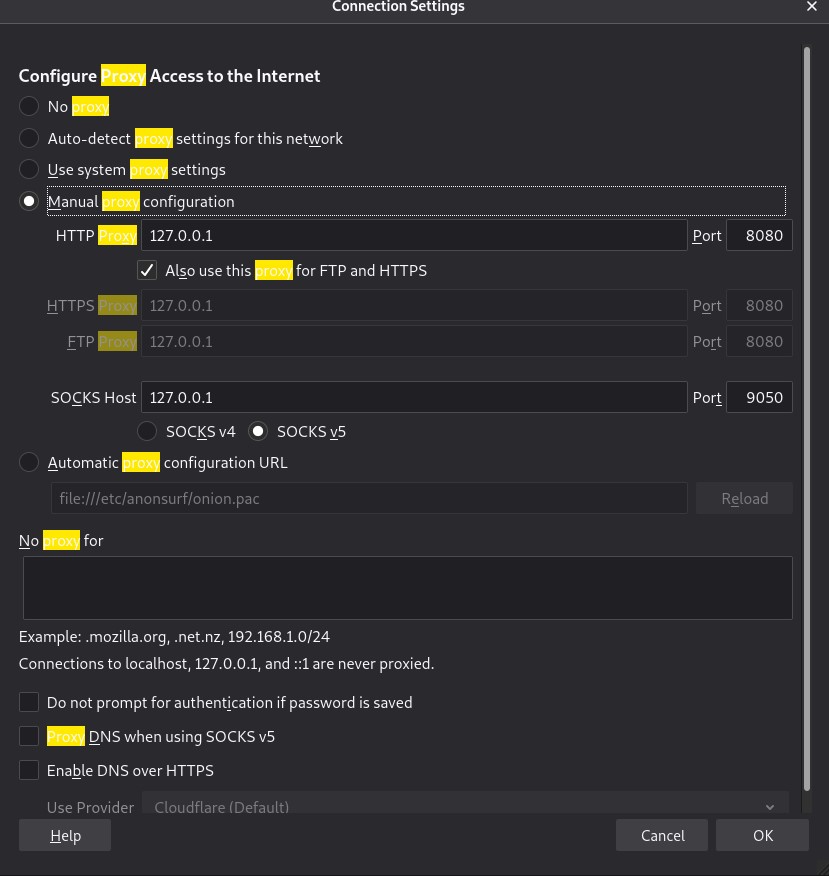
Select both and then hit OK.

=============================================================

**Firefox Proxy Setup:**



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



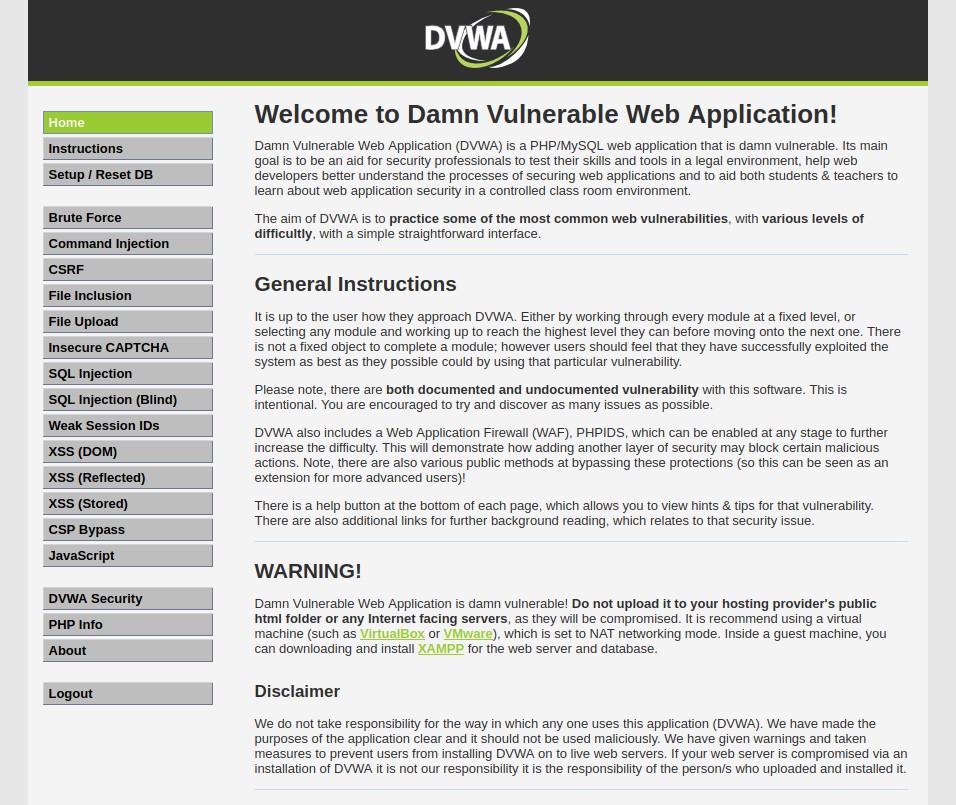
What IP do we use for the proxy?

127.0.0.1

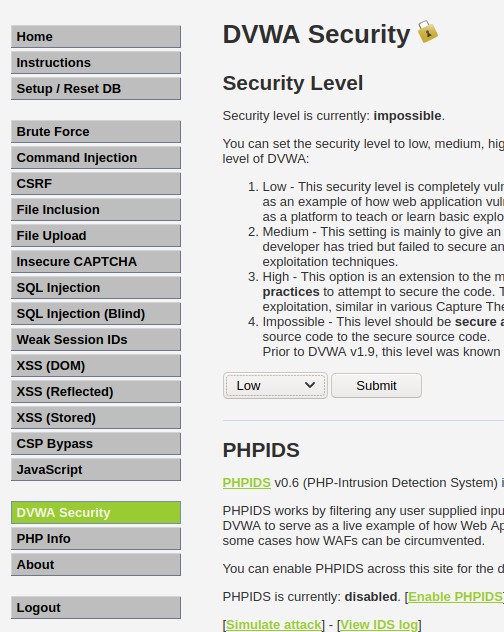
TASK 6:

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.

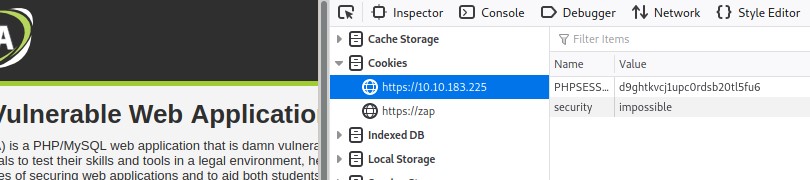


After logging in you should see this.

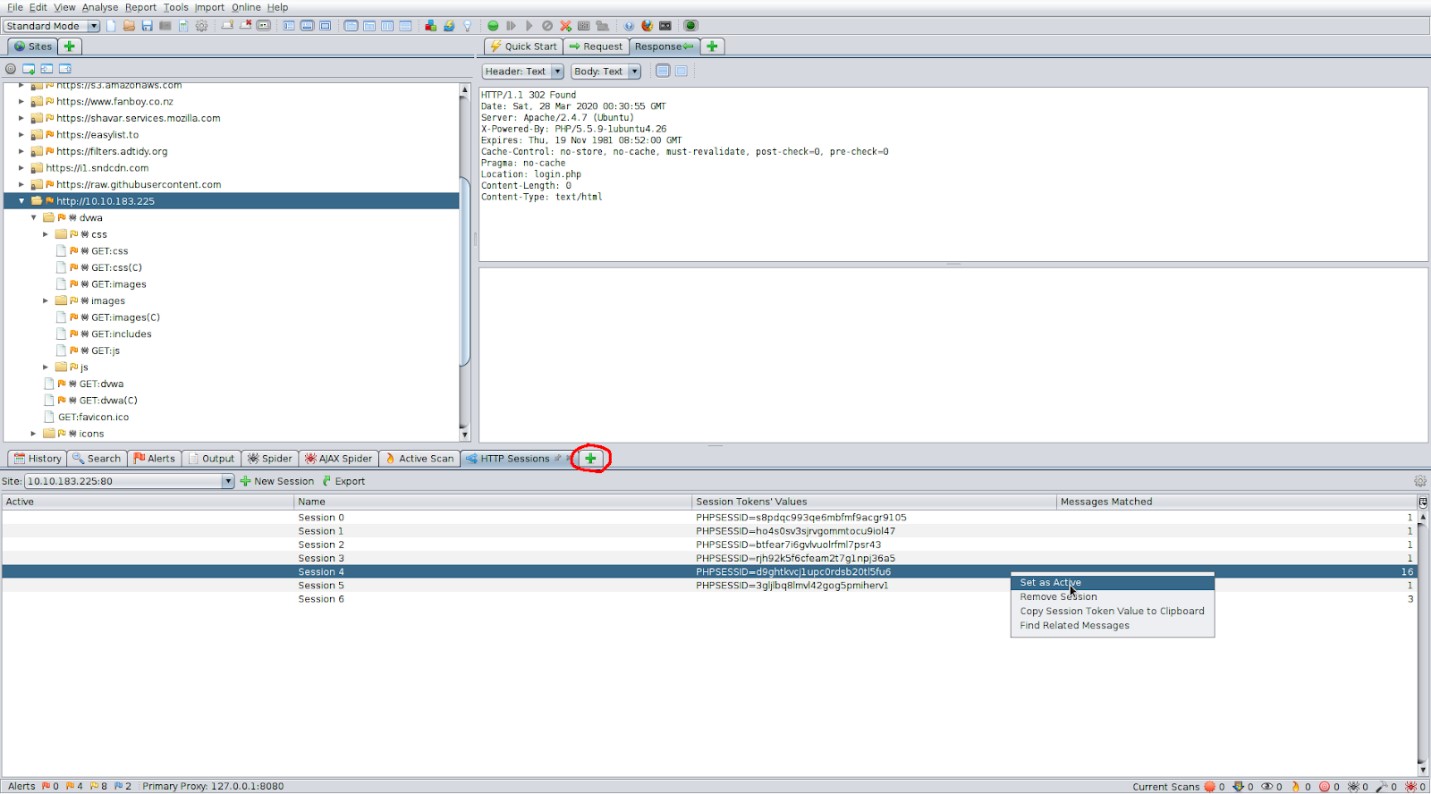


For the purpose of this exercise, once you've logged in, navigate to the DVWA Security tab and set the Security level to Low and then hit submit.

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



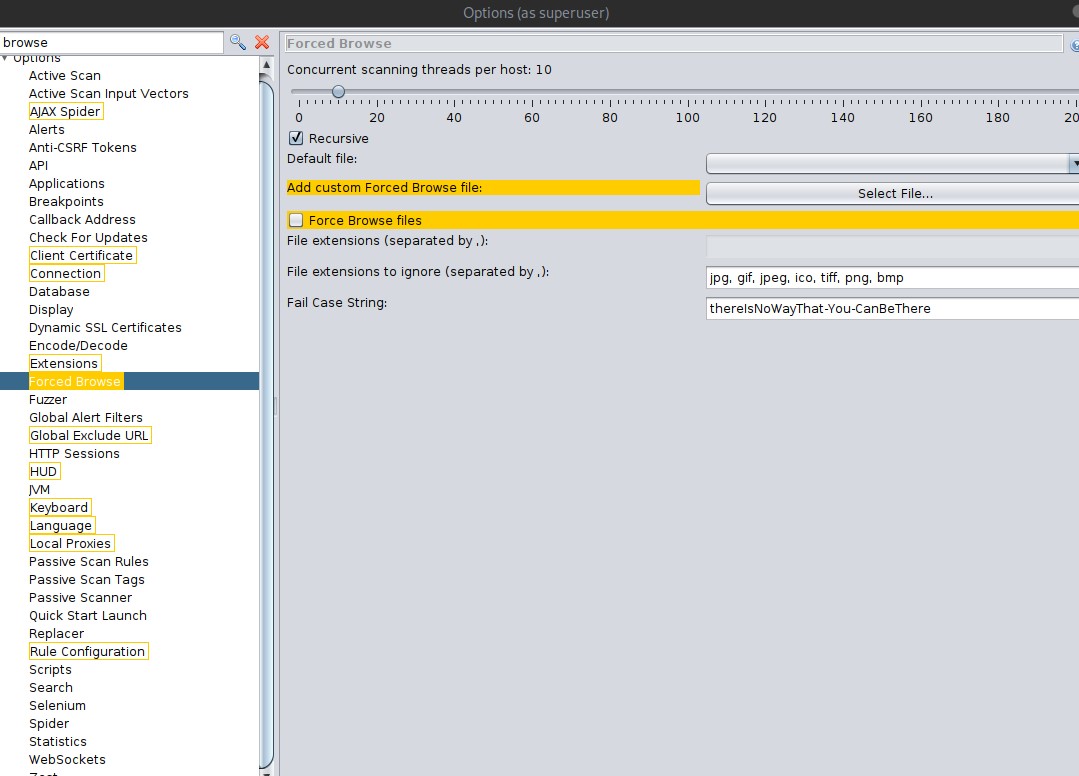
Enter inspect element and take note of your PHPSESSION cookie.



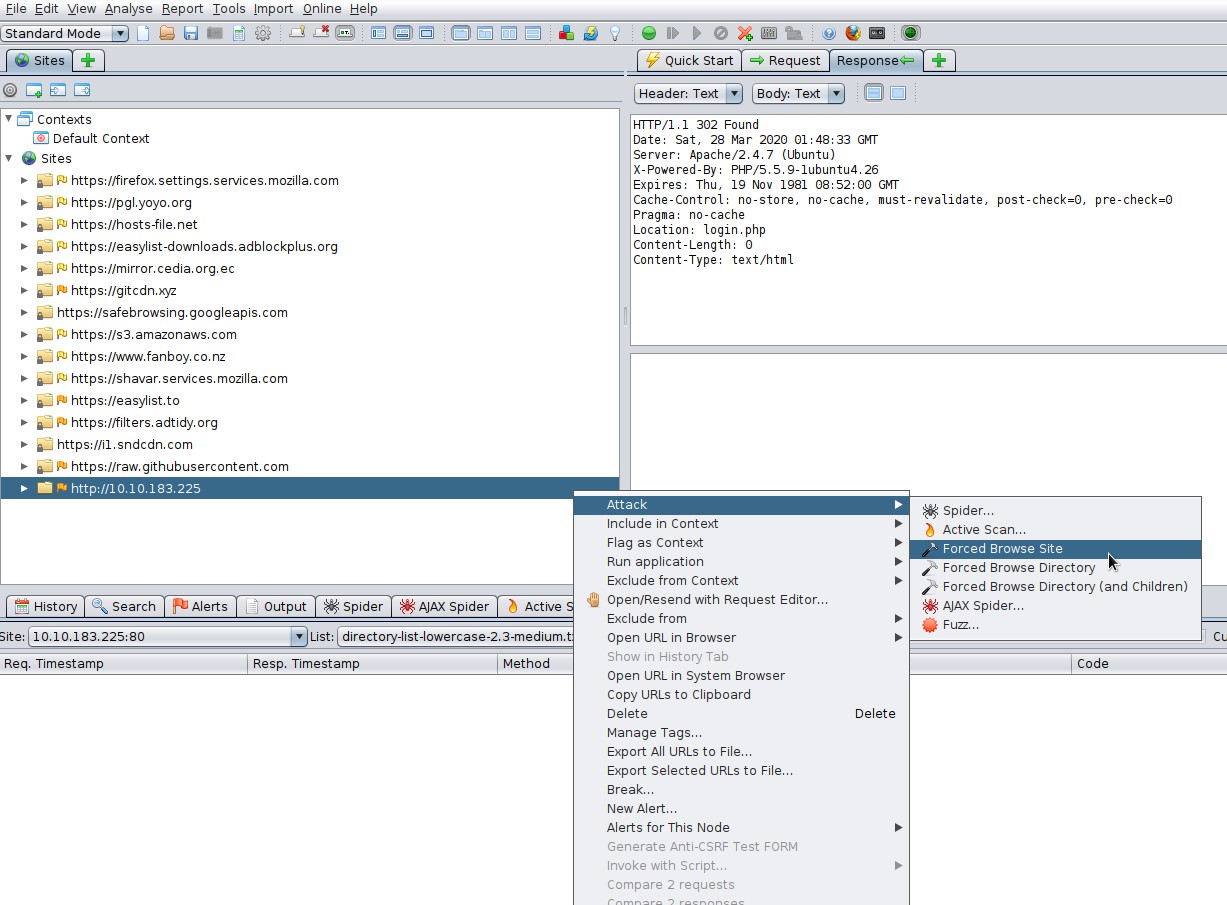
TASK 7:

# Brute-force Directories

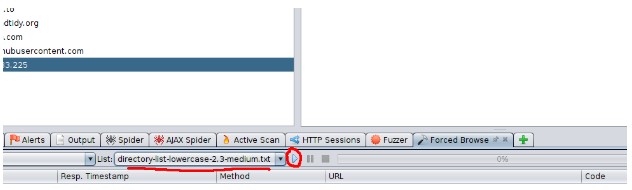
you can use a wordlist attack and directory bruteforce through ZAP just as you would with gobuster. This would pick up pages that are not indexed.



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.



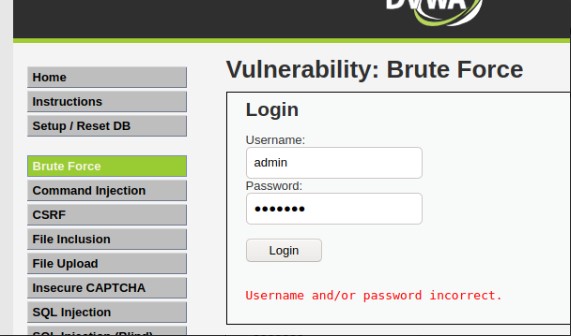
Then, right click the site->attack->forced browse site



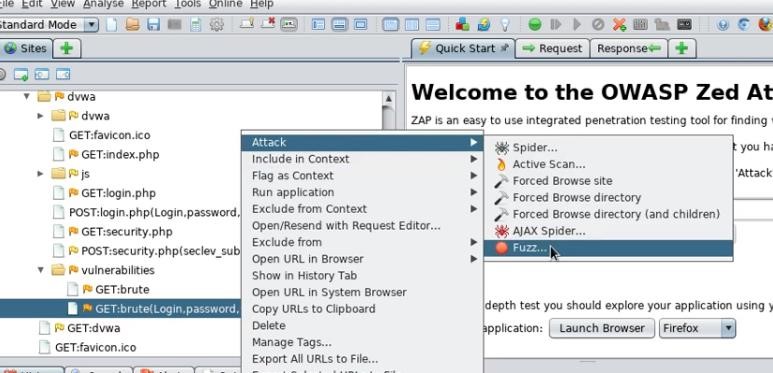
Task 8 - Bruteforce Web Login

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.

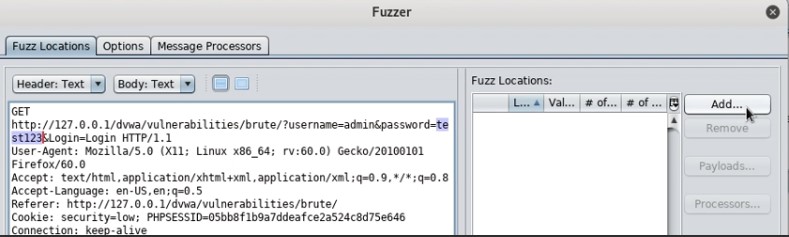
This process is much easier with ZAP!



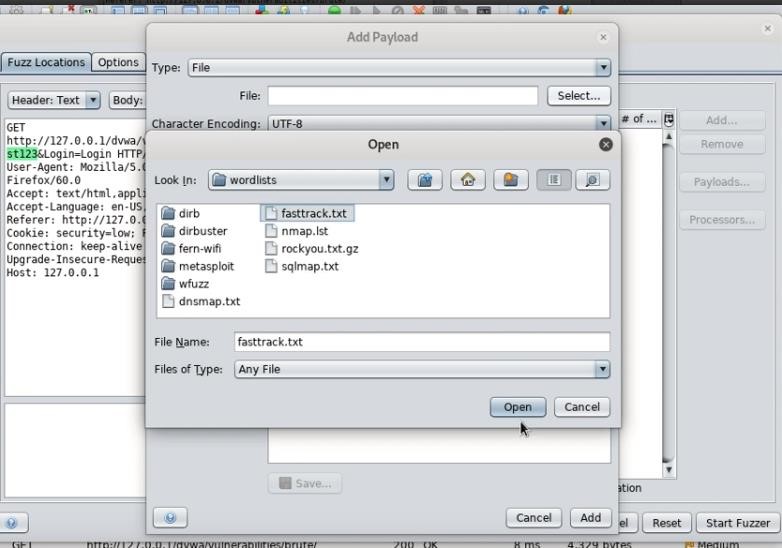
Navigate to the Brute Force page on DVWA and attempt login as “admin” with the password “test123”.



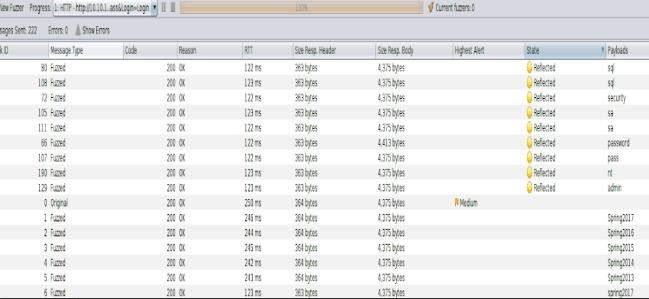
Now we will find the GET request and open the fuzz menu.



Now we will highlight the password we attempted and add a word list.



For Speed we can use fast track which is located in /usr/share/wordlists if using Kali Linux.



Q. Answer the questions

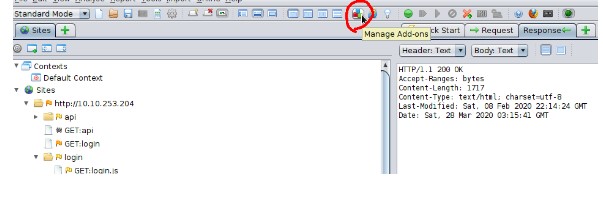
Use ZAP to bruteforce the DVWA 'brute-force' page. What's the password? password

Task 9 - ZAP Extensions

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



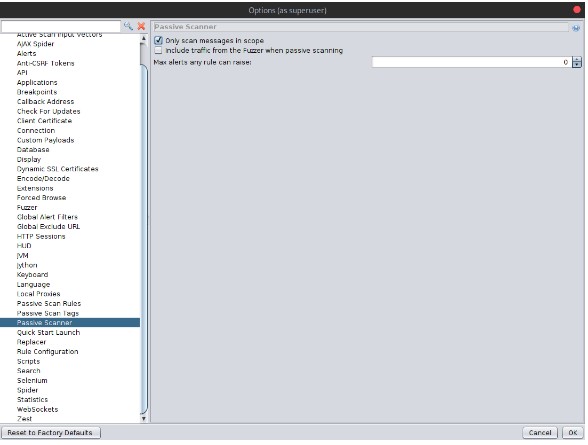
First navigate in your terminal somewhere you'd like to store the scripts https://github.com/bugcrowd/HUNT



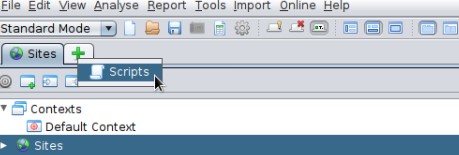
Then in ZAP click the "Manage Add- Ons" icon



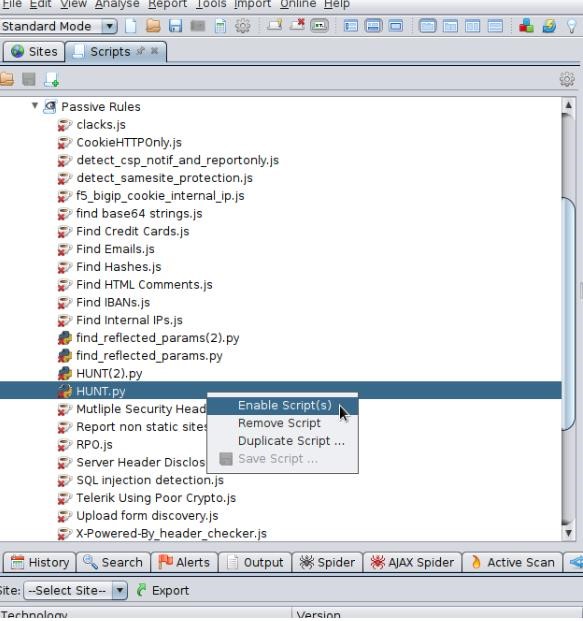
From the marketplace install " python scripting" and " community scripts"



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!