Exploring Web Application Vulnerabilities with OWASP ZAP

ANTHONY CIRILLO

TECHNICAL BACKGROUND

The motivation for choosing this topic

1. Growing Concern

➔ According to the 2022 Verizon DBIR, web application vulnerabilities were responsible for triggering over 90% of the 29,000 breaches examined.

2. Personal Experience

➔ Friend's hacked online shopping account resulted in financial and emotional distress, emphasizing the importance of proactive security measures

3. Industry Relevance

➔ Cybersecurity Ventures predicts global cybercrime costs will grow by 15% per year, reaching $10.5 trillion USD annually by 2025, up from $3 trillion USD in 2015.

Research ~ *What is OWASP ZAP?*

- **OWASP** - *'Open Web Application Security Project'*
- Companies should incorporate OWASP Zap for proactive identification and mitigation of vulnerabilities in web applications.
- OWASP Zap stands out among other tools like Burp Suite.
- It is open-source, cost-effective, and accessible for organizations.
- OWASP Zap offers comprehensive functionalities, including automated scanning, vulnerability detection, and powerful fuzzing capabilities.
- It emphasizes community-driven development and continuous improvement to stay updated with the latest security challenges.

Applied security concepts

WEB APP SECURITY

- safeguarding web applications from threats and vulnerabilities.
- Some examples of these threats:
  - Cross-site scripting
  - SQL injection
  - Cross-site request forgery
  - Path Traversal

HTTP

- *Hypertext Transfer Protocol*, used for communicating between web browsers and web servers.
- Understanding the basics of HTTP, such as request methods (GET, POST, etc.), status codes, headers, and cookies, is crucial.

PENTESTING

- involves simulating real-world attacks to identify and exploit vulnerabilities.
- OWASP ZAP's features e.g. active scanning, lets you simulate attacks and uncover exploitable vulnerabilities

DEFENSE-IN-DEPTH

- layering multiple security measures to provide comprehensive protection.
- OWASP ZAP provides mitigation methods for each vulnerability it exploits.

PREVIEW

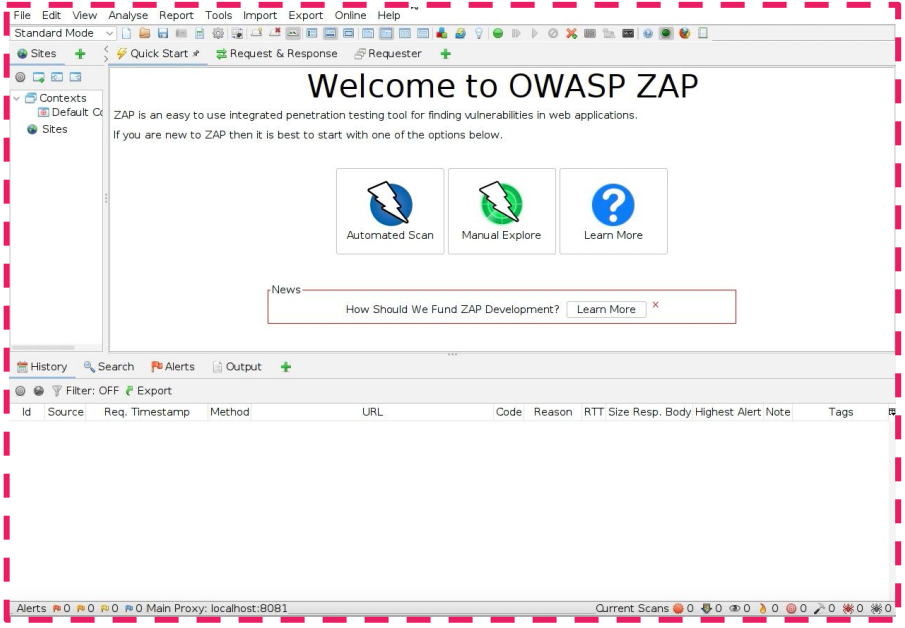
1. Install OWASP ZAP

➔ OWASP ZAP didn't come pre-installed on Ubuntu, so I ran the following to install it:

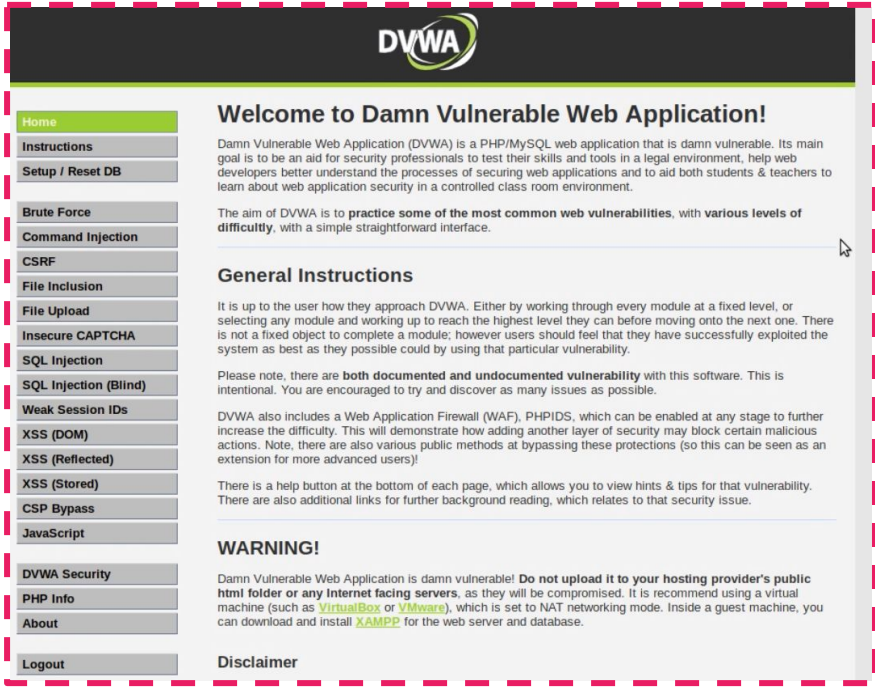
```
~$ snap install zaproxy
```

➔ Then, to open the graphical interface I ran:

```
~$ zaproxy
```



2. Set up DVWA



3. Configure ZAP and DVWA

➔ Setting up target URL + proxy settings for ZAP and web browser

4. Explore target application via Spider Tool

Spider identifies all accessible pages and helps build a comprehensive map of the application

5. Identify vulnerabilities via Active Scan

Active Scan automatically detects and exploits vulnerabilities in the target application

6. Review scan results via Alerts Tab

The Alerts Tab provides a detailed list of identified vulnerabilities, their severity levels, potential impact and mitigation techniques

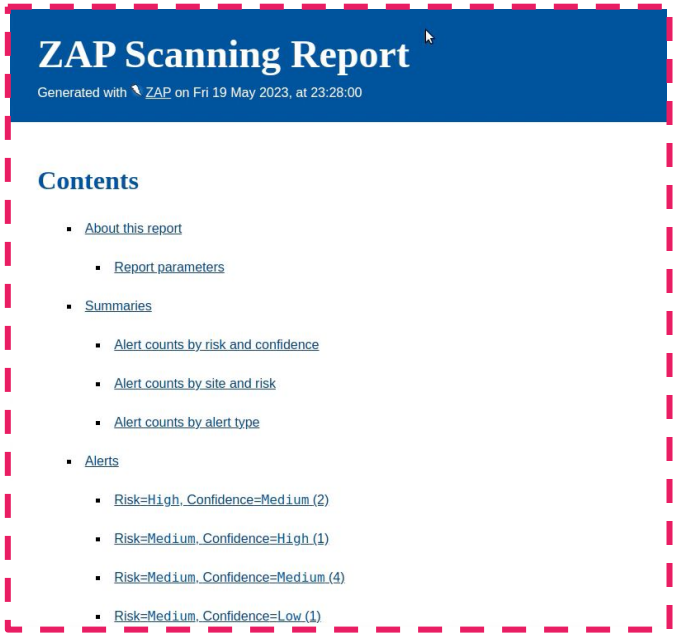
7. Manual exploitation with Fuzzer Tool

- ➔ Fuzzer automates the process of replacing specific strings in requests with different payloads
- ➔ I have prepared a text file containing various directory traversal payloads, which will be utilized by the fuzzer tool

Fuzzer

8. Generate Report

- ➔ Provides overview of identified vulnerabilities, their severity levels and recommended mitigation techniques
- ➔ Can be shared with stakeholders to facilitate resolutions



DEMONSTRATION

SUMMARY

What did we just observe?

1. Proxy configuration and synchronization with the browser
2. Spider tool: Mapping out website structure
3. Active Scan feature: Detecting and exploiting vulnerabilities
4. Fuzzer tool: Systematic testing for weaknesses
5. Path traversal vulnerability: Exploiting critical files
6. Organizing payloads by size for prioritization
7. Comprehensive report generation for communication and decision-making
8. Importance of web application security testing
9. Continuous monitoring and proactive security measures
10. Collaboration among developers, security professionals, and stakeholders

CLOSING THOUGHTS

What can we take away?

1. Heightened awareness of cybersecurity risks and proactive measures

2. Collaboration for robust security

3. Continuous monitoring, testing, and user education

4. Adaptation to evolving threats and societal impact

5. Impact of web app security on society and culture

THANKYOU