OREOLUWATOMIWA IBIKUNLE

C00253239

17/04/2023

CYBERCRIME & IT SECURITY

AUTO-OSINT & VULNERABILITY REPORT TOOL

Contents

Introduction

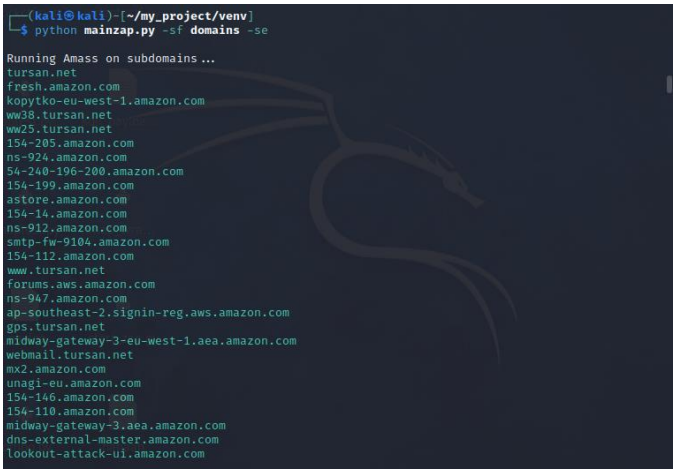
This report presents a comprehensive overview of a Python CLI tool designed to automate web application vulnerability management. The purpose of this document is to describe the journey of the project; its features, functionality and conformance to the original specification and design. The tool, currently called AVS, utilizes widely known open-source tools such as Amass, ZAP (Zed Attack Proxy) and Nmap, to deliver an efficient and user-friendly application for security professionals and developers in companies such as Unum which is actually where this idea stemmed from.

Description of Submitted Project

This tool comes with quite a good arsenal of features. It offers a robust and efficient solution to automate the vulnerability scanning of web applications.

Amass

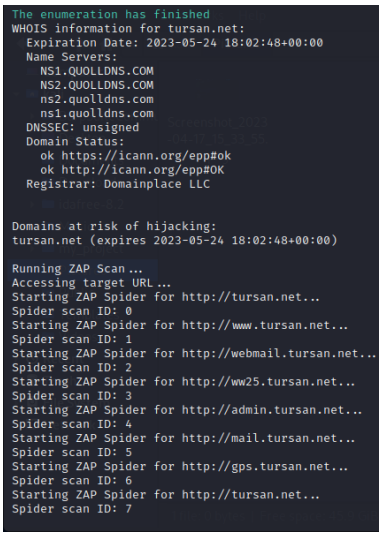
AVS begins by performing subdomain enumeration and whois on the given domain and vulnerability scanning with ZAP. Amass runs on both a single given domain and on subdomain files.



Whois

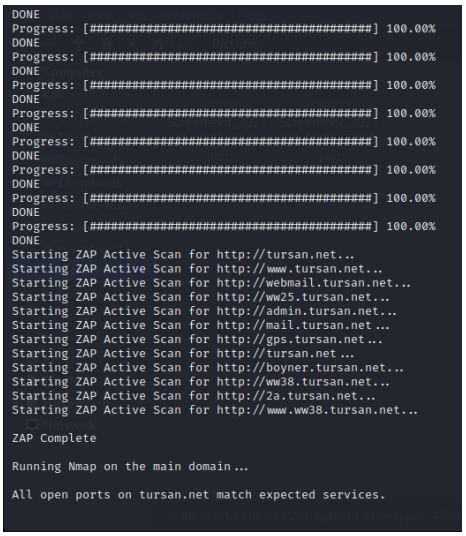
Amass and Whois scans occur concurrently as the information to be obtained from each of them is not dependent on each other.

The information obtained from WHOIS is used to check if the domain given is at risk of hijacking.



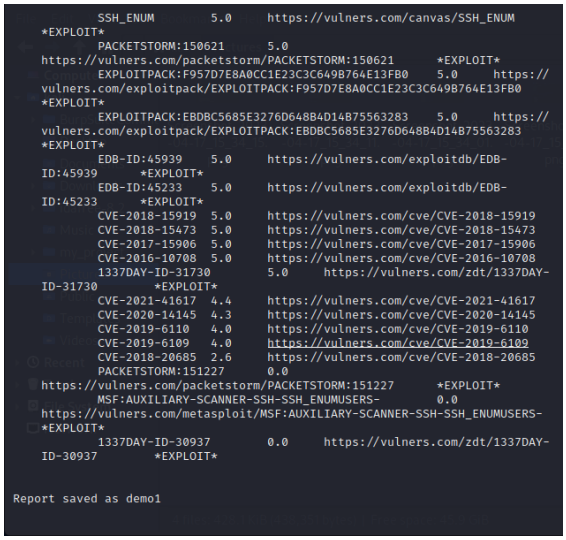
ZAP

The tool automates spider and active scans on the target URL and its subdomains and provides detailed results of identified security issues as well as a graph stating the Severity by Frequency just to give the user a visualization of their company's current security status.

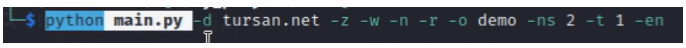


Nmap

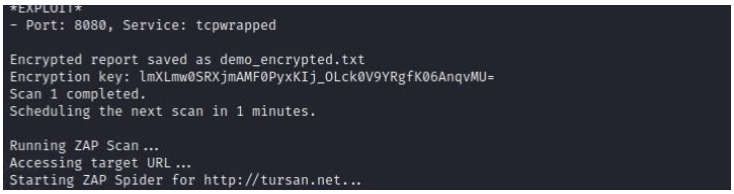
The tool then calls on Nmap to scan for open ports and the found ports and service combinations are matched with those from a database from IANA services to see if any unexpected services can be found on an open port. It goes further to check for vulnerabilities using Nmap's NSE script 'vulners' and considers header information to create a more well-rounded vulnerability report.



AES Encryption



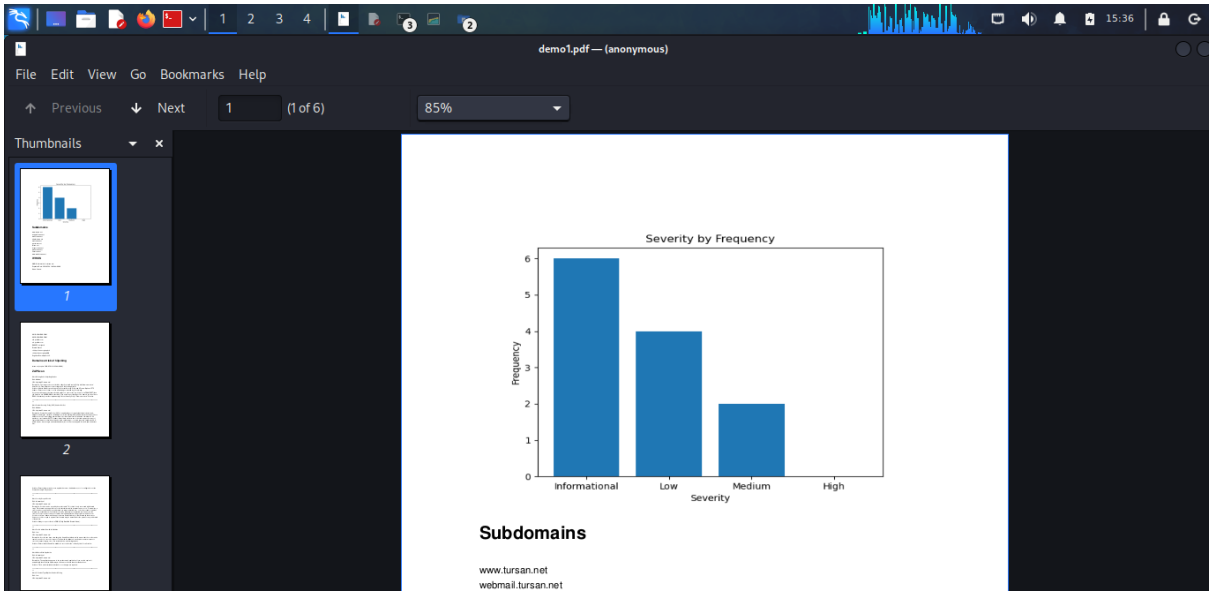
Another feature of the tool is that it can encrypt the files created or existing files and decrypt them using cryptography and keyring libraries in Python. The keys are given to the user for safekeeping for later decryption and a new key is provided for each new scan, i.e., when the user gives the command for 5 scans with a time interval of 10 mins, a new key will be generated for the file for each scan. In the diagrams, the command and the key can be clearly seen.

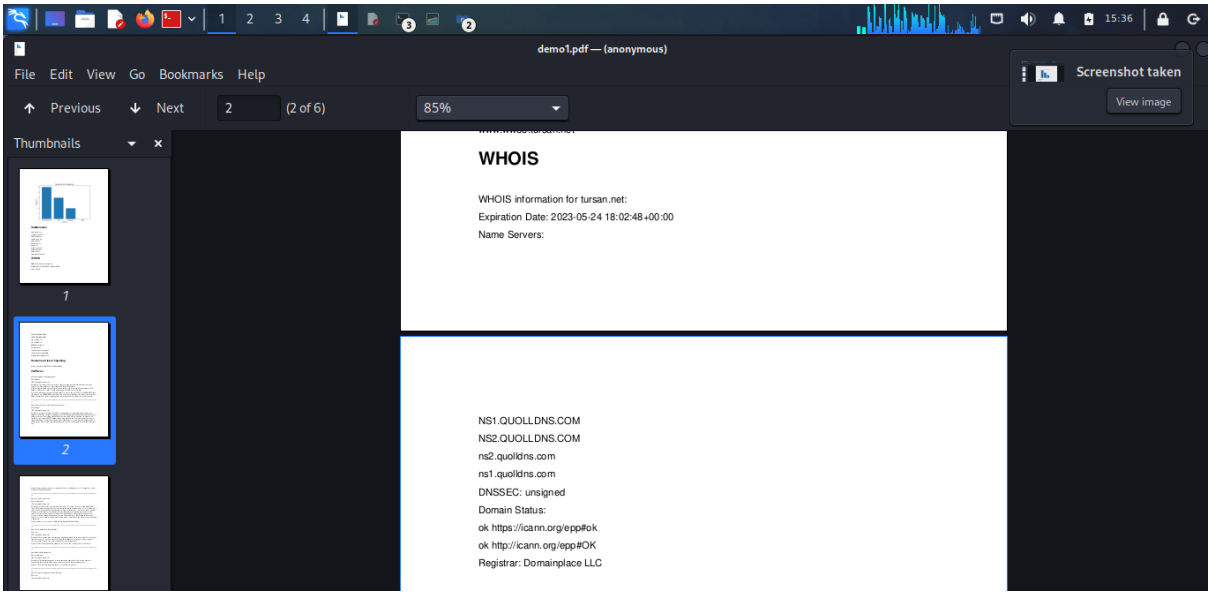


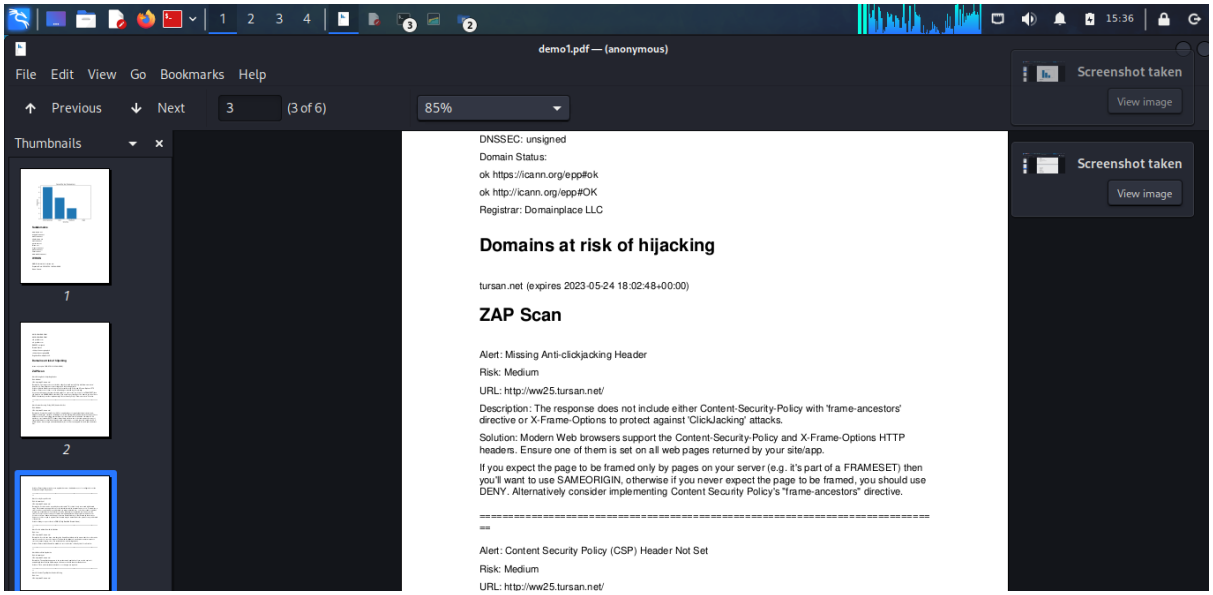
User Interface

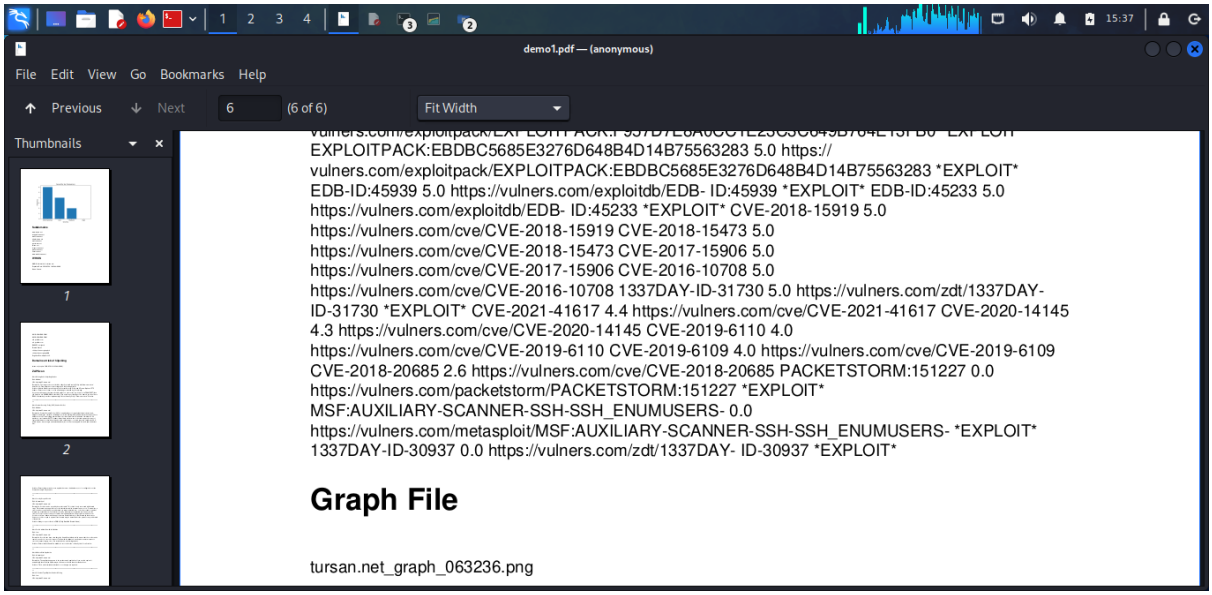
At the moment, the tool does not have a functioning GUI, it is purely CLI-based in both Windows and Linux with clear command-line prompts which do not change in either operating system. It also comes with progress bars to show users the status of the scan.

The Report






demo1.pdf — (anonymous)
File Edit View Go Bookmarks Help
Previous Next 3 (3 of 6) 85%
Thumbnails
DNSSEC: unsigned
Domain Status:
ok https://icann.org/epp#ok
ok http://icann.org/epp#OK
Registrar: Domainplace LLC
Domains at risk of hijacking
tursan.net (expires 2023-05-24 18:02:48+00:00)
ZAP Scan
Alert: Missing Anti-clickjacking Header
Risk: Medium
URL: http://ww25.tursan.net/
Description: The response does not include either Content-Security-Policy with 'frame-ancestors' directive or X-Frame-Options to protect against 'ClickJacking' attacks.
Solution: Modern Web browsers support the Content-Security-Policy and X-Frame-Options HTTP headers. Ensure one of them is set on all web pages returned by your site/app.
If you expect the page to be framed only by pages on your server (e.g. it's part of a FRAMESET) then you'll want to use SAMEORIGIN, otherwise if you never expect the page to be framed, you should use DENY. Alternatively consider implementing Content Security Policy's "frame-ancestors" directive.
Alert: Content Security Policy (CSP) Header Not Set
Risk: Medium
URL: http://ww25.tursan.net/
Screenshot taken
View image
Screenshot taken
View image


demo1.pdf — (anonymous)
File Edit View Go Bookmarks Help
Previous Next 6 (6 of 6) Fit Width
Thumbnails
EXPLOITPACK:EBDBC5685E3276D648B4D14B75563283 5.0 https://
vulners.com/exploitpack/EXPLOITPACK:EBDBC5685E3276D648B4D14B75563283 *EXPLOIT*
EDB-ID:45939 5.0 https://vulners.com/exploitdb/EDB- ID:45939 *EXPLOIT* EDB-ID:45233 5.0
https://vulners.com/exploitdb/EDB- ID:45233 *EXPLOIT* CVE-2018-15919 5.0
https://vulners.com/cve/CVE-2018-15919 CVE-2018-15473 5.0
https://vulners.com/cve/CVE-2018-15473 CVE-2017-15906 5.0
https://vulners.com/cve/CVE-2017-15906 CVE-2016-10708 5.0
https://vulners.com/cve/CVE-2016-10708 1337DAY-ID-31730 5.0 https://vulners.com/zdt/1337DAY-
ID-31730 *EXPLOIT* CVE-2021-41617 4.4 https://vulners.com/cve/CVE-2021-41617 CVE-2020-14145
4.3 https://vulners.com/cve/CVE-2020-14145 CVE-2019-6110 4.0
https://vulners.com/cve/CVE-2019-6110 CVE-2019-6109 4.0 https://vulners.com/cve/CVE-2019-6109
CVE-2018-20685 2.6 https://vulners.com/cve/CVE-2018-20685 PACKETSTORM:151227 0.0
https://vulners.com/packetstorm/PACKETSTORM:151227 *EXPLOIT*
MSF:AUXILIARY-SCANNER-SSH-SSH_ENUMUSERS- 0.0
https://vulners.com/metasploit/MSF:AUXILIARY-SCANNER-SSH-SSH_ENUMUSERS- *EXPLOIT*
1337DAY-ID-30937 0.0 https://vulners.com/zdt/1337DAY- ID-30937 *EXPLOIT*
Graph File
tursan.net_graph_063236.png

Description of Conformance to Specification and Design

The submitted Python CLI tool closely follows the original specification and design, with a few deviations and modifications that were made to enhance functionality, efficiency, and usability.

Some of the changes include the aforementioned progress bar, which was added to provide real-time feedback on the ZAP spider and active scans, thus improving user experience.

In addition to this, the graphical representation of the vulnerability severity and frequencies was added to provide users with a visual representation of their security to make it easier to prioritize the identified vulnerabilities. These changes were made to ensure that the tool delivers optimal performance and usability, ultimately meeting the needs of security professionals and developers who rely on it for web application security scanning and assessment.

I was quite unsure about a lot of things at the start, but I did more rather than less. The final product contains more features than what was described in the functional specification.

Personal Experience

Building this tool was an absolute rollercoaster. From the start, I knew I wanted to work with Open-Source Tools and create something that could contribute to individual startups and big companies at the same time. It needed to be automated and have a high level of scalability. I had become familiar with Open-Source tools like Amass during my internship at Unum, Carlow and they had suggested automating it using Python as a project for the company, however, it seemed like the idea was not big enough at first for the college.

Due to some issues with Blackboard and systems, I was left out of the initial project allocations and this, as well as the trouble of finding someone to supervise me seeing as everyone had their hands full, set me back a couple of weeks. After a few conversations about my project idea with my supervisor, I added vulnerability scanning into it, but the ‘how’ was quite difficult to figure out as there were a lot of tools out there that did well in their own respective fields, and I also had no experience with Python and automation. I thought maybe stepping into the AI world would be best but although I was eager to learn, I was already on a time crunch.

At first, I built a scanner for open ports, XSS and SQLi, but then after much trial and error with other tools, I decided it would be best to use an already existing tool. That was when Whois, Zap, and Nmap came into the mix, and I built my tool around them on my Linux VM.

From there, it was trial and error. The functional specification that I had been working on had reverted to an older version twice and there was no way to find my completed work, then my Kali Linux VM crashed, and it took a while to set up again for weeks. To add to that, it took me a while to figure out how to fully intertwine the tools, amass, nmap, zap and whois and focus on the necessities for the project. I had investigated certain Python libraries to manipulate the data from the scans performed like pandas and beautiful soup but as I progressed, I found a lot more libraries that were useful and did not need pandas for what I wanted to achieve.

Speaking about data manipulation, I underestimated how many errors one could possibly get. As I was practically ‘’learning on the job”, I had a lot of errors with tuples, lists and dictionaries in my code. Also opening ZAP in headless mode and obtaining the data from it to form a report was hard at

first but then I realized that my code and ZAP, on Linux at least could not function from the same console so I had to code to open a separate console to take the start_zap function.

Coming closer to the end, I moved my tool to test it in Windows, and asides from a few menial issues like having to use the python-whois rather than whois itself, it worked just fine, even the production of the graph within the report. I will say, the final implementation in Windows and Kali is quite professional and it can most definitely be used in a company setting which obviously is its prime purpose.

Description of Learning

Technical Achievements

Proficiency in Python Programming, including the implementation of modular code and error handling.

Mastery of cybersecurity tools, including ZAP API and Nmap for vulnerability scanning.

Understanding of web application security concepts and best practices.

Personal Achievements

Effective Project management including planning, scheduling and prioritization.

Adaptability in addressing challenges and modifying the project scope as needed.

Clear communication and documentation of project requirements. Goals and outcomes.

Review of Project

I encountered both successes and challenges during the project's development. Some of my key successes included the successful integration of ZAP and Nmap scanning functionalities, streamlining of the reporting, optimizing the tool in windows and the addition of a graph. On the other hand, the challenges included addressing the learning curve associated with cybersecurity tools and finding an effective method for visualizing vulnerability severity data.

If the project were to be approached differently, the following recommendations could be considered:

I would prioritize doing a deep study of the various cybersecurity tools before implementing them. I spoke to a few cyber security and software development professionals, and this would have helped me understand more of what I wanted and what I could do. The chosen technology stack, including Python, ZAP API, and Nmap, proved to be effective and appropriate for the project's goals. Alternative options that I could have considered include other programming languages, such as Java or JavaScript, or different scanning tools, like Burp Suite or Nessus.

I honestly consider this tool to be a success, as it achieved its goals of automating web application vulnerability scanning and port scanning processes, consolidating multiple tools, and simplifying the identification and mitigation of security issues.

Conclusion

In summary, this project aimed to develop a Python CLI tool that streamlines web application vulnerability scanning and port scanning processes. The resulting tool achieved its goals by integrating ZAP and Nmap functionalities, simplifying reporting, and providing an automated OWASP

mitigation search. Future improvements could include support for additional scanning tools, enhanced visualization options, and further customization of the scanning process. Overall, the tool represents a valuable contribution to the cybersecurity domain, providing a user-friendly solution for security professionals and developers.

Acknowledgments

I would like to say a big thank you to my parents, Uche & Tony Ibikunle, my sisters, my supervisor Paul Barry, my teachers and the Cyber security class of 2023 for helping me through this very incredible time at SETU Carlow.