CYBER THREAT DETECTION TOOL

Mini Project 2022

Done By

VISHNU SHAJI

TVE21MCA-2057

Guided By

Sreerekha V K

Asst. Professor

Dept of Computer Applications
College of Engineering
Trivandrum-695016

ABSTRACT

Cyber threat detection tool is a next generation information gathering tool that provides automated, deep and continuous security for organizations of all sizes. Security tools are expensive and time consuming but with this tool you can save time by automating the execution of various open source and commercial tools to discover vulnerabilities across your entire attack surface. Hacking is a problem that's only getting worse. But with this tool we can find hidden assets and vulnerabilities in an environment. Integrated with leading commercial and open source vulnerability scanners to scan for the latest CVEs and vulnerabilities. Thereby find out quickly how hackers can attack our business or organization before it's too late.

ACKNOWLEDGEMENT

If words are considered as symbols of approval and tokens of acknowledgement, then let words play the heralding role of expressing our gratitude.

First of all I would like to thank God almighty for bestowing with wis- dom,courage and perseverance which had helped to complete this project Cyber Threat Detection Tool. This project has been a reality as a result of the help given by a large number of personalities.

I would like to thank **Dr. Suresh Babu V**, Principal, College of Engineering Trivandrum, who helped me during the entire process of work.

I am extremely grateful to **Prof. Deepa S S**, HOD, Dept of Computer Applications,for providing me with best facilities and atmosphere for the creative work guidance and encouragement.

I express our sincere thanks to our guide **Prof.Sreerekha V K**, Department of Computer Applications, College of Engineering Trivandrum for his valuable guidance, support and advices that aided in the successful completion of our project.

I profusely thank other Asst. Professors in the department and all other staffs of CET, for their guidance and inspirations throughout my course of study.

I owe my thanks to my friends and all others who have directly or indirectly helped me in the successful completion of this project. No words can express my humble gratitude to my beloved parents and relatives who have been guiding me in all walks of my journey.
VISHNU SHAJI

Contents

List of Figures

List of Tables

Chapter 1

Introduction

On the Internet, information is widespread and business operators, alike, risk data theft. Every year, technology becomes more complicated and so do cyber attacks. The world of digital crime is expansive and it isn't exclusive to any particular Internet-accessible platform. Desktops, smartphones, and tablets may each carry a degree of digital defense but each has inherent 'weak points' to which hackers have become attuned

Cyber Threat Detection Tool is a next-generation information gathering tool that provides automated, deep, and continuous security for organizations to find different assets and vulnerabilities in a web application platform. The tool is integrated with various commercial and open source vulnerability scanners to scan for the latest CVEs and vulnerabilities. So the tool is developed for Organizations, Cyber security specialists and bug bounty hunters to find out quickly how hackers can attack your business or organization before it's too late.

Chapter 2

Literature Review

In recent years, the internet has become an integral element of people's everyday lifestyles all across the world. Online criminality, on the other hand, has risen in tandem with the growth of internet activity. Cyber threats isn't just a problem that affects individuals but it also applies to an organization or a government. Cyber attacks are raising concerns about privacy, security and financial compensation. Cyber security is a set of technologies, processes and practices aimed at preventing attacks, damage and illegal access to networks, computer programmes and data. so proper detection of threats and vulnerabilities in a cyberspace is important to safeguards the data and integrity of computing assets that are part of or connected to an organisation's network , with the goal of defending such assets from all threat actors throughout the life cycle of a cyber attack.

Threat analysis includes activities which helps to identify, analyse and prioritize potential security and privacy threats to a software system and the information it handles. Threat analysis is particularly important since many security vulnerabilities are caused due to architectural design flaws. A failure to consider security early-on can be cause for so-called Architectural Technical Debt(ATD). Furthermore, fixing such vulnerabilities after implementation is very costly and requires workarounds which sometimes increase the attack surface.

2.1 Existing System

Now, the security analysts and bug bounty hunters use different tools for finding various threats and vulnerabilities. Because there is no special package which finds different threats by using a single tool. So manually selecting each tool for finding vulnerabilities is time consuming process.

2.1.1 Drawbacks of Existing System

- Using individual tools takes a lot of time
- It is susceptible by human errors

- Inability to analysis result as a whole
- It is costly to maintain manual testers

2.2 Proposed System

The proposed system overcomes these drawbacks by integrating different open source and commercial threat detection and analysis tools into an automated command line tool. The tool automatically find hidden assets and vulnerabilities in a web application and stores the results for later analysis for the user.

2.2.1 Advantages of Proposed System

- Automated process of finding threats
- Report will provide specific advice
- Less expensive and time efficient
- Eliminate stress and increase productivity
- Higher chance of success

Chapter 3

Requirement Analysis

3.1 Purpose

The objective of this tool is to identify different vulnerabilities in a web application by using an automated command line tool.

3.2 Overall Description

Cyber threat detection tool is a next generation information gathering tool that provide automated, deep and continuous security for organizations of all sizes. The tool is capable of finding hidden assets and vulnerabilities in your environment. Integrate with the leading commercial and open source vulnerability scanners to scan for the latest CVEs and vulnerabilities. find out quickly how hackers can attack your business or organisation before it's too late.

3.2.1 Hardware Requirements

- Processor : Intel Core i5
- Storage : 512 GB hard Disk space and 256 GB SSD
- Memory : 8 GB RAM

3.2.2 Software Requirements

- Operating System : Linux (Debian)
- Frameworks : Metasploit

3.3 Functional requirements

The systems intended behaviour is defined by functional requirements. Tasks or functions that the specified system is supposed to do can be used to describe

this behaviour. The parts of the proposed system are as follows. They are given below:

- Metasploit-Framework: The Metasploit framework is a very powerful tool which can be used by ethical hackers to probe systematic vulnerabilities on networks and servers. Because it's an open-source framework, it can be easily customized and used with most operating systems.
- Scanning tools: A vulnerability scanner is a computer program designed to assess computers, networks or applications for known weaknesses. These scanners are used to discover the weaknesses of a given system. There different types of scanners Vulnerability scanners, Web Application scanners, Network Scanning, SSL/TLS scanning. The scanners includes Sc0pe, Nuclei, openVAS, Nessus, Nmap, SSH-Audit, OWASP ZAP, WP-Scan, Arachni, Gobuster, Nikto, Webtech, Shocker, Smuggler, SSLScan etc.
- Reconnaissance Tools: Active reconnaissance is a method of collecting information about the target environment by directly interacting with the target or by sending traffic to the target. The information is further used to exploit the target. This method may be identified by Intrusion Detection System (IDS) used by the target organization. The tools includes massDNS,sublist3r,AMass,Subfinder,Censys,AltDNS etc.
- Open Source Intelligence (OSINT) tools: Open Source Intelligence software, abbreviated as OSINT software, are tools that allow the collection of information that is publicly available or open-source. The goal of OSINT software is mainly to learn more about someone or a business. The tools include Hunter, SpoofCheck, theHarvester, whois, inURLBR, WayBackMachine, Gitgrabber etc.

3.4 Non Functional requirements

Non functional requirements are the quality requirements that the system must meet in order to fulfill the project contract. non behavioural requirements are another name for them.

3.4.1 Performance Requirements

- Usability: The system must be simple to use and comprehend. This tool will be simple to use and uses will be able to access all of the features without difficulty.
- Privacy: The tool must ensure that no personal profile information is shared with other users. only relevant and required information should be accessible.
- Performance: The tool should respond to users in a considerable time window. it should not be too slow or too fast for the users.

- Speed: the system must be capable of offering speed.
- Reliability: The program must be dependable in order to do its responsibilities.

3.4.2 Quality Requirements

- Scalability: The tool will meet all of the functional requirements.
- Maintainability: The system should be maintainable. It should keep backups to atone for system failures and should log it's activities periodically.
- Availability: This system is easily available as the core equipment in building the application is easily obtained.
- Consistency: The data should be consistent and precise the system would need a stable internet connection to store and retrieve data from the database.
- High-Functionality: This system is highly functional in all environment since they are highly adaptable.

Chapter 4

Design And Implementation

The purpose of design phase is to plan a solution of the problem specified by the analysis phase. This phase is the first step in moving from the problem domain to solution domain.

4.1 Overall Design

This tool enables user to automatically check for vulnerabilities in a web applications. This tool combines multiple scanning tools such as Sc0pe, Nuclei, openVAS, Nessus, Nmap, SSHAaudit, Gobuster, Nikto, Webtech, Schocker, Smuggler, SSLscan etc.., Reconnaissance tools such as massdns, sublist3r, AMass, subfinder, Censys,AltDNS etc.., open source intelligence tools such as Hunter,SpoofCheck, theHarvester, whois, inURLBR, WayBAckMachine, Gitgrabber etc.. and integrated with metasploit framework.

The user can use different options provided by the tool for scanning and storing the results. The user can later analyse these results for identifying vulnerabilities and take necessary security measures to ensure security of their web application.

4.2 User Interface Design

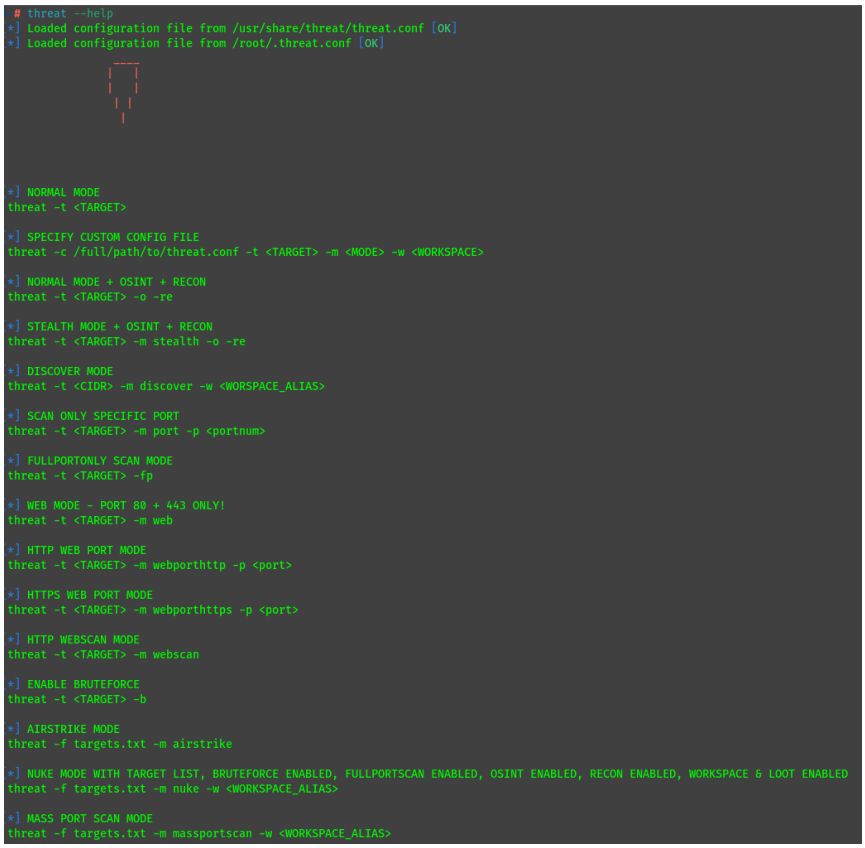

Figure 4.1: Options page

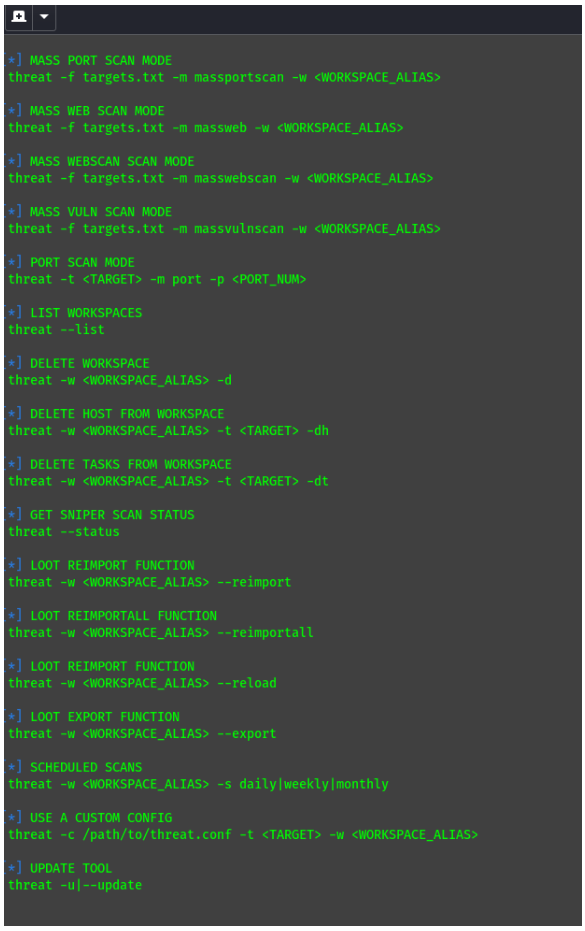

Figure 4.2: Options Terminal

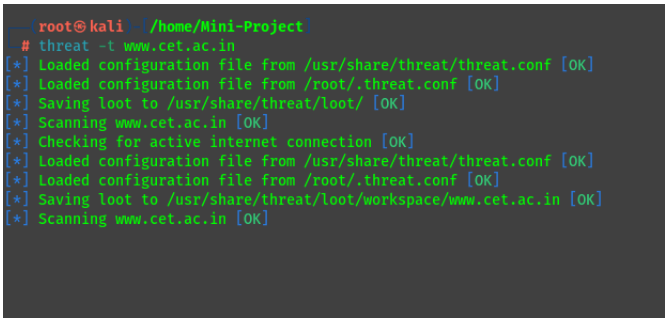

Figure 4.3: Tool start Terminal

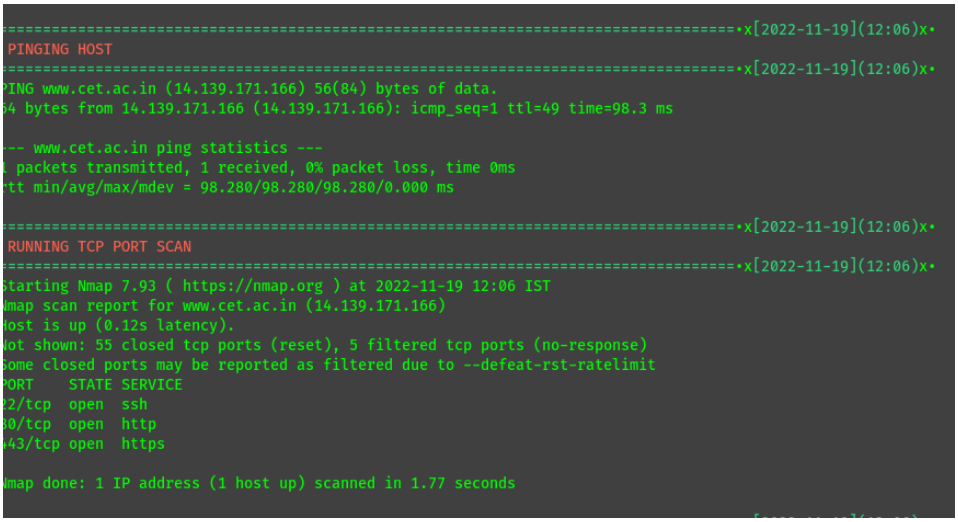

Figure 4.4: Reconnaissance

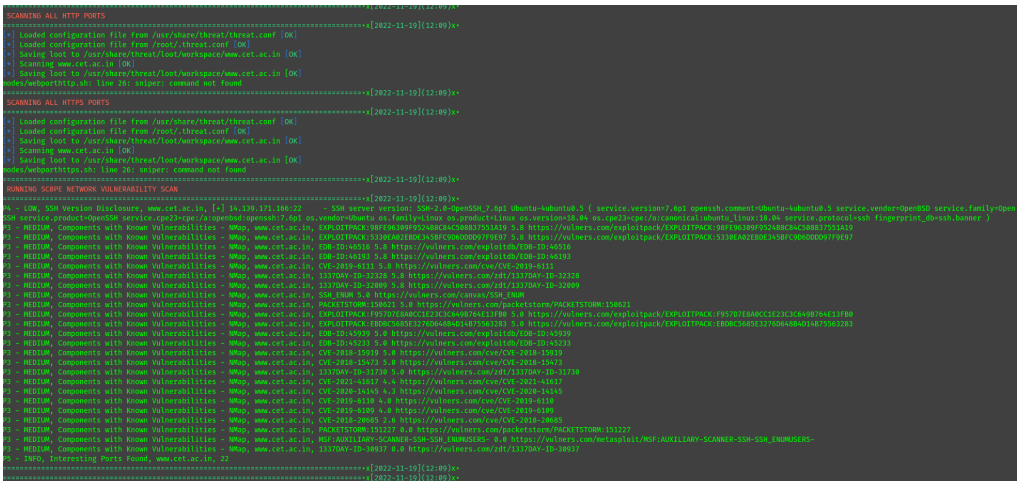

Figure 4.5: Scanning

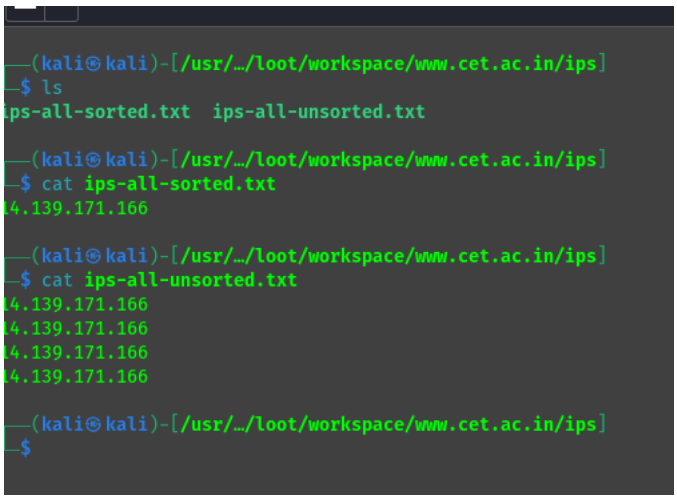

Figure 4.6: IP Address Report

```
┌──(root㉿kali)-[/usr/…/loot/workspace/www.cet.ac.in/output]
└─# cat msf-www.cet.ac.in-port22-ssh_enumusers.txt
USER_FILE => /usr/share/brutex/wordlists/simple-users.txt
RHOSTS => www.cet.ac.in
RHOST => www.cet.ac.in
[*] 14.139.171.166:22 - SSH - Using malformed packet technique
[*] 14.139.171.166:22 - SSH - Starting scan
[+] 14.139.171.166:22 - SSH - User 'admin' found
[+] 14.139.171.166:22 - SSH - User 'administrator' found
[+] 14.139.171.166:22 - SSH - User 'anonymous' found
[+] 14.139.171.166:22 - SSH - User 'backup' found
[+] 14.139.171.166:22 - SSH - User 'bee' found
[+] 14.139.171.166:22 - SSH - User 'ftp' found
[+] 14.139.171.166:22 - SSH - User 'guest' found
[+] 14.139.171.166:22 - SSH - User 'GUEST' found
[+] 14.139.171.166:22 - SSH - User 'info' found
[+] 14.139.171.166:22 - SSH - User 'mail' found
[+] 14.139.171.166:22 - SSH - User 'mailadmin' found
[+] 14.139.171.166:22 - SSH - User 'msfadmin' found
[+] 14.139.171.166:22 - SSH - User 'mysql' found
[+] 14.139.171.166:22 - SSH - User 'nobody' found
[+] 14.139.171.166:22 - SSH - User 'oracle' found
[+] 14.139.171.166:22 - SSH - User 'owaspbwa' found
[+] 14.139.171.166:22 - SSH - User 'postfix' found
[+] 14.139.171.166:22 - SSH - User 'postgres' found
[+] 14.139.171.166:22 - SSH - User 'private' found
[+] 14.139.171.166:22 - SSH - User 'proftpd' found
[+] 14.139.171.166:22 - SSH - User 'public' found
[+] 14.139.171.166:22 - SSH - User 'pi' found
[+] 14.139.171.166:22 - SSH - User 'kali' found
[+] 14.139.171.166:22 - SSH - User 'root' found
```

Figure 4.7: Enumeration report

```
┌──(root㉿kali)-[/usr/…/loot/workspace/www.cet.ac.in/output]
└─# cat nmap-www.cet.ac.in-port22.txt
Starting Nmap 7.93 ( https://nmap.org ) at 2022-11-19 13:46 IST
NSE: Loaded 51 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 13:46
Completed NSE at 13:46, 0.00s elapsed
Initiating NSE at 13:46
Completed NSE at 13:46, 0.00s elapsed
Initiating Parallel DNS resolution of 1 host. at 13:46
Completed Parallel DNS resolution of 1 host. at 13:46, 0.02s elapsed
Initiating SYN Stealth Scan at 13:46
Scanning www.cet.ac.in (14.139.171.166) [1 port]
Discovered open port 22/tcp on 14.139.171.166
Completed SYN Stealth Scan at 13:46, 0.14s elapsed (1 total ports)
Initiating Service scan at 13:46
Scanning 1 service on www.cet.ac.in (14.139.171.166)
Completed Service scan at 13:46, 0.18s elapsed (1 service on 1 host)
Initiating OS detection (try #1) against www.cet.ac.in (14.139.171.166)
Retrying OS detection (try #2) against www.cet.ac.in (14.139.171.166)
Initiating Traceroute at 13:46
Completed Traceroute at 13:46, 3.28s elapsed
Initiating Parallel DNS resolution of 6 hosts. at 13:46
Completed Parallel DNS resolution of 6 hosts. at 13:46, 0.20s elapsed
NSE: Script scanning 14.139.171.166.
Initiating NSE at 13:46
NSE: [ssh-run 14.139.171.166:22] Failed to specify credentials and command to run.
NSE: [ssh-brute 14.139.171.166:22] Trying username/password pair: root:root
NSE: [ssh-brute 14.139.171.166:22] Trying username/password pair: admin:admin
NSE: [ssh-brute 14.139.171.166:22] Trying username/password pair: administrator:administrator
NSE: [ssh-brute 14.139.171.166:22] Trying username/password pair: webadmin:webadmin
NSE: [ssh-brute 14.139.171.166:22] Trying username/password pair: sysadmin:sysadmin
NSE: [ssh-brute 14.139.171.166:22] Trying username/password pair: netadmin:netadmin
NSE: [ssh-brute 14.139.171.166:22] Trying username/password pair: guest:guest
NSE: [ssh-brute 14.139.171.166:22] Trying username/password pair: user:user
NSE: [ssh-brute 14.139.171.166:22] Trying username/password pair: web:web
NSE: [ssh-brute 14.139.171.166:22] Trying username/password pair: test:test
NSE: [ssh-brute 14.139.171.166:22] Trying username/password pair: root:
NSE: [ssh-brute 14.139.171.166:22] Trying username/password pair: admin:
NSE: [ssh-brute 14.139.171.166:22] Trying username/password pair: administrator:
NSE: [ssh-brute 14.139.171.166:22] Trying username/password pair: webadmin:
NSE: [ssh-brute 14.139.171.166:22] Trying username/password pair: sysadmin:
NSE: [ssh-brute 14.139.171.166:22] Trying username/password pair: netadmin:
NSE: [ssh-brute 14.139.171.166:22] Trying username/password pair: guest:
NSE: [ssh-brute 14.139.171.166:22] Trying username/password pair: user:
NSE: [ssh-brute 14.139.171.166:22] Trying username/password pair: web:
NSE: [ssh-brute 14.139.171.166:22] Trying username/password pair: test:
NSE: [ssh-brute 14.139.171.166:22] Trying username/password pair: root:123456
```

Figure 4.8: Nmap and Bruteforcing report

```
┌──(root㉿kali)-[/usr/…/loot/workspace/www.cet.ac.in/scans]
└─# cat tasks.txt
www.cet.ac.in normal 2022-11-17 23:30
www.cet.ac.in webporthttp 2022-11-17 23:33
www.cet.ac.in webporthttps 2022-11-17 23:33
www.cet.ac.in normal 2022-11-18 12:51
www.cet.ac.in webporthttp 2022-11-18 12:54
www.cet.ac.in webporthttps 2022-11-18 12:54
www.cet.ac.in normal 2022-11-18 13:41
www.cet.ac.in webporthttp 2022-11-18 13:43
www.cet.ac.in webporthttps 2022-11-18 13:44
www.cet.ac.in normal 2022-11-19 12:06
www.cet.ac.in webporthttp 2022-11-19 12:09
www.cet.ac.in webporthttps 2022-11-19 12:09
www.cet.ac.in normal 2022-11-19 13:46
```

Figure 4.9: Performed task list

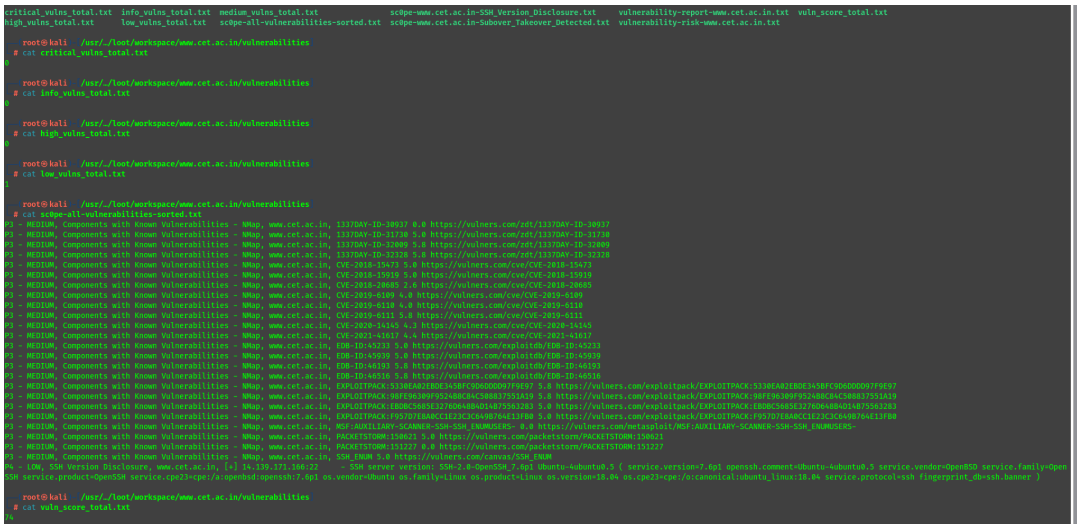

Figure 4.10: Final Threat Analysis Report

Chapter 5

Coding

5.1 Algorithm

Step 1 : Read tool option from the user.

Step 2: Read target domain name or IP address from user.

Step 3: Perform normal scan.

Step 4: Perform reconnaissance and open source intelligence scanning.

Step 5: Scan for specific port.

Step 6: Perform full port only scan mode.

Step 7: Perform web mode scanning.

Step 8: perform HTTP web scan mode.

Step 9: Perform bruteforce.

Step 10: Perform airstrike mode.

Step 11: Perform scheduled scans.

Step 12: Use custom configuration scan.

Step 3: Save the reports of all scans.

Step 3: save the final report of vulnerability analysis.

Chapter 6

Testing and Implementation

6.1 Testing methods

Once an application is developed the major activity is to test whether the actual results match with experimental results. This is called testing. It's used to make sure that the developed system is defect free. The main aim of testing is to find the errors and missing operations by executing the program. It also ensure that all of the objectives of the project are met by the developer. the objective of testing is not only to evaluate the bugs in the created software but also finding the ways to improve the efficiently, usability and accuracy of it. It aims to measure the functionality specification and performance of a software program. tests are performed on the creator software and their results are compared with the expected documentation. When there are too much errors occurred, debugging is performed. and the result after debugging is tested again to make sure that the tool is error free. the major testing process applied to this project are unit testing and integration testing. In unit testing our aim is to test all individual units of the software. It makes sure that all of the units of the software works as intended. In integration testing the combined individual units are tested to check whether it meet the intended function or not. It helps us to find out the faults that may arise when the units are combined. In system testing the enters software is tested to make sure that it satisfies all of the requirements. The table shown below describes that is think process occurred during the development of this project "Cyber Threat Detection Tool: an automated command line tool for threat detection and analysis.

6.1.1 Unit Testing

Sl No	Procedures	Expected result	Actual Result	Result
1	Test with OSINT tools	Test results are displayed	Same as expected	pass
2	Metasploit Framework test	Test results are displayed	Same as expected	pass
3	Test with Scanning Tools	Test results are displayed	Same as expected	pass
4	Data Analysis Testing	Details page is shown	Same as expected	pass
5	Fingerprinting Testing	Details page is shown	Same as expected	pass

Table 6.1: Testing

6.1.2 Integration Testing

Sl No	Procedures	Expected result	Actual Result	Result
1	View Scanning results	Results are displayed in terminal	Same as expected	pass
2	View data analysis results	Results displayed in the terminal	Same as expected	pass
3	View fingerprinting results	Results displayed in the terminal	Same as expected	pass

Table 6.2: Integration testing

6.1.3 System Testing

Sl No	Procedures	Expected result	Actual Result	Result
1	Report files generated	Results are stored and displayed	Same as expected	pass

Table 6.3: System testing

6.2 Advantages and Limitations

The proposed method has more benefit than the current system. the proposed system save a huge amount of time. like every other system this system also have its on disadvantages. But they are negligible while comparing with the advantages and they can be overcame in future.

6.2.1 Advantages

- Reduce risk of damage.

- Reduce downtime thereby increase productivity.
- Automated process of detecting vulnerabilities.
- The tool provides vulnerability analysis reports.

6.2.2 Disadvantages

- The tool is only work with linux(debian) platforms

6.3 Future Extensions if possible

Support for these tool can be extended to various other platforms like Windows.
Realtime analysis of security structure of a web application through a cloud platform for better user interface experience.

Chapter 7

Conclusion

Thus it is better understood how the proposed system is better than the existing system. Different threats and vulnerabilities of a web application can be determined by using this tool. The tool aims at satisfying the requirement of an automated threat detection tool for needy peoples like security analysts and bug bounty hunters. The tool provides several options for scanning and finding treats based on the need of user. An effort focused on finding threats and vulnerabilities before it leads to a data breach or any type of attacks. This work is the initial step to protect every web applications from cyber attacks.

Cyber threats will continue to plague businesses. I will do my best to secure the cyberspace from attackers by identifying threats and removing the threats before any harm happened. The portability of tool may help all types of users other than linux for threat detection. I will try to extend the tool functionality to additional platforms like Windows and iOS.

.

Chapter 8

Reference

- Hacking: The Art of Exploitation, 2nd Edition [Jon Erickson]
- Constructing an Ethical Hacking Knowledge Base for Threat Awareness and Prevention [Sunita Vikrant Dhavale]
- https://codered.eccouncil.org/Ethical-hacking-Essentials [Online course by EC Council]
- CCNP Security Cisco Secure Firewall and Intrusion Prevention System Official Cert Guide [Nazmul Rajib]