

INFOSINT | Informative Open source Intelligence tool for Penetration Testers

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering Degree in Computer Science and
Engineering

By

R KAUSHIK BABU (Reg.NO: 40110579)

KEVIN SAM MATTHEW. G (Reg.NO: 40110587)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING
SATHYABAMA**

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
CATEGORY-1 UNIVERSITY BY UGC**

**Accredited with Grade “A++” by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600119**

APRIL - 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Category - I University by UGC

Accredited "A++" by NAAC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the Bonafide work of R KAUSHIK BABU (Reg.No - 40110579) and KEVIN SAM MATTHEW.G (Reg.No - 40110587) who carried out the project entitled "INFOSINT | Informative Open-source Intelligence tool for Penetration Testers" under our supervision from NOV 2023 to APRIL 2024.


12/4/24
Internal Guide

Dr. M. Selvi, M.Tech., Ph.D.,


Head of the Department
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Jeppiaar Nagar, Rajiv Gandhi Salai,
Chennai - 600 119.

Submitted for Viva voce Examination held on 13 - 04 - 2024


Internal Examiner


External Examiner

DECLARATION

I, Kaushik Babu R (Reg.NO: 40110579), hereby declare that the Project Report entitled "INFOSINT | Informative Open-source Intelligence tool for Penetration Testers" done by me under the guidance of Dr. M. Selvi, M.Tech., Ph.D., is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE: 13/04/2024

PLACE: Chennai



SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

We are pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

We convey my thanks to **Dr. T. Sasikala M.E., Ph.D., Dean**, School of Computing, **Dr. L. Lakshmanan M.E., Ph.D.**, Heads of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

We would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. M. Selvi, M.Tech., Ph.D.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

We wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Information Gathering and Enumeration / Scanning play a vital role in Cybersecurity. It is basically, the first step or the beginning stage of Ethical Hacking, where the penetration testers or hackers (both black hat and white hat) try to gather all the information about the target, in order to use it for Hacking. But the enumeration part of Cybersecurity is a little bit time-consuming because of using multiple tools, comprehensive strategies, Custom Techniques and etc. INFOSINT or General Enumeration is a CLI (Command Line Interface) based Cybersecurity Tool, which is used in Linux (Debian) Operating system. It is Developed using Python3 & Bash. INFOSINT helps security professionals to enumerate Information in various technologies like Networking, Web Applications, & Open Source Intelligence (OSINT) . INFOSINT uses the utilities of Offensive Security tools like Nmap, Nmap Scripting Engine (NSE), Gobuster, Assertfinder, Whois-lookup, OSINT Framework ,and API. It helps Security professionals to navigate quickly and uses the full potential of the listed tools with a Single Comprehensive Script. Network Enumeration a computing activity in which usernames and info on groups, shares, and services of networked computers are retrieved. Web Enumeration is a bit tougher than host discovery. It requires the attacker to use a particular methodology to uncover everything about a target web service. OSINT framework focused on gathering information from free tools or resources. The intention is to help people find free OSINT resources.

TABLE OF CONTENTS

Chapter No	TITLE	Page No.
	ABSTRACT	v
	LIST OF TABLES	vii
	LIST OF FIGURES	viii
1	INTRODUCTION	1
2	LITERATURE SURVEY	4
	2.1 Inferences from literature survey	4
	2.2 Open problems in Existing System	7
3	REQUIREMENTS ANALYSIS	8
	3.1 Feasibility Studies/Risk Analysis of the Project	8
	3.2 Hardware requirements	9
	3.3 Software Requirements Specification Document	10
4	DESCRIPTION OF PROPOSED SYSTEM	11
	4.1 Selected Methodology or process model	11
	4.2 Architecture / Overall Design of Proposed System	13
5	IMPLEMENTATION DETAILS	17
	5.1 Development setup	17
	5.2 Dependency requirement	17
	5.3 Testing the module	18
6	RESULT AND DISCUSSION	20
7	CONCLUSION	21
	7.1 Future work	21
	REFERENCES	22
	APPENDIX	23
	A Source code	23
	B Screenshot	44
	C Research paper	53

LIST OF TABLES

TABLE NO	TABLE NAME	PAGE NO
5.3.1	Testing model	19

LIST OF FIGURES

FIG.NO	FIGURE NAME	PAGE NO
1.1	Penetration testing lifecycle	3
4.2.1.1	Uml class diagram	14
4.2.1.2	Data flowchart diagram	15
4.2.2.1	Uml use case diagram	16
6.1	Result interface	20
B.1	Info about tool	44
B.2	Network enum menu	44
B.3	Nmap comprehensive	45
B.4	Nmap comprehensive	46
B.5	Nmap service ver detection	46
B.6	Nmap os	47
B.7	Web enum menu	47
B.8	Web domain info	48
B.9	Web directory enum	48
B.10	Osint menu	49
B.11	Number info	49
B.12	Trace global ip	49
B.13	Trace global ip	50
B.14	Cryptography menu	50
B.15	John the ripper	51
B.16	John the ripper	51
B.17	hashcat	52

CHAPTER 1

INTRODUCTION

1.1 CYBERSECURITY

Cybersecurity is the practice of protecting systems, networks, and programs from digital attacks. These cyberattacks are usually aimed at accessing, changing, or destroying sensitive information; extorting money from users; or interrupting normal business processes. Implementing effective cybersecurity measures is particularly challenging today because there are more devices than people, and attackers are becoming more innovative.

1.1.1 PENETRATION TESTING

A penetration tester, much like an experienced hacker, performs a deep investigation of the remote system's security flaws. This activity requires methodology and skills. A Penetration Test is both a complex and a very delicate process. And there are six different stages in the Lifecycle of Penetration Testing,

- Engagement
- Information Gathering
- Footprinting And Scanning
- Vulnerability Assessment
- Exploitation
- Reporting

1.1.2 ENGAGEMENT

at the Engagement stage, a professional pentester defines the amount of time for the penetration testing of a Network, Web Application, or the whole organization. Time varies according to,

- Type of engagement (Black Box, Grey Box, etc)
- the Complexity of the applications and services in scope

- The number of targets (IP addresses, Domains, etc)

1.1.3 INFORMATION GATHERING

Information Gathering is the first and one of the most fundamental stages of a successful penetration test. Information Gathering means gathering different kinds of information about the target. It is basically, the first step or the beginning stage of Ethical Hacking, where the penetration testers or hackers (both black hat and white hat) try to gather all the information about the target, in order to use it for Hacking.

1.1.4 FOOTPRINTING AND SCANNING

Refers to the process of collecting as much information as possible about the target system to find ways to penetrate the system. An Ethical hacker has to spend the majority of his time profiling an organization, gathering information about the host, network, and people related to the organization. Information such as IP address, Whois records, DNS information, an operating system used, employee email id, Phone numbers, etc are collected.

1.1.5 VULNERABILITY ASSESSMENT

A vulnerability assessment is a systematic review of security weaknesses in a target. It evaluates if the target is susceptible to any known vulnerabilities, assigns severity levels to those vulnerabilities, and recommends remediation or mitigation. We can carry out a vulnerability assessment in two ways,

- Manually by using data collected in the previous phases
- by utilizing Automated Tools

1.1.6 EXPLOITATION

At this stage, it is time to verify if the vulnerabilities really exist. The exploitation phase takes care of exploiting all the vulnerabilities found during the previous step. During the exploitation phase, a penetration tester checks and validates a vulnerability and also widens and increases the pentester's privileges on the target system and network. A successful exploit of a machine

helps to investigate the target network further, to discover new targets, and to repeat the process from the information-gathering phase. A penetration test is indeed a Cyclic Process.

1.1.7 REPORTING

Lastly, the final penetration test report is as important as the whole testing phase, as it is a way to officially deliver and communicate the results of your tests with,

- Executives
- IT Staff
- Development team

The report shows and explains the result of your tests and is the actual deliverable of your professional engagement.



Fig: 1.1 Penetration Testing Life-cycle

CHAPTER 2

LITERATURE SURVEY

2.1 INFERENCES FROM LITERATURE SURVEY

In [1] 2021, G Jayasuryapal , P Meher Pranay has proposed A Survey on Network Penetration Testing which focussed on all the System along with information gathering to the post exploitation. Penetration testing operation follow specific rules and unity that both the gathering (client and pen testing team). By this testing, the organization's defects will be finding like open servers and open ports etc.

In [2] 2019, Pengfei Shi , Futong Qin has proposed The Penetration Testing Framework for Large-Scale Network Based on Network Fingerprint. The fundamentals and methodology of conventional penetration testing technology for big networks are examined, along with some of its practical limitations. A penetration testing framework for large-scale networks based on network fingerprint is realised by combining network fingerprinting with online search engine technologies.

In [3] 2017, Rodney R Rohrmann , Vincent J Ercolani has proposed Large scale port scanning through tor using parallel Nmap scans to scan large portions of the IPv4 range. Though effective in assuming data-information during scanning, it is not expandable to the point of scanning the entire IPv4 Address space on multiple ports, as scans take appreciably longer to execute through Tor. The benefit of a scanning method that shields researchers from targets who might potentially retaliate after being scanned is that it enables them to source their own scans.

In [4] 2015, Prerna Arote , Karam Veer Arya has proposed Detection and Prevention against ARP Poisoning Attack using Modified ICMP and Voting . An IP network protocol called Internet Control Message Protocol (ICMP) is helpful

for administering and controlling networks. A crucial component of IP implementations is ICMP. ARP was created for a more dependable world. The Data Link Layer is in charge of framing packets as they are processed and passed down the stack (DLL).

In [5] 2016, Enos LETSOALO, Sunday OJO has proposed Survey of Media Access Control address spoofing attacks detection and prevention techniques in Wireless Networks. The Media Access Control addresses of the access point or other users can be extracted from packets intercepted by an attacker using packet sniffer software. In wireless networks, a client is connected to the access point using its MAC address. To disconnect authorised users from the network and seize control of any already established TCP session, an attacker can fake the Media Access Control IP of the realaccess point.

In [6] 2017, Lerato Ramahlapane Moila , Mthulisi Velempini has proposed An Evaluation of the Effectiveness of Cognitive Radio Ad hoc Networks Routing Protocols. Real-time data quality of service needs is not guaranteed by the current protocols. Some of the issues preventing the development of routing protocols that are quality of service aware are caused by the dynamic environment of cognitive networks, which is also responsible for the mobility of nodes, the movement of the spectrum, and the nondeterministic availability of the spectrum.

In [7] 2017, Sathish A.P. Kumar , Brian Xu has proposed Vulnerability Assessment for Security in Aviation Cyber-Physical Systems. We have tested for vulnerabilities in data loaders and other aircraft systems in order to meet the standards for wireless network security set out by the aviation industry. Our efforts include increasing aircraft security and safety as well as identifying cyber vulnerabilities in these aviation systems utilising vulnerability assessment and penetration testing tools like Metasploit Pro and BackTrack.

In [8] 2020, Mehr u Nisa , Kashif Kifayat has proposed Detection of Slow Port Scanning Attacks. In actuality, a scanning assault is a two-part process in which scanning is the first phase where the vulnerability of communication routes is

discovered. The second step is the discovery of the targets, followed by the attack. In order to identify the system that can be abused, available open ports are therefore sought across the network during port scanning.

In [9] 2022, Sabah M. Morsy And Dalia Nashat has proposed D-ARP: An Efficient Scheme to Detect and Prevent ARP Spoofing. An ARP spoof-based MITM attack takes use of a flaw in the ARP protocol by associating the attacker's MAC address with the IP address of the target legitimate host. ARP spoof based- MITM attacks have a variety of defence strategies, although these strategies are incomplete or have a limited effectiveness. since they alter the basic ARP protocol, performance overhead.

In [10] 2020, Ron Andrews , Dalton A. Hahn has proposed Measuring the Prevalence of the Password Authentication Vulnerability in SSH. We suggest a novel technique for probing an SSH service to determine whether password authentication is permitted as part of our review, without being invasive or damaging to the host. We also show that some of these tools and services can be enhanced in order to assess the prevalence of password authentication in SSH particularly.

In [11] 2017, Debdeep Dey , Archisman Dinda has proposed Warezmaster and Warezclient : An Implementation of FTP Base R2L Attacks. The fact that all commands and data are transmitted using clear text indicates that the File Transfer Protocol (FTP) was not created with strict security measures in mind. Subsequent updates to the protocol have added security features, like upgrading the connection to Transport Layer Security (TLS) in FTPS.

In [12] 2016, Thierry Mbah Mbelli , Barry Dwolatzky has proposed Cyber Security, a Threat to Cyber Banking in South Africa. For South African businesses in particular as well as the nation as a whole, the rise of cybercrime is turning into a severe economic problem. The first section of this essay examines the numerous online hazards related to cyber banking. In addition, it suggests a boundary- and application-based approach to cyber-banking security.

2.2 OPEN PROBLEMS IN EXISTING SYSTEM

There are some existing systems in the Information gathering and Enumeration Part of cybersecurity, like Search-that-hash and sooty which are only available in a particular enumeration method.

- **Limited Reporting:** While tools can generate reports, the quality and comprehensiveness of those reports can vary. Human analysis and interpretation are often necessary to provide meaningful results.
- **Scanning Noise:** Running penetration tests can generate network traffic and logs that could alert security systems or network administrators, potentially causing disruptions.
- **Time-Consuming:** Penetration testing can be a time-consuming process, especially if done manually, which may not align with tight project schedules.
- **Skill Required:** Proper use of these tools often requires a high level of expertise. Inexperienced users may misconfigure the tools or misinterpret the results.
- **Cost:** Some commercial penetration testing tools can be expensive, making them less accessible for small organizations with limited budgets.
- **Missing Zero-Day Vulnerabilities:** Penetration testing tools often rely on known vulnerabilities and attack patterns. They may not be effective at identifying zero-day vulnerabilities for which no known defense or signature exists.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDIES/RISK ANALYSIS OF THE PROJECT

A feasibility study is an analysis of how successfully a project can be completed, accounting for that affect it such as economic, technological, legal, and scheduling factors. A feasibility study tests the visibility of an idea, project, or even a new business. The goal of a Feasibility Study is to emphasize potential problems that could occur if a project is pursued and determine if, after all, significant factors are considered, how it will operate, potential obstacles, competition, and the funding needed to get the business up and running. Steps involved in the feasible study

- Analyse the problem: This initial step involves identifying and thoroughly understanding the problem or opportunity that the project aims to address. It's crucial to define the objectives clearly and understand the scope of the proposed endeavor.
- Form a project team and appoint a project leader: Assembling a competent team is essential for conducting a comprehensive feasibility study. Each team member brings valuable expertise and perspective to the table.
- Do your research: Extensive research is conducted to gather relevant data and information pertaining to various aspects of the project. This includes market research, technical assessments, legal considerations, and financial analysis. The goal is to gather comprehensive data to inform decision-making.
- Make a plan: Based on the insights gathered from research, a detailed plan is formulated outlining the steps required to assess the feasibility of the project. This plan includes timelines, resource allocation, methodologies for analysis, and milestones for progress tracking.
- Select the best-proposed system: If multiple proposed solutions or approaches are being considered, this step involves evaluating and

selecting the most promising option based on predetermined criteria. Factors such as cost-effectiveness, technical feasibility, and alignment with project objectives are taken into account.

- Check your data: Before proceeding further, it's essential to validate the accuracy and reliability of the data collected during the research phase. Any inconsistencies or gaps in the data must be addressed to ensure the integrity of the feasibility study.
- Prepare and report the final project directive to management: Finally, the findings of the feasibility study are compiled into a comprehensive report that outlines the conclusions and recommendations.

3.2 HARDWARE SPECIFICATIONS

The Hardware requirements may serve as the basis for a contract for the implementation of the systems and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design.

- | | | |
|---------------|---|--|
| • Processor | : | Intel I5 / AMD Ryzen 5 or above |
| • RAM | : | 8 Gb or above |
| • Hard disk | : | 100 Gb |
| • GPU | : | 2 Gb or above |
| • System type | : | 64-bit operating system, x64-based processor |
| • Keyboard | : | Normal or Multimedia |
| • Mouse | : | Compatible mouse |

3.3 SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT

A Software Requirements Specification is a document or set of documentation that describes the features and behavior of a system or software application. It is useful in estimating the cost, planning team activities, performing tasks, and tracking the team's progress throughout the development activity.

- Host Operating System : Windows 10 or above, Debian 9+ or above
- Virtual Operation System : Kali Linux (Debian) 2020 or above
- Programming Language : Python3 & Bash Scripting
- Terminal : Bash
- Applications : PyCharm
- Hypervisor : VMware or VirtualBox
- Vulnerable Virtual Machine: Skynet from TryHackMe / Metasploitable
- Security Framework & Tools: Nmap, NSE, Whois-lookup, Gobuster, Assertfinder, Hash-Identifier, John the Ripper, Hashcat, APIs & OSINT Framework.

CHAPTER 4

DESCRIPTION OF PROPOSED SYSTEM

4.1 SELECTED METHODOLOGY OR PROCESS MODEL

4.1.1 **Network enumeration module:** plays a crucial role in penetration testing, serving as the initial phase to gather essential information about the target network's infrastructure and identify potential entry points for further exploitation. This process involves various techniques and tools aimed at discovering active hosts, network topology, and potential vulnerabilities. Ping and ARP are fundamental tools utilized for basic network discovery, allowing the tester to determine live hosts and their corresponding IP addresses. Live host detection further aids in identifying active systems within the network, essential for subsequent analysis. GatewayIP detection provides insights into the network's routing architecture, aiding in understanding traffic flow and potential choke points. MacChanger is employed to manipulate MAC addresses, allowing testers to obscure their device's identity and evade detection. Masscan facilitates rapid scanning of large networks, enabling testers to efficiently enumerate open ports and services. Hping3 enables advanced packet manipulation and crafting, useful for network reconnaissance and firewall evasion techniques. Nmap, a versatile tool, offers comprehensive capabilities such as OS detection, port scanning, service version detection, and vulnerability analysis using its NSE (Nmap Scripting Engine). These features allow testers to map out network services, identify potential weaknesses, and assess the overall security posture of the target environment. By leveraging these tools in concert, penetration testers can gain invaluable insights into the target network's infrastructure, aiding in the formulation of effective attack strategies and recommendations for mitigating identified vulnerabilities.

4.1.2 **Web enumeration module:** stands as a pivotal phase, enabling thorough reconnaissance of target systems to identify potential vulnerabilities and attack vectors. Anchored in the pursuit of comprehensive understanding and mapping of web infrastructure, web enumeration encompasses various techniques and tools

tailored to unveil crucial insights. Among these, the WHOIS lookup tool emerges as a cornerstone, facilitating the retrieval of domain registration information, including registrant details, registration and expiration dates, and domain name servers. This insight aids in constructing a comprehensive profile of the target organization's digital footprint, essential for subsequent stages of the penetration testing process. Complementing WHOIS lookup, directory enumeration emerges as another indispensable facet, bolstered by tools like Gobuster. This tool delves into web servers, systematically scanning for hidden directories and files, thus uncovering potential entry points or sensitive data repositories. Moreover, subdomain enumeration emerges as a pivotal tactic, leveraging tools such as Sublist3r to exhaustively catalog subdomains associated with the target domain. By scouring public domain repositories and brute-forcing subdomains, Sublist3r aids in broadening the scope of reconnaissance efforts, potentially revealing overlooked attack surfaces ripe for exploitation. Collectively, these tools and techniques underpin the web enumeration phase, furnishing penetration testers with a nuanced understanding of the target's web infrastructure, thus empowering informed decision-making and precise exploitation of vulnerabilities.

4.1.3 *Cryptographic enumeration module:* The process of counting and decoding cryptography is essential for identifying weaknesses and evaluating how robust the cryptographic processes used in a system are. This phase involves several crucial steps aimed at understanding, analyzing, and potentially breaking cryptographic constructs. Initially, text and files undergo conversion into hash representations, a process integral to cryptographic operations for data integrity and authentication. Subsequently, the hash type identification is carried out utilizing specialized tools such as hashidentifier, which employs pattern recognition algorithms to determine the cryptographic hash function employed. Additionally, tools like Hashcat and John the Ripper are instrumental in the exhaustive search for plaintexts corresponding to the given hash values, employing techniques like brute-force attacks, dictionary attacks, and rule-based cracking methodologies. Furthermore, metadata analysis forms an essential part of cryptography enumeration, facilitated by tools like ExifTool, which unveils hidden information embedded within digital files, aiding in the identification of potential cryptographic artifacts or security flaws. The exploration of whitespace steganography and image

steganography further expands the purview of cryptography enumeration, as these techniques involve concealing data within non-visible areas of digital content, posing challenges to traditional cryptographic analysis. Through a comprehensive approach encompassing hash conversion, hash type identification, hash cracking, metadata analysis, and steganographic exploration, cybersecurity professionals can effectively scrutinize cryptographic implementations, identify weaknesses, and reinforce defenses to mitigate potential security risks and safeguard sensitive information.

4.1.4 Osint enumeration module : Open-Source Intelligence (OSINT) is a methodology for gathering information about potential targets and assessing their security posture. Within this framework, leveraging specialized tools such as PhoneNumber Verifier for validating phone numbers and tracking the public IP address with location emerges as indispensable assets. PhoneNumber Verifier enables penetration testers to validate the authenticity of phone numbers associated with target entities, facilitating the identification of potential communication channels and attack vectors. This tool empowers testers to ascertain the validity and reachability of phone numbers, aiding in the reconnaissance phase of penetration testing. Furthermore, the capability to track public IP addresses with location enhances the OSINT arsenal by providing geolocation data essential for mapping out the digital footprint of target systems and entities. By utilizing this functionality, penetration testers can gain insights into the geographical distribution of network assets and potential vulnerabilities. This information enables the formulation of targeted attack strategies and enhances the overall efficacy of penetration testing initiatives. In essence, the integration of specialized OSINT tools like PhoneNumber Verifier and IP address tracking enhances the reconnaissance capabilities of penetration testers, facilitating the identification of vulnerabilities and the mitigation of security risks within target environments.

4.2 ARCHITECTURE / OVERALL DESIGN OF PROPOSED SYSTEM

4.2.1 Class Diagram:

A class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects

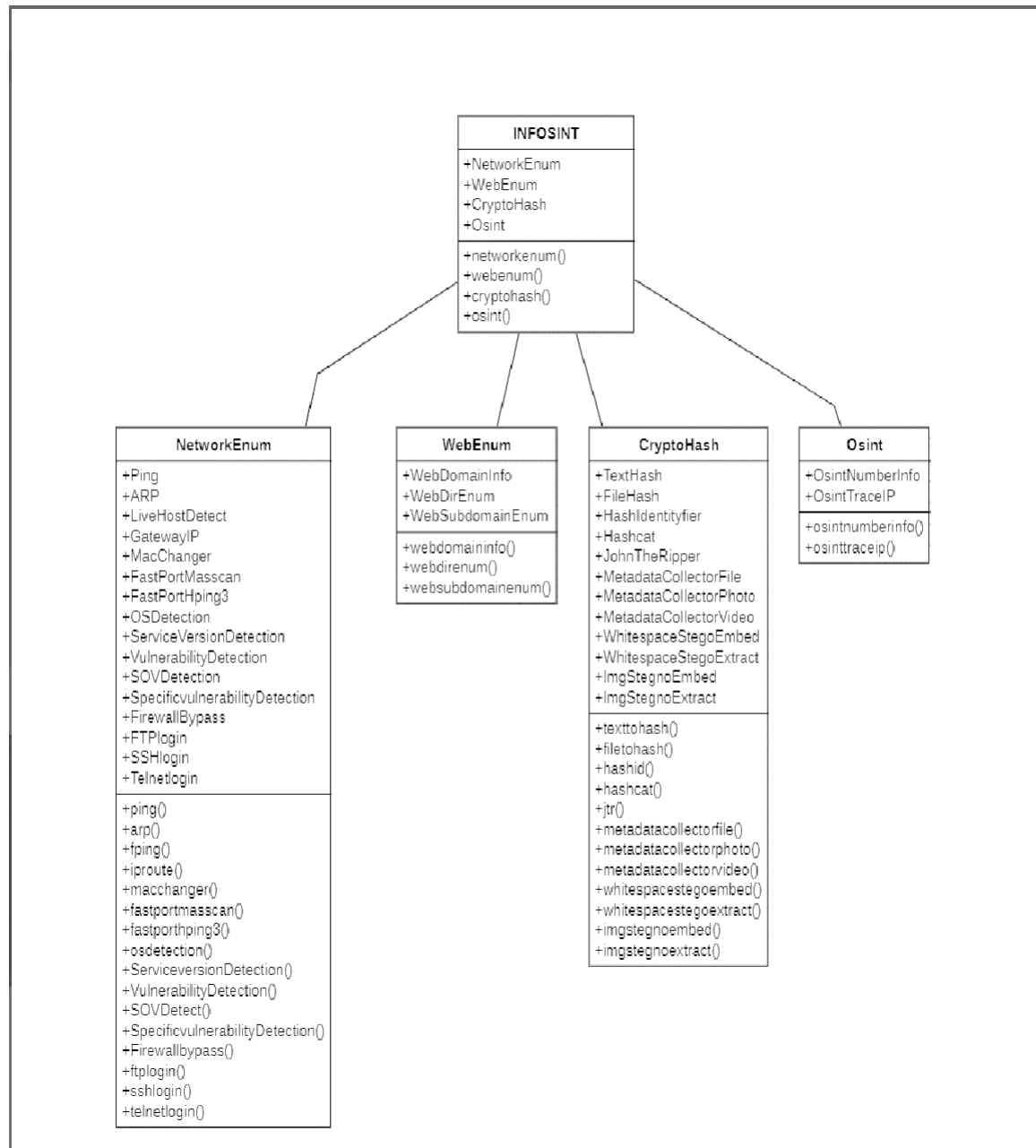


Fig: 4.2.1.1 UML CLASS DIAGRAM

Data flowchart :

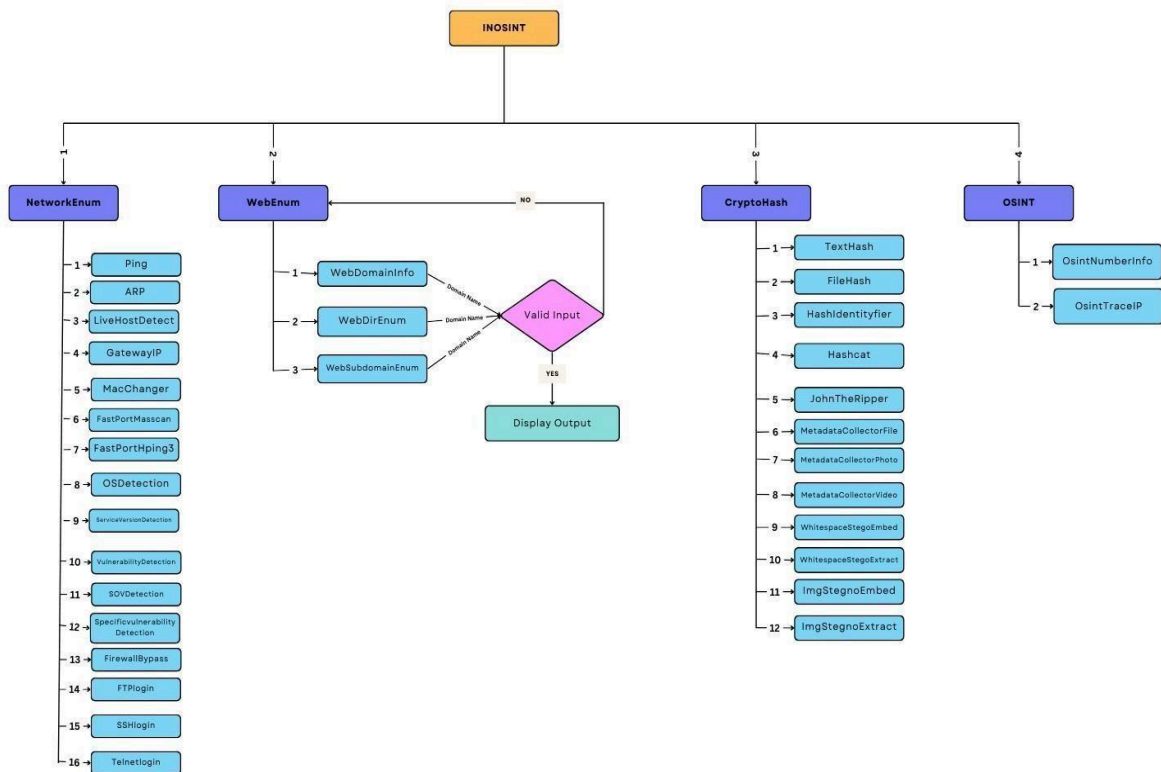


Fig: 4.2.1.2 Data flowchart diagram

4.2.2 Use-Case Diagram :

Use case diagrams to model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform. In this context, a "system" is something being developed or operated, such as a website. The "actors" are people or entities operating under defined roles within the system

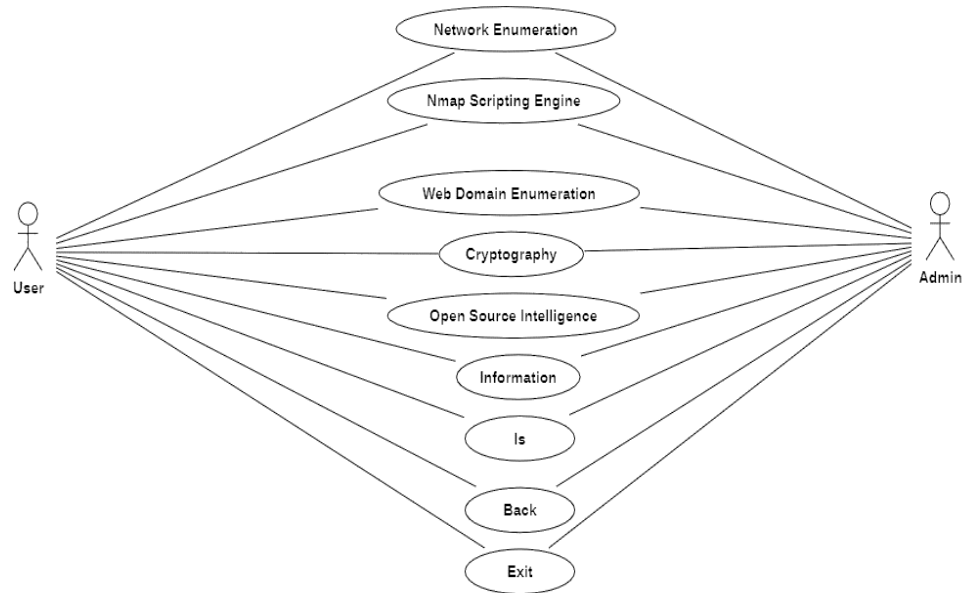


Fig: 4.2.2.1 UML USECASE DIAGRAM

CHAPTER 5

IMPLEMENTATION DETAILS

5.1 DEVELOPMENT SETUP

For our project, we aim to create an environment for security assessments and penetration testing. Our setup revolves around two main components: the host operating system and the virtual operating system. We've meticulously selected the tools and platforms to ensure efficiency and effectiveness in our endeavors.

5.2 DEPENDENCY REQUIREMENTS

Host Operating System:

Our setup supports both Windows 10 or above and Debian 9 or above. This flexibility allows users to choose their preferred environment based on their familiarity and requirements.

Virtual Operating System:

Kali Linux (Debian) 2020 or above is chosen as the virtual operating system. It provides a robust platform for penetration testing and ethical hacking activities with a wide range of pre-installed tools and packages.

Programming Language:

Python 3 and Bash scripting are the primary programming languages utilized in the project. Python offers versatility and ease of use, while Bash scripting facilitates automation and command-line operations.

Terminal Environment:

The terminal environment for executing commands and scripts is set to Bash, providing a familiar and powerful interface for interacting with the system.

Integrated Development Environment (IDE):

PyCharm is selected as the integrated development environment for Python development tasks. Its features such as code completion, debugging, and version control integration enhance the development experience.

Hypervisor Selection:

Users have the option to use either VMware or VirtualBox as the hypervisor for managing virtual machines. Both hypervisors offer features for creating, running, and managing virtualized environments efficiently.

Vulnerable Virtual Machine:

The project utilizes Skynet from TryHackMe or Metasploitable as the vulnerable virtual machine for practicing penetration testing and exploiting security vulnerabilities in a controlled environment.

Security Framework & Tools:

Various security tools and frameworks are incorporated into the project, including:

- Nmap: for network discovery and security auditing.
- NSE (Nmap Scripting Engine): for extending Nmap's functionality with custom scripts.
- Who is-lookup: for retrieving domain registration information.
- Go buster: for directory and file enumeration in web applications.
- Assert finder: for identifying assertions in web applications.
- Hash-Identifier: for identifying the type of hash.
- John the Ripper: for password cracking.
- Hashcat: for advanced password recovery.
- APIs & OSINT (Open-Source Intelligence) Framework: for gathering intelligence and information from publicly available sources.

5.3 TESTING THE MODELS

Introduction Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design, and code generation. Once the source code has been generated, software must be tested to uncover as many errors as possible before delivery to the customer. In order to find the highest possible number of errors, tests must be conducted systematically and

test cases must be designed using disciplined techniques.

Table no: 5.3.1 testing model

Serial no	Test Description	Output	Observed Output
1	If options are not entered	Enter an valid option	As Expected
2	If symbols are entered	Enter an valid option	As Expected
3	If Numbers are entered	Enter an valid option	As Expected

RESULT AND DISCUSSION

RESULT:



CHAPTER 7

CONCLUSIO

N

The INFOSINT project represents a highly adaptable software development endeavor, offering extensive scope for enhancing performance and functionality. Through a comprehensive overview encompassing project objectives, algorithms, and pertinent techniques, INFOSINT stands as a robust tool tailored for penetration testers and Ethical Hackers alike. Central to its design is a user-friendly interface facilitating seamless navigation across diverse functionalities. The project's core functionality is organized into four distinct modules, each serving a pivotal role in Information Gathering and Enumeration processes. These modules, namely Network Enumeration, Web Enumeration, OSINT, and Cryptography, are meticulously designed to cater to specific requirements, offering a range of features tailored to their respective domains. By encapsulating a broad spectrum of capabilities within a cohesive framework, INFOSINT empowers users with versatile tools to streamline their investigative endeavors, thereby contributing to the advancement of cybersecurity practices.

7.1 FUTURE WORK

GUI (Graphical User Interface). Planning to implement vulnerability Scanner based on web attacks. Planning to implement more tool OSINT and Web Enumeration tool for future purpose. Planning to implement this tool in Windows Operating System too. In our upcoming enhancements, we aim to introduce a Graphical User Interface (GUI) for improved user interaction. Additionally, we plan to integrate a Vulnerability Scanner focusing on web attacks, we intend to develop additional tools such as OSINT (Open-Source Intelligence) and Web Enumeration tools to further enrich our offering. Moreover, recognizing the diverse user base, we're expanding compatibility to include the Windows Operating System, broadening accessibility. These enhancements reflect our commitment to continuous improvement and addressing evolving user needs. With these additions, users can

anticipate a more comprehensive and streamlined experience, empowering them to better navigate and address cybersecurity challenges.

REFERENCES: -

- [1] G Jayasurya pal., P Meher Pranay., has proposed "A Survey on Network Penetration Testing" which focussed on all the System along with information gathering to the post exploitation. 2021 2nd International Conference on Intelligent Engineering and Management (ICIEM),IEEE,2021.
- [2] Pengfei Shi., Futong Qin., has proposed "The Penetration Testing Framework for Large-Scale Network Based on Network Fingerprint". 2019 International Conference on Communications, Information System and Computer Engineering (CISCE),IEEE,2019.
- [3] Rodney R Rohrmann., Vincent J Ercolani., has proposed " Large scale port scanning through tor using parallel Nmap" scans to scan large portions of the IPv4 range,2017 IEEE International Conference on Intelligence and Security Informatics (ISI),IEEE,2017.
- [4] Prerna Arote., Karam Veer Arya., has proposed" Detection and Prevention against ARP Poisoning Attack using Modified ICMP and Voting". 2015 International Conference on Computational Intelligence and Networks,IEEE,2015.
- [5] Enos LETSOALO, Sunday OJO., has proposed "Survey of Media Access Control address spoofing attacks detection and prevention techniques in Wireless Networks". 2016 IST-Africa Week Conference.IEEE,2016.
- [6] Lerato Ramahlapane Moila., Mthulisi Velempini., has proposed "An Evaluation of the Effectiveness of Cognitive Radio Ad hoc Networks Routing Protocols". Real-time data quality of service needs is not guaranteed by the current protocols. 2017 IEEE AFRICON,IEEE,2017.
- [7] Sathish A.P. Kumar., Brian Xu., has proposed " Vulnerability Assessment for Security in Aviation Cyber-Physical Systems". We have tested for vulnerabilities in data loaders and other aircraft systems in order to meet the standards for wireless network security set out by the aviation industry 2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud),IEEE,2017.
- [8] Mehr u Nisa.,Kashif Kifayat., has proposed "Detection of Slow Port Scanning Attacks" 2020 International Conference on Cyber Warfare and Security (ICCWS),IEEE,2020.

APPENDIX

A. SOURCE CODE

INFOSINT.sh

```
sudo python3 INFOSINT.py
```

INFOSINT.py

```
#Color
grey  = "\033[1;37m"
red   = "\033[1;31m"
green =
"\033[1;32m" yellow =
"\033[1;33m" cyan
      = "\033[1;96m"
purple = "\033[1;35m"
```

```
def main_menu():
```

```
print(cyan + """)
```

[illegible]

|+| | | ||\ |(| | |)\| | \ || | || | | | |+|

$$\begin{array}{ll} |+\rangle \text{ --- } \rangle \langle \text{ --- } | \rangle \backslash | | \rangle & | \langle \text{ --- } | \rangle \wedge \text{ --- } / / | \\ \rangle \backslash | | \langle \text{ --- } | | \rangle \langle | & |+\rangle \\ |+\rangle \text{ --- } / / \text{ --- } \rangle & \langle \text{ --- } \rangle \backslash \\ \text{--- } / / \text{ --- } \rangle \text{ --- } \langle \rangle / & \backslash | \end{array}$$
 $|+\rangle$

)

Available Scripts

INFOINT()

```
def INFOSINT():  
    from NetworkEnum import networkenum  
    from WebEnum import webenum  
    from CryptoHash import cryptohash  
    from Osint import osint  
  
    inp = (input(cyan + "Choose >> "))  
    if (inp == 'a'):
```

```
    networkenum()  
elif (inp == 'b'):  
    webenum()
```

```
elif (inp == 'c'):
    osint()
elif (inp == 'd'):
    cryptohash()
elif (inp == 'exit'):
    exit()
elif (inp == 'i'):
    print(green
          + """)
```

[illegible]

```

|*
|* Created By :Kaushik Babu R & Kevin Sam Matthew G
*|
|* LinkedIn : https://www.linkedin.com/in/krithik-babu-p-bb97431a9/
*|
|* Project Name : INFOSINT INFOSINT | Informative Open-source
Intelligence Tool *|
|* Created Date : 30.11.2021 *|
|* Domain : Cybersecurity *|
|* Use : Used to enumerate the active host/website
for Penteration Testing *|
|*
|*

```

[illegible]

```
elif (inp == 'x'):
    print(red + "Hint")
    :")
```

```
print(green + "ls " + grey + " - to list scripts.")  
print(green + "back " + grey + " - previous modules.")
```

```

        print(green + "exit " + grey + "- to exit the tool.")
    elif (inp == 'ls'):
        print(green + """
                                Tools available

                                <a> Network Enumeration
                                <b> Web Enumeration
                                <c> Open Source Intelligence (OSINT)
                                <d> Cryptography
                                <i> info
                                <x> Hint

                                """)
    else:
        print(red + "Enter an valid option")

while True:
    INFOSINT()

if __name__ == "__main__":
    main_menu()

```

NetworkEnum.py

```

from NmapComp import nmapcomp
from NmapServ import nmapportserv
from NmapLiveHost import livehost
from NmapOs import nmapos
from NmapNSE import nmapnse

#Color
grey = "\033[1;37m"
red = "\033[1;31m"
green = "\033[1;32m"

```

```
yellow = "\033[1;33m"
cyan = "\033[1;96m"
purple = "\033[1;35m"
```

```
def networkenum():
    from INFOSINT import main_menu
```

```
    print(yellow + ""
```

Available Scripts

```
<a> Nmap Comprehensive (all)
<b> Nmap Service Version Detection
<c> Nmap OS Detection
<d> Nmap Live Host Detection
<e> Nmap Scripting Engine (NSE-ALL)
```

```
""")
```

```
inp = (input(cyan + "Choose >> "))
if (inp == 'a'):
    nmapcomp(
) elif (inp ==
'b'):
    nmappportserv()
elif (inp == 'c'):
    livehost()
elif (inp ==
'd'):
    nmapos()
elif (inp ==
'e'):
    nmapnse()
elif (inp == 'back'):
```

```
main_menu()  
elif (inp == 'exit'):
```

```

        exit()
    elif (inp == 'x'):
        print(red + "Hint
        :")
        print(green + "ls " + grey + " - to list scripts.")
        print(green + "back " + grey + " - previous modules.")
        print(green + "exit " + grey + "- to exit the tool.")
    elif (inp == 'ls'):
        print(green + """

                Tools available

                <a> Nmap Comprehensive (all)
                <b> Nmap Service Version Detection
                <c> Nmap OS Detection
                <d> Nmap Live Host Detection
                <e> Nmap Scripting Engine (NSE-ALL)

                """)
    else:
        print(red + "Enter an valid option")
    while True:
        networkenum()

if __name__ == "__main__":
    networkenum()

```

NmapComp.py

```

#Nmap Comprehensive scan
import os
#Color
grey = "\033[1;37m"

```

```
red = "\033[1;31m"
green = "\033[1;32m"
yellow = "\033[1;33m"
cyan= "\033[0;96m"
purple="\033[0;35m"
```

```
def nmapcomp():
```

```
    IP = input(cyan+"IP >> ")
    print(os.system("nmap -sV -n -v -Pn -p- -T4 " + IP + " -A --open"))
```

```
if __name__=="__main__":
    nmapcomp()
```

NmapServ.py

```
#Nmap Service Detection
```

```
import os
```

```
#Color
```

```
grey = "\033[1;37m"
red = "\033[1;31m"
green = "\033[1;32m"
yellow =
"\033[1;33m" cyan=
"\033[0;96m"
purple="\033[0;35m"
```

```
def nmapportserv():
```

```
    IP = input(cyan+"IP >> ")
    print(os.system("nmap -sV -Pn --min-rate 300 " + IP))
```

```
if __name__ == "__main__":
    nmapportserv()
```

NmapOs.py

```
#Nmap OS scan
import os
#Color
grey = "\033[1;37m"
red = "\033[1;31m"
green = "\033[1;32m"
yellow =
"\033[1;33m" cyan=
"\033[0;96m"
purple="\033[0;35m"

def nmapos():

    IP = input(cyan+"IP >> ")
    print(os.system("sudo nmap -O -T 5 " + IP))

if __name__ == "__main__":
    nmapos()
```

NmapLiveHost.py

```
#Nmap Live Host
import os
#Color
```

```
grey =
"\033[1;37m" red
= "\033[1;31m" green
    = "\033[1;32m"
yellow = "\033[1;33m"
cyan = "\033[0;96m"
purple = "\033[0;35m"
```

```
def livehost():
```

```
    IP = input(cyan+"IP/CIDR >> ")
    print(os.system("nmap -sP " + IP + " | grep -o 192.168.*.*"))
```

```
if __name__=="__main__"
    ": livehost()
```

NmapNSE.py

```
#Nmap
```

```
NSE import
```

```
os
```

```
#Color
```

```
grey = "\033[1;37m"
red = "\033[1;31m"
green = "\033[1;32m"
yellow =
"\033[1;33m" cyan=
"\033[0;96m"
purple="\033[0;35m"
```

```
def nmapnse():
```

```
    IP = input(cyan+"IP >> ")
```

```
    print(os.system("nmap --script=vuln "+ IP ))
```

```
if __name__=="__main__":
    nmapnse()
```

WebEnum.py

```
from WebDomainInfo import webdomaininfo
from WebDirEnum import webdirenum
from WebSubdomainEnum import websubdomainenum
```

```
# Color
```

```
grey = "\033[1;37m"
```

```
red = "\033[1;31m"
```

```
green =
```

```
"\033[1;32m" yellow =
```

```
"\033[1;33m" cyan
```

```
    = "\033[1;96m"
```

```
purple = "\033[1;35m"
```

```
def webenum():
```

```
    from INFOSINT import main_menu
```

```
    print(yellow + """
```

```
        Available Scripts
```

```
        <a> Web Domain Information
```

```
        <b> Web Directory Enumeration
```

```
        <c> Web Subdomain Enumeration
```

```
    """)
```

```
    inp = (input(cyan + "Choose >> "))
```

```
    if (inp == 'a'):
```

```
        webdomaininfo()
```

```
elif (inp == 'b'):
```

```

        webdirenum()
elif (inp == 'c'):
    websubdomainenum()
elif (inp == "back"):
    main_menu()
elif (inp == 'exit'):
    exit()

```

```

elif (inp == 'ls'):
    print(green + ""

```

Available Scripts

- <a> Web Domain Information
- Web Directory Enumeration
- <c> Web Subdomain Enumeration

```

        """)
elif (inp == 'x'):
    print(red + "Hint
:")
    print(green + "ls " + grey + " - to list scripts.")
    print(green + "back " + grey + " - previous modules.")
    print(green + "exit " + grey + "- to exit the tool.")
else:
    print(red + "Enter an valid option")
while True:
    webenum()

if __name__ == "__main__":
    webenum()

```

WebDomainInfo.py

```
# Web domain info
```

```
import os
```

```
#Color
```

```
grey = "\033[1;37m"
```

```
red = "\033[1;31m"
```

```
green = "\033[1;32m"
```

```
yellow =
```

```
"\033[1;33m" cyan=
```

```
"\033[0;96m"
```

```
purple="\033[0;35m"
```

```
def webdomaininfo():
```

```
    domain = input(cyan+"Domain >> ")
```

```
    print(os.system("whois " + domain))
```

```
if __name__=="__main__"
```

```
    ": webdomaininfo()
```

WebDirEnum.py

```
#Hydra Web Enum
```

```
import os
```

```
#Color
```

```
grey = "\033[1;37m"
```

```
red = "\033[1;31m"
```

```
green = "\033[1;32m"
```

```
yellow =
```

```
"\033[1;33m" cyan=
```

```
"\033[0;96m"
```

purple="\033[0;35m"

```
def webdirenum():

    domain = input(cyan+"Domain >> ")
    print(os.system("gobuster dir -e -u " + domain + "
-W
/usr/share/wordlists/dirb/big.txt -t 40 -q"))

if __name__=="__main__":
    webdirenum()
```

WebSubdomainEnum.py

```
# Web sub domain enum

import os

#Color
grey = "\033[1;37m"
red = "\033[1;31m"
green = "\033[1;32m"
yellow = "\033[1;33m"
cyan = "\033[0;96m"
purple = "\033[0;35m"

def websubdomainenum():

    domain = input(cyan+"Domain >> ")
    print(os.system("sudo assetfinder -subs-only " +
domain))

if __name__=="__main__":
    websubdomainenum()
```

Osint.py

```
from OsintNumberInfo import osintnumberinfo
from OsintTraceIP import osinttraceip
```

```
#Color
```

```
grey  = "\033[1;37m"
```

```
red   = "\033[1;31m"
```

```
green =
```

```
"\033[1;32m" yellow =
```

```
"\033[1;33m" cyan
```

```
      = "\033[1;96m"
```

```
purple = "\033[1;35m"
```

```
def osint():
```

```
    from INFOSINT import main_menu
```

```
    print(yellow + ""
```

```
                Available Scripts
```

```
                <a> Number Info
```

```
                <b> Trace (Global) IP Address
```

```
        """)
```

```
    inp = (input(cyan + "Choose >> "))
```

```
    if (inp == 'a'):
```

```
        osintnumberinfo()
```

```
    elif (inp == 'b'):
```

```
        osinttraceip()
```

```
    elif (inp == 'back'):
```

```
        main_menu()
```

```
    elif (inp == 'exit'):
```

```
        exit()
```

```
elif (inp == 'ls'):
    print(green + ""
```

Available Scripts

<a> Number Info

 Trace (Global) IP Address

```
        """)
elif (inp == 'x'):
    print(red + "Hint
:")
    print(green + "ls " + grey + " - to list scripts.")
    print(green + "back " + grey + " - previous modules.")
    print(green + "exit " + grey + "- to exit the tool.")
else:
    print(red + "Enter an valid option")
while True:
    osint()
```

```
if __name__ == "__main__":
    osint()
```

OsintNumberInfo.py

```
# NumberInfo
import requests
```

```
#color
grey =
"\033[1;37m" red =
```

"\033[1;31m" green
= "\033[1;32m"

```
yellow = "\033[1;33m"
```

```
cyan= "\033[0;96m"
```

```
purple="\033[0;35m"
```

```
def osintnumberinfo():
```

```
    phonenum = input(red + "Enter Mobile Number with country code >> ")
```

```
    url =
```

```
("http://apilayer.net/api/validate?access_key=cd3af5f7d1897dc1707c47d05c3759f
d&number="+phonenum)
```

```
    resp = requests.get(url)
```

```
    details = resp.json()
```

```
    print("")
```

```
    print(cyan+"[+] Target                : "+ details['number'])
```

```
    print(cyan+"[+] Target Local Format      : "+
```

```
    details['local_format'])
```

```
    print(cyan+"[+] Target International Format : "+
```

```
    details['international_format']) print(cyan+"[+] Prefix    : "+
```

```
    details['country_prefix'])
```

```
    print(cyan+"[+] Carrier                : "+ details['carrier'])
```

```
    print(cyan+"[+] Code                  : "+ details['country_code'])
```

```
    print(cyan+"[+] Location                : "+ details['location'])
```

```
    print(cyan+"[+] Country                : "+ details['country_name'])
```

```
if __name__=="__main__"
```

```
    ": osintnumberinfo()
```

OsintTracelP.py

```
import requests
```

```
import
```

```
webbrowser
```

```
#Color
```

```
grey = "\033[1;37m"
```

red = "\033[1;31m"

green = "\033[1;32m"

yellow =

"\033[1;33m"

```
cyan= "\033[0;96m"
purple="\033[0;35m"
```

```
def osinttraceip():
```

```
    ipinfo={}
    ip=input(red+"Ip address >>
    ")
    url="http://ip-api.com/json/"+i
    p r=requests.get(url)
    ipinfo=r.json()
    if ipinfo['status'] == 'success':

        lat=ipinfo['lat']
        lon=ipinfo['lon']
        print(green+"Ip location Found !!!")
        print('\n')
        print('[+] Country    :',ipinfo['country'])
        print('[+] Region Name :
        ',ipinfo['regionName']) print('[+] City    :
        ',ipinfo['city'])
        print('[+] Time zone  :',ipinfo['timezone'])
        print('[+] ISP        :',ipinfo['isp'])
        print(cyan+'Opening Location in
        browser')
        mapurl = "https://maps.google.com/maps?q=%s,+%s" % (lat, lon)
        webbrowser.open(mapurl, new=2)
        print("")
    else:
        print(red+"Ip location Not Found !!!")
```

```
if __name__=="__main__"
    ": osinttraceip()
```

CryptoHash.py

```
from CryptoHashIdentifier import cryptohashid
from CryptoHashcat import cryptohashcat
from CryptoJohn import cryptojohn
```

```
#Color
```

```
grey = "\033[1;37m"
```

```
red = "\033[1;31m"
```

```
green =
```

```
"\033[1;32m" yellow =
```

```
"\033[1;33m" cyan
```

```
    = "\033[1;96m"
```

```
purple = "\033[1;35m"
```

```
def cryptohash():
```

```
    from INFOSINT import main_menu
```

```
    print(yellow + ""
```

```
                Available Scripts
```

```
                <a> Hash Identifier
```

```
                <b> Hash Cracking - <John The Ripper>
```

```
                <c> Hash Cracking - <HashCat>
```

```
        """)
```

```
    inp = (input(cyan + "Choose >> "))
```

```
    if (inp == 'a'):
```

```
        cryptohashid()
```

```
    elif (inp == 'b'):
```

```
        cryptojohn()
```

```
    elif (inp == 'c'):
```

```
        cryptohashcat()
```

```
    elif (inp == 'back'):
```

main_menu()

```

elif (inp == 'exit'):
    exit()
elif (inp == 'x'):
    print(red + "Hint
:")
    print(green + "ls " + grey + " - to list scripts.")
    print(green + "back " + grey + " - previous modules.")
    print(green + "exit " + grey + "- to exit the tool.")
elif (inp == 'ls'):
    print(green + """
                Tools available

                <a> Hash Identifier
                <b> Hash Cracking - <John The Ripper>
                <c> Hash Cracking - <HashCat>

                """)
else:
    print(red + "Enter an valid option")
while True:
    cryptohash()

if __name__ == "__main__":
    cryptohash()

```

CryptoHashIdentifier.py

```

#Hydra Web Enum
import os

#Color
grey =

```

"\033[1;37m" red =

"\033[1;31m"

```

green = "\033[1;32m"
yellow = "\033[1;33m"
cyan= "\033[0;96m"
purple="\033[0;35m"

def cryptohashid():

    hash = input(cyan+"HASH >> ")
    print(os.system("haiti -e "+hash))

if _name__=="__main__"
    ": cryptohashid()

```

CryptoJohn.py

```

#john the ripper
import os

#Color
grey = "\033[1;37m"
red = "\033[1;31m"
green = "\033[1;32m"
yellow =
"\033[1;33m" cyan=
"\033[0;96m"
purple="\033[0;35m"

def cryptojohn():

    hash = input(cyan+"HASH >> ")
    htype = input(cyan+"HASH TYPE>> ")
    print(os.system(' echo '+ hash +' >> hash.txt'))
    print(os.system("sudo john --format="+ htype +" --wordlist=rockyou.txt hash.txt"))

```

```
if __name__ == "__main__":
    cryptojohn()
```

CryptoHashcat.py

```
#Hashcat
```

```
import os
```

```
#Color
```

```
grey = "\033[1;37m"
```

```
red = "\033[1;31m"
```

```
green = "\033[1;32m"
```

```
yellow =
```

```
"\033[1;33m" cyan=
```

```
"\033[0;96m"
```

```
purple="\033[0;35m"
```

```
def cryptohashcat():
```

```
    hash = input(cyan+"HASH >> ")
```

```
    htype = input(cyan+"HASH NNUMBER >> ")
```

```
    print(os.system(' echo '+ hash +' >> hash.txt'))
```

```
    print(os.system(" sudo hashcat -m "+ htype +" -a 0 hash.txt rockyou.txt "))
```

```
if __name__ == "__main__":
```

```
    cryptohashcat()
```

B. SCREENSHOTS

[illegible]

B.1: info about tool

```
Choose >> 1

Available Scripts

[1] Ping
[2] ARP (ARP Live Host)
[3] Live Host detection
[4] Find Gateway IP
[5] MAC Changer
[6] Fast Port Scan (masscan)
[7] Fast Port Scan (hgping3)
[8] Service Version Scan
[9] OS Detection
[10] Vulnerability Detection with Report
[11] SDV Detection Scan
[12] Specific vulnerability
[13] Firewall Bypass
[14] FTP Login
[15] SSH Login
[16] TELNET Login
```

B.2: network enum menu

```

Choose >> a
IP >> 10.10.234.59
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times may be slower.
Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-28 23:28 IST
NSE: Loaded 155 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 23:28
Completed NSE at 23:28, 0.00s elapsed
Initiating NSE at 23:28
Completed NSE at 23:28, 0.00s elapsed
Initiating NSE at 23:28
Completed NSE at 23:28, 0.00s elapsed
Initiating SYN Stealth Scan at 23:28
Scanning 10.10.234.59 [65535 ports]
Discovered open port 445/tcp on 10.10.234.59
Discovered open port 22/tcp on 10.10.234.59
Discovered open port 143/tcp on 10.10.234.59
Discovered open port 80/tcp on 10.10.234.59
Discovered open port 110/tcp on 10.10.234.59
Discovered open port 139/tcp on 10.10.234.59
Completed SYN Stealth Scan at 23:29, 45.49s elapsed (65535 total ports)
Initiating Service scan at 23:29
Scanning 6 services on 10.10.234.59
Completed Service scan at 23:29, 11.47s elapsed (6 services on 1 host)
Initiating OS detection (try #1) against 10.10.234.59
Retrying OS detection (try #2) against 10.10.234.59
Retrying OS detection (try #3) against 10.10.234.59
Retrying OS detection (try #4) against 10.10.234.59
Retrying OS detection (try #5) against 10.10.234.59
Initiating Traceroute at 23:29
Completed Traceroute at 23:29, 3.02s elapsed
NSE: Script scanning 10.10.234.59.
Initiating NSE at 23:29
Completed NSE at 23:29, 5.11s elapsed
Initiating NSE at 23:29
Completed NSE at 23:29, 2.40s elapsed
Initiating NSE at 23:29
Completed NSE at 23:29, 0.00s elapsed
Nmap scan report for 10.10.234.59
Host is up (0.15s latency).
Not shown: 65528 closed tcp ports (reset), 1 filtered tcp port (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|   2048 99:23:31:bb:b1:e9:43:b7:56:94:4c:b9:e8:21:46:c5 (RSA)
|   256 57:c0:75:02:71:2d:19:31:83:db:e4:fe:67:96:68:cf (ECDSA)
|_   256 46:fa:4e:fc:10:a5:4f:57:57:d0:6d:54:f6:c3:4d:fe (ED25519)

```

B.3: nmap comprehensive

```

22/tcp open  ssh          OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|   2048 99:23:31:bb:b1:e9:43:b7:56:94:4c:b9:e8:21:46:c5 (RSA)
|   256 57:c0:75:02:71:2d:19:31:83:db:e4:fe:67:96:68:cf (ECDSA)
|_  256 46:fa:4e:fc:10:a5:4f:57:57:d0:6d:54:f6:c3:4d:fe (ED25519)
80/tcp open  http           Apache httpd 2.4.18 ((Ubuntu))
|_ http-server-header: Apache/2.4.18 (Ubuntu)
|_ http-methods:
|_   Supported Methods: POST OPTIONS GET HEAD
|_ http-title: Skynet
110/tcp open  pop3          Dovecot pop3d
|_ pop3-capabilities: TOP CAPA SASL UIDL AUTH-RESP-CODE PIPELINING RESP-CODES
139/tcp open  netbios-ssn   Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
143/tcp open  imap          Dovecot imapd
|_ imap-capabilities: LOGIN-REFERRALS OK LOGINDISABLED0001 listed more post-login have capabilities IMAP4rev1 IDLE Pre-login ID LITERAL+ ENABLE SASL-IR
445/tcp open  netbios-ssn   Samba smbd 4.3.11-Ubuntu (workgroup: WORKGROUP)
No exact OS matches for host (if you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.92%E=4%D=5/1%OT=22%CT=1%CU=31352%PV=Y%DS=5%DC=T%G=Y%TM=626E64B8
OS:%P=x86_64-pc-linux-gnu)SEQ(SP=102%GCD=1%ISR=10F%TI=Z%CI=I%II=I%TS=8)OPS(
OS:O1=M506ST11NW7%O2=M506ST11NW7%O3=M506NNT11NW7%O4=M506ST11NW7%O5=M506ST11
OS:NW7%O6=M506ST11)WIN(W1=68DF%W2=68DF%W3=68DF%W4=68DF%W5=68DF%W6=68DF)ECN(
OS:R=Y%DF=Y%T=40%W=6903%O=M506NNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%F=AS
OS:%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(R=
OS:Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=
OS:%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T
OS:=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD=
OS:S)

Uptime guess: 0.024 days (since Sun May  1 15:40:41 2022)
Network Distance: 5 hops
TCP Sequence Prediction: Difficulty=258 (Good luck!)
IP ID Sequence Generation: All zeros
Service Info: Host: SKYNET; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_ clock-skew: mean: 1h40m05s, deviation: 2h53m12s, median: 4s
|_ smb2-time:
|   date: 2022-05-01T10:45:11
|   start_date: N/A
|_ smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_ message_signing: disabled (dangerous, but default)
|_ nbstat: NetBIOS name: SKYNET, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
|_ Names:
|   SKYNET<00>          Flags: <unique><active>
|   SKYNET<03>          Flags: <unique><active>
|   SKYNET<20>          Flags: <unique><active>

```

B.4: nmap comprehensive

```

Available Scripts
<a> Nmap Comprehensive (all)
<b> Nmap Service Version Detection
<c> Nmap OS Detection
<d> Nmap Live Host Detection
<e> Nmap Scripting Engine (NSE-ALL)

Choose >> b
IP >> 10.10.148.218
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-01 16:48 IST
Nmap scan report for 10.10.148.218
Host is up (0.16s latency).
Not shown: 994 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http         Apache httpd 2.4.18 ((Ubuntu))
110/tcp   open  pop3         Dovecot pop3d
139/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
143/tcp   open  imap         Dovecot imapd
445/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
Service Info: Host: SKYNET; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 17.20 seconds
0

```

B.5: nmap service ver detection

```
Available Scripts

<a> Nmap Comprehensive (all)
<b> Nmap Service Version Detection
<c> Nmap OS Detection
<d> Nmap Live Host Detection
<e> Nmap Scripting Engine (NSE-ALL)

Choose >> c
IP >> 10.10.148.218
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-01 16:28 IST
Nmap scan report for 10.10.148.218
Host is up (0.15s latency).
Not shown: 994 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
110/tcp   open  pop3
139/tcp   open  netbios-ssn
143/tcp   open  imap
445/tcp   open  microsoft-ds
Aggressive OS guesses: Linux 5.4 (95%), Linux 3.10 - 3.13 (95%), ASUS RT-N56U WAP (Linux 3.4) (95%), Linux 3.16 (95%), Linux 3.1 (93%), Linux 3.2 (93%), AXIS
210A or 211 Network Camera (Linux 2.6.17) (92%), Sony Android TV (Android 5.0) (92%), Android 7.1.1 - 7.1.2 (92%), Linux 3.13 - 4.4 (92%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 5 hops

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.66 seconds
0
```

B.6: nmap os

```
Choose >> 2

Available Scripts

[1] Web Domain Information
[2] Web Directory Enumeration
[3] Web Subdomain Enumeration

Choose >> |
```

```
Choose >> 2

Available Scripts

[1] Web Domain Information
[2] Web Directory Enumeration
[3] Web Subdomain Enumeration

Choose >> |
```

B.7: web enum menu

```

Available Scripts

<a> Web Domain Information
<b> Web Directory Enumeration
<c> Web Subdomain Enumeration

Choose >> a
Domain >> google.com
Domain Name: GOOGLE.COM
Registry Domain ID: 2138514_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.markmonitor.com
Registrar URL: http://www.markmonitor.com
Updated Date: 2019-09-09T15:39:04Z
Creation Date: 1997-09-15T04:00:00Z
Registry Expiry Date: 2028-09-14T04:00:00Z
Registrar: MarkMonitor Inc.
Registrar IANA ID: 292
Registrar Abuse Contact Email: abusecomplaints@markmonitor.com
Registrar Abuse Contact Phone: +1.2083895740
Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
Domain Status: serverDeleteProhibited https://icann.org/epp#serverDeleteProhibited
Domain Status: serverTransferProhibited https://icann.org/epp#serverTransferProhibited
Domain Status: serverUpdateProhibited https://icann.org/epp#serverUpdateProhibited
Name Server: NS1.GOOGLE.COM
Name Server: NS2.GOOGLE.COM
Name Server: NS3.GOOGLE.COM
Name Server: NS4.GOOGLE.COM
DNSSEC: unsigned
URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
>>> Last update of whois database: 2022-05-01T11:25:12Z <<<

For more information on Whois status codes, please visit https://icann.org/epp

NOTICE: The expiration date displayed in this record is the date the
registrar's sponsorship of the domain name registration in the registry is
currently set to expire. This date does not necessarily reflect the expiration
date of the domain name registrant's agreement with the sponsoring

```

B.8: web domain info

```

Available Scripts

<a> Web Domain Information
<b> Web Directory Enumeration
<c> Web Subdomain Enumeration

Choose >> b
Domain >> 10.10.148.218
http://10.10.148.218/.htpasswd      (Status: 403) [Size: 278]
http://10.10.148.218/admin        (Status: 301) [Size: 314] [--> http://10.10.148.218/admin/]
http://10.10.148.218/ai           (Status: 301) [Size: 311] [--> http://10.10.148.218/ai/]
http://10.10.148.218/config       (Status: 301) [Size: 315] [--> http://10.10.148.218/config/]
http://10.10.148.218/css          (Status: 301) [Size: 312] [--> http://10.10.148.218/css/]
http://10.10.148.218/js           (Status: 301) [Size: 311] [--> http://10.10.148.218/js/]
http://10.10.148.218/server-status (Status: 403) [Size: 278]
http://10.10.148.218/squirrelmail (Status: 301) [Size: 321] [--> http://10.10.148.218/squirrelmail/]
0

Available Scripts

<a> Web Domain Information
<b> Web Directory Enumeration
<c> Web Subdomain Enumeration

Choose >> 

```

B.9: web directory enum

```
Choose >> 3

Available Scripts

[1] Number Info
[2] Trace (Global) IP Address

Choose >> |
```

B.10: osint menu

```
Available Scripts

<a> Number Info
<b> Trace (Global) IP Address

Choose >> a
Enter Mobile Number with country code >> 919566952654

[+] Target : 919566952654
[+] Target Local Format : 09566952654
[+] Target International Format : +919566952654
[+] Prefix : +91
[+] Carrier : Bharti Airtel Ltd
[+] Code : IN
[+] Location : Tamil Nadu
[+] Country : India (Republic of)

Available Scripts

<a> Number Info
<b> Trace (Global) IP Address

Choose >> |
```

B.11: number info

```
Available Scripts

<a> Number Info
<b> Trace (Global) IP Address

Choose >> b
Ip address >> 202.21.42.239
Ip location Found !!!

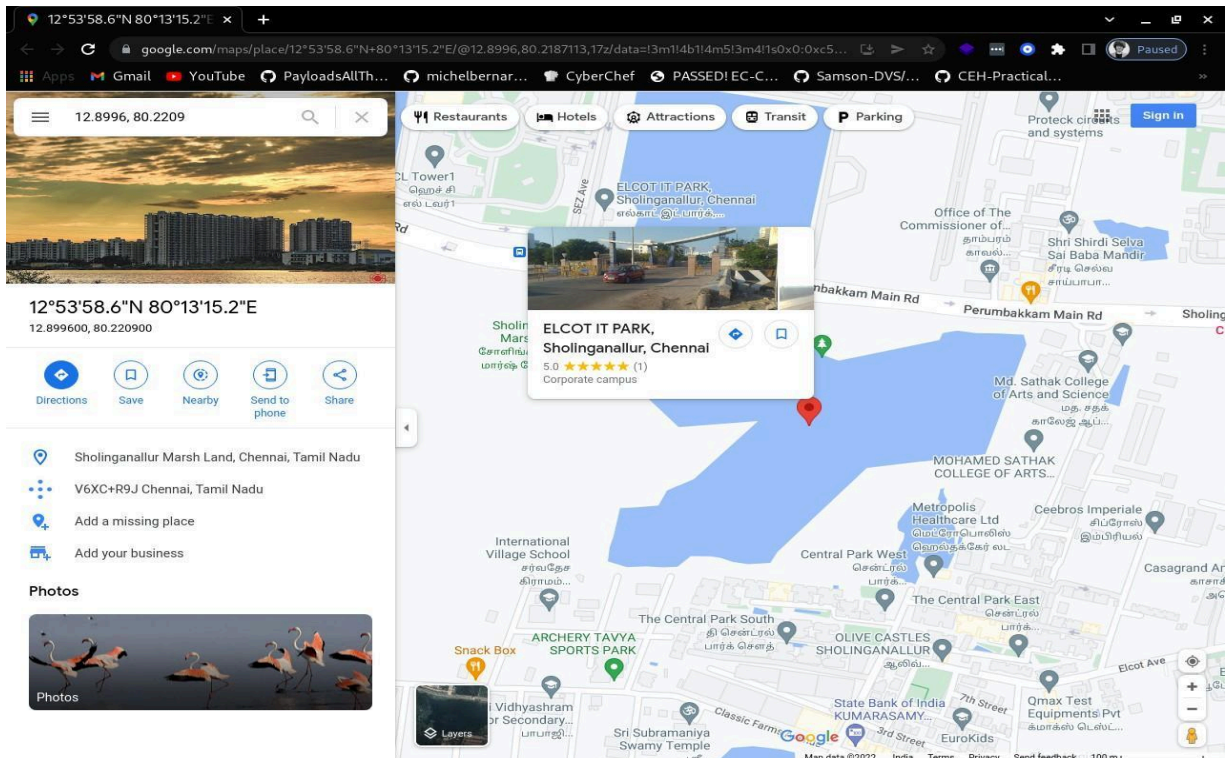
[+] Country : India
[+] Region Name : Tamil Nadu
[+] City : Chennai
[+] Time zone : Asia/Kolkata
[+] ISP : RailTel Corporation
Opening Location in browser

Available Scripts

<a> Number Info
<b> Trace (Global) IP Address

Choose >> [2258:2290:0502/050412.372097:ERROR:bus.cc(398)] Failed to connect to the bus: Could not parse server address: Unknown address type (examples of valid types are "tcp" and on UNIX "unix")
[2258:2290:0502/050412.373192:ERROR:bus.cc(398)] Failed to connect to the bus: Could not parse server address: Unknown address type (examples of valid types are "tcp" and on UNIX "unix")
[2258:2290:0502/050412.495744:ERROR:bus.cc(398)] Failed to connect to the bus: Could not parse server address: Unknown address type (examples of valid types are "tcp" and on UNIX "unix")
[2258:2290:0502/050412.495834:ERROR:bus.cc(398)] Failed to connect to the bus: Could not parse server address: Unknown address type (examples of valid types are "tcp" and on UNIX "unix")
[2258:2290:0502/050412.559109:ERROR:token_service_table.cc(139)] Failed to decrypt token for service AccountId-102963563849787761033
libva error: vaGetDriverNameByIndex() failed with unknown libva error, driver_name = (null)
[2297:2297:0502/050412.632708:ERROR:viz_main_impl.cc(185)] Exiting GPU process due to errors during initialization
[2258:2290:0502/050413.110190:ERROR:bus.cc(398)] Failed to connect to the bus: Could not parse server address: Unknown address type (examples of valid types are "tcp" and on UNIX "unix")
[2258:2290:0502/050413.110265:ERROR:bus.cc(398)] Failed to connect to the bus: Could not parse server address: Unknown address type (examples of valid types are "tcp" and on UNIX "unix")
[2258:2258:0502/050413.485798:ERROR:account_info_fetcher.cc(62)] OnGetTokenFailure: Invalid credentials (credentials missing).
```

B.12: trace global ip



B.13: trace global ip

```

Available Scripts
<a> Hash Identifier
<b> Hash Cracking - <John The Ripper>
<c> Hash Cracking - <HashCat>

Choose >> b
HASH >> 9f9d51bc70ef21ca5c14f307980a29d8
HASH TYPE>> raw-md5
0
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=6
Press 'q' or Ctrl-C to abort, almost any other key for status
bob (?)
1g 0:00:00:00 DONE (2022-05-02 07:07) 33.33g/s 563200p/s 563200c/s 563200C/s kikito..felton
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
0

Available Scripts
<a> Hash Identifier
<b> Hash Cracking - <John The Ripper>
<c> Hash Cracking - <HashCat>

Choose >> 

```

B.14: cryptographic menu

```
Choose >> 4

Available Scripts

[1] Convert Text to Hash using MD5, SHA1, SHA256, SHA512
[2] Convert File to Hash using MD5, SHA1, SHA256, SHA512
[3] Hash Identifier
[4] Hashcat
[5] John The Ripper
[6] Metadata collector - File
[7] Metadata collector - Photo
[8] Metadata collector - Video
[9] Whitespace Steganography Embed
[10] Whitespace Steganography Extract
[11] Image Steganography Embed
[12] Image Steganography Extract

Choose >> |
```

B.15: john the ripper

```
Choose >> d

Available Scripts

<a> Hash Identifier
<b> Hash Cracking - <John The Ripper>
<c> Hash Cracking - <HashCat>

Choose >> a
HASH >> cbfdac6008f9cab4083784cbd1874f76618d2a97
SHA-1 [HC: 100] [JtR: raw-sha1]
RIPEMD-160 [HC: 6000] [JtR: ripemd-160]
Double SHA-1 [HC: 4500]
Haval-160 (3 rounds) [JtR: dynamic_190]
Haval-160 (4 rounds) [JtR: dynamic_200]
Haval-160 (5 rounds) [JtR: dynamic_210]
Tiger-160
HAS-160
LinkedIn [HC: 190] [JtR: raw-sha1-linkedin]
Skein-256(160)
Skein-512(160)
MangosWeb Enhanced CMS [JtR: dynamic_35]
MangosWeb Enhanced CMS v2 [JtR: dynamic_36]
sha1(uppercase($username).:$pass) [JtR: dynamic_35]
sha1($username.:$pass) [JtR: dynamic_36]
sha1(sha1(sha1($pass))) [HC: 4600]
sha1(md5($pass)) [HC: 4700] [JtR: dynamic_23]
sha1($pass.$salt) [HC: 110] [JtR: dynamic_24]
sha1($salt.$pass) [HC: 120] [JtR: dynamic_25]
sha1(unicode($pass).$salt) [HC: 130]
sha1($salt.unicode($pass)) [HC: 140]
HMAC-SHA1 (key = $pass) [HC: 150] [JtR: hmac-sha1]
HMAC-SHA1 (key = $salt) [HC: 160] [JtR: hmac-sha1]
sha1($salt.$pass.$salt) [HC: 4710]
SMF [JtR: dynamic_37]
sha1(lowercase($username).$pass) [JtR: dynamic_37]
Wolt3BB [JtR: dynamic_38]
sha1($salt.sha1($salt.sha1($pass))) [JtR: dynamic_38]
net-sha1 [JtR: dynamic_40]
sha1($salt.pad20($pass)) [JtR: dynamic_40]
sha1($salt.sha1($pass.$salt)) [HC: 24300]
Ruby on Rails Restful Auth (one round, no sitekey) [HC: 27200]
MySQL5.x [HC: 300] [JtR: mysql-sha1]
MySQL4.1 [HC: 300] [JtR: mysql-sha1]
Cisco Type 7
BigCrypt [JtR: bigcrypt]
Umbraco HMAC-SHA1 [HC: 24800]
```

B.16: john the ripper

```

* Raw-Hash

ATTENTION! Pure (unoptimized) backend kernels selected.
Pure kernels can crack longer passwords, but drastically reduce performance.
If you want to switch to optimized kernels, append -O to your commandline.
See the above message to find out about the exact limits.

Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 1 MB

Dictionary cache hit:
* Filename..: rockyou.txt
* Passwords.: 14344384
* Bytes.....: 139921497
* Keyspace..: 14344384

cbfdac6008f9cab4083784cbd1874f76618d2a97:password123

Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 100 (SHA1)
Hash.Target.....: cbfdac6008f9cab4083784cbd1874f76618d2a97
Time.Started.....: Mon May  2 06:59:14 2022 (0 secs)
Time.Estimated...: Mon May  2 06:59:14 2022 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.....: File (rockyou.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 716.9 kH/s (0.37ms) @ Accel:512 Loops:1 Thr:1 Vec:8
Recovered.....: 1/1 (100.00%) Digests
Progress.....: 3072/14344384 (0.02%)
Rejected.....: 0/3072 (0.00%)
Restore.Point....: 0/14344384 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: 123456 -> dangerous
Hardware.Mon.#1..: Util: 17%

Started: Mon May  2 06:58:51 2022
Stopped: Mon May  2 06:59:16 2022
0

Available Scripts

<a> Hash Identifier
<b> Hash Cracking - <John The Ripper>
<c> Hash Cracking - <HashCat>

Choose >>

```

B.17: hashcat

INFOSINT | Informative Open-source Intelligence tool for Penetration Testers

“Dr. M. Selvi
Assistant Professor, Cse Dept
Sathyabama Institute of Science and
Technology, Chennai
selvi.cse@sathyabama.ac.in

R Kaushik Babu
Dept. of Computer Science and Engineering
Sathyabama Institute of Science and
Technology, Chennai
kaushikbabur@gmail.com

Kevin Sam Matthew G
Dept. of Computer Science and
Sathyabama Institute of Science
and Technology, Chennai
kevinsammatthewg@gmail.com

Abstract: Information Gathering and Enumeration / Scanning play a vital role in Cybersecurity. The more they enumerate, the more they get access to the System. But the enumeration part of Cybersecurity is a little bit time-consuming because of using multiple tools, comprehensive strategies, Custom Techniques and etc. INFOSINT or General Enumeration is a CLI (Command Line Interface) based Cybersecurity Tool, which is used in Linux (Debian) Operating system. It is Developed using Python3 & Bash. INFOSINT helps security professionals to enumerate Information in various technologies like Networking, Web Applications, & Open Source Intelligence (OSINT) . INFOSINT uses the utilities of Offensive Security tools like Nmap, Nmap Scripting Engine (NSE), Gobuster, Assertfinder, Whois-lookup, OSINT Framework ,and API. This tool provides security professionals with a streamlined approach, enabling them to efficiently harness the complete capabilities of the listed tools through a single comprehensive script. Network enumeration involves the retrieval of usernames and information on groups, shares, and services from networked computers. On the other hand, web enumeration is a more complex task compared to host discovery, requiring the attacker to employ specific methods to uncover comprehensive details about a target web service. The OSINT framework primarily centers on collecting information from freely available tools and resources, with the objective of assisting individuals in locating free OSINT resources effectively.

Keywords: Cyber security, Penetration,

I. INTRODUCTION

a) What is Cybersecurity :

Proactive steps taken to protect networks, software, and systems from online dangers and attacks are referred to as cybersecurity. The goals of these cyberattacks usually include extorting money from users, corrupting, modifying, or gaining illegal access to sensitive data, or interfering with ordinary corporate activities. Ensuring robust cybersecurity has become increasingly challenging in the present landscape due to the proliferation of devices, often outnumbering human users, and the continual innovation and sophistication exhibited by attackers.

b) What is Penetration Testing :

- A penetration tester, similar to a skilled hacker, conducts a thorough examination of a remote system's vulnerabilities and security weaknesses. This task demands a structured approach and a set of specialized skills. Penetration testing is a multifaceted and highly sensitive process. It comprises six distinct stages within its lifecycle.

- Information Gathering
- Footprinting And Scanning
- Vulnerability Assessment
- Exploitation
- Reporting

c) Engagement :

at the Engagement stage, a professional pentester defines the amount of time for the penetration testing of a Network, Web Application, or the whole organization. Time varies according to,

- Type of engagement (Black Box, Grey Box, etc)
- the Complexity of the applications and services in scope

The number of targets (IP addresses, Domains, etc)

d) Information Gathering :

Information Gathering serves as the initial and crucial phase in the execution of a successful penetration test. It involves the collection of various types of data about the target. This stage essentially represents the starting point of ethical hacking, where penetration testers, whether they are white hat or black hat hackers, strive to accumulate comprehensive information about the target with the intention of utilizing it for hacking purposes.

e) Footprinting And Scanning :

This process entails gathering an extensive amount of information regarding the target system with the objective of identifying potential vulnerabilities and entry points. A large amount of the work that ethical hackers do is profiling an organization, gathering information about the host, network, and people connected to it. This data may encompass details such as IP addresses, Whois records, DNS information, the operating system in use, employee email addresses, phone numbers, and more.

f) Vulnerability Assessment :

A vulnerability assessment is a methodical examination of security weaknesses within a specific target. Assessing the target's susceptibility to known vulnerabilities and the degree of severity associated with them are its goals. and provide recommendations for remediation or mitigation to enhance security. We can carry out a vulnerability assessment in two ways,

- Manually by using data collected in the previous phases
- by utilizing Automated Tools

g) **Exploitation :**

At this stage, it is time to verify if the vulnerabilities really exist. The exploitation phase involves the utilization of vulnerabilities identified in the preceding step. In this phase, a penetration tester verifies and exploits a vulnerability, while also expanding and escalating their privileges within the target system and network. Successfully exploiting a machine not only aids in further exploration of the target network but also enables the discovery of new targets. This process often leads to a repetition of the entire cycle, starting from the information-gathering phase. A penetration test is indeed a Cyclic Process.

h) **Reporting :**

Finally, the penetration test report holds equal importance to the entire testing phase, serving as an official means to deliver and communicate the outcomes of the tests with stakeholders and relevant parties.

- Executives
- IT Staff
- Development team

The report shows and explains the result of your tests and is the actual deliverable of your professional engagement.



Fig.no 1.1: Penetration Testing Life-cycle

II. RELATED WORKS

a) **Existing System**

There are some existing systems in the Information gathering and Enumeration Part of cybersecurity, like Search-that-hash and sooty which are only available in a particular enumeration method.

Limited Reporting: While tools can generate reports, the quality and comprehensiveness of those reports can vary. Human analysis and interpretation are often necessary to provide meaningful results. **Scanning Noise:** Running penetration tests can generate network traffic and logs that could alert security systems or network administrators, potentially causing disruptions. **Time-Consuming:** Penetration testing can be a time-consuming process, especially if done manually, which may not align with tight project schedules. **Skill Required:** Proper use of these tools often requires a high level of expertise. Inexperienced users may misconfigure the tools or misinterpret the results. **Cost:** Some commercial penetration testing tools can be expensive, making them less accessible for small organizations with limited budgets. **Missing Zero- Day Vulnerabilities:** Penetration testing tools often rely on known vulnerabilities and attack patterns. They may not be effective at identifying zero- day vulnerabilities for which no known defense or signature exists.

b) **Proposed System**

After understanding the existing system and the need for developing a new system,

- Satisfies users with enumerated accurate results.
- Saves a lot of Time while Enumeration
- Easily finds Network-based known Vulnerability using NSE
- Enumerates Informative results with High accuracy
- Highly optimized in offensive security operating system
- Preloaded Comprehensive Efficient Script
- Understanding the existing system and recognizing the imperative for developing a new system to address

identified shortcomings.

- Ensuring user satisfaction through the delivery of enumerated results that are precise and accurate, meeting the expectations of users.
- Significantly reducing time spent on enumeration tasks, thereby enhancing efficiency and productivity in cybersecurity operations.
- Streamlining the process of identifying network-based known vulnerabilities using NSE (Nmap Scripting Engine), enabling prompt remediation actions.
- Providing informative results characterized by high accuracy, enhancing the reliability and utility of the system for cybersecurity professionals.
- Optimization for offensive security operations, ensuring that the system is finely tuned to meet the requirements of offensive cybersecurity practices.
- Offering a preloaded repository of comprehensive and efficient scripts, equipping users with a wide range of tools and functionalities for effective cybersecurity enumeration and assessment.

III. LITERATURE SURVEY

In [1] “2021, G Jayasuryapal , P Meher Pranay has proposed A Survey on Network Penetration Testing which focuses on all the System along with information gathering to the post exploitation. Penetration testing operations adhere to a set of guidelines and unity that the customer and pen testing team share. This examination will identify any flaws in the organization, such as open ports and servers, among other things.

In [2] 2019, Pengfei Shi , Futong Qin has proposed The Penetration Testing Framework for Large-Scale Network Based on Network Fingerprint. The fundamentals and methodology of conventional The examination of penetration testing technology for large-scale networks encompasses an analysis of its capabilities and constraints. To address these challenges, a penetration testing framework tailored for extensive networks has been developed. This framework leverages network fingerprinting techniques in conjunction with online search engine technologies to enhance its effectiveness and scope.

In [3] 2017, Rodney R Rohrmann , Vincent J Large-scale port scanning over Tor utilizing parallel Nmap scans to cover a lot of IPv4 range is what Ercolani has suggested. Even if it is efficient

at assuming data-information when scanning, because it takes noticeably longer to run over Tor, it cannot be expanded to the point where it can scan the full IPv4 Address space on many ports. The benefit of a scanning method that shields researchers from targets who might potentially retaliate after being scanned is that it enables them to source their own scans.

In [4] 2015, Prerna Arote , Karam Veer Arya has proposed While it is useful for assuming data-information while scanning, its execution time through Tor makes it impractical to extend to the point of scanning the whole IPv4 Address space on many ports. The Data Link Layer is in charge of framing packets as they are processed and passed down the stack (DLL).

In [5] 2016, Enos LETSOALO, Sunday OJO has proposed Survey of Media Access Control address spoofing attacks detection and prevention techniques in Wireless Networks. The Media Access Control addresses of the access point or other users can be extracted from packets intercepted by an attacker using packet sniffer software. In wireless networks, a client is connected to the access point using its MAC address. To disconnect authorised users from the network and seize control of any already established TCP session, an attacker can fake the Media Access Control IP of the real access point.

In [6] 2017, Lerato Ramahlapane Moila , Mthulisi Velempini has proposed An Evaluation of the Effectiveness of Cognitive Radio Ad hoc Networks Routing Protocols. Real-time data quality of service needs is not guaranteed by the current protocols. Some of the issues preventing the development of routing protocols that are quality of service aware are caused by the dynamic environment of cognitive networks, which is also responsible for the mobility of nodes, the movement of the spectrum, and the nondeterministic availability of the spectrum.

In [7] 2017, Sathish A.P. Kumar , Brian Xu has proposed Vulnerability Assessment for Security in Aviation Cyber-Physical Systems. We have tested for vulnerabilities in data loaders and other aircraft systems in order to meet the standards for wireless network security set out by the aviation industry. Our efforts include increasing aircraft security and safety as well as identifying cyber vulnerabilities in these aviation systems utilising vulnerability assessment and penetration testing tools like Metasploit Pro and BackTrack.

In [8] 2020, Mehr u Nisa , Kashif

Kifayat has proposed Detection of Slow Port Scanning Attacks. In actuality, a scanning assault is a two- part process in which scanning is the first phase

where the vulnerability of communication routes is discovered. The second step is the discovery of the targets, followed by the attack. In order to identify the system that can be abused, available open ports are therefore sought across the network during port scanning.

In [9] 2022, Sabah M. Morsy And Dalia Nashat has proposed D-ARP: An Efficient Scheme to Detect and Prevent ARP Spoofing. An ARP spoof-based MITM attack takes

use of a flaw in the ARP protocol by associating the attacker's MAC address with the IP address of the target legitimate host. ARP spoof based-MITM attacks have a variety of defence strategies, although these strategies are incomplete or have a limited effectiveness. since they alter the basic ARP protocol, performance overhead.

In [10] 2020, Ron Andrews , Dalton A. Hahn has proposed Measuring the Prevalence of the Password Authentication Vulnerability in SSH. We suggest a novel technique for probing an SSH service to determine whether password authentication is permitted as part of our review, without being invasive or damaging to the host. We also show that some of these tools and services can be enhanced in order to assess the prevalence of password authentication in SSH particularly.

In [11] 2017, Debdeep Dey , Archisman Dinda has proposed Warezmaster and Warezclient : An Implementation of FTP Base R2L Attacks. The fact that all commands and data are transmitted using clear text indicates that the File Transfer Protocol (FTP) was not created with strict security measures in mind. Subsequent updates to the protocol have added security features, like upgrading the connection to Transport Layer Security (TLS) in FTPS.

In [12] 2016, Thierry Mbah Mbelli , Barry Dwolatzky has proposed Cyber Security, a Threat to Cyber Banking in South Africa. For South African businesses in particular as well as the nation as a whole, the rise of cybercrime is turning into a severe economic problem. The first section of this essay examines the numerous online hazards related to cyber banking. In addition, it suggests a boundary- and application-based approach to cyber-banking security”.

IV. SYSTEM ARCHITECTURE

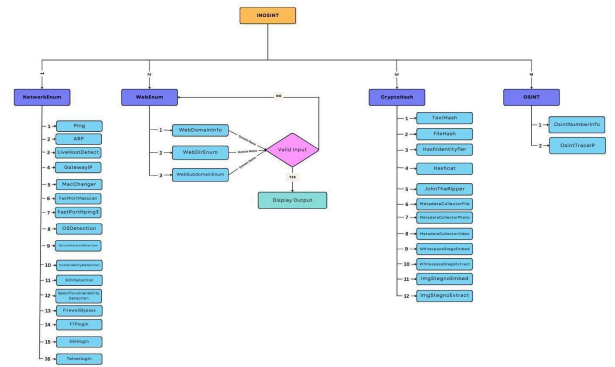


Fig.no 4.1: Data Flow Architecture

V.METHODOLOGY

1. Network enumeration module: plays a crucial role in penetration testing, serving as the initial phase to gather essential information about the target network's infrastructure and identify potential entry points for further exploitation. This process involves various techniques and tools aimed at discovering active hosts, network topology, and potential vulnerabilities. Ping and ARP are fundamental tools utilized for basic network discovery, allowing the tester to determine live hosts and their corresponding IP addresses. Live host detection further aids in identifying active systems within the network, essential for subsequent analysis. GatewayIP detection provides insights into the network's routing architecture, aiding in understanding traffic flow and potential choke points. MacChanger is employed to manipulate MAC addresses, allowing testers to obscure their device's identity and evade detection. Masscan facilitates rapid scanning of large networks, enabling testers to efficiently enumerate open ports and services. Hping3 enables advanced packet manipulation and crafting, useful for network reconnaissance and firewall evasion techniques. Nmap, a versatile tool, offers comprehensive capabilities such as OS detection, port scanning, service version detection, and vulnerability analysis using its NSE (Nmap Scripting Engine). These features allow testers to map out network services, identify potential weaknesses, and assess the overall security posture of the target environment. By leveraging these tools in concert, penetration testers can gain invaluable insights into the target network's infrastructure, aiding in the formulation of effective attack strategies and recommendations for mitigating identified vulnerabilities.

2. Web enumeration module: stands as a pivotal phase, enabling thorough reconnaissance of target systems to identify potential vulnerabilities and attack vectors. Anchored in

the pursuit of comprehensive understanding and mapping of web infrastructure, web enumeration encompasses various techniques and tools tailored to unveil crucial insights. Among

these, the WHOIS lookup tool emerges as a cornerstone, facilitating the retrieval of domain registration information, including registrant details, registration and expiration dates, and domain name servers. This insight aids in constructing a comprehensive profile of the target organization's digital footprint, essential for subsequent stages of the penetration testing process. Complementing WHOIS lookup, directory enumeration emerges as another indispensable facet, bolstered by tools like Gobuster. This tool delves into web servers, systematically scanning for hidden directories and files, thus uncovering potential entry points or sensitive data repositories. Moreover, subdomain enumeration emerges as a pivotal tactic, leveraging tools such as Sublist3r to exhaustively catalog subdomains associated with the target domain. By scouring public domain repositories and brute-forcing subdomains, Sublist3r aids in broadening the scope of reconnaissance efforts, potentially revealing overlooked attack surfaces ripe for exploitation. Collectively, these tools and techniques underpin the web enumeration phase, furnishing penetration testers with a nuanced understanding of the target's web infrastructure, thus empowering informed decision-making and precise exploitation of vulnerabilities.

3. Cryptographic enumeration module: The process of counting and decoding cryptography is essential for identifying weaknesses and evaluating how robust the cryptographic processes used in a system are. This phase involves several crucial steps aimed at understanding, analyzing, and potentially breaking cryptographic constructs. Initially, text and files undergo conversion into hash representations, a process integral to cryptographic operations for data integrity and authentication. Subsequently, the hash type identification is carried out utilizing specialized tools such as hashidentifier, which employs pattern recognition algorithms to determine the cryptographic hash function employed. Additionally, tools like Hashcat and John the Ripper are instrumental in the exhaustive search for plaintexts corresponding to the given hash values, employing techniques like brute-force attacks, dictionary attacks, and rule-based cracking methodologies. Furthermore, metadata analysis forms an essential part of cryptography enumeration, facilitated by tools like ExifTool, which unveils hidden information embedded within digital files, aiding in the identification of potential cryptographic artifacts or security flaws. The exploration of whitespace steganography and image steganography further expands the purview of cryptography enumeration, as these techniques involve concealing data within non-visible areas of digital content, posing challenges to traditional cryptographic analysis. Through a comprehensive approach encompassing hash conversion, hash type identification, hash cracking, metadata analysis, and steganographic exploration, cybersecurity professionals can effectively scrutinize cryptographic implementations, identify weaknesses,

and reinforce defenses to mitigate potential security risks and safeguard sensitive information.

4. Osint enumeration module : Open-Source Intelligence (OSINT) is a methodology for gathering information about potential targets and assessing their security posture. Within this framework, leveraging specialized tools such as PhoneNumber Verifier for validating phone numbers and tracking the public IP address with location emerges as indispensable assets. PhoneNumber Verifier enables penetration testers to validate the authenticity of phone numbers associated with target entities, facilitating the identification of potential communication channels and attack vectors. This tool empowers testers to ascertain the validity and reachability of phone numbers, aiding in the reconnaissance phase of penetration testing. Furthermore, the capability to track public IP addresses with location enhances the OSINT arsenal by providing geolocation data essential for mapping out the digital footprint of target systems and entities. By utilizing this functionality, penetration testers can gain insights into the geographical distribution of network assets and potential vulnerabilities. This information enables the formulation of targeted attack strategies and enhances the overall efficacy of penetration testing initiatives. In essence, the integration of specialized OSINT tools like PhoneNumber Verifier and IP address tracking enhances the reconnaissance capabilities of penetration testers, facilitating the identification of vulnerabilities and the mitigation of security risks within target environments.

VI. RESULTS AND DISCUSSION

Software testing plays a pivotal role in software quality assurance, serving as the final evaluation of specifications, design, and code generation. After the source code has been created, it is imperative to subject the software to testing to identify and rectify as many errors as feasible before it reaches the tester. To maximize error detection, testing should follow a systematic approach, and test cases must be meticulously designed using disciplined techniques.

Menu :

Information about tool :

VII.FUTURE ENHANCEMENT

GUI (Graphical User Interface).Planning to implement vulnerability Scanner based on web attacks.Planning to implement more tool OSINT and Web Enumeration tool for future purpose.Planning to implement this tool in Windows Operating System too. In our upcoming enhancements, we aim to introduce a Graphical User Interface (GUI) for improved user interaction. This GUI will facilitate a more user-friendly experience, enhancing accessibility and usability. Additionally, we plan to integrate a Vulnerability Scanner focusing on web attacks, bolstering security measures. Looking ahead, we intend to develop additional tools such as OSINT (Open-Source Intelligence) and Web Enumeration tools to further enrich our offering. Moreover, recognizing the diverse user base, we're expanding compatibility to include the Windows Operating System, broadening accessibility. These enhancements reflect our commitment to continuous improvement and addressing evolving user needs. With these additions, users can anticipate a more comprehensive and streamlined experience, empowering them to better navigate and address cybersecurity challenges.

VII. REFERENCES

- [1] Plummer: An Ethernet address resolution protocol. RFC 826 (1982).
- [2] <https://nmap.org/book/man-briefoptions.htm>.
- [3] <https://www.linkedin.com/learning/paths/prepare-for-comp-tia-network-plus-n10-007-certification>.
- [4] Gordon Lyon, "Nmap Network Scanning" Sunnyvale, CA: Insecure.com, LLC,2002 p.129
- [5] Huitema, C. (1997), IPv6, The new Internet Protocol. Prentice Hall.
- [6] Unknown Author, "Kali Linux Tutorial: Setting Up ProxyChains + Tor For Anonymity And Security"
<http://www.picateshackz.com/2015/05/kali-linux-tutorial-setting-up-html>
- [7] Nmap cheat sheet see-security hacking defined experts by Nativ.y,2015.
- [8] Lyon Gordon NMAP-the Network Mapper-free security lancer,2017.
- [9] Ethical hacking and penetration guide by Baloch R,2017.
- [10] A guide to penetration testing by Tang A., Network security,2014.

CERTIFICATE :

