IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Streamline Recon CLI Tool

Karthiban R[1], Ajaykumar JS[2], Dharun J[3], Rooban N[4] , Vinayagamoorthy N[5] ,Yuvan krishna P[6]

[1] *Assistant Professor, Department of Computer science and engineering (Cybersecurity) , Sri Shakthi Institude of Engineering and Technology-Coimbatore-62.*

[2,3,4,5,6] *Department of Computer science and engineering (Cybersecurity) , Sri Shakthi Institude of Engineering and Technology-Coimbatore-62.*

Abstract - In the field of cybersecurity, efficient reconnaissance and enumeration processes are vital for identifying potential vulnerabilities in target systems. This paper presents an innovative, automated CLI tool designed to streamline recon and enumeration tasks for cybersecurity professionals and vulnerability assessment teams. The tool integrates essential functionalities such as DNS lookups, reverse DNS, subdomain enumeration, zone transfer testing, web crawling, and sensitive data scanning, including API key and credential detection from popular services like MongoDB, AWS, and Google Cloud. With a user-centric interface and automation-focused design, our tool enhances the speed, accuracy, and depth of information gathering required for effective Vulnerability Assessment and Penetration Testing (VAPT). Extensive testing on real-world systems demonstrated that our solution effectively reduces the manual effort associated with reconnaissance tasks, while providing comprehensive data to support thorough security assessments.

Keywords - Gender, Reconnaissance, Enumeration, Cybersecurity, Vulnerability Assessment, Web Security.

1. INTRODUCTION

In today's cybersecurity landscape, Security is one of the major issues of information systems. The growing connectivity of computers through the internet, the increasing extensibility of systems, and the unbridled growth of the size and complexity of systems have made software security a bigger proble now than in the past. Furthermore, it is a business imperative to adequately protect an organization's information assets by following a comprehensive, and structured approach to provide protection from the risks an organization might face. In an attempt to solve the security problem and comply with the mandated security regulations, security experts have developed various security assurance methods including proof of correctness, layered design, software engineering environments and penetration testing.[3] The five pillars of information assurance are confidentiality, integrity, availability, non-repudiation, and authentication. An attack that violates at least one of these pillars is considered as a cyber-attack. Reconnaissance is the first phase of cyber-attack. In this phase, the weak points of the target are identified.[4]

This paper introduces a command-line interface (CLI) tool designed to automate the key aspects of reconnaissance and enumeration, optimizing the data gathering phase for cybersecurity professionals. Our tool combines various critical features, such as DNS lookups, reverse DNS resolution, zone transfer analysis, subdomain enumeration, and web crawling, along with sensitive data detection techniques. The inclusion of API key and credential scanning enhances the tool's ability to detect potentially exposed sensitive information, a common issue in misconfigured services and public repositories. Our tool is specifically built for usability, offering a straightforward interface and results formatted for easy interpretation, making it suitable for beginners cybersecurity professionals. Initial testing results indicate that the tool significantly reduces the manual overhead associated with reconnaissance tasks, enabling quicker and more reliable data collection. This automation not only benefits penetration testers but also security analysts conducting continuous monitoring and assessment.

The reconnaissance phase in cybersecurity is broadly divided into two types: Passive Reconnaissance and Active Reconnaissance, as shown in **Figure 1.1**.

Passive Reconnaissance: This type of reconnaissance involves gathering information about the target without directly interacting with its systems. The goal is to avoid detection while still collecting valuable data that can aid in understanding the target's infrastructure, vulnerabilities, or security posture. Techniques in passive reconnaissance include analyzing public information, using search engines, monitoring social media, and looking up DNS records. Passive reconnaissance allows attackers or security analysts to obtain insights while staying hidden from the target's monitoring systems, as it does not generate traffic directly toward the target.

Active Reconnaissance: In contrast, active reconnaissance involves directly interacting with the target's systems to gather information. This approach generates network traffic and can often be detected by the target's security mechanisms. Common techniques include port scanning, network mapping, and service enumeration. Active reconnaissance is generally more accurate in revealing open ports, running services, and network configurations. However, it carries the risk of alerting the target to the ongoing investigation due to its intrusive nature.

These two types of reconnaissance are essential in cybersecurity assessments, with passive reconnaissance typically used in initial stages to minimize detection, followed by active reconnaissance to confirm findings and obtain more detailed information.

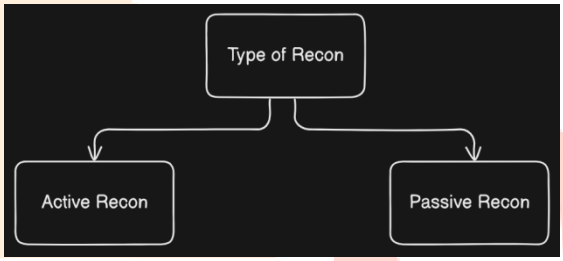

Figure 1.1 Type of recon

2. RELATED WORKS

In recent years, the automation of reconnaissance and enumeration in cybersecurity has gained significant attention, aiming to streamline vulnerability assessments and penetration testing processes. Traditional manual methods for gathering target information, such as subdomain discovery, port scanning, and SSL verification, are not only time-intensive but also prone to oversight, especially with increasingly complex attack surfaces. Several tools, such as Nmap, Amass, and OWASP's Sublist3r, have contributed to automating portions of the reconnaissance process, offering features like network mapping, domain enumeration, and SSL information retrieval. However, these tools often operate in isolation, requiring cybersecurity professionals to combine outputs manually for comprehensive analysis. Recent advancements emphasize integrating multiple functionalities within a single platform to enhance both usability and efficiency. Despite these improvements, there remains a need for an adaptable command-line interface (CLI) tool that consolidates essential features — from subdomain and directory enumeration to port scanning and SSL analysis — into a unified workflow, reducing the manual burden on security analysts and ensuring consistent, accurate data collection.

In [2], it was found that a significant majority, 95.2%, are designed to operate through a command-line interface (CLI), highlighting the widespread preference for CLI among cybersecurity experts. The command-line environment allows users to execute specific, modular commands and customize workflows, enabling rapid and efficient information gathering without the limitations often found in graphical user interfaces (GUIs). Only a small fraction, 4.8%, of the tools do not offer command-line functionality, possibly indicating a focus on ease of use for less technical users or targeting a different type of user experience. Furthermore, the study reveals that 61.9% of these tools are network-focused, meaning they are specifically designed to analyze and gather information related to network infrastructure, such as IP addresses, open ports, network services, and network topology. This network-based focus reflects the essential need in cybersecurity for understanding the target's external-facing network components, which

are often primary points of vulnerability. The high percentage of network-oriented tools underlines the critical role that network reconnaissance plays in modern cybersecurity, allowing for more targeted vulnerability assessments and enhancing the penetration testing process.

In [3], authors have presented a structured overview of Penetration Testing. The paper discusses the advantages and methodologies involved in conducting Penetration Testing. It further demonstrates how to conduct penetration testing using two demo sites. The findings in the paper show that penetration testing is a three-phase methodology consisting of preparation, test, and analysis phase. The test phase includes reconnaissance, scanning and vulnerability exploitation. It can be done manually or using automated tools.

In [4], authors have proposed a tool which helps in fingerprinting an organization. The tool presented is developed using Java which locates and saves organization specific data. The paper discusses the two types of reconnaissance and OSINT. It provides the possibility of network-based passive reconnaissance.

In [6], authors discussed the approach to perform manual penetration testing in web applications and the research paper is suitable to act as a guide for testing OWASP Top 10 vulnerabilities. The paper discusses all the five phases of penetration testing. The objective of the paper is to provide knowledge about all the phases of penetration testing.

In penetration testing, apart from the framework and methodology, other things are no less important, namely the use of work tools that can help testers in certain phases from gathering information, finding vulnerabilities, and analyzing to reporting. There are many penetration testing tools available, especially open source, each tool certainly has advantages and disadvantages. Apart from that, various tools are available for various operating system platforms, starting from Linux (Kali Linux), which is widely used by penetration testers, and Microsoft Windows. Several tools have their own focus areas, including NMAP which can be used to penetrate network activity. Among its functions are port scanning, IP address space mapping, and direct host fingerprint discovery like using different tool use all in one tool to make the process as simple. [8]

3. METHODOLOGY

The methodology of this project revolves around the development of a command-line interface (CLI) tool that automates critical aspects of reconnaissance and enumeration for vulnerability assessment and penetration testing. This tool is designed to simplify and expedite the data gathering phase by integrating multiple modules, each focusing on a specific aspect of information collection. The following steps outline the core functions of the tool and how each module contributes to its overall effectiveness:

3.1 DNS and Reverse DNS Lookups

- **Objective**: To gather network mapping details by resolving domain names to IP addresses and vice versa.
- **Process**: Zone transfers (AXFR queries) are legitimate operations for DNS servers to synchronize data, but if misconfigured, they can allow unauthorized parties to retrieve complete records of DNS zones. The tool sends an AXFR request to each DNS server listed in the NS (Name Server) records.
- **Technical Details**: The tool uses system DNS resolver functions or external DNS API services for faster, reliable lookups. By analyzing response codes and headers, it can identify misconfigurations or open DNS ports, which may suggest vulnerabilities.

3.2 Zone Transfer Analysis

- **Objective**: To attempt a zone transfer from the target's DNS server, potentially exposing network topologies and subdomains.
- **Process**: The tool initiates a DNS lookup for the target domain, obtaining the primary IP address, which serves as an entry point into the network. Following this, reverse DNS resolution queries the IP addresses to ascertain their associated domain names, which can sometimes reveal internal hostnames and non-public assets.
- **Technical Details**: The tool parses NS and SOA records to identify DNS servers before initiating zone transfer requests. If the transfer is successful, it collects data on IP ranges, subdomains, and

configurations, potentially exposing internal systems. The tool flags successful zone transfers as high-severity findings due to the sensitive nature of the data revealed.

3.3 Subdomain Enumeration

- **Objective**: To identify additional attack surfaces by discovering subdomains associated with the target domain.
- **Process**: The tool combines multiple subdomain enumeration techniques, including brute-forcing, certificate transparency logs, DNS enumeration APIs, and search engine scraping to gather a comprehensive list of subdomains. Each discovered subdomain is checked for validity and active status.
- **Technical Details**: The brute-force approach uses a predefined list of common subdomains, querying each to determine if it resolves. Additionally, APIs such as those from Rapid7 or security-focused services may be integrated to expedite the enumeration process. Results from various sources are aggregated and de-duplicated, with active subdomains marked for further investigation, as they may reveal undersecured services or unmonitored endpoints.

3.4 Port Scanning

- **Objective**: To identify open ports on the target IP address, which could expose network services that may be vulnerable or misconfigured, thereby increasing the potential attack surface.
- **Process**: The tool uses a multi-threaded approach to scan a range of ports (default 1-1024) on the target IP address. Each port is checked to determine if it is open and actively listening. The scan is conducted using concurrent threads to improve speed and efficiency, with each thread handling a specific port within the range. Detected open ports are logged and reported, as these ports often serve as entry points for attackers attempting to exploit network services.
- **Technical Details**: Threaded Scanning: The tool leverages Python's concurrent.futures module to perform asynchronous port scanning with a ThreadPoolExecutor, which allows the tool to scan multiple ports simultaneously. This threaded approach reduces scan time, especially on large port ranges or slow networks.
- **Code Execution**: The portscan function initializes a thread pool with a specified number of max_threads. Each thread invokes the scan_port function for a given port within the specified range (start_port to end_port), increasing the tool's efficiency over a single-threaded scan. Interpretation of Results: Open ports typically indicate running services, such as HTTP, SSH, or FTP, which may require additional investigation to check for vulnerabilities or improper configurations. Efficiency Considerations: By using multi-threading and a timeout on each socket connection, the tool optimizes scan time while minimizing network strain on both the target and scanning machine.

3.5 Web Crawling and Internal Link Discovery

- **Objective**: To systematically map accessible resources on the target website and uncover potential entry points, such as login portals, admin panels, or poorly protected directories.
- **Process**: A recursive web crawler is deployed to navigate through internal links, starting from the homepage and following links to discover all reachable paths within the target domain. The crawler collects URLs, page titles, and metadata that can be useful in identifying sensitive information or poorly protected resources.
- **Technical Details**: The tool leverages a headless browser or HTTP client library to fetch HTML content, identifying links, scripts, and forms within the page source. The crawler respects the robots.txt file but can override it if required for testing purposes. Each link is evaluated for response codes, headers, and page content to gauge the potential presence of exploitable endpoints.

3.6 SSL Certificate Information Retrieval

- **Objective**: To obtain SSL certificate details for a given domain, allowing cybersecurity professionals to verify certificate validity and assess expiration dates, issuer information, and other certificate properties. This information is essential for identifying potential vulnerabilities, such as expired or misconfigured certificates.
- **Process**: The tool establishes a secure connection to the domain on port 443, retrieves the SSL certificate details, and displays key certificate information in a readable format. This functionality helps ensure that SSL/TLS security measures are properly configured on the target domain.

- **Technical Details**:
- SSL Context Creation: The tool begins by creating an SSL context using Python's ssl.create_default_context() method, which configures default SSL settings, such as certificate verification and hostname checking, to ensure secure communication. The context is wrapped around a socket to facilitate the secure exchange of certificate information over the TLS protocol.
- Socket Connection and SSL Wrapping: The tool initiates a socket connection to the specified domain on port 443 (the standard port for HTTPS). This step is critical for interacting with the server and retrieving the certificate. The socket is then wrapped in SSL using the context.wrap_socket() method with server_hostname set to the domain name, enabling the secure TLS handshake to complete.
- Certificate Extraction: Once the secure connection is established, the getpeercert() method retrieves the server's SSL certificate, containing details about the certificate issuer, validity period, and subject information. Relevant fields, such as issuer, notBefore, notAfter, and subject, are extracted for analysis, allowing cybersecurity analysts to verify certificate authenticity and assess its expiration status. The tool outputs a structured table containing the SSL certificate details for the specified domain, highlighting crucial information for cybersecurity analysts. If the SSL certificate retrieval fails, a user-friendly error message is displayed, enabling quick troubleshooting.

3.7 Directory Enumeration

- **Objective**: To identify accessible directories on a target URL, which could reveal sensitive files, misconfigurations, or hidden areas that may be vulnerable to unauthorized access.
- **Process**: The tool iterates over a predefined list of directory names and sends HTTP requests to check if each directory exists on the target server. Successfully located directories are stored and displayed for further examination.
- **Technical Details**: The check_directory function is used to send an HTTP GET request for each directory in directories_list. If the response status code is 200 (OK), it indicates that the directory exists, and the URL is saved in found_directories.The tool utilizes Python's concurrent.futures.ThreadPoolExecutor with 20 threads to speed up the enumeration by running multiple directory checks concurrently. This parallelization ensures that the tool can quickly process large lists of directories.

3.8 HTTP Headers Retrieval

- **Objective**: To gather HTTP headers from a target URL, providing insight into server configuration, technologies used, and potential security headers.
- **Process**: The tool sends an HTTP HEAD request to the target URL and extracts the response headers. These headers may contain valuable details about server software, caching behavior, security policies, and more.
- **Technical Details**: An HTTP HEAD request is sent to the target URL, which retrieves only the headers without downloading the entire page content, saving bandwidth and improving speed. Header Extraction and Display: The tool iterates through each header in the response and prints them in a structured format, listing each header key-value pair for easy analysis. Exception Handling: If the URL is inaccessible or another error occurs, an error message is shown to the user, allowing them to troubleshoot or try again later.

3.9 Sensitive Data Detection and API Key Scanning

- **Objective**: To identify exposed sensitive information, such as API keys, access tokens, and credentials, that could enable unauthorized access or further attacks.
- **Process**: The tool scans both HTTP responses and page content collected during crawling for known patterns associated with sensitive data (e.g., AWS access keys, MongoDB credentials, Google API tokens). It applies regular expressions and pre-defined patterns to detect potential secrets.
- **Technical Details**: Patterns are based on known secret formats (e.g., base64 encoding, specific string lengths, and prefixes) and are continually updated to include new services. This module also integrates with GitHub's secret scanning API to check public repositories for the target organization or domain, alerting on accidental exposure of credentials. Findings are categorized by risk, with high-priority flags for keys associated with cloud services or critical infrastructure.

3.6 Comprehensive Report Generation

- **Objective**: To present the data collected in a structured, interpretable format suitable for further analysis or as documentation for vulnerability assessment.
- **Process**: The tool generates a detailed report, organizing results by module (e.g., DNS lookup results, subdomains, zone transfer findings, sensitive data detected). Each section includes metadata, timestamps, and severity ratings, along with actionable insights, such as recommended follow-up actions or mitigation steps.
- **Technical Details**: Reports can be generated in multiple formats, such as JSON, HTML, or PDF, for compatibility with various documentation requirements. The tool structures output hierarchically and includes options for customizable templates, ensuring that data is easy to parse and interpret for users of all experience levels.

3.7 Usages

The Recon CLI Tool is a command-line interface designed to streamline the reconnaissance phase in cybersecurity assessments by automating essential information-gathering tasks. With a range of configurable options, the tool allows users to perform various types of reconnaissance, such as port scanning, subdomain enumeration, reverse DNS lookup, and WHOIS information retrieval. It also includes functionality for decoding encoded strings in formats like Base64, ROT13, and Base32, and retrieves SSL certificate and HTTP header information. This flexible and user-friendly CLI tool consolidates critical reconnaissance processes, enhancing efficiency and accuracy for penetration testers and cybersecurity analysts.

```
usage: master.py [-h] [-auto] [-d] [-p] [-sd] [-dir] [-ip] [-shodan] [-whois]
                 [-ssl] [-hdr] [-geo] [-bs64] [-rot13] [-bs32]

Recon CLI Tool : A tool or Script that makes gathering information about
targets using Basic Recon Process.

options:
  -h, --help            show this help message and exit
  -auto, --autoprocess  Under Construction
  -d , --domain         Target Domain
  -p , --portscan       Target IP for Port Scan
  -sd , --subdomain     Get the Subdomain of the Target System
  -dir , --directory    Get the Directory brute-forcing function
  -ip , --ip            IP address for reverse DNS lookup
  -shodan , --shodan_api_key
                        Under Construction : API Key for Shodan
  -whois , --whois      Perform WHOIS lookup
  -ssl , --ssl          Retrieve SSL Certificate Info
  -hdr , --headers      Retrieve HTTP headers
  -geo , --geoip        Perform GeoIP lookup
  -bs64 , --base64      Decode the base64 String
  -rot13 , --rot_13     Decode the ROT13 String
  -bs32 , --base32      Decode the base322 String
```

Figure 3.1 Usage of the Recon tool

4. CONCLUSION:

In this paper, we introduced an automated command-line tool for reconnaissance and enumeration, streamlining the essential steps of vulnerability assessment. By incorporating multiple features—such as DNS lookups, subdomain enumeration, port scanning, SSL certificate retrieval, WHOIS lookup, HTTP header analysis, and directory enumeration—the tool effectively aids cybersecurity professionals in gathering critical information about target systems. This automation reduces manual effort, speeds up data collection, and enhances overall efficiency, making it a valuable asset for penetration testers and security analysts alike.

For future work, we aim to expand the tool's capabilities by adding advanced cryptographic decryption techniques. This enhancement will allow analysts to handle encrypted data they may encounter during reconnaissance and analyze it for potential vulnerabilities or misconfigurations. Additional features, such as improved web crawling, deeper API key scanning, and network-based data collection, will also be incorporated to provide even more comprehensive insights. With these planned updates, the tool will continue to evolve as a robust, versatile solution for the cybersecurity community, addressing both current and emerging security challenges.

REFERENCES

[1] Redrowthu Ph.D, Vijaya & Ahmed, Iftequar & Reddy, Sriveda & Akshay, S & Reddy, Vrushik & Reddy, Sanjana. (2022). Automation of Recon Process for Ethical Hackers. 1-6. Odun-Ayo, Isaac. (2021).

[2] A Review of Common Tools and Techniques for Reconnaissance Attacks.. Proceedings of the. 141-157. 10.22624/AIMS/iSTEAMS-2021/V28P11.

[3] A.G. Bacudio, X. Yuan, B.T.B.Chu and M. Jones. "An Overview of Penetration Testing". Int. Journal of Network Security & Its Applications Vol.3 (no. 6) (2021, November)

[4] H. P. Sanghvi and M. S. Dahiya, "Cyber Reconnaissance: An Alarm before Cyber Attack," International Journal of Computer Applications, pp. 36-38, 2013.

[5] Sanya Bindlish, Mehak Khurana, "RECON Tool: An Automation of Reconnaissance & Scanning" IJCRT Vol 9 issue 8 Aug 2021.

[6] Nagendran K, Adithyan A, Chethana R, Camillus P, Bala Sri Varshini K B. "Web Application Penetration Testing". International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8 Issue-10 (2019, August 10)

[7] Kathrine, G. Jaspher & Joseph, Ronnie & Veemaraj, Ebenezer. (2020). COMPARATIVE ANALYSIS OF SUBDOMAIN ENUMERATION TOOLS AND STATIC CODE ANALYSIS. 15. 158-173.

[8] Adam, Helmi & Widyawan, Widyawan & Putra, Guntur. (2023). A Review of Penetration Testing Frameworks, Tools, and Application Areas. 319-324. 10.1109/ICITISEE58992.2023.10404397.

[9] Bell, K., Hong, J., McKeown, N. and Voss, C., 2021. The Recon Approach: A new direction for machine learning in criminal law. Berkeley Tech. LJ, 36, p.821.

[10] Mazurczyk, Wojciech & Caviglione, Luca. (2021). Cyber Reconnaissance Techniques. Communications of the ACM. 64. 10.1145/3418293.

[11] Railkar, Dipali. (2022). A Study on Vulnerability Scanning Tools for Network Security. International Journal of Scientific Research in Computer Science Engineering and Information Technology. 8. 340. 10.32628/CSEITCN228641.

[12] Wang, L., Abbas, R., Almansour, F., Gaba, G., Alroobaea, R. and Masud, M. (2021) An empirical study on vulnerability assessment and penetration detection for highly sensitive networks. Journal of Intelligent Systems, Vol. 30 (Issue 1), pp. 592-603. https://doi.org/10.1515/jisys-2020-0145

[13] Alhamed, M.; Rahman, M.M.H. A Systematic Literature Review on Penetration Testing in Networks: Future Research Directions. Appl. Sci. 2023, 13, 6986. https://doi.org/10.3390/app13126986.