ROAR: Red Team Orchestration and Automation Framework for Network and Web Security Assessment

Nameetha V
Guided By:Dr M.Usha Devi
Department of Computer Science, Rathinam College of Arts and Science (Autonomous), Coimbatore, Tamil Nadu, India.

***Abstract*—**Reconnaissance is a fundamental and critical phase in cybersecurity assessment and penetration testing, where comprehensive information about a target system is gathered prior to vulnerability analysis. Traditional reconnaissance approaches rely on multiple independent tools executed manually, generating scattered and unstructured outputs that are time-consuming and difficult to analyze. This paper presents ROAR (Red Team Orchestration and Automation Framework), a Python-based automated reconnaissance system designed for systematic network and web security assessment. The framework integrates multiple information-gathering tools into a unified platform and executes them in a structured, sequential manner. It automates tasks including port scanning, service detection, DNS enumeration, subdomain discovery, directory scanning, and web technology identification. The collected data is aggregated and presented in a structured format, enabling efficient analysis of the target system's attack surface and security posture. Experimental evaluation demonstrates that the ROAR framework significantly reduces manual effort, improves data organization, and enhances the overall effectiveness of the reconnaissance phase in cybersecurity assessments.

Keywords—Reconnaissance, Penetration Testing, Automation Framework, Network Scanning, Web Analysis, DNS Enumeration, Cybersecurity, Python, Kali Linux, Information Gathering.

I. INTRODUCTION

Cybersecurity plays an increasingly critical role in protecting systems, networks, and sensitive data from the growing and evolving landscape of digital threats. Organizations and individuals depend on cybersecurity assessments to identify vulnerabilities and strengthen their overall security posture. One of the most essential phases in this process is reconnaissance, which provides the foundational information required for all subsequent stages of security analysis and penetration testing.

Reconnaissance is the first phase of Vulnerability Assessment and Penetration Testing (VAPT), where information about a target system, network, or web application is systematically collected. This phase involves gathering publicly available data such as IP addresses, domain registration details, DNS records, open ports, running services, active subdomains, and web technologies in use. The primary objective is to map the attack surface of the target and identify potential entry points before any exploitation activity is performed.

Reconnaissance is broadly classified into two categories: passive reconnaissance, which involves collecting information from publicly available sources without directly interacting with the target system, and active reconnaissance, which includes direct interaction techniques such as port scanning and service fingerprinting.

Traditional reconnaissance methods depend on multiple standalone tools, each designed for a specific purpose. This fragmented approach makes the overall process time-consuming, complex, and difficult to manage. Since each tool generates results in different formats, the outputs are often scattered and unstructured, making correlation and analysis challenging. Important security insights may be missed, and the process is highly susceptible to human error and inconsistency.

To overcome these limitations, the ROAR (Red Team Orchestration and Automation Framework) is proposed. The framework integrates multiple network and web reconnaissance tools into a single Python-based platform and automates the information-gathering process. By providing structured and centralized output while reducing manual effort, the ROAR framework improves the efficiency and effectiveness of the reconnaissance phase in modern cybersecurity assessments and contributes to more accurate security analysis.

II. RELATED WORKS

Several approaches have been proposed for automating information gathering and reconnaissance in cybersecurity assessments. Early efforts focused on the use of standalone tools such as Nmap for port scanning and Whois for domain information retrieval. While effective individually, these tools lack integration and require significant manual coordination, making large-scale reconnaissance difficult to manage.

Subsequent research introduced scripted automation using shell scripts and Python, allowing sequential execution of multiple tools. However, these solutions were often limited in scope, lacked modularity, and produced unstructured outputs that were difficult to interpret without post-processing. Studies have highlighted that unstructured reconnaissance output is one of the main factors contributing to missed vulnerabilities during security assessments.

More recent approaches have explored framework-based solutions that integrate multiple tools under a unified interface. Tools such as Recon-ng and theHarvester provide modular reconnaissance capabilities, but they focus primarily on passive open-source intelligence (OSINT) gathering and do not cover active network scanning comprehensively. Research in network

security assessment further emphasizes the importance of combining DNS enumeration, subdomain discovery, and web technology fingerprinting to obtain a complete picture of the attack surface.

Despite these advancements, existing solutions still face challenges such as lack of structured output, limited scalability, and the requirement for extensive user expertise. The ROAR framework addresses these gaps by combining active and web reconnaissance tools in a modular, automated pipeline with centralized, structured output designed for practical cybersecurity assessment scenarios.

III. SYSTEM DESIGN

The background research indicates a clear need for a unified and automated framework for performing network and web reconnaissance in cybersecurity assessments. This study presents the ROAR framework, an optimized approach for systematically gathering target information by executing multiple reconnaissance tools in a structured pipeline. The primary contributions of this study are outlined as follows:

- Implementing a modular reconnaissance pipeline that integrates multiple tools into a single platform, eliminating the need for manual sequential execution.
- Providing automated execution of network reconnaissance tasks including port scanning, service detection, DNS enumeration, and WHOIS information retrieval.
- Incorporating web reconnaissance capabilities including subdomain discovery, directory scanning, and web technology fingerprinting.
- Delivering structured, centralized output that organizes results from all tools into clearly categorized, readable files for efficient analysis.

A. Input Module

The Input Module serves as the entry point of the ROAR framework. It accepts the target domain name or IP address from the user through the command-line interface. The module performs input validation to ensure the provided data is correctly formatted and usable before passing it to the subsequent reconnaissance modules. Invalid or incorrectly formatted inputs are flagged and the user is prompted to re-enter the correct information, ensuring reliable downstream execution.

B. Network Reconnaissance Module

The Network Reconnaissance Module focuses on gathering network-level information about the target system. It integrates and automates the execution of tools including Nmap for port scanning and service detection, WHOIS for domain registration and registrar information retrieval, and DNSenum for DNS record enumeration. The module collects information such as open ports, running services, DNS A records, MX records, and domain registration details. This data provides critical insight into the network exposure and infrastructure of the target system.

C. Web Reconnaissance Module

The Web Reconnaissance Module is responsible for collecting web application-level information about the target. It automates subdomain enumeration using Sublist3r, directory and file scanning using Dirb or Dirbuster, and web technology fingerprinting using WhatWeb and Wappalyzer. This module helps identify hidden directories, active subdomains, the underlying server technology, programming languages in use, and content management systems (CMS) deployed on the target, providing a comprehensive understanding of the web application attack surface.

D. Automation and Control Module

The Automation and Control Module manages the execution flow of the entire framework. It coordinates the workflow by triggering the appropriate reconnaissance modules based on the user-selected mode (network, web, or combined) and ensures proper sequencing and communication between modules. This module is central to reducing manual effort, eliminating redundancy, and maintaining consistent execution across all reconnaissance tasks.

E. Technique Selection

To identify the most suitable approach for automated reconnaissance, various network and web analysis techniques were studied and selected based on their effectiveness in gathering target information. The selected techniques include:

- Port Scanning – Identifying open ports and active services on the target host.
- DNS Analysis – Examining domain-related information including DNS record consistency and domain registration details.
- IP Address Verification – Analyzing the IP address associated with the target for blacklist status and consistency.
- Subdomain Enumeration – Discovering active subdomains to extend the scope of the attack surface.
- Directory Scanning – Identifying hidden or unlisted directories and files within the web application.
- Web Technology Fingerprinting – Detecting server software, programming languages, and CMS platforms.

E.1 DNS Analysis

DNS (Domain Name System) Analysis examines domain-related information of the target system. It identifies suspicious domains by analyzing factors such as domain age, registration details, and DNS record consistency. Phishing or malicious websites often use newly registered domains or frequently alter their DNS records to avoid detection. DNSenum is used in the

ROAR framework to retrieve and analyze DNS A, MX, NS, and TXT records, providing valuable insights into the infrastructure of the target system.

E.2 Port Scanning and Service Detection

Port scanning is performed using Nmap to identify open TCP and UDP ports on the target host, along with the services running on each open port. Service version detection is also performed to identify software versions that may be associated with known vulnerabilities. This technique helps security analysts understand the network exposure of the target and identify potential attack vectors based on exposed services running on the system.

E.3 Web Technology Fingerprinting

Web technology fingerprinting uses WhatWeb and Wappalyzer to identify the server software, web framework, programming language, CMS, and third-party libraries in use on the target web application. This information is critical for understanding the technology stack of the target and identifying software versions that may be subject to known vulnerabilities. The results complement network-level reconnaissance by providing comprehensive application-layer intelligence.

E.4 Output and Data Management

The Output and Data Management component handles the collection, processing, and storage of all results generated by the reconnaissance modules. Python's subprocess module is used to invoke external tools and capture their outputs in real time. The results are processed and organized into clearly labeled output files categorized by data type, such as network information, DNS records, subdomain lists, directory findings, and web technology details. This structured storage ensures that results are easy to access, review, and use for further security analysis.

IV. OBJECTIVES AND SCOPE

The primary objective of this research is to develop an automated and modular reconnaissance framework that integrates multiple network and web information-gathering tools into a single unified platform. The ROAR framework aims to reduce manual effort, eliminate redundancy, improve output organization, and enhance the overall efficiency of the reconnaissance phase in cybersecurity assessments. The system targets security professionals, students, and researchers performing authorized penetration testing and vulnerability assessments.

The scope of this research includes the design, implementation, and evaluation of the ROAR framework covering active network reconnaissance (port scanning, DNS enumeration, WHOIS), web reconnaissance (subdomain discovery, directory scanning, technology fingerprinting), input validation, automated tool execution, and structured output generation. The framework operates in a Linux-based command-line environment and is implemented entirely using Python 3.x along with open-source reconnaissance tools available on Kali Linux.

V. LITERATURE REVIEW

Reconnaissance automation has received significant attention in the cybersecurity research community due to the growing complexity of modern penetration testing workflows. The rise of sophisticated targets with multiple subdomains, diverse technology stacks, and dynamic network configurations has created a strong need for more efficient and intelligent information-gathering systems. Traditional approaches that rely on manual execution of standalone tools are no longer practical for large-scale assessments.

A review of recent literature highlights the increasing adoption of Python-based automation frameworks for cybersecurity operations. Studies have demonstrated that integrating multiple reconnaissance tools under a unified control layer significantly reduces assessment time and improves data consistency. Techniques such as DNS enumeration, port scanning, and subdomain discovery have been widely validated as essential components of effective reconnaissance pipelines in both academic research and industry practice.

Network-based reconnaissance approaches, which analyze communication patterns, service exposure, and domain infrastructure, have proven particularly effective in identifying the attack surface of target systems. Compared to purely passive OSINT approaches, active reconnaissance techniques provide more precise and actionable intelligence, especially when evaluating the security posture of network-exposed services and web applications. The ROAR framework builds upon these findings by combining active network and web reconnaissance in a modular, automated, and structured pipeline that is accessible to both beginners and experienced professionals.

VI. SYSTEM ARCHITECTURE

The system architecture of the ROAR framework illustrates the complete workflow from user input to structured output generation. The process begins with the user providing a target domain or IP address through the command-line interface. The Input Module validates the input before passing it to the Automation and Control Module, which coordinates the sequential execution of the Network and Web Reconnaissance Modules.

The Network Reconnaissance Module invokes Nmap, WHOIS, and DNSenum sequentially, capturing their outputs through Python's subprocess library. The Web Reconnaissance Module follows, executing Sublist3r, Dirb, and WhatWeb in sequence. All captured results are passed to the Output and Data Management component, which processes, categorizes, and stores the

data in structured output files. A final summary is generated and displayed to the user upon completion, providing a consolidated view of all reconnaissance findings.

Fig. 1. System Architecture of ROAR Framework

VII. RESULTS

The ROAR framework was evaluated by executing reconnaissance against controlled test targets in an authorized laboratory environment. The system was assessed across two primary modes — Network Reconnaissance and Web Reconnaissance — to evaluate functionality, output accuracy, and execution efficiency. Performance was measured based on tool execution success, completeness of gathered information, accuracy of identified findings, and quality of structured output generated by the framework.

TABLE I. Reconnaissance Results Summary of ROAR Framework

Reconnaissance Mode	Tools Executed	Observation
Network Reconnaissance	Nmap, WHOIS, DNSenum	Open ports, running services, DNS A/MX/NS records, and domain registration details retrieved successfully.
Web Reconnaissance	Sublist3r, Dirb, WhatWeb	Active subdomains, hidden directories, server software, and CMS technology identified accurately.
Combined Mode	All modules	Comprehensive attack surface mapping with structured, categorized output generated successfully with minimal manual effort.

The system demonstrated stable performance across all tested modes without significant resource overhead. During Network Reconnaissance, the framework successfully identified open ports including SSH (22/tcp), HTTP (80/tcp), and HTTPS (443/tcp), along with DNS records and WHOIS domain registration details. During Web Reconnaissance, active subdomains such as admin.example.com and mail.example.com were discovered, hidden directories including /login, /admin, and /uploads were identified, and the target was fingerprinted as running Apache server with PHP and WordPress CMS.

The automation provided by the framework reduced total reconnaissance time significantly compared to manual sequential tool execution. The structured, categorized output improved data readability and enabled security analysts to quickly understand the attack surface of the target system. No significant false positives were observed during testing in the controlled environment, confirming the reliability of the automated tool execution pipeline.

VIII. CONCLUSION

The ROAR (Red Team Orchestration and Automation Framework) presented in this paper provides an effective and practical solution for automating the reconnaissance phase of cybersecurity assessment. By integrating multiple network and web reconnaissance tools into a unified, modular Python-based platform, the framework successfully eliminates the inefficiencies of manual, standalone tool execution and delivers structured, centralized output that facilitates efficient security analysis.

The results demonstrate that the ROAR framework accurately identifies open ports, running services, DNS records, active subdomains, hidden directories, and web technologies in authorized test environments. The modular architecture ensures scalability and ease of maintenance, allowing additional tools and capabilities to be integrated without disrupting existing functionality. By reducing manual effort, improving data organization, and enhancing output consistency, the framework significantly improves the effectiveness and efficiency of the reconnaissance phase in cybersecurity assessments.

Overall, the ROAR framework provides a reliable, lightweight, and accessible solution suitable for students, security researchers, and professionals conducting authorized cybersecurity assessments. It serves as a strong foundation for future research and development in the field of automated security evaluation. Future enhancements may include a graphical user interface, machine learning-based threat classification, real-time monitoring capabilities, and integration with vulnerability assessment tools for end-to-end security pipeline automation.

ACKNOWLEDGMENT

This article is the outcome of the research work carried out in the Department of Computer Science, Rathinam College of Arts and Science (Autonomous), Coimbatore. The authors would like to express sincere gratitude to the Department for providing the necessary support, guidance, and resources required to successfully complete this work. Special thanks are extended to Dr M.Usha Devi, Assistant Professor,Department of Computer Science, for her valuable suggestions and continuous encouragement. We also thank all faculty members and mentors for their support throughout the research and development of this project.

REFERENCES

[1] NIST, "Framework for Improving Critical Infrastructure Cybersecurity," National Institute of Standards and Technology, 2018.

[2] Center for Internet Security (CIS), "CIS Critical Security Controls," Version 8, 2021.

[3] OWASP Foundation, "OWASP Top 10: The Ten Most Critical Web Application Security Risks," 2021.

[4] G. Lyon, "Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning," Insecure.com LLC, 2009.

[5] J. Weidman, Penetration Testing: A Hands-On Introduction to Hacking, No Starch Press, 2014.

[6] W. Stallings, Network Security Essentials: Applications and Standards, 6th ed., Pearson, 2018.

[7] M. Bishop, Computer Security: Art and Science, Addison-Wesley, 2003.

[8] D. Kim and M. G. Solomon, Fundamentals of Information Systems Security, 3rd ed., Jones & Bartlett Learning, 2018.

[9] K. Beaver, Hacking for Dummies, 6th ed., Wiley, 2018.

[10] T. Wilhelm, Professional Penetration Testing, 2nd ed., Syngress, 2013.