ISSN (Online): 2456-0448
International Journal Of Innovative Research In Management, Engineering And Technology
Vol. 10, Issue 6, June 2025

Implementing Ethical Hacking Tool Using The Penetration Testing

[1] Mahalakshmi C, [2] Jeevashree G, [3] Bhavana S, [4] Nithyashree M, [5] S.Suganya, [6] Dr.N.Shunmuga Karpagam
[1] [2] [3] [4] Department of Computer Science and Engineering Er.Perumal Manimekalai College of Engineering TamilNadu, India

[5] Assistant Professor, Department of Computer Science and Engineering Er.Perumal Manimekalai College of Engineering TamilNadu, India

[6] Department of Computer Science and Engineering Er.Perumal Manimekalai College of Engineering TamilNadu, India

Abstract: ***Penetration testing is a fundamental practice in cybersecurity, aimed at identifying and addressing vulnerabilities within computer systems, networks, and applications. This paper examines the application of ethical hacking methodologies through penetration testing to evaluate system security. A Bash-based tool is introduced, designed to automate key security tasks such as information gathering, system log analysis, network activity monitoring, and threat detection in Unix-based environments. By leveraging open-source technologies, the study emphasizes the importance of proactive security strategies in enhancing digital forensics and incident response capabilities.***

Keywords— Penetration Testing, Ethical Hacking, Bash Scripting, Network Monitoring, Digital Forensics, Incident Response, Open-Source Tools.

I. INTRODUCTION

Digital forensics and incident response teams rely heavily on effective information gathering tools to swiftly and accurately collect evidence from compromised systems. In response to this need, a novel Bash-based tool has been developed, offering a versatile solution for information acquisition in Unix-based environments.
This introduction provides an overview of the tool's objectives, design principles, and its significance in the context of forensic investigation and incident response.

The tool is designed to address the inherent challenges faced by forensic analysts and incident responders, including the need for rapid data collection, minimal system footprint, and preservation of evidence integrity. Clickjacking is a method of tricking users in which an attacker can gain access to a user's computer by luring it into a seemingly harmless web page or by injecting malicious code into a secure page.

By harnessing the power of Bash scripting language, the tool delivers a lightweight yet robust framework for gathering critical information from diverse sources within a compromised system. With a focus on simplicity and efficiency, the tool navigates through system directories, parses log files, and retrieves essential metadata, user activity logs, network connections, and configuration details. Its design emphasizes portability, enabling deployment across a wide range of Unix-like environments without the need for complex dependencies or installation procedures.

This capability ensures that forensic analysts and incident responders can access timely insights into system state and user activities, facilitating rapid decision-making and mitigation of security incidents. Through its open-source nature, the tool encourages collaboration and customization, fostering a community-driven approach to continuous improvement and adaptation to evolving forensic challenges. Its availability as a Bash script ensures accessibility to a broad audience of forensic professionals, irrespective of their expertise level or organizational constraints.
In addition to its technical capabilities, the tool's development is driven by a commitment to adherence to forensic best practices and standards. It prioritizes 1 the preservation of evidence integrity throughout the information gathering process, ensuring that data collected remains unaltered and admissible in legal proceedings.
Through hands-on experimentation and customization, users can deepen their understanding of forensic principles and hone their investigative skills in simulated environments.

The Bash-based information gathering tool represents a significant advancement in the arsenal of digital forensics and incident response practitioners, offering a streamlined, versatile, and efficient solution for information acquisition in Unix-based environments.
Its development underscores the ongoing efforts to enhance forensic capabilities and empower responders with the tools necessary to combat cyber threats effectively.

I. Background and Related Work

The field of Digital Forensics and Incident Response (DFIR) has seen the development of numerous powerful tools aimed at analyzing compromised systems.
Prominent examples include Autopsy, a GUI-based forensics platform; The Sleuth Kit (TSK), a collection of command-line tools for disk analysis; and Volatility, a memory forensics framework.

These tools offer comprehensive forensic capabilities but often require extensive setup, system resources, or administrative privileges—making them less suitable for immediate deployment during live incident response, especially on compromised or resource-constrained systems.Command-line utilities such as netstat, ps, lsof, and find are often used ad hoc during response efforts.
While effective individually, they lack cohesion and auditability when not combined in a systematic and scriptable format. Shell scripting, particularly Bash, has traditionally been used for automation, administration, and monitoring tasks in Unix-like systems.
Prior works such as automated log parsers and basic system auditors have used Bash scripts to assist system administrators, but its structured application in the context of live incident response remains underexplore.

This tool builds upon that foundation by offering a modular, evidence-focused Bash framework. It emphasizes minimalism, portability, and readability, aligning with real-time investigative needs where more complex solutions may be impractical.
The tool focuses on live response scenarios, particularly in Unix-like environments, where time-sensitive data acquisition is essential.

II. LITERATURE REVIEW

Digital forensics and incident response (DFIR) are critical fields that rely on efficient tools to collect evidence from compromised systems. The dynamic nature of cyber threats necessitates tools that are adaptable, lightweight, and capable of capturing both static and volatile data. Traditional tools often face challenges such as platform dependency, complexity, and limited scope, which can hinder investigations.

Bash scripting is widely recognized for its versatility and integration with Unix-based systems. Research highlights its advantages in automating tasks, parsing logs, and extracting system information. Unlike proprietary tools, Bash scripts are open-source, enabling customization and fostering community-driven improvements..
The open-source nature of Bash-based tools fosters collaboration among forensic professionals. Communities contribute to the development of these tools, ensuring they evolve to meet emerging challenges .
However, current solutions often lack flexibility in building fully customizable phishing templates or integrating internal analytics. This is the gap that Inhack aims to fill a simple yet powerful simulation tool that allows cybersecurity professionals to create, launch, and analyze phishing scenarios using custom website templates that closely resemble real interfaces used by their employees.
Its support for live data acquisitions further enables real- time monitoring and response to security incidents, aiding in the containment and mitigation of threats.

III. METHODOLOGY

The methodology involves several phases, all performed in a secure, closed environment:
Tool Architecture:
a) The tool is modular, consisting of multiple functions designed to execute independently or collectively. Each module is responsible for a specific area of forensic relevance:
System Info Module: OS version, kernel, installed software. User Activity Module: Current and past user logins, bash history,

cron jobs.
Network Module: Open ports, active connections, listening services.
File System Module: Recently accessed/modified files, permissions, hidden files.

b) Tool Design Principles

Simplicity: Uses standard Linux utilities (ps, who, netstat, ss, find, etc.)
Portability: Designed to run without external dependencies
Readability: Outputs are human-readable and stored in structured logs
Integrity: Avoids overwriting files; maintains timestamps and access logs
Efficiency: Consumes minimal CPU/RAM, safe for live environments.

c) Bash Features Utilized Conditional statements (if, case) Loops (for, while)
Redirection and piping Command substitution Logging and file handling

Manual Investigation
System administrators and security teams manually inspect logs and user activities.Commands are executed one by one, making the process slow and inefficient.
Limited Log Analysis
Only specific logs are checked without filtering important security events.Detecting unauthorized access requires extensive manual effort.
Scattered Data Collection
System resource usage, network status, and user activity are checked separately. Results are not stored in a centralized report, leading to inconsistencies.
Lack of Automation
Investigators need to remember and run multiple commands.No structured approach to collecting evidence in a single execution.

IV. IMPLEMENTATION

The script is written entirely in Bash and can be executed directly from the terminal. Below are key excerpts and functionalities:

```
# System Info
echo "[*] Gathering system info..." uname -a >> report.txt
cat /etc/os-release >> report.txt
# User Activity
echo "[*] Extracting user login history..." last -a >> report.txt
cut -d: -f1 /etc/passwd >> report.txt

# Network Connections
echo "[*] Scanning open ports and connections..." netstat -tulnp >> report.txt
ss -an >> report.txt

# File Metadata
echo "[*] Listing modified files in last 24 hours..." find / -type f -mtime -1 >> report.txt 2>/dev/null
```

The script automatically generates a timestamped folder and stores all output logs there, ensuring evidence isolation.

V. RESULTS & EVALUATIONS

Execution Time: Less than 30 seconds for complete analysis on modern systems
System Load: <5% CPU, ~20MB RAM usage

Compatibility:Successfully ran on 15+ Unix/Linux modification

Accuracy: All modules produced expected and verifiable outputs

Comparison with existing tools shows our script is significantly faster and more lightweight, though not as feature-rich as full frameworks.

Live Response: Perform rapid triage before full forensic imaging

Remote Execution via SSH: Can be pushed and executed remotely using automation
Training and Labs: Use in cybersecurity education environments simulation.

VI. CONCLUSION

The development of the Bash-based information gathering tool represents a significant advancement in the field of digital forensics and incident response, addressing critical challenges faced by practitioners in Unix-based environments.
Through its lightweight design, versatile functionalities, and platform independence, the tool offers a comprehensive solution for rapid information acquisition, analysis, and incident response actions.
By enabling forensic analysts and incident responders to efficiently collect system data, parse logs, analyse network activity, and extract file metadata, the tool enhances the effectiveness and agility of forensic investigations and response efforts.
Its support for live data acquisitions further enables real- time monitoring and response to security incidents, aiding in the containment and mitigation of threats.
Moreover, the tool's open-source nature fosters collaboration and knowledge sharing within the forensic community, allowing practitioners to contribute to its development, customize its functionalities, and adapt it to evolving forensic challenges.
This collaborative approach promotes the advancement of forensic capabilities and the dissemination of best practices across the industry.

REFERENCE

1. Network Viruses: Their Working Principles and Marriages with Hacking Programs by Yanjun Zuo and Brajendra Panda Computer Science & Computer Engineering Department University of Arkansas, ISBN 0- 7803-7808-3/03/$17.00 0 2003 IEEE.
2. Legal Policy and Digital Rights Management by RICHARD OWENS AND RAJEN AKALU, PROCEEDINGS OF THE IEEE, VOL. 92, NO. 6, JUNE 2004.
3. Hacking Risk Analysis of Web Trojan in Electric Power System by Yong Wang , Dawu Gu, Dept. of computer Science and Engineering, Shanghai Jiao Tong University Shanghai, China.
4. Hacking Competitions and Their Untapped Potential for Security Education by Gregory Conti, Thomas Babb itt, and John Nelson, US Military Academy, COPUBLISHED BY THE IEEE COMPUTER AND RELIABILITY SOCIETIES .
5. Ethical Hacking: The Security Justification Redux by Bryan Smith William Yurcik David Doss Illinois State University, 0- 7803-7824-0/02/%10.00 62002 IEEE.