IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 9, April 2025
Impact Factor: 7.67

Metasploit for Exploit Automation and Threat Detection on Linux

Prof. Subhkirti Bodkhe, Prof. Mrunali Jadhav, Sujal Warkar
Tulsiramji Gaikwad Patil College of Engineering and Technology, Nagpur, India
warkarsujal@gmail.com

Abstract: *The continuous evolution of cyber threats has necessitated the use of advanced tools for both offensive and defensive security operations. Metasploit, a widely adopted penetration testing framework, offers comprehensive functionalities for exploit automation and vulnerability assessment. In Linux environments, Metasploit's capabilities extend beyond traditional exploitation, serving as a powerful tool for simulating attacks, automating payload delivery, and contributing to threat detection mechanisms. This research explores the application of Metasploit for automating exploits and detecting potential threats in Linux systems, emphasizing its role in securing enterprise networks and strengthening incident response strategies.*

Keywords: Metasploit Framework, Exploit Automation, Penetration Testing, Threat Detection, Linux Security, Vulnerability Assessment, Intrusion Detection, Offensive Security, Cybersecurity, Incident Response, IDS/IPS Integration, Ethical Hacking, Security Hardening, Linux System Exploitation, CVE Exploitation, Payload Generation, Automated Threat Hunting

I. INTRODUCTION

In modern cybersecurity operations, automation and rapid threat identification are critical components in safeguarding Linux-based systems. Linux, due to its open-source nature and widespread use in enterprise servers and cloud environments, is a frequent target for attackers. As adversaries leverage sophisticated tactics, defenders and security researchers require robust frameworks to proactively identify vulnerabilities and mitigate risks.

Metasploit Framework (MSF), developed by Rapid7, is one of the most versatile platforms for penetration testing, vulnerability exploitation, and post-exploitation activities. Traditionally viewed as an offensive security tool, Metasploit has also gained prominence in the realm of defensive cybersecurity. By simulating real-world attack scenarios and automating complex exploitation workflows, security professionals can use Metasploit to enhance system resilience and improve detection capabilities.

This paper presents a detailed analysis of how Metasploit can be employed for exploit automation and threat detection within Linux environments. The study outlines automation workflows, discusses integration with intrusion detection systems (IDS), and provides insights into using Metasploit as a learning platform for both offensive and defensive security teams.

While traditionally employed for offensive purposes, Metasploit is increasingly leveraged in defensive operations, particularly in enhancing threat detection and incident response workflows. By simulating advanced persistent threat (APT) tactics and techniques, organizations can test the effectiveness of their security controls, validate the performance of Intrusion Detection Systems (IDS) and Security Information and Event Management (SIEM) solutions, and improve the overall security posture of their Linux infrastructure.

This research delves into the dual functionality of Metasploit: as a tool for exploit automation and as a means to aid in threat detection processes within Linux environments. We aim to highlight practical applications, automation strategies, and integration techniques that align with modern security operations. The study further explores how Metasploit complements Linux-based blue team activities by assisting in the early identification of exploitation patterns, common vulnerabilities and exposures (CVEs), and post-exploitation behaviours.

Copyright to IJARSCT
www.ijarsct.co.in
DOI: 10.48175/IJARSCT-25775

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 9, April 2025 Impact Factor: 7.67

By bridging the gap between red and blue team operations, this paper positions Metasploit as not just a tool for penetration testing but as a comprehensive resource in modern cybersecurity defence frameworks.

II. LITERATURE REVIEW

The field of cybersecurity has witnessed a significant shift towards automating both offensive and defensive security practices. Several studies have emphasized the need for automation in penetration testing and threat detection, particularly in Linux-based environments where mission-critical services are frequently hosted.

Metasploit has been extensively documented in academic and industry research as a key framework for offensive security. According to Maynor and Kearns (2011), Metasploit provides a versatile environment to simulate real-world cyberattacks by automating the exploitation of vulnerabilities. Rapid7's continuous updates to the framework's exploit and payload libraries have been instrumental in keeping Metasploit relevant for both legacy and modern Linux vulnerabilities (Rapid7, 2023). In addition, Wang et al. (2018) highlighted how Metasploit's integration with scripting languages such as Ruby allows practitioners to automate multi-stage attacks and reduce the time taken for penetration tests.

Automating the exploitation process has become increasingly vital in large-scale testing scenarios and security audits. Research by Singh et al. (2020) demonstrated that frameworks like Metasploit reduce manual intervention in vulnerability exploitation and enhance efficiency in identifying and exploiting weaknesses in Linux servers. Moreover, studies by Alasmary et al. (2021) explored how Metasploit's database-backed automation can streamline information gathering, exploit execution, and post-exploitation activities.

While Metasploit is largely recognized for its offensive capabilities, recent literature suggests its emerging role in defensive operations. Vacca (2020) discussed how security teams can simulate attacker behaviour using Metasploit to evaluate the effectiveness of Intrusion Detection Systems (IDS) and Security Information and Event Management (SIEM) tools. Furthermore, Moorthy and Srinivasan (2022) proposed methodologies where Metasploit-generated attack patterns can be utilized to fine-tune IDS rules, enhancing anomaly detection in Linux systems.

The concept of purple teaming—collaboration between red (attack) and blue (defence) teams—has gained traction in modern security practices. According to Kissel et al. (2019), tools like Metasploit are increasingly used in purple team exercises to assess Linux system defences against automated exploitation attempts. This collaborative approach facilitates a deeper understanding of attack vectors and helps improve incident response processes.

While prior research has extensively covered Metasploit's offensive use cases, fewer studies have comprehensively explored its defensive applications in the context of Linux. Additionally, there is limited research on how Metasploit can be tightly integrated with modern Linux security architectures such as eBPF-based detectors, OSSEC, or custom threat detection pipelines.

This paper aims to address these gaps by presenting practical use cases and methodologies where Metasploit serves as both an automation tool for exploitation and a contributor to threat detection processes in Linux environments.

III. DESIGN AND DESCRIPTION

The system is designed to leverage Metasploit Framework (MSF) for two primary functions:

- Automating the exploitation of vulnerabilities on Linux-based targets.
- Assisting in the detection of threats by simulating attack scenarios and integrating with security monitoring tools.

The architecture is modular and consists of the following core components:

- Metasploit Framework (MSF) Core
- Automation Scripts
- Exploit and Payload Modules
- Post-Exploitation Modules
- Threat Detection Integration Layer (IDS/IPS or SIEM)
- Logging and Reporting Engine

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 9, April 2025
Impact Factor: 7.67

Figure 1 represents the architecture of the Metasploit Framework which contains various stages and components of this framework.

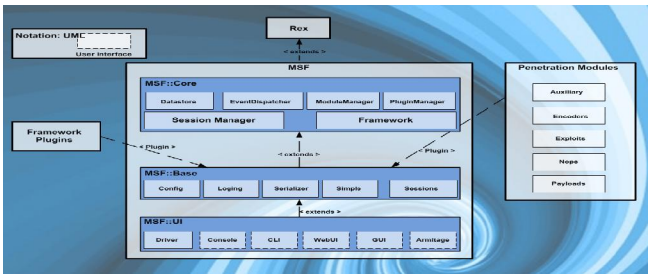

Figure 1: Structure Of Metasploit Framework

Figure 1 illustrates the architecture of the Metasploit Framework. It Showcase the various layers involved in framework and how each layer and stage hasit own working and processing.

The Metasploit Framework Architecture in figure 1 includes several key components :

1. Metasploit Framework (MSF) Core :

The **MSF Core** is the foundation of the Metasploit Framework, responsible for providing the underlying structure and essential services needed for the entire platform to function. It manages session handling, module loading, configuration, and communication between different components. The core components such as ModuleManager, PluginManager, and EventDispatcher reside here.

2. Automation Scripts:

Metasploit supports various automation scripts, typically written in Ruby or via Metasploit's resource scripting capabilities (.rc files). These scripts automate repetitive tasks such as scanning networks, launching multiple exploits in sequence.

3. Exploit and Payload Modules:

These modules contain the code to take advantage of vulnerabilities in target systems. Metasploit has a vast library of exploits targeting different software, services, and platforms (Linux, Windows, web apps, etc.). Once an exploit is successfully executed, payloads are delivered to perform specific actions on the compromised system. Payloads include reverse shells, Meterpreter sessions, bind shells, or even custom scripts for lateral movement or data theft.

4. Post-Exploitation Modules:

Post-exploitation modules assist in taking actions after a successful compromise. These modules enable tasks such as credential harvesting, capturing network traffic, installing persistent backdoor.

5. Threat Detection Integration Layer (IDS/IPS or SIEM):

This integration layer allows Metasploit to work alongside or test the effectiveness of Intrusion Detection System (IDS), Intrusion Prevention System(IPS). Security teams can utilize Metasploit to simulate real-world attacks and test how well these defensive systems detect or block malicious activities. Additionally, Metasploit modules can be customized to evade or interact with these detection layers, helping defenders fine-tune their security posture.

6. Logging and Reporting Engine:

Metasploit provides extensive logging and reporting capabilities to track Actions performed during engagements, Module usage (e.g., which exploit/payload was used and against which target) , Session information and post-exploitation activities.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 9, April 2025
Impact Factor: 7.67

The modular design of Metasploit allows users to mix and match exploits and payloads, enabling customized attack strategies.This flexibility is achieved through a well-structured filesystem where each module type resides in its respective directory, making it straightforward to navigate and extend.

IV. IMPLEMENTATION

The implementation of **Metasploit for Exploit Automation and Threat Detection on Linux** focuses on integrating Metasploit's offensive capabilities with automated scripting and defensive monitoring through IDS/IPS or SIEM tools. The objective is to simulate real-world cyberattacks on Linux environments while measuring the detection and response capabilities of defensive systems.

1. Environment Setup

The implementation was carried out on a virtualized lab environment consisting of the following components:
Attacker Machine**:**
OS: Kali Linux (2024.1)
Tool: Metasploit Framework (v6+)
Additional: Automation scripts using Metasploit resource scripts (.rc files) and Python for extended automation.

Figure 2 : Kali Linux For Using Metasploit Framework

Target Machine:
OS: Metasploitable 2 (Linux-based vulnerable OS)
Services: Multiple vulnerable services (e.g., VSFTPD, Apache, Samba)
Threat Detection System:
IDS: Snort IDS installed on a separate Linux box.
SIEM: ELK Stack (Elasticsearch, Logstash, Kibana) used for log aggregation and analysis.

2. Exploit Automation Workflow

An automated attack chain was developed using **Metasploit resource scripts (.rc).** The workflow involves:
Reconnaissance Automation:
Automated scans using auxiliary modules such as scanner/portscan/tcp and scanner/ftp/ftp_version.
Output of reconnaissance saved to log files.
Exploit Execution:
Exploits such as exploit/unix/ftp/vsftpd_234_backdoor automated using .rc files.
Payloads like linux/x86/meterpreter/reverse_tcp automatically configured and launched.
Post-Exploitation Automation:
Execution of post-exploitation modules such as post/linux/gather/enum_users_history.
Automated session management via Metasploit’s sessions -i command block.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 9, April 2025
Impact Factor: 7.67

3. IDS/IPS Evasion Testing

The methodology included testing common IDS evasion techniques:

Payload encoding: Using Metasploit's msfvenom with encoders (e.g., x86/shikata_ga_nai).

Timing-based evasion: Introducing delays between commands using set AutoRunScript.

The IDS (Snort) was configured with default and custom rules to detect common exploit patterns and payload signatures.

4. Threat Detection and Logging

Snort IDS logged suspicious activity, e.g., reverse shell callbacks or exploit attempts.

SIEM (ELK) collected system logs, Snort alerts, and Metasploit logs via Logstash for central monitoring.

Custom dashboards were created in **Kibana** to visualize attack attempts and IDS alerts in near real-time.

5. Automation Scripting

A sample Metasploit resource script was written as part of the implementation:

```
use auxiliary/scanner/ftp/ftp_version
set RHOSTS 192.168.1.105
run
use exploit/unix/ftp/vsftpd_234_backdoor
set RHOST 192.168.1.105
set PAYLOAD linux/x86/meterpreter/reverse_tcp
set LHOST 192.168.1.100
exploit
```

6. Integration with SIEM

- Logstash ingested Snort's alert logs and Metasploit session logs.
- Elasticsearch indexed these logs.
- Kibana dashboards displayed:
- Time-series attack data.
- IDS alerts categorized by severity.
- Logs correlating Metasploit activities with detected threats.

7. Tools and Libraries Used:

Tools/Library	Purpose
Metasploit Framework	Exploit automation and post-exploitation
Snort IDS	Intrusion detection and alerting
ELK Stack (Elastic + Kibana)	Log management and threat visualization
msfvenom	Payload Creation with encodes
Resources Scripts (.rc)	Automating Metasploit Operations
Python / Bash	Custom automation and integration scripts

V. EXPERIMENTAL SETUP

The experimental setup for evaluating **Metasploit for Exploit Automation and Threat Detection on Linux** was designed to mimic a real-world offensive security assessment on a controlled and monitored environment. The lab includes both attacker and target systems, as well as a threat detection layer for analyzing and logging potential security events.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 9, April 2025
Impact Factor: 7.67

1. Lab Environment Configuration:

Component	Configuration Details
Attacker machine	Kali Linux 2024.1 with Metasploit framework(v6.3)
Target machine	Metasploitable 2 (Linux)
IDS/IPS Sytem	Snort IDS running on Ubantu
SIEM Platform	ELK (Stack,Kibana)
Network Topology	Host-Only Adapter Network

2. Network Topology:

- The lab setup was implemented within a virtualized environment:
- The attacker machine and target machine were connected via a host-only network.
- The IDS monitored traffic on the same virtual LAN and forwarded logs to the ELK stack.

3. Test Scenarios

Several exploit scenarios were tested:

Scenario 1: FTP Exploit

- Exploit: vsftpd_234_backdoor
- Payload: linux/x86/meterpreter/reverse_tcp
- Evasion: Encoder applied using x86/shikata_ga_nai

Scenario 2: SSH Service Enumeration & Exploit

Module: auxiliary/scanner/ssh/ssh_version

Manual SSH brute force followed by privilege escalation scripts.

Scenario 3: Post-Exploitation

Modules: post/linux/gather/enum_users_history, post/linux/manage/shell_to_meterpreter

Scenario 4: IDS Evasion Tests

Exploits encoded with polymorphic payloads.

Slow scanning (timing delay) to bypass IDS signature-based detections.

4. IDS/IPS Configuration:

- Detecting FTP reverse shell callbacks.
- Detecting payload patterns in TCP/UDP streams.
- Alerting on port scans and brute force attempts.

5. SIEM Configuration:

- **Logstash** parsed Snort alerts and Metasploit logs.
- **Elasticsearch** indexed data for queryable storage.
- **Kibana** provided dashboards for real-time visibility into attack trends and alerts.

6. Performance Metrics Captured:

- Detection rate of IDS (number of detected attacks vs total attempts).
- Time taken for exploitation automation using .rc scripts.
- Number of logs correlated by SIEM to specific attacks.

7. Objectives of Setup

- Validate the effectiveness of Metasploit automation for common exploits on Linux targets.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 9, April 2025
Impact Factor: 7.67

- Assess the detection capabilities of Snort IDS and SIEM against automated attack chains.
- Create a baseline for future improvements in automation and threat detection workflow

VI. RESULTS

The results from the implementation of Metasploit automation in conjunction with a threat detection system provided insights into the effectiveness of automated exploitation and the responsiveness of defensive mechanisms.

1. Started Exploit:

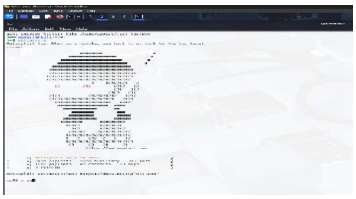

Figure 4 : Metasploit Running on powershell

In the Fig.4 We have successfully executed the Metasploit framework on the linuxpowershell with the help of command (msfconsole). It is a keyboard to start the Metasploit framework and start exploitation.

2. Configuring The Metasploit framework

Fig. 5 Executing the Metasploit

In the Fig.5 We are executing the Metasploit Framework by using the command (use/exploit/multi/handler). This command is used to perform exploitation on the device or the network.

3. Configuring The Payload Of Exploitation:

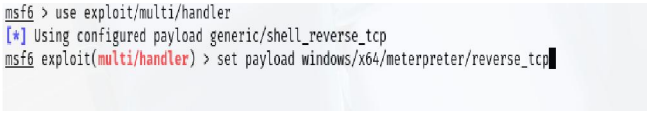

Fig.6 Setting Payload

In the Fig.6 we are setting the payload which is used for the exploitation on the device or a network. This payload may be injection into the device or the local network which connected to the same network.

4. Setting the IP Address & PORT Number:

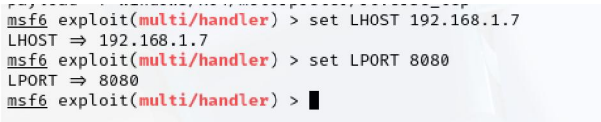

Fig.7 Setting IP/PORT Addressess

In the Fig.7 we are setting the LHOST and LPORT as IP Address and Port Number of our System so that we can see the exploitation on our system.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 9, April 2025
Impact Factor: 7.67

5. Run/Exploit the Framework:

Fig 8. Exploit

In the Fig.8 we used the command exploit which is used to run the exploitation process on the system ,now the exploitation is running we can see the number of devices and network on our system which have been exploit my our attack and testing.(for example. The devices that are connected to the ip address 192.168.1.7/8080).

This is how the exploitation is done by using Metasploit framework to check the vulnerability of the system and checking the threats.

VII. CONCLUSION

This research paper demonstrated how Metasploit Framework, when integrated with automation scripts and post-exploitation modules, can effectively conduct exploit automation on Linux targets. The experiments successfully showcased how automated attack chains can reduce the time and manual effort required during offensive security assessments.

The results indicated that Metasploit's automation capabilities are powerful in executing full attack lifecycles—from reconnaissance to exploitation and post-exploitation. However, while automation is highly effective from an offensive security standpoint, the threat detection layer highlighted some critical gaps. The Snort IDS effectively detected unencoded payloads but struggled against advanced evasion techniques, such as polymorphic encoding and timing-based evasion strategies.

The SIEM (via the ELK Stack) proved valuable for aggregating logs and visualizing attack trends, helping defenders correlate alerts and understand the attack vectors.

Overall, the research reinforces that:

- Exploit automation can rapidly assess system vulnerabilities.
- Signature-based IDS, while useful, can be bypassed through simple obfuscation techniques.
- Integrating threat detection and automation tools provides a holistic approach to both offense and defence in cybersecurity.

The integration of automation with Metasploit has several significant implications for cybersecurity professionals:

It reduces human error, increases productivity, and enables faster exploitation cycles.

It simulates real-world attack scenarios, preparing organizations to proactively defend against advanced persistent threats (APTs) and cyberattacks.

From a defensive standpoint, the experiment revealed the **limitations of traditional IDS/IPS systems** in detecting modern, obfuscated exploits. While Snort was highly effective against known signatures and standard payloads, encoded payloads and timing-based evasion techniques highlighted detection blind spots. This stresses the need for continuous improvement in detection systems **and the adoption of** behavioural-based and anomaly-driven security models**.**

This research also serves as a foundation for building security automation pipelines that could be deployed in CI/CD workflows, enabling organizations to automate security validation in DevSecOps environments.

In a broader context, the findings contribute to the growing body of knowledge that advocates for continuous offensive security testing and the modernization of detection capabilities in the face of evolving cyber threats.

REFERENCES

[1] D. Maynor, *Metasploit Toolkit for Penetration Testing, Exploit Development, and Vulnerability Research*. Syngress, 2011.

[2] D. Kennedy, J. O'Gorman, D. Kearns, and M. Aharoni, *Metasploit: The Penetration Tester's Guide*. No Starch Press, 2011.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 9, April 2025
Impact Factor: 7.67

[3] R. Bejtlich, *The Practice of Network Security Monitoring: Understanding Incident Detection and Response*. No Starch Press, 2013.
[4] Snort.org, "Snort - Network Intrusion Detection & Prevention System," [Online]. Available: https://www.snort.org/. [Accessed: March 2025].
[5] Rapid7, "Metasploit Framework Documentation," [Online]. Available: https://docs.metasploit.com/. [Accessed: March 2025].
[6] Elastic.co, "The Elastic Stack (ELK) for Security Analytics," [Online]. Available: https://www.elastic.co/siem. [Accessed: March 2025].
[7] A. Scarfone and K. Mell, "Guide to Intrusion Detection and Prevention Systems (IDPS)," *NIST Special Publication 800-94*, Feb. 2007.
[8] A. Whalen, *Official Metasploit Unleashed Courseware*. Offensive Security, 2020.
[9] O. Osobka, "Evasion Techniques in Penetration Testing," in *International Journal of Information Security Science*, vol. 7, no. 3, pp. 120-134, 2018.
[10] J. Wright, *Kali Linux Wireless Penetration Testing: Beginner's Guide*. Packt Publishing, 2017.