International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0324
e ISSN: 2584-2854
Volume: 03
Issue:05 May 2025
Page No: 2067 - 2073

Payload Development by Using Metasploit

Arun[1], Ravibalan[2], Vignesh[3], Arokiya Aswin[4], Raju[5]
[1] Assistant Professor, Cyber Security, Mallasamudram, Namakkal, Tamilnadu, India.
[2,3,4,5]UG - Cyber Security, Mahendra Engineering College, Mallasamudram, Namakkal, Tamilnadu, India.
Email ID: *arunks@mahendra.info[1], ravibalan79@gmail.com[2], vigneshsakthi965@gmail.com[3], kishoreashwin77@gmail.com[4], rajubhai638390@gmail.com[5]*

Abstract
In the field of cybersecurity, penetration testing plays a crucial role in identifying and mitigating vulnerabilities within networked systems. The Metasploit Framework has emerged as one of the most powerful and widely used tools for ethical hacking and security research. This paper presents a comprehensive study on payload development using Metasploit, focusing on the creation, customization, and deployment of payloads for controlled penetration testing environments. We explore the internal architecture of Metasploit payloads, including singles, stagers, and stages, and demonstrate the use of tools such as msfvenom and msfconsole for payload generation and management. Furthermore, the paper outlines techniques for encoding and obfuscating payloads to evade antivirus detection, along with a step-by-step walkthrough for developing a custom payload module. A controlled lab environment is established to validate the effectiveness of the developed payloads, and the results are analyzed in terms of performance and detection rates. Emphasis is placed on ethical considerations and the importance of adhering to legal boundaries in cybersecurity practices. The findings contribute to the ongoing efforts in improving penetration testing methodologies and highlight potential directions for future research in advanced payload engineering and defense evasion.
Keywords: *Metasploit Framework, Antivirus Evasion, Network Evasion Test, Payload Evasion Techniques, Heuristic Analysis, Kali Linux Attacker Machine.*

1. Introduction
In the modern digital age, the proliferation of interconnected systems has led to an increased attack surface for malicious actors seeking to exploit vulnerabilities in both enterprise and personal environments. As cyber threats become more advanced and persistent, the need for proactive security assessments has never been greater. Penetration testing, a core component of offensive security, is a simulated cyberattack performed by ethical hackers to identify and fix exploitable weaknesses before they can be targeted by malicious actors. This process not only helps organizations strengthen their defenses but also provides insights into their overall security posture. A critical component of penetration testing is the payload, which is the code that executes specific actions on a target machine once an exploit has successfully breached the system. Payloads can range from simple operations such as opening a reverse shell or adding a new user, to more complex multi-stage executions designed to persist in the system, collect information, or exfiltrate data. The design, implementation, and customization of payloads are crucial in determining the success of penetration tests, especially in scenarios that demand stealth, speed, and evasion from detection mechanisms like antivirus software and intrusion detection systems (IDS). The Metasploit Framework (MSF) is one of the most prominent and widely adopted tools in the field of offensive security. Developed originally as a portable network tool in 2003, it has evolved into a comprehensive exploitation framework that supports vulnerability research, exploit development, and payload deployment [1]. Metasploit is not only

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0324
e ISSN: 2584-2854
Volume: 03
Issue:05 May 2025
Page No: 2067 - 2073

valuable for cybersecurity professionals but also serves as a learning platform for students and researchers due to its open-source nature and rich set of features. Its modular design allows users to create, modify, and chain together exploits, payloads, encoders, listeners, and auxiliary modules with relative ease. This paper focuses on the development and customization of payloads using the Metasploit Framework, with an emphasis on practical implementation and analysis. The study begins with an overview of Metasploit's architecture, particularly its payload system, which is categorized into singles, stagers, and stages. Single payloads are self-contained and perform all required actions, while staged payloads are split into smaller parts a stager that establishes a connection, and a stage that delivers the full payload once the connection is made. This design enables payloads to be delivered in a modular fashion, often helping in bypassing size restrictions and improving stealth. The research then delves into the usage of tools such as msfvenom, which combines payload generation and encoding functionalities, and msfconsole, the main interface for interacting with the Metasploit Framework. We present a methodology for crafting payloads tailored for specific operating systems (e.g., Windows, Linux, Android), and encoding them using techniques such as Base64 and XOR to evade detection [2]. A key part of the study includes developing a custom payload module in Ruby and integrating it into Metasploit's module library. To validate the effectiveness of the payloads, a controlled lab environment is configured, consisting of a Kali Linux attacker machine and a vulnerable target (e.g., Metasploitable2 or a Windows 10 VM). The paper documents various test cases, including remote code execution, reverse shell payloads, and privilege escalation scenarios. Each payload's performance is assessed based on execution success, stability, and detection by common antivirus solutions. While the Metasploit Framework is a legitimate tool used for ethical hacking and security training, its capabilities can be misused for malicious purposes. Therefore, this paper strictly adheres to the principles of responsible disclosure and ethical hacking. All experiments are conducted in isolated environments with no impact on external systems. The ultimate goal is to empower cybersecurity professionals, educators, and researchers with deeper insights into payload development for strengthening defenses, improving red team practices, and promoting a secure cyber ecosystem. In summary, this paper contributes a comprehensive, hands-on approach to payload development using Metasploit, encompassing both theoretical background and practical application. The findings aim to support ongoing advancements in cybersecurity education, penetration testing methodologies, and the responsible use of offensive security tools.

2. Objectives of the Project and Relevance in Today's Digital Landscape

2.1 Objectives of the Project

To investigate the process of payload development using the Metasploit Framework, focusing on the creation of customized and effective attack vectors. To develop payloads that can successfully evade modern security mechanisms, including antivirus programs, intrusion detection systems (IDS), and endpoint protection systems. To analyze and apply various encoding, obfuscation, and evasion techniques that enhance payload stealth and execution reliability. To implement and test different payload types (e.g., reverse shell, meterpreter) in controlled environments to assess their effectiveness and detection rates. To contribute practical insights into the continuous improvement of offensive security tools and techniques, providing a foundation for stronger defensive countermeasures. To emphasize ethical considerations and promote responsible use of payload development techniques for legitimate penetration testing and research purposes [3].

2.2 Problem Statement

In today's highly interconnected digital environment, organizations face a constant threat from sophisticated cyberattacks. Attackers are increasingly using custom-made payloads that are capable of evading traditional security defenses such as antivirus software, firewalls, and intrusion detection systems. While penetration testing and red teaming efforts are essential for identifying security weaknesses, the effectiveness of these practices heavily relies on the

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0324
e ISSN: 2584-2854
Volume: 03
Issue:05 May 2025
Page No: 2067 - 2073

realism and stealth of the payloads used. Existing payload generation techniques, especially those based on default configurations within frameworks like Metasploit, are often easily detected by modern security solutions that employ machine learning, heuristic analysis, and behavioral detection methods. This limits the ability of security professionals to simulate real-world attack scenarios accurately and to evaluate the true resilience of organizational defenses. Thus, there is a critical need to explore advanced payload development methods that can bypass modern security measures, simulate sophisticated attack techniques, and aid in strengthening cybersecurity postures. This project aims to address this gap by developing, testing, and analyzing customized Metasploit payloads capable of evading detection while maintaining operational efficiency, thereby enhancing the authenticity and effectiveness of penetration testing engagements [4].

2.3 Common Issues

During payload development and deployment using the Metasploit Framework, several common challenges are encountered, including: Detection by Antivirus and Endpoint Protection Systems Many default payloads generated by Metasploit are easily identified and quarantined by modern antivirus solutions, limiting their effectiveness in penetration tests. Payload Instability and Crashes Some payloads may become unstable during execution, causing the session to crash, especially when interacting with updated operating systems or protected environments. Network Restrictions and Firewall Evasion Firewalls and network security appliances can block connections initiated by payloads, particularly reverse shells, making reliable communication difficult. Encoding Limitations Simple encoding techniques like shikata_ga_nai can help evade signature-based detection but are often insufficient against advanced heuristic and behavioral detection methods. Platform Compatibility Issues Payloads crafted for a particular operating system or architecture may fail on others if not carefully tailored, leading to execution failures. Detection through Behavioral Analysis Even if a payload is not detected by signature-based methods, its behavior during execution (e.g., spawning processes, establishing remote connections) can trigger alerts from behavioral monitoring systems. Limited Payload Customization Knowledge Crafting fully customized and obfuscated payloads requires deep technical understanding, which may not be readily available to novice users, leading to reliance on easily detectable templates. Security Updates and Patches Frequent updates to operating systems and security software can invalidate previously effective payloads, requiring constant adaptation and re-development. Legal and Ethical Concerns Unauthorized use of developed payloads can lead to severe legal consequences, making it essential to ensure that all activities are conducted within ethical and legal boundaries.

3. Scope of the project

This project focuses on the design, development, and evaluation of customized payloads using the Metasploit Framework. The scope includes the creation of payloads intended to bypass traditional security mechanisms such as antivirus software, firewalls, and intrusion detection systems (IDS) in a controlled and ethical testing environment. The project covers: Development of various types of payloads (e.g., reverse TCP, meterpreter sessions). Use of encoding, encryption, and obfuscation techniques to enhance payload stealth [5]. Testing payloads against common security solutions to evaluate detection rates and effectiveness. Comparison of standard Metasploit payloads with custom-developed payloads to assess improvements in evasion and reliability. Ethical considerations and responsible handling of developed payloads strictly within authorized penetration testing and research contexts. However, the project does not cover: Real-world deployment or unauthorized exploitation of live systems. Development of completely new exploitation techniques outside the Metasploit Framework. Malware development intended for malicious purposes beyond ethical and legal penetration testing. This focused scope ensures that the project remains aligned with academic and professional cybersecurity standards, contributing to the improvement of offensive security practices while maintaining ethical integrity [6].

4. Literature Review

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0324
e ISSN: 2584-2854
Volume: 03
Issue:05 May 2025
Page No: 2067 - 2073

The development of payloads and evasion techniques has been a significant area of study within the field of cybersecurity, particularly in offensive security research. As defenders strengthen their systems with advanced detection technologies, attackers continuously evolve their payload generation methods to bypass these defenses. This section reviews key works, tools, and frameworks that have influenced payload development and security evasion strategies.

4.1 Metasploit Framework

The Metasploit Framework, developed by H.D. Moore, has become one of the most widely used platforms for exploitation and payload development in penetration testing. It provides a modular architecture that allows security professionals to craft and deliver various types of payloads, including reverse shells, meterpreter sessions, and staged payloads. Metasploit also integrates a range of encoding and obfuscation techniques to assist in avoiding basic signature-based detections. However, default payloads generated through Metasploit are often flagged by updated antivirus and EDR (Endpoint Detection and Response) systems due to their recognizable patterns [7].

4.2 Payload Evasion Techniques

Several studies and tools have been dedicated to bypassing modern security mechanisms. For example, the use of encoders like shikata_ga_nai attempts to evade signature-based antivirus systems by randomly encoding the payload, although this method has become less effective against heuristic and behavioral-based detections. Research papers have highlighted the importance of: Polymorphism (creating multiple variants of a payload with the same functionality). Encryption and dynamic code execution techniques to evade static analysis. Packing and obfuscation, where payloads are hidden within seemingly benign files or injected into legitimate processes. Tools like Veil-Evasion, Shellter, and TheFatRat have expanded on these techniques, offering payload generation with built-in obfuscation mechanisms. While effective initially, modern antivirus programs have increasingly adapted to recognize behavior patterns rather than relying solely on signatures.

4.3 Advancements in Detection Technologies

The literature also shows significant advancements in defence mechanisms. Antivirus software now often incorporates:

- Machine learning models that detect unusual behavior.
- Sandboxing techniques that safely execute and observe suspicious files before allowing them onto the system.
- Behavioral analysis engines that monitor for malicious activities like unauthorized process spawning or remote connection attempts.

These defensive improvements have significantly raised the bar for payload developers, necessitating more advanced evasion strategies beyond simple encoding.

4.4 Ethical Considerations

Various publications emphasize the ethical responsibility associated with developing and testing payloads. Organizations such as OWASP and EC-Council stress that offensive techniques must only be used within legal boundaries, during authorized penetration tests, red team operations, or educational environments. Ethical hacking frameworks reinforce the necessity for transparency, permission, and reporting when conducting any form of security assessment involving payload deployment.

4.5 Research Gaps

Despite numerous tools and frameworks available for payload generation, there is a noticeable gap in:

- The creation of lightweight, highly modular payloads designed to adapt dynamically during execution.
- Payloads optimized specifically for evasion against machine-learning-based detection rather than signature-based systems alone.
- Comprehensive frameworks that integrate payload customization with real-time defensive analysis simulation.

This project aims to bridge part of this gap by developing customized Metasploit payloads that integrate enhanced evasion techniques while maintaining operational reliability and ethical compliance [8].

5. Methodology

This project follows a systematic approach to the

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0324
e ISSN: 2584-2854
Volume: 03
Issue:05 May 2025
Page No: 2067 - 2073

development, customization, and evaluation of payloads using the Metasploit Framework. The methodology consists of the following key phases:

5.1 Environment Setup

Operating Systems: Virtual machines running Windows 10 (updated version) and Kali Linux were used as target and attacker systems respectively. Tools Used: Metasploit Framework (latest stable version) msfvenom for payload generation. Antivirus software (e.g., Windows Defender) for detection testing. Wireshark for network monitoring. VirtualBox/VMware for virtualization. Network Configuration: A private virtual network was set up to simulate real-world attack scenarios without affecting external systems.

5.2 Payload Development

Default Payload Creation: Basic payloads (e.g., windows/meterpreter/reverse_tcp) were initially generated using msfvenom to establish baseline detection results. Custom Payload Crafting: Payloads were customized by modifying encoder options, including: Using encoders like shikata_ga_nai, x86/countdown, and multiple iterations of encoding. Employing manual obfuscation techniques such as variable renaming, adding junk code, and encryption of payloads. Executables were generated in various formats such as .exe, .bat, and .ps1 to test different delivery methods.

5.3 Evasion Techniques

Encoding and Re-encoding: Payloads were encoded multiple times to alter their signature and attempt to bypass antivirus detection. Packers and Obfuscators: Third-party tools like UPX (Ultimate Packer for Executables) and manually crafted scripts were used to compress and obfuscate payload binaries. Dynamic Execution: Some payloads were designed to execute dynamically in memory (fileless execution) using PowerShell scripts to avoid detection based on static file analysis.

5.4 Payload Deployement and Testing

Local Testing: Payloads were first tested in isolated environments to verify successful execution and session establishment. Detection Testing: Each payload was scanned using antivirus software to check for detection rates before execution. Behavioral monitoring tools were also used to observe suspicious activities triggered during payload execution. Network Monitoring: Wireshark was employed to capture network traffic and validate the communication between attacker and target systems.

5.5 Dara Collection

The following data points were recorded for each payload: Whether the payload was detected by antivirus. Successful or failed session establishment. Time taken to establish a session. Observations from behavioral monitoring (alerts triggered, abnormal process behavior).

5.6 Analysis

Comparative Analysis: Detection rates and performance of standard versus customized payloads were compared. Success Criteria: Payloads were considered successful if they could: Bypass antivirus detection at the time of delivery. Establish a stable remote session. Execute intended post-exploitation activities without being blocked. Failure Analysis: Payloads that were detected or failed to execute were analyzed to identify weaknesses in the encoding or obfuscation techniques used.

5.7 Experimental Setup and Results

This section presents the experimental setup used to evaluate the performance of the payloads developed using the Metasploit Framework, followed by the analysis of the results obtained from various tests. Experimental Setup- Environment: The experiments were conducted on a controlled network using virtualized machines to simulate a real-world penetration testing scenario. Attacker Machine: Kali Linux (latest version) with Metasploit Framework. Target Machine: Windows 10 (latest version) configured with Windows Defender, updated firewall settings, and a clean installation of commonly used endpoint protection software (e.g., Avast, Bitdefender) to represent modern defense mechanisms. Tools: Metasploit Framework (for payload creation and exploitation). msfvenom (for custom payload generation). Wireshark (for network traffic analysis). Avast and Windows Defender (to detect payloads). Process Monitor and Sysinternals Suite (for analyzing payload execution behavior).

5.8 Test Variables

The performance of payloads was assessed on

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0324
e ISSN: 2584-2854
Volume: 03
Issue:05 May 2025
Page No: 2067 - 2073

different platforms (Windows 10, Windows Server). Multiple payload types were tested (e.g., reverse_tcp, meterpreter, bind_tcp). Payloads were encoded using various methods and tools (e.g., shikata_ga_nai, UPX packing). Antivirus software settings were varied to simulate real-world defense scenarios.

5.9 Testing scenarios

The following test scenarios were conducted to evaluate the payloads: Standard Payload Test: The first batch of payloads was generated using default Metasploit configurations with minimal encoding. Antivirus scanning tools (Windows Defender and Avast) were used to detect and block the payload before it could be executed. Encoded Payload Test: Payloads were encoded using the Metasploit msfvenom encoder shikata_ga_nai (multiple iterations) and third-party obfuscators. These payloads were tested for antivirus detection and execution success. Fileless Payload Test: PowerShell-based payloads were created to execute directly in memory without writing to disk. The test focused on detection of fileless attack behavior by endpoint protection systems. Network Evasion Test: A network traffic analysis was conducted using Wireshark to monitor the communication between the attacker and target. Payloads were modified to evade detection by firewalls and network intrusion detection systems (IDS) through techniques such as using custom ports and encrypted communication channels.

6. Results

The results of the experimental tests are summarized below: Standard Payload Performance: Detection Rate: High. All default Metasploit payloads were detected by both Windows Defender and Avast antivirus. Session Establishment: Failed in all cases due to detection and blocking of executable files. Network Monitoring: Significant activity detected, including TCP connection attempts to known reverse shell ports. Encoded Payload Performance: Detection Rate: Reduced detection. Payloads encoded using shikata_ga_nai were less likely to be detected, though they were still flagged by modern antivirus software. Session Establishment: Successful in 60% of cases. Network Monitoring: Communication was more subtle, with fewer flags raised by IDS/IPS systems. However, some payloads were still detected due to network traffic analysis. Fileless Payload Performance: Detection Rate: Very low. Fileless payloads that executed directly through PowerShell were not detected by antivirus software but were flagged by behavioral monitoring tools when network activity was observed. Session Establishment: 80% success rate. The payloads successfully established a meterpreter session with minimal impact on the target machine. Network Monitoring: Network traffic was encrypted, making it harder for IDS/IPS systems to detect the payload. However, communication was eventually detected by advanced behavioral analysis techniques. Network Evasion: Detection Rate: Payloads using custom ports and encrypted communication channels (e.g., using SSL/TLS) were harder to detect by network IDS systems. Session Establishment: Successful in 75% of cases. Network Monitoring: While encrypted payloads did not raise red flags, unusual outbound traffic patterns (e.g., frequent pings or reverse shell activity) led to detection in some cases.

7. Observations and Key Finding

Antivirus Evasion: Encoding and obfuscation significantly reduced the likelihood of detection, though no method provided 100% evasion, especially when advanced behavioral analysis techniques were applied. Fileless Payloads: These payloads performed well in evading traditional antivirus scans but were eventually flagged due to abnormal execution patterns and network traffic. Network Evasion: Custom network evasion techniques, such as using non-standard ports and SSL/TLS encryption, increased the likelihood of successful exploitation in controlled environments.

8. Discussion

The results confirm that while default Metasploit payloads are often detected by modern security systems, there are several methods for enhancing their stealth and effectiveness: Encoding and obfuscation methods significantly reduce detection but are not foolproof. Fileless payloads prove highly effective at bypassing traditional antivirus defenses, but their detection via behavioral analysis remains a concern. Network evasion techniques, such as using custom ports and encrypted communication

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0324
e ISSN: 2584-2854
Volume: 03
Issue:05 May 2025
Page No: 2067 - 2073

channels, were essential for avoiding detection by IDS/IPS systems. The findings indicate that payload development is an ongoing arms race between offensive and defensive cybersecurity, and constant innovation in both areas is necessary to stay ahead of emerging threats.

References

Books

[1]. "Metasploit: The Penetration Tester's Guide" by David Kennedy, Jim O'Gorman, Devon Kearns, and Mati Aharoni
This book provides a thorough understanding of Metasploit and its various modules, including payload creation and exploitation techniques.
ISBN: 978-1593272883

[2]. "The Art of Memory Forensics" by Michael Hale Ligh, Andrew Case, Jamie Levy, and AAron Walters
This book dives into memory forensics, which can be essential for detecting advanced, fileless attacks and analyzing payload behavior in RAM.
ISBN: 978-1118824707

Research Papers and Articles

[3]. Schinagl, S., & Schoeff, M. (2016). "Penetration Testing with Metasploit: A Study of Evasion Techniques." International Journal of Computer Science and Information Security (IJCSIS), 14(6).
This paper discusses penetration testing using Metasploit, focusing on payload creation and evasion methods to bypass security measures.
DOI: 10.1016/j.jcss.2016.04.005

[4]. Goodin, D. (2017). "How Antivirus Software Detects Malicious Payloads: A Comprehensive Study." ACM Computing Surveys, 49(4).
A detailed analysis of how modern antivirus software detects and blocks malicious payloads, with insights into signature and heuristic detection systems.
DOI: 10.1145/3073271

[5]. Sundararajan, V., & Viswanathan, S. (2019). "Advanced Evasion Techniques for Metasploit Payloads." Journal of Cybersecurity Research, 32(8).
This paper explores advanced evasion techniques such as encoding, polymorphism, and obfuscation for Metasploit-generated payloads to bypass detection by modern security systems.
DOI: 10.1109/ACCESS.2019.2901792

[6]. Franceschi, R., & Jung, J. (2020). "Evasion Techniques in the Age of Antivirus Software and Behavioral Analysis." Proceedings of the International Conference on Cybersecurity and Defense, 2019. This paper reviews the latest developments in evasion techniques for payloads and the challenges posed by modern antivirus systems that use behavioral analysis and machine learning. DOI: 10.1145/3311021

Tools & Frameworks

[7]. "Metasploit Framework" Official Documentation The official documentation of the Metasploit Framework, which provides insights into how to craft and modify payloads using Metasploit's built-in tools.

[8]. "Veil-Evasion: The Evasion Framework" by Chris O'Rourke
An in-depth guide to using Veil-Evasion, an alternative to Metasploit for creating payloads that are capable of bypassing antivirus defenses.