Web Application Vulnerabilities and Exploitation Techniques

Vulnerability Exploit and Patch

Team 10.2

Animikha Sinha

Practice Website: Metasploitable2

Target Website: www.palashbitan.com

Chapter Index

Figures Index

Acknowledgement

I would like to acknowledge the following individuals and organization for their support and guidance throughout my project on vulnerability exploitation:

Firstly, I would like to express my gratitude to SmartBridge for providing the course and creating an environment where I could enhance my knowledge and skills in the field of Cyber Security and Ethical Hacking. The resources, materials, and hands-on training offered by SmartBridge were invaluable in shaping my understanding of this subject.

I am extremely thankful to my mentors, Manoj Sir and Mahesh Sir, for their unwavering support and expert guidance throughout the project. Their depth of knowledge, patience, and willingness to share their expertise made a significant impact on my learning experience. Their insights and feedback helped me navigate complex challenges and develop a deeper understanding of vulnerability exploitation techniques.

I would like to express my immense gratitude to the owners of Palash Bitan for permitting me to use their official website as my target website. Through this I gained real-world experience.

I would also like to extend my appreciation to my fellow students who actively participated in discussions, shared their insights, and contributed to a collaborative learning environment. Their diverse perspectives and engagement enriched the overall learning experience.

Lastly, I would like to thank all the individuals and organizations whose vulnerability exploitation research, tools, and resources I utilized during the project. Their contributions to the field laid the foundation for my work and allowed me to explore and experiment with different techniques.

Without the support and guidance of these individuals and organizations, this project would not have been possible. I am truly grateful for their contributions to my learning journey in vulnerability exploitation.

ANIMIKHA SINHA 20BCE2388

1. Introduction

1.1. Overview

In an increasingly interconnected digital world, the security of websites and online systems has become a critical concern. The rise in cyberattacks and data breaches has exposed the vulnerabilities present within these systems, highlighting the urgent need for robust security measures. Vulnerability exploitation, a complex process involving the identification and exploitation of weaknesses in software applications, networks, and systems, plays a crucial role in understanding and mitigating potential risks.

This comprehensive project delves deep into the intriguing field of vulnerability exploitation, with a specific focus on a carefully selected target website. Conducted with explicit permission from the website owner, the project aims to provide valuable insights into the vulnerabilities faced by websites and the effectiveness of various tools and techniques in identifying and exploiting them.

The primary objective of this project is to gain a comprehensive understanding of the methods and techniques employed by attackers to exploit vulnerabilities in web applications and systems. To achieve this, the project leverages the powerful arsenal of tools available in Kali Linux, including renowned tools like Metasploit, Nmap, Telnet, and others. These tools serve as essential assets in exploring and comprehending the vulnerability discovery, analysis, and exploitation process.

The project adheres to a well-structured methodology, encompassing several key stages. First and foremost, a meticulous selection process is conducted to choose a target website with the explicit consent of its owner, ensuring ethical boundaries are maintained throughout the project. Subsequently, the project embarks on a thorough port scanning phase, utilizing tools such as Nmap to identify open ports on the target website. This initial reconnaissance provides crucial insights into potential entry points and vulnerabilities that can be exploited.

Following the port scanning phase, an in-depth vulnerability assessment is carried out. The project employs a range of advanced tools and techniques to analyze the target website's software, network configuration, and application design, aiming to uncover potential weaknesses. Leveraging tools like Metasploit, the project meticulously scrutinizes and evaluates the identified vulnerabilities, determining their potential for exploitation.

Once armed with a comprehensive understanding of the vulnerabilities, controlled attempts to gain access to the target website are initiated. The project employs various exploitation techniques, including but not limited to brute-forcing, SQL injection, and other relevant methods, always adhering to ethical boundaries. The objective is to demonstrate the potential impact of successful exploitation and emphasize the criticality of implementing robust security measures to prevent such attacks.

Throughout the project, thorough and meticulous documentation is maintained, meticulously capturing findings, techniques utilized, and outcomes. This comprehensive record serves as a valuable resource for in-depth analysis, enabling a deeper understanding of the vulnerabilities and their potential consequences. Furthermore, this documentation significantly contributes to the broader field of cybersecurity, providing valuable insights into the effectiveness of vulnerability scanning tools and shedding light on the challenges faced in securing web applications and systems.

By raising awareness among website owners, administrators, developers, and the broader community, this project seeks to emphasize the significance of proactive security measures and continuous monitoring. Serving as an educational resource, it promotes best practices in secure web development, highlights the importance of staying informed about potential vulnerabilities and their exploitation methods, and encourages a proactive approach to web security.

In conclusion, this comprehensive project endeavours to bridge the gap between theoretical knowledge and practical implementation by exploring vulnerability exploitation within a controlled and ethical environment. Focusing on a carefully chosen target website and harnessing the power of cutting-edge tools, it seeks to enhance our collective understanding of website security, empower website owners with knowledge to protect their digital assets, and make valuable contributions to the ever-evolving field of cybersecurity.

1.2. Purpose

The primary purpose of a project on vulnerability exploitation is to gain a deep understanding of the methods and techniques employed by attackers to exploit vulnerabilities in web applications and systems. By simulating real-world scenarios in a controlled environment, the project allows researchers to explore the vulnerabilities that websites may face and the potential impact of successful exploitation. It provides insights into the weaknesses that can be targeted by malicious actors and highlights the critical need for implementing comprehensive security measures.

One of the central objectives of the project is to identify and document vulnerabilities in the target system. By leveraging tools and techniques like port scanning, vulnerability assessment, and exploitation attempts, researchers can uncover potential weaknesses and vulnerabilities that could be exploited by attackers. This process enables a comprehensive understanding of the specific risks faced by the target system, facilitating the development of effective mitigation strategies.

Furthermore, the project aims to demonstrate the efficacy and limitations of various vulnerability scanning and exploitation tools. By utilizing tools found in Kali Linux, such as Metasploit, Nmap, and Telnet, researchers can assess the effectiveness of these tools in identifying vulnerabilities, validating their potential impact, and understanding their role in strengthening overall security measures. This knowledge can contribute to the improvement and advancement of vulnerability scanning tools, benefiting the broader cybersecurity community.

Another vital purpose of the project is to raise awareness among website owners, administrators, developers, and the general public about the importance of proactive security measures. By showcasing the potential risks and consequences of vulnerabilities, the project underscores the criticality of implementing robust security practices and continuously monitoring systems. It serves as an educational resource, enlightening stakeholders about the evolving threat landscape and encouraging the adoption of best practices in secure web development and system administration.

Additionally, the project contributes to the broader field of cybersecurity by providing valuable insights and data regarding vulnerability exploitation. The documented findings, techniques employed, and outcomes serve as a rich resource for further research and analysis. They offer researchers, industry professionals, and policymakers an in-depth understanding of the current state of vulnerabilities, the effectiveness of existing security measures, and the areas that require attention and improvement.

The project on vulnerability exploitation holds immense significance in the realm of cybersecurity. It serves as a proactive approach to identifying and mitigating potential risks, allowing website owners and administrators to fortify their systems against malicious activities. By understanding the vulnerabilities and exploitation techniques, stakeholders can take informed actions to protect sensitive data, maintain business continuity, and safeguard the privacy and trust of their users.

Furthermore, the project contributes to the broader cybersecurity community by fostering knowledge sharing and collaboration. The documentation of vulnerabilities, exploitation techniques, and

mitigation strategies enables researchers, cybersecurity professionals, and developers to learn from the project's insights, furthering the collective understanding of vulnerabilities and enhancing the effectiveness of security measures.

In today's rapidly evolving digital landscape, the security of websites and online systems has become a paramount concern. The increasing frequency and sophistication of cyberattacks highlight the vulnerabilities that exist within these systems, underscoring the urgent need for robust security measures. Vulnerability exploitation, the process of identifying and exploiting weaknesses in software applications, networks, and systems, plays a crucial role in understanding and mitigating potential risks. The project's significance also lies in its role in promoting an ethical approach to vulnerability exploration. By seeking explicit permission from website owners, adhering to ethical guidelines, and maintaining a controlled environment, the project demonstrates the importance of responsible behaviour in the cybersecurity domain. It emphasizes the need for ethical hacking practices, encouraging individuals to apply their skills and knowledge for positive purposes, ultimately contributing to a safer digital ecosystem.

In conclusion, a project focused on vulnerability exploitation holds a crucial purpose and significant value in the realm of cybersecurity. By exploring the methods and techniques employed by attackers, identifying vulnerabilities, and highlighting the importance of proactive security measures, such a project enables website owners, administrators, developers, and the broader community to better protect digital assets. It fosters knowledge sharing, enhances the understanding of vulnerabilities, and contributes to the ongoing efforts in strengthening cybersecurity practices. Ultimately, the project aims to create a more secure and resilient digital environment, safeguarding critical data and ensuring the trust and confidence of users.

2. Literature Survey

2.1. Existing Problem

The practise of preventing cyberattacks on critical infrastructure and private data is known as cybersecurity. Government agencies, banks, hospitals, and businesses of all sizes are increasing their investments in cybersecurity infrastructure to protect their operations and the millions of customers who entrust them with their personal information. In a world where businesses are more interconnected than ever before, cyber threat activity is concerning, raising questions about how well organisations can protect themselves from widespread attacks. Natural Language Processing is used in threat intelligence solutions to read and interpret the meaning of words and technical data in various languages and identify trends. NLP is making it easier for machines to analyse various data sources in multiple languages. **Singh et al.[1]** intend to create a system that treats software vulnerability detection as a Natural Language Processing (NLP) problem, with source code treated as texts, and addresses automated software vulnerability detection with advanced deep learning NLP models that are currently available. We created and compared various deep learning models for accuracy, and the best performer achieved 95% accuracy. In addition, we attempted to predict which vulnerability class a specific source code belongs to and created a robust dashboard using FastAPI and ReactJS.

In recent years, an increasing number of security flaws have been discovered and reported. Much of the information about these vulnerabilities is currently available to the public in the form of rich, textual data (for example, vulnerability reports). Many of the cutting-edge techniques used to process such textual data today rely on so-called word embeddings. Several pre-trained embeddings have been created as of today, many of which rely on general-purpose training datasets like Google News and Wikipedia. Other domain-specific word embeddings have recently been developed (for example, in the context of software development) to address the terminology and ambiguity limitations of existing general-purpose embeddings. The availability of specialised domain-specific word embeddings is critical for the effectiveness of domain-specific tasks that rely on this technique. **Mumtaz et al.** proposed a word embedding for the domain of cyber security vulnerabilities. They trained the embedding model on a variety of rich and diverse security vulnerability information sources freely available on the internet. The advantages of such specialised word embedding are demonstrated by a qualitative comparison of word similarity and the illustrative task of matching security professionals to vulnerability discovery tasks posted to bug bounty programmes. They also presented a new dataset of word pairs that are similar to human judgements and can be used as a benchmark.

As academic fields of study, ethical hacking and vulnerability assessments are rapidly gaining traction. Nonetheless, it is not always clear what research areas are included in the categories or how they fit into the traditional academic framework. Previous studies in the field reviewed literature, but they relied on manual analysis and thus failed to provide a comprehensive view of the domain. 537,629 related articles from the Scopus database were analysed to gain a better understanding of how the topic is treated in academia. A Python script was used for data mining as well as data analysis, and the final synthesis included 23,459 articles. **Heiding et al.[3]** noted that the articles' publication dates ranged from 1975 to 2022. They were written by 53,495 different people and received a total of 836,956 citations. The Louvain community detection algorithm detected fifteen research communities (smart grids, attack graphs, security testing, software vulnerabilities, Internet of Things (IoT), network vulnerability, vulnerability analysis, Android, cascading failures, authentication, Software-Defined Networking (SDN), spoofing attacks, malware, trust models, and red teaming). Furthermore, each community had several individual subcommunities, for a total of 126. According to the trends of the studies examined, research interest in ethical hacking and vulnerability assessment is growing.

Since IEC 61850, cyber security vulnerabilities in smart substations have emerged as a growing concern. Intelligent substations relied heavily on information and communication technologies. **Chai and Liu [4]** proposed a comprehensive vulnerability assessment platform to assess the cyber security vulnerability of smart substation automation system devices and networks. Cyber-attack simulation and fuzz testing based on TCP/IP and IEC 61850 are two assessment methods used in the platform to detect potential system and protocol loopholes. The platform also includes a compliance evaluation approach for determining smart substation management compliance in accordance with IEC 62443-3-3 requirements. The evaluation platform was built on a test substation using a real-time digital simulator (RTDS).

The cyber system is critical in supervising and controlling the power system. Aside from providing much convenience to the power industry, the cyber system poses some potential danger due to its inherent vulnerability. It is critical to assess the vulnerability of the cyber system, determine its risk to the power industry, identify weak points, develop appropriate strategies to avoid potential accidents, and improve the cyber system's safety. **Yu et al.[5]** propose two methods to assess cyber security vulnerability after analysing the threats and vulnerabilities of cyber systems, primarily the vulnerability of SCADA (supervisory control and data acquisition) systems, EMS (energy management systems), and MIS (management information systems).

The threat of cyber-attacks on vulnerable organisations has grown significantly in recent years. These attacks may combine to exploit a vulnerability breach in a system's protection strategy, potentially resulting in asset loss, damage, or destruction. As a result, every vulnerability comes with a risk, which is defined as the "intersection of assets, threats, and vulnerabilities." **Coleman et al.[6]** intend to compare the similarity-based ranking of cyber security information using a recommendation environment experimentally. The Memory-Based Collaborative Filtering technique, specifically the User-Based and Item-Based approaches, were used. These systems used data from the National Vulnerability Database to identify and rank cyber-security vulnerabilities in hardware and software applications based on similarity. Experiments were carried out using the Item-Based technique to identify the best system parameters, which were then evaluated using the AUC metric. Once identified, the Item-Based technique was compared to the User-Based technique, which used the previously identified parameters. The Pearson's Correlation Coefficient and the Cosine Similarity Measure were used in these experiments. These experiments revealed that when the Item-Based technique was used with the Cosine similarity measure, an AUC evaluation metric of 0.80225 was obtained.

Robots have become more integrated than ever before in various domains such as agriculture, medicine, industry, military, police (law enforcement), and logistics as a result of the recent digital revolution. Robots are dedicated to serving, facilitating, and improving human life. However, numerous incidents have occurred, resulting in serious injuries and devastating consequences such as the unnecessary loss of human lives. Unintentional accidents will always occur, but those caused by malicious attacks pose a particularly difficult problem. This includes maliciously hijacking and controlling robots, resulting in significant economic and financial losses. **Yaacoub et al.[7]** examine the primary security vulnerabilities, threats, risks, and their consequences, as well as the primary security attacks in the robotics domain. Various approaches and recommendations are presented in this context in order to enhance and improve the security level of robotic systems, such as multi-factor device/user authentication schemes and multi-factor cryptographic algorithms. We also look at some of the most recent security solutions for robotic systems.

Brandao et al.[8] show how the elements of a cybersecurity incident can be systematically analysed and propose an alternative method to mitigate the causes and consequences of such incidents. Cybersecurity incidents can be explained in terms of a series of elements that connect the attacking agents to their objectives: the attacking agent uses tools to exploit vulnerabilities, causing actions on a specific target to obtain unauthorised results, thereby achieving their objectives. Stopping the flow of the attack by mitigating one or more elements of the process can improve cyber security. Unfortunately, most of these elements have characteristics that limit mitigation options. Vulnerability is the easiest element to mitigate. The current vulnerability mitigation model has performed well in the corporate environment, which can afford specialised tools and consulting. This is an excellent business model, but it is not available to the general public. To prevent cybersecurity incidents on a broader, more inclusive scale, a new model is required. The main vulnerability mitigation proposal is multisector collaboration to create an independent, trustworthy, and secure vulnerability database based on a new vulnerability report protocol developed in collaboration with researchers, businesses, governments, and society. This proposal, however, raises some social, political, and technical issues.

Gamez [9] describes the basic elements of Ethical Hacking to understand the necessary concepts for the use of automated remote intrusion tools, the modules that are necessary to carry out a successful attack using Metasploit are detailed, the basic remote manipulation commands are also described, once the intrusion is achieved.

The technological era has seen many new inventions emerge, and with them, the need to secure our systems. In this paper, we discussed how the older generation is falling behind in terms of keeping up with technology and losing track of the knowledge required to process it. Furthermore, this factor contributes to the leakage of sensitive personal information. **Thapa et al.[10]** describe the steps taken to exploit the pre-existing operating system, Windows 7, Ultimate, by utilising a widely used framework, Metasploit. It entails installing a backdoor on the victim machine from a remote setup, typically a Kali Linux operating system. This backdoor enable attacker to create executable files and deploy them in the Windows system in order to gain remote access to the machine. Manipulation of sensitive data becomes simple once access is gained. Access to admin rights on any system is a red flag because it indicates that an outsider has extensive access to a human being's personal information, and data about someone reveals a lot about them. It is basically exposing, and humans despise that. It robs them of their personal identity. As a result, security is not something to be taken lightly. It is supposed to be handled with extreme caution.

Mirai is a living malware that targets and constantly threatens IoT devices. IoT malware illegally infiltrates IoT devices, causes them to download other malware such as bots, and infects them. As a result, in order to improve the security of IoT devices, it is necessary to analyse the behaviours of IoT malware and implement countermeasures. **Yamauchi et al.[11]** analysed Telnet logs collected by an IoT device honeypot in this study to analyse the behaviours of IoT malware after entering IoT devices and propose new security functions for operating systems to prevent activities such as IoT malware infection. Following that, we present the findings of our analysis of IoT malware input commands. The results show that IoT malware frequently executes commands related to shell execution, file download, and changing file permissions.

The unprecedented growth in information technology and information explosion, with more and more data in electronic forms has put the computer into the hands of users with very little technical knowledge. The fact that the systems are not inherently immune, and that open up a number of vulnerabilities, leading to potential attacks, and the most prominent is in the form of open ports. **Mathew et al.[12]** attempted to do a survey on common user computing devices including start devices to discover open ports and thereby explore vulnerabilities that can lead to potential attack targets.

2.2. Proposed Solution

In an increasingly digital world, ensuring the security of websites is paramount. Identifying vulnerabilities is a crucial step towards strengthening the security posture of websites. Kali Linux, a powerful penetration testing platform, offers a range of tools, including Metasploit, Nmap, and Telnet, that can aid in vulnerability identification.

1. **Nmap:** Network Scanning and Enumeration Nmap is a versatile network scanning tool that helps identify open ports, services, and potential vulnerabilities. By performing comprehensive port scans, Nmap allows security professionals to assess the network landscape. Through techniques such as OS fingerprinting and service version detection, Nmap enables the identification of outdated software versions and potential entry points for attackers. By running regular scans, administrators can ensure that all ports are properly secured, reducing the risk of unauthorized access.
2. **Metasploit Framework:** Exploiting Vulnerabilities Metasploit is a widely used exploitation framework that aids in the identification and exploitation of vulnerabilities. With an extensive database of exploits, Metasploit allows security professionals to simulate attacks and assess the system's response. By identifying vulnerable software versions, misconfigurations, or weak passwords, Metasploit enables security teams to proactively patch vulnerabilities before malicious actors can exploit them. Regular vulnerability assessments using Metasploit contribute to maintaining an up-to-date defence posture.
3. **Telnet:** Remote Access Vulnerability Telnet is a network protocol that allows remote access to servers. However, its use poses significant security risks as it transmits data in plaintext, making it susceptible to eavesdropping and interception. Kali Linux provides Telnet clients to test remote servers for this vulnerability. By attempting to connect to remote systems via Telnet, security professionals can assess whether this service is active and flag it as a potential security weakness. It is recommended to disable Telnet in favour of secure alternatives such as SSH (Secure Shell).
4. **OpenVAS:** Vulnerability Scanning and Management OpenVAS (Open Vulnerability Assessment System) is a powerful vulnerability scanner available in Kali Linux. It helps identify known vulnerabilities in target systems by scanning for outdated software, weak configurations, and common misconfigurations. OpenVAS automates the process of vulnerability management by providing detailed reports and recommendations for remediation. By regularly running vulnerability scans using OpenVAS, website administrators can prioritize and address potential vulnerabilities promptly, reducing the window of opportunity for attackers.

Ensuring the security of websites requires a proactive approach that involves identifying vulnerabilities and promptly addressing them. Kali Linux tools such as Nmap, Metasploit, Telnet, and OpenVAS provide powerful capabilities for vulnerability identification and assessment. By utilizing these tools effectively, website administrators can gain insights into potential weaknesses, prioritize remediation efforts, and strengthen their overall security posture.

3. Theoretical Analysis

3.1. Block Diagram

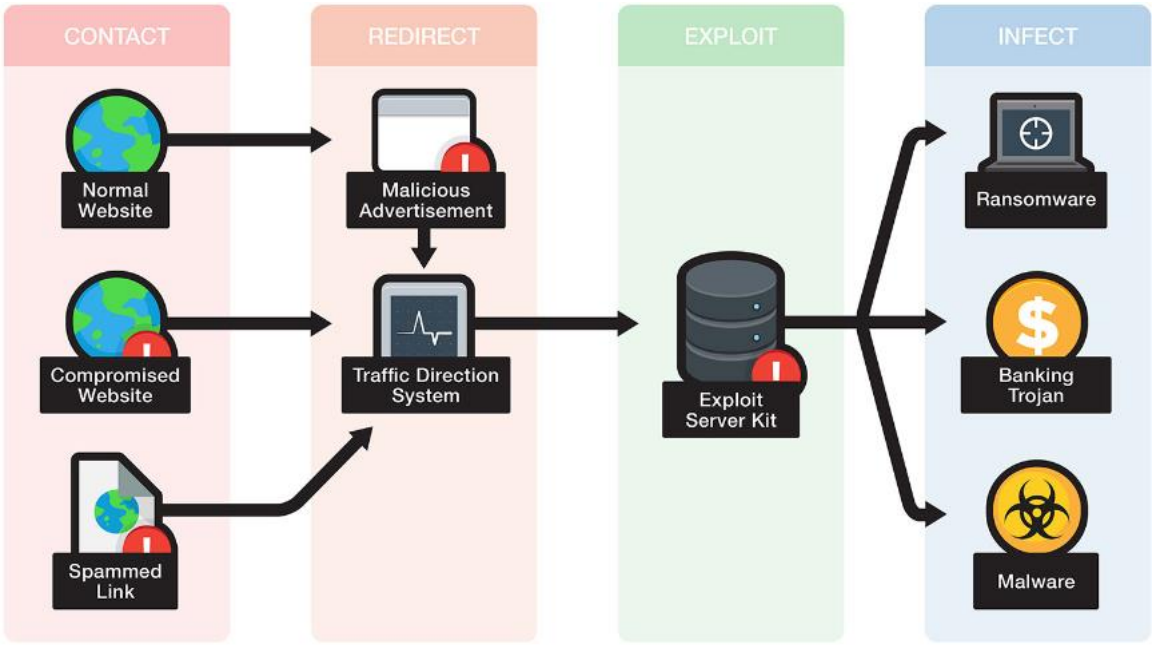


Figure 3.1.1: Stages of an exploit kit infection

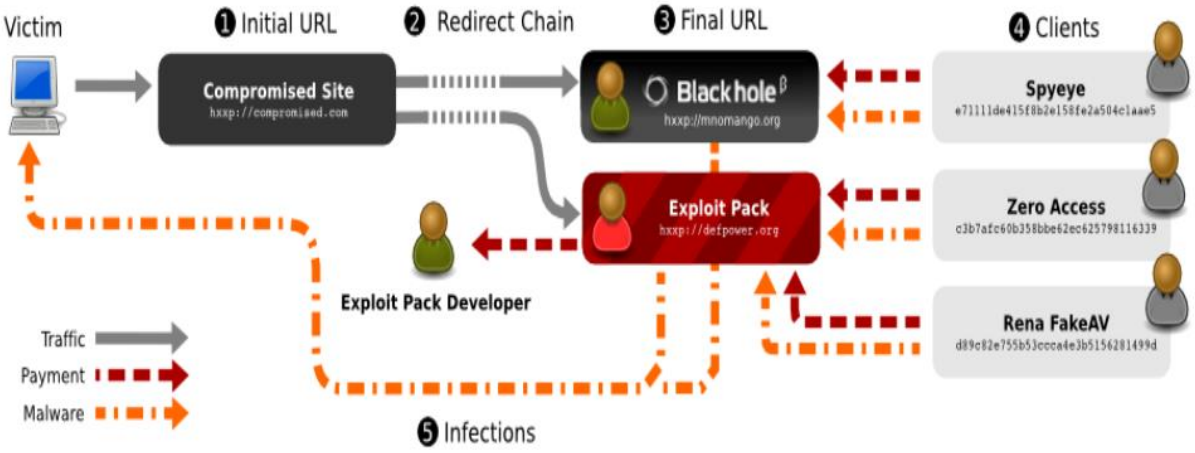


Figure 3.1.2: Vulnerabilities and Exploits

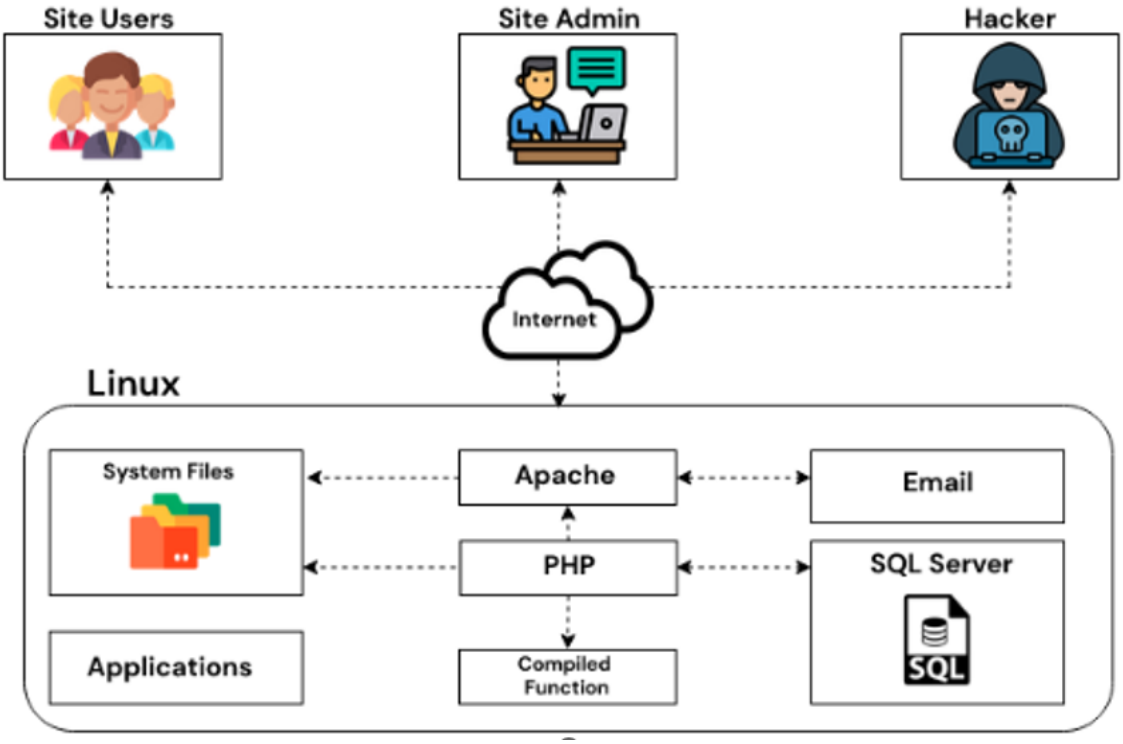


Figure 3.1.3: Web Server Hacking

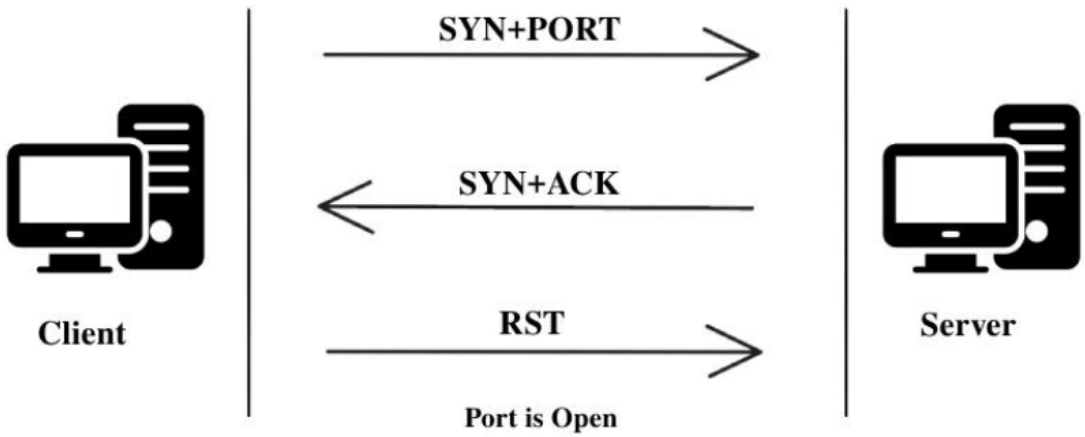


Figure 3.1.4: Port scanning attack

3.2 Hardware and Software Designing

Hardware

Performing vulnerability exploitation involves a different set of considerations compared to vulnerability scanning. Exploitation often requires more specialized hardware and may involve additional factors to consider. Here are some hardware-related aspects to consider when engaging in vulnerability exploitation:

1. **Testing Environment:** To conduct vulnerability exploitation safely, it is recommended to set up a controlled and isolated testing environment. This environment typically includes a separate network segment or virtualized infrastructure where the exploitation activities can be performed without affecting production systems. The hardware required for this environment may include servers or virtual machines to simulate the target systems.
2. **Powerful Workstation:** A powerful workstation is essential for running the necessary exploitation tools, virtualization software, and operating systems. This includes having sufficient CPU power, memory (RAM), and storage to handle resource-intensive activities such as running virtual machines, multiple instances of tools, and handling large amounts of data.
3. **Network Interface:** A network interface card (NIC) is needed to connect the workstation to the testing environment or target systems. The NIC should support the required network protocols and provide reliable connectivity.
4. **Capture/Analysis Equipment:** In some cases, vulnerability exploitation may involve capturing network traffic for analysis or reverse engineering purposes. Specialized hardware such as network sniffers or packet capture devices can aid in capturing and analyzing network traffic.
5. **Custom Hardware/Devices:** Certain vulnerabilities may require specific hardware or devices to exploit successfully. For example, hardware-based exploits targeting specific devices, USB-based attacks, or attacks against embedded systems. These scenarios may require the use of specific hardware components or devices relevant to the target systems.
6. **Collaboration Tools:** Depending on the scale and complexity of vulnerability exploitation, collaboration tools such as remote desktop software, video conferencing tools, or virtual collaboration platforms might be necessary for team coordination, knowledge sharing, and real-time collaboration during the exploitation process.
7. **Processor (CPU):** A multi-core processor with a higher clock speed is beneficial for faster scanning and handling resource-intensive tasks. Vulnerability scanning tools often utilize the CPU extensively.
8. **Memory (RAM):** Sufficient RAM is essential to accommodate the memory requirements of the scanning tools and efficiently store the results. The more RAM available, the better the scanning performance, especially when dealing with large networks or multiple concurrent scans.
9. **Storage:** Adequate storage space is necessary to store the vulnerability scanning tools, their associated databases, and the generated reports. Additionally, enough disk space is needed to store the results of the scans, especially when performing frequent or large-scale scans.

10. **Operating System:** The choice of operating system depends on the vulnerability scanning tool being used. Common options include Windows, Linux, or specialized security-focused operating systems like Kali Linux, which comes preloaded with numerous vulnerability scanning tools.

11. **Network Bandwidth:** While not directly related to hardware, having sufficient network bandwidth is crucial for efficient vulnerability scanning. A higher bandwidth connection allows faster communication between the scanning system and the devices being scanned, reducing scanning time and potential network congestion.

Software

1. Oracle VM VirtualBox

Oracle VM VirtualBox is a powerful and popular open-source virtualization software developed by Oracle Corporation. It enables users to create and manage virtual machines on their host systems.

- **Virtualization Platform:** Oracle VM VirtualBox allows the creation and management of virtual machines, which are isolated and independent environments that can run multiple operating systems simultaneously on a single physical machine. This virtualization platform provides a safe and flexible way to test software, run different operating systems, or create virtualized environments.
- **Cross-Platform Support:** Oracle VM VirtualBox is compatible with various host operating systems, including Windows, macOS, Linux, and Oracle Solaris. This cross-platform support allows users to install VirtualBox on their preferred host operating system and create virtual machines running different guest operating systems.
- **Guest OS Support:** VirtualBox supports a wide range of guest operating systems, including Windows (all versions), Linux distributions, macOS, Oracle Solaris, BSD, and more. It provides compatibility and virtualization support for different operating systems, enabling users to run various software configurations within virtual machines.
- **Snapshot and Clone:** VirtualBox offers snapshot and clone functionality, allowing users to take snapshots of a virtual machine's state or create clones of virtual machines. Snapshots enable users to capture the state of a virtual machine at a specific point in time, making it easy to revert to a previous state if needed. Cloning allows users to create identical copies of virtual machines for backup or experimentation purposes.
- **Networking Options:** VirtualBox provides flexible networking options to configure network connectivity for virtual machines. It supports various networking modes, including NAT (Network Address Translation), Bridged, Internal, and Host-Only networking, allowing virtual machines to communicate with the host system, each other, and the external network.
- **USB Device Support:** VirtualBox enables the connection of USB devices to virtual machines. Users can attach USB peripherals to virtual machines, allowing them to utilize USB devices within the guest operating system. This feature is useful for testing software with specific USB dependencies or accessing USB devices from virtualized environments.
- **Extension Packs:** Oracle VM VirtualBox offers optional Extension Packs that enhance its functionality. These packs provide additional features, such as support for USB 2.0/3.0 devices, Remote Desktop Protocol (RDP) server, PXE booting for Intel network cards, and more. Users can download and install the Extension Packs to extend the capabilities of VirtualBox.

- **Command-Line Interface:** VirtualBox provides a command-line interface (CLI) that allows users to manage virtual machines, control their configurations, and automate virtualization tasks. The CLI offers extensive control and customization options, making it suitable for scripting, automation, and integration with other tools or systems.

Oracle VM VirtualBox is widely used by individuals, software developers, system administrators, and organizations for a variety of purposes, including software testing, virtualized environments, development environments, security testing, and more. Its user-friendly interface, extensive guest OS support, and flexibility make it a popular choice for virtualization needs.

2. Kali Linux

Kali Linux is a specialized Linux distribution designed for advanced penetration testing, ethical hacking, and security auditing. It is developed and maintained by Offensive Security, a leading provider of security training and certification.

- **Penetration Testing Tools:** Kali Linux comes preloaded with a vast collection of over 600 penetration testing tools and utilities. These tools cover various aspects of security assessment, including network scanning, vulnerability assessment, web application testing, wireless attacks, password cracking, forensics, and more. Some popular tools included in Kali Linux are Metasploit Framework, Nmap, Wireshark, Burp Suite, John the Ripper, Aircrack-ng, and Hydra.
- **Security-focused Operating System:** Kali Linux is specifically designed to meet the needs of security professionals, penetration testers, and ethical hackers. It provides a robust and secure environment with built-in tools and configurations optimized for security testing and offensive security operations.
- **Customization and Flexibility:** Kali Linux allows extensive customization to tailor the operating system according to specific requirements. Users can add or remove tools, install additional packages, and customize the desktop environment to suit their preferences and workflows.
- **Regular Updates:** Kali Linux receives regular updates, ensuring that the included tools and packages are up to date. The updates include security patches, bug fixes, and the addition of new tools or features. This ensures that security professionals have access to the latest tools and vulnerabilities to effectively assess and secure systems.
- **Live Boot Capability:** Kali Linux can be run as a live system directly from a bootable USB or DVD without installing it on the host system. This feature allows security professionals to use Kali Linux on different machines without leaving traces on the host system. It also enables quick deployment during security assessments or forensics investigations.
- **Documentation and Community Support:** Kali Linux provides comprehensive documentation, including user guides, tutorials, and a well-maintained community forum. The documentation assists users in understanding the tools, methodologies, and best practices for conducting security assessments and penetration testing. The active community support allows users to seek help, share knowledge, and collaborate with other security professionals.
- **Compatibility and Integration:** Kali Linux is compatible with a wide range of hardware platforms and architectures. It can be installed on desktops, laptops, virtual machines, and even ARM-based devices like Raspberry Pi. Kali Linux can also be integrated with other security tools and platforms to create customized security solutions and workflows.

- **Ethical Hacking Training and Certifications:** Offensive Security, the organization behind Kali Linux, offers training courses and certifications in ethical hacking and penetration testing. These courses, such as the Offensive Security Certified Professional (OSCP) certification, provide in-depth knowledge and practical skills in using Kali Linux and performing real-world security assessments.

Kali Linux is widely used by security professionals, penetration testers, system administrators, and individuals interested in learning and practicing ethical hacking techniques. Its comprehensive toolset, focus on security testing, and active community support make it a popular and powerful platform for conducting security assessments and securing systems.

3. Metasploit

Metasploit is a popular and powerful penetration testing framework developed by Rapid7. It provides a comprehensive collection of tools and exploits for penetration testing, vulnerability assessment, and security research

- **Exploit Development:** Metasploit allows security professionals to develop, test, and execute exploits against various software vulnerabilities. It includes a wide range of exploit modules that can target common vulnerabilities like buffer overflows, SQL injection, remote code execution, and more. These modules simplify the process of discovering and exploiting vulnerabilities in target systems.
- **Post-Exploitation:** Once a system has been compromised, Metasploit provides post-exploitation modules that enable further exploration, privilege escalation, and lateral movement within the compromised network. These modules help security professionals gather information, extract credentials, pivot to other systems, and maintain persistence within the compromised infrastructure.
- **Vulnerability Scanning:** Metasploit integrates with vulnerability scanners such as Nexpose, Rapid7's vulnerability management solution, to identify and exploit vulnerabilities in target systems. It can import vulnerability scan results and automatically generate exploit modules or payloads to target the identified vulnerabilities.
- **Payloads:** Metasploit offers a wide range of payload options that allow security professionals to deliver and execute specific actions on compromised systems. These payloads can be customized to suit specific requirements, such as remote shell access, file upload/download, keylogging, and more. The flexibility of payloads makes Metasploit a versatile tool for both offensive and defensive security operations.
- **Resource Scripting:** Metasploit allows users to create and automate complex attack scenarios using resource scripts. These scripts combine multiple Metasploit commands and modules, allowing users to define and execute a series of actions in a streamlined manner. Resource scripts are particularly useful for repetitive tasks, custom exploitation scenarios, or multi-step attack simulations.
- **Social Engineering:** Metasploit includes modules for conducting social engineering attacks, such as phishing campaigns or email-based attacks. These modules help security professionals test the effectiveness of security awareness training programs and simulate real-world social engineering scenarios.
- **Exploit Development Framework:** Metasploit provides an exploit development framework that allows users to create their own custom exploits or modify existing modules. This

framework provides the necessary tools and resources for security researchers and developers to analyse vulnerabilities, develop proof-of-concept exploits, and contribute to the Metasploit community.

- **Metasploit Community and Updates:** Metasploit has a vibrant and active community of security professionals, researchers, and enthusiasts. The community contributes to the development of new exploits, modules, and enhancements to the framework. Metasploit receives regular updates and releases, ensuring that the framework stays up to date with the latest vulnerabilities, exploits, and security techniques.

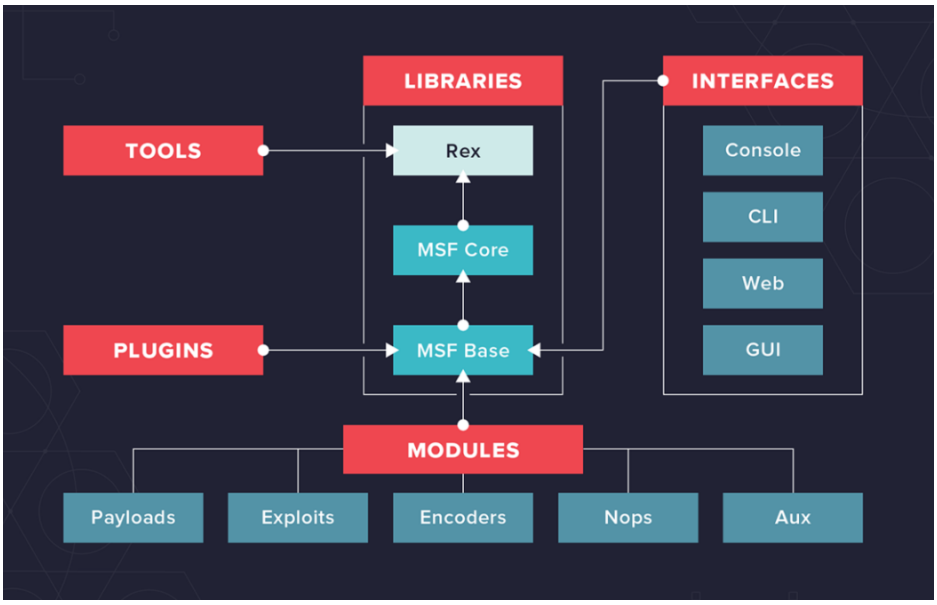


Figure 3.2.1: Typical Metasploit Architecture

4. Nmap

Nmap (Network Mapper) is a widely used open-source network scanning and reconnaissance tool. It is designed to discover and analyze network hosts, services, and vulnerabilities.

- **Host Discovery:** Nmap can scan and discover hosts on a network by sending ICMP echo requests (ping) or performing more advanced host discovery techniques like ARP requests, TCP/UDP probes, or even DNS queries. This helps identify live hosts on a network and determine their availability.
- **Port Scanning:** Nmap is known for its powerful port scanning capabilities. It can scan for open ports on target hosts to determine which services are running and listening for connections. Nmap supports various scanning techniques, including TCP SYN scan, TCP connect scan, UDP scan, and more. By identifying open ports, security professionals can assess potential attack vectors and identify services that may have vulnerabilities.
- **Service Version Detection:** Nmap can perform service version detection by analysing the response of a target's open ports. It attempts to identify the specific version of the service running on each open port. This information helps security professionals understand the

software versions in use and assess the potential security risks associated with outdated or vulnerable service versions.

- **Operating System Detection:** Nmap can also determine the operating system of a target host based on various network characteristics and responses. It analyzes network packets, TTL values, TCP/IP stack behavior, and other indicators to make an educated guess about the underlying operating system. This feature helps in network mapping and understanding the target environment.
- **Scripting Engine:** Nmap incorporates a flexible scripting engine called Nmap Scripting Engine (NSE). NSE allows users to write and execute custom scripts to automate tasks, perform advanced network scanning, gather additional information, or check for specific vulnerabilities. The NSE library includes a wide range of pre-built scripts that can be used for tasks like vulnerability scanning, brute-forcing credentials, or performing specific network tests.
- **Output Formatting:** Nmap offers various output formats to present scan results in a clear and organized manner. Users can choose between interactive console output, text files, XML files, or even a Zenmap graphical interface for result visualization. This flexibility allows users to process and analyse scan results efficiently.
- **Extensibility and Integration:** Nmap is extensible and can be integrated with other security tools and frameworks. It provides APIs and libraries that enable developers to incorporate Nmap functionality into their own applications or scripts. Additionally, Nmap can work alongside vulnerability scanners, such as OpenVAS or Nessus, to provide comprehensive security assessments.
- **Continuous Development and Community Support:** Nmap has an active community of developers and security professionals who contribute to its development and maintenance. Regular updates and new releases ensure that Nmap remains up to date with the latest network scanning techniques, improvements, and bug fixes.

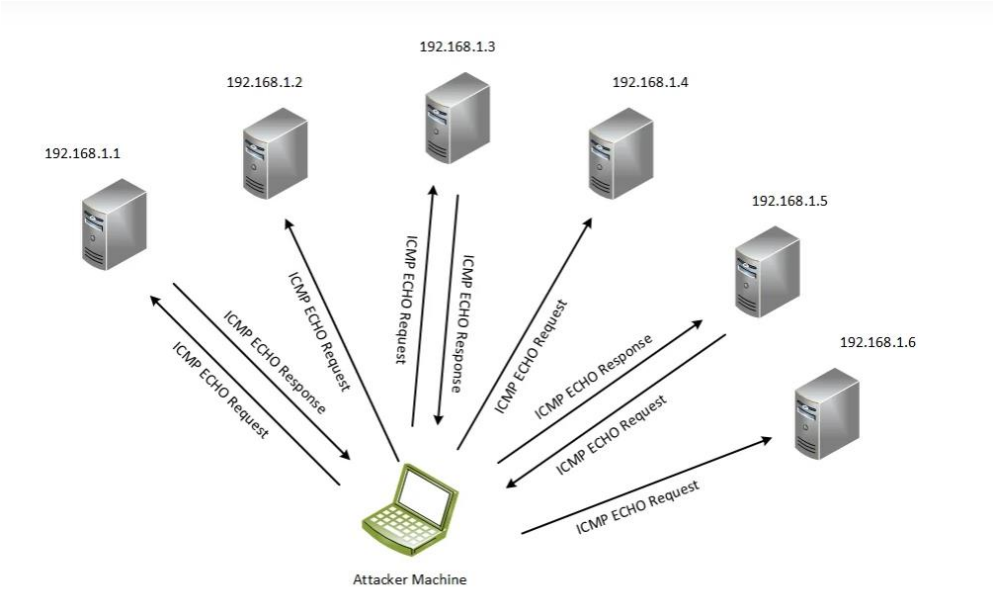


Figure 3.2.2: Scanning and reconnaissance using Nmap

5. Telnet

Telnet is a network protocol and a command-line tool used for remote access to network devices or systems. It provides a way to establish a virtual terminal session with a remote host over a network connection.

- **Remote Access:** Telnet allows users to remotely access and manage devices or systems that support the Telnet protocol. By establishing a Telnet session, users can interact with the remote device as if they were physically present, providing a command-line interface for administration, configuration, or troubleshooting purposes.
- **Terminal Emulation:** Telnet provides terminal emulation, which means it allows a local terminal or computer to emulate the behavior of a physical terminal connected to the remote system. This enables users to execute commands and receive responses from the remote host, just as if they were using a direct console connection.
- **Text-based Communication:** Telnet uses a text-based communication method, where users type commands or input data on their local system, and the text is transmitted to the remote host. The remote host processes the commands and sends back responses or output, which is displayed on the local system. This makes Telnet suitable for managing devices or systems that have a command-line interface.
- **Port 23:** Telnet typically operates on port 23, which is the default port for Telnet communication. When connecting to a remote host via Telnet, users specify the IP address or hostname of the remote system along with the port number, if different from the default.
- **Lack of Encryption:** It's important to note that Telnet sends all communication, including usernames, passwords, and data, in plain text. This means that the data transmitted over a Telnet connection is not encrypted and can be intercepted by anyone with access to the network. Due to this security vulnerability, Telnet is considered insecure for transmitting sensitive information and is often replaced by more secure protocols like SSH (Secure Shell).
- **Scripting and Automation:** Telnet can be automated and scripted to perform repetitive tasks or automate configurations. By writing scripts that send predefined commands and process responses, users can automate tasks such as configuration backups, firmware upgrades, or device monitoring.
- **Troubleshooting and Diagnostics:** Telnet can be used for troubleshooting network connectivity or testing services on remote hosts. It allows users to manually connect to a specific port on a remote system and verify if the service is responding. This can help in diagnosing network issues, checking if a service is available, or troubleshooting communication problems.

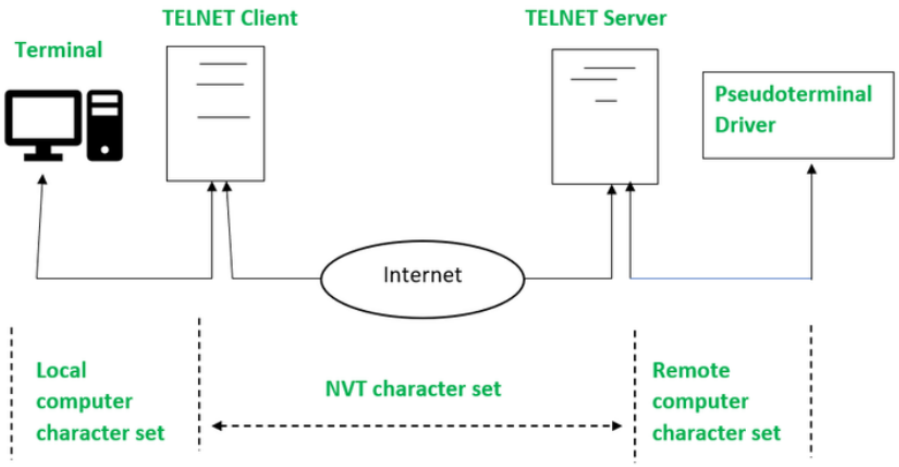


Figure 3.2.3: Network Virtual Terminal (NVT) in Telnet

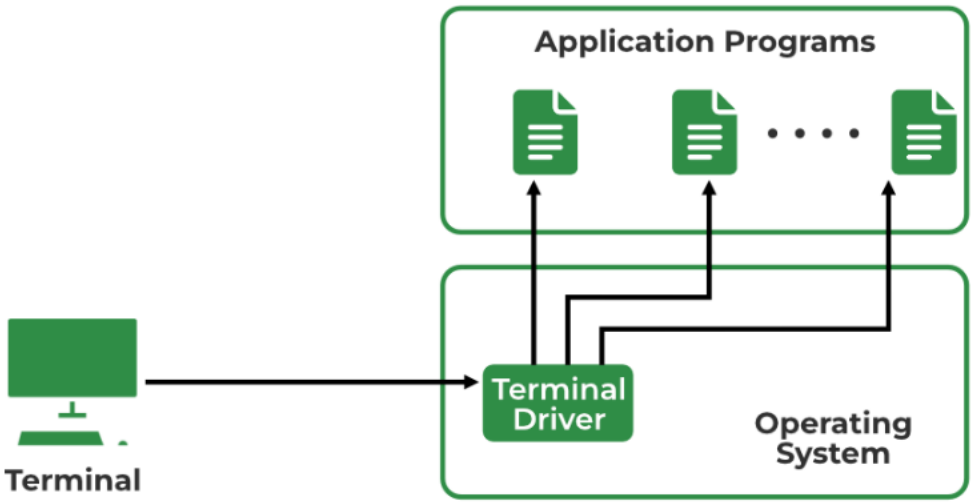


Figure 3.2.4: Local login

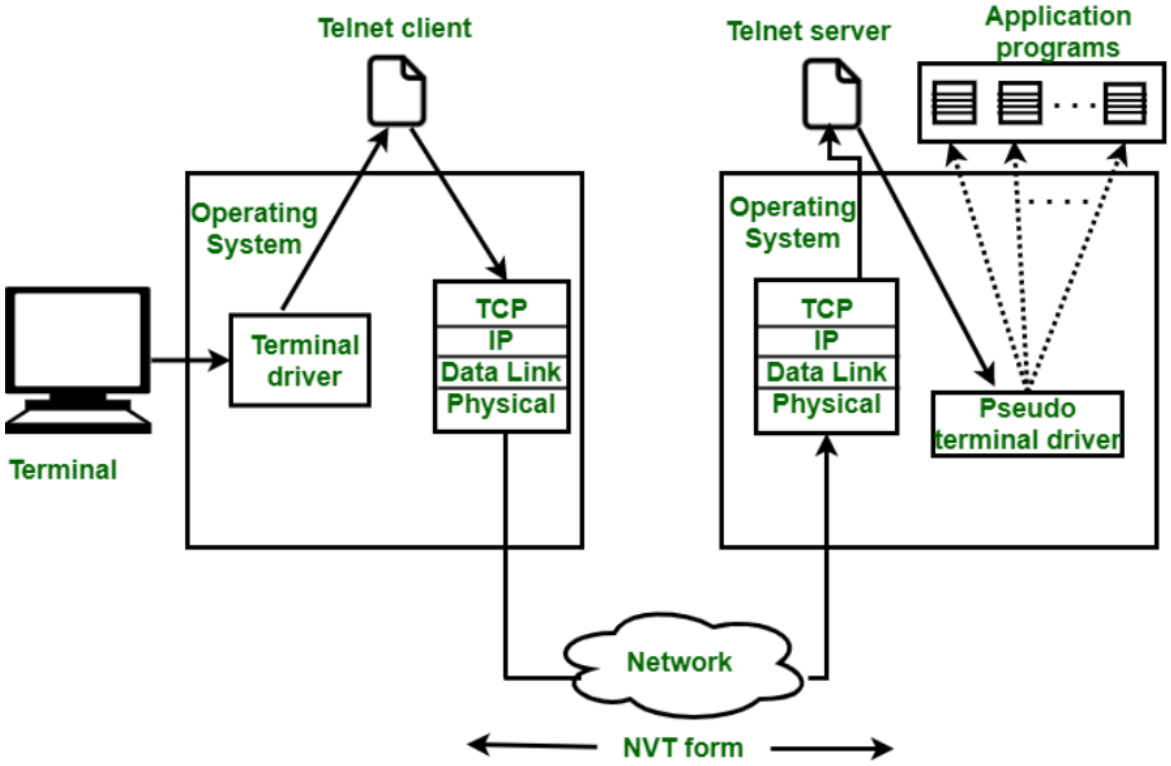


Figure 3.2.5: Remote login

6. Metasploitable 2 (Practice Website)

Metasploitable 2 is a purposely vulnerable virtual machine that is designed for practicing and learning penetration testing techniques using the Metasploit Framework. It is a pre-configured virtual machine that contains numerous vulnerabilities and misconfigurations, making it an ideal target for security testing and exploitation exercises.

- **Purposefully Vulnerable:** Metasploitable 2 is intentionally created with vulnerable services and weak configurations. It simulates a realistic target environment with a wide range of security vulnerabilities, including outdated software versions, default credentials, misconfigured services, and more. These vulnerabilities allow security professionals and students to practice exploiting and securing systems in a controlled environment.
- **Metasploit Integration:** Metasploitable 2 is designed to be used in conjunction with the Metasploit Framework, a powerful penetration testing tool. The vulnerable services and configurations in Metasploitable 2 can be exploited using various Metasploit modules, including exploits, payloads, auxiliary modules, and post-exploitation tools. This integration allows users to gain hands-on experience in using Metasploit for exploitation and post-exploitation activities.
- **Diverse Range of Vulnerabilities:** Metasploitable 2 encompasses a wide variety of vulnerabilities, covering multiple services and technologies. It includes vulnerable versions of popular software, such as web servers (Apache, Tomcat), database servers (MySQL, PostgreSQL), FTP servers (vsftpd, ProFTPD), and more. These vulnerabilities include remote code execution, buffer overflows, SQL injection, command injection, and other common security flaws.
- **Realistic Simulation:** Metasploitable 2 aims to simulate a real-world vulnerable system, providing users with a practical learning experience. It mimics the type of security weaknesses and misconfigurations that may be found in actual environments, helping users understand the impact of these vulnerabilities and the techniques used to exploit them.
- **Educational Tool:** Metasploitable 2 is primarily used as an educational tool for security professionals, students, and enthusiasts. It facilitates hands-on learning and practical exercises in a safe and controlled environment. By exploiting the vulnerabilities in Metasploitable 2, users can enhance their understanding of common attack vectors, improve their exploit development skills, and gain knowledge about effective mitigation and security best practices.

4. Experimental Investigation

We can diagnose email errors through the Linux command line using the TELNET command. We can communicate back and forth with the receiving server with TELNET to see why the email is being rejected to verify email bounce errors or to check if the server email ports are working.

TELNET to an Email Server

We can communicate with a server using TELNET to find the exact error on the server in question. In this case, we will see if we can replicate the error when emails sent from name@example.com are bouncing sending to notexample.com.

Basic syntax:

Command Domain or IP Port # telnet example.com 25

We can type HELO localhost to receive a response from the receiving server. This will display the server's name and IP. We can also type EHLO localhost to receive more information for the receiving server. This shows the extended server response with server name IP and some supported settings.

Now that we verified that the server is communicating back and forth, next, we ask the server to check the sending email address. Type MAIL FROM: <name@example.com> FROM: SIZE=100 and hit enter.

We enter the email subject by typing Subject: Your email subject and press Enter. Replace Your email subject with the desired subject line.

Press Enter twice to leave a blank line, and then enter the body of the email. You can include multiple lines of text as the email content.

To end the email message, type a period (.) on a new line and press Enter.

Finally, type the following command to quit the Telnet session:

5. Flowchart

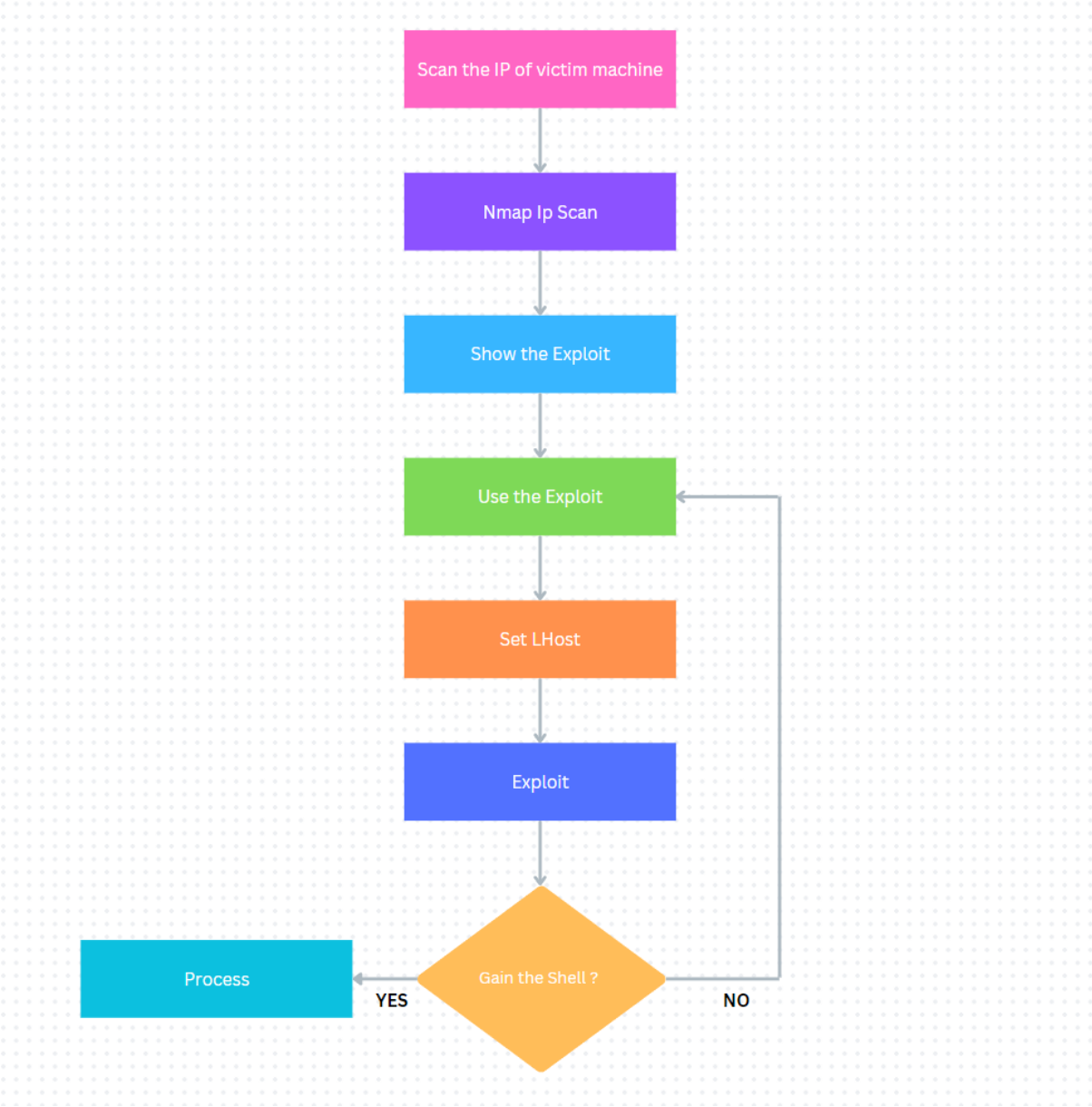


Figure 5.1: Flowchart of vulnerability exploit process

6. Result

6.1. Searching for open ports in the main website

1. **Nslookup:** The nslookup command queries internet domain name servers in two modes. Interactive mode allows you to query name servers for information about various hosts and domains, or to print a list of the hosts in a domain. In noninteractive mode, the names and requested information are printed for a specified host or domain.

```
┌──(root💀 kali)-[~]
└─# nslookup www.palashbitan.com
Server:         192.168.0.1
Address:        192.168.0.1#53

Non-authoritative answer:
www.palashbitan.com     canonical name = palashbitan.com.
Name:   palashbitan.com
Address: 135.125.153.179


┌──(root💀 kali)-[~]
└─#
```

We get the Ip address of www.palashbitan.com : **135.125.153.179**

2. **Nmap:** Nmap is a network scanner created by Gordon Lyon. Nmap is used to discover hosts and services on a computer network by sending packets and analyzing the responses. Nmap provides a number of features for probing computer networks, including host discovery and service and operating system detection

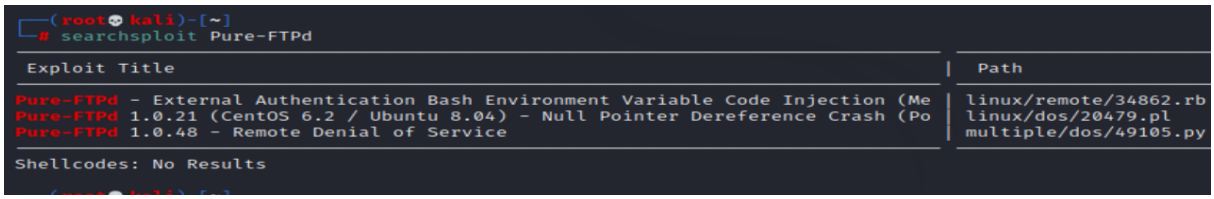
```
┌──(root💀 kali)-[~]
└─# nmap -sV 135.125.153.179
Starting Nmap 7.92 ( https://nmap.org ) at 2023-06-27 10:17 EDT
Nmap scan report for ns2.server333.iseencloud.com (135.125.153.179)
Host is up (0.24s latency).
Not shown: 979 filtered tcp ports (no-response)
PORT      STATE  SERVICE  VERSION
21/tcp    open   ftp      Pure-FTPd
22/tcp    closed ssh
53/tcp    open   domain   PowerDNS Authoritative Server 4.7.3
80/tcp    open   http     LiteSpeed httpd
110/tcp   open   pop3     Dovecot pop3d
143/tcp   open   imap     Dovecot imapd
443/tcp   open   ssl/http LiteSpeed httpd
587/tcp   open   smtp     Exim smtpd 4.96
993/tcp   open   imaps?
995/tcp   open   pop3s?
3306/tcp  open   mysql    MySQL 5.5.5-10.3.39-MariaDB
49154/tcp closed unknown
49161/tcp closed unknown
49176/tcp closed unknown
50002/tcp closed iiimsf
50003/tcp closed unknown
52673/tcp closed unknown
54328/tcp closed unknown
63331/tcp closed unknown
64623/tcp closed unknown
65129/tcp closed unknown
Service Info: Host: server333.iseencloud.com

Service detection performed. Please report any incorrect results at https:/
Nmap done: 1 IP address (1 host up) scanned in 87.60 seconds
```

Open Ports

PORT	STATE	SERVICE	VERSION
21/tcp	open	ftp	Pure-FTPd
53/tcp	open	domain	PowerDNS Authoritative Server 4.7.3
80/tcp	open	http	LiteSpeed httpd
110/tcp	open	pop3	Dovecot pop3d
143/tcp	open	imap	Dovecot imapd
443/tcp	open	ssl/http	LiteSpeed httpd
587/tcp	open	smtp	Exim smtpd 4.96
993/tcp	open	imaps?	
995/tcp	open	pop3s?	
3306/tcp	open	mysql	MySQL 5.5.5-10.3.39-MariaDB

3. **Searchsploit:** Searchsploit is a command-line search tool for Exploit-DB that allows you to take a copy of the Exploit Database with you. Searchsploit is included in the Exploit Database repository on GitHub. Searchsploit is very useful for security assessments when you don't have Internet access because it gives you the power to perform detailed offline searches for exploits in the saved Exploit-DB.



We used searchsploit on Pure-FTPd

4. **msfconsole:** MSFconsole provides a command line interface to access and work with the Metasploit Framework. The MSFconsole is the most commonly used interface to work with the Metasploit Framework. The console lets you do things like scan targets, exploit vulnerabilities, and collect data.

5. Search: The msfconsole includes an extensive regular-expression based search functionality. If you have a general idea of what you are looking for, you can search for it via search. In the output below, a search is being made for MS Bulletin MS09-011. The search function will locate this string within the module names, descriptions, references, etc.

```
msf6 > search Pure-FTPd

Matching Modules
================

   #  Name                                      Disclosure Date  Rank       Check  Description
   -  ----                                      ---------------  ----       -----  -----------
   0  exploit/multi/ftp/pureftpd_bash_env_exec  2014-09-24       excellent  Yes    Pure-FTPd External Authentication
 Bash Environment Variable Code Injection (Shellshock)


Interact with a module by name or index. For example info 0, use 0 or use exploit/multi/ftp/pureftpd_bash_env_exec
```

We used search on Pure-FTPd

```
msf6 > use exploit/multi/ftp/pureftpd_bash_env_exec
[*] No payload configured, defaulting to linux/x86/meterpreter/reverse_tcp
msf6 exploit(multi/ftp/pureftpd_bash_env_exec) > options

Module options (exploit/multi/ftp/pureftpd_bash_env_exec):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   RHOSTS                     yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki
                                        /Using-Metasploit
   RPATH     /bin             yes       Target PATH for binaries used by the CmdStager
   RPORT     21               yes       The target port (TCP)
   SRVHOST   0.0.0.0          yes       The local host or network interface to listen on. This must be an address o
                                        n the local machine or 0.0.0.0 to listen on all addresses.
   SRVPORT   8080             yes       The local port to listen on.
   SSL       false            no        Negotiate SSL for incoming connections
   SSLCert                    no        Path to a custom SSL certificate (default is randomly generated)
   URIPATH                    no        The URI to use for this exploit (default is random)

Payload options (linux/x86/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.0.2.15        yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Linux x86
```

We use **exploit/multi/ftp/pureftpd_bash_env_exec** for further exploitation

5. telnet: Telnet is a network protocol and application that allows a user to establish a remote terminal connection to a computer or network device over a TCP/IP network. It was one of the earliest protocols used for remote access and remains in use today, although it has been largely superseded by more secure alternatives. Telnet operates on the client-server model, where the client establishes a connection to the server using the Telnet protocol. Once connected, the client can send commands and receive responses from the server, effectively controlling the remote device as if it were a local terminal. Telnet was designed to be a simple and lightweight protocol, providing basic terminal emulation and text-based communication. It allows users to access and manage remote systems, such as servers, routers, switches, and other network devices, over a network connection.

Open Ports exploited

i) Pure-FTPd: Pure-FTPd is a free and open-source FTP (File Transfer Protocol) server software that allows users to transfer files between computers over a network. It is available for various operating systems, including Linux, Unix, macOS, and Windows. **Could not hack into Pure-FTPd**

```
msf6 exploit(multi/ftp/pureftpd_bash_env_exec) > telnet 135.125.153.179 21
[*] exec: telnet 135.125.153.179 21

Trying 135.125.153.179 ...
Connected to 135.125.153.179.
Escape character is '^]'.
220---------- Welcome to Pure-FTPd [privsep] [TLS] ----------
220-You are user number 1 of 50 allowed.
220-Local time is now 17:45. Server port: 21.
220-This is a private system - No anonymous login
220-IPv6 connections are also welcome on this server.
220 You will be disconnected after 25 minutes of inactivity.
VERSION
530 You aren't logged in
```

ii) LiteSpeed hhtpd: LiteSpeed HTTP Server (LSWS) is a high-performance web server software that delivers excellent speed and scalability for serving websites and web applications. It is a commercial product developed by LiteSpeed Technologies and is designed to be a drop-in replacement for the popular Apache HTTP Server while providing significant performance improvements. Followed the same procedure for LiteSpeed httpd. **Could briefly connect to it**.

```
msf6 auxiliary(scanner/http/litespeed_source_disclosure) > telnet 135.125.153.179 80
[*] exec: telnet 135.125.153.179 80

Trying 135.125.153.179 ...
Connected to 135.125.153.179.
Escape character is '^]'.
Connection closed by foreign host.
```

iii) Dovecot: Dovecot is a popular open-source email server software that provides both the Internet Message Access Protocol (IMAP) and the Post Office Protocol (POP) for retrieving emails from a server. It is designed to be secure, scalable, and efficient, making it a reliable choice for handling email storage and access. **Connection Successful!**

```
msf6 exploit(linux/smtp/exim4_dovecot_exec) > telnet 135.125.153.179 110
[*] exec: telnet 135.125.153.179 110

Trying 135.125.153.179 ...
Connected to 135.125.153.179.
Escape character is '^]'.
+OK Dovecot ready.
```

```
msf6 exploit(linux/smtp/exim4_dovecot_exec) > telnet 135.125.153.179 143
[*] exec: telnet 135.125.153.179 143

Trying 135.125.153.179...
Connected to 135.125.153.179.
Escape character is '^]'.
* OK [CAPABILITY IMAP4rev1 SASL-IR LOGIN-REFERRALS ID ENABLE IDLE NAMESPACE LITERAL+ STARTTLS AUTH=PLAIN AUTH=LOGIN
] Dovecot ready.
```

iv) Exim: Exim is a popular open-source mail transfer agent (MTA) that is widely used for routing and delivering email messages on Unix-like operating systems. It is known for its flexibility, scalability, and extensive feature set, making it a versatile choice for handling email services. **Made a successful connection!**

```
┌──(root💀kali)-[~]
└─# telnet 135.125.153.179 587
Trying 135.125.153.179 ...
Connected to 135.125.153.179.
Escape character is '^]'.
220-server333.iseencloud.com ESMTP Exim 4.96 #2 Tue, 27 Jun 2023 19:33:45 +0000
220-We do not authorize the use of this system to transport unsolicited,
220 and/or bulk e-mail.
HELP
214-Commands supported:
214 AUTH STARTTLS HELO EHLO MAIL RCPT DATA BDAT NOOP QUIT RSET HELP
MAIL
500 unrecognized command
HELO
250 server333.iseencloud.com Hello  [103.240.97.214]
mail from: animonjoyee@gmail.com
550 HELO required before MAIL
HELO mail from: animonjoyee@gmail.com
250 server333.iseencloud.com Hello mail from: animonjoyee@gmail.com [103.240.97.214]
HELO mail to: joyeemonani@gmail.com
250 server333.iseencloud.com Hello mail to: joyeemonani@gmail.com [103.240.97.214]
HELO rtcp to: joyeemonani@gmail.com
250 server333.iseencloud.com Hello rtcp to: joyeemonani@gmail.com [103.240.97.214]
HELO rtcp to: sinhaanimikha@gmail.com
250 server333.iseencloud.com Hello rtcp to: sinhaanimikha@gmail.com [103.240.97.214]
a
500 unrecognized command
AUTH_CRAM_MD5=no
500 unrecognized command
ehlo client.example
250-server333.iseencloud.com Hello client.example [103.240.97.214]
250-SIZE 52428800
250-8BITMIME
250-PIPELINING
250-PIPECONNECT
250-STARTTLS
250 HELP
```

iii) Mysql: MySQL is an open-source relational database management system (RDBMS) that provides a robust and scalable solution for storing, managing, and retrieving data. It is one of the most popular databases used in web applications and is widely supported by various programming languages and frameworks. **Connection Successful!**

```
┌──(root💀kali)-[~]
└─# telnet 135.125.153.179 3306
Trying 135.125.153.179 ...
Connected to 135.125.153.179.
Escape character is '^]'.
Y
5.5.5-10.3.39-MariaDB@@S5Nf;1%c???Fr.*),DN<$<Dmysql_native_password
```

7. Advantages and Disadvantages

Advantages

Vulnerability exploitation and patching are crucial steps in securing web applications. Penetration testing tools, such as those available in Kali Linux, can play a significant role in this process.

- **Identifying vulnerabilities:** Penetration testing tools can scan web applications for known vulnerabilities, including common vulnerabilities such as SQL injection, cross-site scripting (XSS), and insecure direct object references. By actively searching for vulnerabilities, organizations can gain a better understanding of their web application's security weaknesses.

- **Exploiting vulnerabilities:** Once vulnerabilities are identified, penetration testing tools can be used to exploit them. This step involves simulating real-world attack scenarios to determine the potential impact of a vulnerability. By exploiting vulnerabilities in a controlled environment, security professionals can assess the severity and potential consequences of an attack.

- **Assessing security controls:** Penetration testing tools can help evaluate the effectiveness of existing security controls and defences implemented in web applications. By attempting to exploit vulnerabilities, security professionals can identify weaknesses and gaps in the security measures. This enables organizations to take proactive steps to strengthen their defences and mitigate potential risks.

- **Patching vulnerabilities:** After vulnerabilities are identified and exploited, the next step is to patch and fix them. Penetration testing tools assist in this process by providing insights into the specific vulnerabilities that need to be addressed. By prioritizing the identified vulnerabilities based on their severity, organizations can focus their resources on patching the most critical issues first.

- **Validating patches:** Once patches are implemented, penetration testing tools can be used again to validate the effectiveness of the fixes. By retesting the patched web application, organizations can ensure that the vulnerabilities have been adequately addressed and that the fixes have been properly implemented.

- **Continuous improvement:** Penetration testing tools facilitate an iterative approach to security testing. Regularly conducting penetration tests and using tools to exploit vulnerabilities help organizations identify recurring patterns and common weaknesses. This information can then be used to improve development practices, security controls, and ongoing security measures.

Disadvantages

While penetration testing tools, including those in Kali Linux, can be valuable for securing web applications, there are also some potential disadvantages to be aware of:

- **False positives and false negatives:** Penetration testing tools are not infallible and can produce false positives (indicating vulnerabilities that don't actually exist) or false negatives (failing to detect actual vulnerabilities). This can lead to wasted time and resources addressing non-existent issues or overlooking genuine vulnerabilities, respectively. It's important to supplement automated testing with manual analysis and verification.

- **Limited scope:** Penetration testing tools have limitations in terms of the scope of vulnerabilities they can detect. They rely on pre-configured vulnerability databases and may not cover all possible attack vectors or newly emerging vulnerabilities. This means that certain vulnerabilities might go undetected, requiring additional testing techniques or manual analysis to uncover them.

- **Technical expertise required:** Using penetration testing tools effectively requires a certain level of technical expertise and understanding of security concepts. Without proper knowledge and skills, there is a risk of misconfiguring or misusing the tools, leading to inaccurate results or unintended consequences. Adequate training and expertise are crucial for meaningful and effective penetration testing.

- **Limited understanding of business context:** Penetration testing tools primarily focus on identifying technical vulnerabilities but may not consider the broader business context. They may not evaluate the impact of vulnerabilities on business processes, data sensitivity, or compliance requirements. This can result in a lack of alignment between security priorities and business goals.

- **Time and resource constraints:** Penetration testing, especially when performed comprehensively, can be time-consuming and resource-intensive. Organizations might face challenges in allocating sufficient time, budget, and skilled personnel to conduct thorough and regular testing. This can limit the frequency and depth of penetration testing activities, potentially leaving vulnerabilities undetected.

- **Disruption and false alarms:** Penetration testing involves actively probing systems for vulnerabilities, which can sometimes result in service disruptions or false alarms. These disruptions can impact normal business operations and cause inconvenience. Additionally, false alarms can lead to wasted resources and unnecessary panic if not properly managed.

- **Ethical and legal considerations**: Penetration testing must always be conducted with proper authorization and in adherence to legal and ethical guidelines. Using these tools without appropriate permissions can result in legal consequences and damage an organization's reputation. It's essential to obtain consent and follow relevant laws, regulations, and ethical standards when performing penetration testing activities.

8. Applications

Vulnerability exploitation and patching should be applied in various areas to enhance the security of systems and applications. Here are some key areas where vulnerability exploit and patching practices are commonly applied:

1. **Operating Systems:** Vulnerabilities in operating systems can be targeted by malicious actors to gain unauthorized access or control over a system. Regularly applying patches and updates for the operating system helps mitigate these vulnerabilities and strengthens the overall security of the system.
2. **Web Applications:** Web applications are a common target for attackers due to their exposure to the internet. Exploiting vulnerabilities in web applications can lead to data breaches, unauthorized access, and other security incidents. By conducting vulnerability assessments and applying patches to address identified vulnerabilities, the security of web applications can be significantly improved.
3. **Network Infrastructure:** Network devices, such as routers, switches, and firewalls, can have vulnerabilities that expose the network to attacks. Exploiting these vulnerabilities can compromise the network's integrity, availability, and confidentiality. Regularly updating and patching the firmware and software of network devices helps mitigate these vulnerabilities and reduces the risk of unauthorized access or control over the network infrastructure.
4. **Databases:** Databases contain sensitive and valuable data, making them attractive targets for attackers. Vulnerabilities in database management systems (DBMS) can lead to data leaks, data manipulation, and unauthorized access. Patching the DBMS software and applying security updates help protect the integrity and confidentiality of the stored data.
5. **Mobile Applications:** Mobile applications often process and store sensitive user data. Vulnerabilities in mobile applications can expose this data to unauthorized access or misuse. Applying patches and updates to mobile applications, along with conducting rigorous security testing, helps minimize the risk of exploitation and protect user data.
6. **Internet of Things (IoT) Devices:** IoT devices, such as smart home appliances and industrial sensors, are becoming increasingly interconnected. However, many IoT devices lack robust security measures, making them susceptible to attacks. Identifying vulnerabilities in IoT devices and promptly applying patches and firmware updates is crucial to mitigate potential risks and secure these devices.
7. **Third-party Software and Libraries:** Many applications rely on third-party software components and libraries. Vulnerabilities in these components can introduce security weaknesses into the application. Regularly monitoring and updating third-party software and libraries to the latest patched versions help address known vulnerabilities and reduce the attack surface.

9. Conclusion

Vulnerability exploitation and patching are critical components of a comprehensive cybersecurity strategy aimed at enhancing the security posture of systems and applications. By actively identifying and addressing vulnerabilities, organizations can significantly reduce the risk of unauthorized access, data breaches, and other security incidents.

Vulnerability exploitation using penetration testing tools, such as those available in Kali Linux, is an essential practice in understanding the weaknesses and potential attack vectors present in various areas. It allows security professionals to simulate real-world attack scenarios and assess the severity and potential consequences of vulnerabilities. Through targeted exploitation, organizations can gain valuable insights into their systems' vulnerabilities and take proactive measures to remediate them.

The patching process is equally important as it involves the application of updates, fixes, and security patches to address identified vulnerabilities. By promptly applying patches and updates, organizations close security gaps and minimize the window of opportunity for attackers. Regular patch management ensures that systems and applications are protected against known vulnerabilities, reducing the risk of exploitation.

Vulnerability exploitation and patching should always be performed with proper authorization, adherence to legal and ethical guidelines, and a comprehensive understanding of the business context. It is crucial to obtain explicit consent from relevant stakeholders and follow established policies and regulations. This ensures that security testing activities are conducted responsibly and minimize any potential disruptions or legal implications.

In addition to their benefits, vulnerability exploitation and patching also come with potential disadvantages. False positives and false negatives can occur during vulnerability assessments, leading to wasted time and resources addressing non-existent issues or overlooking genuine vulnerabilities. The scope of penetration testing tools may be limited, and they might not cover all possible attack vectors or emerging vulnerabilities. It is important to supplement automated testing with manual analysis and verification to achieve more accurate results.

Technical expertise is necessary for the effective use of penetration testing tools and patch management. Without proper knowledge and skills, there is a risk of misconfiguring or misusing the tools, leading to inaccurate results or unintended consequences. Adequate training and expertise are essential to ensure meaningful and effective vulnerability management.

Time and resource constraints can also pose challenges in vulnerability exploitation and patching. Conducting thorough and regular vulnerability assessments requires allocating sufficient time, budget, and skilled personnel. Organizations need to prioritize and allocate resources appropriately to maintain a proactive and effective security testing program.

Moreover, vulnerability exploitation and patching should not be seen as standalone activities. They should be integrated into a broader cybersecurity strategy that encompasses continuous monitoring, risk assessment, incident response, and employee education. Organizations should adopt a holistic approach to cybersecurity that addresses people, processes, and technology to ensure comprehensive protection against evolving threats.

In conclusion, vulnerability exploitation and patching, when conducted responsibly and comprehensively, significantly contribute to a more secure digital environment. By reducing vulnerabilities, protecting sensitive data, and mitigating the risk of cyber-attacks, organizations can enhance their overall security posture and better safeguard their systems and applications against potential threats.

10. Future Scope

Generally, email is insecure and not the best medium for sending confidential files. But it remains the primary means of communication for most people, especially organizations. SMTP is the most common protocol for sending emails. Furthermore, SMTP is unencrypted. This makes it a vulnerable protocol. SMTP (Simple Mail Transfer Protocol) has several security vulnerabilities and challenges that can be exploited by malicious actors. Some of the security problems associated with SMTP include:

- **Email Spoofing:** SMTP does not provide built-in mechanisms to verify the authenticity of the sender's address. This allows attackers to easily forge the "From" address of an email, leading to email spoofing and phishing attacks.
- **Lack of Encryption:** By default, SMTP operates in plain text, which means that email messages and login credentials can be intercepted and read by attackers during transit. This lack of encryption exposes sensitive information and leaves it vulnerable to eavesdropping.
- **Spam:** SMTP is often abused for sending unsolicited bulk emails (spam). Spammers exploit weaknesses in SMTP servers to send a large volume of unwanted emails, which can overload email systems and disrupt legitimate email communication.
- **Relay Abuse:** SMTP allows relaying, where an email server forwards messages to other servers. Malicious actors can exploit insecurely configured or open relays to send spam or launch denial-of-service (DoS) attacks.
- **Phishing Attacks:** SMTP-based phishing attacks are common, where attackers send fraudulent emails that appear to be from a legitimate source, tricking users into revealing sensitive information such as passwords or financial details.
- **Email Content Modification:** SMTP does not provide inherent protection against email content modification during transit. Attackers can intercept and modify email content, which can lead to the dissemination of false information or malicious attachments.
- **Lack of Sender Verification:** SMTP does not have built-in mechanisms to verify the sender's identity. This makes it difficult to differentiate legitimate senders from malicious ones, increasing the risk of accepting malicious emails.
- **Directory Harvesting Attacks:** Attackers can perform directory harvesting attacks by attempting to gather valid email addresses from an SMTP server. This information can be used for spamming or targeted attacks.

DKIM signature, along with other methods, such as SPF or DMARC, is one of the most common methods of authenticating yourself as the sender of an email message. DKIM, or DomainKeys Identified Mail, is an email authentication method that uses a digital signature to let the receiver of an email know that the message was sent and authorized by the owner of a domain. Once the receiver determines that an email is signed with a valid DKIM signature it can be confirmed that the email's content has not been modified. In most cases, DKIM signatures are not visible to end-users, the validation is done on a server level. If DKIM is used together with DMARC, or SPF you can protect your domain against malicious emails sent from domains impersonating your brand.

Working of DKIM

1) The sender's email server generates a pair of cryptographic keys: a private key and a corresponding public key.
2) The sender's email server uses the private key to create a unique digital signature for each outgoing email message. This signature is inserted into the email headers as a DKIM signature.
3) The public key is published in the DNS (Domain Name System) records of the sender's domain.
4) When the recipient's email server receives the email, it can retrieve the public key from the DNS records of the sender's domain.
5) The recipient's email server uses the retrieved public key to verify the authenticity of the DKIM signature. If the signature is valid, it means the email has not been modified during transit and has originated from the claimed sender domain.

While the protocol is useful, it is not a foolproof method of preventing spoofing attacks. The DKIM information is not visible to non-technical users and does nothing to address the possibility that the sender is spoofing the emails "from" address, which is the only information most users see. Hackers can steal the private keys used to sign messages with DKIM. Managing public keys can also be time-consuming for email security teams.

Importance of DMARC

DMARC (Domain-based Message Authentication, Reporting & Conformance) is a standard that prevents spammers from spoofing your domain and sending email without your permission. Spammers can forge the "From" address on messages, making the spam appear to be sent by a user in your domain. PayPal spoofing is an example of this, in which a spammer sends you a fraudulent email posing as PayPal in order to obtain your account information. DMARC ensures that these fraudulent emails are blocked before they reach your inbox. Furthermore, DMARC provides you with detailed visibility and reporting into who is sending email on behalf of your domain, ensuring that only legitimate email is received.

Working of DMARC

DMARC is built on two email authentication technologies:

1) DKIM (DomainKeys Identified Mail) which verifies that a message wasn’t altered in transit
2) SPF (Sender Policy Framework) which verifies that the message came from an approved server

DMARC essentially tells an inbox provider that as long as one of these two authentications passes, it should consider the message as legitimate. If they both fail or are not present, the provider should consider the message as suspicious and act according to that domain’s DMARC policy.

11. Bibliography

1. K. Singh, S. S. Grover and R. K. Kumar, "Cyber Security Vulnerability Detection Using Natural Language Processing," 2022 IEEE World AI IoT Congress (AIIoT), Seattle, WA, USA, 2022, pp. 174-178, doi: 10.1109/AIIoT54504.2022.9817336.

2. S. Mumtaz, C. Rodriguez, B. Benatallah, M. Al-Banna and S. Zamanirad, "Learning Word Representation for the Cyber Security Vulnerability Domain," 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 2020, pp. 1-8, doi: 10.1109/IJCNN48605.2020.9207140.

3. HEIDING, F.; KATSIKEAS, S.; LAGERSTRÖM, R. Research communities in cyber security vulnerability assessments: A comprehensive literature review. Computer Science Review, [s. l.], v. 48, 2023. DOI 10.1016/j.cosrev.2023.100551.

4. Chai Jiwen and Liu Shanmei, "Cyber security vulnerability assessment for Smart substations," 2016 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC), Xi'an, China, 2016, pp. 1368-1373, doi: 10.1109/APPEEC.2016.7779741.

5. YU JIAXI; MAO ANJIA; GUO ZHIZHONG. Cyber Security Vulnerability Assessment of Power Industry. TENCON 2006 - 2006 IEEE Region 10 Conference, TENCON 2006. 2006 IEEE Region 10 Conference, [s. l.], p. 1–4, 2006. DOI 10.1109/TENCON.2006.343799.

6. S. Coleman, P. Doody and A. Shields, "Machine Learning for Real-Time Data-Driven Security Practices," 2018 29th Irish Signals and Systems Conference (ISSC), Belfast, UK, 2018, pp. 1-6, doi: 10.1109/ISSC.2018.8585360.

7. YAACOUB, J.-P. A. et al. Robotics cyber security: vulnerabilities, attacks, countermeasures, and recommendations. International journal of information security, [s. l.], v. 21, n. 1, p. 115–158, 2022. DOI 10.1007/s10207-021-00545-8.

8. BRANDÃO, J. E. M. de S. Toward a Vulnerability Mitigation Model. *[s. l.]*: Oxford University Press, 2021. ISBN 978-0-19-880068-2. DOI 10.1093/oxfordhb/9780198800682.013.39.

9. E. de la Cruz Gámez, "Ethical Hacking to remote systems using Metasploit and Kali Linux," 2022 11th International Conference On Software Process Improvement (CIMPS), Acapulco, Guerrero, Mexico, 2022, pp. 224-226, doi: 10.1109/CIMPS57786.2022.10035712.

10. R. Thapa, B. Sehl, S. Gupta and A. Goyal, "Security of operating system using the Metasploit framework by creating a backdoor from remote setup," 2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, 2022, pp. 2618-2622, doi: 10.1109/ICACITE53722.2022.9823460.

11. T. Yamauchi, R. Yoshimoto, T. Baba and K. Yoshioka, "Analysis of commands of Telnet logs illegally connected to IoT devices," 2021 10th International Congress on Advanced Applied Informatics (IIAI-AAI), Niigata, Japan, 2021, pp. 913-915, doi: 10.1109/IIAI-AAI53430.2021.00160.

12. K. Mathew, M. Tabassum and M. V. Lu Ai Siok, "A study of open ports as security vulnerabilities in common user computers," 2014 International Conference on Computational Science and Technology (ICCST), Kota Kinabalu, Malaysia, 2014, pp. 1-6, doi: 10.1109/ICCST.2014.7045193.