CloudSEK

AI Infrastructure as a Strategic Target in Modern Cyber Conflict

Understanding How Exposed MLOps Platforms and Credential Leaks Allow Attackers to Steal Models, Poison Training Pipelines, Harvest Cloud Credentials, Execute Code on Agents, and Compromise the Entire AI Supply Chain

Category Research

Executive Summary

Over the past several years, Artificial Intelligence has rapidly transitioned from experimental research into operational infrastructure embedded within governments, critical industries, and national security environments. Machine learning models now support applications ranging from intelligence analysis and cyber defense to autonomous systems, financial monitoring, and information operations.

The geopolitical landscape of March 2026 lends particular urgency to this research. On February 28, US and Israeli forces launched coordinated strikes against Iranian nuclear sites, leadership infrastructure, and military installations, marking a significant escalation in regional tensions that has had immediate and observable consequences in the cyber domain. Iranian state-sponsored threat groups including MuddyWater (MOIS), APT34 (OilRig), APT33, and APT35 have shown indicators of heightened operational activity following the escalation. Notably, threat intelligence reporting has historically documented MuddyWater's establishment of persistent footholds inside networks spanning the defense, financial, aviation, and non-governmental sectors across North America and Western Europe, well in advance of periods of heightened geopolitical tension. This pattern is consistent with a well-documented Iranian APT doctrine of establishing persistent access inside strategically relevant networks during periods of relative calm, maintaining those footholds through geopolitical escalation, and activating them when operational conditions warrant. The pre-positioning observed here was not reactive to the conflict. It preceded it.

This pattern reflects a broader and well-documented evolution in how nation-state threat actors operate. The cyber domain is no longer a secondary front, it is simultaneous, persistent, and often more durable than kinetic action. **North Korean APT clusters like Lazarus Group and TraderTraitor have spent years perfecting long-dwell supply chain infiltration, embedding malicious packages inside npm and PyPI ecosystems to silently compromise developer infrastructure at scale. Iranian groups have followed a parallel path — targeting CI/CD pipelines, credential stores, and developer tooling, understanding that compromising the build environment is harder to detect, harder to attribute, and more strategically valuable than a direct attack on production systems.**

MLOps platforms are the natural next target for this class of adversary, and this research argues that the threat intelligence community has not yet recognized this. These platforms sit at the convergence of everything APT groups historically prize: centralized authentication credentials, always-on execution agents, persistent cloud storage integrations, and direct access to an organization's most sensitive intellectual assets — its AI models and training data. An adversary embedded within an MLOps environment does not need to repeatedly breach a perimeter. The pipelines run continuously. The workers are always executing. The cloud credentials are always present. This is precisely the kind of silent, persistent, low-visibility access that defines the most consequential state-sponsored intrusions.

The threat extends beyond intelligence collection. In a conflict environment, an adversary with write access to a training pipeline can do something far more damaging than steal a model — they can quietly manipulate it. Subtle corruption of training datasets, poisoned retraining inputs, or silently swapped model artifacts can degrade the reliability of AI systems that militaries and intelligence agencies depend on for surveillance analysis, threat detection, and operational decision-making, with no traditional indicator of compromise, and no obvious point of attribution. In the context of an active conflict where AI increasingly sits inside targeting and battlefield decision pipelines, this is not a theoretical risk.

While much of the public discussion around AI security focuses on model-level attacks such as prompt injection, jailbreaks, and adversarial inputs, these represent only a fraction of the attack surface. The operational infrastructure responsible for training and maintaining AI systems and specifically MLOps platforms introduces a far more powerful and far less scrutinized attack vector, one that is structurally aligned with the long-dwell, supply-chain-focused doctrine that Iran, North Korea, and China have each demonstrated the intent and capability to execute.

This research exposes that gap.

Why This Matters in Modern Conflict

The conflict that erupted on February 28, 2026, with US and Israeli forces striking Iranian nuclear and military infrastructure, has accelerated a dynamic the security community has been slow to articulate clearly: that in modern warfare, the systems that *build* AI capabilities are as much a target as the systems that *deploy* them. This is not a bilateral conflict. The broader Middle East and North Africa region has aligned along fault lines that bring in a complex web of state and proxy threat actors, each with established cyber capabilities, established APT infrastructure, and established doctrine around supply chain and long-dwell operations.

Iranian APT groups including APT34, APT33, MuddyWater, and APT35 do not operate alone. Hamas-affiliated groups like MOLERATS and Gaza Cybergang have historically conducted cyber operations in parallel with kinetic conflict, targeting Israeli infrastructure and Western supporters. Hezbollah-linked actors have demonstrated growing sophistication in persistent access operations. Houthi-aligned threat actors in Yemen, backed by Iranian operational support, have shown increasing interest in targeting logistics, maritime, and defense supply chains across the region. Beyond the immediate conflict, Russian and Chinese APT groups, each with strategic interest in the outcome and in studying Western AI-assisted military capabilities, represent an additional layer of threat that cannot be separated from this geopolitical moment.

None of these actors need to destroy an adversary's AI-assisted surveillance or targeting system. They need to make it untrustworthy. A model that misclassifies at a critical moment, an anomaly detection system quietly tuned to ignore a specific behavioral pattern, a drone coordination model whose decision thresholds have been shifted through poisoned retraining data, these represent forms of battlefield sabotage that leave no forensic trace and generate no traditional security alert. The weapon is the training pipeline itself.

This is where MLOps exposure becomes a national security issue rather than just an enterprise security one. Across this threat landscape, the common thread is patience and pre-positioning. These groups have demonstrated the technical sophistication to embed inside target networks for months before activating. An MLOps environment offers exactly the kind of persistent, always-on foothold that makes long-dwell operations viable, execution agents running continuously, pipelines consuming new data on schedule, cloud credentials permanently authenticated. For a threat actor whose doctrine prioritizes persistence over noise, there is no better place to wait, watch, and when the moment is right, act.

Compromising the systems that build AI may ultimately prove more consequential than compromising the systems that deploy it. In an active conflict where AI sits inside targeting, surveillance, and autonomous platform workflows across an entire region, that distinction has stopped being theoretical.

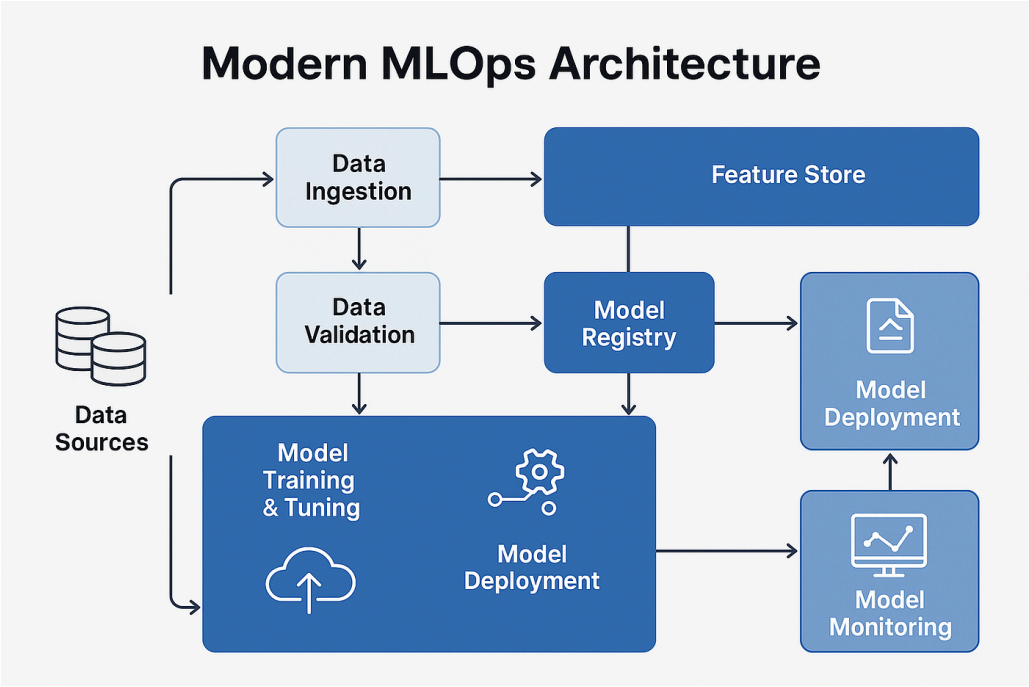
Introduction — Everyone's Attacking the AI, Nobody's Looking at the Control Panel

Over the past few years, Artificial Intelligence has rapidly transitioned from experimental research to production-critical infrastructure. Organizations are deploying machine learning models across nearly every domain: fraud detection, healthcare diagnostics, recommendation systems, autonomous systems, internal copilots, and large-scale automation platforms. As adoption accelerated, a new layer of infrastructure emerged to support the operational lifecycle of these models: **MLOps platforms**.

MLOps platforms exist to solve a fundamental problem. Training a machine learning model is not a single event, it is a continuous process involving datasets, training pipelines, experiment tracking, artifact storage, versioning, and deployment orchestration. These platforms centralize and automate this lifecycle, acting as the operational backbone that connects data, compute, models, and production environments.

At a high level, MLOps platforms function as a control plane for AI systems. They coordinate how models are trained, where datasets are stored, how artifacts are versioned, and how execution agents retrieve and

run training workloads. They also integrate directly with external infrastructure such as cloud storage providers, container registries, and compute environments.



Because of this central role, access to an MLOps platform is fundamentally different from access to a typical application dashboard. It provides visibility and control over the entire machine learning lifecycle, including training data, trained models, execution pipelines, and the infrastructure used to run them. In environments where AI systems increasingly support critical operations, such access can present a strategic opportunity for advanced threat actors seeking to observe, replicate, or manipulate AI capabilities.

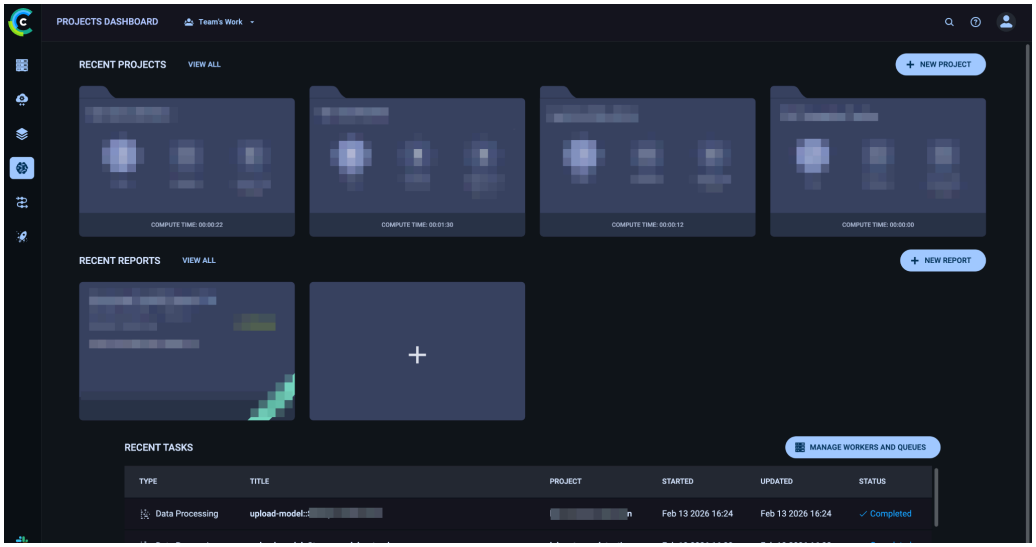
Despite this, much of the current security conversation around AI remains focused on model-layer attacks such as prompt injection, adversarial inputs, and output manipulation. These attacks target the behavior of the model itself. While important, they represent only one layer of the attack surface.

This research focuses on a different and far more foundational layer: the infrastructure used to build and operate AI systems.

During the course of this research, exposed credentials associated with MLOps platforms were identified across publicly accessible code repositories. In parallel, multiple internet-accessible MLOps instances were discovered with weak or absent access controls, allowing unauthenticated or low-privileged access

to internal machine learning environments. In adversarial contexts, such exposure could provide entry points for APT groups to weaponize AI infrastructure for intelligence collection, model theft, or manipulation of training pipelines.

In several cases, access to these platforms provided direct visibility into machine learning projects, training pipelines, experiment configurations, and model artifacts.



Snapshot showing the dashboard of a ClearML Platform

This represents a structural shift in how AI systems can be compromised. The attack surface is no longer limited to the model interface, it extends to the operational infrastructure that powers the entire machine learning lifecycle.

As MLOps platforms continue to proliferate across organizations, they are increasingly becoming high-value targets. Understanding how exposed credentials, insecure deployments, and insufficient credential isolation affect these platforms is critical to securing modern AI environments.

This research explores how exposed MLOps credentials and publicly accessible platforms introduce a scalable attack surface capable of enabling unauthorized access to models, datasets, pipelines, and underlying cloud infrastructure, often without exploiting a single software vulnerability.

Research Scope Note

This research spanned multiple MLOps platforms, including ClearML, MLflow, Kubeflow, Metaflow, ZenML, and Weights & Biases (wandb) etc. The observations, attack scenarios, and exposure patterns discussed throughout this report were validated across these platforms to ensure the findings reflect ecosystem-wide risks rather than platform-specific issues.

To identify exposed credentials, a custom API key crawler was developed to scan publicly accessible GitHub repositories. The crawler leveraged regex-based pattern matching to detect authentication artifacts associated with MLOps platforms, including keys and credentials such as `BIGML_API_KEY`, `MLFLOW_TRACKING_USERNAME`, `KUBEFLOW_USERNAME`, `METADATA_SERVICE_AUTH_KEY`, `ZENML_STORE_API_KEY`, `WANDB_API_KEY`, and `CLEARML_API_SECRET_KEY`. Using the GitHub API, the crawler analyzed repository contents and extracted contextual information including repository identifiers, service endpoints, usernames, access keys, and secret values for validation purposes.

This process was conducted continuously **across thousands of public repositories**, revealing exposed credentials in source code, configuration scripts, and environment files. In parallel, internet-wide scanning platforms such as Shodan and FOFA were used to identify publicly accessible MLOps instances by leveraging service fingerprints, including HTTP response signatures and exposed dashboard interfaces. While exposure levels varied across platforms, this research confirmed over 100+ exposed credential instances and more than 80+ publicly accessible MLOps deployments (Crawling happened for 3 days). The numbers can drastically increase considering the fact that AI adoption is getting popular.

Due to practical limitations in embedding screenshots and visual evidence across multiple platforms, ClearML was selected as the primary reference for demonstrating user interfaces, credential exposure scenarios, and platform interactions. These examples are provided strictly for illustrative purposes. **The security implications and attack surfaces described in this research apply broadly to MLOps platforms with similar architectural and operational models.**

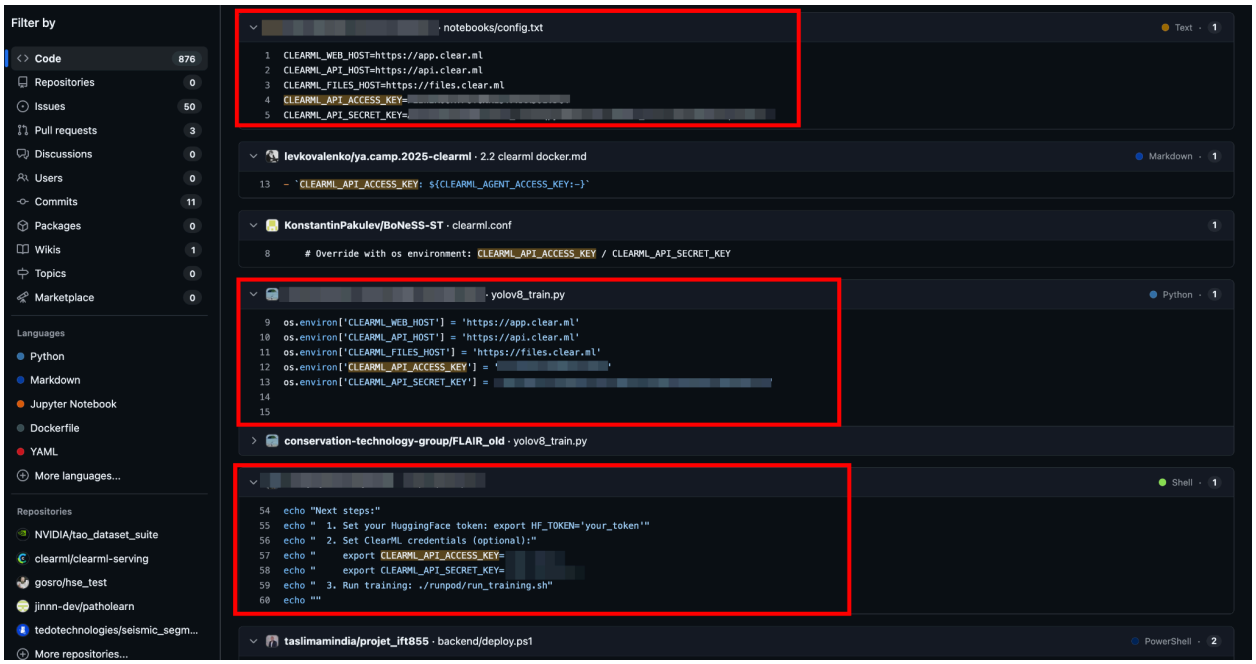
The Root Cause: It's Still Credential Exposure — But Now It Owns the AI Pipeline!

For over a decade, exposed credentials in public repositories have remained one of the most consistent and effective entry points for infrastructure compromise. API keys, access tokens, and configuration files are frequently hardcoded during development for convenience, often ending up committed to version control systems and inadvertently exposed through public repositories.

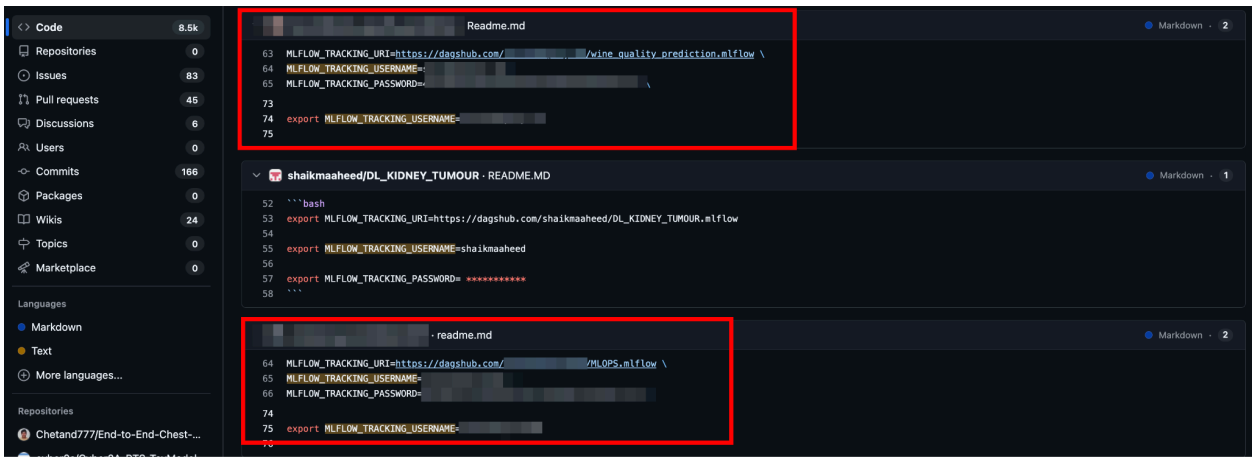
Traditionally, exposed credentials have provided attackers access to cloud resources, internal APIs, or database systems. However, as organizations increasingly adopt MLOps platforms to operationalize machine learning workflows, exposed credentials **now grant access to something far more critical, the control plane responsible for building, managing, and storing AI models.**

Unlike typical application credentials, MLOps credentials **provide visibility into the entire machine learning lifecycle.** These credentials authenticate directly to platforms responsible for orchestrating training pipelines, managing datasets, storing model artifacts, and coordinating execution agents. As a result, exposure of these credentials provides attackers with access not just to an application, but to the infrastructure responsible for creating and maintaining AI systems.

During this research, exposed credentials associated **with multiple MLOps platforms were identified across publicly accessible code repositories**. These credentials appeared in various forms, including hardcoded API keys, configuration files, environment variable definitions, and deployment scripts. In many cases, these credentials were committed directly into source code repositories, often as part of development workflows or automation scripts.



Exposure of ClearML related access keys in GitHub repositories

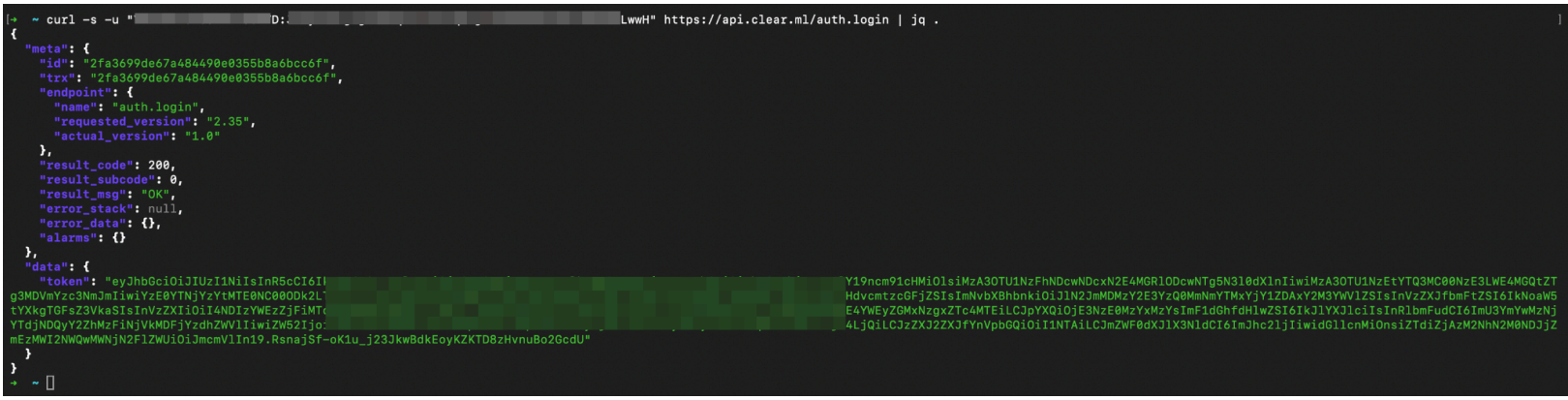


Exposure of MLFlow credentials in Github repositories

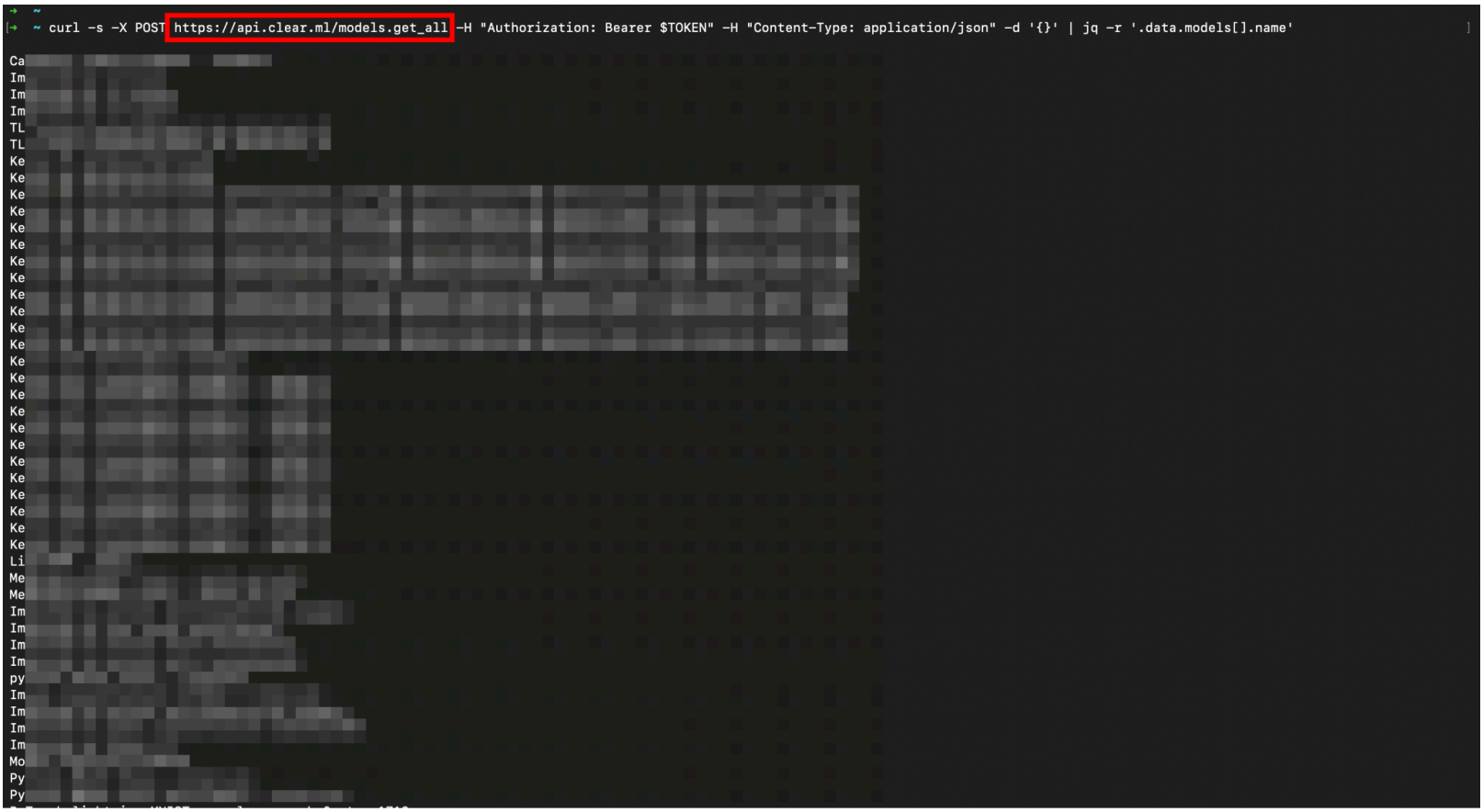
These credentials typically consisted of access key and secret key pairs or API tokens used by developers to authenticate local training environments, execution agents, or automation pipelines with the MLOps platform. Because these credentials are designed to enable programmatic access, they often remain valid for extended periods and are rarely rotated unless explicitly revoked.

Once exposed, these credentials can be used by attackers to authenticate directly to the MLOps platform, gaining access equivalent to the original user or service account.

Unlike traditional credential exposure scenarios where attackers must identify additional vulnerabilities to escalate access, authentication to an MLOps platform inherently provides broad visibility into machine learning operations. This includes access to project structures, experiment configurations, training pipelines, and references to stored datasets and model artifacts.



Screenshot showing successful authentication to an MLOps platform using exposed credentials



Snapshot displaying all model information, filtered to show only the model names due to the large output size.

In addition to credentials exposed in public repositories, configuration files associated with training environments and execution agents frequently contained embedded authentication tokens. These configurations are often distributed across development systems, CI/CD pipelines, and compute nodes, increasing the likelihood of exposure through misconfigured repositories or improperly secured environments.

This creates a scalable and reliable entry point for attackers. Rather than exploiting software vulnerabilities, adversaries can authenticate using valid credentials and gain immediate access to the MLOps control plane.

For nation-state threat actors and government-backed APT groups, this type of access represents a strategic foothold. Because MLOps platforms maintain references to datasets, model artifacts, and external storage integrations, credential-based access allows adversaries to systematically enumerate and monitor the assets that power an organization's AI development pipeline.

In geopolitical conflict scenarios, such visibility can support long-term intelligence operations. Access to the control plane allows threat actors to understand how AI systems are trained, what data sources they rely on, and how models are deployed or updated. This information can be leveraged to replicate capabilities, extract sensitive datasets at scale, or identify opportunities to subtly influence training workflows during critical periods.

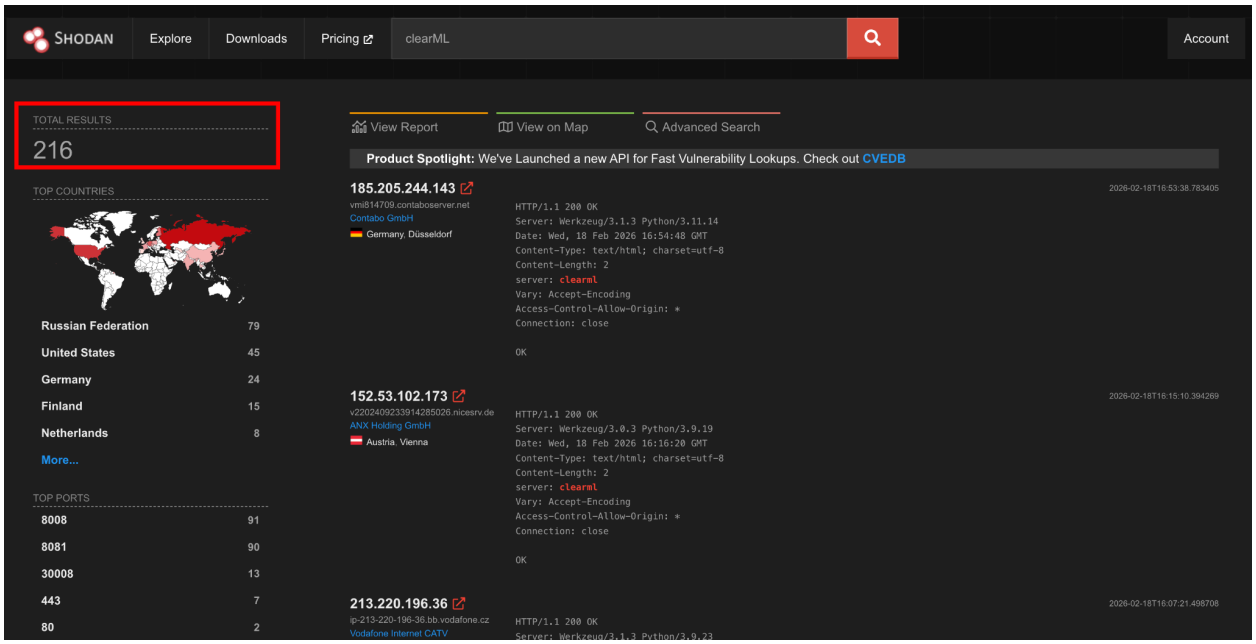
As a result, the consequences of credential exposure extend far beyond traditional data theft. A single exposed MLOps credential may provide a persistent vantage point into an organization's AI infrastructure, enabling adversaries to observe, harvest, or potentially manipulate the systems responsible for generating AI capabilities during times of strategic or geopolitical tension.

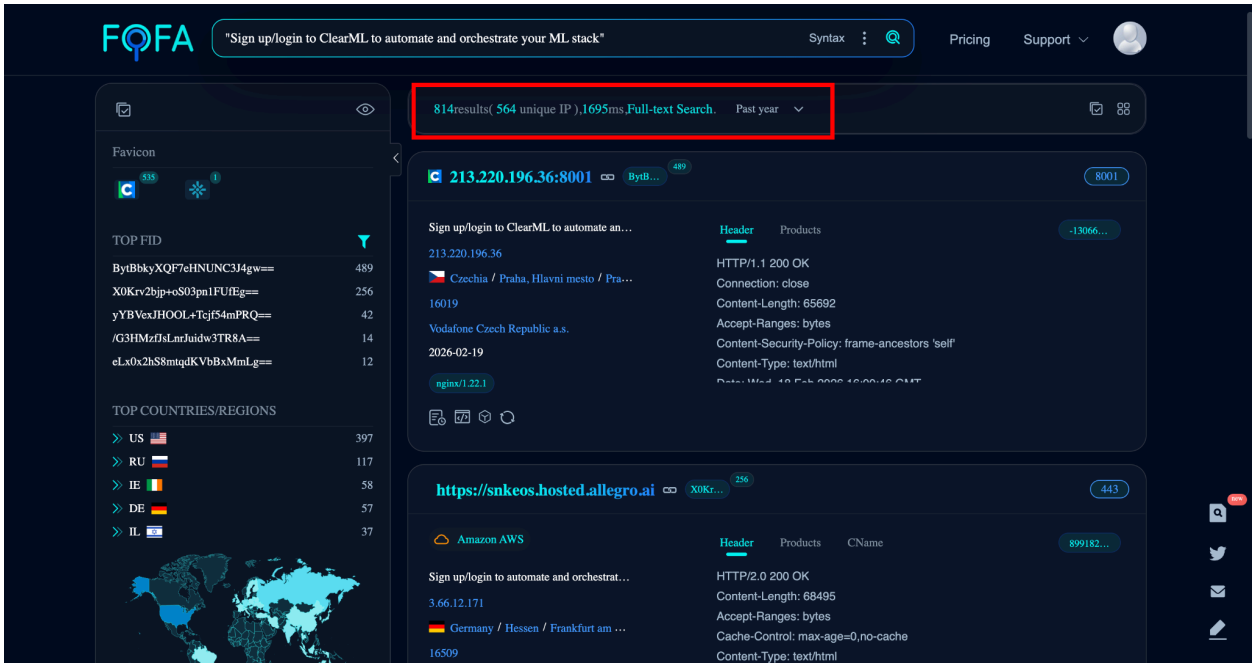
Internet-Exposed MLOps Platforms: When the Control Plane Doesn't Even Require Credentials

While exposed credentials provide a direct pathway into MLOps platforms, an equally concerning issue emerged during this research: numerous MLOps instances were found to be directly accessible over the internet with weak, misconfigured, or completely absent authentication controls.

Unlike credential exposure scenarios, these platforms did not require attackers to obtain API keys or tokens. The control plane itself was already exposed.

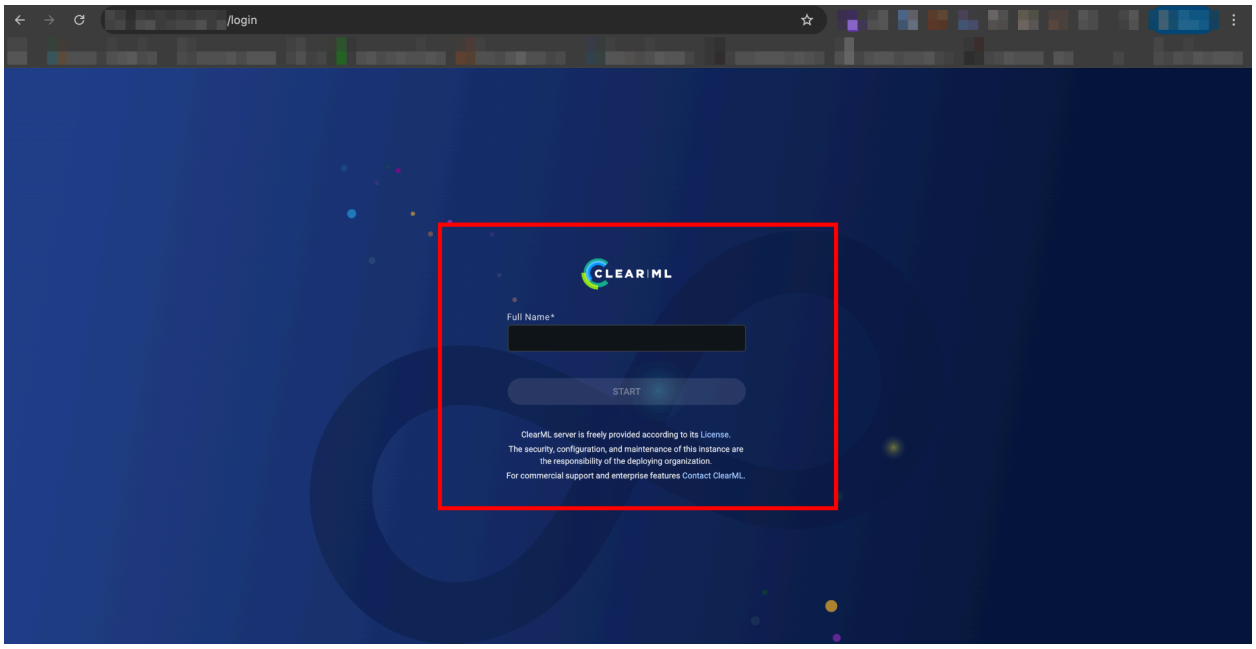
Using internet-wide asset discovery platforms such as Shodan and FOFA, multiple publicly accessible MLOps instances were identified. These instances were discoverable through simple service fingerprinting and HTTP response analysis, often revealing web interfaces exposed on default ports associated with MLOps platforms.



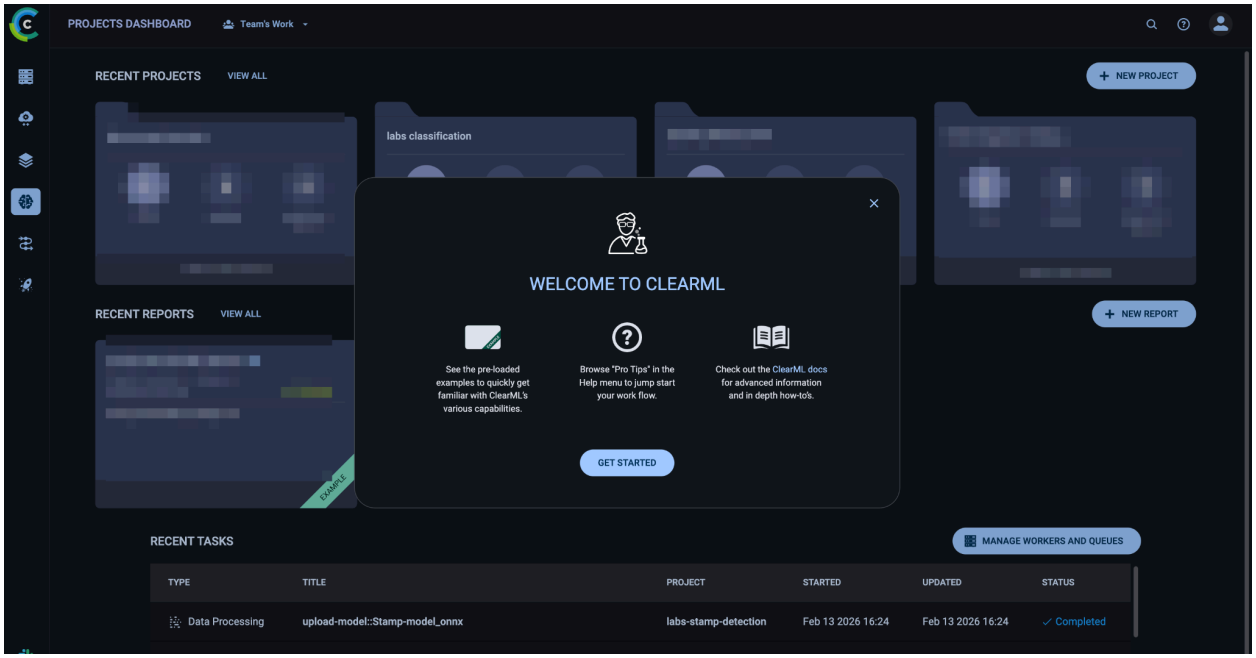


In several cases, accessing the platform's web interface presented a fully functional dashboard without requiring authentication. Project listings, experiment tracking dashboards, and pipeline configurations were directly visible to any unauthenticated user.

In other instances, the platforms permitted unrestricted user registration. This allowed attackers to create new accounts and gain authenticated access to the environment without prior authorization. Because these platforms were deployed without proper access restrictions, attackers could onboard themselves as legitimate users.

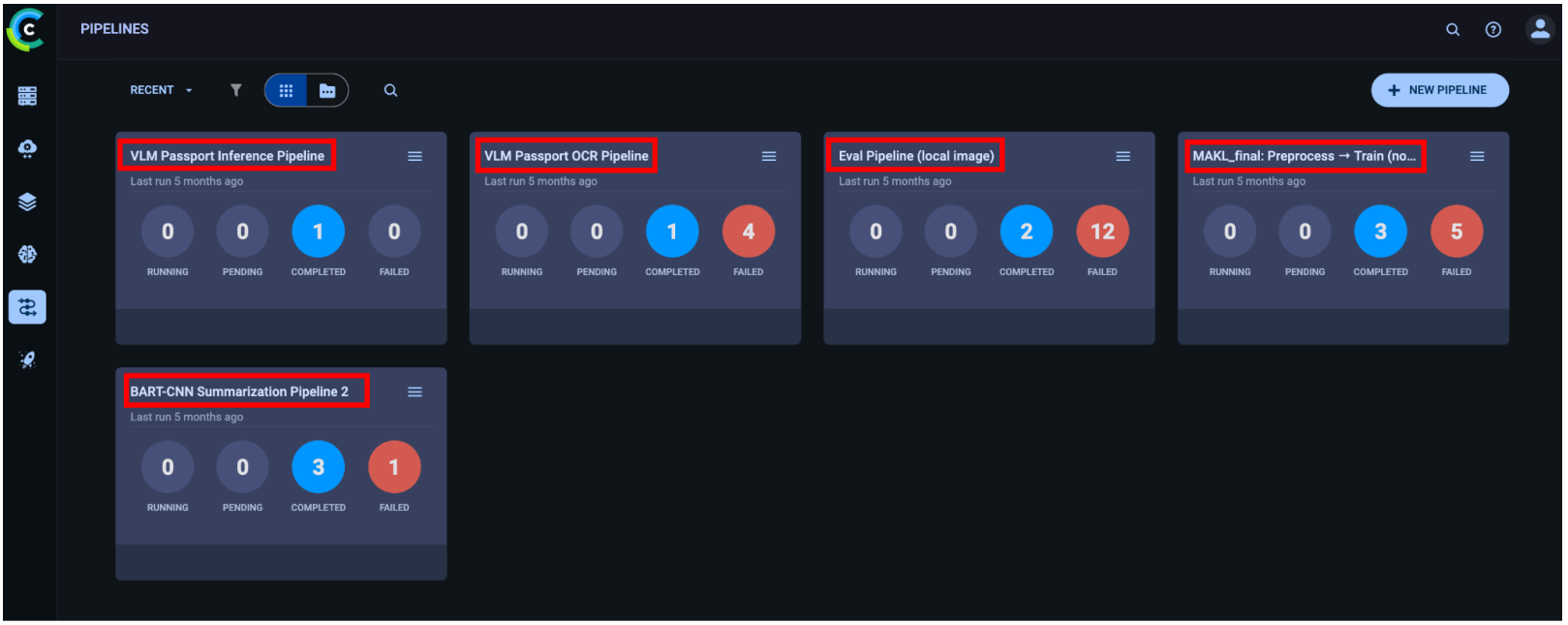


Snapshot showing the exposed ClearML user registration page.

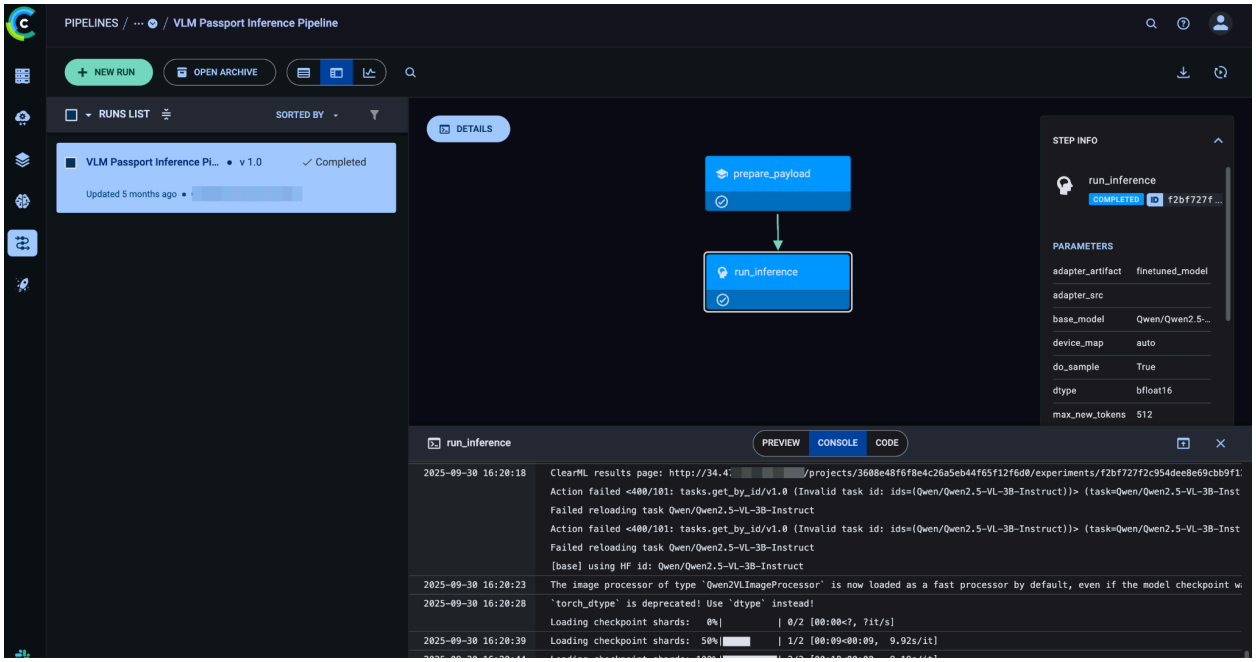


Screenshot displaying the dashboard accessible after user registration using a randomly created username.

Once inside, attackers could enumerate internal machine learning projects, view experiment histories, and observe training activity. These platforms often exposed detailed metadata about ongoing and completed training tasks, including execution parameters, dataset references, and artifact storage locations.



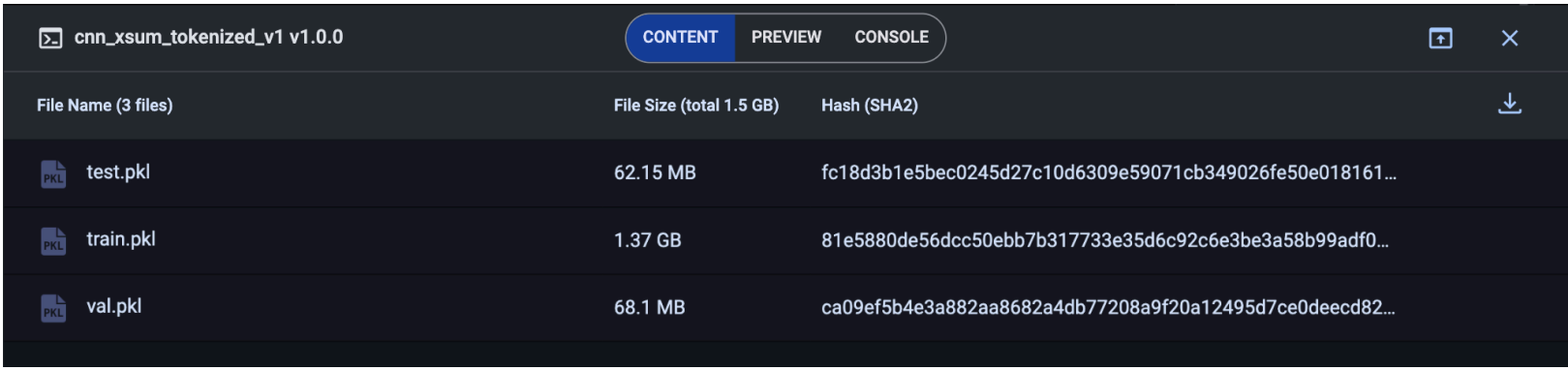
Snapshot showing the available pipelines in one of the exposed ClearML platform instances.



Snapshot showing the inference output of one of the completed pipelines.

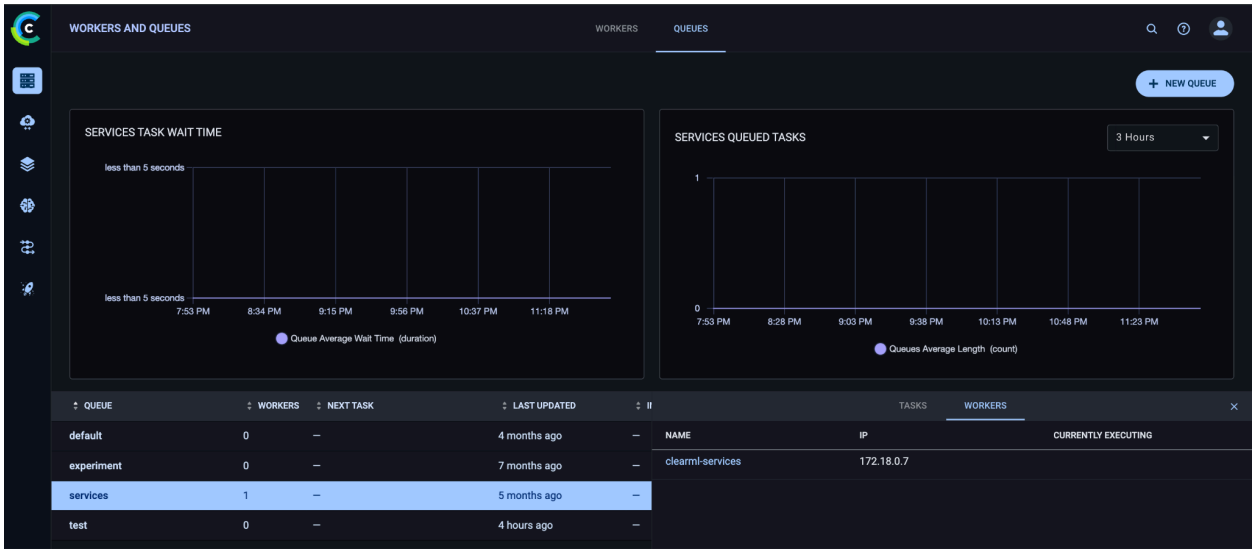
When these platforms are deployed on publicly accessible infrastructure without proper hardening, the entire machine learning control plane becomes exposed.

Unlike traditional application exposures, where attackers may gain access to a limited subset of application functionality, exposed MLOps platforms provide direct visibility into the infrastructure responsible for training and managing AI models. This includes access to project structures, experiment metadata, and execution activity.



Screenshot demonstrating the ability to download model artifacts

In many cases, these platforms also exposed connected execution agents and storage integrations, revealing the broader infrastructure ecosystem associated with the machine learning environment.



Snapshot showing the workers and Queue of the current instance

This type of exposure significantly lowers the barrier to compromise. Attackers do not need to discover credential leaks or exploit software vulnerabilities—they can simply discover exposed instances through internet scanning and access the platform directly.

Because MLOps platforms serve as centralized orchestration systems, this access provides attackers with a starting point to enumerate machine learning workflows, identify sensitive assets, and potentially escalate access further into connected infrastructure.

The exposure of these platforms represents a structural security gap. As organizations rapidly deploy MLOps platforms to operationalize machine learning workflows, insufficient attention is often given to access control and deployment hardening. This results in control planes being exposed beyond intended trust boundaries.

The impact of this exposure becomes even more severe when considering the credentials and infrastructure integrations managed by these platforms. Access to the MLOps platform can reveal references to external storage systems, execution environments, and authentication credentials—creating pathways for attackers to extend their access beyond the platform itself.

What Happens After Login: Attacks Enabled by Access to MLOps Control Planes

Authentication to an MLOps platform fundamentally changes the attacker's position within the environment. Instead of interacting with AI systems through external interfaces, the attacker gains access to the orchestration layer responsible for building and maintaining those systems.

In modern conflicts, artificial intelligence increasingly sits inside military decision pipelines. AI models are now used for intelligence analysis, target identification, surveillance interpretation, and battlefield simulations. Recent operations have demonstrated how large-scale AI systems can process massive volumes of intelligence data and generate prioritized targets for military operations.

This means the infrastructure responsible for training and maintaining those models becomes a strategic attack surface.

Access to an MLOps control plane provides visibility into how these AI capabilities are developed, what data sources they rely on, and how models are updated over time. From this vantage point, attackers can observe or influence the lifecycle of systems that may ultimately support operational or strategic decision-making.

The following attack scenarios become possible once access is obtained.

1. Dataset Exfiltration: Intelligence Collection Through Training Data

Training datasets form the foundation of modern AI-driven intelligence systems. These datasets often contain large-scale telemetry, surveillance feeds, behavioral analytics, or operational records used to train models responsible for analyzing real-world events.

Authenticated access to the MLOps platform allows attackers to enumerate datasets, identify their storage locations, and retrieve them through platform integrations.

In geopolitical conflicts, access to such data provides significant intelligence value. By analyzing training datasets, adversaries can understand what signals a model uses to classify activity, what patterns it prioritizes, and what data sources it trusts.

This allows attackers to study how AI systems interpret intelligence and potentially design operations that exploit blind spots within those models.

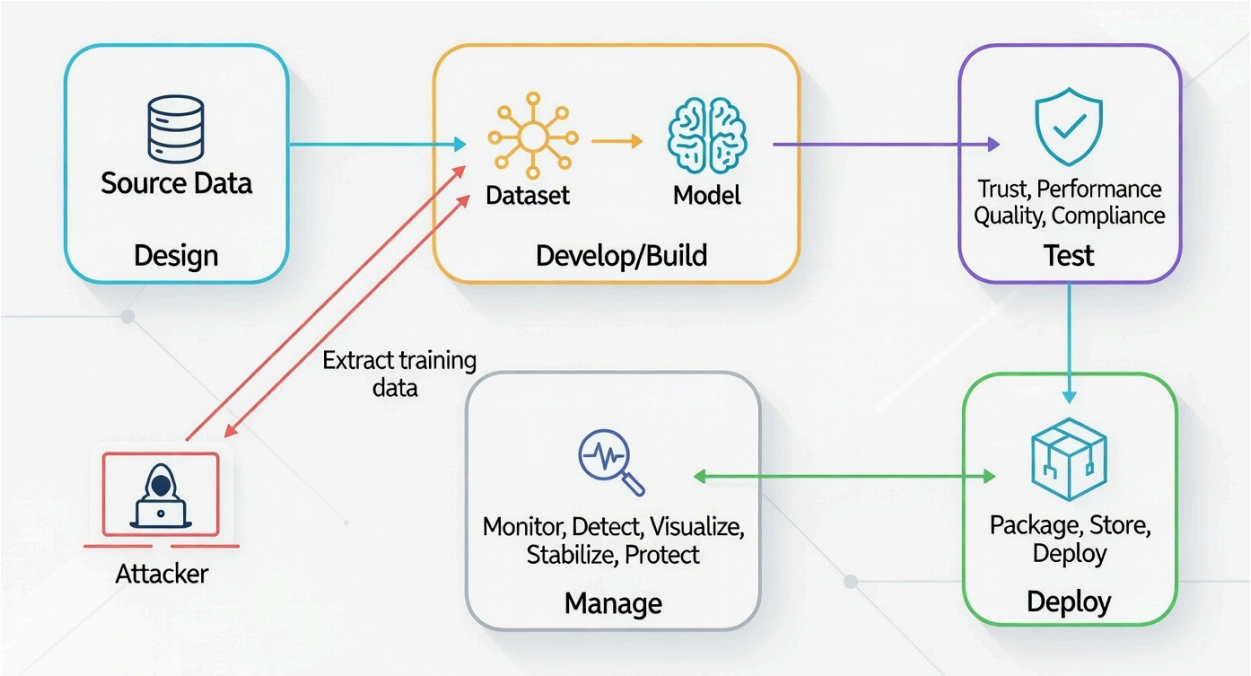
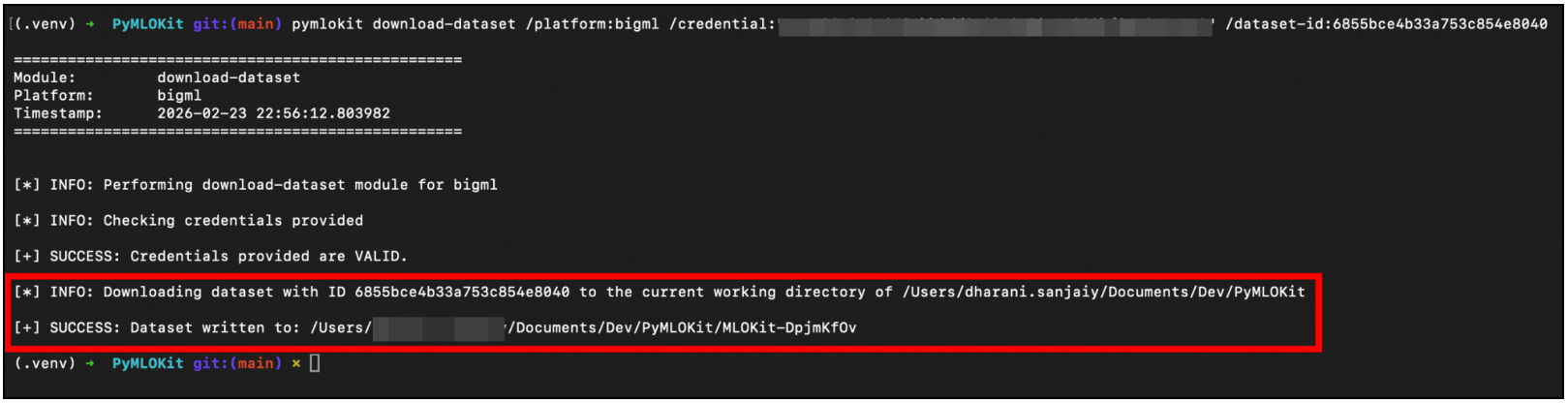


Diagram showing data extraction attack

```
(.venv) ➜  PyMLOKit git:(main)
(.venv) ➜  PyMLOKit git:(main) pymlokit list-datasets /platform:bigml /credential:"

================================================
Module:         list-datasets
Platform:       bigml
Timestamp:      2026-02-23 22:55:01.859838
================================================


[*] INFO: Performing list-datasets module for bigml

[*] INFO: Checking credentials provided

[+] SUCCESS: Credentials provided are VALID.

                                        Name | Visibility |              Creation Date |              Dataset ID
-------------------------------------------------------------------------------------------------------------------------
                  iris-dataset | Training (80%) |    Private | 2025-06-20T19:56:21.051000 | 6855bce5b33a753c854e8043
                      iris-dataset | Test (20%) |    Private | 2025-06-20T19:56:20.817000 | 6855bce4b33a753c854e8040
                                   iris-dataset |    Private | 2025-06-20T19:55:36.754000 | 6855bcb88baab6a917b8e68d
                         adult-processed-test |    Private | 2025-06-20T15:43:39.804000 | 685581abb30f2da831717d41
                        adult-processed-train |    Private | 2025-06-20T15:10:18.632000 | 685579da1fabd8342a67507c
SourceTest Animals [filtered] | Training (80%) |    Private | 2025-06-17T21:36:08.987000 | 6851dfc8dd54b2205d865568
    SourceTest Animals [filtered] | Test (20%) |    Private | 2025-06-17T21:36:08.672000 | 6851dfc8e2a31f95ca1b64e2
                SourceTest Animals [filtered] |    Private | 2025-06-17T21:35:34.217000 | 6851dfa68baab6a917b8e2f6
                              SourceTest Animals |    Private | 2025-06-17T21:31:04.923000 | 6851de98e2262f3ec0f39487
   Copy of Animals [filtered] | Training (80%) |    Private | 2025-06-15T17:10:01.899000 | 684efe6912ac6d80c5d924be
       Copy of Animals [filtered] | Test (20%) |    Private | 2025-06-15T17:10:01.554000 | 684efe69b33a753c854e7834
                  Copy of Animals [filtered] |    Private | 2025-06-15T17:09:40.564000 | 684efe54b30f2da8317176b1
                               Copy of Animals |    Private | 2025-06-15T17:07:58.910000 | 684efdeecb8cac68cb152dae
                                       Animals |    Private | 2025-06-15T17:02:20.771000 | 684efc9c32afd54b4b70862a

(.venv) ➜  PyMLOKit git:(main)
```

Snapshot showing the list of available datasets



Snapshot showing the successful retrieval of one of the dataset

MLOKit-ZRLTSmAU

sepalLenthg	sepalWidth	petalLength	petalWidth	Target
5.1	3.5	1.4	0.2	Iris-setosa
4.6	3.1	1.5	0.2	Iris-setosa
5.8	4.0	1.2	0.2	Iris-setosa
5.4	3.4	1.5	0.4	Iris-setosa
4.4	3.0	1.3	0.2	Iris-setosa

Snapshot showing some of the content of the downloaded dataset

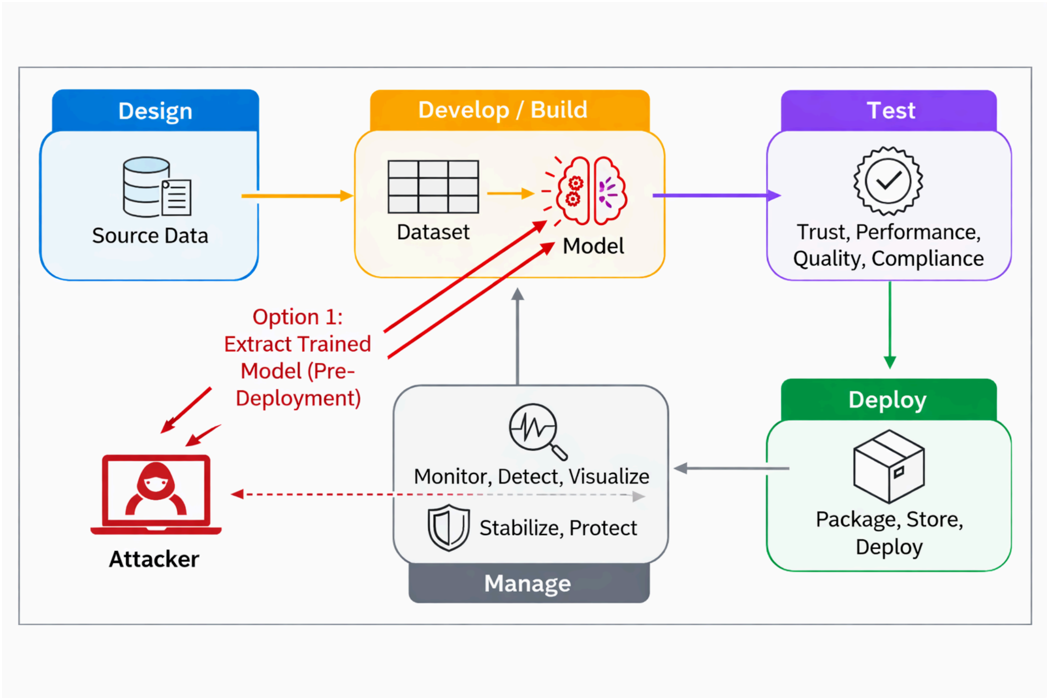
2. Model Artifact Theft: Understanding Autonomous Decision Systems

Machine learning models encapsulate the logic used by automated systems to interpret complex datasets and generate decisions or recommendations. In many modern environments, these models are not limited to analytical tasks but increasingly power systems involved in surveillance analysis, autonomous navigation, and target identification.

When attackers obtain model artifacts from an MLOps platform, they gain the ability to analyze these models offline. This allows them to study how the system processes sensor inputs, how it prioritizes signals, and what conditions influence its outputs.

In the context of modern conflict, such models may support systems used for large-scale surveillance analysis, autonomous drone navigation, sensor fusion, or target prioritization. Access to these artifacts allows adversaries to understand how automated systems interpret the environment, which signals they trust, and where potential blind spots exist.

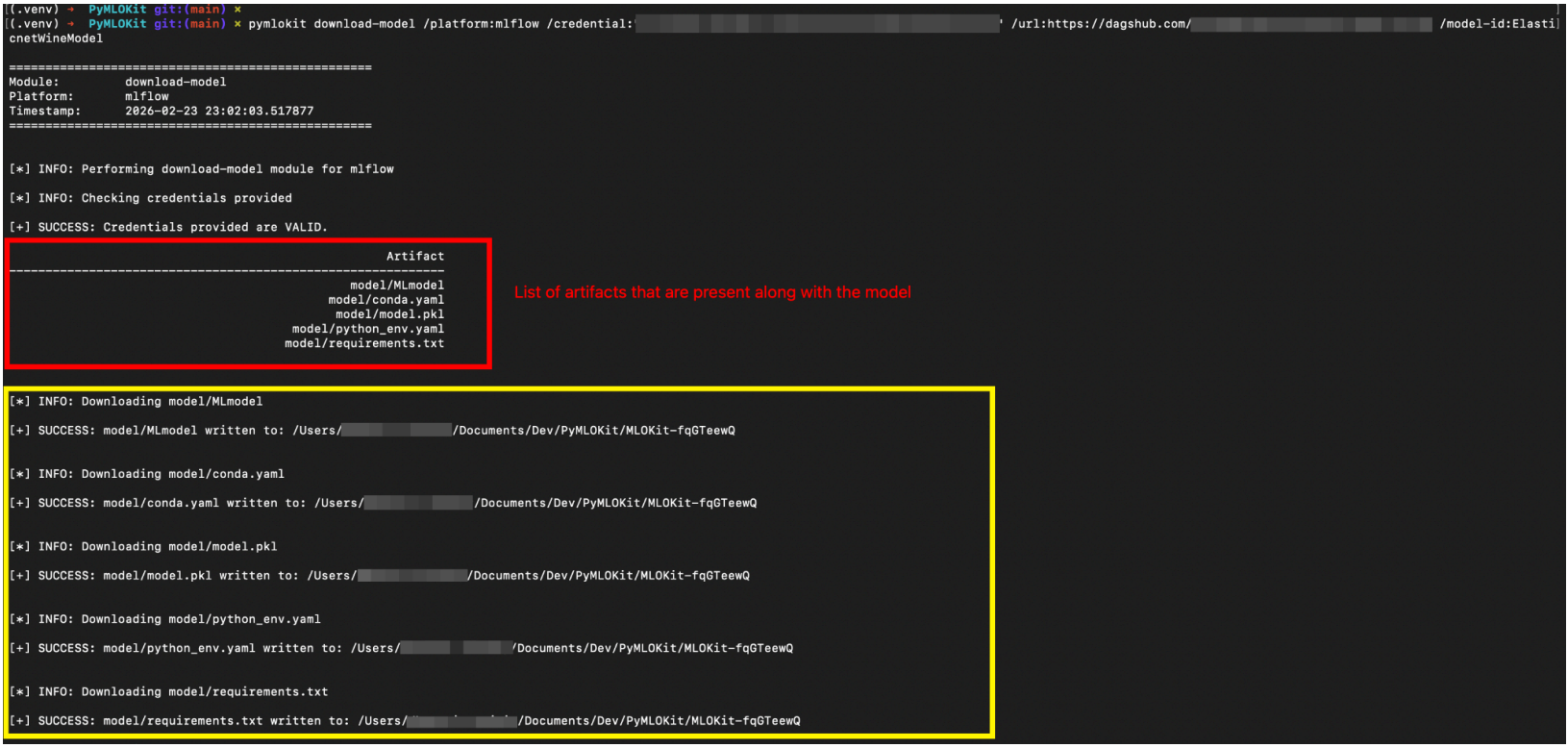
For nation-state threat actors, this type of insight can provide a strategic advantage. By studying these systems offline, adversaries can better understand the decision logic behind automated analysis pipelines or autonomous platforms that increasingly support military and intelligence operations.



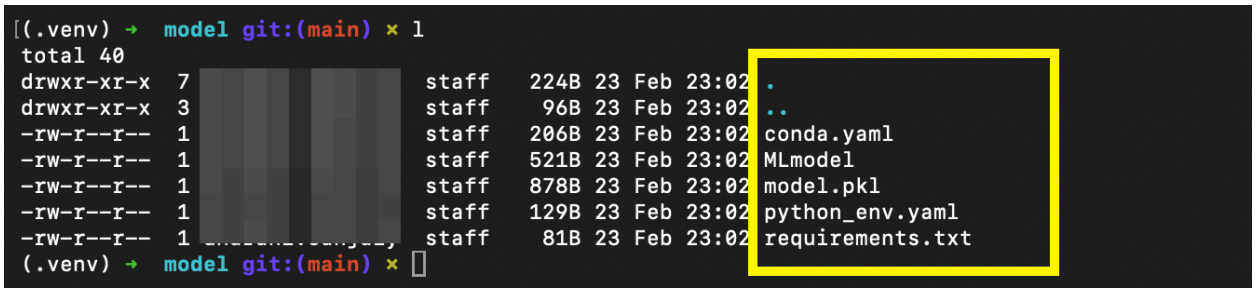
Performing model extraction before deployment

```
(.venv) → PyMLOKit git:(main) ×
(.venv) → PyMLOKit git:(main) × pymlokit list-models /platform:mlflow /credential:[redacted] /url:https://dagshub.com/[redacted]

================================================
Module:          list-models
Platform:        mlflow
Timestamp:       2026-02-23 23:00:45.774868
================================================


[*] INFO: Performing list-models module for mlflow

[*] INFO: Checking credentials provided

[+] SUCCESS: Credentials provided are VALID.

               Name | Version | Status | Description |                                              Artifact Location
----------------------------------------------------------------------------------------------------------------------------------------------------------
              model |       1 |  READY |             |                                    runs:/ffd71dd6bf774ba18ebeb9efeaa595c5/model
ElasticnetWineModel |       2 |  READY |             | mlflow-artifacts:/709c9596874c46978c89c3f6c37d8c7d/72ea597b27084c25ad079052a3901ff1/artifacts/model
(.venv) → PyMLOKit git:(main) ×
```

Snapshot showing the list of available models



Snapshot showing the successful retrieval of model along with its artifacts



3. Training Data Poisoning: Influencing Model Behavior

Modern AI systems rely on continuous retraining as new data becomes available. MLOps platforms orchestrate this lifecycle by connecting datasets, training pipelines, and execution environments responsible for updating models.

Attackers with access to the control plane may gain visibility into how these pipelines operate and how new data influences model behavior over time. In environments where machine learning models support surveillance analysis, autonomous navigation systems, or automated threat detection, this lifecycle becomes critical to maintaining system accuracy.

Because machine learning systems inherently trust their training data, any compromise within the training pipeline introduces the possibility that model behavior may evolve in unintended ways. Subtle changes in datasets, labeling processes, or training inputs can propagate through retraining cycles and influence how models interpret signals in the future.

In operational environments where AI assists with surveillance interpretation, automated targeting analysis, or autonomous platform behavior, even small shifts in model behavior could affect how those systems respond to real-world inputs.

4. Execution Environment Abuse: Running Arbitrary Code Through Workers

MLOps platforms rely on distributed execution agents to run training jobs, model evaluations, and data processing tasks. These workers trust instructions originating from the control plane and execute them within compute environments that often have access to internal datasets, model repositories, and development infrastructure.

Authenticated attackers can leverage this trust relationship to execute workloads through legitimate platform mechanisms.

Because these execution environments frequently operate within high-performance computing clusters or internal research networks, they may provide access to the broader infrastructure used to develop and test AI systems. In environments supporting advanced machine learning research, these systems may also host simulation environments, sensor datasets, or model evaluation frameworks used to test autonomous systems.

As a result, access to the execution layer can extend the attacker's visibility beyond individual models and into the infrastructure responsible for developing and validating AI capabilities.

These Aren't Exploits, These Are Features!

The attacks described in the previous section do not rely on exploiting software vulnerabilities. They are performed using legitimate functionality exposed by machine learning platforms as part of normal operational workflows.

Machine learning platforms are designed to allow authenticated users to access datasets, download model artifacts, execute training pipelines, and interact with integrated storage systems. These capabilities are essential for development and automation, but they also mean that once authentication is obtained, the platform itself provides direct access to sensitive assets.

There is no need for privilege escalation or exploit chaining. The same interfaces used by engineers and automation systems can be used by attackers. This makes compromise both practical and difficult to detect. Actions such as downloading models, accessing datasets, or executing pipelines appear as normal platform activity.

The effectiveness of these attacks is not due to software vulnerabilities, but due to how these platforms are designed and operated, an issue that becomes more apparent when examining the security maturity of many open-source MLOps platforms.

Security Immaturity of Open-Source MLOps Platforms:

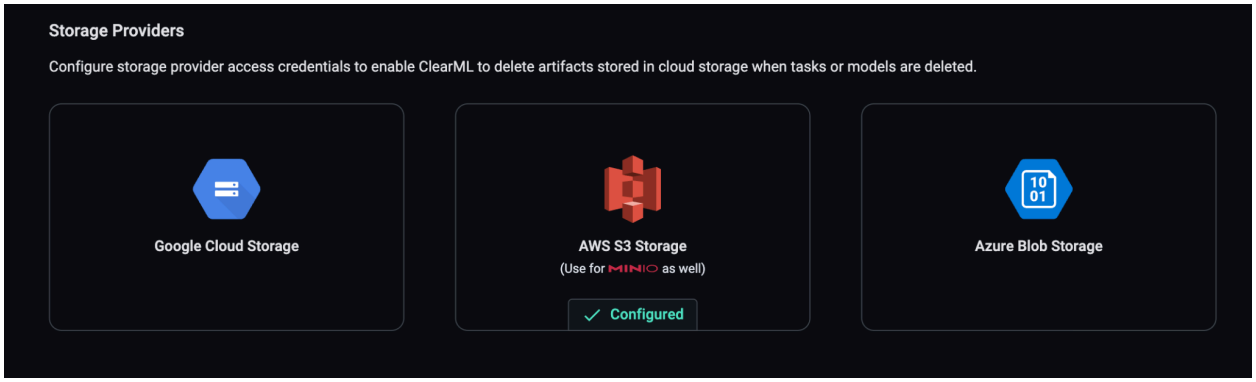
One of the most consistent patterns observed throughout this research was not the presence of traditional software vulnerabilities, but rather the relative security immaturity of many open-source MLOps platforms themselves.

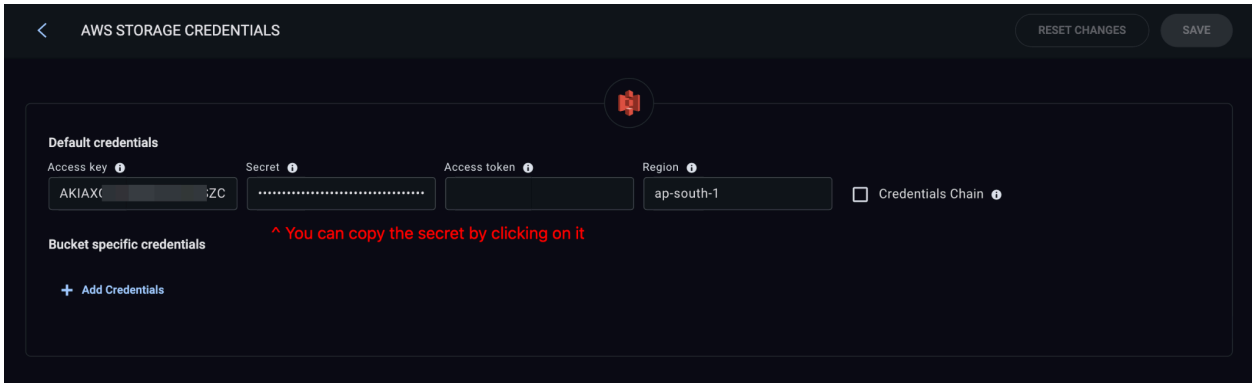
Unlike traditional infrastructure components such as CI/CD systems, cloud IAM services, or container orchestration platforms, which have undergone years of security hardening, **many MLOps platforms are comparatively new.** Their rapid adoption has been driven primarily by the need to accelerate model development, automate training workflows, and simplify experimentation, often prioritizing usability and operational flexibility over secure-by-default design principles.

This immaturity manifests in several ways, particularly in how these platforms handle sensitive credentials, access controls, and trust boundaries.

A common example observed during this research was the way cloud storage integrations were configured and managed. MLOps platforms frequently require access to external storage providers such as AWS S3, Google Cloud Storage, or Azure Blob Storage in order to store datasets, model artifacts, and experiment outputs. To enable this functionality, users configure storage credentials directly within the platform.

In multiple instances, these credentials were stored and displayed in retrievable form through the platform's web interface.





This design introduces a critical trust assumption: that anyone with access to the MLOps platform should also be able to retrieve and use the underlying cloud credentials. In practice, this assumption significantly expands the blast radius of any platform compromise.

This design introduces a critical trust assumption: that anyone with access to the platform can also retrieve and use the underlying cloud credentials. In practice, this allows attackers to directly access external storage systems, exposing datasets, model artifacts, and other sensitive assets.

This reflects the relative security immaturity of the MLOps ecosystem compared to mature infrastructure platforms such as Jenkins. Over more than a decade of widespread deployment, Jenkins has undergone extensive security hardening, formal vulnerability disclosure processes, and architectural improvements driven by years of real-world attack exposure. In contrast, many MLOps platforms have only recently transitioned from research and experimentation tools into production-critical infrastructure.

Security research has already uncovered numerous vulnerabilities across MLOps platforms, including authentication bypasses, remote code execution vectors, and unsafe handling of models and datasets. However, a more fundamental issue lies in inherent risks within machine learning workflows themselves.

Unlike traditional software artifacts, machine learning models and datasets often support embedded code execution as part of normal functionality. Loading a model or dataset can execute arbitrary code by design (yes, looking at you pickle), not as a software bug, but as an intended feature of the ecosystem. These inherent risks may never receive CVEs or patches, yet still introduce real compromise potential.

As a result, MLOps platforms frequently operate as centralized aggregation points for both infrastructure access and executable artifacts, while lacking the mature security boundaries and operational hardening seen in traditional control plane systems like Jenkins.

This creates a structural risk where access to a single platform can expose machine learning workflows, sensitive datasets, model artifacts, and the underlying cloud infrastructure supporting them.

Securing MLOps Platforms: Reducing the Attack Surface

The risks associated with exposed MLOps platforms are largely driven by credential exposure, insecure deployments, and excessive trust in platform integrations. Reducing this attack surface starts with proper credential management.

Authentication tokens, API keys, and cloud credentials should never be hardcoded into source code or configuration files. Instead, they should be managed using secure secret management systems, rotated regularly, and scoped with the principle of least privilege.

Access to MLOps platforms should also be restricted. Platforms should not be exposed directly to the public internet without proper access controls. Network restrictions, enforced authentication, and disabling unrestricted account registration can significantly reduce unauthorized access.

Cloud storage integrations should use short-lived credentials or role-based access instead of static keys. This limits the impact of credential exposure and prevents attackers from directly accessing underlying storage infrastructure.

Finally, MLOps platforms should be treated as critical infrastructure. Monitoring access to datasets, models, and pipelines, and applying the same security controls used for CI/CD and cloud control planes, can significantly reduce the risk of compromise.

Securing AI systems requires securing the platforms and credentials behind them, not just the models themselves.

Responsible Disclosure and Ethical Considerations

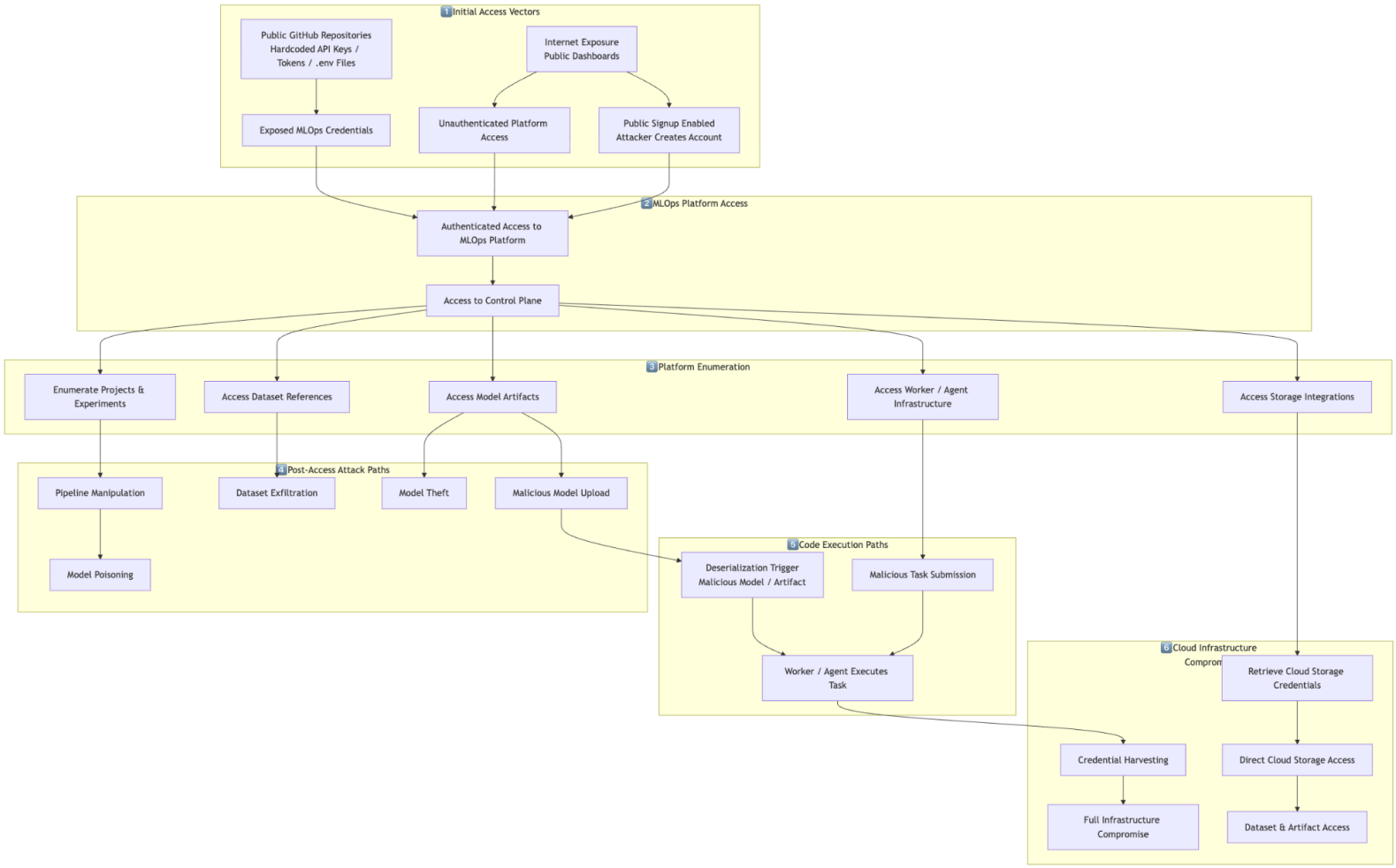
This research was conducted using publicly accessible information and exposed credentials discovered in open-source repositories and internet-accessible systems. Validation was performed in a controlled and passive manner, limited strictly to confirming access and identifying exposed assets.

No modifications were made to any systems, pipelines, datasets, or models. No data was altered, deleted, or exfiltrated beyond what was necessary to confirm exposure.

The purpose of this research is to highlight systemic security risks affecting machine learning infrastructure and to raise awareness about the importance of securing MLOps platforms and associated credentials.

All sensitive details, including platform identifiers, credential values, and organizational information, have been redacted to prevent misuse.

End-to-end attack surface of exposed MLOps platforms



For an interactive view of this flowchart, Please go to this [Mermaid.live link]

References:

https://azure.microsoft.com/en-us/blog/mlops-blog-series-part-4-testing-security-of-secure-machine-learning-systems-using-mlops/

https://iarjset.com/wp-content/uploads/2024/11/IARJSET.2024.111025.pdf

CloudSEK

We Predict Cyber Threats

Monitor. Analyse. Predict.

Secure your Tomorrow,Today!

Request for a Free Demo of our platform:

SCAN ME

OR

Mail us at **info@cloudsek.com**
or visit **https://cloudsek.com**

Gain access to a free trial and
Detailed POC on CloudSEK Platform

Registered Office:

CloudSEK Research Pte Ltd.
51 Chin Swee Rd. #07-12 Manhattan House,
Singapore 169876

Regional Office: United States

CloudSEK Inc.
8 The Green, Ste A, Dover, DE - 19901
United States

Regional Office: India

CloudSEK Information Security Pvt Ltd
16/1, WINGS, Cambridge Rd, Halasuru,
Cambridge Layout, Jogupalya,
Bengaluru, Karnataka, India - 560008

Regional Office: United Kingdom

CloudSEK, 4th floor, Rex House,
4, 12 Regent Street, London,
SW1Y 4PE - United Kingdom