Detecting AI resource-hijacking with Composite Alerts

In recent years, the rise of generative AI services in the cloud has revolutionized various industries, from art and design to healthcare and finance. These advanced systems have the ability to create realistic images, texts, and even music, pushing the boundaries of what technology can achieve. However, with great innovation comes great risk, as these AI services have become targets for attacks in the cloud. Lacework has witnessed this first hand as many of our customers have integrated AI into their services and have subsequently been affected by related attacks.

In this blog, we explore one such attack in the wild, which was detected by our Composite Alerts. In this incident, we first observed reconnaissance followed by privilege escalation with Amazon Web Services (AWS) Management Console access and then subsequent resource-hijacking (T1496) of the AWS Bedrock service. We believe the attacker was able to leverage the issue by first gaining initial access to the customer's AWS environment, likely through stolen or compromised credentials. AWS Bedrock is a fully managed service on AWS that allows developers to create generative AI applications with access to various foundational models (FMs). Many of these models are inherently resource-intensive and expensive to operate. This has made cloud compute even more of a commodity in the same way it has historically been targeted for illicit cryptomining.

Initial access and privilege escalation

In a cloud based attack leveraging stolen credentials, the primary intrusion is often preceded with automated reconnaissance. Attackers typically cast as wide a net as possible, which results in the same indicators across many environments. In the case of the actor documented here, we observed the same user agent in multiple customer environments. This is characterized by the usage of various python request versions and AWS Signature version 4. The following are variations observed in the wild:

```
python-requests/2.20.0 auth-aws-sigv4/0.7

python-requests/2.28.2 auth-aws-sigv4/0.7

python-requests/2.32.2 auth-aws-sigv4/0.7

python-requests/2.31.0 auth-aws-sigv4/0.7
```

In all cases, the initial reconnaissance APIs included `GetCallerIdentity` (STS service), `ListSecrets` (Secrets Manger) and `ListVaults` (S3 Glacier). The first two are commonly seen for malicious reconnaissance; however, the latter is relatively rare. Amazon S3 Glacier is an Amazon S3 storage class for data archiving and long-term backup. A “vault” in Glacier represents a container for archived data. In the context of targeting generative AI services and resources, the motive may be data theft of archived training data or related datasets.

During this automated reconnaissance, Lacework generated a Composite Alert that correlated several weak signals including:

- Impossible travel – two or more source countries in a short period of time
- Anomalous login – previously unseen source IPs
- Anomalous behavior associated with the focal cloud identity
- Usage of APIs associated with cloud discovery and credential access techniques

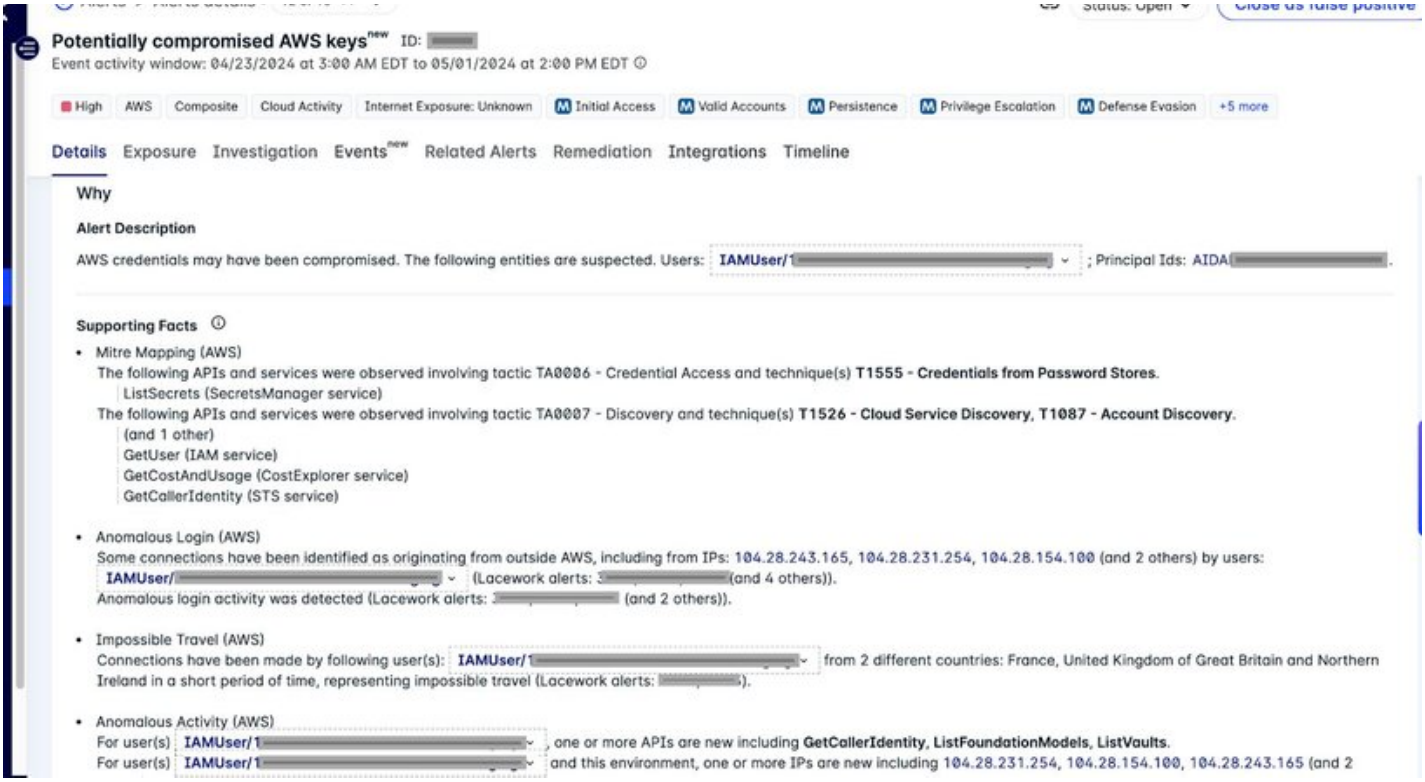

Figure 1: Composite Alert for automated reconnaissance

Each one of these weak signals would be quite noisy if they generated an alert on their own; however, with Composite Alerts, we only alert if these various signals are correlated. This results in much higher accuracy and low false positives. Unfortunately, the activity reported by our first Composite Alert was not fully mitigated so there were follow-on actions taken by the attacker several weeks later. These likewise triggered Composite Alerts; however, there was privilege escalation and subsequent access to the AWS Management Console which was also reported within the alerts.

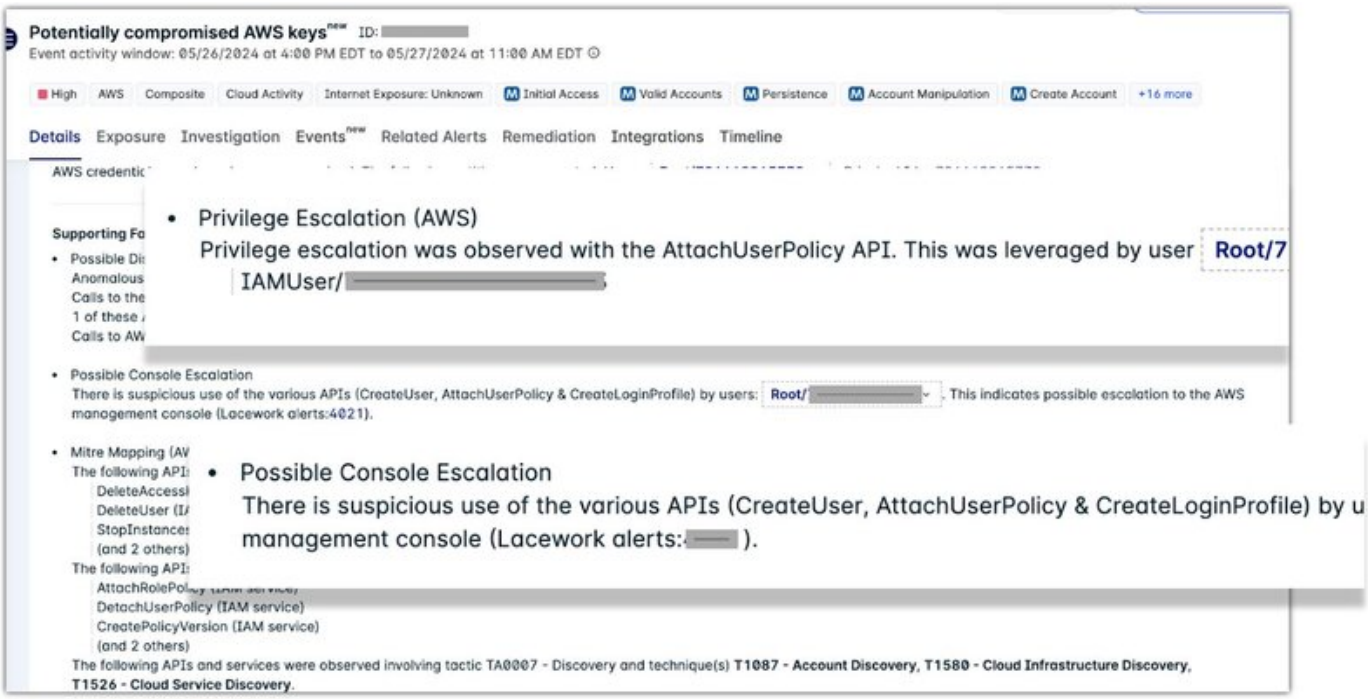

Figure 2: Privilege escalation and console access reported by Composite Alerts

In this case, we observed a common privilege escalation technique (TA0004), which involves the following steps, enabling access to the AWS Management Console:

- `CreateUser` (T1136) – creates new user
- `AttachUserPolicy` (T1098) – used to attach administrative policy to the new user, effectively escalating privileges
- `CreateLoginProfile` – creates a login profile for the new user to allow access to the AWS Management Console, with administrative privileges

Actions on objectives

Upon obtaining access to the AWS Management Console, the actor quickly identified available applications and then started performing reconnaissance against AWS Bedrock. This initial recon was performed within less than a minute of the console login indicating the attacker already knew what they were looking for. Actions resulted in calls to `ListFoundationModels` and `GetFoundationModelAvailability`, respectively enumerating accessible models. Subsequently we observed several write APIs in preparation for using the foundation models. These included:

- `GetUseCaseForModelAccess`
- `PutUseCaseForModelAccess`
- `PutFoundationModelEntitlement`
- `CreateFoundationModelAgreement`

While the activity on the management console was ongoing, we also observed both successful and unsuccessful calls against `InvokeModel` and `InvokeModelWithResponseStream` APIs.The `InvokeModel` API invokes a specified AWS Bedrock model. This allows for generation of text, images, or embeddings. All model invocations were for `anthropic.claude-*` models and occurred from separate infrastructure than what was leveraged for reconnaissance and to access the management console.

One of the traffic sources was identified as an illicit reverse OAI proxy known as “Scylla” (`scylla.dragonetwork.pl`). The site for the proxy currently hosts links to a Discord server, a user board, and contact info for purchasing access with either Bitcoin or Monero. This activity is consistent with “LLMjacking” tactics detailed by Sysdig, specifically the use of a reverse proxy for monetizing hijacked infrastructure.

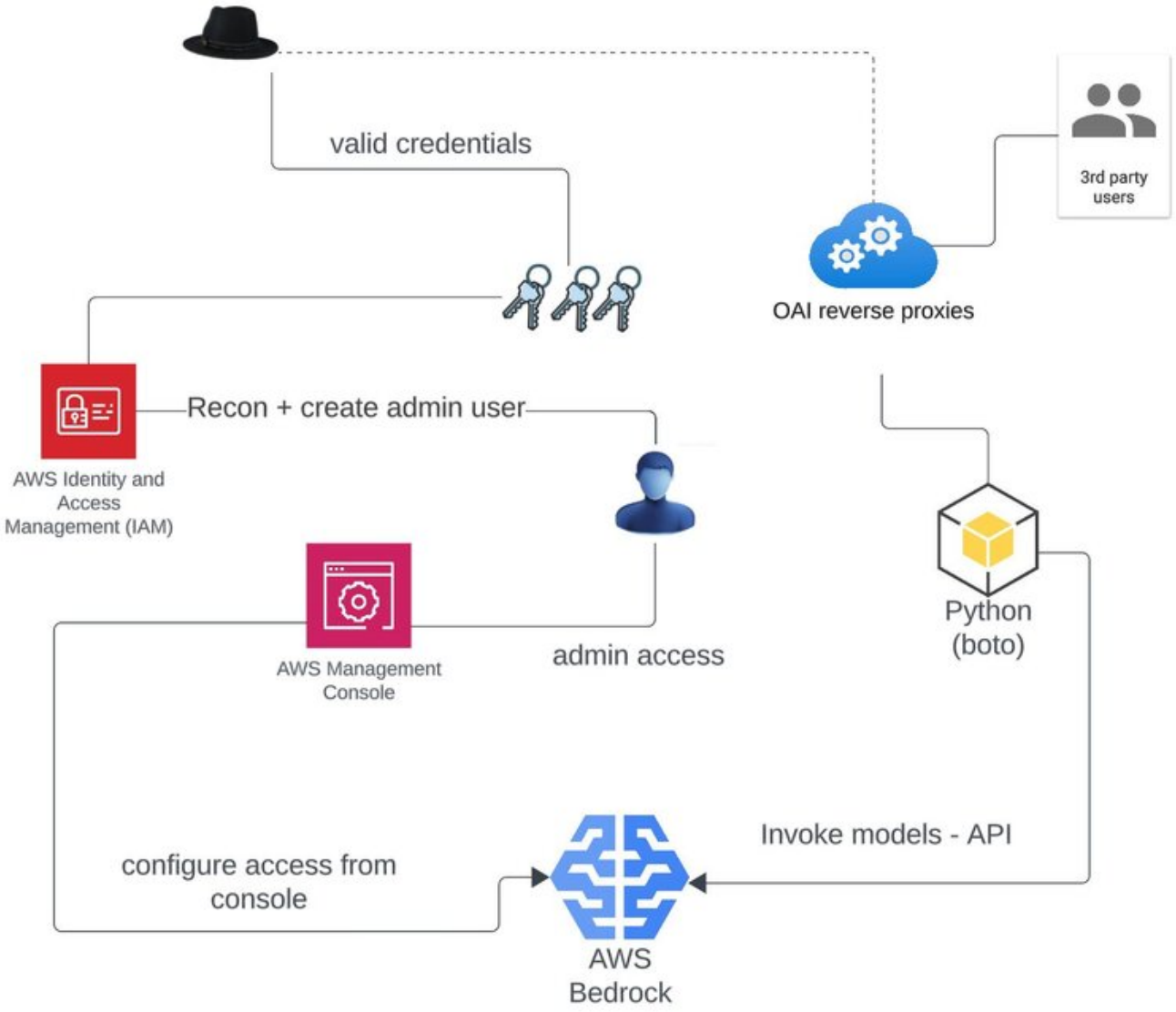

Figure 3: Bedrock attack overview

On the AWS management console end, there were additional calls to `ListProvisionedModelThroughputs`, presumably to gather information on existing inputs and outputs for models in use. Throughput refers to the number and rate of inputs and outputs that a model processes and returns. These throughputs may be purchased and are required in order to leverage customized models. While possibly related, there was also interaction with the AWS Cost Explorer indicating an interest in the target's overall cloud costs. In this context, collecting information on cloud costs is likely part of an effort to track consumption or stay within certain cost parameters and not raise attention to the resource hijacking operation.

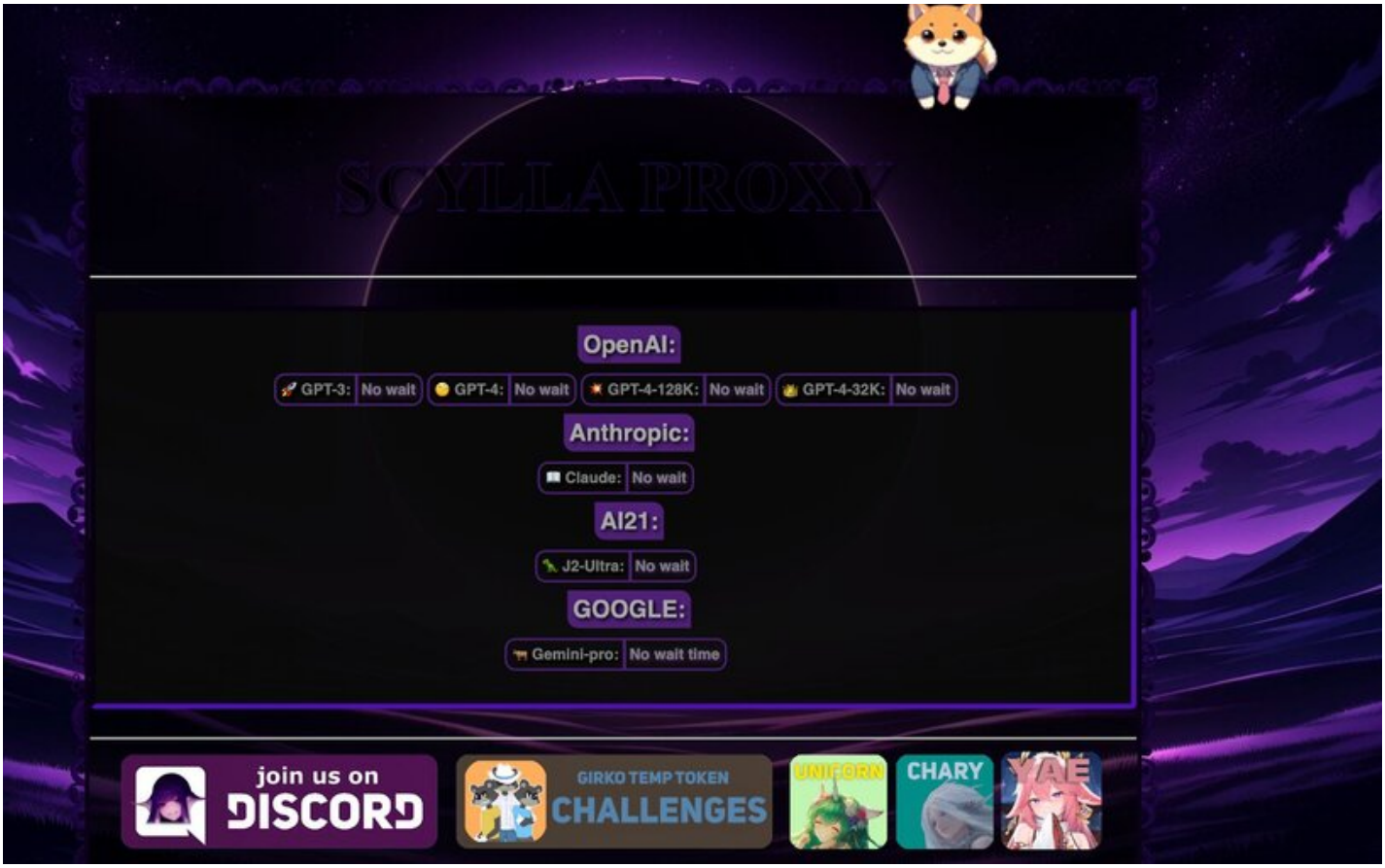

Figure 4: "Scylla" reverse proxy providing access to hijacked AI services

Following this activity, there was further persistence the next day with the creation of an additional admin — using the same tactic described previously. The created user in this incident was a clear reference to a service provided by the targeted customer, indicating targeting or at least familiarity with the environment. This newly created user performed nearly identical tasks as the previous day. Shortly after this, the illicit access to the cloud environment was mitigated so there was no further observed activity. And incidentally, the Anthropic proxy hosted by Scylla is no longer available.

Lessons learned

Regardless of the targeted cloud service, many things remain consistent through the attack chain. In nearly all cases, we observe anomalous behavior, anomalous network activity, plus specific actions necessary to escalate privileges and carry out objectives. Attacks targeting AI services are no different, especially when run on top of the same monitored cloud infrastructure.

The detection challenge however involves correlation of these weak signals which on their own mean very little. It's only when we correlate these together that we can provide much needed context and trigger an alert for our customers.

During the incident documented in this blog, there were several Composite Alerts that were triggered by these correlated anomalies and actions. The first of these represented attacker reconnaissance, and only later did we see higher severity activity. With resource-hijacking incidents such as these this can become especially costly as running various models on AWS Bedrock can incur extremely large cloud costs. This underscores the importance of Lacework anomaly detection as an early warning system because privilege escalation and actions on objectives are often carried out days or even weeks after testing of credentials.

Indicators

Observed AWS Bedrock APIs leveraged by attacker:

Observed Bedrock API	Description	Read/Write/List
CreateFoundationModelAgreement	Grants permission to create a new foundation model agreement	Write
GetFoundationModelAvailability	Grants permission to get the availability of a foundation model	Read
GetUseCaseForModelAccess	Grants permission to retrieve a use case for model access	Read
ListFoundationModelAgreementOffers	Grants permission to get a list of foundation model agreement offers	List
ListFoundationModels	Grants permission to list Bedrock foundation models that you can use	List
ListProvisionedModelThroughputs	Grants permission to list provisioned model throughputs that you created earlier	List
PutFoundationModelEntitlement	Grants permission to put entitlement to access a foundation model	Write
PutUseCaseForModelAccess	Grants permission to put a use case for model access	Write
InvokeModel	Invokes the specified Amazon Bedrock model to run inference using the prompt and inference parameters provided in the request body.	Read
InvokeModelWithResponseStream	Grants permission to invoke the specified Bedrock model to run inference using the input provided in the request body with streaming response	Read

Network indicators

We've observed a total of 51 source IPs since tracking this threat. Many of the IPs are VPNs, CloudFlarenet or both. As such they may not represent good stand-alone indicators of compromise. Other common source networks included 212238 (Datacamp Limited) and 18403 (FPT Telecom Company), with a large number geolocated to Vietnam.

IPs leveraged for reconnaissance and console access:

- `183.80.32.29`
- `104.28.214.18`

- 104.28.237.72
- 148.252.146.75
- 58.187.68.163
- 104.28.212.85
- 104.28.200.2
- 89.187.170.169
- 85.255.235.112
- 104.28.232.1
- 104.28.154.100
- 104.28.205.70
- 104.28.205.72
- 104.28.243.165
- 104.28.200.1
- 1.53.56.66
- 104.28.244.85
- 58.187.189.153
- 104.28.205.71
- 104.28.212.86
- 104.28.237.71
- 58.187.68.220
- 27.65.42.168
- 104.28.199.254
- 58.187.68.218
- 212.102.51.245
- 104.28.246.18
- 104.28.232.21
- 104.28.242.246
- 104.28.232.6
- 104.28.205.73
- 104.28.231.254
- 104.28.232.20
- 104.28.237.70
- 37.19.205.195
- 42.118.236.39
- 104.28.246.17
- 58.187.68.217
- 104.28.244.86
- 104.28.214.17
- 183.80.38.101
- 193.107.109.72

IPV6:

- 2a09:bac5:37aa:d2::15:1c2
- 2a09:bac5:382a:ebe::178:11f
- 2a09:bac5:37aa:165a::23a:32
- 2a09:bac5:376a:1f19::319:66

Reverse proxies leveraged for invoking models:

- 54.243.246.120
- 54.80.185.234
- 51.75.163.93

Learn more about Composite Alerts

Lacework Composite Alerts detect malicious activity by automatically tying together low severity signals. Read our feature brief to learn more.

Suggested for you