Tales from the cloud trenches: The Attacker doth persist too much, methinks | Datadog Security Labs

By Martin McCloskey

Published: 2025-05-13 · Archived: 2026-04-06 01:56:20 UTC

As a result of a recent threat hunt, we observed attacker activity originating from a leaked long-term AWS access key (`AKIA*`). Within a 150-minute period, we detected five distinct IP addresses attempting to leverage this access key to perform malicious techniques, tactics, and procedures (TTPs). This post presents several techniques that, to our knowledge, have never been reported in the wild.

Key points and observations

- A long-term AWS access key associated with an IAM user in an AWS organization management account was exposed.
- We observed follow-up activity from this access key for a number of tactics, including both common and innovative ones.
- Previously unreported tactics involve creating "persistence-as-a-service" infrastructure, creating AWS Identity Center users, and disabling organization-level services.

Routine attacker tactics

We observed several tactics that attackers commonly use in cloud intrusions. We list them below for the sake of completeness but don't analyze them in further detail:

- SES enumeration through API calls such as GetAccount, ListIdentities, and GetSendQuota.
- Attempt to create an EC2 security group called `Administratorsz` with the description `We Are There But Not Visible` , which has been attributed to the JavaGhost group.
- Creation of several IAM users, subsequently granted administrative permissions either directly through `AttachUserPolicy` or indirectly through `AttachGroupPolicy` . The attacker sometimes attempted to create a login profile on the IAM user to facilitate using the AWS console.
- Generating temporary STS credentials from long-lived access keys, which allows an attacker to authenticate to the AWS console even from long-lived credentials.

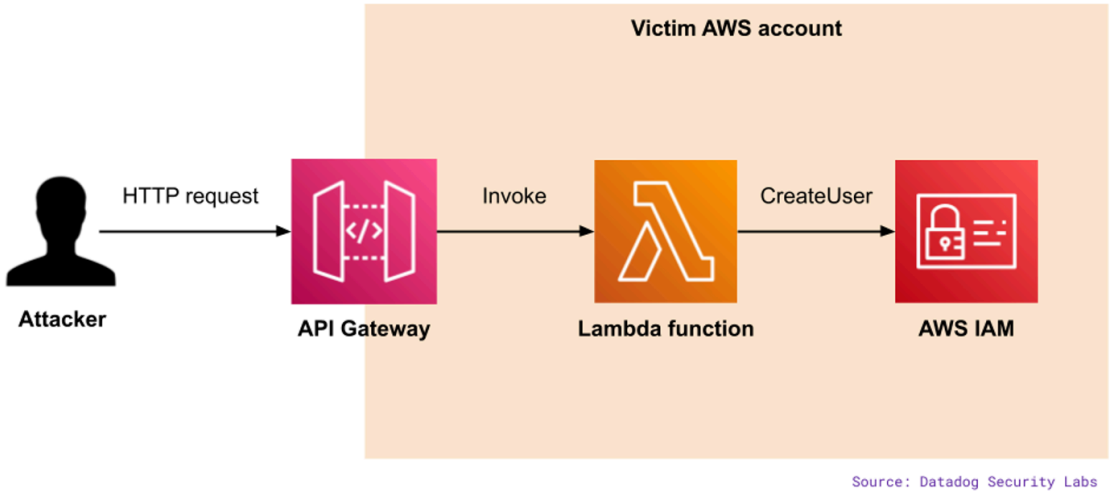
Notable tactics

Besides the common techniques listed above, we observed new tactics that have never been reported before (to the best of our knowledge).

Persistence as a service with API Gateways and Lamba function

In one case, the attacker created a Lambda function named `buckets555` and attached its execution role to a new policy `AWSLambdaBasicExecutionRole-b69e3024-5a7f-4fff-a576-cf54fc986b93`. They then created an HTTP API Gateway, and a Lambda function trigger so the function would automatically get invoked when an HTTP request to a specific URL is sent. We later determined that this Lambda function ran code with the capability to create IAM users dynamically, on demand.

This effectively creates a "persistence-as-a-service" mechanism: The attacker, even after the compromised credentials are revoked, is able to perform external HTTP requests to the API Gateway and dynamically create further malicious IAM users.



Source: Datadog Security Labs

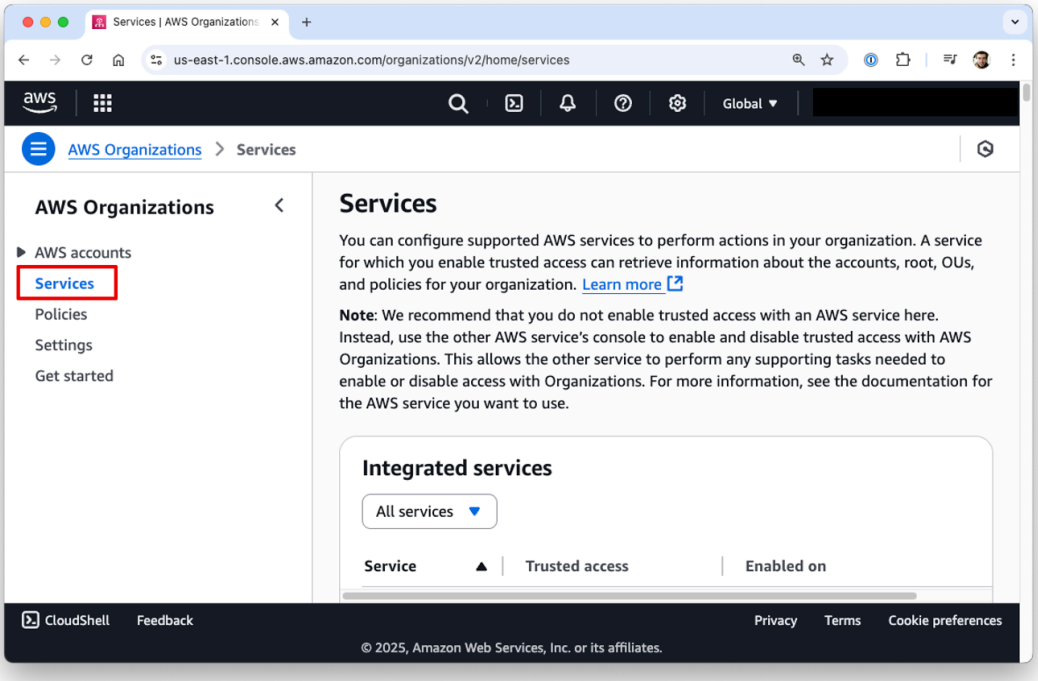
ConsoleLogin events from Telegram IP addresses

As part of this attack, we identified several `ConsoleLogin` events in a short amount of time from the IP address `149.154.161[.]235`, which belongs to the ASN `Telegram Messenger Inc`. This indicates that part of the attacker's operations are based on Telegram.

At first sight, it may seem unusual that the ConsoleLogin events themselves would originate from the Telegram IP space. We believe that after compromising long-lived credentials, the attacker may have a Telegram bot automatically generating sign-in URLs for the AWS console. The Telegram preview service would then follow this link and generate ConsoleLogin events.

Disabling trusted access for organization-level services

In one case, the attacker authenticated to the AWS Console and navigated to the **Services** tab under the AWS Organizations service and began to disable the integration of six AWS services.



In CloudTrail, this is recorded with the API call DisableAWSServiceAccess. The attacker disabled trusted access for the following services:

- `access-analyzer.amazonaws.com` (IAM Access Analyzer)
- `account.amazonaws.com and am.amazonaws.com` (AWS Account Management)
- `member.org.stacksets.cloudformation.amazonaws.com` (CloudFormation StackSets)
- `ssm.amazonaws.com` (AWS Systems Manager)
- `tagpolicies.tag.amazonaws.com` (Tag Policies)

We were unable to discern what the attacker's intent was with this action, as this only affects new AWS accounts, and the order in which the attacker disabled these services is the way they are presented in the AWS console. One theory is that the attacker intended to eventually add a new AWS account to the organization, which may have allowed them to evade some security controls so they could act on their objective.

Persistence through AWS Identity Center (AWS SSO)

AWS Identity Center is a cloud-based identity and access management solution that enables centralized user access control across AWS accounts and integrated applications. Actions taken in Identity Center require access to an organization's management AWS account.

We observed the attacker enumerating the SSO instance to look at SSO configurations, users, groups, and applications. Afterward, they created a group called `secure` and a user called `Secret`, which the attacker added to their group, and assigned a new permission set to that group.

Following this, the attacker updated two configuration options within the SSO instance. First, they modified the MFA configuration of the SSO instance to allow themselves to sign in without MFA. They then extended the

session duration for Amazon Q Developer to 90 days, indicating a likely intent to leverage this service in the future.

Later, we observed a successful sign-in event associated with a password-only sign-in flow for the newly created user `Secret`.

Summary of attacker activity

TA0001 - Initial Access
- T1078.004 - Valid Accounts

TA0007 - Discovery
- T1078.004 - Valid Accounts
- T1526 - Cloud Service Discovery

TA0003 - Persistence
- T1098.001- Additional Cloud Credentials
- T1098.003- Additional Cloud Roles
- T1036.003 - Cloud Account

TA0006 - Credential Access
- T1556.006 - Multi-Factor Authentication

TA0040 - Impact
- T1485 - Data Destruction

Detection opportunities

Here are some suggestions to help to identify this type of activity:

- Identify creation/modification actions of a login profile.
- Identify the attachment of the managed policy `arn:aws:iam::aws:policy/AdministratorAccess` and `arn:aws:iam::aws:policy/AmazonSESFullAccess`.
- Identify unusual console logins from unexpected networks like Telegram.
- Identify attempts to enumerate AWS SES settings and configurations.
- Activity from a long-term access key is generally rare.
- Identify attempts to create a new IAM user from Lambda.
- The user agent will contain the string `exec-env/AWS_Lambda`.
- Identify updates to your AWS IAM Identity Center configuration.
- Look for changes to MFA settings `requestParameters.configurationType:APP_AUTHENTICATION_CONFIGURATION`.
- Identify `GetFederationToken` API calls with a highly privileged policy attached.
- Identify `DisableAWSServiceAccess` API calls disabling the integration of AWS services.
- Identify the deletion of a high number of Lambda functions.

- Identify EC2 security group creations with the name `Java_Ghost` or description `We Are There But Not Visible`.

All of these detection ideas should be assessed within the context of your environment.

How Datadog can help

Datadog Cloud SIEM and Cloud Security Management (CSM) come with the following out-of-the-box rules to identify suspicious activity relevant to these attacks in an AWS environment. The Cloud SIEM rules help identify potential threats, while the CSM rules help identify overprivileged identities. Long-lived access keys tend to carry a higher risk of being associated with a compromise.

- AWS SES discovery attempt by long-term access key
- Possible privilege escalation via AWS login profile manipulation
- AWS IAM Identity Center SSO configuration updated
- Anomalous number of AWS Lambda functions deleted
- Temporary AWS security credentials generated for user
- AWS IAM AdministratorAccess policy was applied to a user
- AWS console login without MFA
- AWS Java Ghost security group creation attempt
- AWS IAM AmazonSESFullAccess policy was applied to a user
- AWS IAM AdministratorAccess policy was applied to a group
- AWS IAM User created with AdministratorAccess policy attached
- Amazon SES enumeration attempt by previously unseen user
- IAM users should not have both Console access and Access Keys
- IAM users should not have the 'AdministratorAccess' policy attached
- Multi-factor authentication should be enabled for all IAM users with console access

Indicators of compromise

IP Addresses used

```
 129.146.24[.]173
134.199.148[.]132
103.131.213[.]89
80.85.141[.]238
54.95.125[.]167
149.154.161[.]235
103.131.213[.]89
182.185.156[.]45
```

Created IAM user names:

```
 adminslabs
buckets488
s3s684
git-lab965
git-lab555
```

Created IAM role names:

```
 LambdaExecutionRole
buckets555-role-c6s4hhdi
curdfunctionsme-role-zw1zxamc
```

Created IAM group name:

```
 Administrators
```

Created Lambda function names:

```
 buckets555
curdfunctionsme
```

Lambda function SHA256:

```
 HAPq9EReJVEC5gLavtc/gyd5vZtd9eiUGF932t0jBxY= (1c03eaf4445e255102e602dabed73f832779bd9b5df5e894185f77dadd230716)
HXGHpm9uGbfTRsBh2YwHKSlF5xxwrAggliHsuoD3OGY= (1d7187a66f6e19b7d346c061d98c07292945e71c70ac08209621ecba80f73866)
```

Created IAM Identity Center user name:

```
 Secret
```

Created IAM Identity Center group name:

```
 secure
```

Source: https://securitylabs.datadoghq.com/articles/tales-from-the-cloud-trenches-the-attacker-doth-persist-too-much/