

AWS re:Inforce

JUNE 10 - 12, 2024 | PHILADELPHIA, PA

TDR 432

New tactics and techniques for proactive threat detection

Ben Fletcher

(he/him)

AWS EMEA CIRT LEAD
Amazon Web Services

Steve de Vera

(he/him)

AWS CIRT Manager
Amazon Web Services



Agenda

- About AWS CIRT
- Statistics
- Current threat actor tactics
- New threat actor tactics
- Security best practices

THIS SESSION IS INTERACTIVE!

**Feel free to ask questions,
make comments, participate**

About AWS CIRT



© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Customer Incident Response Team (CIRT)

A specialized team that assists and advises customers during suspected active security events, on the **customer's side** of the **AWS Shared Responsibility Model**

Respond



Global team 24/7, follow-the-sun model

Recover



Assist and advise customers with active triage and recovery from their security event on AWS

Learn



Assist in root cause analysis of a customer's AWS service logs for their active security event

Educate



Provide advice to customers for long-term recovery from their active security event

Statistics



Threat actors use which initial access method most often?

Cross-account permissions

#1

Vulnerable web apps

#2

Brute force

#3

Lost/leaked access keys/credentials

#4

Open S3 buckets

#5

DDoS

#6

Threat actors use which initial access method most often?

Lost/leaked access
keys/credentials

#4

Threat actors use which initial access method most often?

Lost/leaked access
keys/credentials

#4



66%

valid IAM credentials

Threat actors use which initial access method most often?

Lost/leaked access
keys/credentials

#4



66%

valid IAM credentials



1/3

of those are **root credentials**
[20% of all initial access method use]

Threat actors use which initial access method most often?

Lost/leaked access
keys/credentials

#4

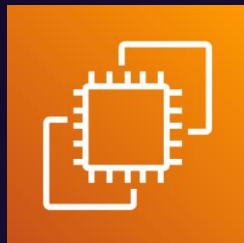


66%

valid IAM credentials

↑
1/3

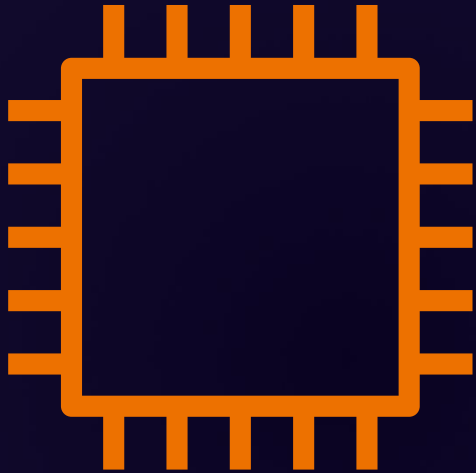
of those are **root credentials**
[20% of all initial access method use]



13%

Public-facing EC2 instance

Threat primary tactics



Resource hijack



Ransom events



Opportunistic
destruction

A zero trust strategy

Get the keys

MITRE ATT&CK
Tactic: Initial access
Technique: Valid cloud credentials

Access key best practices & alternatives

Info

Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.

Use case

☒ Command Line Interface (CLI)

You plan to use this access key to enable the AWS CLI to access your AWS account.

☐ Local code

You plan to use this access key to enable application code in a local development environment to access your AWS account.

☐ Application running on an AWS compute service

You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.

☐ Third-party service

You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.

☐ Application running outside AWS

You plan to use this access key to authenticate workloads running in your data center or other infrastructure outside of AWS that needs to access your AWS resources.

☐ Other

Your use case is not listed here.

⚠ Alternatives recommended

• Use [AWS CloudShell](#), a browser-based CLI, to run commands. [Learn more](#)

• Use the [AWS CLI V2](#) and enable authentication through a user in IAM Identity Center. [Learn more](#)

Confirmation

☐ I understand the above recommendation and want to proceed to create an access key.

Cancel

Next

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

If keys are posted on GitHub, how long until they are used?

~Two weeks

#1

~One week

#2

24 hours

#3

4 hours

#4

Minutes, if not
seconds

#5

GitHub

MITRE ATT&CK

Tactic: Initial access

Technique: Valid cloud credentials

Minutes, if not
seconds

#5



Prevent secret leaks.

```
→ ~/my_project git:(branch_name) git push  
remote: error GH009: Secrets detected!  
This push failed.
```

```
this push failed.  
remote: error GH009: secrets detected!
```

<https://thehackernews.com/2024/03/github-rolls-out-default-secret.html>

Current threat actor tactics

DISCLAIMER:

**Tactics and techniques presented
do not constitute vulnerabilities
within AWS**

Resource hijacking: Premise

MITRE ATT&CK

Tactic: Impact

Technique: Resource hijacking

Resource hijacking: Premise

MITRE ATT&CK

Tactic: Impact

Technique: Resource hijacking

- 1) Threat actor obtains access to AWS account or hosted resource

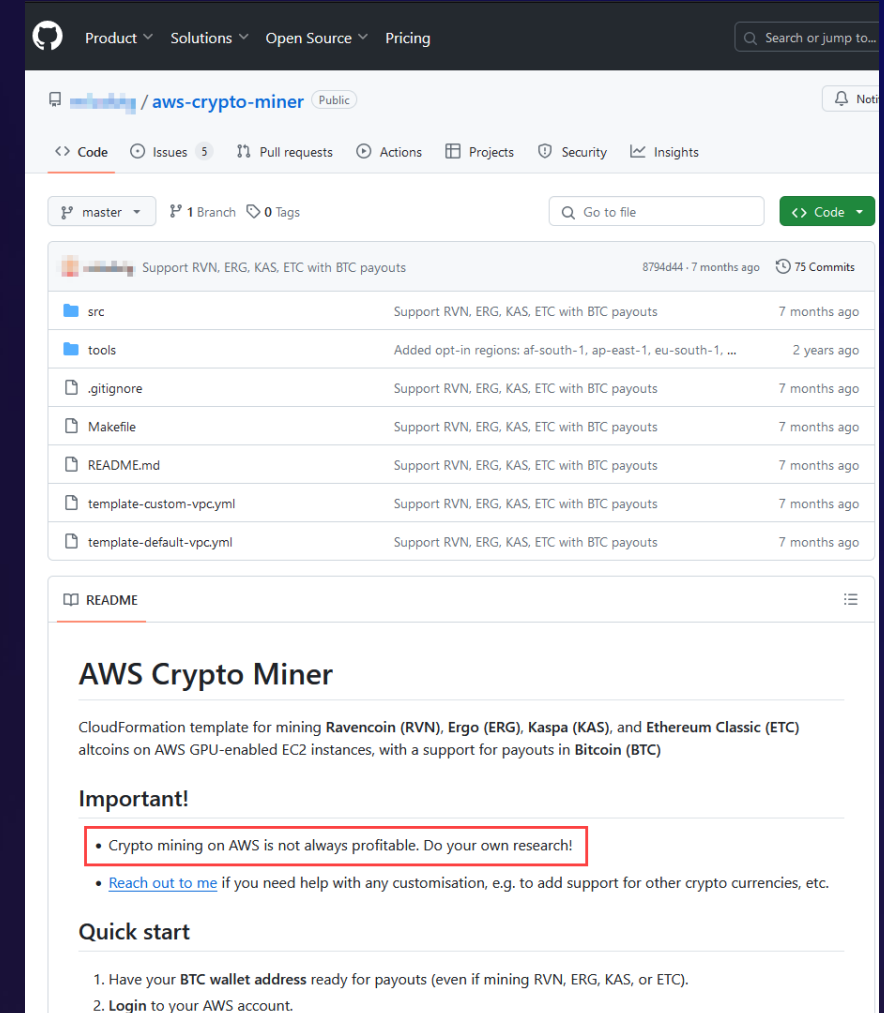
Resource hijacking: Premise

- 1) Threat actor obtains access to AWS account or hosted resource
- 2) Threat actor will mine cryptocurrency from the resource

MITRE ATT&CK

Tactic: Impact

Technique: Resource hijacking



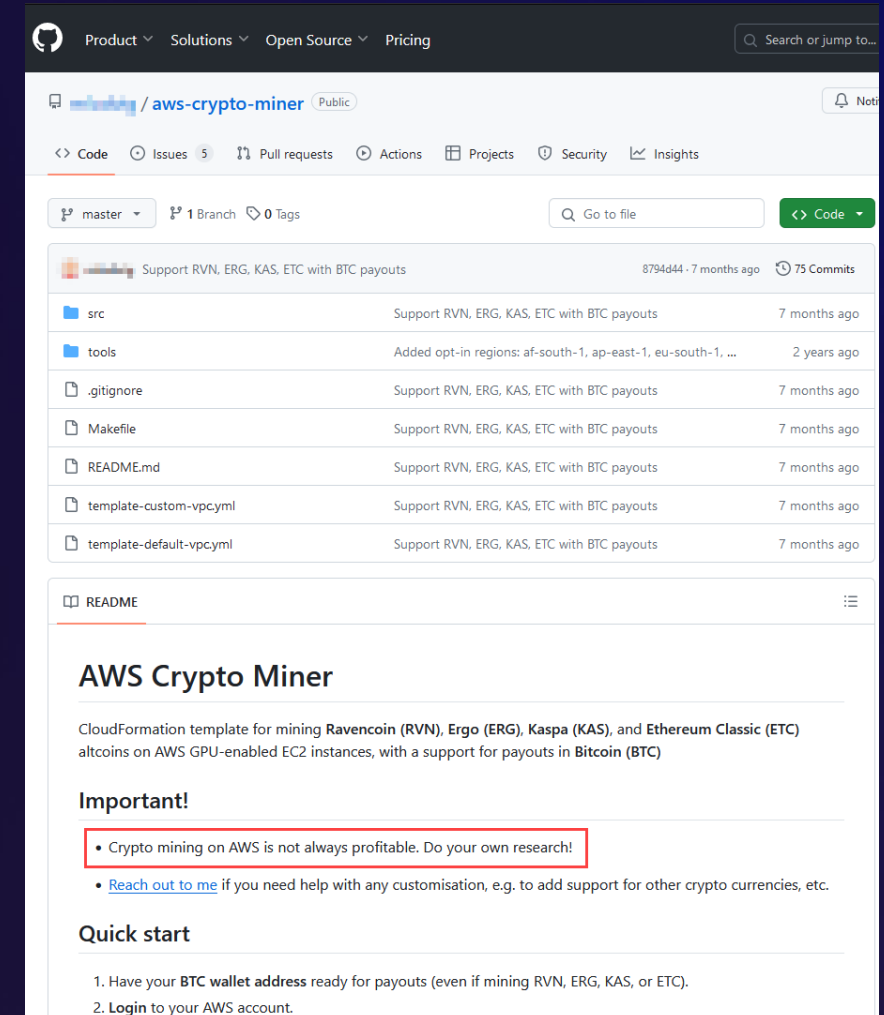
Resource hijacking: Premise

MITRE ATT&CK

Tactic: Impact

Technique: Resource hijacking

- 1) Threat actor obtains access to AWS account or hosted resource
- 2) Threat actor will mine cryptocurrency from the resource
- 3) Resources created in AWS account:
 - RunInstances
 - CreateStack
 - CreateCluster



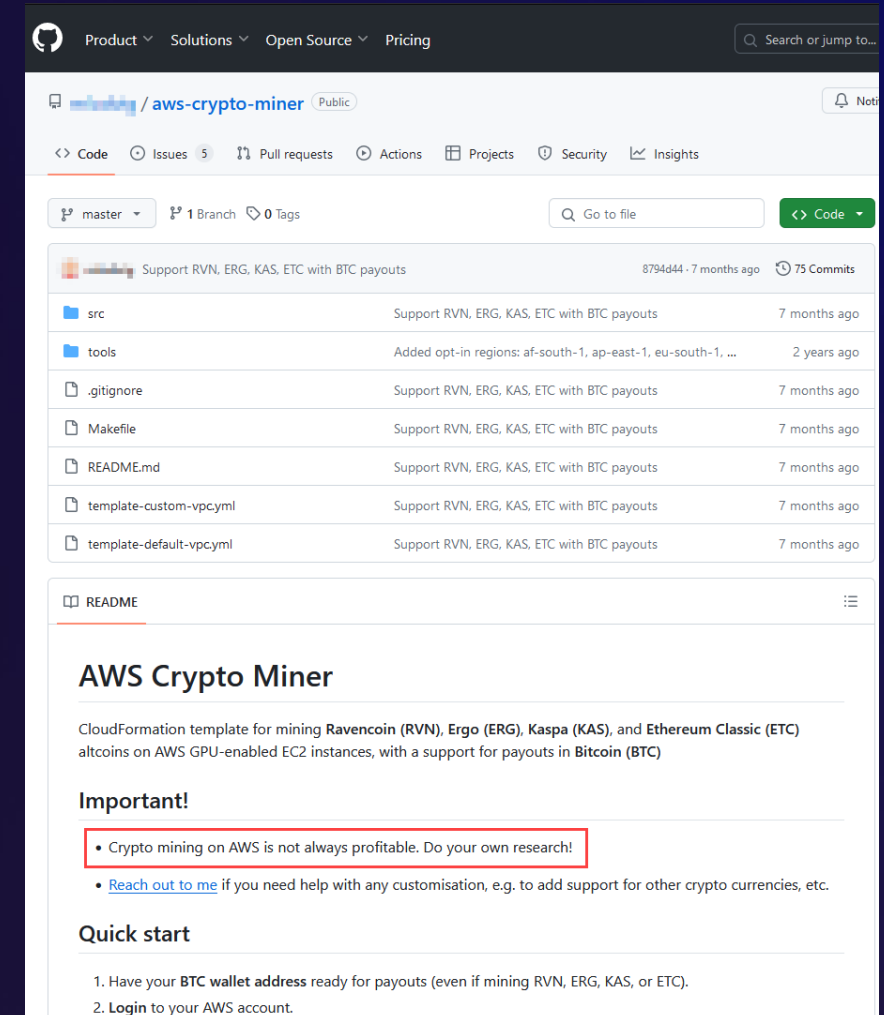
Resource hijacking: Premise

MITRE ATT&CK

Tactic: Impact

Technique: Resource hijacking

- 1) Threat actor obtains access to AWS account or hosted resource
- 2) Threat actor will mine cryptocurrency from the resource
- 3) Resources created in AWS account:
 - RunInstances
 - CreateStack
 - CreateCluster
- 4) Resources created in unused AWS Regions



Resource hijacking: Mitigations

MITRE ATT&CK

Tactic: Impact

Technique: Resource hijacking

- Use SCPs to prevent resource creation – especially in unused Regions
- Apply principle of least privilege to assigned permissions

SubDomain takeover: Premise

MITRE ATT&CK

Tactic: Impact

Technique: Defacement

SubDomain takeover: Premise

MITRE ATT&CK

Tactic: Impact

Technique: Defacement

- 1) Customer has CNAME pointing to a resource (S3 bucket, EC2 instance, Elastic IP)

SubDomain takeover: Premise

MITRE ATT&CK

Tactic: Impact

Technique: Defacement

- 1) Customer has CNAME pointing to a resource (S3 bucket, EC2 instance, Elastic IP)
- 2) The resource is deleted, but the CNAME still exists

SubDomain takeover: Premise

MITRE ATT&CK

Tactic: Impact

Technique: Defacement

- 1) Customer has CNAME pointing to a resource (S3 bucket, EC2 instance, Elastic IP)
- 2) The resource is deleted, but the CNAME still exists
- 3) Threat actor creates a resource that the CNAME still points to

SubDomain takeover: Premise

SubDomain takeover: Premise



Customer

SubDomain takeover: Premise

S3 bucket

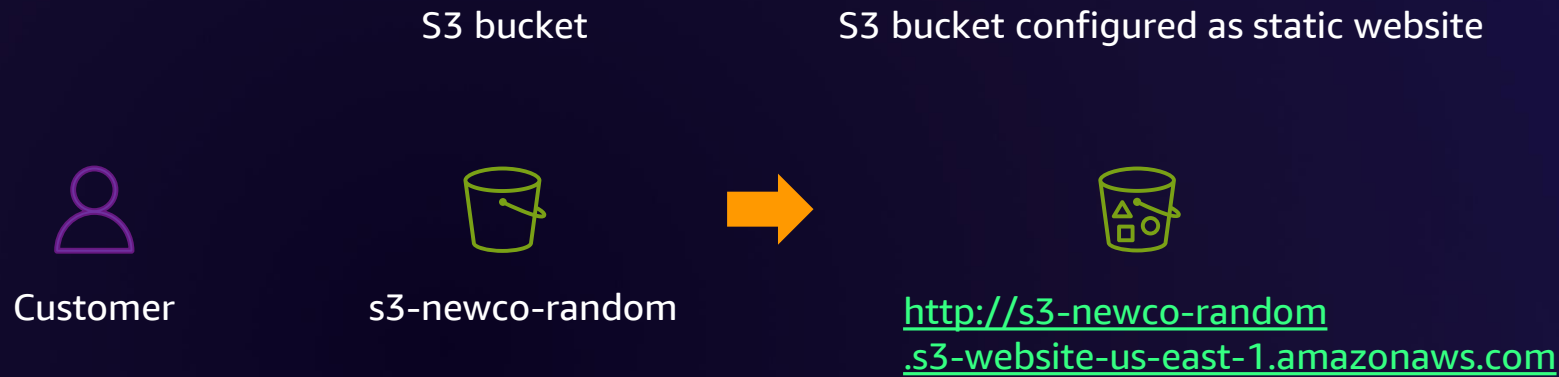


Customer

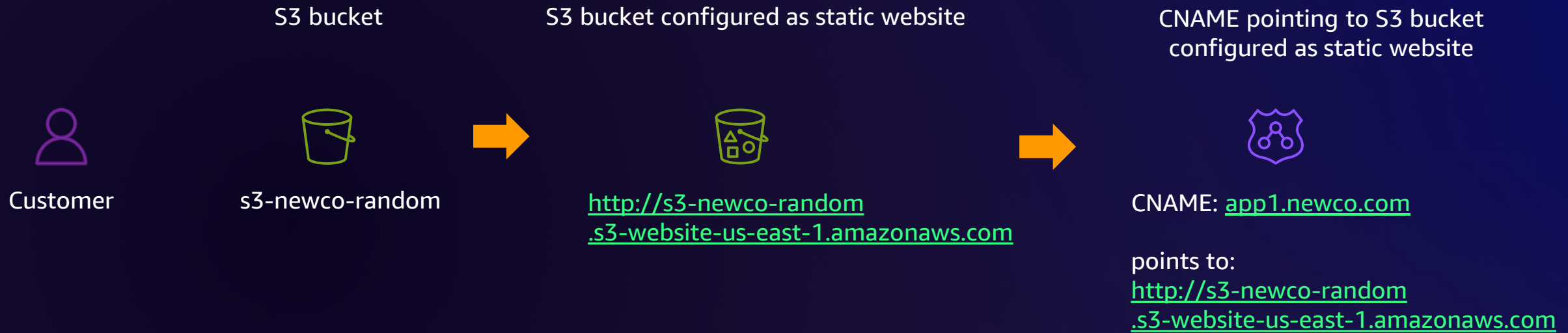


s3-newco-random

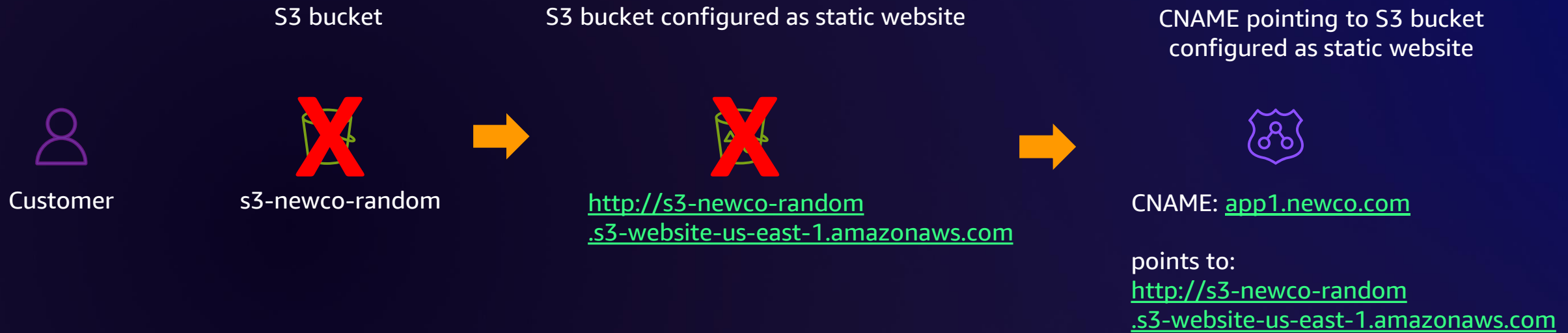
SubDomain takeover: Premise



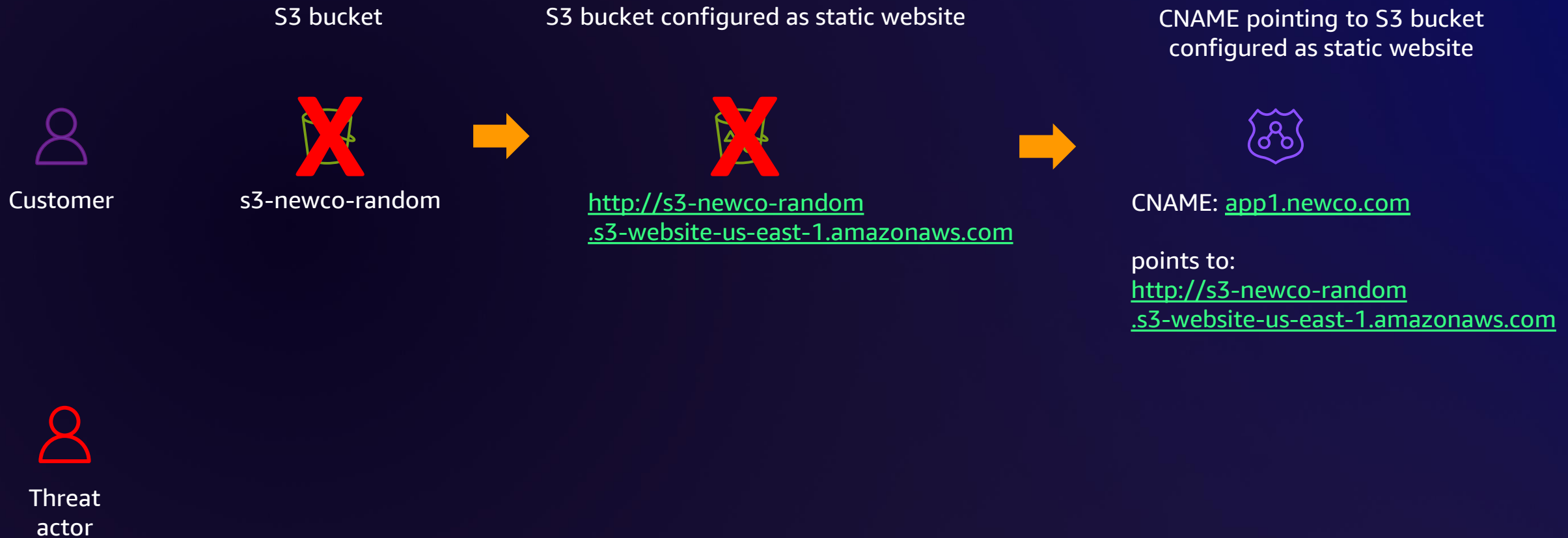
SubDomain takeover: Premise



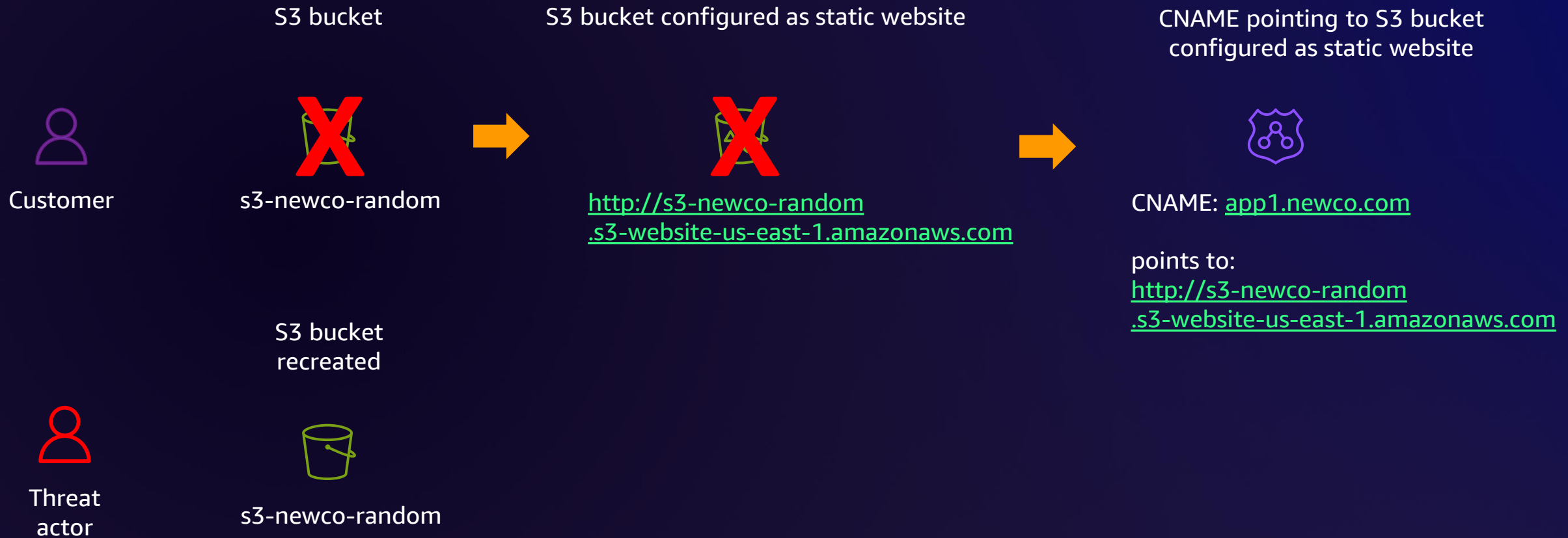
SubDomain takeover: Premise



SubDomain takeover: Premise



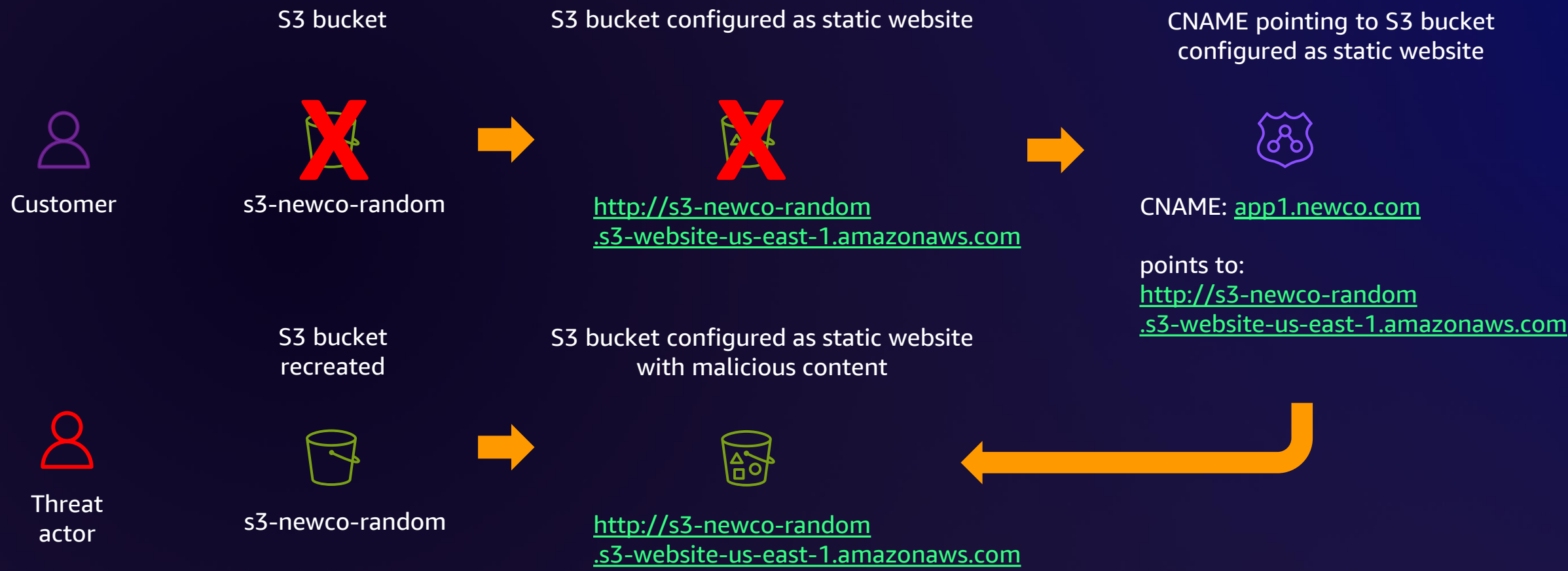
SubDomain takeover: Premise



SubDomain takeover: Premise



SubDomain takeover: Premise



SubDomain takeover: Mitigations

- Review hosted zones and delete unused CNAMEs
- When de-provisioning, remove CNAMEs first

Data destruction: Premise

MITRE ATT&CK

Tactic: Impact

Technique: Data destruction

Data destruction: Premise

MITRE ATT&CK

Tactic: Impact

Technique: Data destruction

- 1) Threat actor obtains access to AWS account or resource (Amazon S3 or Amazon RDS)

Data destruction: Premise

MITRE ATT&CK

Tactic: Impact

Technique: Data destruction

- 1) Threat actor obtains access to AWS account or resource (Amazon S3 or Amazon RDS)
- 2) Threat actor will attempt to delete resources or data

Data destruction: Premise

MITRE ATT&CK

Tactic: Impact

Technique: Data destruction

- 1) Threat actor obtains access to AWS account or resource (Amazon S3 or Amazon RDS)
- 2) Threat actor will attempt to delete resources or data
- 3) Resources deleted in AWS account:
 - DeleteBucket
 - DeleteObject
 - DeleteDBInstance
 - DeleteDBCluster
 - DeleteDBSnapshot
 - AuthorizeSecurityGroupIngress

Data destruction: Mitigations

MITRE ATT&CK

Tactic: Impact

Technique: Data destruction

- Apply and review policies (resource policies and lifecycle policies), S3 Object Lock
- Principle of least privilege
- Use and test backup methodologies

IMDSv1 credential access: Premise

MITRE ATT&CK

Tactic: Credential access

Technique: Unsecured credentials

IMDSv1 credential access: Premise

MITRE ATT&CK

Tactic: Credential access

Technique: Unsecured credentials

- 1) Threat actor obtains ability to obtain IMDSv1 credentials from resource

IMDSv1 credential access: Premise

MITRE ATT&CK

Tactic: Credential access

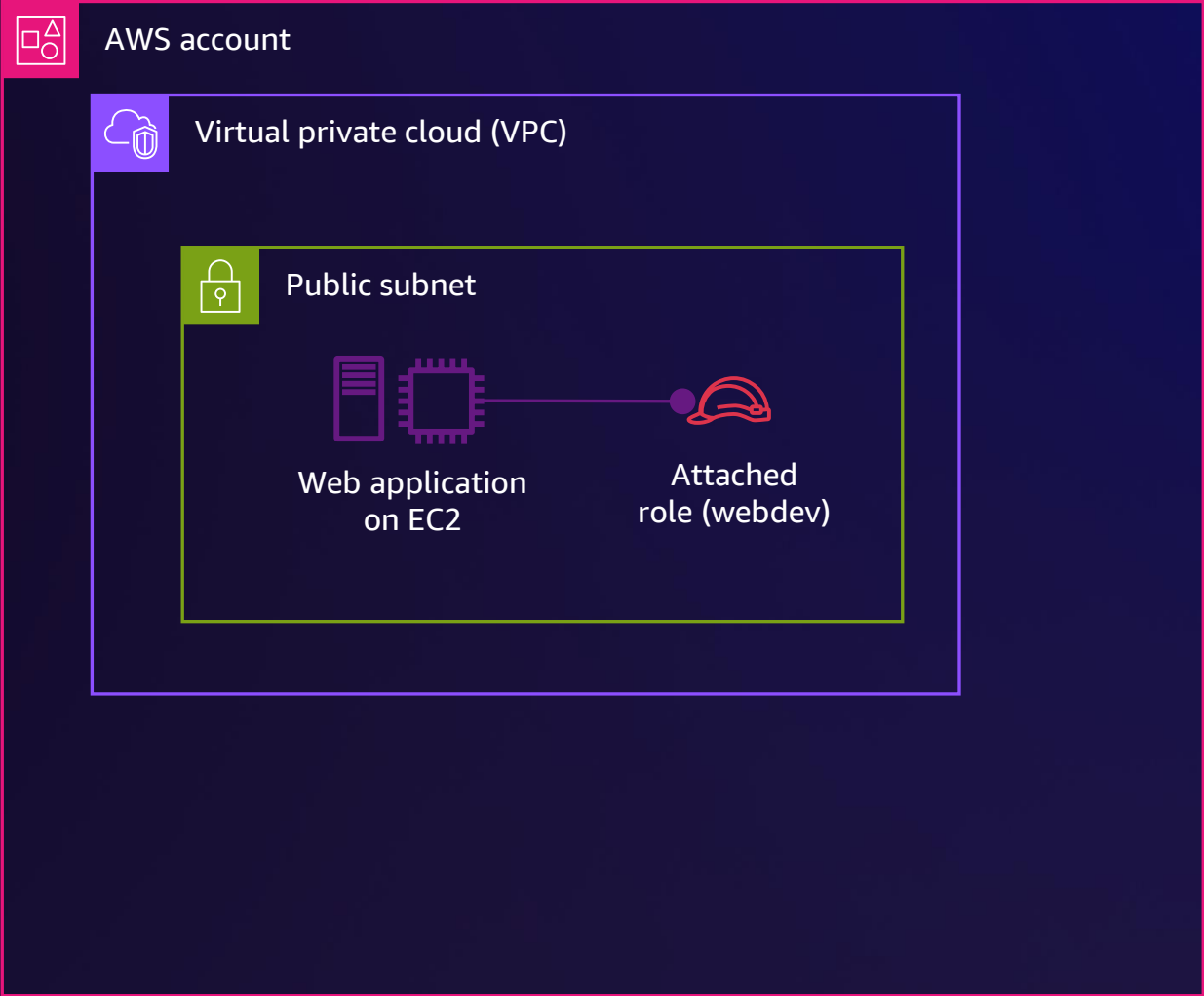
Technique: Unsecured credentials

- 1) Threat actor obtains ability to obtain IMDSv1 credentials from resource
- 2) Threat actor exports and uses credentials

IMDSv1 credential access: Premise



Threat actor

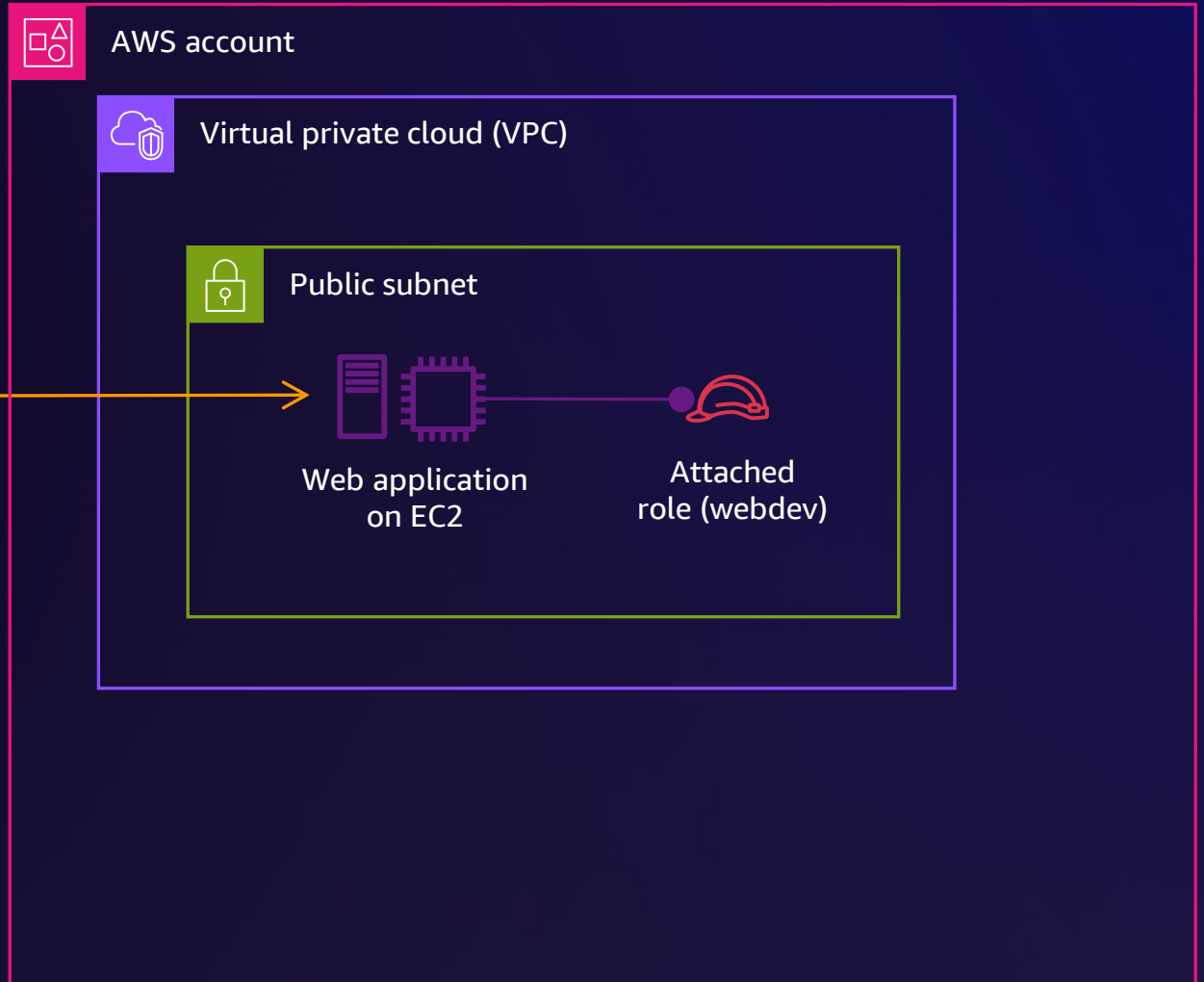


IMDSv1 credential access: Premise

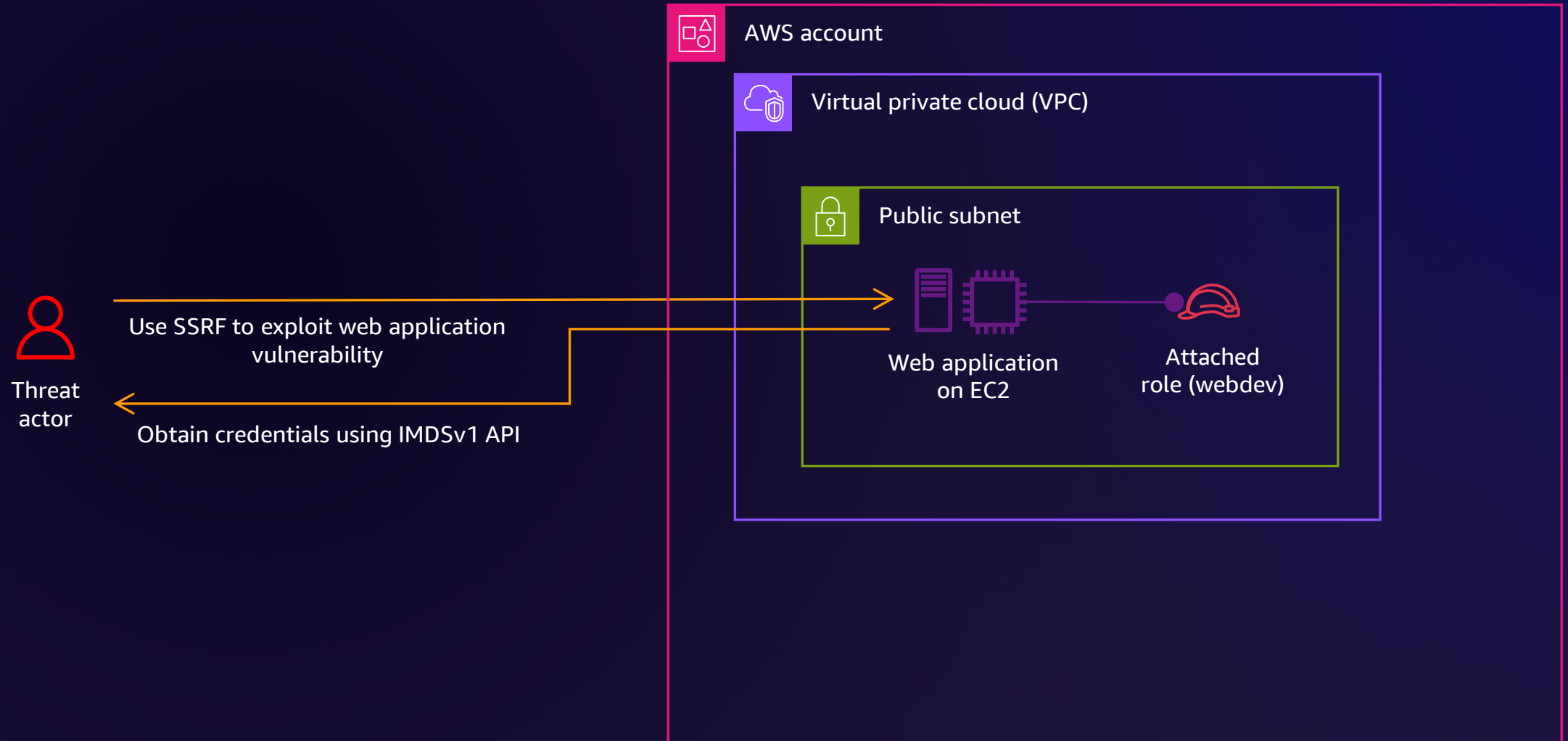


Threat actor

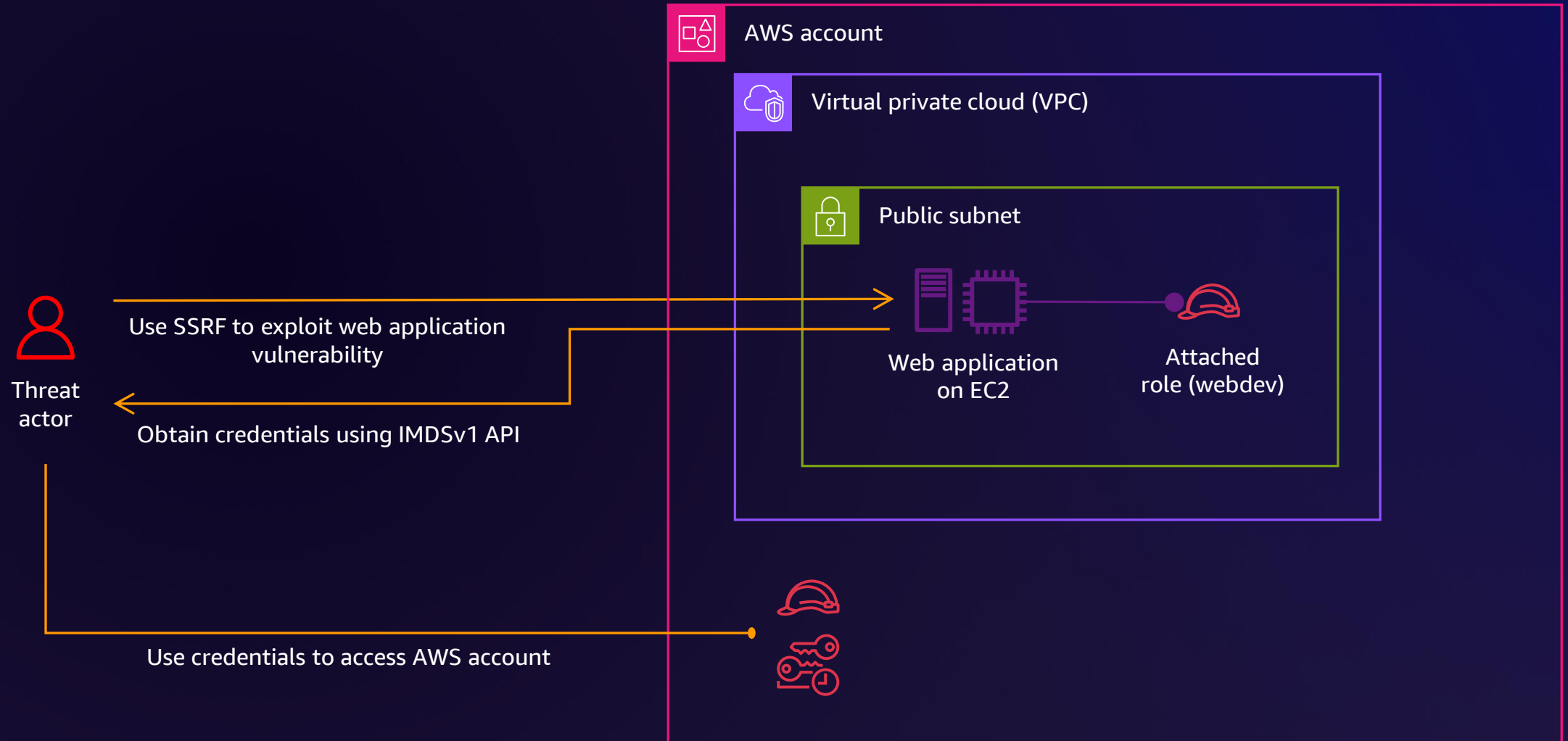
Use SSRF to exploit web application vulnerability



IMDSv1 credential access: Premise

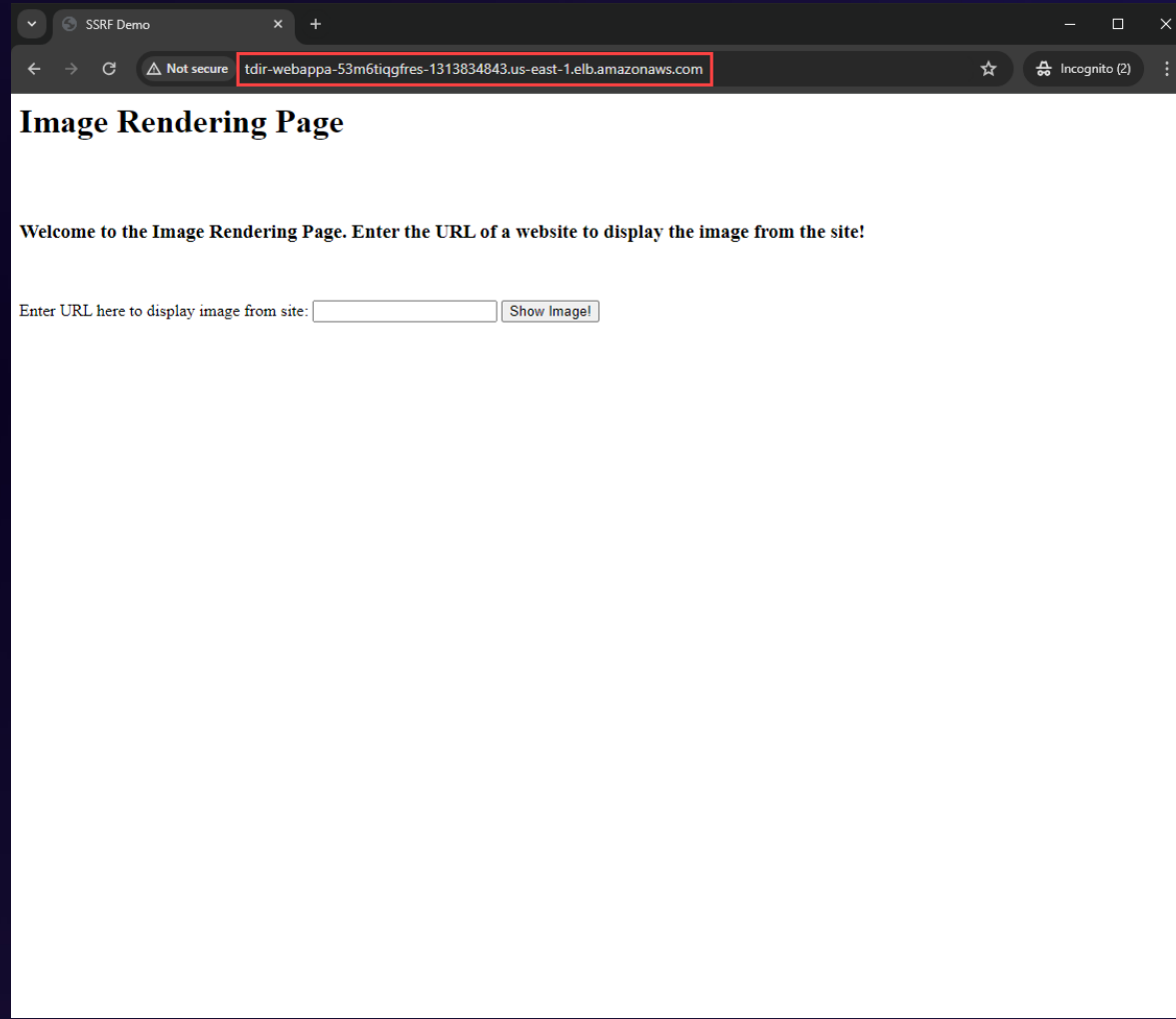


IMDSv1 credential access: Premise

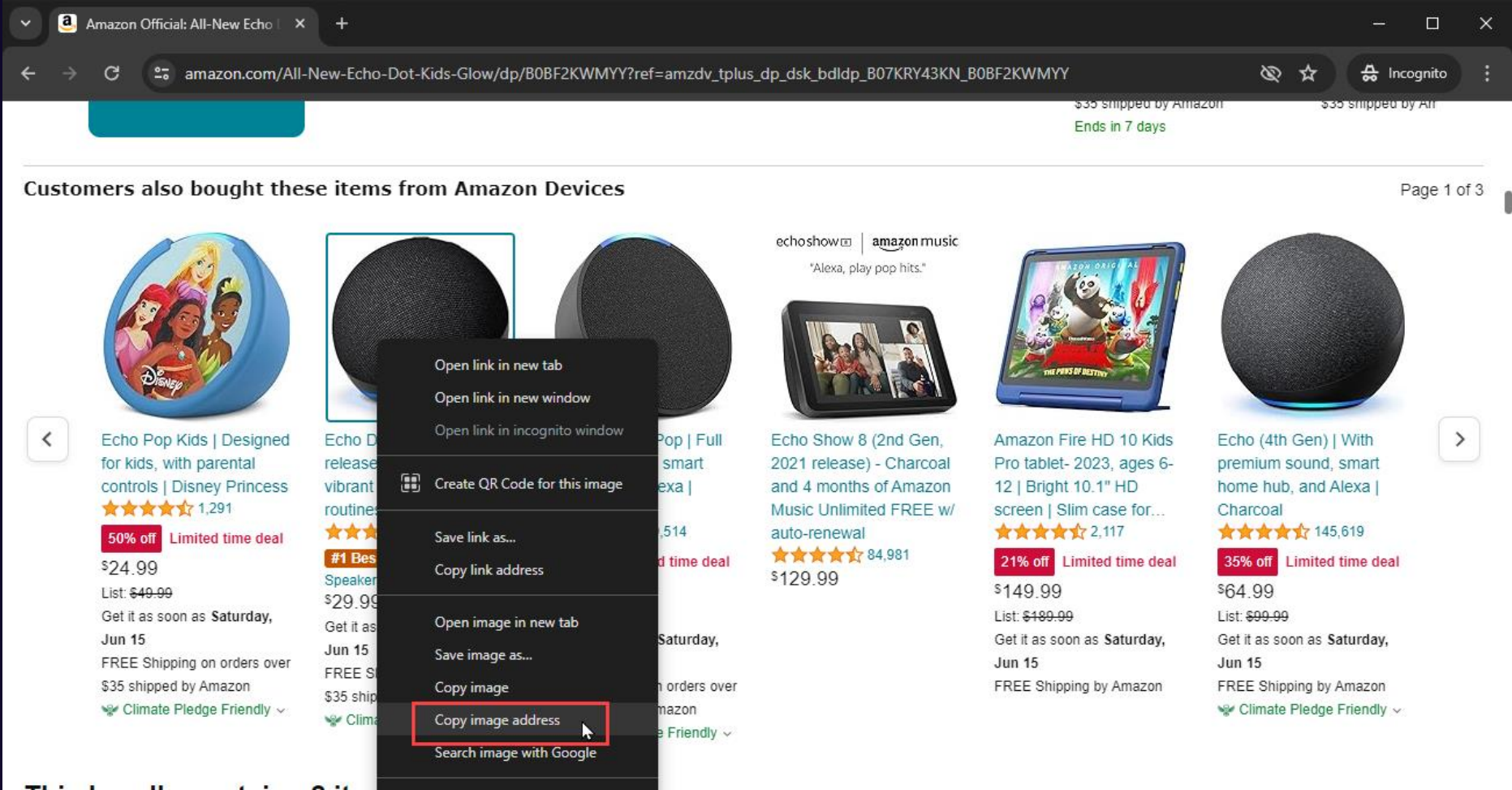


IMDSv1 credential access: Premise

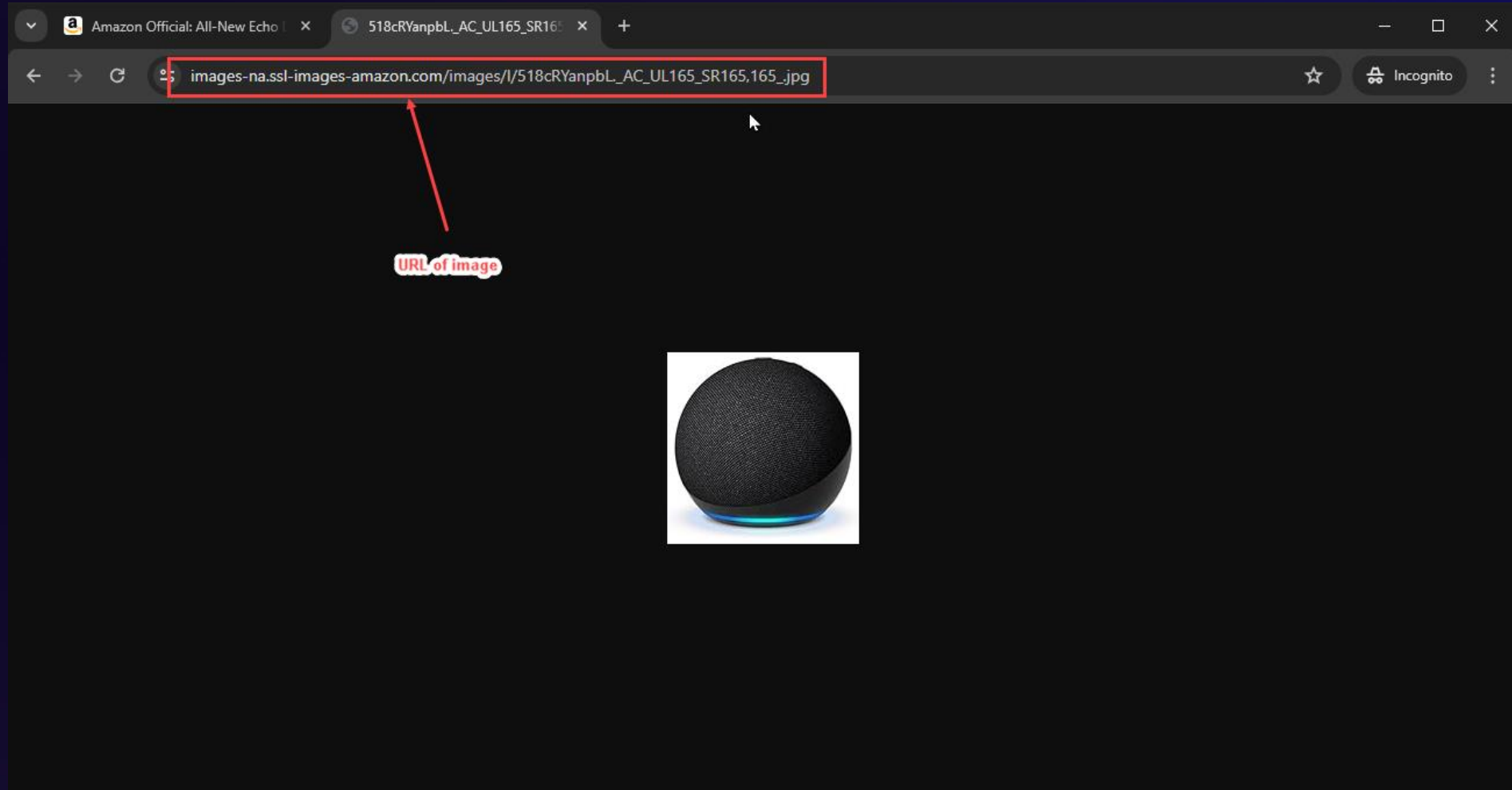
IMDSv1 credential access: Premise



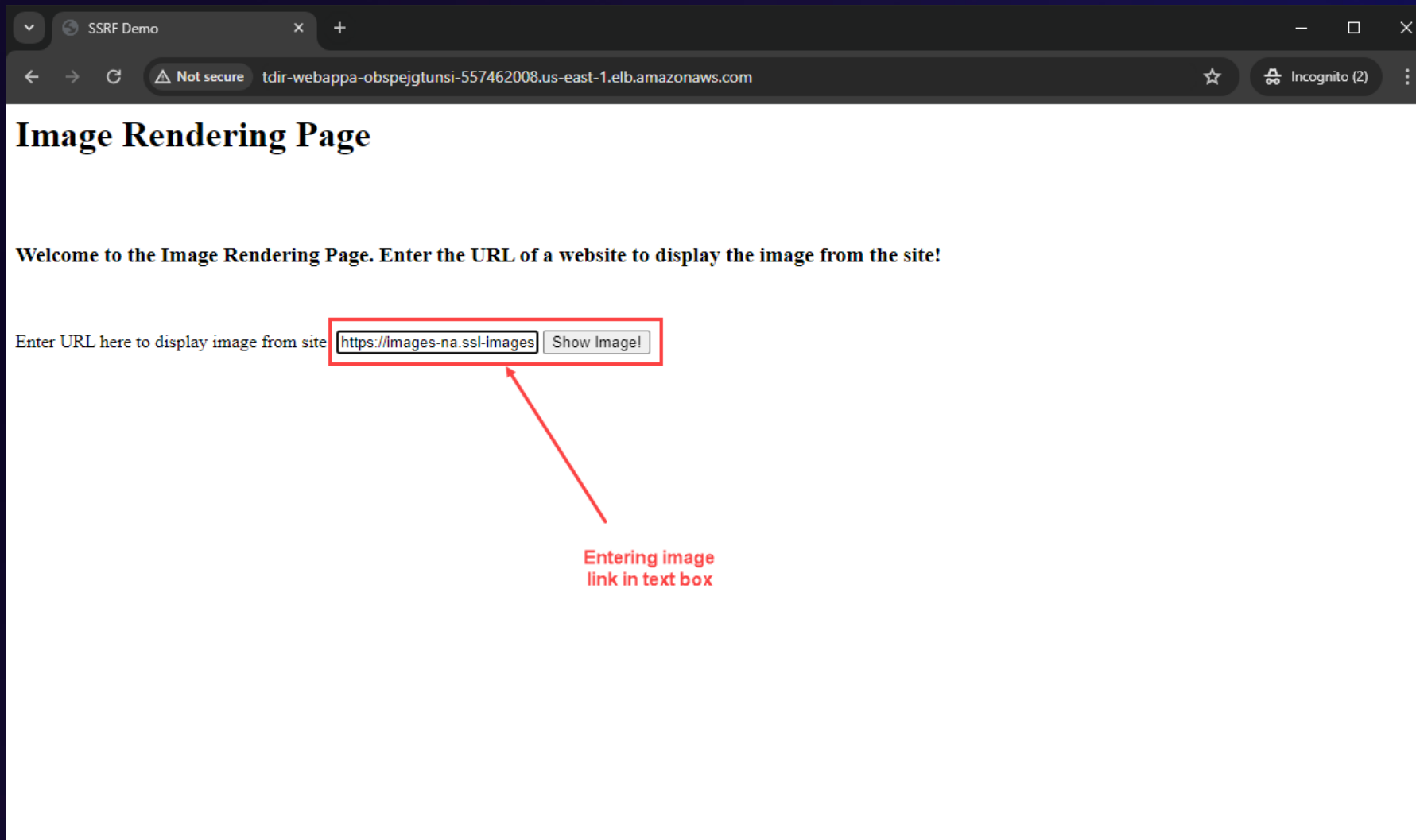
IMDSv1 credential access: Premise



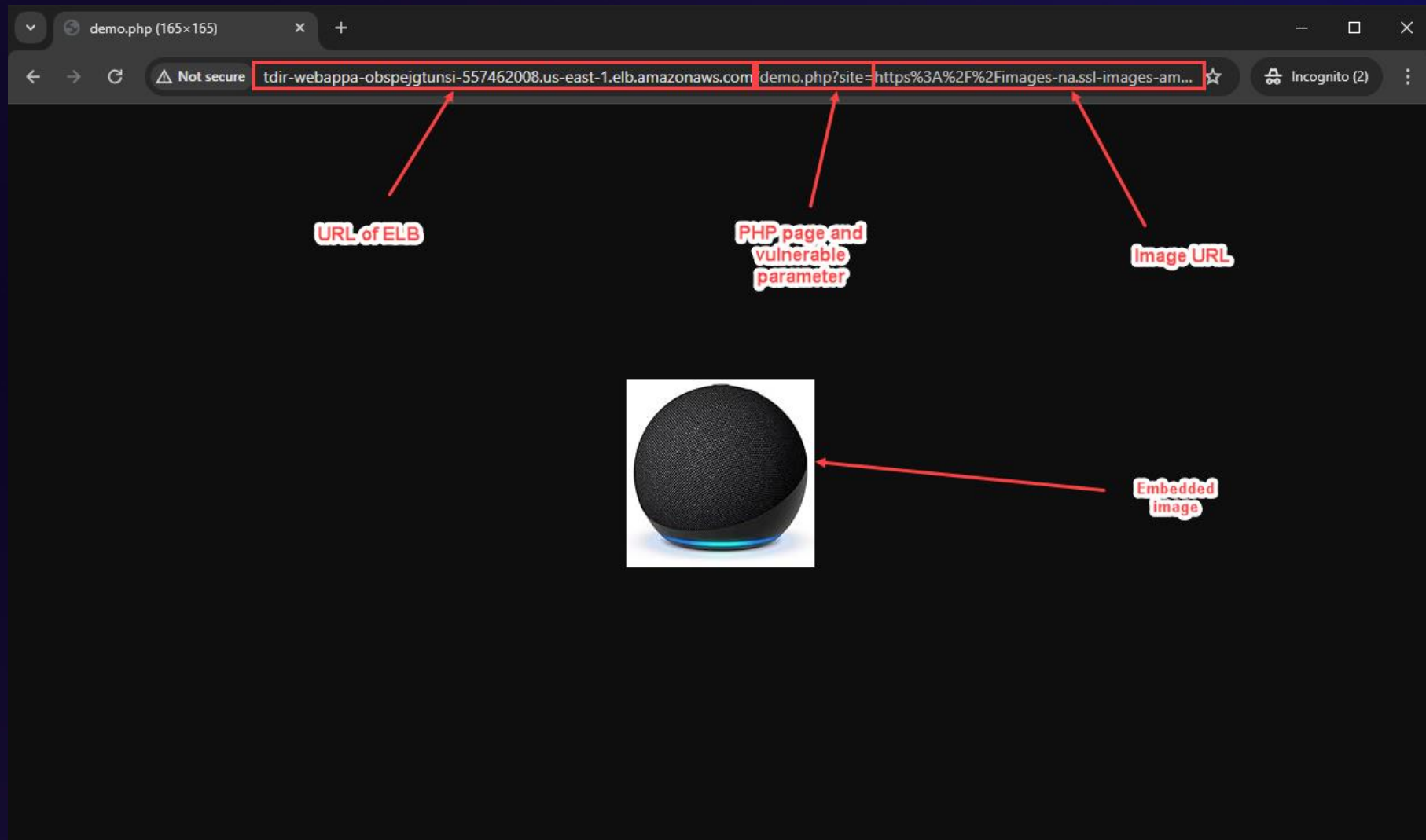
IMDSv1 credential access: Premise



IMDSv1 credential access: Premise



IMDSv1 credential access: Premise



IMDSv1 credential access: Premise

```
sdevera@SEA-1801401059: ~  
:~$  
:~$ curl "http://tdir-webappa-53m6tiqgfres-1313834843.us-east-1.elb.amazonaws.com/demo.php?site=http://169.254.169.254/latest/meta-data"  
ami-id  
ami-launch-index  
ami-manifest-path  
block-device-mapping/  
events/  
hostname  
iam/  
identity-credentials/  
instance-action  
instance-id  
instance-life-cycle  
instance-type  
local-hostname  
local-ipv4  
mac  
metrics/  
network/  
placement/  
profile  
public-hostname  
public-ipv4  
reservation-id  
security-groups  
services/  
system  
:~$
```

URL of ELB

PHP page and vulnerable parameter

URL of IMDS

IMDSv1 credential access: Premise

```
~$ curl "http://tdir-webappa-53m6tiqgfres-1313834843.us-east-1.elb.amazonaws.com/demo.php?site=http://169.254.169.254/latest/meta-data/iam/security-credentials/webdev"
{
  "Code" : "Success",
  "LastUpdated" : "2024-05-25T22:46:55Z",
  "Type" : "AWS-HMAC",
  "AccessKeyId" : "ASIA...",
  "SecretAccessKey" : "...",
  "Token" : "...",
  "Expiration" : "2024-05-26T05:10:18Z"
}
```

Requesting credentials from IMDS

Credentials

IMDSv1 credential access: Mitigations

MITRE ATT&CK

Tactic: Credential access

Technique: Unsecured credentials

IMDSv1 credential access: Mitigations

- *Use require IMDSv2*

MITRE ATT&CK

Tactic: Credential access

Technique: Unsecured credentials

Modify instance metadata options

IMDSv2 uses session-oriented requests. With session-oriented requests, you create a session token that defines the session duration, which can be a minimum of 1 second and a maximum of 6 hours. [Learn more](#)

Instance ID
i-0b7698a84dbd7a660 (test)

Instance metadata service
☒ Enable

IMDSv2
☐ Optional
☒ Required

Set to 'Required'

Ensure that your instance is making no IMDSv1 calls before setting IMDSv2 to required. IMDSv1 calls are recorded by the MetadataNoToken metric in CloudWatch. [View MetadataNoToken for your instance](#)

Cancel Save

IMDSv1 credential access: Mitigations

- ~~Use~~ require IMDSv2
- Use principle of least privilege on EC2 instance profile

MITRE ATT&CK

Tactic: Credential access

Technique: Unsecured credentials

Modify instance metadata options

IMDSv2 uses session-oriented requests. With session-oriented requests, you create a session token that defines the session duration, which can be a minimum of 1 second and a maximum of 6 hours. [Learn more](#)

Instance ID
i-0b7698a84dbd7a660 (test)

Instance metadata service
☒ Enable

IMDSv2
☐ Optional
☒ Required

Set to 'Required'

⚠ Ensure that your instance is making no IMDSv1 calls before setting IMDSv2 to required. IMDSv1 calls are recorded by the MetadataNoToken metric in CloudWatch. [View MetadataNoToken for your instance](#)

Cancel Save

IMDSv1 credential access: Mitigations

- ~~Use~~ *require* IMDSv2
- Use principle of least privilege on EC2 instance profile
- Use the `aws:EC2InstanceSourceVPC` or `aws:EC2InstanceSourcePrivateIPv4` global condition keys in Service Control Policies

MITRE ATT&CK

Tactic: Credential access

Technique: Unsecured credentials

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "*",
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
          "aws:ec2InstanceSourceVPC": "${aws:SourceVpc}"
        }
      },
      "Null": {
        "ec2:SourceInstanceARN": "false"
      },
      "BoolIfExists": {
        "aws:ViaAWSService": "false"
      },
      "ArnNotLike": {
        "aws:PrincipalArn": [
          "arn:aws:iam::*:role/aws:ec2-infrastructure"
        ]
      }
    }
  ]
}
```

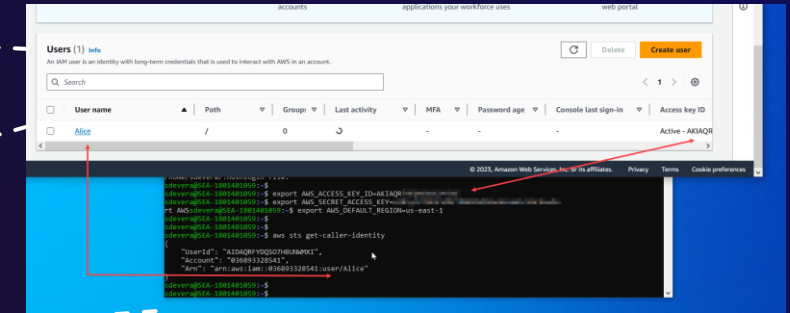
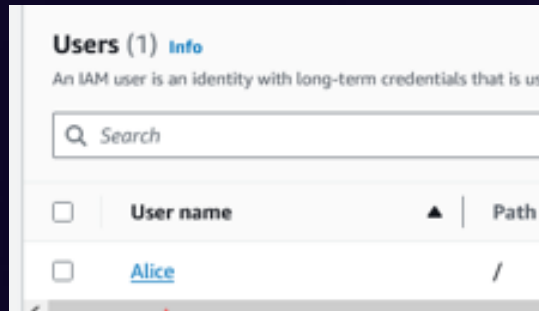
GetFederationToken: Premise

MITRE ATT&CK

Tactic: Persistence

Technique: Additional cloud credentials

1) Credentials exported



```
sdevera@SEA-1801401059:~$  
sdevera@SEA-1801401059:~$ export AWS_ACCESS_KEY_ID=AKIAQR...  
sdevera@SEA-1801401059:~$ export AWS_SECRET_ACCESS_KEY=...  
sdevera@SEA-1801401059:~$ export AWS_DEFAULT_REGION=us-east-1  
sdevera@SEA-1801401059:~$  
sdevera@SEA-1801401059:~$  
sdevera@SEA-1801401059:~$ aws sts get-caller-identity  
{  
  "UserId": "AIDAQRFYDQSO7HBUNWMMXI",  
  "Account": "036893328541",  
  "Arn": "arn:aws:iam::036893328541:user/Alice"
```

GetFederationToken: Premise

2) Federation token generated

```
sdevera@SEA-1801401059:~$ aws sts get-federation-token --name HIDDEN_DUE_TO_SECURITY_REASONS --duration-seconds 129600 -policy-arns arn=arn:aws:iam::aws:policy/AdministratorAccess
```

```
{  
    "Credentials": {  
        "AccessKeyId": "[REDACTED]",  
        "SecretAccessKey": "[REDACTED]",  
        "SessionToken": "[REDACTED]"  
    },  
    "Expiration": "2023-10-06T09:45:07+00:00"  
},  
"FederatedUser": {  
    "FederatedUserId": "036893328541:HIDDEN_DUE_TO_SECURITY_REASONS",  
    "Arn": "arn:aws:sts::036893328541:federated-user/HIDDEN_DUE_TO_SECURITY_REASONS"  
},  
"PackedPolicySize": 7  
}
```

GetFederationToken: Premise

3) Threat actor exports and assumes federation token credentials

```
sdevera@SEA-1801401059:~$ export AWS_ACCESS_KEY_ID=ASIA...
AWS_sdevera@SEA-1801401059:~$ export AWS_SECRET_ACCESS_KEY=...
t AWS_sdevera@SEA-1801401059:~$ export AWS_DEFAULT_REGION=us-east-1
sdevera@SEA-1801401059:~$ export AWS_SESSION_TOKEN=...

sdevera@SEA-1801401059:~$ aws sts get-caller-identity
{
  "UserId": "036893328541:HIDDEN_DUE_TO_SECURITY_REASONS",
  "Account": "036893328541",
  "Arn": "arn:aws:sts::036893328541:federated-user/HIDDEN_DUE_TO_SECURITY_REASONS"
}
sdevera@SEA-1801401059:~$
```

Exporting
credentials

GetFederationToken: Premise

4) Use exported credentials from federation token

The screenshot illustrates the process of using federation token credentials in the AWS Management Console. It is divided into two main sections: a terminal window at the top and the AWS console interface at the bottom.

Terminal Window (Top):

- The user exports AWS credentials from a federation token:

```
sdevera@SEA-1801401059:~$ export AWS_ACCESS_KEY_ID=ASIA[redacted]
sdevera@SEA-1801401059:~$ export AWS_SECRET_ACCESS_KEY=[redacted]
sdevera@SEA-1801401059:~$ export AWS_DEFAULT_REGION=us-east-1
sdevera@SEA-1801401059:~$ export AWS_SESSION_TOKEN=[redacted]
```

A red box highlights these export commands, with a red arrow pointing to the text "Exporting credentials".
- The user runs `aws sts get-caller-identity` to verify the identity:

```
sdevera@SEA-1801401059:~$ aws sts get-caller-identity
{
  "UserId": "[redacted]:HIDDEN_DUE_TO_SECURITY_REASONS",
  "Account": "[redacted]",
  "Arn": "arn:aws:sts::[redacted]:federated-user/[redacted]:HIDDEN_DUE_TO_SECURITY_REASONS"
}
```

A red box highlights the `"Arn"` field, which contains the user name `HIDDEN_DUE_TO_SECURITY_REASONS`.
- The user runs `aws s3 ls` to list buckets:

```
sdevera@SEA-1801401059:~$ aws s3 ls
2023-09-21 14:16:19 cloudtrail-awslogs-[redacted]-amok46uu-isengard-do-not-delete
2023-09-22 08:58:39 do-not-delete-gatedgarden-audit-[redacted]
```

A red box highlights the `aws s3 ls` command, with a red arrow pointing to the text "ListBuckets".

AWS Management Console (Bottom):

- The console shows the "CloudTrail" service, with the "Event history" tab selected.
- The "ListBuckets" event is highlighted in the event history table.
- The "Details" view for the "ListBuckets" event is shown, with the following information:

Details	
Event time	October 04, 2023, 18:07:48 (UTC-04:00)
User name	HIDDEN_DUE_TO_SECURITY_REASONS
Event name	ListBuckets
Event source	
AWS access key	ASIA[redacted]
Source IP address	12.116.165.10
Event ID	2db69531-b828-413a-8957-6f3835f5ab81
Request ID	

A red box highlights the "User name" field, which contains the value `HIDDEN_DUE_TO_SECURITY_REASONS`. A red arrow points from this box to the text "User name = HIDDEN_DUE_TO_SECURITY_REASONS".

GetFederationToken: Premise

- The session name or 'user name' can be changed
- Still need to review actions by 'masked' user

```
{
  "eventVersion": "1.09",
  "userIdentity": {
    "type": "FederatedUser",
    "principalId": "036[REDACTED]:HIDDEN_DUE_TO_SECURITY_REASONS",
    "arn": "arn:aws:sts:036[REDACTED]:federated-user/HIDDEN_DUE_TO_SECURITY_REASONS",
    "accountId": "036[REDACTED]",
    "accessKeyId": "ASIA[REDACTED]",
    "sessionContext": {
      "sessionIssuer": {
        "type": "IAMUser",
        "principalId": "AIDAQRFYDQSO7HBUNWMXI",
        "arn": "arn:aws:iam:036[REDACTED]:user/Alice",
        "accountId": "036[REDACTED]",
        "userName": "Alice"
      },
      "attributes": {
        "creationDate": "2023-10-04T21:45:07Z",
        "mfaAuthenticated": "false"
      }
    }
  },
  "eventTime": "2023-10-04T22:07:48Z",
  "eventSource": "s3.amazonaws.com",
  "eventName": "ListBuckets",
  "awsRegion": "us-east-1",
}
```

Named session

Originating principal

GetFederationToken: Mitigations

- `GetSessionToken` also used
- Generally considered unauthorized if observed
- With both `GetFederationToken` and `GetSessionToken`, you can delete the originating access key and the session will persist
- Can delete/recreate the user

GetFederationToken: Mitigations

- Apply inline policy to IAM user (deny based on `aws:TokenIssueTime`)

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Deny",
    "Action": "*",
    "Resource": "*",
    "Condition": {
      "DateLessThan": {"aws:TokenIssueTime": "2014-05-07T23:47:00Z"}
    }
  }
}
```

Time in ISO 8601 format
based on time of
GetFederationToken
event



Novel threat actor tactics



Create account: Premise

1) Threat actor creates an account in an AWS organization

MITRE ATT&CK

Tactic: Defense evasion

Technique: Unused/unsupported cloud regions

AWS Organizations X

AWS Organizations > AWS accounts > Add an AWS account

Add an AWS account

You can add an AWS account to your organization either by creating an account or by inviting one or more existing AWS accounts to join your organization.

☒ **Create an AWS account**
Create an AWS account that is added to your organization.

☐ **Invite an existing AWS account**
Send an email request to the owner of the account. If they accept, the account joins the organization.

Create an AWS account

AWS account name
Sandbox

Email address of the account's owner
account@domain.com

IAM role name
The management account can use this IAM role to access resources in the member account.
OrganizationAccountAccessRole

Tags
Tags are key-value pairs that you can add to AWS resources to help identify, organize, and secure your AWS resources.
No tags are associated with the resource.
Add tag
You can add 50 more tags.

Cancel **Create AWS account**

Create account: Premise

- 1) Threat actor creates an account in an AWS organization
- 2) Created account is used for defense evasion, resource hijacking

MITRE ATT&CK

Tactic: Defense evasion

Technique: Unused/unsupported cloud regions

AWS Organizations X

AWS Organizations > AWS accounts > Add an AWS account

Add an AWS account

You can add an AWS account to your organization either by creating an account or by inviting one or more existing AWS accounts to join your organization.

☒ **Create an AWS account**
Create an AWS account that is added to your organization.

☐ **Invite an existing AWS account**
Send an email request to the owner of the account. If they accept, the account joins the organization.

Create an AWS account

AWS account name
Sandbox

Email address of the account's owner
account@domain.com

IAM role name
The management account can use this IAM role to access resources in the member account.
OrganizationAccountAccessRole

Tags
Tags are key-value pairs that you can add to AWS resources to help identify, organize, and secure your AWS resources.
No tags are associated with the resource.
Add tag
You can add 50 more tags.

Cancel **Create AWS account**

Create account: Alternative

MITRE ATT&CK

Tactic: Defense evasion

Technique: Unused/unsupported cloud regions

- 1) Threat actor creates a standalone account with a stolen credit card

Create account: Alternative

- 1) Threat actor creates a standalone account with a stolen credit card
- 2) Invites account to compromised AWS organization

MITRE ATT&CK

Tactic: Defense evasion

Technique: Unused/unsupported cloud regions

AWS Organizations ×

[AWS Organizations](#) > [AWS accounts](#) > Add an AWS account

Add an AWS account

You can add an AWS account to your organization either by creating an account or by inviting one or more existing AWS accounts to join your organization.

☐ Create an AWS account
Create an AWS account that is added to your organization.

☒ Invite an existing AWS account
Send an email request to the owner of the account. If they accept, the account joins the organization.

Invite one or more existing AWS accounts to join your organization

Email address or account ID of the AWS accounts to invite

Message to include in the invitation email message - optional
This text is included in the email message sent to the owners of the invited AWS accounts.

Tags

Tags are key-value pairs that you can add to AWS resources to help identify, organize, and secure your AWS resources.

No tags are associated with the resource.

You can add 50 more tags.

Create account: Premise

MITRE ATT&CK

Tactic: Defense evasion

Technique: Unused/unsupported cloud regions

- 1) Threat actor can remove *OrganizationAccountAccessRole*

Create account: Premise

MITRE ATT&CK

Tactic: Defense evasion

Technique: Unused/unsupported cloud regions

- 1) Threat actor can remove *OrganizationAccountAccessRole*
- 2) Victim can apply SCPs, but this prevents new actions (existing threat actor resources not affected)

Create account: Premise

MITRE ATT&CK

Tactic: Defense evasion

Technique: Unused/unsupported cloud regions

- 1) Threat actor can remove *OrganizationAccountAccessRole*
- 2) Victim can apply SCPs, but this prevents new actions (existing threat actor resources not affected)
- 3) May need support case to remove account

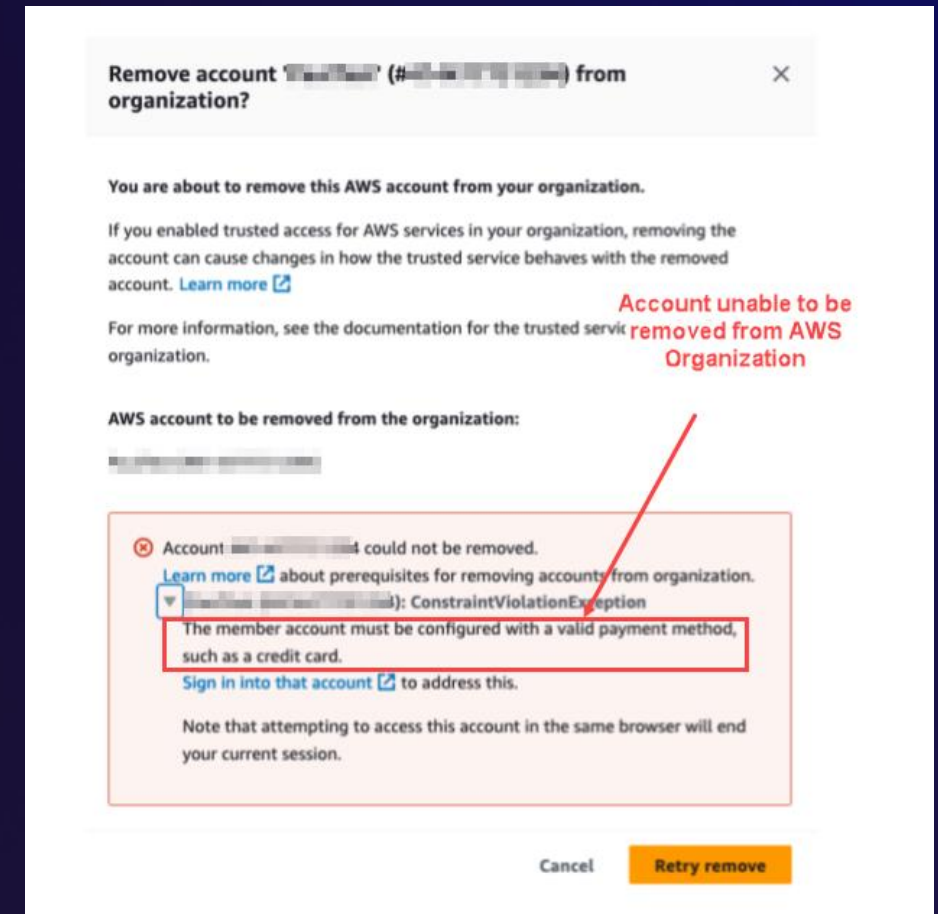
Create account: Premise

- 1) Threat actor can remove *OrganizationAccountAccessRole*
- 2) Victim can apply SCPs, but this prevents new actions (existing threat actor resources not affected)
- 3) May need support case to remove account

MITRE ATT&CK

Tactic: Defense evasion

Technique: Unused/unsupported cloud regions



Create account: Mitigations

MITRE ATT&CK

Tactic: Defense evasion

Technique: Unused/unsupported cloud regions

- Create custom groups or roles
- Use principle of least privilege to restrict account creation
- Amazon CloudWatch alarm/SCP for `InviteAccountToOrganization` API call

Lifecycle deletion: Premise

MITRE ATT&CK

Tactic: Impact

Technique: Data destruction

- 1) Threat actor uses S3 lifecycle policies to set parameters to delete objects within 1 day

Lifecycle deletion: Premise

MITRE ATT&CK

Tactic: Impact

Technique: Data destruction

- 1) Threat actor uses S3 lifecycle policies to set parameters to delete objects within 1 day

```

{
  "eventTime": "2023-10-18T10:29:40Z",
  "eventSource": "s3.amazonaws.com",
  "eventName": "PutBucketLifecycle",
  "awsRegion": "eu-west-1",
  "sourceIPAddress": "10.0.0.1",
  "userAgent": "[aws-cli/2.13.2 Python/3.11.4 Darwin/22.6.0 source/x86_64 prompt/off command/s3api.put-bucket-lifecycle-configuration]",
  "requestParameters": {
    "lifecycle": "",
    "bucketName": "s3-eu-west-1-123456789012",
    "LifecycleConfiguration": {
      "xmlns": "http://s3.amazonaws.com/doc/2006-03-01/",
      "Rule": {
        "Status": "Enabled",
        "Filter": "",
        "NoncurrentVersionExpiration": {
          "NoncurrentDays": 1
        },
        "Expiration": {
          "Days": 1
        },
        "ID": "Delete_all_Oct",
        "AbortIncompleteMultipartUpload": {
          "DaysAfterInitiation": 1
        }
      }
    }
  },
  "Host": "s3-eu-west-1.amazonaws.com"
},
{
  "responseElements": null,
  "additionalEventData": {
    "SignatureVersion": "SigV4",
    "CipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
    "bytesTransferredIn": 408,
    "AuthenticationMethod": "AuthHeader"
  }
}
```

Policy sets parameters to minimum values

Lifecycle deletion: Premise

MITRE ATT&CK

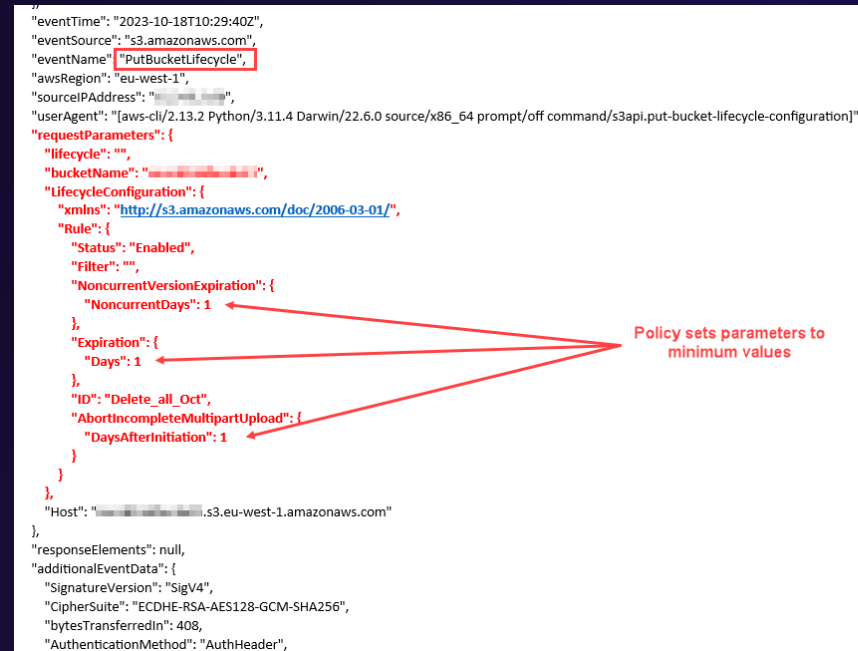
Tactic: Impact

Technique: Data destruction

1) Threat actor uses S3 lifecycle policies to set parameters to delete objects within 1 day

2) Form of data destruction

```
"eventTime": "2023-10-18T10:29:40Z",
"eventSource": "s3.amazonaws.com",
"eventName": "PutBucketLifecycle",
"awsRegion": "eu-west-1",
"sourceIPAddress": "10.0.0.1",
"userAgent": "[aws-cli/2.13.2 Python/3.11.4 Darwin/22.6.0 source/x86_64 prompt/off command/s3api.put-bucket-lifecycle-configuration]",
"requestParameters": {
  "lifecycle": "",
  "bucketName": "s3-eu-west-1-123456789012",
  "LifecycleConfiguration": {
    "xmlns": "http://s3.amazonaws.com/doc/2006-03-01/",
    "Rule": {
      "Status": "Enabled",
      "Filter": "",
      "NoncurrentVersionExpiration": {
        "NoncurrentDays": 1
      },
      "Expiration": {
        "Days": 1
      },
      "ID": "Delete_all_Oct",
      "AbortIncompleteMultipartUpload": {
        "DaysAfterInitiation": 1
      }
    }
  }
},
"Host": "s3-eu-west-1.amazonaws.com"
},
"responseElements": null,
"additionalEventData": {
  "SignatureVersion": "SigV4",
  "CipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
  "bytesTransferredIn": 408,
  "AuthenticationMethod": "AuthHeader",
```



Policy sets parameters to minimum values

Lifecycle deletion: Premise

MITRE ATT&CK

Tactic: Impact

Technique: Data destruction

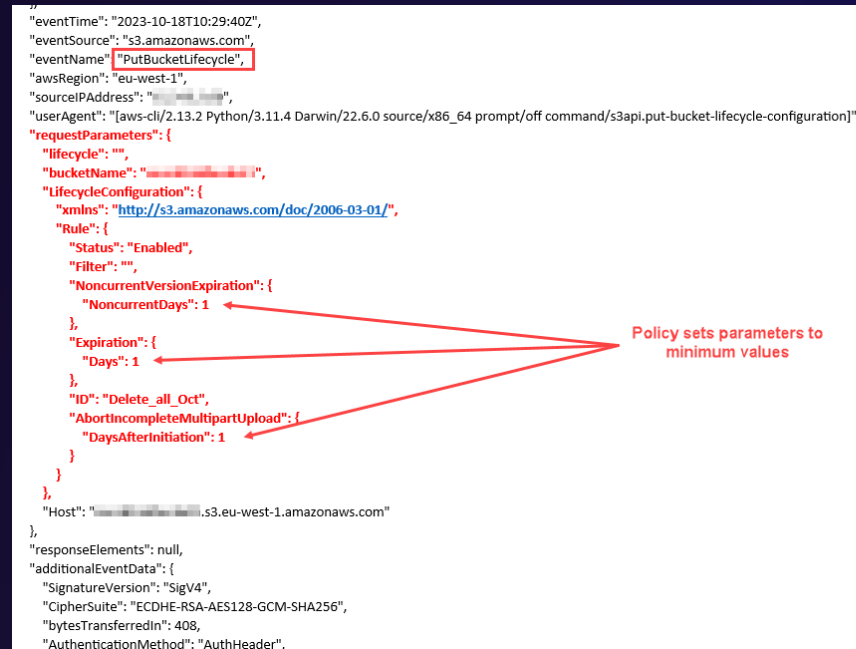
1) Threat actor uses S3 lifecycle policies to set parameters to delete objects within 1 day

2) Form of data destruction

3) Bypasses permissions and detections against DeleteObject

```

{
  "eventTime": "2023-10-18T10:29:40Z",
  "eventSource": "s3.amazonaws.com",
  "eventName": "PutBucketLifecycle",
  "awsRegion": "eu-west-1",
  "sourceIPAddress": "10.0.0.1",
  "userAgent": "[aws-cli/2.13.2 Python/3.11.4 Darwin/22.6.0 source/x86_64 prompt/off command/s3api.put-bucket-lifecycle-configuration]",
  "requestParameters": {
    "lifecycle": "",
    "bucketName": "s3-eu-west-1-123456789012",
    "LifecycleConfiguration": {
      "xmlns": "http://s3.amazonaws.com/doc/2006-03-01/",
      "Rule": {
        "Status": "Enabled",
        "Filter": "",
        "NoncurrentVersionExpiration": {
          "NoncurrentDays": 1
        },
        "Expiration": {
          "Days": 1
        },
        "ID": "Delete_all_Oct",
        "AbortIncompleteMultipartUpload": {
          "DaysAfterInitiation": 1
        }
      }
    }
  },
  "Host": "s3-eu-west-1.amazonaws.com"
},
{
  "responseElements": null,
  "additionalEventData": {
    "SignatureVersion": "SigV4",
    "CipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
    "bytesTransferredIn": 408,
    "AuthenticationMethod": "AuthHeader"
  }
}
```



Policy sets parameters to minimum values

Lifecycle deletion: Mitigations

MITRE ATT&CK

Tactic: Impact

Technique: Data destruction

- Apply SCPs to prevent use of `PutBucketLifecycle`
- Use principle of least privilege
- AWS Config rule for `s3-lifecycle-policy-check`

SMS pumping: Premise

MITRE ATT&CK

Tactic: Impact

Technique: Resource hijacking

- 1) Threat actor obtains block of high rate SMS phone numbers from telecom provider
- 2) Threat actor identifies service that sends SMS text messages
- 3) Service used to send numerous text messages

SMS pumping: Premise

MITRE ATT&CK

Tactic: Impact

Technique: Resource hijacking

4) Amazon Cognito used

SMS pumping: Premise

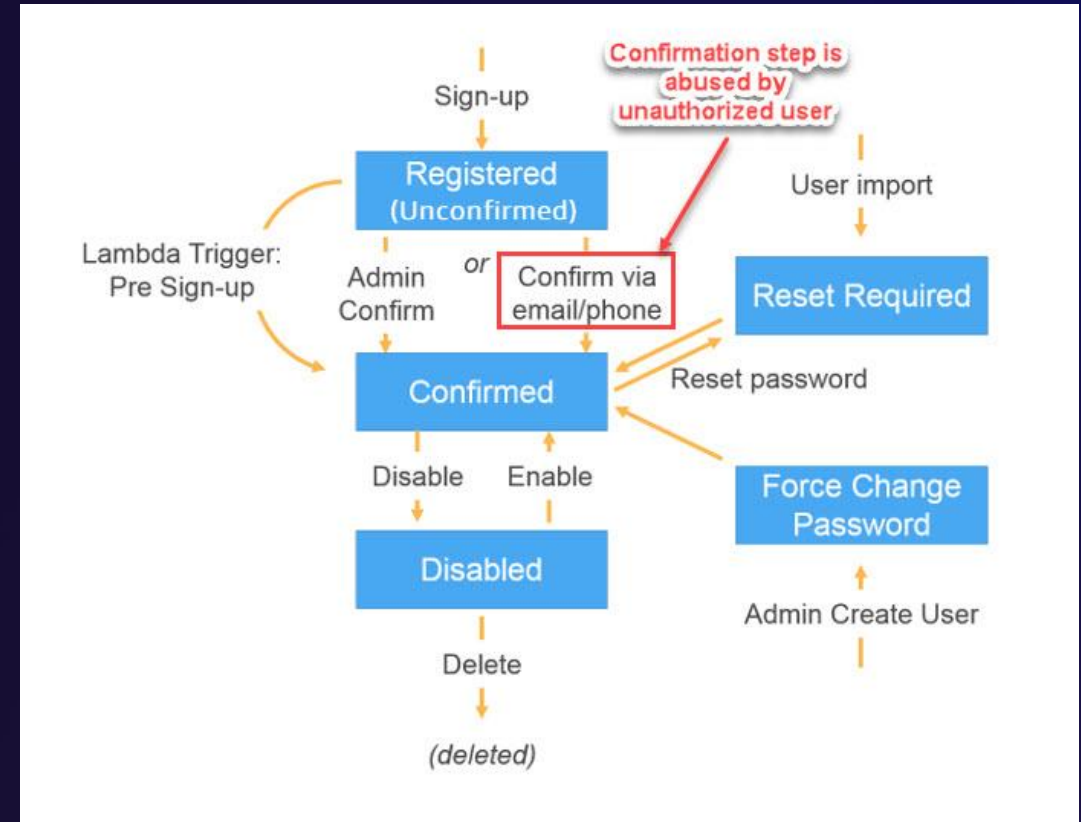
MITRE ATT&CK

Tactic: Impact

Technique: Resource hijacking

4) Amazon Cognito used

5) APIs observed are SignUp or ResendConfirmationCode



SMS pumping: Mitigations

MITRE ATT&CK

Tactic: Impact

Technique: Resource hijacking

- Change attribute verification and user account confirmation
- Apply AWS WAF to present CAPTCHA
- Apply web ACL rule to inspect request body and match the SMS area code
- Amazon Fraud Detector (may require rearchitected solution)

Leave organization: Premise

MITRE ATT&CK

Tactic: Defense evasion

Technique: Indicator removal

- 1) Threat actor attempts to leave an AWS organization

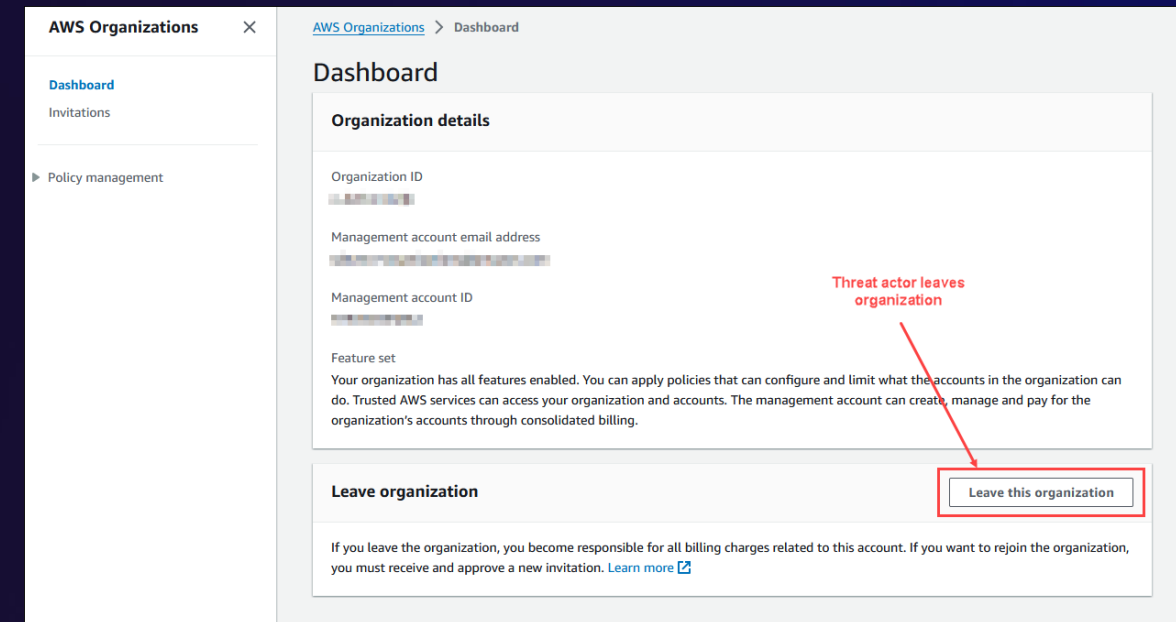
Leave organization: Premise

MITRE ATT&CK

Tactic: Defense evasion

Technique: Indicator removal

1) Threat actor attempts to leave an AWS organization



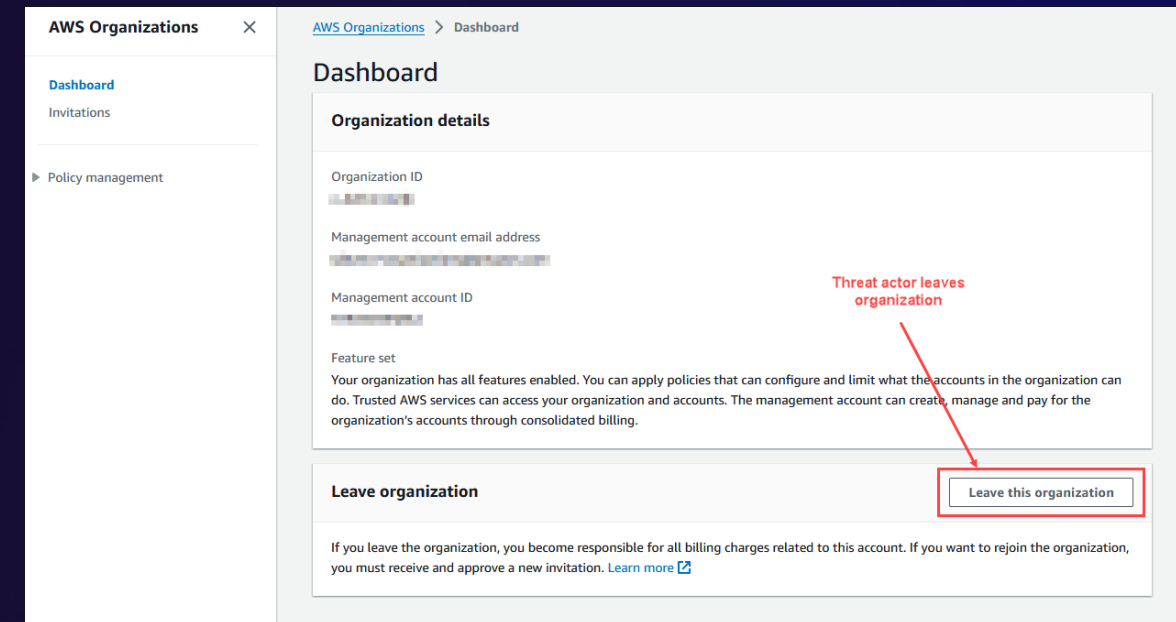
Leave organization: Premise

- 1) Threat actor attempts to leave an AWS organization
- 2) Prevents SCPs from being applied, used for resource hijacking

MITRE ATT&CK

Tactic: Defense evasion

Technique: Indicator removal



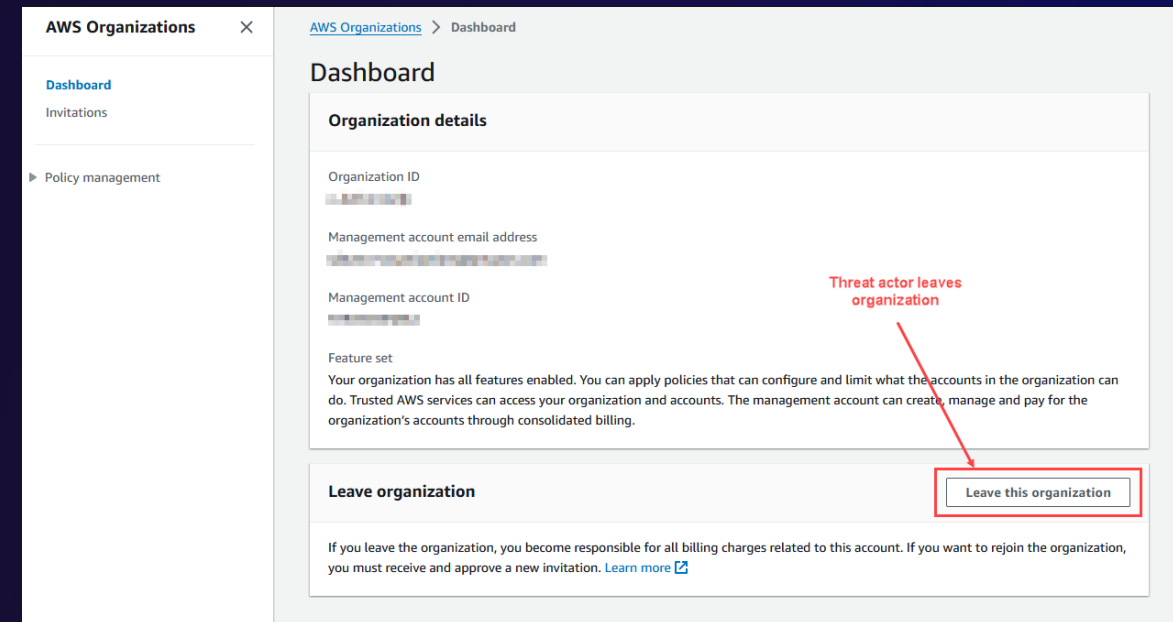
Leave organization: Premise

MITRE ATT&CK

Tactic: Defense evasion

Technique: Indicator removal

- 1) Threat actor attempts to leave an AWS organization
- 2) Prevents SCPs from being applied, used for resource hijacking
- 3) Form of defense evasion, AWS billing reports migrate



Leave organization: Mitigations

MITRE ATT&CK

Tactic: Defense evasion

Technique: Indicator removal

- Apply SCPs to prevent `LeaveOrganization` API call in member account
- Use principle of least privilege to limit use of `RemoveAccountFromOrganization` in management account

Create identity provider: Premise

MITRE ATT&CK

Tactic: Persistence

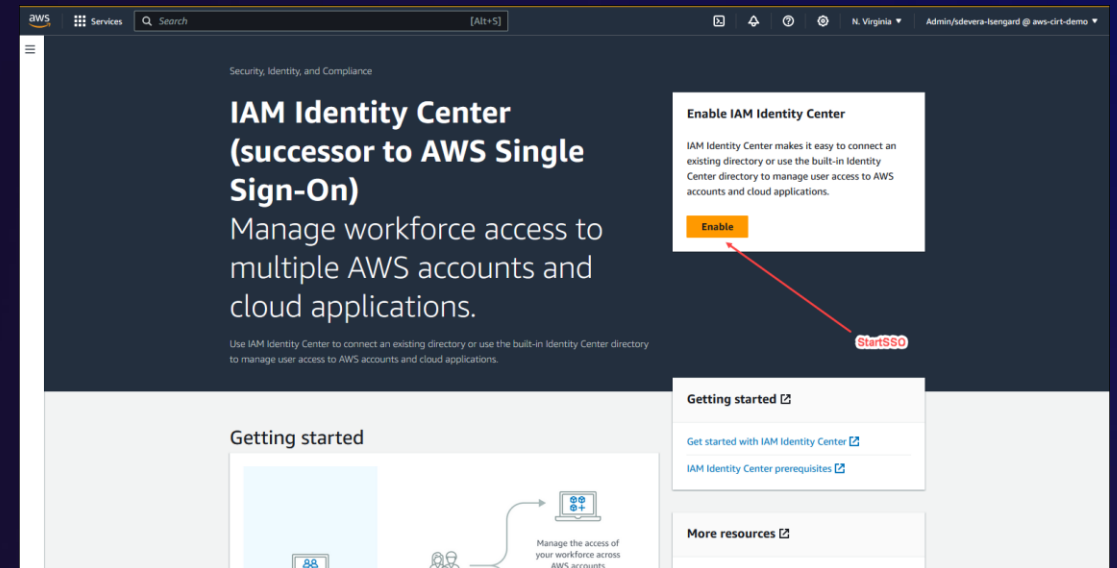
Technique: Create account

- 1) Threat actor gains access to an AWS organization

Create identity provider: Premise

- 1) Threat actor gains access to an AWS organization
- 2) AWS IAM Identity Center enabled to provision access to accounts

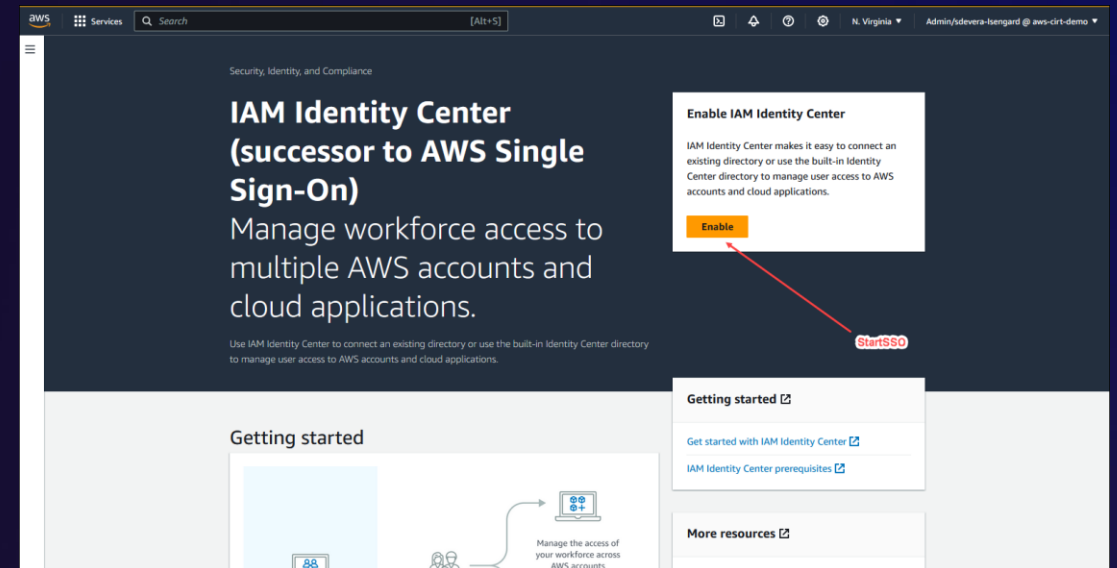
MITRE ATT&CK
Tactic: Persistence
Technique: Create account



Create identity provider: Premise

- 1) Threat actor gains access to an AWS organization
- 2) AWS IAM Identity Center enabled to provision access to accounts
- 3) Adds extra steps to containment

MITRE ATT&CK
Tactic: Persistence
Technique: Create account



Create identity provider: Alternative

MITRE ATT&CK

Tactic: Persistence

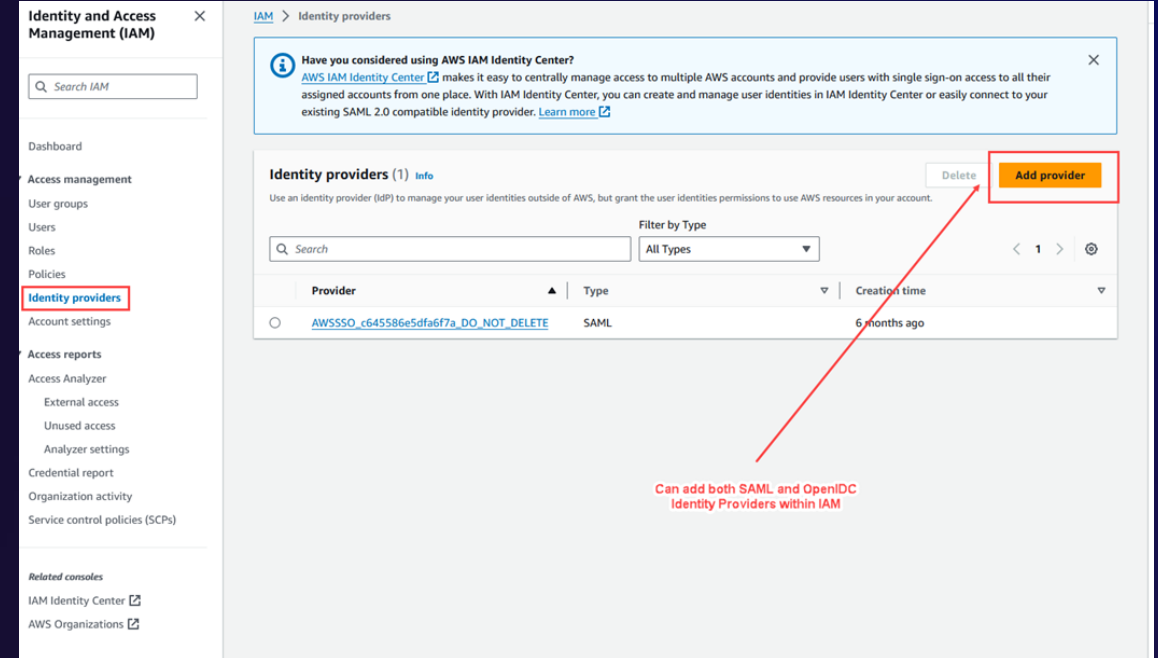
Technique: Create account

- 3) Access to a specific account/s within an AWS organization

Create identity provider: Alternative

- 3) Access to a specific account/s within an AWS organization
- 4) IAM used to add a SAML or OpenIDC provider

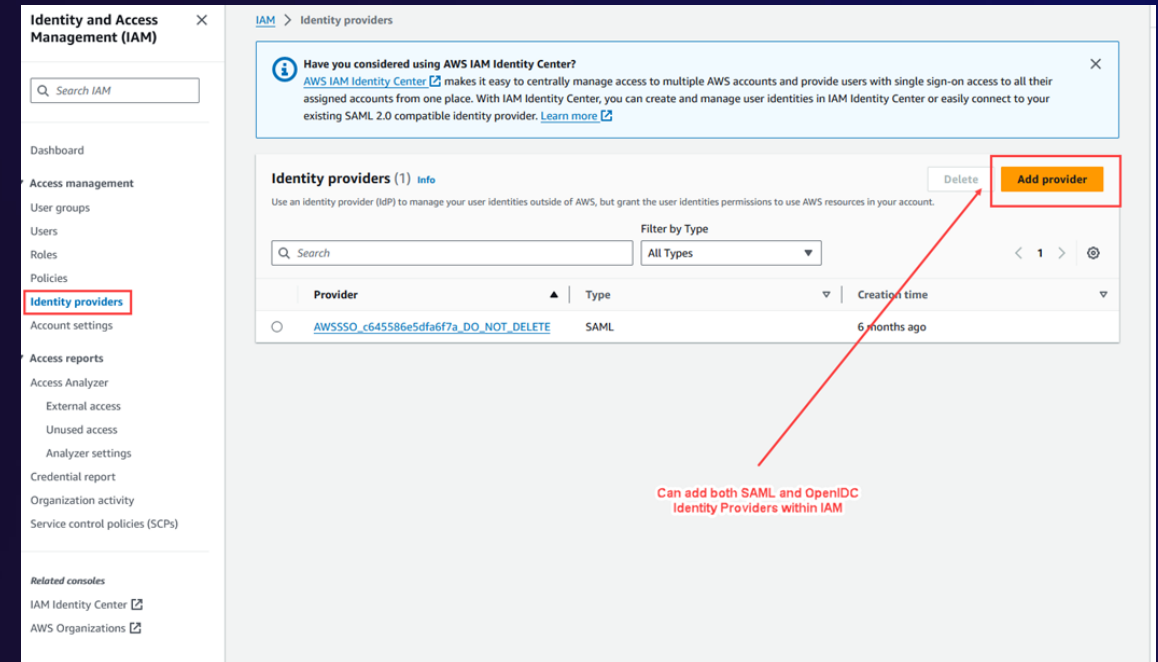
MITRE ATT&CK
Tactic: Persistence
Technique: Create account



Create identity provider: Alternative

- 3) Access to a specific account/s within an AWS organization
- 4) IAM used to add a SAML or OpenIDC provider
- 5) Look for `CreateSAMLProvider` or `CreateOIDCProvider` events in AWS CloudTrail

MITRE ATT&CK
Tactic: Persistence
Technique: Create account



Create identity provider: Mitigations

- Remove identity provider from IAM Identity Center or IAM

MITRE ATT&CK

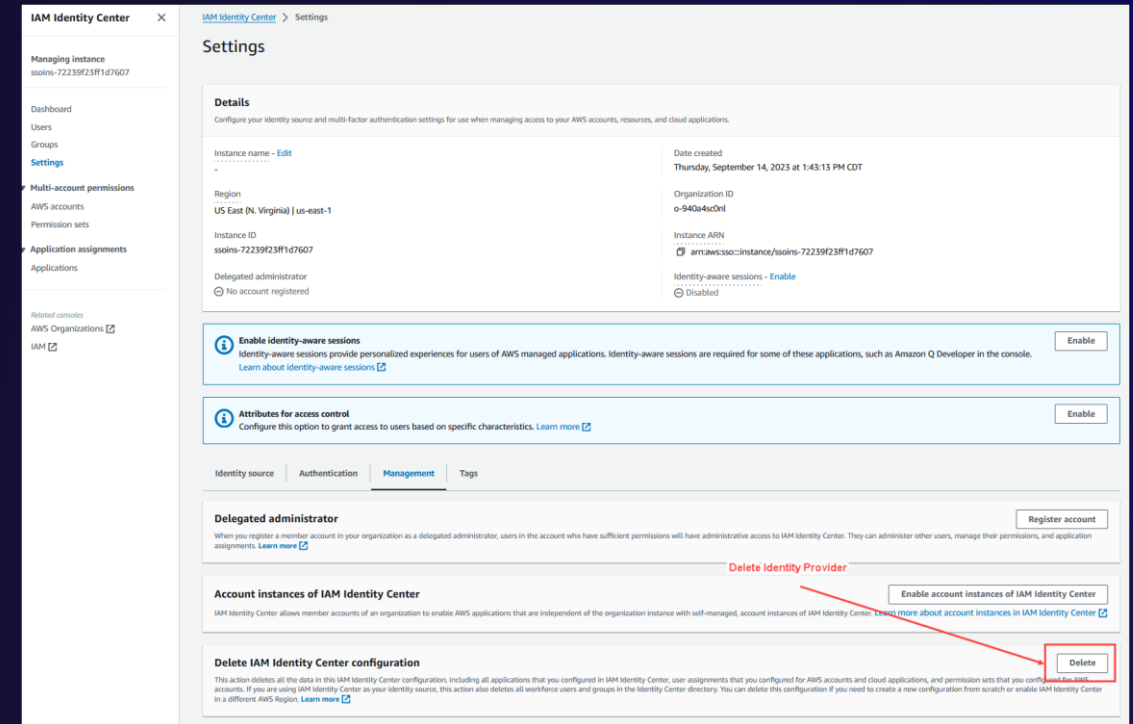
Tactic: Persistence

Technique: Create account

Create identity provider: Mitigations

- Remove identity provider from IAM Identity Center or IAM
- Use Amazon EventBridge to watch for StartSSO, CreateSAMLProvider or CreateOIDCProvider events in CloudTrail

MITRE ATT&CK
Tactic: Persistence
Technique: Create account

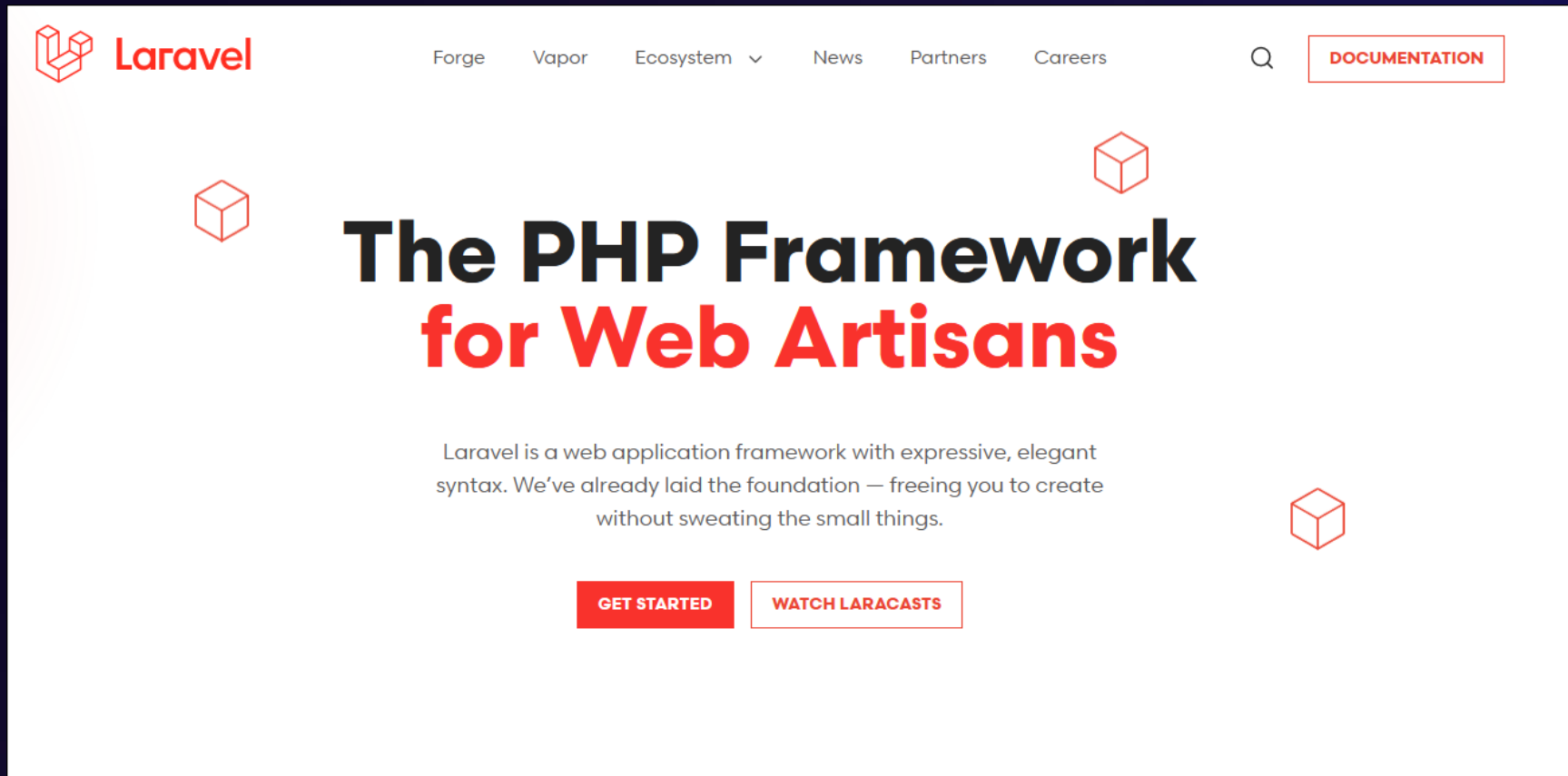


Laravel framework access: Premise

MITRE ATT&CK

Tactic: Initial access

Technique: Exploit public-facing application



Laravel framework access: Premise

MITRE ATT&CK

Tactic: Initial access

Technique: Exploit public-facing application

1) Threat actor identifies vulnerable version of Laravel

- CVE-2021-3129
- Debug mode

Laravel framework access: Premise

1) Threat actor identifies vulnerable version of Laravel

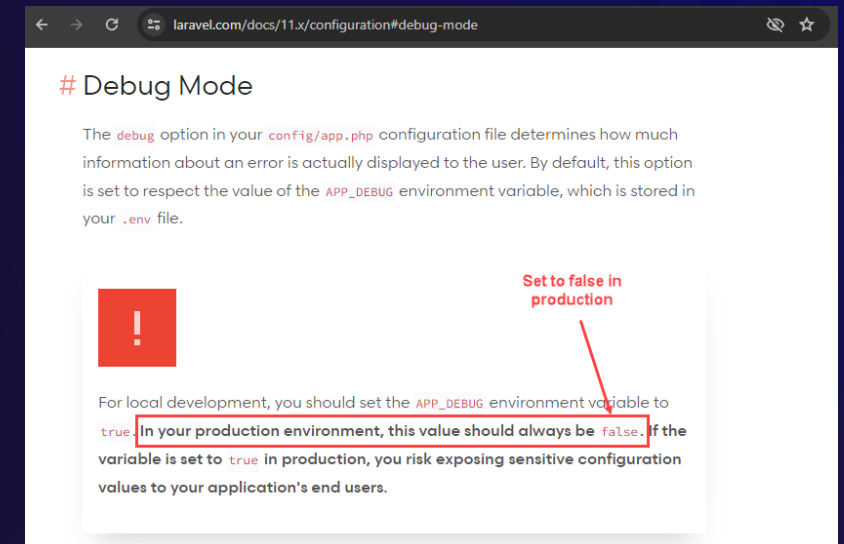
- CVE-2021-3129
- Debug mode

2) Debug mode allows access to .env file

MITRE ATT&CK

Tactic: Initial access

Technique: Exploit public-facing application



Laravel framework access: Premise

- 1) Threat actor identifies vulnerable version of Laravel
 - CVE-2021-3129
 - Debug mode
- 2) Debug mode allows access to .env file
- 3) .env configured with AWS credentials

MITRE ATT&CK

Tactic: Initial access

Technique: Exploit public-facing application

```
APP_NAME=My App
APP_ENV=local
APP_KEY=base64:
APP_DEBUG=true
APP_URL=http://localhost

LOG_CHANNEL=stack

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=my_database
DB_USERNAME=root
DB_PASSWORD=password

BROADCAST_DRIVER=log
CACHE_DRIVER=file
QUEUE_CONNECTION=sync
SESSION_DRIVER=file
SESSION_LIFETIME=120

REDIS_HOST=127.0.0.1
REDIS_PASSWORD=null
REDIS_PORT=6379

MAIL_MAILER=smtp
MAIL_HOST=smtp.mailtrap.io
MAIL_PORT=2525
MAIL_USERNAME=null
MAIL_PASSWORD=null
MAIL_ENCRYPTION=null
MAIL_FROM_ADDRESS=null
MAIL_FROM_NAME="${APP_NAME}"

AWS_ACCESS_KEY_ID=
AWS_SECRET_ACCESS_KEY=
AWS_DEFAULT_REGION=us-east-1
AWS_BUCKET=

PUSHER_APP_ID=
PUSHER_APP_KEY=
PUSHER_APP_SECRET=
PUSHER_APP_CLUSTER=mt1

MIX_PUSHER_APP_KEY="${PUSHER_APP_KEY}"
MIX_PUSHER_APP_CLUSTER="${PUSHER_APP_CLUSTER}"
```

Laravel framework access: Premise

MITRE ATT&CK

Tactic: Initial access

Technique: Exploit public-facing application

- 3) For server in debug mode, specific data sent generates a debug file

Laravel framework access: Premise

MITRE ATT&CK

Tactic: Initial access

Technique: Exploit public-facing application

- 3) For server in debug mode, specific data sent generates a debug file
- 4) File contains .env variables including AWS credentials

```
:/tmp$ cat laravel.dump |grep -A 1 AWS_ACCESS_KEY
<td>AWS_ACCESS_KEY_ID</td>
<td><pre class=sf-dump id=sf-dump-639057879 data-indent-pad=" "><span class=sf-dump-str title="11 characters"
>AKIAEXAMPLE</span>"
--
<td>AWS_ACCESS_KEY_ID</td>
<td><pre class=sf-dump id=sf-dump-1669199535 data-indent-pad=" "><span class=sf-dump-str title="11 characters"
">AKIAEXAMPLE</span>"
:/tmp$
:/tmp$ cat laravel.dump |grep -A 1 SECRET_ACCESS_KEY
<td>AWS_SECRET_ACCESS_KEY</td>
<td><pre class=sf-dump id=sf-dump-1009507608 data-indent-pad=" "><span class=sf-dump-str title="11 characters"
">AKIAEXAMPLE</span>"
--
<td>AWS_SECRET_ACCESS_KEY</td>
<td><pre class=sf-dump id=sf-dump-874671302 data-indent-pad=" "><span class=sf-dump-str title="11 characters"
">AKIAEXAMPLE</span>"
:/tmp$
:/tmp$
:/tmp$
:/tmp$
:/tmp$
```

Keys

Laravel framework access: Mitigations

MITRE ATT&CK

Tactic: Initial access

Technique: Exploit public-facing application

- Confirm Laravel is up-to-date and fully patched
- Disable debug mode in production – set `APP_DEBUG = FALSE`
- Use principle of least privilege for credentials in Laravel `.env`
- AWS Secrets Manager for hardcoded secrets

CloudTrail modification: Premise

MITRE ATT&CK

Tactic: Defense evasion

Technique: Impair defenses

- 1) Threat actor gains access to AWS account

CloudTrail modification: Premise

- 1) Threat actor gains access to AWS account
- 2) Modifies CloudTrail using PutEventSelectors

MITRE ATT&CK

Tactic: Defense evasion

Technique: Impair defenses

Events [Info](#)
Record API activity for individual resources, or for all current and future resources in AWS account. [Additional charges apply](#)

Event type
Choose the type of events that you want to log.

☒ **Management events**
Capture management operations performed on your AWS resources.

Management events [Info](#)
Management events show information about management operations performed on resources in your AWS account.

Multiple management events trails detected. Charges apply to duplicated logged management events. [Additional charges apply](#)

API activity
Choose the activities you want to log.

☒ **Read**

☐ Exclude AWS KMS events

☐ Exclude Amazon RDS Data API events

☒ **Write**

Write events deselected

[Cancel](#) [Save changes](#)

CloudTrail modification: Premise

- 1) Threat actor gains access to AWS account
- 2) Modifies CloudTrail using PutEventSelectors
- 3) Prevents logging of mutating events

MITRE ATT&CK

Tactic: Defense evasion

Technique: Impair defenses

```
"eventTime": "██████████",
"eventSource": "cloudtrail.amazonaws.com",
"eventName": "PutEventSelectors",
"awsRegion": "us-west-2",
"sourceIPAddress": "██████████",
"userAgent": "AWS Internal",
"requestParameters": {
  "trailName": "██████████",
  "eventSelectors": [
    {
      "readWriteType": "ReadOnly",
      "includeManagementEvents": true,
      "dataResources": [],
      "excludeManagementEventSources": []
    }
  ]
},
"responseElements": {
```

PutEventSelectors event record

CloudTrail trail name

Event selector set to ReadOnly

CloudTrail modification: Alternative

MITRE ATT&CK

Tactic: Defense evasion

Technique: Impair defenses

- 1) Threat actor gains access to AWS account
- 2) Modifies CloudTrail using `PutEventSelectors`

CloudTrail modification: Alternative

- 1) Threat actor gains access to AWS account
- 2) Modifies CloudTrail using PutEventSelectors
- 3) Prevents logging of management events

MITRE ATT&CK

Tactic: Defense evasion

Technique: Impair defenses

Events [Info](#)
Record API activity for individual resources, or for all current and future resources in AWS account. [Additional charges apply](#)

Event type
Choose the type of events that you want to log.

☒ **Management events**
Capture management operations performed on your AWS resources.

Management events [Info](#)
Management events show information about management operations performed on resources in your AWS account.

API activity
Choose the activities you want to log.

☒ Read ☒ Write

☐ Exclude AWS KMS events

☐ Exclude Amazon RDS Data API events

[Cancel](#) [Save changes](#)

CloudTrail modification: Alternative

- 1) Threat actor gains access to AWS account
- 2) Modifies CloudTrail using PutEventSelectors
- 3) Prevents logging of management events

MITRE ATT&CK

Tactic: Defense evasion

Technique: Impair defenses

```
"eventTime": "██████████",
"eventSource": "cloudtrail.amazonaws.com",
"eventName": "PutEventSelectors",
"awsRegion": "us-west-2",
"sourceIPAddress": "██████████",
"userAgent": "AWS Internal",
"requestParameters": {
  "trailName": "██████████",
  "eventSelectors": [
    {
      "readWriteType": "All",
      "includeManagementEvents": false,
      "dataResources": [
        {
          "type": "AWS::S3::Object",
          "values": [
            "██████████",
            "██████████",
            "██████████"
          ]
        }
      ]
    }
  ]
}
```

PutEventSelectors event record

CloudTrail trail name

Event selector with includeManagementEvents to false

CloudTrail modification: Mitigations

MITRE ATT&CK

Tactic: Defense evasion

Technique: Impair defenses

- Use SCPs to restrict CloudTrail modification including use of `PutEventSelectors` API
- Consider AWS Config remediation rules for CloudTrail

LLM resource hijacking: Premise

MITRE ATT&CK

Tactic: Impact

Technique: Resource hijacking

- 1) Threat actor obtains access to AWS account

LLM resource hijacking: Premise

MITRE ATT&CK

Tactic: Impact

Technique: Resource hijacking

- 1) Threat actor obtains access to AWS account
- 2) Threat actor enables access to LLMs through Amazon Bedrock

LLM resource hijacking: Premise

MITRE ATT&CK
Tactic: Impact
Technique: Resource hijacking

- 1) Threat actor obtains access to AWS account
- 2) Threat actor enables access to LLMs through Amazon Bedrock

Amazon Bedrock > Model access

Model access info

To use Bedrock, you must request access to Bedrock's FMs. To do so, you will need to have the correct [IAM Permissions](#). For certain models, you may first need to submit use case details before you are able to request access. More information about these models is available on the [Providers](#) page.

Base models (29)

Manage model access

Models	Access status	Modality	EULA
AI21 Labs			
— Jurassic-2 Ultra	Access granted	Text	EULA
— Jurassic-2 Mid	Access granted	Text	EULA
Amazon			
— Titan Embeddings G1 - Text	Access granted	Embedding	EULA
— Titan Text G1 - Lite	Access granted	Text	EULA
— Titan Text G1 - Express	Access granted	Text	EULA
— Titan Image Generator G1	Access granted	Image	EULA
— Titan Multimodal Embeddings G1	Access granted	Embedding	EULA
— Titan Text Embeddings V2	Available to request	Embedding	EULA
Anthropic			
— Claude 3 Opus	Unavailable	Text & Vision	EULA
— Claude 3 Sonnet	Available to request	Text & Vision	EULA
— Claude 3 Haiku	Available to request	Text & Vision	EULA
— Claude	Access granted	Text	EULA
— Claude Instant	Access granted	Text	EULA
Cohere			



LLM resource hijacking: Premise

MITRE ATT&CK

Tactic: Impact

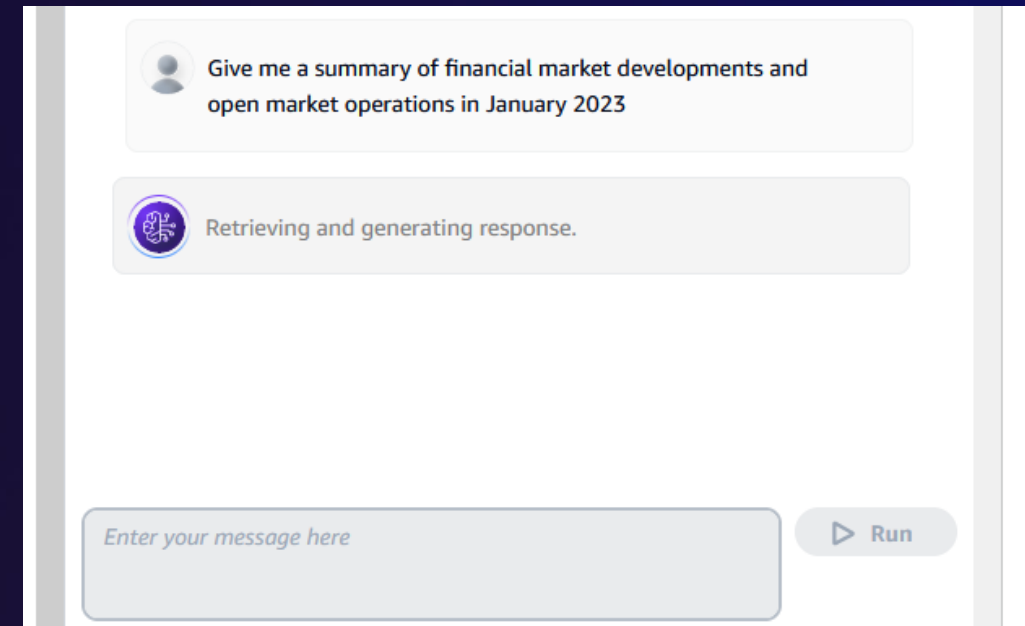
Technique: Resource hijacking

1) Threat actor obtains access to AWS account

2) Threat actor enables access to LLMs through Amazon Bedrock

3) Models used and prompts sent:

- `InvokeModel`
- `InvokeModelWithResponseStream`



LLM resource hijacking: Premise

MITRE ATT&CK

Tactic: Impact

Technique: Resource hijacking

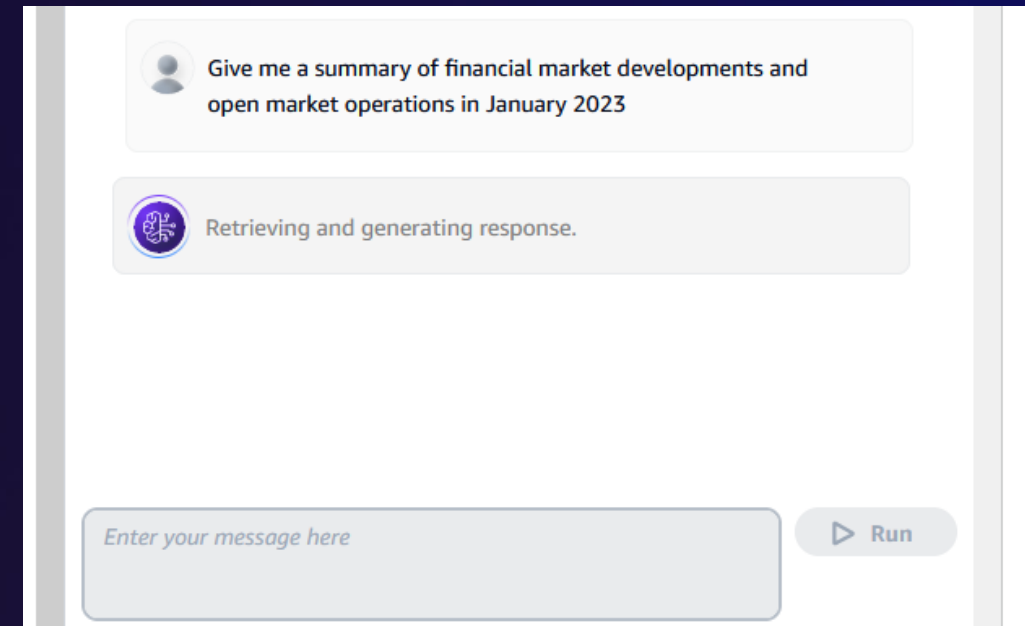
1) Threat actor obtains access to AWS account

2) Threat actor enables access to LLMs through Amazon Bedrock

3) Models used and prompts sent:

- `InvokeModel`
- `InvokeModelWithResponseStream`

4) Can be performed in unused AWS Regions



LLM resource hijacking: Mitigations

MITRE ATT&CK

Tactic: Impact

Technique: Resource hijacking

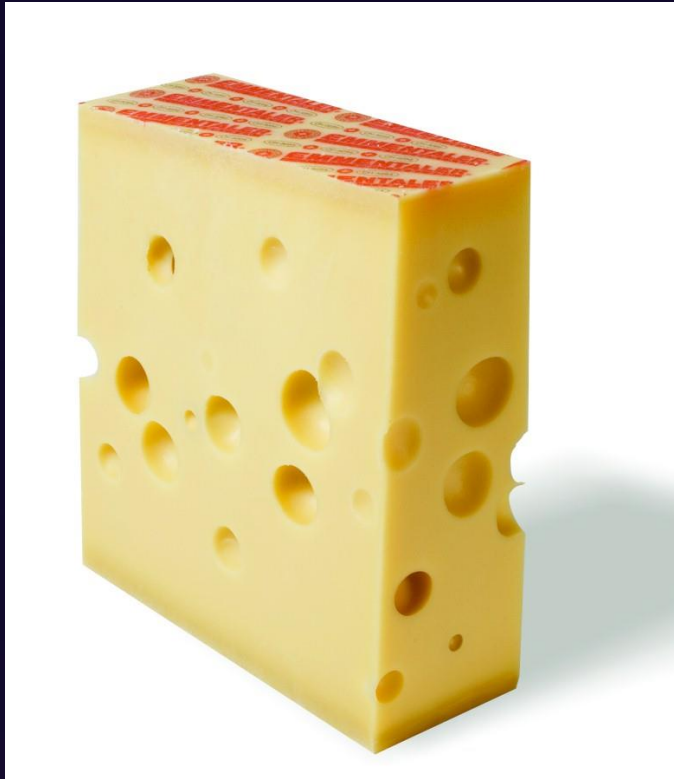
Use SCPs to limit access to Amazon Bedrock using

- Specific principals
- Specific Regions

Security best practices

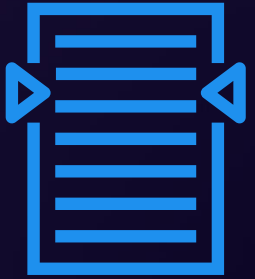


Swiss cheese model (industrial accidents)

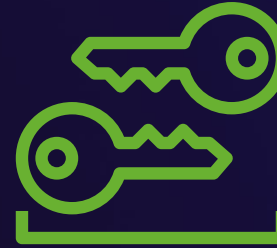


A failure **cannot** be traced back to a **single root cause**;
accidents are often the **result of a combination of factors**

Get the basics right



Inaccurate AWS account contact information



Unintended disclosure of credentials and secrets



Ineffective response to detective controls



Lack of continuous vulnerability management



Insecure AWS resource configuration