ownux

SNDK Corp

Report Name: First Report

Cloud Configuration Review Report

Friday, April 07, 2023

ADDRESS

1117, 11th Floor, Shivalik Satyamev,
Near Vakil Bridge, SP. Ringroad Bopal
Ahmedabad – 380058, Gujarat

CONTACT

P : +91 8469305424
E: contact@ownuxglobal.com
W: www.ownuxglobal.com

ownux

Key Agenda

Sec 01	**COMPANY OVERVIEW**	
03	About Ownux	
04	Work Process	
Sec 02	**PROJECT DETAILS**	
05	About Project	
05	Scope of Work	
05	Timelines	
06	Attack Narrative	
Sec 03	**TECHNICAL DETAILS**	
07	Findings	
09	Detailed Proof of Concept	
Sec 04	**RECOMMENDATION**	
98	Recommendation Summary	

Ownux is an Information Security Consultation firm specializing in the field of Penetration Testing of every channel which classifies different security areas of interest within an organization.

We are focused on Application Security, however, we are not limited to physical cyber security, reviewing the configurations of applications and security appliances.

We easily get inclined with your in-house team to work collaboratively and with less hassles to kick-start new projects and to bring the best out of the on-going engagements. We follow and use Workflow Process Management which includes best methods and tools that can be used to evaluate risk, both within the project and within the client's network architecture which ensures the best deliverables.

We understand that identifying vulnerabilities and exploits within a professional engagement is not enough. We strive to diligently follow the process in order to ensure desired quality and efficiency.

ownux

Work Process

1

Scope Definition

We understand and define a scope here where the Penetration Testing has to be done. The scope include an application, devices or a network. It also defines time duration of the entire engagement that has to be conducted.

2

Information Gathering

Information Gathering is the process where we try to collect scope information using various techniques to understand the attacking approach. This helps our security analysts to carry out and plan the attack.

3

Vulnerability Discovery

It is a process where we find weaknesses in the defined scope using manual and automated approach against the defined scope. This includes both Technical (NIST and OWASP Controls) and Business Logic Vulnerabilities.

4

Exploitation & Post Exploitation

This process includes exploiting the weaknesses found in the previous process. We try to penetrate into the given asset to find the maximum impact if it was an attacker instead. This gives a better idea of even internal Vulnerabilities if any.

5

Analysis & Reporting

We analyse and review the real impact of the exploit and put them down for you in a fine report which is easy to understand. This includes overviews and technical depth to help developers or system administrators to fix the issue.

About Project

The Project is AWS Cloud configuration review of Servers for Sndkcorp which also covers CIS Benchmarking and best practices as per the platform.

Created By:

Rudransh Jani

Approved By:

Bhashit Pandya

Date: 04/07/2023

Ownux Infosec Private Limited

1117, 11th Floor, Shivalik Satyamev,
Near Vakil Bridge, SP. Ringroad Bopal
Ahmedabad – 380058,
Gujarat

Scope of Work

The Following hosts were considered to be the part of this engagement.

No.	Scope	Description
1.	AWS	One-Time Cloud Configuration Review

Timelines

The testing activities were performed between 04/03/2023 and 04/07/2023.

Milestone	Start Date	End Date
Information Gathering	04/03/2023	04/03/2023
Technical Analysis	04/04/2023	04/04/2023
Technical Audit	04/05/2023	04/07/2023
Reporting	04/07/2023	04/07/2023
Approval	04/07/2023	04/07/2023

Attack Narrative

Day 1: We collected all the information required for the assessment. We did set up the environment for further analysis.

Day 2: We did technical analysis and verified all the findings. Collected evidences for the same.

Day 3: We initiated reporting. Verified all the instances with our findings. Removed false-positives.

Day 4: We did approvals and further eliminated false-positives from the report. Created a final draft of the report.

ownux

Findings

Title	Risk	Status
AMI Storage Not Encrypted	High	Vulnerable
EFS Encryption Not Enabled	High	Vulnerable
S3 – GET Actions Authorized To All Principal	High	Vulnerable
SSH Open To All	High	Vulnerable
IAM Managed Policy Allows iam:PassRole*	High	Vulnerable
Root Account Has Active X.509 Cert	High	Vulnerable
Lack Of Key Rotation	High	Vulnerable
RDP Port Open To All	Medium	Vulnerable
TCP Port Open To All	Medium	Vulnerable
EBS Volume Not Encrypted	Medium	Vulnerable
ELBv2 Access Log Not Enabled	Medium	Vulnerable
ELBv2 Lack Of Deletion Protection	Medium	Vulnerable
CloudFront Viewer Protocol Policy	Medium	Vulnerable
FTP Port Open	Medium	Vulnerable
Security Group Whitelist Too Permissive CIDRs	Medium	Vulnerable
Unencrypted ECR Repositories	Medium	Vulnerable
ECR Scan-On-Push Disabled	Medium	Vulnerable
IAM Cross-Account Access Lacks External ID and MFA	Medium	Vulnerable
Inline Role Policy Allows iam:PassRole	Medium	Vulnerable

Bucket Allowing Clear Text (HTTP) Communication	Medium	Vulnerable
Redshift SSL Not Enabled	Medium	Vulnerable
S3 Bucket Without Delete MFA	Medium	Vulnerable
Lack Of MFA (Root Account)	Medium	Vulnerable
MsSQL Port Open To All	Medium	Vulnerable
User Without MFA	Low	Vulnerable
S3 Bucket Logging Disabled	Low	Vulnerable
S3 Bucket Without Versioning	Low	Vulnerable
Single AZ RDS Instance	Low	Vulnerable

AMI Storage Not Encrypted

Risk:	**High**	**STATUS:** Vulnerable
CVSS Score:	**8.2**	

(AV:Network/AC:Low/PR:None/UI:None/C:Low/I:High/A:None)

Description

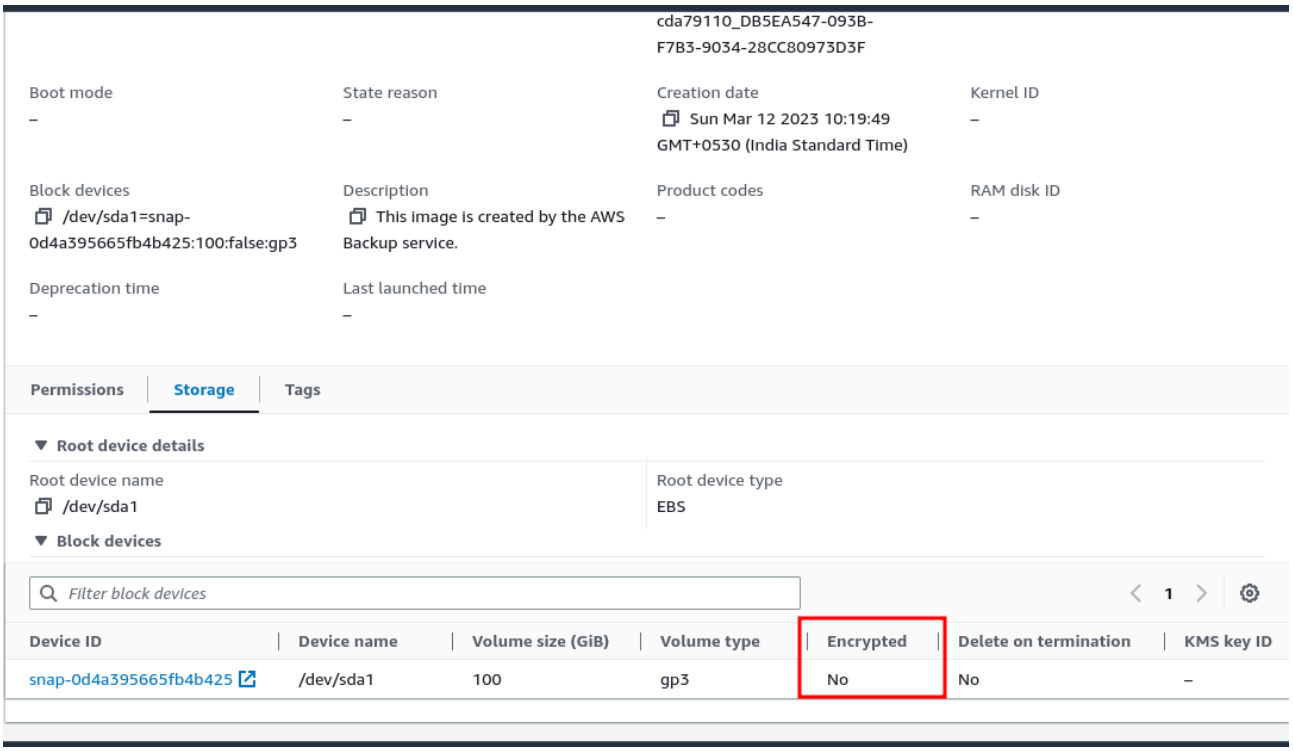
When dealing with production data that is crucial to your business, it is highly recommended to implement data encryption in order to protect it from attackers or unauthorized personnel. The AMI encryption keys are using AES-256 algorithm and are entirely managed and protected by the AWS key management infrastructure through AWS Key Management Service (KMS).

Ensure that your Amazon Machine Images (AMIs) are encrypted to fulfill compliance requirements for data-at-rest encryption. The Amazon Machine Image (AMI) data encryption and decryption is handled transparently and does not require any additional action from your applications. This rule can help you with the following compliance standards:

1. PCI
2. HIPAA
3. GDPR
4. APRA
5. MAS
6. NIST4

Step by Step Proof

1. Sign in to the AWS Management Console.
2. Navigate to EC2 dashboard at https://console.aws.amazon.com/ec2/.
3. In the left navigation panel, under IMAGES section, choose AMIs and observe that ecryption is not enabled.



Remediation

To encrypt any unencrypted Amazon Machine Images available within your AWS account, you need to create AMIs with encrypted snapshots from AMIs with unencrypted snapshots by copying them. To implement the AMI encryption process, perform the following:

01. Sign in to the AWS Management Console.
02. Navigate to EC2 dashboard at https://console.aws.amazon.com/ec2/.
03. In the left navigation panel, under **IMAGES** section, choose **AMIs**.
04. Select the image that you want to encrypt.
05. Click the **Actions** dropdown button from the dashboard top menu and select **Copy AMI**.
06. Inside **Copy AMI** dialog box, perform the following actions:
    a) Select the new AMI destination region from the **Destination region** dropdown list.
    b) Within **Name** box, provide a name for your new AMI.
    c) (Optional) Edit the image description available within **Description** box.
    d) Next to **Encryption**, select **Encrypt target EBS snapshots** checkbox then choose the required KMS master key (the key used to encrypt the target snapshot) from the **Master Key** dropdown list. If there are no KMS CMK keys already created, you can use the default master key (i.e. **(default) aws/ebs**) that protects your EBS volumes and snapshots when no other key is defined.
    e) Click **Copy AMI** to confirm the action then click **Done** to return to the

EC2 dashboard. The copy operation should take few minutes. Once the process is complete, the new AMI status should change from **pending** to **available**.

07. Repeat steps no. 4-6 to encrypt other unencrypted AMIs available within the current region.
08. Change the AWS region from the navigation bar and repeat the entire process for the other regions.

Instances

- Through out

References

- https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AMIEncryption.html
- https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AMIs.html

EFS Encryption Not Enabled

Risk:	**High**	**STATUS:** Vulnerable
CVSS Score:	**8.2**	

(AV:Network/AC:Low/PR:None/UI:None/C:High/I:Low/A:None)

Description

It is recommended to encrypt EFS file systems in order to protect your data and metadata from unauthorized access and fulfill compliance requirements for data-at-rest encryption within your organization.

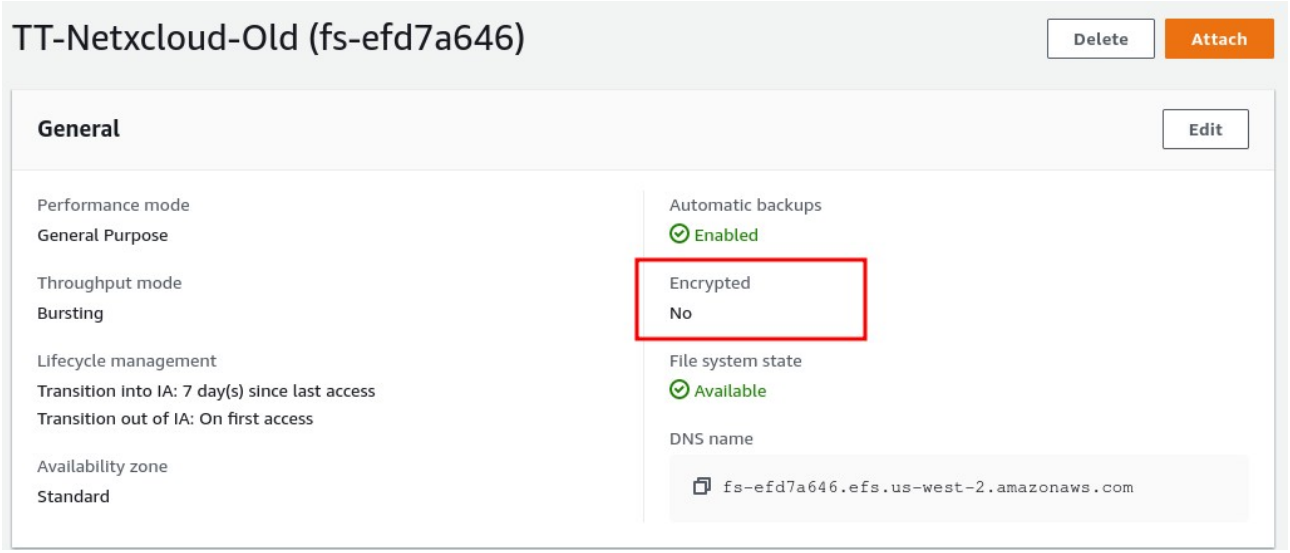
Ensure that your Amazon EFS file systems are encrypted in order to meet security and compliance requirements. Your data is transparently encrypted while being written and transparently decrypted while being read from your file system, therefore the encryption process does not require any additional action from you or your application. Encryption keys are managed by AWS KMS service, eliminating the need to build and maintain a secure key management infrastructure.

This rule can help you with the following compliance standards:

1. HIPAA
2. GDPR
3. APRA
4. MAS
5. NIST4

Step by Step Proof

1. Login to the AWS Management Console.
2. Navigate to Elastic File System (EFS) dashboard at https://console.aws.amazon.com/efs/.
3. In the left navigation panel, select File Systems.



Remediation

To encrypt an existing AWS EFS file system you must copy the data from the existing file system onto the new one, that has the encryption feature enabled. To set up the new EFS file system, enable encryption, and copy your existing data to it, perform the following:

01. Login to the AWS Management Console.
02. Navigate to Elastic File System (EFS) dashboard at https://console.aws.amazon.com/efs/.
03. In the left navigation panel, select **File Systems**.
04. Click **Create File System** button from the dashboard top menu to start the file system setup process.
05. On the **Configure file system** access configuration page, perform the following actions:
    A. Choose the right **VPC** from the VPC dropdown list. Note that only the EC2 instances provisioned within the selected VPC can access the new file system.
    B. Within **Create mount targets** section, select the checkboxes for all of the Availability Zones (AZs) within the selected VPC. These are your mount targets. By selecting a mount target for each Availability Zone within the VPC, all the EC2 instances across your VPC can access the new file system.
    C. Click **Next step** to continue the setup process.

06. On the **Configure optional settings** page, perform the following:
    a. Inside **Add tags** section, create tags to describe your new file system.
    b. Select **General Purpose (default)** or **Max I/O** from **Choose performance mode** section to set up the performance mode for your file system based on your requirements.

c. Check Enable encryption checkbox and choose aws/elasticfilesystem from Select KMS master key dropdown list to enable encryption for the new file system using the default master key provided and managed by AWS KMS. This default master key is an AWS-managed key that is created automatically for the EFS service within your AWS account. To achieve better control over who can use the KMS key and access the encrypted data, you can create and manage your own KMS Customer Master Key (CMK) by following the instructions outlined in this conformity rule. Once your KMS CMK key is created, choose its alias from the Select KMS master key dropdown list. To use a CMK key from another AWS account, choose Enter a KMS key ARN from another account option and provide the ARN of the foreign KMS key inside the ARN/ID box.

d. Click **Next Step** to continue.

07. On the **Review and create** page, review the file system configuration details then click **Create File System** to create your new AWS EFS file system.

08. Now you can mount your file system from an EC2 instance with an NFSv4 client installed. You can also mount your file system from a on-premises server over an AWS Direct Connect connection. For EC2 mount and on-premises mount instructions use the links provided within the EFS confirmation message.

09. Copy the data from the source (old) EFS file system onto the new one.

10. As soon as the data migration process is completed and all the data is loaded into your new (encrypted) file system, you can remove the unencrypted file system from your AWS account to avoid further charges by performing the following actions:

a. Connect to your AWS EC2 instance and unmount the unencrypted EFS file system.

b. Choose the Amazon EFS file system that you want to delete from the list of file systems available.

c. Click the **Action dropdown** button from the dashboard top menu and select **Delete file system** option.

d. Inside the **Permanently delete file system** dialog box, type the file system ID for the EFS file system that you want to delete, then choose **Delete File System** to confirm the action. The removal process may take a few minutes to complete.

11. Repeat steps no. 4 - 10 to enable data-at-rest encryption for other Amazon EFS file system available in the current region.

12. Change the AWS region from the navigation bar and repeat the entire process for other regions.

Instances

- TT-Netxcloud-Old

References

- https://aws.amazon.com/efs/faq/
- https://docs.aws.amazon.com/efs/latest/ug/managing.html
- https://docs.aws.amazon.com/efs/latest/ug/gs-step-two-create-efs-resources.html

S3 - GET Actions Authorized To All Principal

Risk: **High** **STATUS:** Vulnerable

CVSS Score: **8.2**

(AV:Network/AC:Low/PR:None/UI:None/C:High/I:Low/A:None)

Description

The Principal element specifies the user, account, service, or other entity that is allowed or denied access to a resource. In Amazon S3, a Principal is the account or user who is allowed access to the actions and resources in the statement. When added to a bucket policy, the principal is the user, account, service, or other entity that is the recipient of this permission.

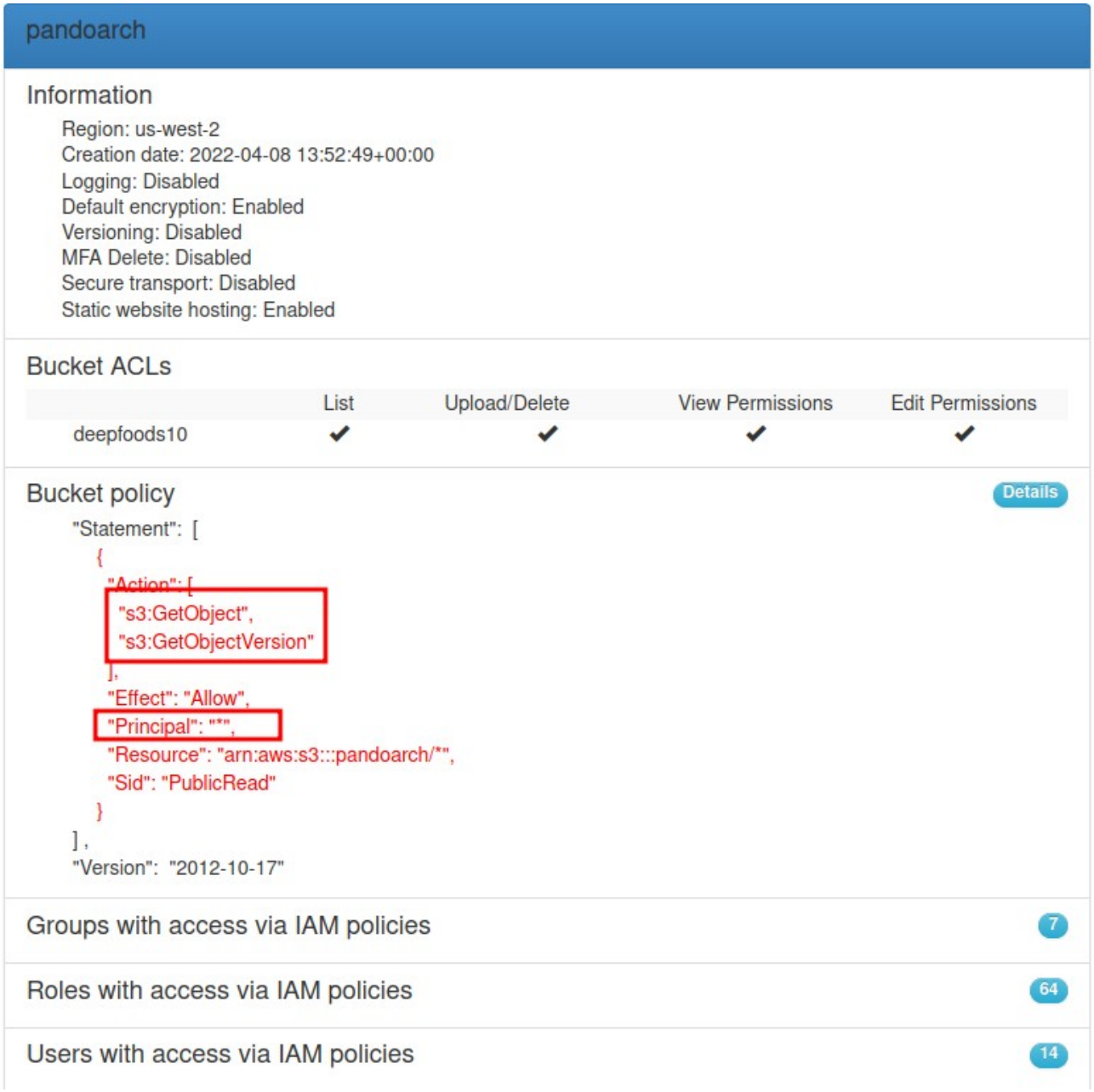
When you set the wildcard ("*") as the Principal value you essentially grant permission to everyone. This is referred to as anonymous access. The following statements are all considered Anonymous Permissions.

```
## Example 1
"Principal":"*"

## Example 2
"Principal":{"AWS":"*"}

## Example 2
"Principal":{"AWS":["*", ...]}
```

When you grant anonymous access, anyone in the world can access your bucket. It is highly recommend to **never** grant any kind of anonymous write access to your S3 bucket.

Step by Step Proof

1. Log in to the AWS Management Console at https://console.aws.amazon.com/
2. Open the Amazon S3 console.
3. Select the Permissions tab, then select Bucket Policy.



Remediation

To change the policy using the AWS Console, follow these steps:
01. Log in to the AWS Management Console at https://console.aws.amazon.com/
02. Open the Amazon S3 console.
03. Select the **Permissions** tab, then select **Bucket Policy**.
04. Remove policies for s3:List *actions for principals* ''. If necessary, modify the policy instead, to limit the access to specific principals.

Instances

- pandoarch
- devendra-img
- accordionwagebump
- betameet.teamlocus.com
- gbabu-img
- pandoarch-empschedule
- ttdrive
- meet.teamlocus.com
- call.teamlocus.com
- deepfoods

References

- https://docs.aws.amazon.com/AmazonS3/latest/userguide/managing-acls.html
- https://docs.aws.amazon.com/AmazonS3/latest/userguide/acl-overview.html
- https://docs.aws.amazon.com/AmazonS3/latest/userguide/s3-access-control.html

SSH Port Open To All

Risk: **High** **STATUS:** Vulnerable

CVSS Score: **7.3**

(AV:Network/AC:Low/PR:None/UI:None/C:Low/I:Low/A:Low)

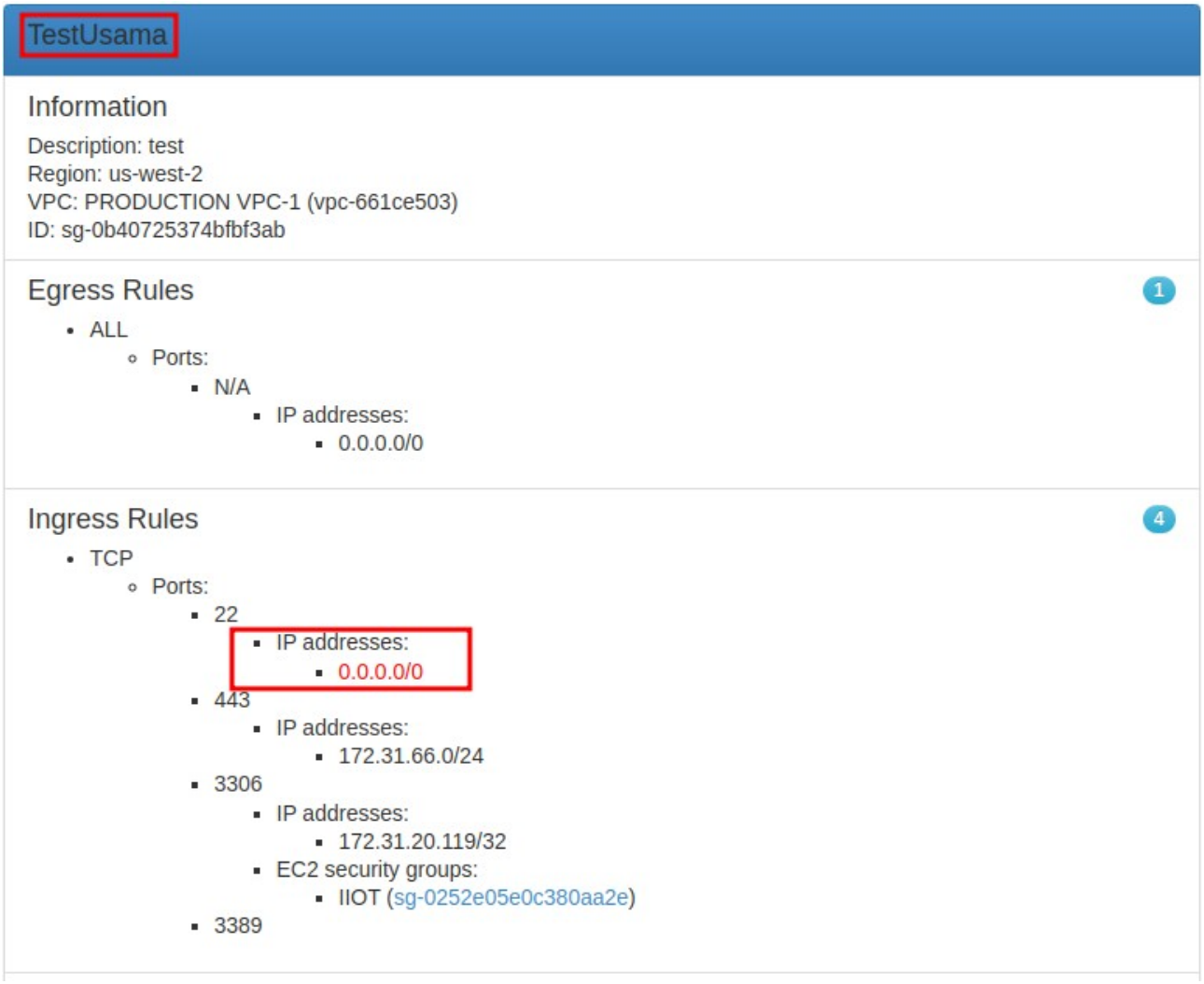
Description

Check your EC2 security groups for inbound rules that allow unrestricted access (i.e. 0.0.0.0/0 or ::/0) to TCP port 22. Restrict access to only those IP addresses that require it, in order to implement the principle of least privilege and reduce the possibility of a breach. TCP port 22 is used for secure remote login by connecting an SSH client application with an SSH server: https://en.wikipedia.org/wiki/Secure_Shell

It was observed that the configurations were allowing unrestricted SSH access that can increase opportunities for malicious activity such as hacking, man-in-the-middle attacks (MITM) and brute-force attacks.

Step by Step Proof

1. Sign in to the AWS Management Console.
2. Navigate to EC2 dashboard at https://console.aws.amazon.com/ec2/.
3. In the navigation panel, under NETWORK & SECURITY section, choose Security Groups.



Remediation

To update your security groups inbound/ingress configuration in order to restrict SSH access to specific entities (IP addresses, IP ranges, etc), perform the following:

01. Sign in to the AWS Management Console.
02. Navigate to EC2 dashboard at https://console.aws.amazon.com/ec2/.
03. In the navigation panel, under **NETWORK & SECURITY** section, choose **Security Groups**.
04. Select the appropriate security group (refer to the Audit section to identify the appropriate security group(s)).
05. Select the **Inbound** tab and click the **Edit** button.
06. In the **Edit inbound rules** dialog box, change the traffic **Source** for any inbound rules that allow unrestricted access through TCP port 22 by performing one of the following actions:

a. Select **My IP** from the **Source** dropdown list to allow inbound traffic only from your machine (from your IP address).
b. Select **Custom** from the **Source** dropdown list and enter one of the following options based on your access requirements:

i. The static IP/Elastic IP address of the permitted host with the suffix set to /32, e.g. 54.164.53.201/32.

ii. The IP address range of the permitted hosts in CIDR notation, for example 54.164.53.201/24.

c. The name or ID of another security group available in the same AWS region.
07. Click **Save** to apply the changes.
08. Repeat steps no. 4-7 to update other EC2 security groups that allow unrestricted SSH access.
09. Change the AWS region from the navigation bar and repeat the process for other regions.

Instances

- TestUsama
- default

References

- https://docs.aws.amazon.com/workspaces/latest/adminguide/connect-to-linux-workspaces-with-ssh.html

IAM Managed Policy Allows iam:PassRole*

Risk:	**High**	**STATUS:** Vulnerable
CVSS Score:	**7.3**	

(AV:Network/AC:Low/PR:None/UI:None/C:Low/I:Low/A:Low)

Description

Providing the right permissions for your IAM roles will significantly reduce the risk of unauthorized access (through API requests) to your AWS resources and services.

Note: You can set additional permission criteria within the rule settings available in the Cloud Conformity console.

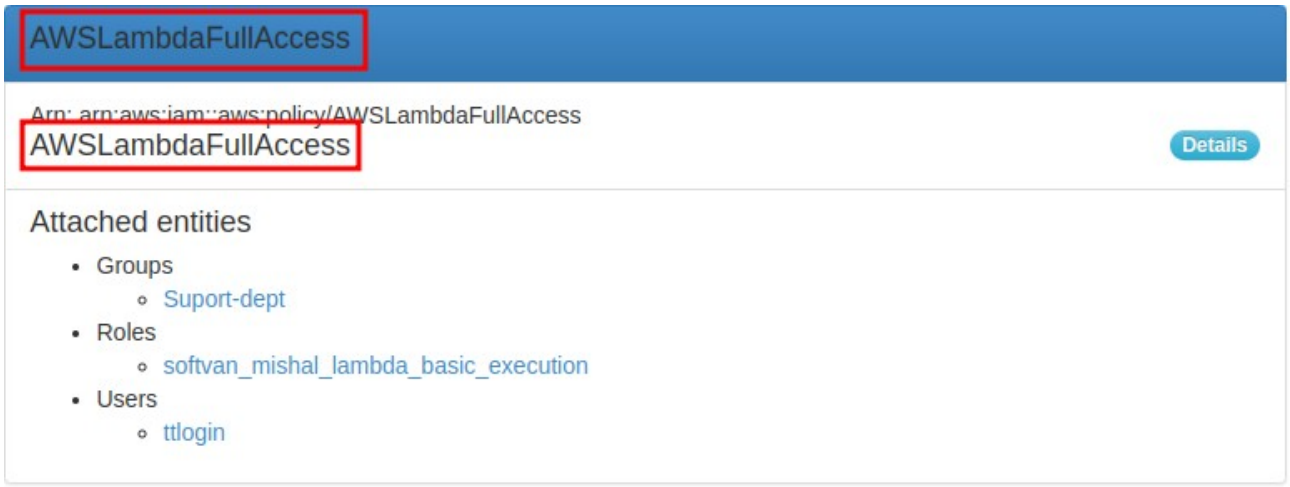
Ensure that the access policies attached to your IAM roles adhere to the principle of least privilege by giving the roles the minimal set of actions required to perform their tasks successfully.
This rule can help you with the following compliance standards:

1. APRA
2. MAS
3. NIST4

During the review, it was observed that the iam:passRole* was implemented.

Step by Step Proof

1. Sign in to the AWS Management Console.
2. Navigate to IAM dashboard at https://console.aws.amazon.com/iam/.
3. In the left navigation panel, choose Roles.
4. Click on the IAM role that you want to reconfigure.
5. On the IAM role configuration page, select the Permissions tab from the bottom panel.



Remediation

To update the IAM role policies that allow passing any roles to EC2 instances using "iam:PassRole" action, perform the following:

01. Sign in to the AWS Management Console.
02. Navigate to IAM dashboard at https://console.aws.amazon.com/iam/.
03. In the left navigation panel, choose **Roles**.
04. Click on the IAM role that you want to reconfigure.
05. On the IAM role configuration page, select the**Permissions** tab from the bottom panel.
06. Inside the **Managed Policies** and/or **Inline Policies** section(s), click on the policy name (link) to open the attached IAM policy for editing.
07. On the **Policy Details** page, select the **Policy Document** tab and click the **Edit** button to enter in the edit mode.
08. Update the selected policy by replacing the wildcard character (*) at the end of the resource ARN (e.g. "arn:aws:iam::123456789012:role/*") with a specific role name in order to limit this permission to a certain IAM role.
09. Click **Validate Policy** to validate the changes.
10. Click the **Save** button to apply the policy changes.

Instances

- AWSLambdaFullAccess
- IAMFullAccess
- AutoScalingServiceRolePolicy
- ServerMigrationServiceLaunchRole
- AWSDataPipelineRole
- AWSCloudTrailFullAccess
- AmazonSSMServiceRolePolicy
- AWSEC2SpotServiceRolePolicy

- AmazonEC2SpotFleetTaggingRole
- AWSElasticBeanstalkService
- ServerMigrationServiceRole
- passrole-apttoweruser
- Appsync_Amplify
- elasticbeanstalkfullpermssionforsatoru
- lambda-activate-deactivate-access-key-policy

References

- https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_use_passrole.html

Root Account Has Active X.509 Cert

Risk: **High** **STATUS:** Vulnerable

CVSS Score: **7.1**

(AV:Network/AC:Low/PR:Low/UI:None/C:High/I:Low/A:None)

Description

Disabling X.509 signing certificates created for your AWS root account eliminates the risk of unauthorized access to certain AWS services and resources, in case the private certificate keys are stolen or shared accidentally.
To secure your Amazon Web Services account and adhere to security best practices, ensure that your AWS root user is not using X.509 certificates to perform SOAP-protocol requests to AWS services. An X.509 certificate is a signing certificate utilized for API request validation purposes. Some AWS services use X.509 certificates to approve requests that are signed with a corresponding private key. Cloud Conformity strongly recommends disabling any active X.509 certificates deployed for your root account because using the root user to perform daily operations and develop AWS applications is not a best practice.

This rule can help you with the following compliance standards:

- APRA
- MAS
- NIST4

Step by Step Proof

1. Sign in to the AWS Management Console using the root account credentials.
2. Click on the AWS account name or number available in the upper-right corner of the management console and select My Security Credentials from the dropdown menu.
3. On Your Security Credentials page, click on the X.509 certificate tab to expand the panel with the X.509 certificates deployed for your root account.

Remediation

To disable any active X.509 signing certificates created for your AWS root account, perform the following actions:

Note: Disabling X.509 certificates deployed for your AWS root user via Command Line Interface (CLI) is not currently supported.

01. Sign in to the AWS Management Console using the root account credentials.
02. Click on the AWS account name or number available in the upper-right corner of the management console and select **My Security Credentials** from the dropdown menu.
03. On **Your Security Credentials** page, click on the **X.509 certificate** tab to expand the panel with the X.509 certificates deployed for your root account.
04. Choose the X.509 certificate that you want to disable (see Audit section part I to identify the right resource), then click on the required **Make Inactive** button, available within the **Actions** column, to disable the selected signing certificate. Once the certificate become inoperative, its status should change to **Inactive**.
05. Repeat steps no. 1-4 for each AWS root account that you want to secure by disabling its active X.509 certificates.

Instances

- AWS root account

References

- https://aws.amazon.com/iam/faqs/
- https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html
- https://docs.aws.amazon.com/iot/latest/developerguide/x509-client-certs.html
- https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_getting-report.html

Lack Of Key Rotation

Risk:	**High**	**STATUS:** Vulnerable
CVSS Score:	**7.1**	

(AV:Network/AC:Low/PR:Low/UI:None/C:Low/I:High/A:None)

Description

Rotating Identity and Access Management (IAM) credentials periodically will significantly reduce the chances that a compromised set of access keys can be used without your knowledge to access certain components within your AWS account.

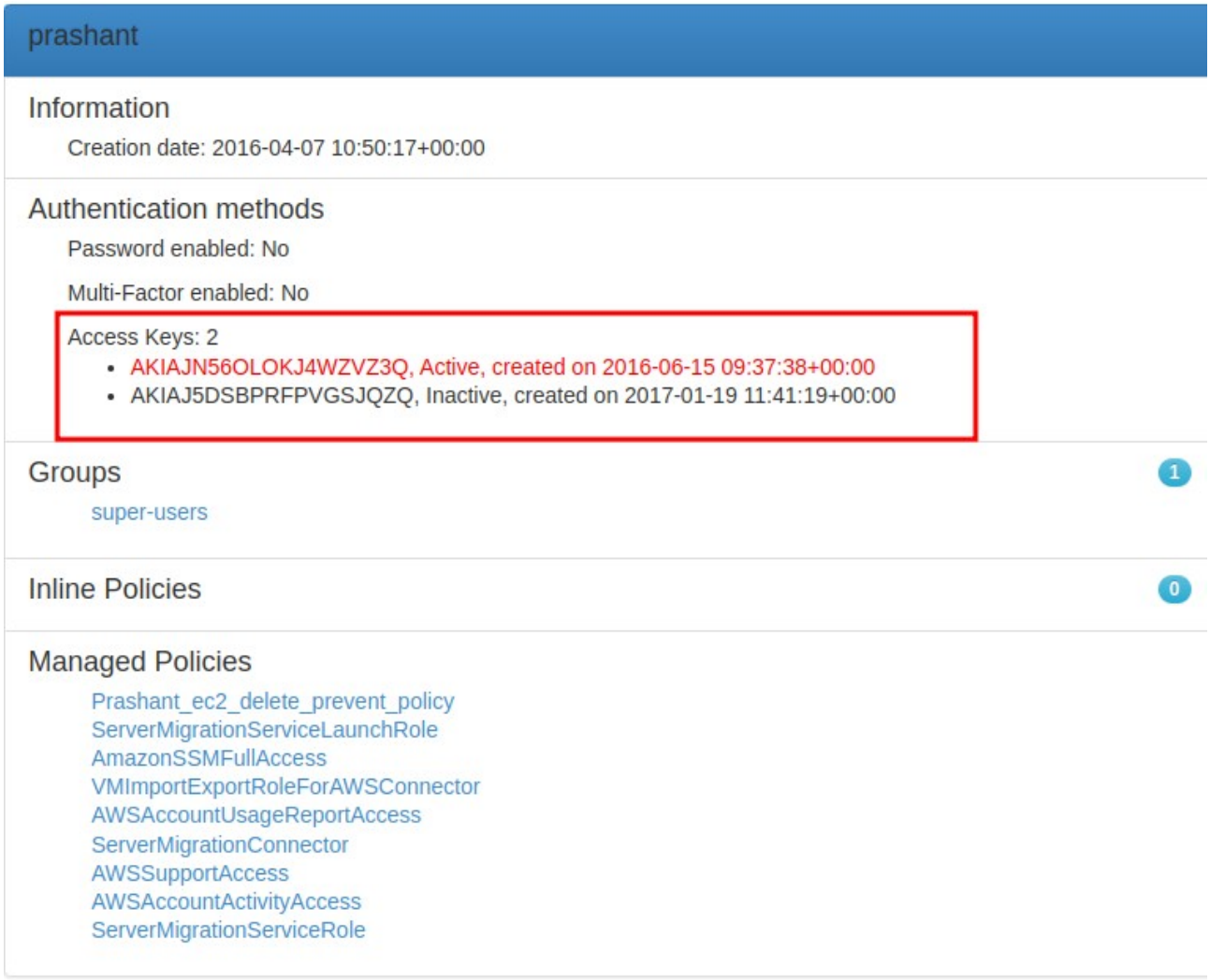
Ensure that all your IAM user access keys are rotated every month in order to decrease the likelihood of accidental exposures and protect your AWS resources against unauthorized access.

This rule can help you with the following compliance standards:

1. CISAWSF
2. APRA
3. MAS
4. NIST4

Step by Step Proof

1. Sign in to the AWS Management Console.
2. Navigate to IAM dashboard at https://console.aws.amazon.com/iam/
3. In the left navigation panel, choose **Users**.
4. Click on the IAM user name that you want to examine.



Remediation
To rotate (change) your outdated IAM access keys, you need to perform the following:
01. Sign in to the AWS Management Console.
02. Navigate to IAM dashboard at https://console.aws.amazon.com/iam/
03. In the left navigation panel, choose **Users**.
04. Click on the IAM user name that you want to examine.
05. On the IAM user configuration page, select **Security Credentials** tab.
06. Click **Create Access Key** to create a new set of access keys that will replace the old ones.
07. In the **Create Access Key** dialog box, click **Download Credentials** to save the newly created access key ID and secret access key to a CSV file on your machine. (!) IMPORTANT: AWS IAM will not provide access to the new secret access key again once the **Create Access Key** dialog box closes so make sure you save your credentials in a safe location on your machine.
08. Click **Close** to close the dialog box and return to the configuration page.

The IAM user should have now two active access keys.
09. Now update your application(s) code and replace the existing access key ID and secret access key with the new ones. Test your application(s) to make sure that the new access key pair is working.
10. Once the new key is validated, return to the IAM user configuration page, select the outdated (previous) key and click **Make Inactive** to change the state of the access key to inactive.
11. In the **Change Key Status** confirmation box, click **Deactivate** to deactivate the selected key. The access key status should change from **Active** to **Inactive**. (!) IMPORTANT: Cloud Conformity strongly recommends waiting few days before going forward with the next step in order to ensure that the original (outdated) key is no longer used by your application(s).
12. Once you are sure that the application(s) is/are no longer using the original key, return to the IAM user configuration page and remove the key by clicking the Delete link available in the Actions column.
13. In the **Delete Access Key** confirmation box, click **Delete** to remove the selected key.
14. Repeat steps no. 4-13 for each outdated (older than 90 days) IAM access key, available in your AWS account.

Instances

- s3backup
- prashant
- ttlogin
- zimbra_s3_user
- tasktower_sns
- apttower
- ses-beta-sm-grocerybabu-com
- secret-manager
- ses-smtp-sm-grocerybabu-com
- Mayurpatel
- kirtanchatbot
- apt-s3-backup
- Parth_BJP
- betateamlocus
- lamda_teamlocus
- ses-smtp-sm-themasalacompany-com
- satoru
- ses-smtp-user-sm-shopdeepfoods-com
- teamlocus_backup
- iiot
- parth-new-tlchat
- vijay-deepfoods

References

1. https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html
2. https://docs.aws.amazon.com/accounts/latest/reference/best-practices.html
3. https://aws.amazon.com/iam/faqs/

RDP Port Open To All

Risk:	**Medium**	**STATUS:** Vulnerable
CVSS Score:	**6.5**	

(AV:Network/AC:Low/PR:None/UI:None/C:Low/I:None/A:Low)

Description

Check your EC2 security groups for inbound rules that allow unrestricted access (i.e. 0.0.0.0/0 or ::/0) to TCP port 3389 and restrict access to only those IP addresses that require it in order to implement the principle of least privilege and reduce the possibility of a breach. TCP port 3389 is used for secure remote GUI login to Microsoft servers by connecting an RDP (Remote Desktop Protocol) client application with an RDP server: https://en.wikipedia.org/wiki/Remote_Desktop_Protocol.

This rule can help you with the following compliance standards:

1. CISAWSF
2. PCI
3. APRA
4. MAS
5. NIST4

It was observed that existing configurations allow unrestricted RDP access that can increase opportunities for malicious activity such as hacking, man-in-the-middle attacks (MITM) and Pass-the-Hash (PtH) attacks.

Step by Step Proof

1. Sign in to the AWS Management Console.
2. Navigate to EC2 dashboard at https://console.aws.amazon.com/ec2/.
3. In the navigation panel, under NETWORK & SECURITY section, choose Security Groups.

Remediation

To update your security groups inbound/ingress configuration in order to restrict RDP access to specific entities (IP addresses, IP ranges, etc), perform the following:

01. Sign in to the AWS Management Console.
02. Navigate to EC2 dashboard at https://console.aws.amazon.com/ec2/.
03. In the navigation panel, under **NETWORK & SECURITY** section, choose **Security Groups**.

04. Select the appropriate security group (see Audit section to identify the right one(s)).
05. Select the **Inbound** tab from the dashboard bottom panel and click the Edit button.
06. In the **Edit inbound rules** dialog box, change the traffic **Source** for any inbound rules that allow unrestricted access through TCP port 3389 by performing one of the following actions:
a. Select **My IP** from the **Source** dropdown list to allow inbound traffic only from your machine (from your IP address).
b. Select **Custom** from the **Source** dropdown list and enter one of the following options based on your access requirements:

i. The static IP/Elastic IP address of the permitted host with the suffix set to /32, e.g. 54.164.53.105/32.

ii. The IP address range of the permitted hosts in CIDR notation, for example 54.164.53.105/24.

c.The name or ID of another security group available in the same AWS region.
07. Click **Save** to apply the changes.
08. Repeat steps no. 4-7 to update other EC2 security groups that allow unrestricted RDP access.
09. Change the AWS region from the navigation bar and repeat the process for other regions.

Instances

- oops work

References

- https://www.intelligentdiscovery.io/controls/ec2/aws-ec2-rdp-open

TCP Ports Open To All

Risk: **Medium** **STATUS:** Vulnerable

CVSS Score: **5.3**

(AV:Network/AC:Low/PR:None/UI:None/C:Low/I:None/A:None)

Description

Ensure that your Amazon EC2 security groups don't allow unrestricted access (i.e. 0.0.0.0/0 or ::/0) in order to protect against attackers that use brute force methods to gain access to the EC2 instances associated with your security groups.

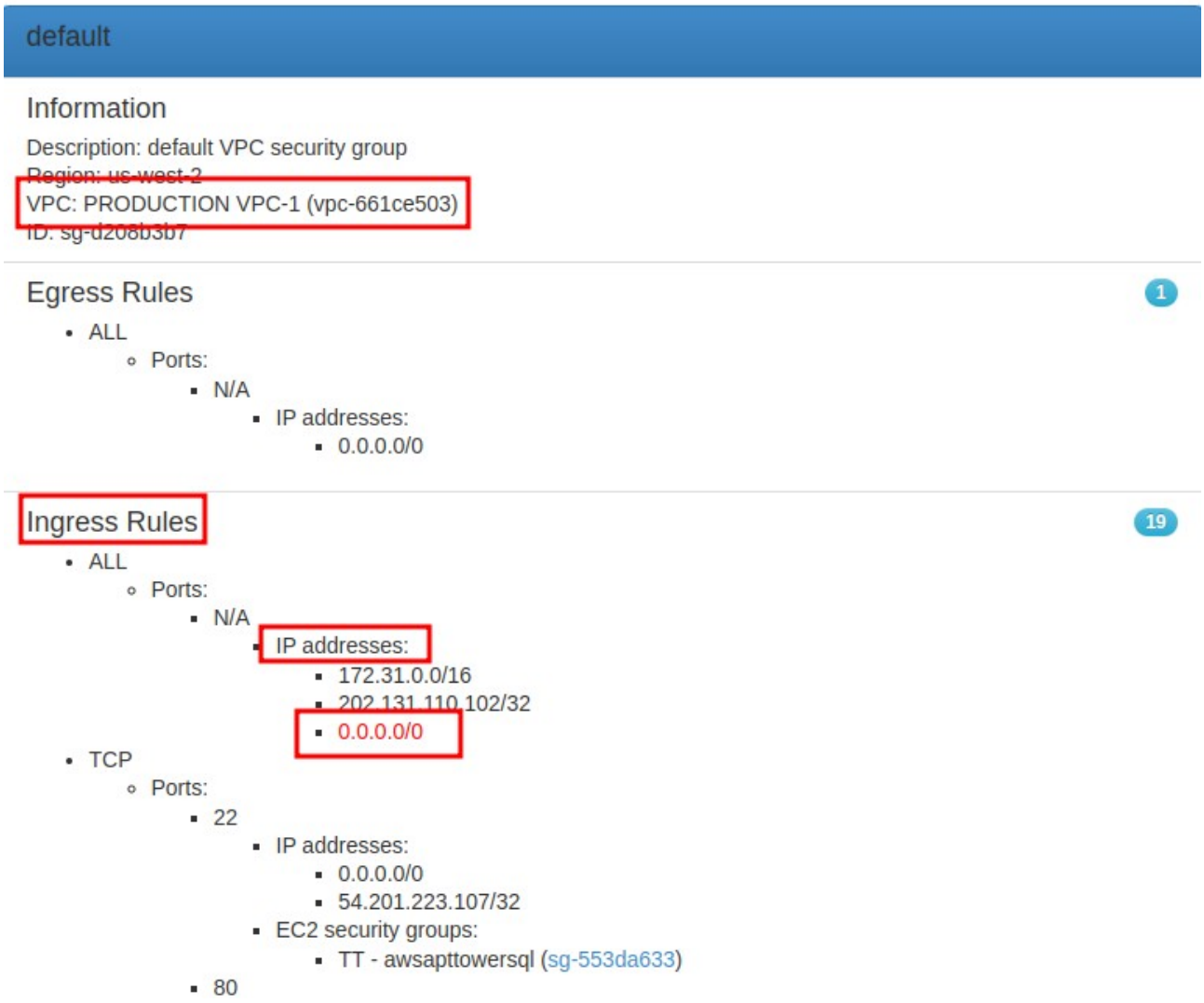
Ports can be any TCP/UDP port that is not inluded or included in the common service ports category, i.e. not commonly used or commonly used ports such as 80 (HTTP), 443 (HTTPS), 20/21 (FTP), 22 (SSH), 23 (Telnet), 53 (DNS), 3389 (RDP), 25/465/587 (SMTP), 3306 (MySQL), 5432 (PostgreSQL), 1521 (Oracle Database), 1433 (SQL Server), 135 (RPC), and 137/138/139/445 (SMB/CIFS).
This rule can help you with the following compliance standards:

1. PCI
2. HIPAA
3. APRA
4. MAS
5. NIST4

It was observed during the assessment that infrastructure allow unrestricted inbound/ingress access to Amazon EC2 instances on uncommon and common TCP/UDP ports that can increase opportunities for malicious activities such as hacking, data capture, and all kinds of attacks (brute-force attacks, Man-in-the-Middle attack, and Denial-of-Service attacks).

Step by Step Proof

1. Sign in to the AWS Management Console.
2. Navigate to EC2 dashboard at https://console.aws.amazon.com/ec2/.
3. In the navigation panel, under NETWORK & SECURITY section, choose Security Groups.



Remediation

To update the inbound rule configuration for your Amazon EC2 security groups in order to restrict access to trusted entities only (i.e. authorized IP addresses and IP ranges, or other security groups), perform the following operations:
CloudFormation template (JSON):

```
{
   "AWSTemplateFormatVersion":"2010-09-09",
   "Description":"Allow inbound access on uncommon ports to
trusted entities only",
   "Resources":{
    "CustomSecurityGroup" : {
         "Type" : "AWS::EC2::SecurityGroup",
         "Properties" : {
            "GroupDescription" : "Custom security group",
            "GroupName" : "custom-security-group",
```

```
            "VpcId" : "vpc-1234abcd",
            "SecurityGroupIngress" : [{
               "IpProtocol" : "tcp",
               "FromPort" : 8040,
               "ToPort" : 8040,
               "CidrIp" : "10.0.0.35/32"
            }],
            "SecurityGroupEgress" : [{
               "IpProtocol" : "-1",
               "FromPort" : 0,
               "ToPort" : 65535,
               "CidrIp" : "0.0.0.0/0"
            }]
         }
      }
   }
}
```

CloudFormation template (YAML):

```
AWSTemplateFormatVersion: '2010-09-09'
Description: Allow inbound access on uncommon ports to trusted entities only
Resources:
 CustomSecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: Custom security group
      GroupName: custom-security-group
      VpcId: vpc-1234abcd
      SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 8040
        ToPort: 8040
        CidrIp: 10.0.0.35/32
      SecurityGroupEgress:
      - IpProtocol: "-1"
        FromPort: 0
        ToPort: 65535
        CidrIp: 0.0.0.0/0
```

01. Sign in to the AWS Management Console.
02. Navigate to Amazon EC2 console at https://console.aws.amazon.com/ec2/.
03. In the navigation panel, under **Network & Security**, choose **Security Groups**.
04. Select the Amazon EC2 security group that you want to reconfigure (see Audit section part I to identify the right resource).
05. Select the **Inbound rules** tab from the console bottom panel and choose **Edit inbound rules**.
06. On the **Edit inbound rules** configuration page, change the traffic source for the inbound rule that allows unrestricted access through uncommon TCP/UDP ports, by performing one of the following actions:

a. Select **My IP** from the **Source** dropdown list to allow inbound traffic only from your current IP address.
b. Select **Custom** from the **Source** dropdown list and enter one of the following options based on your access requirements:

i. The static IP address of the permitted host in CIDR notation (e.g. 10.0.0.35/32).

ii. The IP address range of the permitted network/subnetwork in CIDR notation, for example 10.0.5.0/24.

c. The name or ID of another security group available in the same AWS cloud region.
d. Choose **Save rules** to apply the configuration changes.
07. Repeat steps no. 4 – 6 to reconfigure other EC2 security groups that allow unrestricted access on uncommon TCP/UDP ports.
08. Change the AWS cloud region from the navigation bar and repeat the remediation process for other regions.

Instances

- awseb-e-jnzjgnt3i2-stack-AWSEBSecurityGroup-VFB4DQN8B4FD
- Zimbra Mail SG
- Livekit Teamlocas
- credit-card-temp-test
- bhargav-python
- AWSTL Batcher

References

- https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/authorizing-access-to-an-instance.html

EBS Volumn Not Encrypted

Risk:	**Medium**	**STATUS:** Vulnerable
CVSS Score:	**5.3**	

(AV:Network/AC:Low/PR:None/UI:None/C:None/I:Low/A:None)

Description

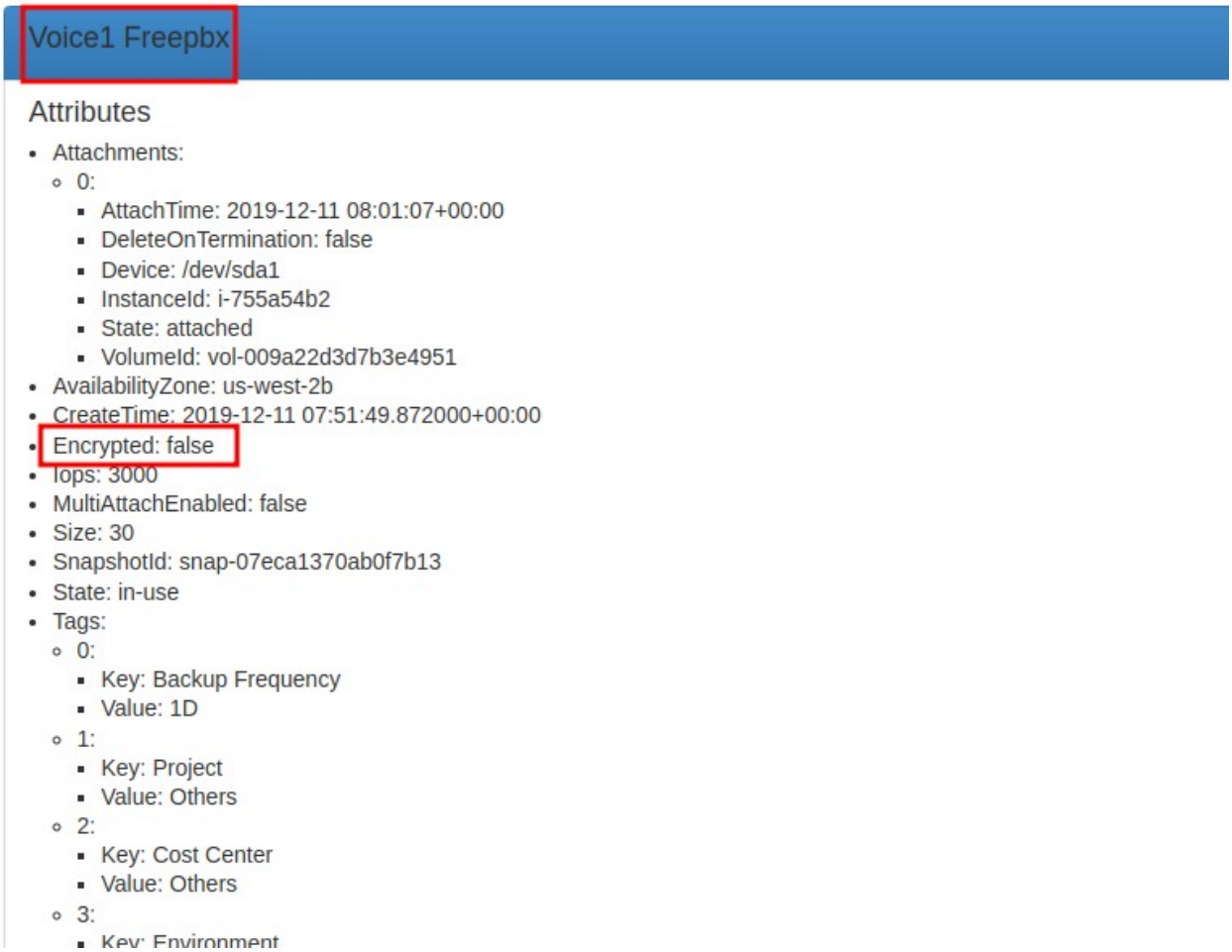
Ensure that all your Amazon Elastic Block Store (EBS) volumes are encrypted in order to meet security and compliance requirements. With encryption enabled, your EBS volumes can hold sensitive, confidential, and critical data. The data encryption and decryption process is handled transparently and does not require any additional action from you, your server instance, or your application.

When working with EBS data that is crucial to your business, it is strongly recommended to implement encryption at rest in order to protect your data from attackers or unauthorized personnel. When Encryption by Default feature is enabled, all new Amazon EBS volumes and copies of snapshots created in the specified region(s), are encrypted by default. If you implement Amazon IAM policies that require the use of encrypted EBS volumes, you can use this feature to avoid launch failures that would occur if unencrypted volumes were inadvertently referenced when an instance is launched. In this case, your SecOps team can enable encryption by default without having to coordinate with your development team and without performing additional operational changes. Your new EBS volumes can be encrypted with the AWS-managed master key, unless you specify a different key at launch time.

Note: Enabling this feature does not affect existing unencrypted Amazon EBS volumes.

Step by Step Proof

1. Sign in to AWS Management Console.
2. Navigate to Amazon EC2 console at https://console.aws.amazon.com/ec2/v2/.
3. Select the AWS cloud region that you want to access from the console navigation bar.
4. In the Account attributes section, under Settings, choose EBS encryption to access the EBS configuration settings available for the EBS volumes



Remediation

To enable encryption by default for your new Amazon EBS volumes, perform the following operations:

01. Sign in to AWS Management Console.
02. Navigate to Amazon EC2 console at https://console.aws.amazon.com/ec2/v2/.
03. Select the AWS cloud region that you want to access from the console navigation bar.
04. In the **Account attributes** section, under **Settings**, choose **EBS encryption** to access the EBS configuration settings available for the EBS volumes within the selected AWS region.
05. On the **Settings** page, select the **EBS encryption** tab, and click on the **Manage** button to modify the EBS feature settings.
06. On the **Modify EBS encryption** page, select **Enable** under **Always encrypt new EBS volumes** and click inside the Default encryption key configuration box to choose the master key to encrypt your EBS volumes. Choose **Update EBS encryption** to save the configuration changes. After you

enable EBS encryption by default, the Amazon EBS volumes that you create are always encrypted, either using the default master key or the Customer Master Key (CMK) that you specified when you created each volume.
07. Change the AWS region from the console navigation bar and repeat step no. 5 and 6 to enable encryption by default for the Amazon EBS volumes in other AWS cloud regions.

Instances

- Through out

References

- https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSEncryption.html

ELBv2 Access Log Not Enabled

Risk:	**Medium**	**STATUS:** Vulnerable
CVSS Score:	**5.3**	

(AV:Network/AC:Low/PR:None/UI:None/C:None/I:Low/A:None)

Description

Ensure that your Amazon Application Load Balancers (ALBs) have Access Logging feature enabled for security, troubleshooting and statistical analysis purposes.
This rule can help you with the following compliance standards:

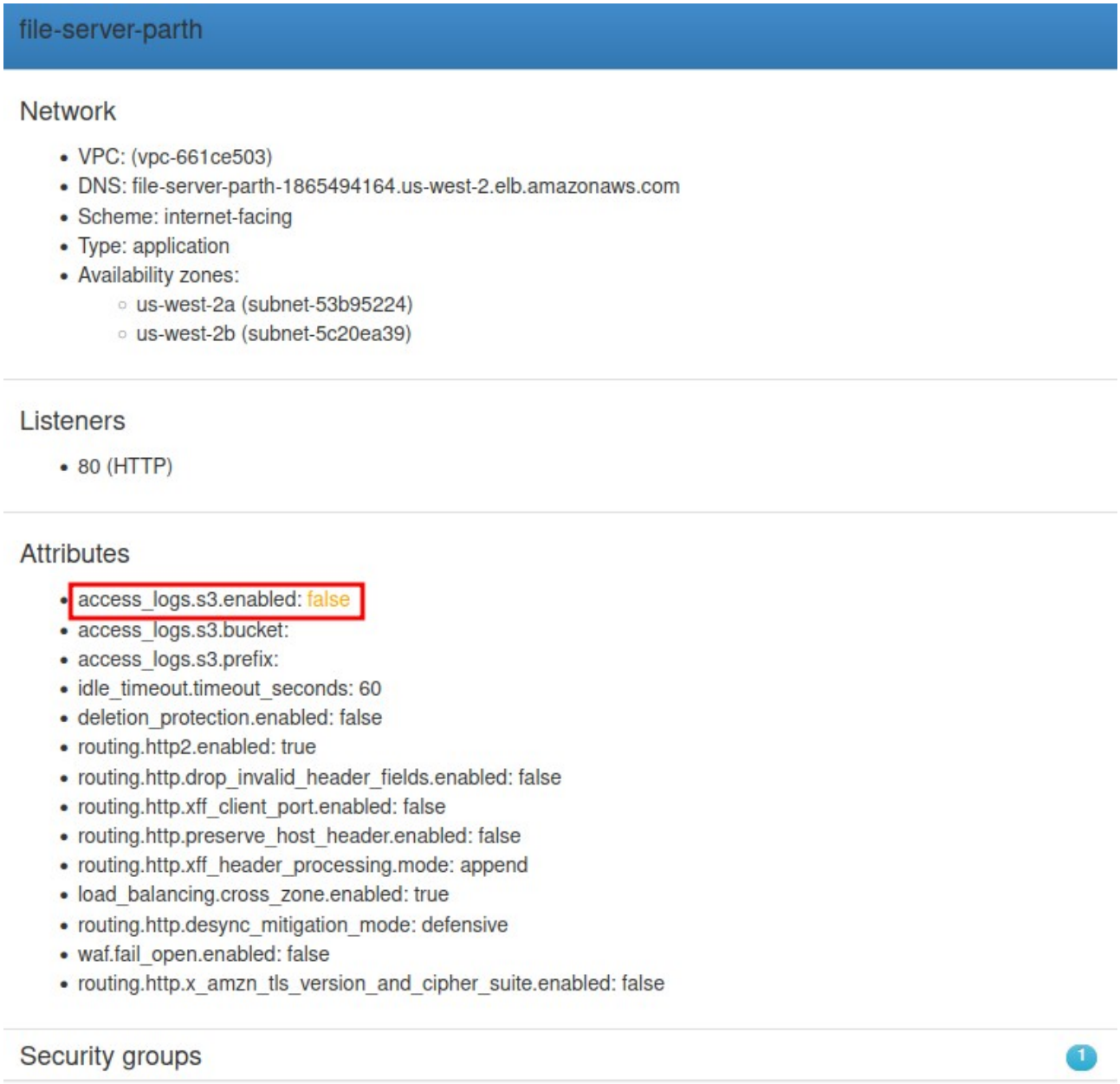
1. PCI
2. GDPR
3. APRA
4. MAS
5. NIST4

After you enable and configure access logging for your AWS Application Load Balancers, the log files will be delivered to the S3 bucket of your choice. The log files contain data about each HTTP/HTTPS request processed by the load balancer, data that can be extremely useful for analyzing traffic patterns, implementing protection plans and identifying and troubleshooting security issues.

During the review, we observed that Elbv2 logging of access is misconfigured and is disabled.

Step by Step Proof

1. Sign in to the AWS Management Console.
2. Navigate to EC2 dashboard at https://console.aws.amazon.com/ec2/.
3. In the left navigation panel, under LOAD BALANCING section, choose Load Balancers.
4. Select the Application Load Balancer that you want to reconfigure (see Audit section part I to identify the right resource).



Remediation

To enable access logging for your AWS Application Load Balancers (ALBs), perform the following actions:

01. Sign in to the AWS Management Console.
02. Navigate to EC2 dashboard at https://console.aws.amazon.com/ec2/.
03. In the left navigation panel, under **LOAD BALANCING** section, choose **Load Balancers**.
04. Select the Application Load Balancer that you want to reconfigure (see Audit section part I to identify the right resource).

05. Select **Description** tab from the dashboard bottom panel to view the ELBv2 resource description.
06. Within **Attributes** section, click **Edit attributes** button to access the load balancer attributes configuration.
07. Inside **Edit load balancer** attributes dialog box, set the following:
a. Check **Enable access logs** checkbox to enable the feature.
b. For **S3 location**, enter a unique name (e.g. alb-access-logging) and a prefix (optional) for the S3 bucket that will store the log files.
c. Check **Create this location for me** checkbox to enable Amazon Web Services to create the new bucket for you. If you don’t request this option, you must provide the name of an existing S3 bucket available in the same region with the selected load balancer.
d. Click **Save** to apply the changes. The **Access Logs** attribute value should now change to **Enabled**.
08. Repeat steps no. 4-7 to enable access logging for other AWS Application Load Balancers provisioned in the current region.
09. Change the AWS region from the navigation bar and repeat the remediation process for other regions.

Instances

- file-server-parth
- ALB-all-websites

References

- https://docs.aws.amazon.com/elasticloadbalancing/latest/application/load-balancer-access-logs.html

ELBv2 Lack of Deletion Protection

Risk:	**Medium**	**STATUS:** Vulnerable
CVSS Score:	**5.3**	

(AV:Network/AC:Low/PR:None/UI:None/C:None/I:None/A:Low)

Description

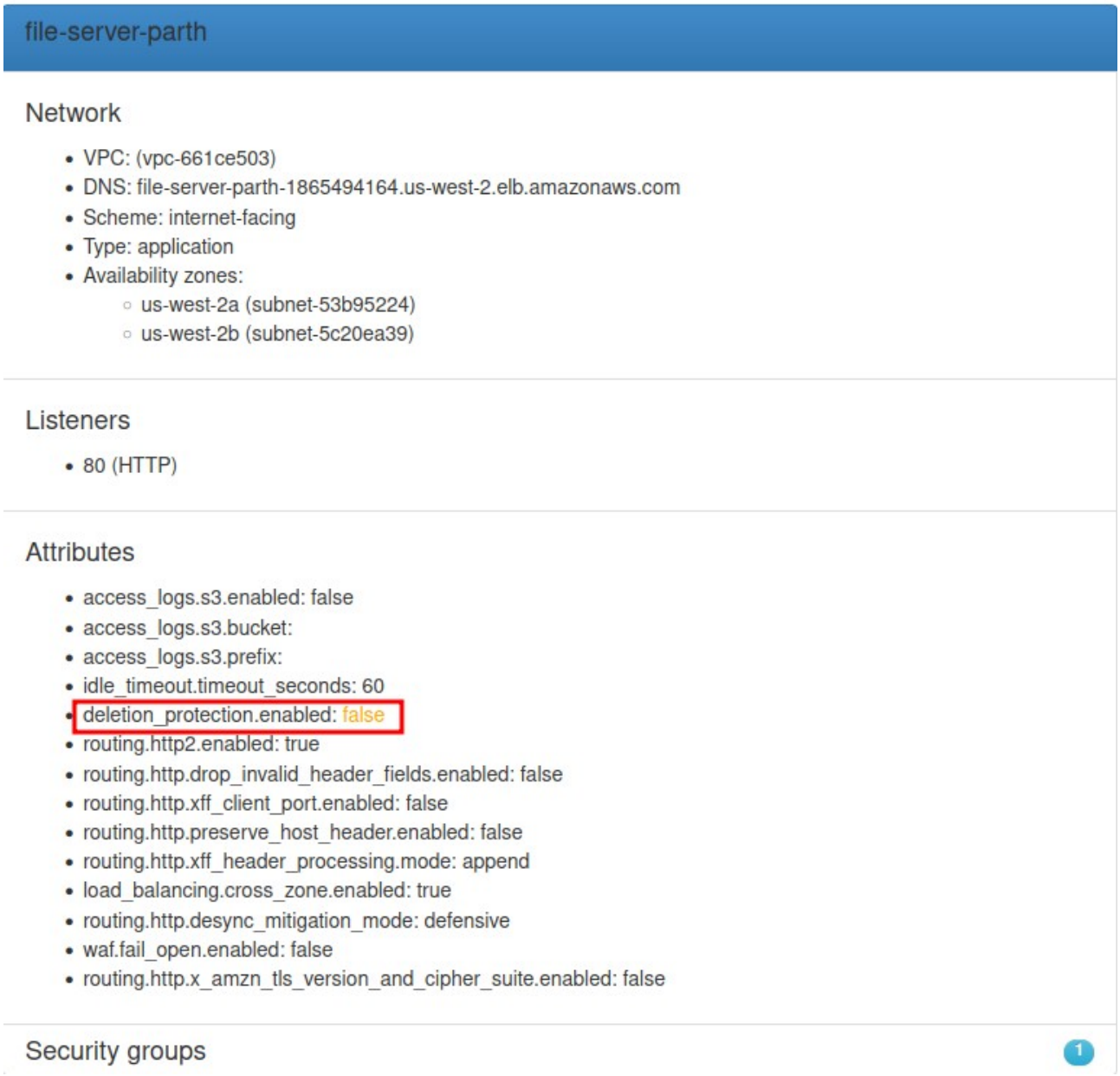
With Deletion Protection safety feature enabled, you have the guarantee that your AWS load balancers cannot be accidentally deleted and make sure that your load-balanced environments remain safe.

Ensure ELBv2 Load Balancers have Deletion Protection feature enabled in order to protect them from being accidentally deleted.

During the review it was discovered that it is misconfigured for detection and protection on ELBv2.

Step by Step Proof

01. Sign in to the AWS Management Console.
02. Navigate to EC2 dashboard at https://console.aws.amazon.com/ec2/.
03. In the left navigation panel, under **LOAD BALANCING** section, choose **Load Balancers**.
04. Select the ELBv2 load balancer that you want to reconfigure (see Audit section part I to identify the right resource).



Remediation

Using AWS Console:

01. Sign in to the AWS Management Console.
02. Navigate to EC2 dashboard at https://console.aws.amazon.com/ec2/.
03. In the left navigation panel, unde **LOAD BALANCING** section, choose **Load Balancers**.
04. Select the ELBv2 load balancer that you want to reconfigure (see Audit section part I to identify the right resource).
05. Select **Description** tab from the dashboard bottom panel to view the

resource description.
06. Within **Attributes** section, click **Edit attributes** button to access the load balancer attributes configuration.
07. Inside **Edit load balancer attributes** dialog box, select the checkbox next to **Enable deletion protection**, then click **Save** to apply the change and enable the **Deletion Protection** feature. The Deletion Protection attribute value should change now to **Enabled**.
08. Repeat steps no. 4-7 to enable deletion protection for other AWS load balancers provisioned in the current region.
09. Change the AWS region from the navigation bar and repeat the remediation process for other regions.

Instances

- file-server-parth
- ALB-all-websites

References

- https://www.trendmicro.com/cloudoneconformity/knowledge-base/aws/ELBv2/deletion-protection.html#

CloudFront Viewer Protocol Policy

Risk:	**Medium**	**STATUS:** Vulnerable
CVSS Score:	**5.3**	

(AV:Network/AC:Low/PR:None/UI:None/C:Low/I:None/A:None)

Description

Using HTTPS for your CloudFront CDN distribution can guarantee that the encrypted traffic between the edge (cache) servers and the application viewers cannot be decrypted by malicious users in case they are able to intercept packets sent across the CDN distribution network.

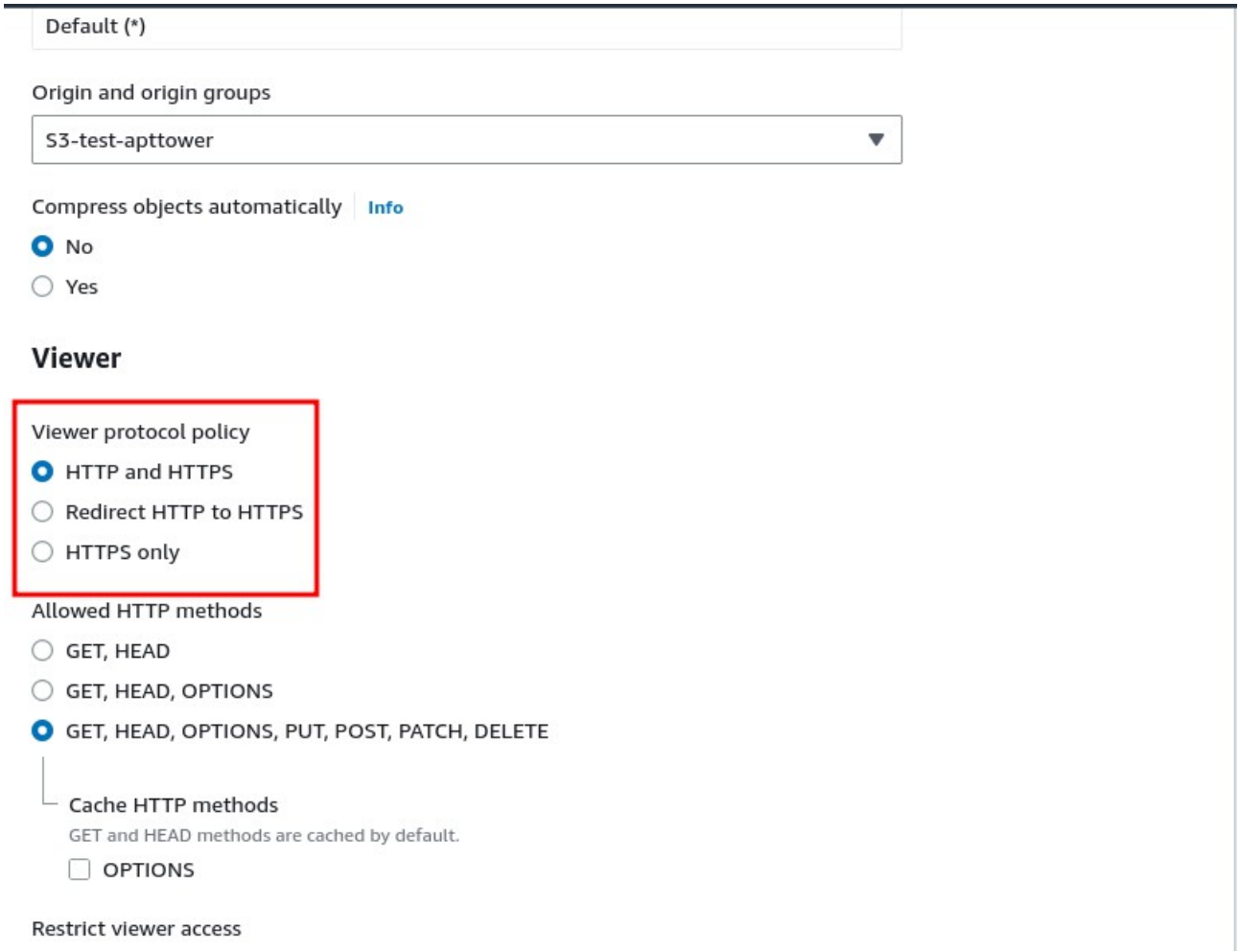
Ensure that the communication between your Amazon CloudFront CDN distribution and its viewers (end users) is encrypted using HTTPS in order to secure the delivery of your web application content. To enable data in transit encryption, you need to configure the web distribution viewer protocol policy to redirect HTTP requests to HTTPS requests or to require the viewers to use only the HTTPS protocol to access your web content available in the CloudFront distribution cache.

This rule can help you with the following compliance standards:

1. PCI
2. HIPAA
3. NIST4

Step by Step Proof

1. Sign in to the AWS Management Console.
2. Navigate to CloudFront dashboard at https://console.aws.amazon.com/cloudfront/
3. In the left navigation panel, click **Distributions**.
4. On **CloudFront Distribution** page, under the main menu, select **Web** and **Enabled** from **Viewing** dropdown menus to list all active web distributions available within your AWS account.
5. Select the web distribution that you want to reconfigure (see Audit section part I to identify the right distribution).



Remediation

To ensure that your web content is encrypted between your CloudFront distribution edge locations and your application viewers, perform the following actions:

01. Sign in to the AWS Management Console.
02. Navigate to CloudFront dashboard at https://console.aws.amazon.com/cloudfront/
03. In the left navigation panel, click **Distributions**.
04. On **CloudFront Distribution** page, under the main menu, select **Web** and **Enabled** from **Viewing** dropdown menus to list all active web distributions available within your AWS account.
05. Select the web distribution that you want to reconfigure (see Audit section part I to identify the right distribution).
06. Click the **Distribution Settings** button from the dashboard top menu to

access the resource configuration page.
07. Choose the **Behaviors** tab and select the distribution default behavior.
08. Click the **Edit** button to access the behavior configuration settings.
09. On the **Edit Behavior** page, under **Default Cache Behavior Settings**, perform one of the following actions to enforce encryption for your web content:
a. Set the **Viewer Protocol Policy** configuration attribute to **Redirect HTTP to HTTPS** so that any HTTP requests are automatically redirected to HTTPS requests. Click **Yes, Edit** to apply the changes.
b. Set the **Viewer Protocol Policy** attribute to **HTTPS Only** so that your application viewers can only access your web content using HTTPS. Choosing this option will drop any HTTP traffic between edge servers and viewers. Click **Yes, Edit** to apply the configuration changes.
10. Repeat steps no. 5-9 to reconfigure the viewer protocol policy for other Amazon CloudFront CDN distributions available within your AWS account.

Instances

- ENJT91FE6V7F
- E3CHI3YXQXW4RB
- E2F74E3WPFHIN0
- E3QDJ3E7K0RAHX
- EGCUCGS7OZV5R
- E36XYQMTWE7HYN
- E1K403FLOG4KF

References

- https://aws.amazon.com/cloudfront/features/?whats-new-cloudfront.sort-by=item.additionalFields.postDateTime&whats-new-cloudfront.sort-order=desc#faq
- https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/distribution-overview.html
- https://docs.aws.amazon.com/cli/latest/reference/cloudfront/index.html

FTP Port Open

Risk:	**Medium**	**STATUS:** Vulnerable
CVSS Score:	**5.3**	

(AV:Network/AC:Low/PR:None/UI:None/C:Low/I:None/A:None)

Description

Allowing unrestricted FTP access can increase opportunities for malicious activity such as brute-force attacks, FTP bounce attacks, spoofing attacks and packet capture.

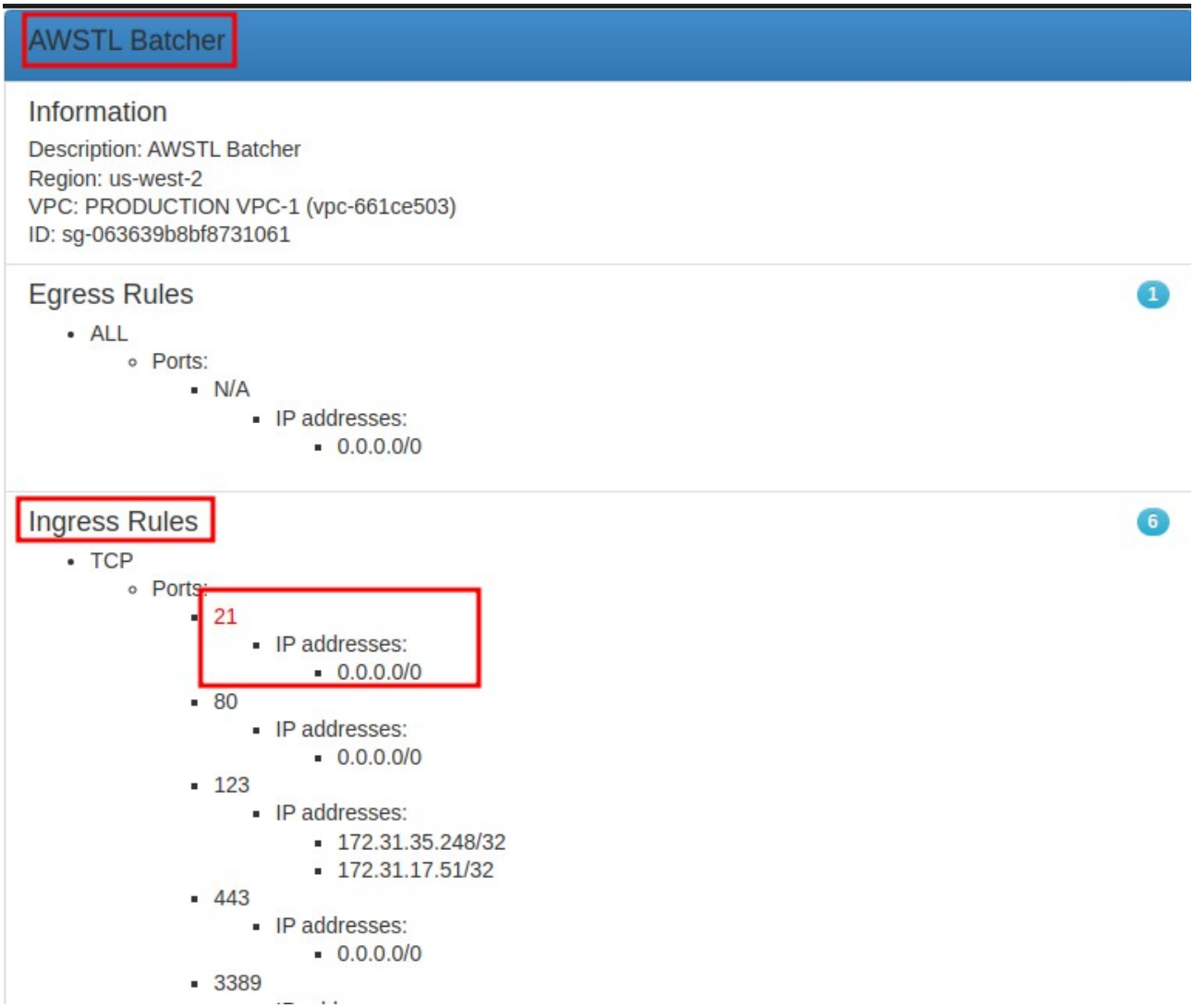
Check your EC2 security groups for inbound rules that allow unrestricted access (i.e. 0.0.0.0/0 or ::/0) to TCP ports 20 and 21 and restrict access to only those IP addresses that require it in order to implement the principle of least privilege and reduce the possibility of a breach. TCP ports 20 and 21 are used for data transfer and communication by the File Transfer Protocol (FTP) client-server applications.

This rule can help you with the following compliance standards:

1. PCI
2. APRA
3. MAS
4. NIST4

Step by Step Proof

01. Sign in to the AWS Management Console.
02. Navigate to EC2 dashboard at https://console.aws.amazon.com/ec2/home.
03. In the navigation panel, under NETWORK & SECURITY section, choose Security Groups.
04. Select the appropriate security group (see Audit section to identify the right one(s)).



Remediation

To update your security groups inbound/ingress configuration in order to restrict FTP access to specific entities (IP addresses, IP ranges, etc), perform the following:

01. Sign in to the AWS Management Console.
02. Navigate to EC2 dashboard at https://console.aws.amazon.com/ec2/home.
03. In the navigation panel, under **NETWORK & SECURITY** section, choose **Security Groups**.
04. Select the appropriate security group (see Audit section to identify the right one(s)).
05. Select the **Inbound** tab from the dashboard bottom panel and click the **Edit** button.
06. In the **Edit inbound rules** dialog box, change the traffic **Source** for any inbound rules that allow unrestricted access through TCP ports 20 and 21 by

performing one of the following actions:
a. Select **My IP** from the **Source** dropdown list to allow inbound traffic only from your machine (from your IP address).
b. Select **Custom** from the **Source** dropdown list and enter one of the following options based on your access requirements:
- The static IP/Elastic IP address of the permitted host with the suffix set to /32, e.g. 54.164.53.214/32.
- The IP address range of the permitted hosts in CIDR notation, for example 54.164.53.214/24.
- The name or ID of another security group available in the same AWS region.
07. Click **Save** to apply the changes.
08. Repeat steps no. 4 – 7 to update other EC2 security groups that allow unrestricted FTP access.
09. Change the AWS region from the navigation bar and repeat the process for other regions.

Instances

- AWSTL Batcher

References

- https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-security-groups.html
- https://docs.aws.amazon.com/vpc/latest/userguide/vpc-security-groups.html
- https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/authorizing-access-to-an-instance.html

Security Group Whitelists Too Permissive CIDRs

Risk:	**Medium**	**STATUS:** Vulnerable
CVSS Score:	**5.3**	

(AV:Network/AC:Low/PR:None/UI:None/C:Low/I:None/A:None)

Description

AWS Security Group for your EC2 instances have an unknown or too permissive CIDR origin allowed for inbound/outbound traffic. This audit rule expects another Security Group reference as originator or destiny of the traffic that pass through it.

When an unknown CIDR is found, the Unknown CIDR caption is added to the report, which facilitates detection of EC2 security group rules that whitelist network traffic from untrusted IP ranges.

Using RFC-1918 CIDRs within your Amazon EC2 security groups to allows an entire private network to access the associated EC2 instances can be overly permissive, therefore the security groups configuration does not adhere to AWS cloud security best practices.

Check your Amazon EC2 security groups for inbound rules that allow access from IP address ranges specified in RFC-1918 (i.e. 10.0.0.0/8, 172.16.0.0/12 and 192.168.0.0/16) and restrict access to only those private IP addresses/IP ranges that require it in order to implement the Principle of Least Privilege (POLP).

This rule can help you with the following compliance standards:

1. NIST4

Step by Step Proof

1. Sign in to the AWS Management Console.
2. Navigate to Amazon EC2 console at https://console.aws.amazon.com/ec2/home
3. In the navigation panel, under **Network & Security**, choose **Security Groups**.
4. Select the Amazon EC2 security group that you want to reconfigure.

Remediation

To update the inbound access configuration for the Amazon EC2 security groups with RFC-1918 CIDRs in order to restrict access to trusted entities only (i.e. authorized IP addresses or other security groups), perform the following operations:

01. Sign in to the AWS Management Console.
02. Navigate to Amazon EC2 console at https://console.aws.amazon.com/ec2/home
03. In the navigation panel, under **Network & Security**, choose **Security Groups**.
04. Select the Amazon EC2 security group that you want to reconfigure.
05. Select the **Inbound rules** tab from the console bottom panel and choose **Edit inbound rules**.
06. On the **Edit inbound rules** configuration page, change the traffic source for the inbound rule that allows access from RFC-1918 CIDRs

(regardless of the port used), by performing one of the following actions:
a. Select **Custom** from the **Source** dropdown list and enter one of the following options based on your access requirements:
- A specific IPv4 address with the suffix set to /32 (e.g. 192.168.0.5/32), representing the private IP address of the trusted host that requires access to the Amazon EC2 instance(s) associated with the selected security group.
- The name or ID of another security group available in the same AWS cloud region.
b. Choose **Save rules** to apply the configuration changes.
07. Repeat steps no. 4 – 6 to reconfigure other Amazon EC2 security groups that allow inbound traffic from RFC-1918 CIDRs.
08. Change the AWS cloud region from the console navigation bar and repeat the remediation process for other regions.

Instances

- Voice Freepbx-16 Indikitch
- CiscoCRV1000vMiraj
- TT – awsapttowersql
- freepbx-indikitch
- twilio
- default
- Cisco Cloud Services Router (CSR) 1000V - Security Technology Package-03.16.04.aS-AutogenByAWSMP-

References

1. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-security-groups.html
2. https://docs.aws.amazon.com/vpc/latest/userguide/vpc-security-groups.html
3. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ec2/index.html
4. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ec2/revoke-security-group-ingress.html
5. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ec2/describe-security-groups.html

Unencrypted ECR Repositories

Risk:	**Medium**	**STATUS:** Vulnerable
CVSS Score:	**5.3**	

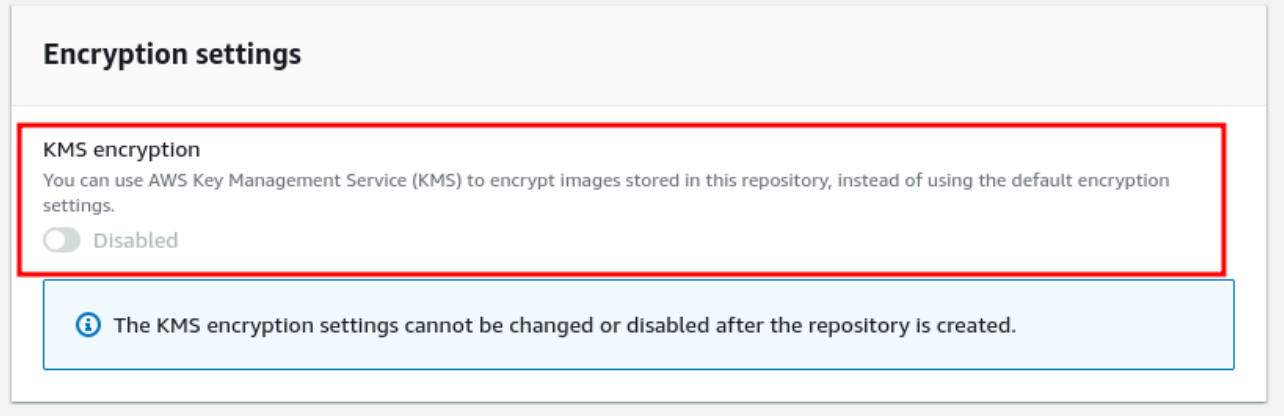
(AV:Network/AC:Low/PR:None/UI:None/C:Low/I:None/A:None)

Description

Encrypting your ECR repositories helps protect your data from unauthorized access or tampering. That way, you can ensure that only authorized users can access and modify the contents of your repositories. Such action can help protect against external threats such as hackers or malware, as well as internal threats such as accidental or unauthorized access.

Step by Step Proof

1. When creating a repository, Amazon ECR sends a DescribeKey call to AWS KMS to validate and retrieve the Amazon Resource Name (ARN) of the KMS key specified in the encryption configuration.
2. Amazon ECR sends two CreateGrant requests to AWS KMS to create grants on the KMS key to allow Amazon ECR to encrypt and decrypt data using the data key.
3. When pushing an image, a GenerateDataKey request is made to AWS KMS that specifies the KMS key to use for encrypting the image layer and manifest.



Remediation

Use AWS Key Management Service (KMS) to encrypt images. KMS is a managed service that makes it easy to create and manage encryption keys.

When we encrypt images,it is possible to specify who has access to the encryption keys. This enables to control who can decrypt and view images. Encrypting images is a great way to protect your data if your images are compromised.

The following provides a high-level understanding of how Amazon ECR is integrated with AWS KMS to encrypt and decrypt your repositories:

01. When creating a repository, Amazon ECR sends a DescribeKey call to AWS KMS to validate and retrieve the Amazon Resource Name (ARN) of the KMS key specified in the encryption configuration.
02. Amazon ECR sends two CreateGrant requests to AWS KMS to create grants on the KMS key to allow Amazon ECR to encrypt and decrypt data using the data key.
03. When pushing an image, a GenerateDataKey request is made to AWS KMS that specifies the KMS key to use for encrypting the image layer and manifest.
04. AWS KMS generates a new data key, encrypts it under the specified KMS key, and sends the encrypted data key to be stored with the image layer metadata and the image manifest.
05. When pulling an image, a Decrypt request is made to AWS KMS, specifying the encrypted data key.
06. AWS KMS decrypts the encrypted data key and sends the decrypted data key to Amazon S3.
07. The data key is used to decrypt the image layer before the image layer being pulled.
08. When a repository is deleted, Amazon ECR sends two RetireGrant requests to AWS KMS to retire the grants created for the repository.
Check reference link for further documentation as per your environment

Instances

- deep
- hello-world
- next-cloud
- sndk-web
- tet

References

- https://docs.aws.amazon.com/AmazonECR/latest/userguide/encryption-

at-rest.html

ECR Scan-On-Push Disabled

Risk:	**Medium**	**STATUS:** Vulnerable
CVSS Score:	**5.3**	

(AV:Network/AC:Low/PR:None/UI:None/C:Low/I:None/A:None)

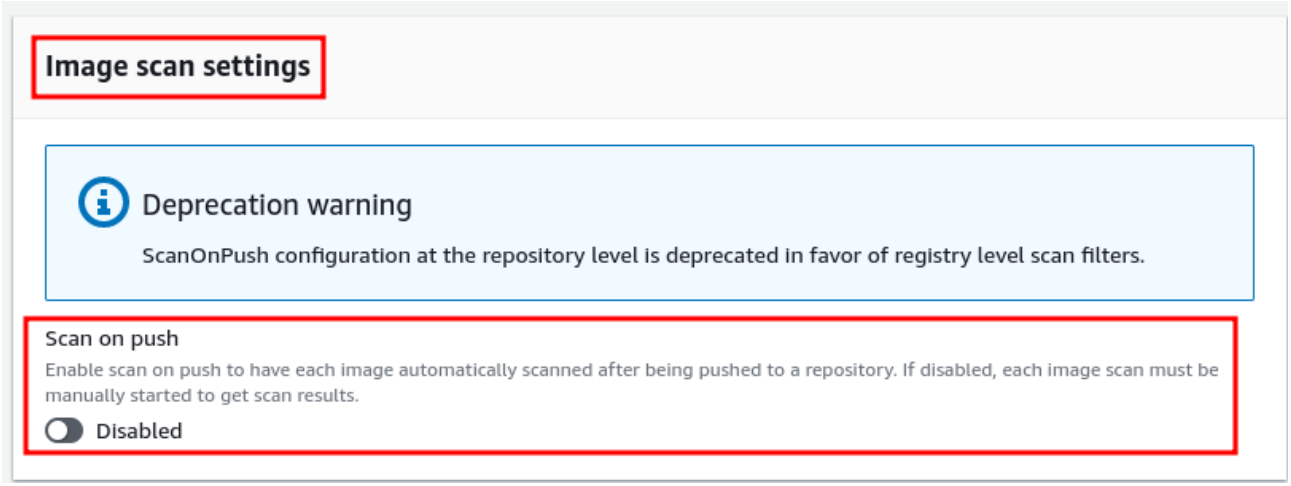
Description

For the security and compliance status of your applications it is crucial to detect and respond to Amazon ECR container image vulnerabilities in the early stages of deployment. When Scan on Push security feature is enabled, your container images are automatically scanned after being pushed to your Amazon ECR repository. If Scan on Push is disabled on your repository, then each image scan must be manually started to get scan results.

Ensure that all your Amazon ECR container images are automatically scanned for security vulnerabilities and expenses after being pushed to a repository. Scan on Push for Amazon ECR is an automated vulnerability assessment feature that helps you improve the security of your ECR container images by scanning them for a broad range of Operating System (OS) vulnerabilities after being pushed to an ECR repository.

Step by Step Proof

1. Sign in to AWS Management Console.
2. Navigate to Amazon ECR console at https://console.aws.amazon.com/ecr
3. In the left navigation panel, under Amazon ECR, select Repositories to access your ECR image repositories.



Remediation

Amazon ECR Scan on Push helps you identify software vulnerabilities within your container images by checking each image against an aggregated set of Common Vulnerabilities and Exposures (CVEs). To configure each Amazon ECR repository to automatically scan your container images for security vulnerabilities when you push them to the repository, perform the following operations:

01. Sign in to AWS Management Console.
02. Navigate to Amazon ECR console at https://console.aws.amazon.com/ecr
03. In the left navigation panel, under **Amazon ECR**, select **Repositories** to access your ECR image repositories.
04. Select the container image repository that you want to reconfigure, and choose **Edit**.
05. On the **Edit repository** configuration page, toggle **Enabled** under **Scan on push** to enable Scan on Push security feature and have each container image automatically scanned after being pushed to the selected repository. This will apply to future image pushes. Choose Save to apply the configuration changes.
06. Repeat step no. 4 and 5 to enable Scan on Push feature for other Amazon ECR image repositories available within the current AWS cloud region.
07. Change the AWS region from the navigation bar to repeat the remediation process for other regions.

Instances

- deep
- hello-world
- next-cloud
- sndk-web
- tet

References

- https://aws.amazon.com/ecr/faqs/
- https://docs.aws.amazon.com/AmazonECR/latest/userguide/Repositories.html
- https://docs.aws.amazon.com/AmazonECR/latest/userguide/image-scanning.html

IAM Cross-Account Access Lacks External ID and MFA

Risk:	**Medium**	**STATUS:** Vulnerable
CVSS Score:	**5.3**	

(AV:Network/AC:Low/PR:None/UI:None/C:Low/I:None/A:None)

Description

Increase the security of your cross-account IAM role by requiring either an optional external ID (similar to a password) or an MFA device to secure further the access to your AWS cloud resources and prevent "confused deputy" attacks. This is highly recommended if you don't own or have administrative access to the AWS account that can assume this IAM role. To assume this cross-account role, users must be available in the trusted account and provide the external ID or the unique passcode generated by the MFA device configured.

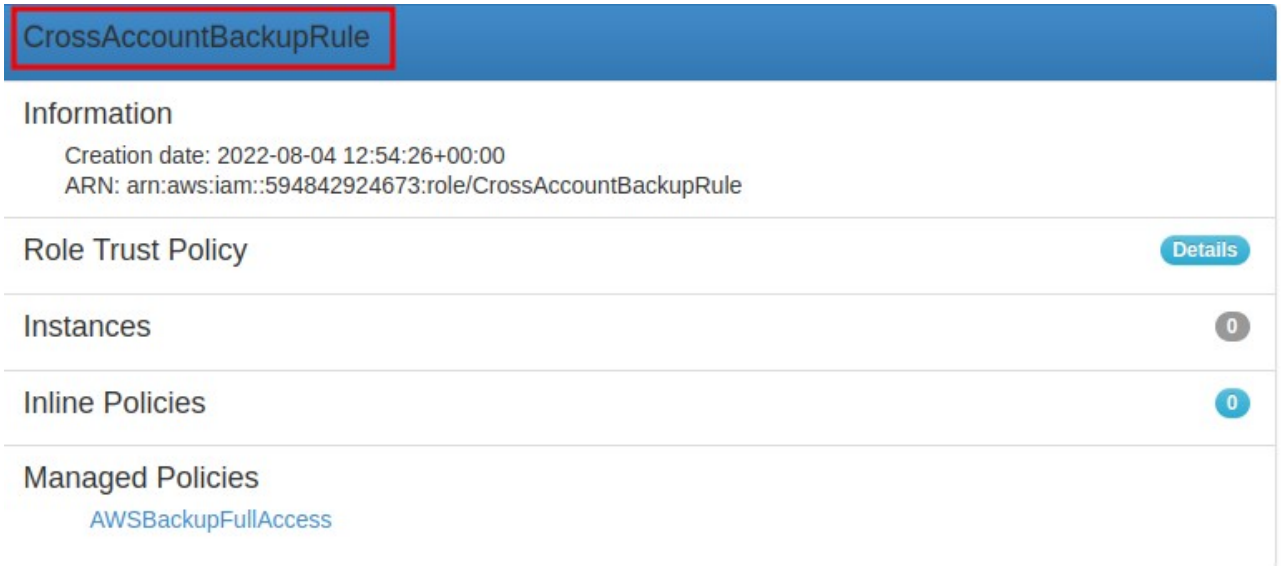
Ensure that Amazon IAM roles used to establish a trusted relationship between your AWS cloud account and a third-party entity (also known as cross-account access roles) are using Multi-Factor Authentication (MFA) or external IDs to secure the access to your resources and to prevent "confused deputy" attacks. The MFA/external ID adds an extra layer of security on top of role's temporary security credentials and facilitates external third-party accounts to access your AWS resources in a secure way.

This rule can help you with the following compliance standards:

1. APRA
2. MAS
3. NIST4

Step by Step Proof

1. Sign in to the AWS Management Console.
2. Navigate to Amazon IAM console at https://console.aws.amazon.com/iam/
3. In the navigation panel, under **Access management**, choose **Roles**.
4. Click on the name of the cross-account IAM role that you want to reconfigure.



Remediation

To update the trust relationship policies defined for your Amazon IAM cross-account roles in order to enable Multi-Factor Authentication (MFA) and/or external ID support for secure access, perform the following operations:

01. Sign in to the AWS Management Console.
02. Navigate to Amazon IAM console at https://console.aws.amazon.com/iam/
03. In the navigation panel, under **Access management**, choose **Roles**.
04. Click on the name of the cross-account IAM role that you want to reconfigure.
05. Select the **Trust relationships** tab and choose **Edit trust relationship**.
06. On the Edit Trust Relationship page, add one of the following blocks to the existing policy:a. To enable Multi-Factor Authentication (MFA) and force the IAM users in the trusted account(s) to provide the passcode generated by the MFA device upon accessing your AWS cloud resources, add the following Condition element block:
`"Condition": { "Bool": { "aws:MultiFactorAuthPresent": "true" }` to the trust relationship policy. Once updated, the policy document configured for the selected IAM role should look like this:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": "sts:AssumeRole",
            "Effect": "Allow",
            "Condition": {
                "Bool": {
                    "aws:MultiFactorAuthPresent": "true"
                }
```

```
            },
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:root"
            }
        }
    ]
}
```

B. To enable external ID support in order to force the users within the trusted AWS accounts to provide the required ID (passphrase) upon accessing your AWS cloud resources, add the following **"Condition"** element block: **"Condition": { "StringEquals": { "sts:ExternalId": "<external_id>" }** to the trust relationship policy, then replace **<external_id>** with your own passphrase. Once updated, the policy document configured for the selected IAM role, should look like this:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": "sts:AssumeRole",
            "Effect": "Allow",
            "Condition": {
                "StringEquals": {
                    "sts:ExternalId": "<external_id>"
                }
            },
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:root"
            }
        }
    ]
}
```

C. (Optional) To enable both MFA protection and external ID support for the selected cross-account IAM role, add the following "Condition" element block: "Condition": { "Bool": { "aws:MultiFactorAuthPresent": "true" }, "StringEquals": { "sts:ExternalId": "<external_id>" } }, to the trust relationship policy. Once updated, the policy document defined for the selected role should look like this:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": "sts:AssumeRole",
            "Effect": "Allow",
            "Condition": {
                "Bool": {
                    "aws:MultiFactorAuthPresent": "true"
                },
                "StringEquals": {
                    "sts:ExternalId": "<external_id>"
                }
            },
```

```
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:root"
            }
        }
    ]
}
```

D. Choose **Update Trust Policy** to apply the configuration changes.
07. Repeat steps no. 4 – 6 to enable MFA and/or external ID support for other cross-account IAM roles, available within your AWS cloud account.

Instances

- CrossAccountBackupRule
- gbabu_cross_acc_role

References

- https://aws.amazon.com/blogs/security/how-to-use-external-id-when-granting-access-to-your-aws-resources/
- https://aws.amazon.com/iam/faqs/
- https://docs.aws.amazon.com/IAM/latest/UserGuide/id.html
- https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles.html

Inline Role Policy Allows iam:passrole

Risk:	**Medium**	**STATUS:** Vulnerable
CVSS Score:	**5.3**	

(AV:Network/AC:Low/PR:None/UI:None/C:Low/I:None/A:None)

Description

Providing the right permissions for your Amazon IAM roles will significantly reduce the risk of unauthorized access to your AWS cloud services and resources.

Ensure that the policies attached to your Amazon IAM roles are not too permissive. To adhere to IAM security best practices, the policies configured for your IAM roles should implement the Principle of Least Privilege (also known as the principle of least authority, i.e. the security concept of providing every identity, process, or system the minimal set of permissions required to successfully perform its tasks).

This rule can help you with the following compliance standards:

1. APRA
2. MAS
3. NIST4

Step by Step Proof

01. Sign in to the AWS Management Console.
02. Navigate to Amazon IAM console at https://console.aws.amazon.com/iam/
03. In the navigation panel, under Access management, choose Roles.
04. Click on the name of the Amazon IAM role that you want to reconfigure.

Remediation

To update your Amazon IAM role permissions through IAM policies in order to implement the Principle of Least Privilege (POLP), perform the following operations:

01. Sign in to the AWS Management Console.
02. Navigate to Amazon IAM console at https://console.aws.amazon.com/iam/
03. In the navigation panel, under **Access management**, choose **Roles**.
04. Click on the name of the Amazon IAM role that you want to reconfigure.
05. Select the **Permissions** tab to access the identity-based policies attached to the selected role.
06. In the **Permissions policies** section, perform the following actions based on the policy type:
a. Choose the overly permissive inline policy embedded within the selected IAM role, click on the Expand button (right arrow icon), and select **Edit policy**.
b. Select the **JSON** tab and customize the policy document according to your IAM role access requirements. Follow the Principle of Least Privilege (the security concept of providing every identity the minimal set of permissions required to perform successfully its tasks) when editing the inline policy associated with to your IAM role. For example, replace the **"Action"** element value **"*"** with specific Amazon EC2 service actions such as **"ec2:DescribeInstances"** and **"ec2:DescribeImages"** if you want your IAM role to grant permission to describe one or more EC2 instances and AMIs. Or pass a specific and compliant IAM role to AWS cloud services when **"Action"** is set to **"iam:PassRole"**.
c. Choose **Review policy** to review the inline policy before you save your changes.
d. Choose **Save changes** to apply the permission changes.

07. Repeat steps no. 4-6 for each Amazon IAM role that you want to reconfigure, available in your AWS cloud account.

Instances

- aws-opsworks-service-role

References

- https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html
- https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-ec2.html
- https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_manage_modify.html
- https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_policy-validator.html

Bucket Allowing Cleat Text (HTTP) Communication

Risk:	**Medium**	**STATUS:** Vulnerable
CVSS Score:	**5.3**	

(AV:Network/AC:Low/PR:None/UI:None/C:Low/I:None/A:None)

Description

When Amazon S3 buckets are not configured to strictly require SSL connections, the communication between the buckets and their clients (users and applications) is vulnerable to eavesdropping and Man-in-the-Middle (MITM) attacks. It is strongly recommended to enforce SSL-only access by denying all regular, unencrypted HTTP requests to your Amazon S3 buckets when dealing with business-critical, sensitive, or private data.

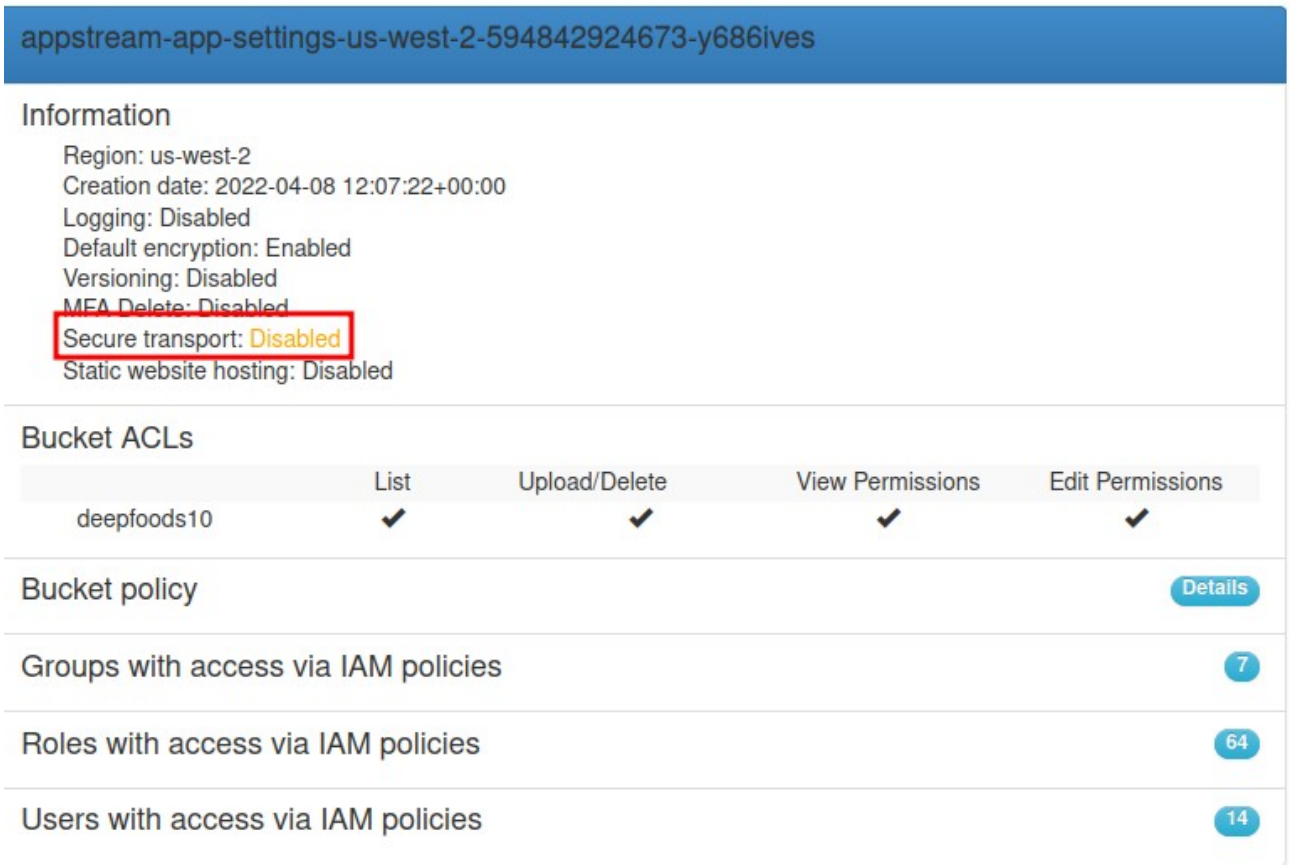
Ensure that your Amazon S3 buckets enforce encryption of data over the network, as it travels to and from Amazon S3, using Secure Sockets Layer (SSL).

This rule can help you with the following compliance standards:

1. PCI
2. APRA
3. MAS
4. NIST4

Step by Step Proof

1. Sign in to the AWS Management Console.
2. Navigate to Amazon S3 console at https://console.aws.amazon.com/s3/
3. Click on the name of the S3 bucket that you want to reconfigure (see Audit section part I to identify the right resource).
4. Select the Permissions tab from the console menu to access the bucket permissions.



Remediation

To enforce in-transit encryption for your Amazon S3 buckets via bucket policies, perform the following actions:
01. Sign in to the AWS Management Console.
02. Navigate to Amazon S3 console at https://console.aws.amazon.com/s3/
03. Click on the name of the S3 bucket that you want to reconfigure (see Audit section part I to identify the right resource).
04. Select the **Permissions** tab from the console menu to access the bucket permissions.
05. In the **Bucket policy** section, choose **Edit** to modify the bucket policy attached to the selected bucket.
06. In the **Policy** editor box, perform one of the following actions based on the current access configuration:
a. If there is no policy attached to the selected S3 bucket, paste the following policy document in the **Policy** editor box, then choose **Save changes** to apply the changes. This bucket policy will deny all non-encrypted (non-SSL) access to an S3 bucket named "cc-web-app-assets":

```
{
  "Version": "2012-10-17",
  "Id": "cc-secure-transport-bucket-policy",
  "Statement": [
    {
      "Effect": "Deny",
      "Principal": { "AWS": "*" },
      "Action": "s3:*",
      "Condition": {
        "Bool": { "aws:SecureTransport": false }
      },
      "Resource":"arn:aws:s3:::cc-web-app-assets/*"
    }
  ]
}
```

B. If the selected S3 bucket has a bucket policy attached, append the following policy statement (highlighted) to the existing policy document in the Policy editor box, as shown in the following example, then choose Save changes to apply the changes. This bucket policy will enable Amazon S3 to serve bucket content over SSL only and deny all regular (unencrypted) access:

```
{
  "Id": "cc-web-assets-bucket-policy",
  "Version": "2012-10-17",
  "Statement": [

    ...

    {
      "Sid": "cc-secure-transport-bucket-policy",
      "Effect": "Deny",
      "Principal": { "AWS": "*" },
      "Action": "s3:*",
      "Condition": {
        "Bool": { "aws:SecureTransport": false }
      },
      "Resource":"arn:aws:s3:::cc-web-app-assets/*"
    }
  ]
}
```

07. Repeat steps no. 3 – 6 to enable in-transit encryption for other Amazon S3 buckets available in your AWS cloud account.

Instances

- appstream-app-settings-us-west-2-594842924673-y686ives
- iiotreport
- apttower
- ttchat
- tasktower
- ttmail-beanstalk
- teamlocusdbbackup
- compute-optimizer-deep
- pandoarch
- indikitch.com
- ttmail-beanstalk-backup
- devendra-img
- accordionwagebump
- cny-nas-drive-81.30
- datasync-restore-pandoarchdb
- cost-and-usage-report-deep
- betameet.teamlocus.com
- deepdocsses
- gbabu-img
- teamlocus-elastic-backup-all
- cf-templates-82lne1hxbwuv-us-west-2
- audiofilestotext
- ttmail-beanstalk-mail-pending
- pandoarch-empschedule
- appstream2-36fb080bb8-us-west-2-594842924673
- testdeepkiran
- chatbot-cf-new
- ttdrive
- elasticbeanstalk-us-west-2-594842924673
- awsaptdb02bucket
- allwebsites
- pandoarchdbbkp
- athena-result-deepkiran
- cloudtrail-deep
- datasync-restore-teamlocus
- pandodb-deleted-bkp
- meet.teamlocus.com
- tlchatelasticsearchbackup
- www.indikitch.com
- call.teamlocus.com
- deepfoods-cloudformation-template
- datasync-restore-apt

References

- https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_elements.html
- https://docs.aws.amazon.com/AmazonS3/latest/userguide/access-policy-language-overview.html
- https://docs.aws.amazon.com/AmazonS3/latest/userguide/using-iam-policies.html

Redshift SSL Not Enabled

Risk:	**Medium**	**STATUS:** Vulnerable
CVSS Score:	**5.3**	

(AV:Network/AC:Low/PR:None/UI:None/C:Low/I:None/A:None)

Description

When Redshift clusters are not configured to require Secure Sockets Layer (SSL) connections, the communication between the clients and these clusters is vulnerable to eavesdropping and man-in-the-middle (MITM) attacks. Cloud Conformity strongly recommends enabling SSL for your clusters front-end connection when dealing with sensitive or private data.

This policy identifies Redshift databases in which data connection to and from is occurring on an insecure channel. SSL connections ensures the security of the data in transit.

Ensure that all the parameter groups associated with your Amazon Redshift clusters have the require_ssl parameter enabled in order to keep your data secure in transit by encrypting the connection between the clients (applications) and your warehouse clusters.

This rule can help you with the following compliance standards:

1. PCI
2. HIPAA
3. APRA
4. MAS
5. NIST4

Step by Step Proof

1. Login to the AWS and navigate to the Amazon Redshift service.
2. Expand the identified Redshift cluster and make a note of the Cluster Parameter Group
3. In the navigation panel, click on the Parameter group.

Remediation

To enable require_ssl parameter within your Amazon Redshift non-default parameter groups in order to use SSL for the client-cluster connection, perform the following:

01. Login to the AWS and navigate to the Amazon Redshift service.
02. Expand the identified Redshift cluster and make a note of the Cluster Parameter Group
03. In the navigation panel, click on the Parameter group.
04. Select the identified Parameter Group and click on Edit Parameters.
05. Review the require_ssl flag. Update the parameter require_ssl to **true** and save it.
Note: If the current parameter group is a Default parameter group, it cannot be edited. You will need to create a new parameter group and point it to an affected cluster.
06. To take effect immediately, the Amazon Redshift cluster associated with the selected parameter group must be rebooted. To reboot a cluster, perform the following actions:

a. In the navigation panel, under **Redshift Dashboard**, click **Clusters**.
b. Choose the cluster that you want to reboot then click on its identifier link available in the **Cluster** column.
c. On the configuration page, click the **Cluster** dropdown button from the dashboard top menu and select **Reboot**.

d. Within **Reboot Cluster** dialog box, click **Continue** to reboot the selected AWS Redshift cluster. The cluster status should change now to **rebooting**. IMPORTANT: The reboot process can take several minutes. During this time your Redshift cluster becomes unavailable.

07. Repeat steps no. 2-6 to enable the require_ssl parameter for other non-default parameter groups created in the current region.
08. Change the AWS region from the navigation bar and repeat the entire process for other regions.

Instances

- cf1eb88c6b50532f9176f3a3a4eb587159ddebc1

References

- https://docs.aws.amazon.com/redshift/latest/mgmt/managing-parameter-groups-console.html
- https://docs.aws.amazon.com/redshift/latest/mgmt/connecting-ssl-support.html
- https://docs.aws.amazon.com/redshift/latest/mgmt/working-with-clusters.html

S3 Bucket without Delete MFA

Risk:	**Medium**	**STATUS:** Vulnerable
CVSS Score:	**5.3**	

(AV:Network/AC:Low/PR:None/UI:None/C:Low/I:None/A:None)

Description

Using MFA-protected Amazon S3 buckets will add an extra layer of protection on top of existing ones to ensure that your S3 objects can't be accidentally or intentionally deleted by other users that have access to your S3 buckets.

Note 1: The MFA Delete feature requires bucket versioning as dependency. Bucket versioning is a method of keeping multiple variations of an S3 object in the same bucket.

Note 2: Only the bucket owner that is logged in as AWS root account can enable MFA Delete feature and perform DELETE actions on Amazon S3 buckets.

Ensure that your Amazon S3 buckets are configured to use the Multi-Factor Authentication (MFA) Delete feature in order to prevent the deletion of versioned S3 objects available within your buckets.

This rule can help you with the following compliance standards:

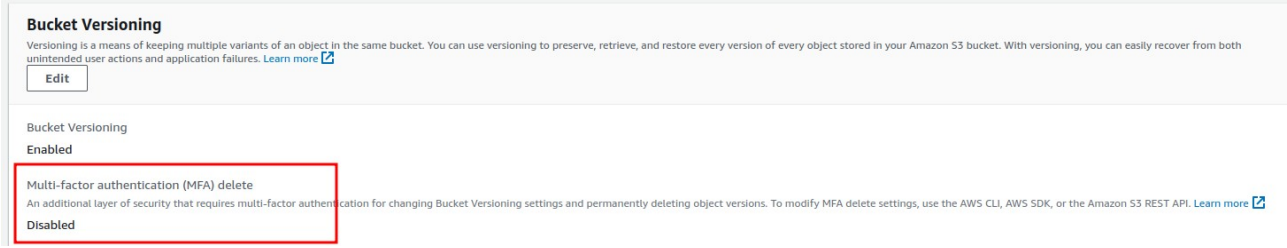
1. PCI
2. GDPR
3. APRA
4. MAS
5. NIST4

During the review, it was discovered that MFA is disabled for operation for S3 Bucket.

Step by Step Proof

1. Sign in to the AWS Management Console.
2. Navigate to Amazon S3 console at https://console.aws.amazon.com/s3/
3. Click on the name of the S3 bucket that you want to reconfigure (see Audit section part I to identify the right resource).
4. Select the Permissions tab from the console menu to access the bucket

permissions.



Remediation

Terraform configuration file (.tf):

```
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 3.27"
    }
  }

  required_version = ">= 0.14.9"
}

provider "aws" {
  profile = "default"
  region  = "us-east-1"
}

resource "aws_s3_bucket" "mfa-protected" {
  bucket = "cc-prod-web-data"
  versioning {
    enabled = true
    mfa_delete = true
  }
}
```

Instances

- apttower
- ttchat
- tasktower
- gbabu-img
- teamlocus-elastic-backup-all
- allwebsites
- tlchatelasticsearchbackup
- deepfoods

References

- https://docs.aws.amazon.com/AmazonS3/latest/userguide/MultiFactorAuthenticationDelete.html

Lack Of MFA (root account)

Risk:	**Medium**	**STATUS:** Vulnerable
CVSS Score:	**4.8**	

(AV:Network/AC:High/PR:None/UI:None/C:Low/I:Low/A:None)

Description

Having an MFA-protected root account is one of the best ways to protect your AWS cloud resources against attackers. An MFA device signature adds an extra layer of protection on top of your existing root credentials making your AWS root account virtually impossible to penetrate without the unique passcode generated by the MFA device.

Ensure that the Multi-Factor Authentication (MFA) feature is enabled for your AWS root account in order to secure your cloud environment and adhere to IAM security best practices.

This rule can help you with the following compliance standards:

1. CISAWSF
2. PCI
3. GDPR
4. APRA
5. MAS
6. NIST4

Step by Step Proof

1. Sign in to the AWS Management Console using the root account credentials.
2. Click on the AWS account name/number available in the upper-right corner of the Management Console and select My Security Credentials from the dropdown menu.
3. On Your Security Credentials page, click on the Multi-factor authentication (MFA) tab to expand the panel with the MFA configuration settings available for the root account.

Remediation

01. Sign in to the AWS Management Console using the root account credentials.
02. Click on the AWS account name/number available in the upper-right corner of the Management Console and select **My Security Credentials** from the dropdown menu.
03. On **Your Security Credentials** page, click on the **Multi-factor authentication (MFA)** tab to expand the panel with the MFA configuration settings available for the root account.
04. On the **Multi-factor authentication (MFA)** panel, check for any MFA devices enabled for the AWS root account. If there are no MFA devices configured and the Amazon IAM console shows the **Activate MFA** button, your AWS root account is not MFA-protected and the authentication process for the root user is not following Amazon IAM security best practices.
05. Repeat steps no. 1 – 4 for each AWS root account that you want to examine.

Instances

- AWS root account

References

- https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_mfa_enable_virtual.html
- https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_mfa_enable.html
- https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_mfa.html
- https://pages.awscloud.com/how-to-enable-multi-factor-authentication-for-aws-account.html

MsSQL Ports Open to All

Risk:	**Medium**	**STATUS:** Vulnerable
CVSS Score:	**4.3**	

(AV:Network/AC:Low/PR:Low/UI:None/C:Low/I:None/A:None)

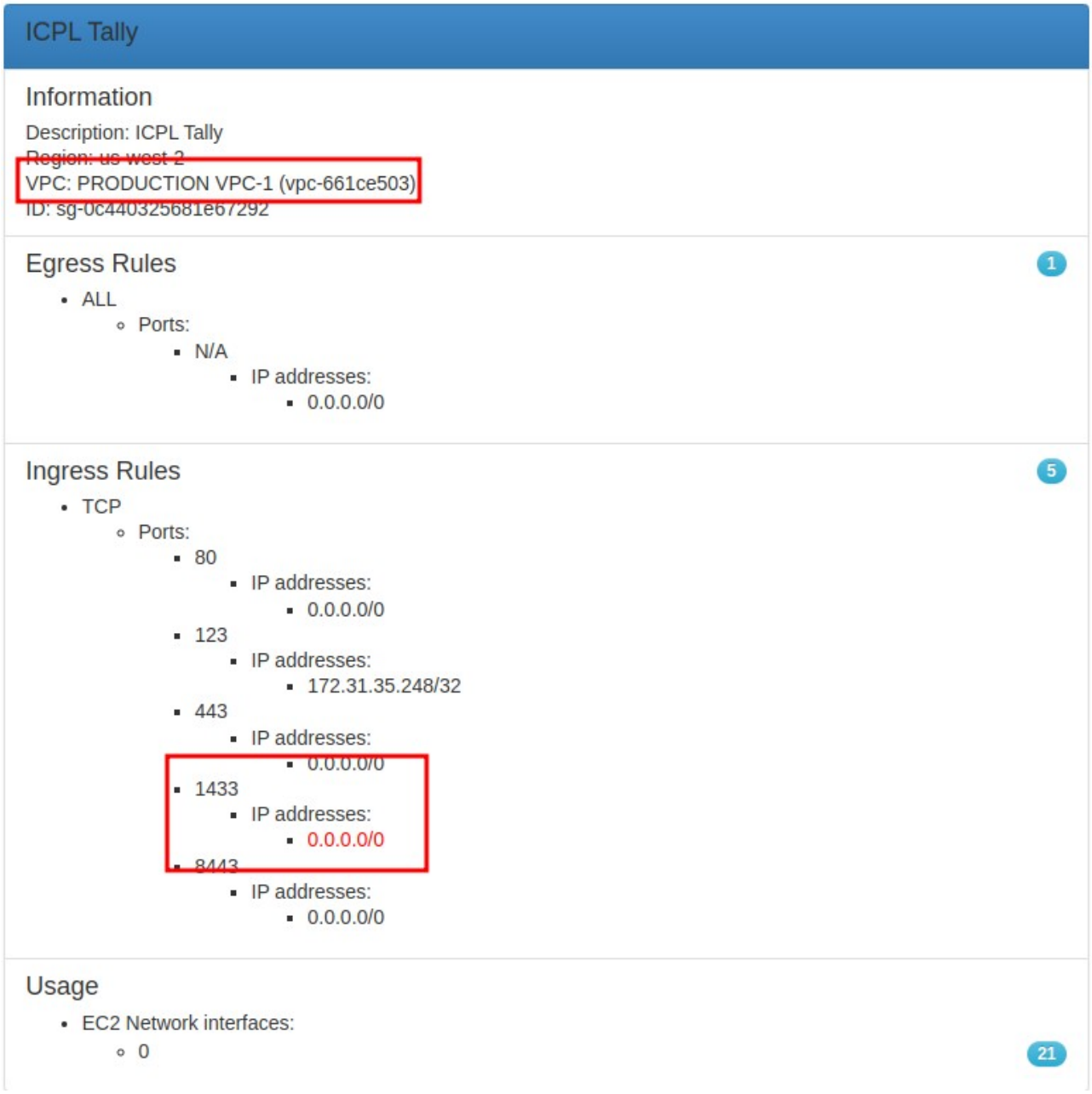
Description

Check your EC2 security groups for inbound rules that allow unrestricted access (i.e. 0.0.0.0/0 or ::/0) to TCP port 1433 and restrict access to only those IP addresses that require it in order to implement the principle of least privilege and reduce the possibility of a breach. TCP port 1433 is used by the MsSQL Server which is an open-source relational database management system (RDBMS) server.

It was observed that the configurations allow unrestricted MsSQL access can increase opportunities for malicious activity such as hacking, denial-of-service (DoS) attacks and loss of data.

Step by Step Proof

1. Sign in to the AWS Management Console.
2. Navigate to EC2 dashboard at https://console.aws.amazon.com/ec2/.
3. In the navigation panel, under NETWORK & SECURITY section, choose Security Groups.
4. Select the appropriate security group (see Audit section to identify the right one(s)).



Remediation

1. Sign in to the AWS Management Console.
2. Navigate to EC2 dashboard at https://console.aws.amazon.com/ec2/.
3. In the navigation panel, under **NETWORK & SECURITY** section, choose Security Groups.
4. Select the appropriate security group (see Audit section to identify the right one(s)).
5. Select the **Inbound** tab from the dashboard bottom panel and click the **Edit** button.

6. In the **Edit inbound rules** dialog box, change the traffic **Source** for any inbound rules that allow unrestricted access through TCP port 1433 by performing one of the following actions:
a. Select **My IP** from the **Source** dropdown list to allow inbound traffic only from your machine (from your IP address).
b. Select **Custom** from the **Source** dropdown list and enter one of the following options based on your access requirements:

i. The static IP/Elastic IP address of the permitted host with the suffix set to /32, e.g. 54.164.53.205/32.

ii. The static IP/Elastic IP address of the permitted host with the suffix set to /32, e.g. 54.164.53.205/32.

iii. The IP address range of the permitted hosts in CIDR notation, for example 54.164.53.205/24.

c. The name or ID of another security group available in the same AWS region.
7. Click **Save** to apply the changes.
8. Repeat steps no. 4-7 to update other EC2 security groups that allow unrestricted MsSQL access.
9. Change the AWS region from the navigation bar and repeat the process for other regions.

Instances

- ICPL Tally

References

- https://www.intelligentdiscovery.io/controls/ec2/aws-ec2-mssql-open

User Without MFA

Risk:	**Low**	**STATUS:** Vulnerable
CVSS Score:	**3.7**	

(AV:Network/AC:High/PR:None/UI:None/C:Low/I:None/A:None)

Description

Having MFA-protected IAM users is one of the best ways to protect your AWS services and resources against hacking. An MFA device signature adds an extra layer of protection on top of your existing IAM user credentials (username and password), making your AWS account virtually impossible to penetrate without the MFA-generated passcode.

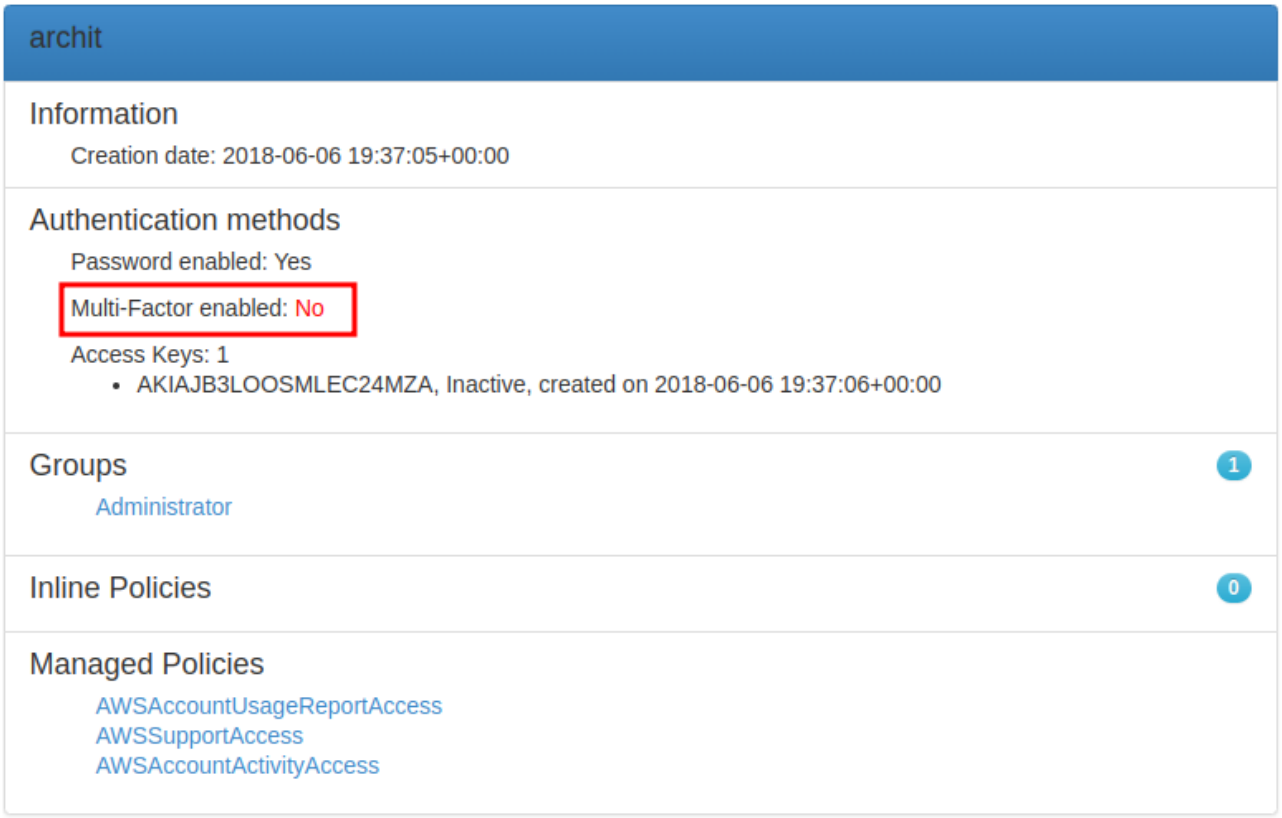
Ensure that Multi-Factor Authentication (MFA) is enabled for all the IAM users console access within your AWS account in order to secure your AWS cloud environment and adhere to IAM security best practices.

This rule can help you with the following compliance standards:

1. CISAWSF
2. PCI
3. GDPR
4. APRA
5. MAS
6. NIST4

Step by Step Proof

01. Sign in to the AWS Management Console.
02. Navigate to Amazon IAM console at https://console.aws.amazon.com/iam/.
03. In the navigation panel, under Access management, choose Users.
04. Click on the name of the Amazon IAM user that you want to reconfigure.
05. Select the Security credentials tab to access the configuration information available for the IAM user credentials.



Remediation

To enable Multi-Factor Authentication (MFA) protection for your Amazon IAM users, perform the following operations:
01. Sign in to the AWS Management Console.
02. Navigate to Amazon IAM console at https://console.aws.amazon.com/iam/.
03. In the navigation panel, under **Access management**, choose **Users**.
04. Click on the name of the Amazon IAM user that you want to reconfigure.
05. Select the **Security credentials** tab to access the configuration information available for the IAM user credentials.
06. In the **Sign-in credentials** section, click on the **Manage** link available next to **Assigned MFA device** to initiate the MFA device setup process.
07. Inside the **Sign-In Credentials** section, click the **Manage MFA Device** button next to **Multi-Factor Authentication Device** to initiate the MFA device setup process.
08. On the **Multi-factor authentication (MFA)** panel choose **Activate MFA** to initiate the MFA setup.
09. In the **Manage MFA device** configuration box, select **Virtual MFA device** from **Choose the type of MFA device to assign**, then click **Continue**.
10. Install the MFA-compatible device. The MFA virtual device used in this example is Google Authenticator. This guide assumes that you have already installed the Google Authenticator application on your smartphone, otherwise follow the official Google documentation to install the required application.

11. In the **Set up virtual MFA device** configuration box, perform the following actions:
    A. Click on the **Show QR code** link under **Use your virtual MFA app and your device's camera to scan the QR code**.
    B. Scan the QR code using the Google Authenticator application. Enter two consecutive authentication passcodes in the **MFA code 1** and **MFA code 2** text fields.
    C. Choose **Assign MFA** to complete the Multi-Factor Authentication (MFA) setup process. If successful, the following message will be displayed: "**You have successfully assigned virtual MFA**". Choose **Close** to return to the Amazon IAM console. The new virtual MFA device will be required during IAM user sign-in.
12. Repeat steps no. 4 - 11 for each Amazon IAM user that you want to protect using Multi-Factor Authentication (MFA).

Instances

- archit
- amit
- ownux
- Usama
- satoru

References

- https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_examples_aws_my-sec-creds-self-manage-mfa-only.html

S3 Bucket Logging Disabled

Risk:	**Low**	**STATUS:** Vulnerable
CVSS Score:	**3.7**	

(AV:Network/AC:High/PR:None/UI:None/C:Low/I:None/A:None)

Description

The Server Access Logging feature provides detailed records for the requests that are made to your Amazon S3 buckets. The log data includes the request type, the resources that are specified in the request, and the time and date that the request was processed.

Once enabled, the feature can provide useful data for security and compliance audits, and can help you learn about your user base and understand your Amazon S3 bill.

Ensure that Server Access Logging feature is enabled for your Amazon S3 buckets in order to track access requests useful for security and access audits. By default, Server Access Logging is not enabled for S3 buckets.

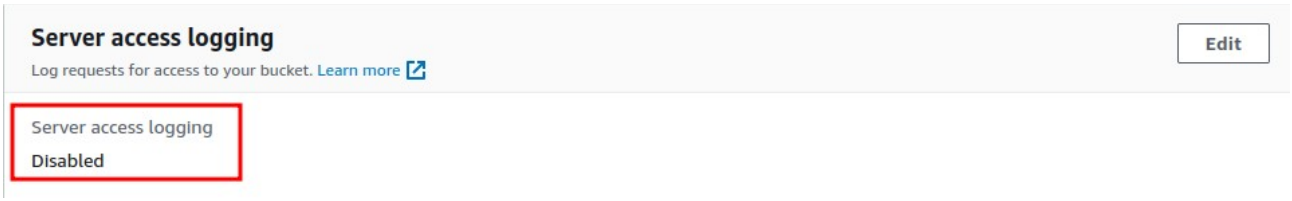
This rule can help you with the following compliance standards:

1. PCI
2. HIPAA
3. GDPR
4. APRA
5. MAS
6. NIST4

During the review of S3, it was discovered that logging of access is disabled which makes it hard to trace the compromised account incase of cyber incident.

Step by Step Proof

01. Sign in to the AWS Management Console.
02. Navigate to Amazon S3 console at https://console.aws.amazon.com/s3/.
03. Click on the name of the S3 bucket that you want to reconfigure.
04. Select the Properties tab from the console menu to access the bucket properties.
05. In the Server access logging section, choose Edit to modify the feature configuration.



Remediation

Using AWS Console:

01. Sign in to the AWS Management Console.
02. Navigate to Amazon S3 console at https://console.aws.amazon.com/s3/.
03. Click on the name of the S3 bucket that you want to reconfigure.
04. Select the **Properties** tab from the console menu to access the bucket properties.
05. In the **Server access logging** section, choose **Edit** to modify the feature configuration.
06. On the **Edit server access logging** page, perform the following actions:
Choose **Enable** under **Server access logging** to enable the Server Access Logging feature for the selected Amazon S3 bucket.
For **Target bucket**, choose **Browse S3** and select the name of the destination bucket and folder for the access logs. You can use the same bucket for logs storage, however, when your source bucket and destination (target) bucket are the same, additional logs are created for the logs that are written to the bucket. These extra logs can increase your storage billing and make it harder to find the logs that you're looking for.
Choose **Save changes** to apply the configuration changes. Once the feature is enabled, Amazon S3 console will automatically update your bucket access control list (ACL) to include access to the S3 log delivery group.
07. Repeat steps no. 3-6 to enable Server Access Logging feature for other Amazon S3 buckets available in your AWS cloud account.

Instances

- appstream-app-settings-us-west-2-594842924673-y686ives
- iiotreport
- apttower
- ttchat
- tasktower
- ttmail-beanstalk
- teamlocusdbbackup
- compute-optimizer-deep
- pandoarch
- indikitch.com
- ttmail-beanstalk-backup
- devendra-img
- accordionwagebump
- cny-nas-drive-81.30
- datasync-restore-pandoarchdb
- cost-and-usage-report-deep
- betameet.teamlocus.com
- deepdocsses
- gbabu-img
- teamlocus-elastic-backup-all
- cf-templates-82lne1hxbwuv-us-west-2
- audiofilestotext
- ttmail-beanstalk-mail-pending
- pandoarch-empschedule
- appstream2-36fb080bb8-us-west-2-594842924673
- testdeepkiran
- chatbot-cf-new
- ttdrive
- elasticbeanstalk-us-west-2-594842924673
- awsaptdb02bucket
- allwebsites
- pandoarchdbbkp
- athena-result-deepkiran
- cloudtrail-deep
- datasync-restore-teamlocus
- pandodb-deleted-bkp
- meet.teamlocus.com
- tlchatelasticsearchbackup
- www.indikitch.com
- call.teamlocus.com
- deepfoods-cloudformation-template
- datasync-restore-apt
- deepfoods

References

- https://docs.aws.amazon.com/AmazonS3/latest/userguide/enable-server-access-logging.html

S3 bucket without versioning

Risk:	**Low**	**STATUS:** Vulnerable
CVSS Score:	**3.7**	

(AV:Network/AC:High/PR:None/UI:None/C:Low/I:None/A:None)

Description

Versioning-enabled Amazon S3 buckets will allow you to preserve, retrieve, and restore every version of an S3 object.

S3 versioning can be used for data protection and retention scenarios such as recovering objects that have been accidentally/intentionally deleted or overwritten by AWS users or applications and archiving previous versions of objects to Amazon S3 Glacier for long-term low-cost storage. With S3 versioning, you can easily recover from both unintended user actions and application failures.

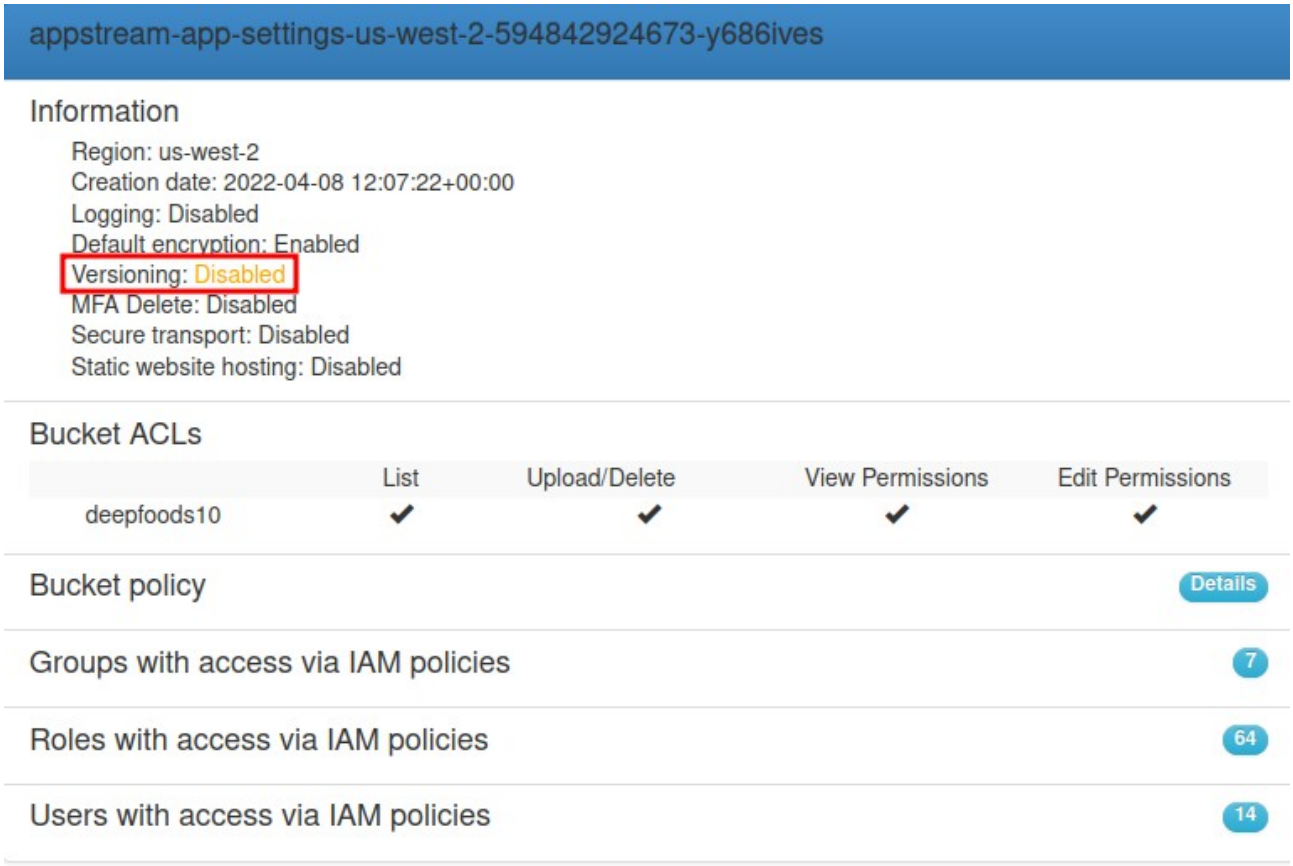
Ensure that S3 object versioning is enabled for your Amazon S3 buckets in order to preserve and recover overwritten and deleted S3 objects as an extra layer of data protection and/or data retention.
This rule can help you with the following compliance standards:

1. PCI
2. APRA
3. MAS
4. NIST4

During the review of S3 bucket, it was discovered that the versioning is disabled which makes it near impossible to retrive previous version of the object in case of cyber incident.

Step by Step Proof

01. Sign in to the AWS Management Console.
02. Navigate to Amazon S3 console at https://console.aws.amazon.com/s3/.
03. Click on the name of the S3 bucket that you want to reconfigure.
04. Select the Properties tab from the console menu to access the bucket properties.



Remediation

01. Sign in to the AWS Management Console.
02. Navigate to Amazon S3 console at https://console.aws.amazon.com/s3/.
03. Click on the name of the S3 bucket that you want to reconfigure.
04. Select the **Properties** tab from the console menu to access the bucket properties.
05. In the **Bucket Versioning** section, choose **Edit** to modify the object versioning configuration.
06. On the **Edit Bucket Versioning** page, select **Enable** under **Bucket Versioning** to enable the feature. Choose **Save changes** to apply the configuration changes. After enabling object versioning, you might need to update your lifecycle rules to manage previous versions of objects.
07. Repeat steps no. 3-6 to enable S3 object versioning for other Amazon S3 buckets available in your AWS cloud account.

Instances

- appstream-app-settings-us-west-2-594842924673-y686ives
- iiotreport
- ttmail-beanstalk

- compute-optimizer-deep
- pandoarch
- indikitch.com
- ttmail-beanstalk-backup
- devendra-img
- accordionwagebump
- cny-nas-drive-81.30
- datasync-restore-pandoarchdb
- cost-and-usage-report-deep
- betameet.teamlocus.com
- deepdocsses
- cf-templates-82lne1hxbwuv-us-west-2
- audiofilestotext
- ttmail-beanstalk-mail-pending
- pandoarch-empschedule
- appstream2-36fb080bb8-us-west-2-594842924673
- testdeepkiran
- chatbot-cf-new
- ttdrive
- elasticbeanstalk-us-west-2-594842924673
- awsaptdb02bucket
- athena-result-deepkiran
- cloudtrail-deep
- datasync-restore-teamlocus
- pandodb-deleted-bkp
- meet.teamlocus.com
- www.indikitch.com
- call.teamlocus.com
- deepfoods-cloudformation-template
- datasync-restore-apt

References

- https://docs.aws.amazon.com/AmazonS3/latest/userguide/manage-versioning-examples.html

Single AZ RDS instance

Risk:	**Low**	**STATUS:** Vulnerable
CVSS Score:	**3.1**	

(AV:Network/AC:High/PR:Low/UI:None/C:Low/I:None/A:None)

Description

When Multi-AZ is enabled, AWS automatically provision and maintain a synchronous database standby replica on a dedicated hardware in a different datacenter (known as Availability Zone). AWS RDS will automatically switch from the primary cluster to the available standby replica in the event of a failure such as an Availability Zone outage, an internal hardware or network outage, a software failure or in case of planned interruptions such as software patching or changing the RDS cluster type.

Ensure that your RDS Aurora clusters are using Multi-AZ deployment configurations for high availability and automatic failover support fully managed by AWS.

This rule can help you with the following compliance standards:

- NIST4

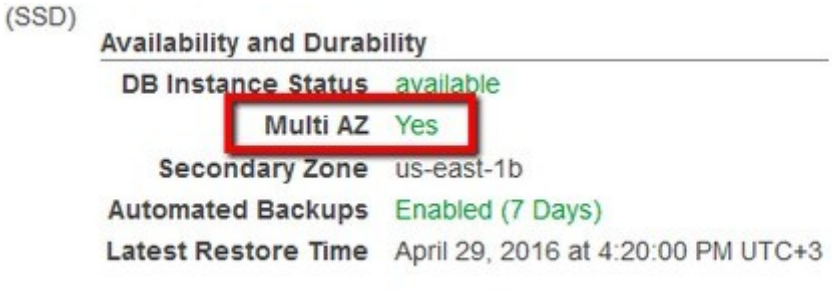
Step by Step Proof

01. Login to the AWS Management Console.
02. Navigate to RDS dashboard at https://console.aws.amazon.com/rds/.
03. In the navigation panel, under RDS Dashboard, click Clusters.
04. Select the RDS cluster that you want to examine.
05. Click Cluster Actions button from the dashboard top menu and select Modify.

Remediation

To update your RDS clusters configuration and enable Multi-AZ deployment, perform the following:

01. Login to the AWS Management Console.
02. Navigate to RDS dashboard at https://console.aws.amazon.com/rds/.
03. In the navigation panel, under **RDS Dashboard**, click **Clusters**.
04. Select the RDS cluster that you want to examine.
05. Click **Cluster Actions** button from the dashboard top menu and select **Modify**.
06. On the **Modify DB Cluster: <cluster identifier>** page, under Cluster Specifications section, select **Yes** from the **Multi-AZ Deployment** dropdown list.
07. At the bottom of the page, check **Apply Immediately** to apply the changes immediately.
08. Click **Continue**.
09. Review the changes and click **Modify DB Cluster**. The cluster status should change from available to modifying and back to available. Once the feature is enabled, the **Multi AZ** status should change to **Yes**:



10. Repeat steps no. 4 - 9 for each RDS cluster available in the current region. Change the AWS region from the navigation bar to repeat the process for other regions.

Instances

- iiot-db

References

- https://aws.amazon.com/rds/faqs/
- https://aws.amazon.com/rds/features/multi-az/
- https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Concepts.-MultiAZ.html

Recommendation Summary

- Implement Encryption on AMI Storage
- Implement Encryption Not Enabled
- Restrict Usage of S3 GET Action
- Close or filter SSH port
- Disallow iam:PassRole*
- Disable or Expire Root Account’s X.509 Cert
- Implement Key Rotation
- Close or Filter RDP port
- Implement restriction over tcp port
- Enable encryption of EBS
- Enable access logs to ELBv2
- Enable deletion protection for ELBv2
- Implement Redirect HTTP to HTTPS
- Close or filter FTP Port
- Disable Whitelist of CIDRs
- Enable Encryption of Ecr Repositories
- Enable Scan-On-Push in ECR
- Enable MFA in Cross-Account Access
- Disable inline iam:passrole
- Disable Bucket allowing HTTP
- Implement SSL in Redshift
- Enable Deletion MFA in S3
- Enable MFA in Root Account
- Close or restrict MsSQL port
- Enable MFA for users
- Enable S3 logging
- Implement S3 versioning
- Enable Multi Availiability Zone