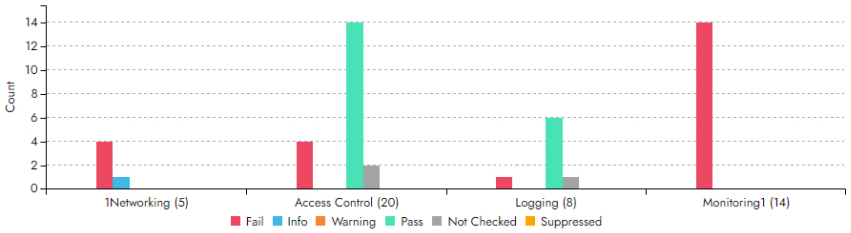
Compliance Score : 45 %

| 23 | 1 | 0 | 20 | 3 | 0 |
|---|---|---|---|---|---|
| Fail | Info | Warning | Pass | Not Checked | Suppressed |




### 1Networking

| Tags | Control | Reason | Severity | Status |
|---|---|---|---|---|
| | Ensure no security groups allow ingress from 0.0.0.0/0 to port 22 | default - demo-11111111(ap-south ... | HIGH | Fail |
| | Ensure no security groups allow ingress from 0.0.0.0/0 to port 3389 | default - demo-11111111(ap-south ... | HIGH | Fail |
| | Ensure VPC flow logging is enabled in all VPCs | No VPC flow logs found in regi ... | HIGH | Fail |
| | Ensure the default security group of every VPC restricts all traffic | Security Group (default) in VP ... | HIGH | Fail |
| | Ensure routing tables for VPC peering are 'least access' | No VPC peering found in region ap-south-1 ... | HIGH | Info |

### Access Control

| Tags | Control | Reason | Severity | Status |
|---|---|---|---|---|
| | Avoid the use of the root account | - | HIGH | Pass |
| | Ensure IAM password policy prevents password reuse | - | MEDIUM | Pass |
| | Ensure IAM password policy expires passwords within 90 days or less | - | MEDIUM | Pass |
| | Ensure no root account access key exists | - | HIGH | Pass |
| | Ensure MFA is enabled for the root account | - | HIGH | Pass |
| | Ensure hardware MFA is enabled for the account | Hardware based MFA is not enab ... | HIGH | Fail |
| | Ensure security questions are registered in the AWS account | This activity c ... | LOW | Not Checked |
| | Ensure IAM policies are attached only to groups or roles | User (abhijeet) has 3 policy/p ... | MEDIUM | Fail |
| | Ensure security contact information is registered | This activity c ... | LOW | Not Checked |
| | Ensure IAM instance roles are used for AWS resource access from instances | - | MEDIUM | Pass |
| | Ensure multi-factor authentication (MFA) is enabled for all IAM users that have a console password | - | HIGH | Pass |
| | Do not setup access keys during initial user setup for all IAM users that have a console password | - | MEDIUM | Pass |
| | Ensure IAM policies that allow full '*:*' administrative privileges are not created | - | HIGH | Pass |
| | Ensure credentials unused for 90 days or greater are disabled | Password not used since last 9 ... | MEDIUM | Fail |
| | Ensure access keys are rotated every 90 days or less | Access key 1 has not been rota ... | MEDIUM | Fail |
| | Ensure IAM password policy requires at least one uppercase letter | - | MEDIUM | Pass |
| | Ensure IAM password policy require at least one lowercase letter | - | MEDIUM | Pass |
| | Ensure IAM password policy require at least one symbol | - | MEDIUM | Pass |
| | Ensure IAM password policy require at least one number | - | MEDIUM | Pass |
| | Ensure IAM password policy requires minimum length of 14 or greater | - | MEDIUM | Pass |

### Logging

| Tags | Control | Reason | Severity | Status |
|---|---|---|---|---|
| | Maintain current contact details | This activity c ... | LOW | Not Checked |
| | Ensure S3 bucket access logging is enabled on the CloudTrail S3 bucket | - | HIGH | Pass |
| | Ensure CloudTrail is enabled in all regions for atleast one trail | - | MEDIUM | Pass |
| | Ensure CloudTrail log file validation is enabled | - | LOW | Pass |
| | Ensure CloudTrail trails are integrated with CloudWatch Logs | - | MEDIUM | Pass |
| | Ensure AWS Config is enabled in all regions | - | HIGH | Pass |
| | Ensure CloudTrail logs are encrypted at rest using KMS CMKs | - | HIGH | Pass |
| | Ensure rotation for customer created CMKs is enabled | Rotation for customer created ... | HIGH | Fail |

### Monitoring1

| Tags | Control | Reason | Severity | Status |
|---|---|---|---|---|
| | Ensure a log metric filter and alarm exist for unauthorized API calls | A log metric alarm does not ex ... | MEDIUM | Fail |
| | Ensure a log metric filter and alarm exist for security group changes | A log metric alarm for 'Securi ... | MEDIUM | Fail |

| | | | | |
|---|---|---|---|---|
| | Ensure a log metric filter and alarm exist for changes to Network Access Control Lists (NACL) | A log metric alarm for 'NACL C ... | MEDIUM | ⊗ |
| | Ensure a log metric filter and alarm exist for changes to network gateways | A log metric alarm for 'Change ... | MEDIUM | ⊗ |
| | Ensure a log metric filter and alarm exist for route table changes | A log metric alarm for 'Change ... | MEDIUM | ⊗ |
| | Ensure a log metric filter and alarm exist for VPC changes | A log metric alarm for 'VPC Ch ... | MEDIUM | ⊗ |
| | Ensure a log metric filter and alarm exist for Management Console sign-in without MFA | A log metric alarm for 'Manage ... | MEDIUM | ⊗ |
| | Ensure a log metric filter and alarm exist for usage of root account | A log metric alarm for 'Usage ... | MEDIUM | ⊗ |
| | Ensure a log metric filter and alarm exist for IAM policy changes | A log metric alarm for 'IAM Po ... | MEDIUM | ⊗ |
| | Ensure a log metric filter and alarm exist for CloudTrail configuration changes | A log metric alarm for 'Cloudt ... | MEDIUM | ⊗ |
| | Ensure a log metric filter and alarm exist for AWS Management Console authentication failures | A log metric alarm for 'Consol ... | MEDIUM | ⊗ |
| | Ensure a log metric filter and alarm exist for disabling or scheduled deletion of customer created CMKs | A log metric alarm for 'Disabl ... | MEDIUM | ⊗ |
| | Ensure a log metric filter and alarm exist for S3 bucket policy changes | A log metric alarm for 'S3 Buc ... | MEDIUM | ⊗ |
| | Ensure a log metric filter and alarm exist for AWS Config configuration changes | A log metric alarm for 'AWS Co ... | MEDIUM | ⊗ |

**Tags:**

### Control : Ensure no security groups allow ingress from 0.0.0.0/0 to port 22

**Status:** ⊗ Fail **Configuration Name:** Demo-conf **Group Name:** 1Networking

**Severity :** HIGH

**Reason:**

1. default - demo-11111111(ap-south-1) allows access to Port 22 from 0.0.0.0/0

**Description :**

*Security groups provide stateful filtering of ingress/egress network traffic to AWS resources. It is recommended that no security group allows unrestricted ingress access to port 22 .*

**Remediation Steps:**

*Perform the following to implement the prescribed state:*

1. *Login to the AWS Management Console at https://console.aws.amazon.com/vpc/home*
2. *In the left pane, click Security Groups*
3. *For each security group, perform the following:*
    1. *Select the security group*
    2. *Click the Inbound Rules tab*
    3. *Identify the rules to be removed*
    4. *Click the x in the Remove column*
    5. *Click Save*

*Impact:*
*For updating an existing environment, care should be taken to ensure that administrators currently relying on an existing ingress from 0.0.0.0/0 have access to ports 22 and/or 3389 through another security group.*

**Tags:**

### Control : Ensure no security groups allow ingress from 0.0.0.0/0 to port 3389

**Status:** ⊗ Fail **Configuration Name:** Demo-conf **Group Name:** 1Networking

**Severity :** HIGH

**Reason:**

1. default - demo-11111111(ap-south-1) allows access to Port 3389 from 0.0.0.0/0

**Description :**

*Security groups provide stateful filtering of ingress/egress network traffic to AWS resources. It is recommended that no security group allows unrestricted ingress access to port 3389 .*

**Remediation Steps:**

*Perform the following to implement the prescribed state:*

1. *Login to the AWS Management Console at https://console.aws.amazon.com/vpc/home*
2. *In the left pane, click Security Groups*
3. *For each security group, perform the following:*
    1. *Select the security group*
    2. *Click the Inbound Rules tab*
    3. *Identify the rules to be removed*
    4. *Click the x in the Remove column*
    5. *Click Save*

*Impact:*
*For updating an existing environment, care should be taken to ensure that administrators currently relying on an existing ingress from 0.0.0.0/0 have access to ports 22 and/or 3389 through another security group.*

**Tags:**

### Control : Ensure VPC flow logging is enabled in all VPCs

**Status:** ⊗ Fail **Configuration Name:** Demo-conf **Group Name:** 1Networking

**Severity :** HIGH

**Reason:**

1. No VPC flow logs found in region eu-north-1

**Description :**

*VPC Flow Logs is a feature that enables you to capture information about the IP traffic going to and from network interfaces in your VPC. After you've created a flow log, you can view and retrieve its data in Amazon CloudWatch Logs.*
*It is recommended that VPC FlowLogs be enabled for packet "Rejects" for VPCs.*

**Remediation Steps:**

*01)Sign in to the AWS Management Console.*

*02)Navigate to VPC dashboard at https://console.aws.amazon.com/vpc/.*

*03)In the left navigation panel, select Your VPCs.*

*04)Select the VPC that you need to check.*

*05)Select the Flow Logs tab from the bottom panel and click Create Flow Log:*

*06)In the Create Flow Log dialog box, enter the following details:*

*Filter: select the filter that describes the type of traffic to be logged – accepted, rejected, or all.*

*Role: enter the name of the IAM role that will allow permissions to publish to the CloudWatch Logs log group.*

*Destination Log Group: enter a name for the new CloudWatch Logs log group, where the flow logs will be published.*

*07Review the flow log configuration and click Create Flow Log:*

*The log group will be available in approximately 10 minutes after you create the flow log. To access it, just click on the log group name listed under the CloudWatch Logs Group column:*

*or open the CloudWatch Logs dashboard at https://console.aws.amazon.com/cloudwatch/home#logs:*

---

**Tags:**

## Control : Ensure the default security group of every VPC restricts all traffic

**Status:** Fail **Configuration Name:** Demo-conf **Group Name:** 1Networking

**Severity :** HIGH

**Reason:**

1. Security Group (default) in VPC C2-DEV-VPC-1:demo-11111111 from (us-east-1) allows egress traffic from 0.0.0.0/0
2. Security Group (default) in VPC :vpc-799f7110 from (eu-north-1) allows egress traffic from 0.0.0.0/0
3. Security Group (default) in VPC :vpc-46de3f2f from (ap-south-1) allows ingress traffic from 0.0.0.0/0
4. Security Group (default) in VPC :vpc-46de3f2f from (ap-south-1) allows ingress traffic from 0.0.0.0/0

**Description :**

*A VPC comes with a default security group whose initial settings deny all inbound traffic, allow all outbound traffic, and allow all traffic between instances assigned to the security group. If you don't specify a security group when you launch an instance, the instance is automatically assigned to this default security group. Security groups provide stateful filtering of ingress/egress network traffic to AWS resources. It is recommended that the default security group restrict all traffic.*
*The default VPC in every region should have it's default security group updated to comply. Any newly created VPCs will automatically contain a default security group that will need remediation to comply with this recommendation.*

*NOTE: When implementing this recommendation, VPC flow logging is invaluable in determining the least privilege port access required by systems to work properly because it can log all packet acceptances and rejections occurring under the current security groups. This dramatically reduces the primary barrier to least privilege engineering - discovering the minimum ports required by systems in the environment. Even if the VPC flow logging recommendation in this benchmark is not adopted as a permanent security measure, it should be used during any period of discovery and engineering for least privileged security groups.*

**Remediation Steps:**

*Security Group Members*
*Perform the following to implement the prescribed state:*

1. *Identify AWS resources that exist within the default security group*
2. *Create a set of least privilege security groups for those resources*
3. *Place the resources in those security groups*
4. *Remove the resources noted in #1 from the default security group*

*Security Group State*

1. *Login to the AWS Management Console at https://console.aws.amazon.com/vpc/home*
2. *Repeat the next steps for all VPCs - including the default VPC in each AWS region:*
3. *In the left pane, click Security Groups*
4. *For each default security group, perform the following:*
    1. *Select the default security group*
    2. *Click the Inbound Rules tab*
    3. *Remove any inbound rules*
    4. *Click the Outbound Rules tab*
    5. *Remove any inbound rules*

*Recommended:*
*IAM groups allow you to edit the "name" field. After remediating default groups rules for all VPCs in all regions, edit this field to add text similar to "DO NOT USE. DO NOT ADD RULES"*

*Impact:*
*Implementing this recommendation in an existing VPC containing operating resources requires extremely careful migration planning as the default security groups are likely to be enabling many ports that are unknown. Enabling VPC flow logging (of accepts) in an existing environment that is know to be breach free will reveal the current pattern of ports being used for each instance to communicate successfully.*

*References:*

1. *http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-network-security.html*

---

**Tags:**

## Control: Ensure routing tables for VPC peering are 'least access'

**Status:** Info **Configuration Name:** Demo-conf **Group Name:** 1Networking

**Severity :** HIGH

**Reason:**

1. No VPC peering found in region us-west-2
2. No VPC peering found in region us-west-1
3. No VPC peering found in region us-east-2
4. No VPC peering found in region us-east-1
5. No VPC peering found in region eu-central-1
6. No VPC peering found in region ap-southeast-2
7. No VPC peering found in region ap-southeast-1
8. No VPC peering found in region sa-east-1
9. No VPC peering found in region ca-central-1
10. No VPC peering found in region ap-northeast-1
11. No VPC peering found in region ap-northeast-2
12. No VPC peering found in region eu-west-1
13. No VPC peering found in region eu-west-2
14. No VPC peering found in region eu-west-3
15. No VPC peering found in region eu-north-1
16. No VPC peering found in region ap-south-1

**Description :**

*Once a VPC peering connection is estalished, routing tables must be updated to establish any connections between the peered VPCs. These routes can be as specific as desired even peering a VPC to only a single host on the other side of the connection.*

**Remediation Steps:**

*N.A*

Tags:

### Control: Avoid the use of the root account

**Status:** Pass **Configuration Name:** Demo-conf **Group Name:** Access Control

**Severity :** HIGH

**Reason:** N.A

**Description :**
*The "root" account has unrestricted access to all resources in the AWS account. It is highly recommended that the use of this account be avoided.*

**Remediation Steps:**
*N.A*

---

Tags:

### Control: Ensure IAM password policy prevents password reuse

**Status:** Pass **Configuration Name:** Demo-conf **Group Name:** Access Control

**Severity :** MEDIUM

**Reason:** N.A

**Description :**
*IAM password policies can prevent the reuse of a given password by the same user. It is recommended that the password policy prevent the reuse of passwords.*

**Remediation Steps:**
*N.A*

---

Tags:

### Control: Ensure IAM password policy expires passwords within 90 days or less

**Status:** Pass **Configuration Name:** Demo-conf **Group Name:** Access Control

**Severity :** MEDIUM

**Reason:** N.A

**Description :**
*IAM password policies can require passwords to be rotated or expired after a given number of days. It is recommended that the password policy expire passwords after 90 days or less.*

**Remediation Steps:**
*N.A*

---

Tags:

### Control: Ensure no root account access key exists

**Status:** Pass **Configuration Name:** Demo-conf **Group Name:** Access Control

**Severity :** HIGH

**Reason:** N.A

**Description :**
*The root account is the most privileged user in an AWS account. AWS Access Keys provide programmatic access to a given AWS account. It is recommended that all access keys associated with the root account be removed.*

**Remediation Steps:**
*N.A*

---

Tags:

### Control: Ensure MFA is enabled for the root account

**Status:** Pass **Configuration Name:** Demo-conf **Group Name:** Access Control

**Severity :** HIGH

**Reason:** N.A

**Description :**
*The root account is the most privileged user in an AWS account. MFA adds an extra layer of protection on top of a user name and password. With MFA enabled, when a user signs in to an AWS website, they will be prompted for their user name and password as well as for an authentication code from their AWS MFA device.*

**Remediation Steps:**
*N.A*

---

Tags:

### Control : Ensure hardware MFA is enabled for the account

**Status:** Fail **Configuration Name:** Demo-conf **Group Name:** Access Control

**Severity :** HIGH

**Reason:**
1. Hardware based MFA is not enabled for root user

**Description :**
*The root account is the most privileged user in an AWS account. MFA adds an extra layer of protection on top of a user name and password. With MFA enabled, when a user signs in to an AWS website, they will be prompted for their user name and password as well as for an authentication code from their AWS MFA device. For Level 2, it is recommended that the root account be protected with a hardware MFA.*

**Remediation Steps:**

*Perform the following to establish a hardware MFA for the root account:*

*1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.*
*Note: To manage MFA devices for the root AWS account, you must use your root account credentials to sign in to AWS. You cannot manage MFA devices for the root account using other credentials.*
*2. Choose Dashboard , and under Security Status , expand Activate MFA on your root account.*
*3. Choose Activate MFA*
*4. In the wizard, choose A hardware MFA device and then choose Next Step .*
*5. In the Serial Number box, enter the serial number that is found on the back of the MFA device.*
*6. In the Authentication Code 1 box, enter the six-digit number displayed by the MFA device. You might need to press the button on the front of the device to display the number.*
*7. Wait 30 seconds while the device refreshes the code, and then enter the next six- digit number into the Authentication Code 2 box. You might need to press the button on the front of the device again to display the second number.*
*8. Choose Next Step . The MFA device is now associated with the AWS account. The next time you use your AWS account credentials to sign in, you must type a code from the hardware MFA device.*

*References:*

1. *Order Hardware MFA: http://onlinenoram.gemalto.com/*
2. *http://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_mfa_enable_virtual.html*
3. *http://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_mfa_enable_physical.html#enable-hw-mfa-for-root*

---

**Tags:**

## Control: Ensure security questions are registered in the AWS account

**Status:** ☐ Not Checked **Configuration Name:** Demo-conf **Group Name:** Access Control

**Severity :** LOW

**Reason:**

1. This activity can only be performed via the AWS Console by logging into the Root account

**Description**

*The AWS support portal allows account owners to establish security questions that can be used to authenticate individuals calling AWS customer service for support. It is recommended that security questions be established.*

**Remediation Steps:**

*1. Perform the following in the AWS Management Console:*

*1. Login to the AWS account as root*
*2. Click on the <Root_Account_Name> from the top right of the console*
*3. From the drop-down menu Click My Account*
*4. Scroll down to the Configure Security Questions section*
*5. Click on Edit*
*6. Click on each Question*
*From the drop-down select an appropriate question*
*Click on the Answer section*
*Enter an appropriate answer*
*Follow process for all 3 questions*
*7. Click Update when complete*
*8. Place Questions and Answers and place in a secure physical location*

---

**Tags:**

## Control : Ensure IAM policies are attached only to groups or roles

**Status:** ⊗ Fail **Configuration Name:** Demo-conf **Group Name:** Access Control

**Severity :** MEDIUM

**Reason:**

1. User (temp-user) has 1 policy/policies [AmazonEC2ReadOnlyAccess] directly attached to it
2. User (abc.abc@abc.com) has 1 policy/policies [ReadOnlyAccess] directly attached to it
3. User (CloudwatchCustomMetrics) has 1 policy/policies [CloudwatchCustomMetrics] directly attached to it
4. User (cloudlytics-notifications) has 2 policy/policies [AmazonSESFullAccess, AmazonSNSFullAccess] directly attached to it
5. User (abc-cmp) has 1 policy/policies [ReadOnlyAccess] directly attached to it
6. User (abhijeet) has 3 policy/policies [Billing+BudgetAccess, Billing, Cost_Explorer] directly attached to it

**Description :**

*By default, IAM users, groups, and roles have no access to AWS resources. IAM policies are the means by which privileges are granted to users, groups, or roles. It is recommended that IAM policies be applied directly to groups and roles but not users.*

**Remediation Steps:**

*Perform the following to create an IAM group and assign a policy to it:*

*1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.*

*2. In the navigation pane, click Groups and then click Create New Group .*

*3. In the Group Name box, type the name of the group and then click Next Step .*

*4. In the list of policies, select the check box for each policy that you want to apply to all members of the group. Then click Next Step .*
*5. Click Create Group.*

*Perform the following to add a user to a given group:*

*1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.*

*2. In the navigation pane, click Groups*

*3. Select the group to add a user*

*4. Click Add Users To Group*

*5. Select the users to be added to the group*

*6. Click Add Users*

*Perform the following to remove a direct association between a user and policy:*

*1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.*

*2. In the left navigation pane, click on Users*

*3. For each user:*

*1. Select the user*

*2. Click on the Permissions tab*

*3. Expand Managed Policies*

*4. Click Detach Policy for each policy*

*5. Expand Inline Policies*

*6. Click Remove Policy for each policy*

*References:*

1. *http://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html*
2. *http://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_managed-vs-inline.html*

**Tags:**

### Control: Ensure security contact information is registered

**Status:** ☐ Not Checked **Configuration Name:** Demo-conf **Group Name:** Access Control

**Severity :** LOW

**Reason:**
1. This activity can only be performed via the AWS Console

**Description**
*AWS provides customers with the option of specifying the contact information for account's security team. It is recommended that this information be provided.*

**Remediation Steps:**
*Perform the following in the AWS Management Console to establish security contact information:*
*1. Click on your account name at the top right corner of the console.*

*2. From the drop-down menu Click My Account*

*3. Scroll down to the Alternate Contacts section*

*4. Enter contact information in the Security section*

*Note: Consider specifying an internal email distribution list to ensure emails are regularly monitored by more than one individual.*

**Tags:**

### Control: Ensure IAM instance roles are used for AWS resource access from instances

**Status:** ✓ Pass **Configuration Name:** Demo-conf **Group Name:** Access Control

**Severity :** MEDIUM

**Reason:** N.A

**Description :**
*AWS access from within AWS instances can be done by either encoding AWS keys into AWS API calls or by assigning the instance to a role which has an appropriate permissions policy for the required access. "AWS Access" means accessing the APIs of AWS in order to access AWS resources or manage AWS account resources.*

**Remediation Steps:**
*N.A*

**Tags:**

### Control: Ensure multi-factor authentication (MFA) is enabled for all IAM users that have a console password

**Status:** ✓ Pass **Configuration Name:** Demo-conf **Group Name:** Access Control

**Severity :** HIGH

**Reason:** N.A

**Description :**
*Multi-Factor Authentication (MFA) adds an extra layer of protection on top of a user name and password. With MFA enabled, when a user signs in to an AWS website, they will be prompted for their user name and password as well as for an authentication code from their AWS MFA device. It is recommended that MFA be enabled for all accounts that have a console password.*

**Remediation Steps:**
*N.A*

**Tags:**

### Control: Do not setup access keys during initial user setup for all IAM users that have a console password

**Status:** ✓ Pass **Configuration Name:** Demo-conf **Group Name:** Access Control

**Severity :** MEDIUM

**Reason:** N.A

**Description :**
*AWS console defaults the checkbox for creating access keys to enabled. This results in many access keys being generated unnecessarily. In addition to unnecessary credentials, it also generates unnecessary management work in auditing and rotating these keys.*

**Remediation Steps:**
*N.A*

**Tags:**

### Control: Ensure IAM policies that allow full '*:*' administrative privileges are not created

**Status:** ✓ Pass **Configuration Name:** Demo-conf **Group Name:** Access Control

**Severity :** HIGH

**Reason:** N.A

**Description :**

*IAM policies are the means by which privileges are granted to users, groups, or roles. It is recommended and considered a standard security advice to grant least privilege—that is, granting only the permissions required to perform a task. Determine what users need to do and then craft policies for them that let the users perform only those tasks, instead of allowing full administrative privileges.*

**Remediation Steps:**

*N.A*

---

**Tags:**

### Control : Ensure credentials unused for 90 days or greater are disabled

**Status:** ⊗ Fail **Configuration Name:** Demo-conf **Group Name:** Access Control

**Severity :** MEDIUM

**Reason:**

1. Access Key 1 not used since last 90 days for entity (temp-user)
2. Password not used since last 90 days for entity (abc.abc@abc.com)

**Description :**

*AWS IAM users can access AWS resources using different types of credentials, such as passwords or access keys. It is recommended that all credentials that have been unused in 90 or greater days be removed or deactivated.*

**Remediation Steps:**

*Perform the following to remove or deactivate credentials:*

1. *Login to the AWS Management Console*
2. *Click Services*
3. *Click IAM*
4. *Click on Users*
5. *Click on Security Credentials*
6. *As an Administrator*
    1. *Click on Make Inactive for credentials that have not been used in 90 Days*
7. *As an IAM User*

*Click on Make Inactive or Delete for credentials which have not been used in 90 Days*

---

**Tags:**

### Control : Ensure access keys are rotated every 90 days or less

**Status:** ⊗ Fail **Configuration Name:** Demo-conf **Group Name:** Access Control

**Severity :** MEDIUM

**Reason:**

1. Access key 2 has not been rotated since last 90 days or Access key 2 has not been used since last key rotation for entity (root)
2. Access key 1 has not been rotated since last 90 days or Access key 1 has not been used since last key rotation for entity (temp-user)
3. Access key 1 has not been rotated since last 90 days or Access key 1 has not been used since last key rotation for entity (CloudwatchCustomMetrics)
4. Access key 1 has not been rotated since last 90 days or Access key 1 has not been used since last key rotation for entity (cloudlytics-notifications)
5. Access key 1 has not been rotated since last 90 days or Access key 1 has not been used since last key rotation for entity (cloudlytics)
6. Access key 1 has not been rotated since last 90 days or Access key 1 has not been used since last key rotation for entity (abc-cmp)
7. Access key 1 has not been rotated since last 90 days or Access key 1 has not been used since last key rotation for entity (abhijeet)
8. Access key 1 has not been rotated since last 90 days or Access key 1 has not been used since last key rotation for entity (cloudlytics_mail_user)

**Description :**

*Access keys consist of an access key ID and secret access key, which are used to sign programmatic requests that you make to AWS. AWS users need their own access keys to make programmatic calls to AWS from the AWS Command Line Interface (AWS CLI), Tools for Windows PowerShell, the AWS SDKs, or direct HTTP calls using the APIs for individual AWS services. It is recommended that all access keys be regularly rotated.*

**Remediation Steps:**

*Perform the following to rotate access keys:*

*1. Login to the AWS Management Console*

*2. Click Services*

*3. Click IAM*

*4. Click on Users*

*5. Click on Security Credentials*

*6. As an Administrator*

*Click on Make Inactive for keys that have not been rotated in 90 Days*

*7. As an IAM User*

*Click on Make Inactive or Delete for keys which have not been rotated or used in 90 Days*

*8. Click on Create Access Key*

*9. Update programmatic call with new Access Key credentials*

---

**Tags:**

### Control: Ensure IAM password policy requires at least one uppercase letter

**Status:** ⊘ Pass **Configuration Name:** Demo-conf **Group Name:** Access Control

**Severity :** MEDIUM

**Reason:** N.A

**Description :**

*Password policies are, in part, used to enforce password complexity requirements. IAM password policies can be used to ensure password are comprised of different character sets. It is recommended that the password policy require at least one uppercase letter.*

**Remediation Steps:**

*N.A*

---

**Tags:**

### Control: Ensure IAM password policy require at least one lowercase letter

**Status:** Pass **Configuration Name:** Demo-conf **Group Name:** Access Control

**Severity :** MEDIUM

**Reason:** N.A

**Description :**

*Password policies are, in part, used to enforce password complexity requirements. IAM password policies can be used to ensure password are comprised of different character sets. It is recommended that the password policy require at least one lowercase letter.*

**Remediation Steps:**

*N.A*

**Tags:**

### Control: Ensure IAM password policy require at least one symbol

**Status:** Pass **Configuration Name:** Demo-conf **Group Name:** Access Control

**Severity :** MEDIUM

**Reason:** N.A

**Description :**

*Password policies are, in part, used to enforce password complexity requirements. IAM password policies can be used to ensure password are comprised of different character sets. It is recommended that the password policy require at least one symbol.*

**Remediation Steps:**

*N.A*

**Tags:**

### Control: Ensure IAM password policy require at least one number

**Status:** Pass **Configuration Name:** Demo-conf **Group Name:** Access Control

**Severity :** MEDIUM

**Reason:** N.A

**Description :**

*Password policies are, in part, used to enforce password complexity requirements. IAM password policies can be used to ensure password are comprised of different character sets. It is recommended that the password policy require at least one number.*

**Remediation Steps:**

*N.A*

**Tags:**

### Control: Ensure IAM password policy requires minimum length of 14 or greater

**Status:** Pass **Configuration Name:** Demo-conf **Group Name:** Access Control

**Severity :** MEDIUM

**Reason:** N.A

**Description :**

*Password policies are, in part, used to enforce password complexity requirements. IAM password policies can be used to ensure password are at least a given length. It is recommended that the password policy require a minimum password length 14.*

**Remediation Steps:**

*N.A*

**Tags:**

### Control: Maintain current contact details

**Status:** Not Checked **Configuration Name:** Demo-conf **Group Name:** Logging

**Severity :** LOW

**Reason:**

1. This activity can only be performed via the AWS Console, with a user who has permission to read and write Billing information (aws-portal:*Billing )

**Description**

*Ensure contact email and telephone details for AWS accounts are current and map to more than one individual in your organisation. An AWS account supports a number of contact details, and AWS will use these to contact the account owner if activity judged to be in breach of Acceptable Use Policy or indicative of likely security compromise is observed by the AWS Abuse team. Contact details should not be for a single individual, as circumstances may arise where that individual is unavailable. Email contact details should point to a mail alias which forwards email to multiple individuals within the organisation; where feasible, phone contact details should point to a PABX hunt group or other call-forwarding system.*

**Remediation Steps:**

*This activity can only be performed via the AWS Console, with a user who has permission to read and write Billing information (aws-portal:*Billing ).*

*1. Sign in to the AWS Management Console and open the Billing and Cost Management console at https://console.aws.amazon.com/billing/home#/.*

*2. On the navigation bar, choose your account name, and then choose My Account.*

*3. On the Account Settings page, next to Account Settings, choose Edit.*

*4. Next to the field that you need to update, choose Edit.*

*5. After you have entered your changes, choose Save changes.*

*6. After you have made your changes, choose Done.*

*7. To edit your contact information, under Contact Information, choose Edit.*

*8. For the fields that you want to change, type your updated information, and then choose Update.*

*References:*

1. *https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/manage-account-payment.html#contact-info*

Tags:

### Control: Ensure S3 bucket access logging is enabled on the CloudTrail S3 bucket

**Status:** Pass **Configuration Name:** Demo-conf **Group Name:** Logging

**Severity :** HIGH

**Reason:** N.A

**Description :**
*S3 Bucket Access Logging generates a log that contains access records for each request made to your S3 bucket. An access log record contains details about the request, such as the request type, the resources specified in the request worked, and the time and date the request was processed. It is recommended that bucket access logging be enabled on the CloudTrail S3 bucket.*

**Remediation Steps:**
*N.A*

---

Tags:

### Control: Ensure CloudTrail is enabled in all regions for atleast one trail

**Status:** Pass **Configuration Name:** Demo-conf **Group Name:** Logging

**Severity :** MEDIUM

**Reason:** N.A

**Description :**
*AWS CloudTrail is a web service that records AWS API calls for your account and delivers log files to you. The recorded information includes the identity of the API caller, the time of the API call, the source IP address of the API caller, the request parameters, and the response elements returned by the AWS service. CloudTrail provides a history of AWS API calls for an account, including API calls made via the Management Console, SDKs, command line tools, and higher-level AWS services (such as CloudFormation).*

**Remediation Steps:**
*N.A*

---

Tags:

### Control: Ensure CloudTrail log file validation is enabled

**Status:** Pass **Configuration Name:** Demo-conf **Group Name:** Logging

**Severity :** LOW

**Reason:** N.A

**Description :**
*CloudTrail log file validation creates a digitally signed digest file containing a hash of each log that CloudTrail writes to S3. These digest files can be used to determine whether a log file was changed, deleted, or unchanged after CloudTrail delivered the log. It is recommended that file validation be enabled on all CloudTrails.*

**Remediation Steps:**
*N.A*

---

Tags:

### Control: Ensure CloudTrail trails are integrated with CloudWatch Logs

**Status:** Pass **Configuration Name:** Demo-conf **Group Name:** Logging

**Severity :** MEDIUM

**Reason:** N.A

**Description :**
*AWS CloudTrail is a web service that records AWS API calls made in a given AWS account. The recorded information includes the identity of the API caller, the time of the API call, the source IP address of the API caller, the request parameters, and the response elements returned by the AWS service. CloudTrail uses Amazon S3 for log file storage and delivery, so log files are stored durably. In addition to capturing CloudTrail logs within a specified S3 bucket for long term analysis, realtime analysis can be performed by configuring CloudTrail to send logs to CloudWatch Logs. For a trail that is enabled in all regions in an account, CloudTrail sends log files from all those regions to a CloudWatch Logs log group. It is recommended that CloudTrail logs be sent to CloudWatch Logs.*

*Note: The intent of this recommendation is to ensure AWS account activity is being captured, monitored, and appropriately alarmed on. CloudWatch Logs is a native way to accomplish this using AWS services but does not preclude the use of an alternate solution.*

**Remediation Steps:**
*N.A*

---

Tags:

### Control: Ensure AWS Config is enabled in all regions

**Status:** Pass **Configuration Name:** Demo-conf **Group Name:** Logging

**Severity :** HIGH

**Reason:** N.A

**Description :**
*AWS Config is a web service that performs configuration management of supported AWS resources within your account and delivers log files to you. The recorded information includes the configuration item (AWS resource), relationships between configuration items (AWS resources), any configuration changes between resources. It is recommended to enable AWS Config be enabled in all regions.*

**Remediation Steps:**
*N.A*

---

Tags:

### Control: Ensure CloudTrail logs are encrypted at rest using KMS CMKs

**Status:** Pass **Configuration Name:** Demo-conf **Group Name:** Logging

**Severity :** HIGH

**Reason:** N.A

**Description :**

*AWS CloudTrail is a web service that records AWS API calls for an account and makes those logs available to users and resources in accordance with IAM policies. AWS Key Management Service (KMS) is a managed service that helps create and control the encryption keys used to encrypt account data, and uses Hardware Security Modules (HSMs) to protect the security of encryption keys. CloudTrail logs can be configured to leverage server side encryption (SSE) and KMS customer created master keys (CMK) to further protect CloudTrail logs. It is recommended that CloudTrail be configured to use SSE-KMS.*

**Remediation Steps:**

*N.A*

---

**Tags:**

## Control : Ensure rotation for customer created CMKs is enabled

**Status:** ⊗ Fail **Configuration Name:** Demo-conf **Group Name:** Logging

**Severity :** HIGH

**Reason:**

1. Rotation for customer created CMKs is not enabled for (11111111-2222-eeee-qqqq-11111111111)
2. Rotation for customer created CMKs is not enabled for (11111111-2222-eeee-qqqq-11111111111)

**Description :**

*AWS Key Management Service (KMS) allows customers to rotate the backing key which is key material stored within the KMS which is tied to the key ID of the Customer Created customer master key (CMK). It is the backing key that is used to perform cryptographic operations such as encryption and decryption. Automated key rotation currently retains all prior backing keys so that decryption of encrypted data can take place transparently. It is recommended that CMK key rotation be enabled.*

**Remediation Steps:**

*Perform the following to enable rotation for customer created CMKs via the Management Console:*

1. *Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam.*
2. *In the left navigation pane, choose Encryption Keys .*
3. *Select a customer created master key (CMK)*
4. *Under the Key Policy section, move down to Key Rotation .*
5. *Check the Rotate this key every year checkbox.*

*Via CLI*

1. *Run the following command to enable key rotation:*
   *aws kms enable-key-rotation –key-id <kms_key_id>*

*References:*

1. *https://aws.amazon.com/kms/pricing/*

---

**Tags:**

## Control : Ensure a log metric filter and alarm exist for unauthorized API calls

**Status:** ⊗ Fail **Configuration Name:** Demo-conf **Group Name:** Monitoring1

**Severity :** MEDIUM

**Reason:**

1. A log metric alarm does not exist for 'Unauthorized API Calls' for Filter(s) [ 31UnauthAPICalls ] in ( us-east-1 )

**Description :**

*Real-time monitoring of API calls can be achieved by directing CloudTrail Logs to CloudWatch Logs and establishing corresponding metric filters and alarms. It is recommended that a metric filter and alarm be established for unauthorized API calls.*

**Remediation Steps:**

*Perform the following to Ensure a log metric filter and alarm exist for unauthorized API calls:*

*Note : Filter pattern for unauthorized API calls*

*"filterPattern": "[ ($.errorCode = \"*UnauthorizedOperation\") || ($.errorCode = \"AccessDenied*\") ]"*

*Perform the following to setup the metric filter, alarm, SNS topic, and subscription:*

1. *Identify the log group name configured for use with CloudTrail*
2. *Note the <cloudtrail_log_group_name> value associated with CloudWatchLogsLogGroupArn :*
3. *Create a metric filter based on filter pattern provided which checks for unauthorized API calls and the <cloudtrail_log_group_name> taken from step 2.*
   *Note : You can choose your own metricName and metricNamespace strings. Using the same metricNamespace for all Foundations Benchmark metrics will group them together.*
4. *Create an SNS topic that the alarm will notify*
   *Note : you can re-use the same topic for all monitoring alarms.*
5. *Create an SNS subscription to the topic created in step 4*
   *Note : you can re-use the same SNS subscription for all monitoring alarms.*
6. *Create an alarm that is associated with the CloudWatch Logs Metric Filter created in step 3 and an SNS topic created in step 4*
   *Note : set the period and threshold to values that fit your organization.*

*Impact:*

*This alert may be triggered by normal read-only console activities that attempt to opportunistically gather optional information, but gracefully fail if they don't have permissions.*
*If an excessive number of alerts are being generated then an organization may wish to consider adding read access to the limited IAM user permissions simply to quiet the alerts.*
*In some cases doing this may allow the users to actually view some areas of the system - any additional access given should be reviewed for alignment with the original limited IAM user intent.*

*References:*

1. *https://aws.amazon.com/sns/*

---

**Tags:**

## Control : Ensure a log metric filter and alarm exist for security group changes

**Status:** ⊗ Fail **Configuration Name:** Demo-conf **Group Name:** Monitoring1

**Severity :** MEDIUM

**Reason:**

1. A log metric alarm for 'Security Group Changes' does not exist for Filter(s) [ 310SecGroupChanges ] in ( us-east-1 )

**Description :**

*Real-time monitoring of API calls can be achieved by directing CloudTrail Logs to CloudWatch Logs and establishing corresponding metric filters and alarms. Security Groups are a stateful packet filter that controls ingress and egress traffic within a VPC. It is recommended that a metric filter and alarm be established changes to Security Groups.*

**Remediation Steps:**

*Perform the following to ensure a log metric filter and alarm exist for security group changes*

*Note : Filter pattern for security group changes*

*"filterPattern": "[ ($.eventName = AuthorizeSecurityGroupIngress) || ($.eventName = AuthorizeSecurityGroupEgress) || ($.eventName = RevokeSecurityGroupIngress) || ($.eventName = RevokeSecurityGroupEgress) || ($.eventName = CreateSecurityGroup) || ($.eventName = DeleteSecurityGroup)]"*

*Perform the following to setup the metric filter, alarm, SNS topic, and subscription:*

1. *Identify the log group name configured for use with CloudTrail*
2. *Note the <cloudtrail_log_group_name> value associated with CloudWatchLogsLogGroupArn :*
3. *Create a metric filter based on filter pattern provided which checks for security groups changes and the <cloudtrail_log_group_name> taken from step 2.*
   *Note : You can choose your own metricName and metricNamespace strings. Using the same metricNamespace for all Foundations Benchmark metrics will group them together.*
4. *Create an SNS topic that the alarm will notify*
   *Note : you can re-use the same topic for all monitoring alarms.*
5. *Create an SNS subscription to the topic created in step 4*
   *Note : you can re-use the same SNS subscription for all monitoring alarms.*
6. *Create an alarm that is associated with the CloudWatch Logs Metric Filter created in step 3 and an SNS topic created in step 4*
   *Note : set the period and threshold to values that fit your organization.*

---

**Tags:**

## Control : Ensure a log metric filter and alarm exist for changes to Network Access Control Lists (NACL)

**Status:** ⊗ Fail **Configuration Name:** Demo-conf **Group Name:** Monitoring1

**Severity :** MEDIUM

**Reason:**

1. A log metric alarm for 'NACL Changes' does not exist for Filter [ 311NACLChanges ] in ( us-east-1 )

**Description :**

*Real-time monitoring of API calls can be achieved by directing CloudTrail Logs to CloudWatch Logs and establishing corresponding metric filters and alarms. NACLs are used as a stateless packet filter to control ingress and egress traffic for subnets within a VPC. It is recommended that a metric filter and alarm be established for changes made to NACLs.*

**Remediation Steps:**

*Perform the following to ensure a log metric filter and alarm exist for changes to Network Access Control Lists (NACL)*

*Note : Filter pattern for changes to Network Access Control Lists (NACL)*

*"filterPattern": "[ ($.eventName = CreateNetworkAcl) || ($.eventName = CreateNetworkAclEntry) || ($.eventName = DeleteNetworkAcl) || ($.eventName = DeleteNetworkAclEntry) || ($.eventName = ReplaceNetworkAclEntry) || ($.eventName = ReplaceNetworkAclAssociation) ]"*

*Perform the following to setup the metric filter, alarm, SNS topic, and subscription:*

1. *Identify the log group name configured for use with CloudTrail*
2. *Note the <cloudtrail_log_group_name> value associated with CloudWatchLogsLogGroupArn :*
3. *Create a metric filter based on filter pattern provided which checks for NACL changes and the <cloudtrail_log_group_name> taken from step 2.*
   *Note : You can choose your own metricName and metricNamespace strings. Using the same metricNamespace for all Foundations Benchmark metrics will group them together.*
4. *Create an SNS topic that the alarm will notify*
   *Note : you can re-use the same topic for all monitoring alarms.*
5. *Create an SNS subscription to the topic created in step 4*
   *Note : you can re-use the same SNS subscription for all monitoring alarms.*
6. *Create an alarm that is associated with the CloudWatch Logs Metric Filter created in step 3 and an SNS topic created in step 4*
   *Note : set the period and threshold to values that fit your organization.*

---

**Tags:**

## Control : Ensure a log metric filter and alarm exist for changes to network gateways

**Status:** ⊗ Fail **Configuration Name:** Demo-conf **Group Name:** Monitoring1

**Severity :** MEDIUM

**Reason:**

1. A log metric alarm for 'Changes to Network Gateways' does not exist for Filter(s) [ 312NetworkGatewayChanges ] in ( us-east-1 )

**Description :**

*Real-time monitoring of API calls can be achieved by directing CloudTrail Logs to CloudWatch Logs and establishing corresponding metric filters and alarms. Network gateways are required to send/receive traffic to a destination outside of a VPC. It is recommended that a metric filter and alarm be established for changes to network gateways.*

**Remediation Steps:**

*Note : set the period and threshold to values that fit your organization.*
*To Create Metric Filter and Cloudwatch Alarm*

*1.Navigate to Cloudwatch dashboard at https://console.aws.amazon.com/cloudwatch/.*

*2.In the left navigation panel, select Logs.*

*3.Select the log group created for your CloudTrail trail event logs and click Create Metric Filter button.*

*4.On the Define Logs Metric Filter page, paste the following pattern inside the Filter Pattern box: [ ($.eventName = CreateCustomerGateway) || ($.eventName = DeleteCustomerGateway) || ($.eventName = AttachInternetGateway) || ($.eventName = CreateInternetGateway) || ($.eventName = DeleteInternetGateway) || ($.eventName = DetachInternetGateway) ]. This pattern will be used for scanning the AWS CloudTrail logs for event names like "CreateInternetGateway", "AttachInternetGateway" or "DeleteInternetGateway".*

*5.Review the metric filter config details then click Assign Metric.*

*6.On the Create Metric Filter and Assign a Metric page, perform the following:*

*a)In the Filter Name box, enter a unique name for the new filter, e.g. VPCGatewayConfigChanges.*

*b)In the Metric Namespace box, type CloudTrailMetrics.*

*c)In the Metric Name box, type GatewayEventCount for the metric identifier.*

*d)Click Show advanced metric settings to slide down the advanced settings section.*

*e)In the Metric Value box, enter 1.*

*7.Review the details then click Create Filter to generate your new CloudWatch Logs metric filter.*

*8.On the current page click Create Alarm: .*

*9.In the Create Alarm dialog box, provide the following information:*

*Within the Alarm Threshold section, in the Name and Description fields, enter a unique name and a short description for the new CloudWatch alarm.*

*Under Whenever: <Metric Name>, select >= (greater than or equal to) from the is dropdown list and enter 1 as the threshold value in the box next to the dropdown list to trigger the alarm every time a configuration change involving a VPC Network Customer/Internet Gateway is made.*

*In the Actions section, click the + Notification button, select State is ALARM from the Whenever this alarm dropdown menu and choose the AWS SNS topic name created at Step 1 from Send notification to.*

*In the Alarm Preview section, select 5 Minutes from the Period dropdown list and Sum from the Statistic list.*

*Review the CloudWatch alarm configuration details then click Create Alarm. Once created, the new alarm will be listed on the Alarms page.*

**Tags:**

## Control : Ensure a log metric filter and alarm exist for route table changes

**Status:** ⊗ Fail **Configuration Name:** Demo-conf **Group Name:** Monitoring1

**Severity :** MEDIUM

**Reason:**

1. A log metric alarm for 'Changes to Route Table' does not exist for Filter [ 313RouteTableChanges] in ( us-east-1 )

**Description :**

*Real-time monitoring of API calls can be achieved by directing CloudTrail Logs to CloudWatch Logs and establishing corresponding metric filters and alarms. Routing tables are used to route network traffic between subnets and to network gateways. It is recommended that a metric filter and alarm be established for changes to route tables.*

**Remediation Steps:**

*Perform the following to ensure a log metric filter and alarm exist for route table changes*

*Note : Filter pattern for route table changes*

*"filterPattern": "[ ($.eventName = CreateRoute) || ($.eventName = CreateRouteTable) || ($.eventName = ReplaceRoute) || ($.eventName = ReplaceRouteTableAssociation) || ($.eventName = DeleteRouteTable) || ($.eventName = DeleteRoute) || ($.eventName = DisassociateRouteTable) ]"*

*Perform the following to setup the metric filter, alarm, SNS topic, and subscription:*

1. *Identify the log group name configured for use with CloudTrail*
2. *Note the <cloudtrail_log_group_name> value associated with CloudWatchLogsLogGroupArn :*
3. *Create a metric filter based on filter pattern provided which checks for route table changes and the <cloudtrail_log_group_name> taken from step 2.*
   *Note : You can choose your own metricName and metricNamespace strings. Using the same metricNamespace for all Foundations Benchmark metrics will group them together.*
4. *Create an SNS topic that the alarm will notify*
   *Note : you can re-use the same topic for all monitoring alarms.*
5. *Create an SNS subscription to the topic created in step 4*
   *Note : you can re-use the same SNS subscription for all monitoring alarms.*
6. *Create an alarm that is associated with the CloudWatch Logs Metric Filter created in step 3 and an SNS topic created in step 4*
   *Note : set the period and threshold to values that fit your organization.*

**Tags:**

## Control : Ensure a log metric filter and alarm exist for VPC changes

**Status:** ⊗ Fail **Configuration Name:** Demo-conf **Group Name:** Monitoring1

**Severity :** MEDIUM

**Reason:**

1. A log metric alarm for 'VPC Changes' does not exist for Filter [ 314VPCChanges ] in ( us-east-1 )

**Description :**

*Real-time monitoring of API calls can be achieved by directing CloudTrail Logs to CloudWatch Logs and establishing corresponding metric filters and alarms. It is possible to have more than 1 VPC within an account, in addition it is also possible to create a peer connection between 2 VPCs enabling network traffic to route between VPCs. It is recommended that a metric filter and alarm be established for changes made to VPCs.*

**Remediation Steps:**

*Perform the following to ensure a log metric filter and alarm exist for VPC changes*

*Note : Filter pattern for VPC changes*

*"filterPattern": "[ ($.eventName = CreateVpc) || ($.eventName = DeleteVpc) || ($.eventName = ModifyVpcAttribute) || ($.eventName = AcceptVpcPeeringConnection) || ($.eventName = CreateVpcPeeringConnection) || ($.eventName = DeleteVpcPeeringConnection) || ($.eventName = RejectVpcPeeringConnection) || ($.eventName = AttachClassicLinkVpc) || ($.eventName = DetachClassicLinkVpc) || ($.eventName = DisableVpcClassicLink) || ($.eventName = EnableVpcClassicLink) ]"*

*Perform the following to setup the metric filter, alarm, SNS topic, and subscription:*

1. *Identify the log group name configured for use with CloudTrail*
2. *Note the <cloudtrail_log_group_name> value associated with CloudWatchLogsLogGroupArn :*
3. *Create a metric filter based on filter pattern provided which checks for VPC changes and the <cloudtrail_log_group_name> taken from step 2.*
   *Note : You can choose your own metricName and metricNamespace strings. Using the same metricNamespace for all Foundations Benchmark metrics will group them together.*
4. *Create an SNS topic that the alarm will notify*
   *Note : you can re-use the same topic for all monitoring alarms.*
5. *Create an SNS subscription to the topic created in step 4*
   *Note : you can re-use the same SNS subscription for all monitoring alarms.*
6. *Create an alarm that is associated with the CloudWatch Logs Metric Filter created in step 3 and an SNS topic created in step 4*
   *Note : set the period and threshold to values that fit your organization.*

**Tags:**

## Control : Ensure a log metric filter and alarm exist for Management Console sign-in without MFA

**Status:** ⊗ Fail **Configuration Name:** Demo-conf **Group Name:** Monitoring1

**Severity :** MEDIUM

**Reason:**
1. A log metric alarm for 'Management Console Sign-In without MFA' does not exist for Filter [ 32MgmtConsoleNoMFA ] in ( us-east-1 )

**Description :**
*Real-time monitoring of API calls can be achieved by directing CloudTrail Logs to CloudWatch Logs and establishing corresponding metric filters and alarms. It is recommended that a metric filter and alarm be established for console logins that are not protected by multi-factor authentication (MFA).*

**Remediation Steps:**
*Perform the following to ensure a log metric filter and alarm exist for Management Console sign-in without MFA :*

*Note : Filter pattern for Management Console sign-in without MFA*

*"filterPattern": "[ ($.eventName = "ConsoleLogin") && ($.additionalEventData.MFAUsed != "Yes") ]"*

*Perform the following to setup the metric filter, alarm, SNS topic, and subscription:*

1. *Identify the log group name configured for use with CloudTrail*
2. *Note the <cloudtrail_log_group_name> value associated with CloudWatchLogsLogGroupArn :*
3. *Create a metric filter based on filter pattern provided which checks for AWS Management Console sign-in without MFA and the <cloudtrail_log_group_name> taken from step 2.*
   *Note : You can choose your own metricName and metricNamespace strings. Using the same metricNamespace for all Foundations Benchmark metrics will group them together.*
4. *Create an SNS topic that the alarm will notify*
   *Note : you can re-use the same topic for all monitoring alarms.*
5. *Create an SNS subscription to the topic created in step 4*
   *Note : you can re-use the same SNS subscription for all monitoring alarms.*
6. *Create an alarm that is associated with the CloudWatch Logs Metric Filter created in step 3 and an SNS topic created in step 4*
   *Note : set the period and threshold to values that fit your organization.*

*References:*

1. *http://docs.aws.amazon.com/AmazonCloudWatch/latest/DeveloperGuide/viewing_metrics_with_cloudwatch.html*

---

**Tags:**

## Control : Ensure a log metric filter and alarm exist for usage of root account

**Status:** ⊗ Fail **Configuration Name:** Demo-conf **Group Name:** Monitoring1

**Severity :** MEDIUM

**Reason:**
1. A log metric alarm for 'Usage of root account' does not exist for Filter [ 33UseOfRootAcct ] in ( us-east-1 )

**Description :**
*Real-time monitoring of API calls can be achieved by directing CloudTrail Logs to CloudWatch Logs and establishing corresponding metric filters and alarms. It is recommended that a metric filter and alarm be established for root login attempts.*

**Remediation Steps:**
*Perform the following to ensure a log metric filter and alarm exist for usage of root account*

*Note : Filter pattern for for usage of "root" account*

*"filterPattern": "[ $.userIdentity.type = \"Root\" && $.userIdentity.invokedBy NOT EXISTS && $.eventType != \"AwsServiceEvent\" ] "*

*Perform the following to setup the metric filter, alarm, SNS topic, and subscription:*

1. *Identify the log group name configured for use with CloudTrail*
2. *Note the <cloudtrail_log_group_name> value associated with CloudWatchLogsLogGroupArn :*
3. *Create a metric filter based on filter pattern provided which checks for "Root" account usage and the <cloudtrail_log_group_name> taken from step 2*
   *Note : You can choose your own metricName and metricNamespace strings. Using the same metricNamespace for all Foundations Benchmark metrics will group them together.*
4. *Create an SNS topic that the alarm will notify*
   *Note : you can re-use the same topic for all monitoring alarms.*
5. *Create an SNS subscription to the topic created in step 4*
   *Note : you can re-use the same SNS subscription for all monitoring alarms.*
6. *Create an alarm that is associated with the CloudWatch Logs Metric Filter created in step 3 and an SNS topic created in step 4*
   *Note : set the period and threshold to values that fit your organization.*

*References:*
*1. CCE-79188-9*
*2. CIS CSC v6.0 #4.6, #5.1, #5.5*

---

**Tags:**

## Control : Ensure a log metric filter and alarm exist for IAM policy changes

**Status:** ⊗ Fail **Configuration Name:** Demo-conf **Group Name:** Monitoring1

**Severity :** MEDIUM

**Reason:**
1. A log metric alarm for 'IAM Policy Changes' does not exist for Filter [ 34IAMPolicyChanges ] in ( us-east-1 )

**Description :**
*Real-time monitoring of API calls can be achieved by directing CloudTrail Logs to CloudWatch Logs and establishing corresponding metric filters and alarms. It is recommended that a metric filter and alarm be established changes made to Identity and Access Management (IAM) policies.*

**Remediation Steps:**
*Perform the following to ensure a log metric filter and alarm exist for IAM policy changes*

*Note : Filter pattern for IAM policy changes*

*"filterPattern": "{($.eventName=DeleteGroupPolicy)||($.eventName=DeleteRolePolicy)||($.eventName=DeleteUserPolicy)||($.eventName=PutGroupPolicy)||($.eventName=PutRolePolicy)||($.eventName=PutUserPolicy)||($.eventName=CreatePolicy)||($.eventName=DeletePolicy)||($.eventName=CreatePolicyVersion)||($.eventName=DeletePolicyVersion)||($.eventName=AttachRolePolicy)||($.eventName=DetachRolePolicy)||($.eventName=AttachUserPolicy)||($.eventName=DetachUs*

*($.eventName=AttachGroupPolicy)||($.eventName=DetachGroupPolicy)) }*

*Perform the following to setup the metric filter, alarm, SNS topic, and subscription:*

1. *Identify the log group name configured for use with CloudTrail*
2. *Note the <cloudtrail_log_group_name> value associated with CloudWatchLogsLogGroupArn :*
3. *Create a metric filter based on filter pattern provided which checks for IAM Policy changes and the <cloudtrail_log_group_name> taken from step 2.*
   *Note : You can choose your own metricName and metricNamespace strings. Using the same metricNamespace for all Foundations Benchmark metrics will group them together.*
4. *Create an SNS topic that the alarm will notify*
   *Note : you can re-use the same topic for all monitoring alarms.*
5. *Create an SNS subscription to the topic created in step 4*
   *Note : you can re-use the same SNS subscription for all monitoring alarms.*
6. *Create an alarm that is associated with the CloudWatch Logs Metric Filter created in step 3 and an SNS topic created in step 4*
   *Note : set the period and threshold to values that fit your organization.*

---

**Tags:**

## Control : Ensure a log metric filter and alarm exist for CloudTrail configuration changes

**Status:** ⊗ Fail **Configuration Name:** Demo-conf **Group Name:** Monitoring1

**Severity :** MEDIUM

**Reason:**

1. A log metric alarm for 'Cloudtrail configuration Changes' does not exist for Filter [ 35CloudTrailConfigChanges ] in ( us-east-1 )

**Description :**

*Real-time monitoring of API calls can be achieved by directing CloudTrail Logs to CloudWatch Logs and establishing corresponding metric filters and alarms. It is recommended that a metric filter and alarm be established for detecting changes to CloudTrail's configurations.*

**Remediation Steps:**

*Perform the following to ensure a log metric filter and alarm exist for CloudTrail configuration changes*

*Note : Filter pattern for CloudTrail configuration changes*

*"filterPattern": "[ ($.eventName = CreateTrail) || ($.eventName = UpdateTrail) || ($.eventName = DeleteTrail) || ($.eventName = StartLogging) || ($.eventName = StopLogging) ]"*

*Perform the following to setup the metric filter, alarm, SNS topic, and subscription:*

1. *Identify the log group name configured for use with CloudTrail*
2. *Note the <cloudtrail_log_group_name> value associated with CloudWatchLogsLogGroupArn :*
3. *Create a metric filter based on filter pattern provided which checks for Cloudtrail configuration changes and the <cloudtrail_log_group_name> taken from step 2.*
   *Note : You can choose your own metricName and metricNamespace strings. Using the same metricNamespace for all Foundations Benchmark metrics will group them together.*
4. *Create an SNS topic that the alarm will notify*
   *Note : you can re-use the same topic for all monitoring alarms.*
5. *Create an SNS subscription to the topic created in step 4*
   *Note : you can re-use the same SNS subscription for all monitoring alarms.*
6. *Create an alarm that is associated with the CloudWatch Logs Metric Filter created in step 3 and an SNS topic created in step 4*
   *Note : set the period and threshold to values that fit your organization.*

---

**Tags:**

## Control : Ensure a log metric filter and alarm exist for AWS Management Console authentication failures

**Status:** ⊗ Fail **Configuration Name:** Demo-conf **Group Name:** Monitoring1

**Severity :** MEDIUM

**Reason:**

1. A log metric alarm for 'Console Authentication Failure' does not exist for Filter [ 36ConsoleAuthFailures] in ( us-east-1 )

**Description :**

*Real-time monitoring of API calls can be achieved by directing CloudTrail Logs to CloudWatch Logs and establishing corresponding metric filters and alarms. It is recommended that a metric filter and alarm be established for failed console authentication attempts.*

**Remediation Steps:**

*Perform the following to ensure a log metric filter and alarm exist for AWS Management Console authentication failures*

*Note : Filter pattern for AWS Management Console authentication failures*

*"filterPattern": "[ ($.eventName = ConsoleLogin) && ($.errorMessage = \"Failed authentication\") ]"*

*Perform the following to setup the metric filter, alarm, SNS topic, and subscription:*

1. *Identify the log group name configured for use with CloudTrail*
2. *Note the <cloudtrail_log_group_name> value associated with CloudWatchLogsLogGroupArn :*
3. *Create a metric filter based on filter pattern provided which checks for AWS Management Console authentication failures and the <cloudtrail_log_group_name> taken from step 2.*
   *Note : You can choose your own metricName and metricNamespace strings. Using the same metricNamespace for all Foundations Benchmark metrics will group them together.*
4. *Create an SNS topic that the alarm will notify*
   *Note : you can re-use the same topic for all monitoring alarms.*
5. *Create an SNS subscription to the topic created in step 4*
   *Note : you can re-use the same SNS subscription for all monitoring alarms.*
6. *Create an alarm that is associated with the CloudWatch Logs Metric Filter created in step 3 and an SNS topic created in step 4*
   *Note : set the period and threshold to values that fit your organization.*

---

**Tags:**

## Control : Ensure a log metric filter and alarm exist for disabling or scheduled deletion of customer created CMKs

**Status:** ⊗ Fail **Configuration Name:** Demo-conf **Group Name:** Monitoring1

**Severity :** MEDIUM

**Reason:**

1. A log metric alarm for 'Disabling or Scheduled Deletion of Customer Created CMKs' does not exist for Filter [ 37DisableDeleteCMK ] in ( us-east-1 )

**Description :**

*Real-time monitoring of API calls can be achieved by directing CloudTrail Logs to CloudWatch Logs and establishing corresponding metric filters and alarms. It is recommended that a metric filter and alarm be established for customer created CMKs which have changed state to disabled or scheduled deletion.*

**Remediation Steps:**

*Perform the following to ensure a log metric filter and alarm exist for disabling or scheduled deletion of customer created CMKs*

*Note : Filter pattern for disabling or scheduled deletion of customer created CMKs*

*"filterPattern": "[($.eventSource = kms.amazonaws.com) && (($.eventName=DisableKey)|||($.eventName=ScheduleKeyDeletion))] }"*

*Perform the following to setup the metric filter, alarm, SNS topic, and subscription:*

1. *Identify the log group name configured for use with CloudTrail*
2. *Note the <cloudtrail_log_group_name> value associated with CloudWatchLogsLogGroupArn*
3. *Create a metric filter based on filter pattern provided which checks for disabled or scheduled for deletion CMK's and the <cloudtrail_log_group_name> taken from step 2.*
   *Note : You can choose your own metricName and metricNamespace strings. Using the same metricNamespace for all Foundations Benchmark metrics will group them together.*
4. *Create an SNS topic that the alarm will notify*
   *Note : you can re-use the same topic for all monitoring alarms.*
5. *Create an SNS subscription to the topic created in step 4*
   *Note : you can re-use the same SNS subscription for all monitoring alarms.*
6. *Create an alarm that is associated with the CloudWatch Logs Metric Filter created in step 3 and an SNS topic created in step 4*
   *Note : set the period and threshold to values that fit your organization.*

---

**Tags:**

## Control : Ensure a log metric filter and alarm exist for S3 bucket policy changes

**Status:** ⊗ Fail **Configuration Name:** Demo-conf **Group Name:** Monitoring1

**Severity :** MEDIUM

**Reason:**

1. A log metric alarm for 'S3 Bucket Policy Changes' does not exist for Filter(s) [ 38S3BucketPolicyChanges ] in ( us-east-1 )

**Description :**

*Real-time monitoring of API calls can be achieved by directing CloudTrail Logs to CloudWatch Logs and establishing corresponding metric filters and alarms. It is recommended that a metric filter and alarm be established for changes to S3 bucket policies.*

**Remediation Steps:**

*Perform the following to ensure a log metric filter and alarm exist for S3 bucket policy changes*

*Note : Filter pattern for S3 bucket policy changes*

*"filterPattern": "[ ($.eventSource = s3.amazonaws.com) && (($.eventName = PutBucketAcl) || ($.eventName = PutBucketPolicy) || ($.eventName = PutBucketCors) || ($.eventName = PutBucketLifecycle) || ($.eventName = PutBucketReplication) || ($.eventName = DeleteBucketPolicy) || ($.eventName = DeleteBucketCors) || ($.eventName = DeleteBucketLifecycle) || ($.eventName = DeleteBucketReplication)) ]"*

*Perform the following to setup the metric filter, alarm, SNS topic, and subscription:*

1. *Identify the log group name configured for use with CloudTrail*
2. *Note the <cloudtrail_log_group_name> value associated with CloudWatchLogsLogGroupArn :*
3. *Create a metric filter based on filter pattern provided which checks for S3 Bucket Policy changes and the <cloudtrail_log_group_name> taken from step 2.*
   *Note : You can choose your own metricName and metricNamespace strings. Using the same metricNamespace for all Foundations Benchmark metrics will group them together.*
4. *Create an SNS topic that the alarm will notify*
   *Note : you can re-use the same topic for all monitoring alarms.*
5. *Create an SNS subscription to the topic created in step 4*
   *Note : you can re-use the same SNS subscription for all monitoring alarms.*
6. *Create an alarm that is associated with the CloudWatch Logs Metric Filter created in step 3 and an SNS topic created in step 4*
   *Note : set the period and threshold to values that fit your organization.*

---

**Tags:**

## Control : Ensure a log metric filter and alarm exist for AWS Config configuration changes

**Status:** ⊗ Fail **Configuration Name:** Demo-conf **Group Name:** Monitoring1

**Severity :** MEDIUM

**Reason:**

1. A log metric alarm for 'AWS Config Changes' does not exist for Filter [39AWSConfigChanges] in ( us-east-1 )

**Description :**

*Real-time monitoring of API calls can be achieved by directing CloudTrail Logs to CloudWatch Logs and establishing corresponding metric filters and alarms. It is recommended that a metric filter and alarm be established for detecting changes to AWS Config configuration.*

**Remediation Steps:**

*Perform the following to ensure a log metric filter and alarm exist for AWS Config configuration changes*

*Note : Filter pattern for AWS Config configuration changes*

*"filterPattern": "{($.eventSource = config.amazonaws.com) && ($.eventName=StopConfigurationRecorder)||($.eventName=DeleteDeliveryChannel)||($.eventName=PutDeliveryChannel)||($.eventName=PutConfigurationRecorder)}"*

*Perform the following to setup the metric filter, alarm, SNS topic, and subscription:*

1. *Identify the log group name configured for use with CloudTrail*
2. *Note the <cloudtrail_log_group_name> value associated with CloudWatchLogsLogGroupArn :*
3. *Create a metric filter based on filter pattern provided which checks for AWS Config changes and the <cloudtrail_log_group_name> taken from step 2.*
   *Note : You can choose your own metricName and metricNamespace strings. Using the same metricNamespace for all Foundations Benchmark metrics will group them together.*
4. *Create an SNS topic that the alarm will notify*
   *Note : you can re-use the same topic for all monitoring alarms.*
5. *Create an SNS subscription to the topic created in step 4*
   *Note : you can re-use the same SNS subscription for all monitoring alarms.*
6. *Create an alarm that is associated with the CloudWatch Logs Metric Filter created in step 3 and an SNS topic created in step 4*
   *Note : set the period and threshold to values that fit your organization.*