

CIS Oracle Cloud Infrastructure Foundations Benchmark

v1.0.0 - 08-06-2020

Terms of Use

Please see the below link for our current terms of use:

<https://www.cisecurity.org/cis-securesuite/cis-securesuite-membership-terms-of-use/>

Table of Contents

| | |
|--|----|
| Terms of Use | 1 |
| Overview | 5 |
| Intended Audience | 5 |
| Consensus Guidance..... | 5 |
| Typographical Conventions | 6 |
| Assessment Status..... | 6 |
| Profile Definitions | 7 |
| Acknowledgements | 8 |
| Recommendations | 9 |
| 1 Identity and Access Management..... | 9 |
| 1.1 Ensure service level admins are created to manage resources of particular service (Manual)..... | 10 |
| 1.2 Ensure permissions on all resources are given only to the tenancy administrator group (Manual)..... | 13 |
| 1.3 Ensure IAM administrators cannot update tenancy Administrators group (Manual) | 15 |
| 1.4 Ensure IAM password policy requires minimum length of 14 or greater (Manual) | 17 |
| 1.5 Ensure IAM password policy requires at least one uppercase letter (Manual) | 19 |
| 1.6 Ensure IAM password policy requires at least one lowercase letter (Manual) | 21 |
| 1.7 Ensure IAM password policy requires at least one symbol (Manual) | 23 |
| 1.8 Ensure IAM password policy requires at least one number (Manual) | 25 |
| 1.9 Ensure IAM password policy expires passwords within 90 days or less (Manual) | 27 |
| 1.10 Ensure IAM password policy prevents password reuse (Manual)..... | 28 |
| 1.11 Ensure MFA is enabled for all users with a console password (Automated) | 29 |
| 1.12 Ensure user API keys rotate within 90 days or less (Automated) | 31 |
| 1.13 Ensure API keys are not created for tenancy administrator users (Automated)..... | 33 |

| | |
|---|----|
| 2 Networking | 35 |
| 2.1 Ensure no security lists allow ingress from 0.0.0.0/0 to port 22 (Automated) | 36 |
| 2.2 Ensure no security lists allow ingress from 0.0.0.0/0 to port 3389 (Automated) | 39 |
| 2.3 Ensure no network security groups allow ingress from 0.0.0.0/0 to port 22 (Manual) | 42 |
| 2.4 Ensure no network security groups allow ingress from 0.0.0.0/0 to port 3389 (Manual) | 44 |
| 2.5 Ensure the default security list of every VCN restricts all traffic except ICMP (Automated) | 46 |
| 3 Logging and Monitoring | 48 |
| 3.1 Ensure audit log retention period is set to 365 days (Automated) | 49 |
| 3.2 Ensure default tags are used on resources (Manual) | 51 |
| 3.3 Create at least one notification topic and subscription to receive monitoring alerts (Manual) | 54 |
| 3.4 Ensure a notification is configured for Identity Provider changes (Manual) .. | 57 |
| 3.5 Ensure a notification is configured for IdP group mapping changes (Manual) | 61 |
| 3.6 Ensure a notification is configured for IAM group changes (Manual) | 64 |
| 3.7 Ensure a notification is configured for IAM policy changes (Manual) | 67 |
| 3.8 Ensure a notification is configured for user changes (Manual) | 70 |
| 3.9 Ensure a notification is configured for VCN changes (Manual) | 74 |
| 3.10 Ensure a notification is configured for changes to route tables (Manual) | 77 |
| 3.11 Ensure a notification is configured for security list changes (Manual) | 80 |
| 3.12 Ensure a notification is configured for network security group changes (Manual) | 83 |
| 3.13 Ensure a notification is configured for changes to network gateways (Manual) | 86 |
| Appendix: Summary Table | 91 |
| Appendix: Change History | 93 |

Overview

This document, CIS Oracle Cloud Infrastructure Foundations Benchmark, provides prescriptive guidance for establishing a secure baseline configuration for the Oracle Cloud Infrastructure environment. The scope of this benchmark is to establish a base level of security for anyone utilizing the included Oracle Cloud Infrastructure services. The benchmark is, however, not an exhaustive list of all possible security configurations and architecture. You should take the benchmark as a starting point and do the required site-specific tailoring wherever needed and when it is prudent to do so.

To obtain the latest version of this guide, please visit <https://www.cisecurity.org/cis-benchmarks/>. If you have questions, comments, or have identified ways to improve this guide, please write us at benchmarkinfo@cisecurity.org.

Intended Audience

This document is intended for system and application administrators, security specialists, auditors, help desk, platform deployment, and/or DevOps personnel who plan to develop, deploy, assess, or secure solutions in the Oracle Cloud Infrastructure.

Consensus Guidance

This benchmark was created using a consensus review process comprised of subject matter experts. Consensus participants provide perspective from a diverse set of backgrounds including consulting, software development, audit and compliance, security research, operations, government, and legal.

Each CIS benchmark undergoes two phases of consensus review. The first phase occurs during initial benchmark development. During this phase, subject matter experts convene to discuss, create, and test working drafts of the benchmark. This discussion occurs until consensus has been reached on benchmark recommendations. The second phase begins after the benchmark has been published. During this phase, all feedback provided by the Internet community is reviewed by the consensus team for incorporation in the benchmark. If you are interested in participating in the consensus process, please visit <https://workbench.cisecurity.org/>.

Typographical Conventions

The following typographical conventions are used throughout this guide:

| Convention | Meaning |
|--|---|
| <code>Stylized Monospace font</code> | Used for blocks of code, command, and script examples. Text should be interpreted exactly as presented. |
| Monospace font | Used for inline code, commands, or examples. Text should be interpreted exactly as presented. |
| <i><italic font in brackets></i> | Italic texts set in angle brackets denote a variable requiring substitution for a real value. |
| <i>Italic font</i> | Used to denote the title of a book, article, or other publication. |
| Note | Additional information or caveats |

Assessment Status

An assessment status is included for every recommendation. The assessment status indicates whether the given recommendation can be automated or requires manual steps to implement. Both statuses are equally important and are determined and supported as defined below:

Automated

Represents recommendations for which assessment of a technical control can be fully automated and validated to a pass/fail state. Recommendations will include the necessary information to implement automation.

Manual

Represents recommendations for which assessment of a technical control cannot be fully automated and requires all or some manual steps to validate that the configured state is set as expected. The expected state can vary depending on the environment.

Profile Definitions

The following configuration profiles are defined by this Benchmark:

- **Level 1**

Items in this profile intend to:

- be practical and prudent;
- provide a clear security benefit; and
- not inhibit the utility of the technology beyond acceptable means.

- **Level 2**

This profile extends the "Level 1" profile. Items in this profile exhibit one or more of the following characteristics:

- are intended for environments or use cases where security is paramount
- acts as defense in depth measure
- may negatively inhibit the utility or performance of the technology.

Acknowledgements

This benchmark exemplifies the great things a community of users, vendors, and subject matter experts can accomplish through consensus collaboration. The CIS community thanks the entire consensus team with special recognition to the following individuals who contributed greatly to the creation of this guide:

Contributor

Mike Wicks

Editor

Vani Naik

Andy Tael

Richard Jacobs

Manasi Vaishampayan

Christopher Johnson

Clinton Edwards

Logan Kleier

Sanjay Basu

Ryan Cronk

Recommendations

1 Identity and Access Management

This section contains recommendations for configuring identity and access management related options.

1.1 Ensure service level admins are created to manage resources of particular service (Manual)

Profile Applicability:

- Level 1

Description:

To apply least-privilege security principle, one can create service-level administrators in corresponding groups and assigning specific users to each service-level administrative group in a tenancy. This limits administrative access in a tenancy.

It means service-level administrators can only manage resources of a specific service.

Example policies for global/tenant level service-administrators

```
Allow group VolumeAdmins to manage volume-family in tenancy
Allow group ComputeAdmins to manage instance-family in tenancy
Allow group NetworkAdmins to manage virtual-network-family in tenancy
```

Organizations have various ways of defining service-administrators. Some may prefer creating service administrators at a tenant level and some per department or per project or even per application environment (dev/test/production etc.). Either approach works so long as the policies are written to limit access given to the service-administrators.

Example policies for compartment level service-administrators

```
Allow group NonProdComputeAdmins to manage instance-family in compartment dev
Allow group ProdComputeAdmins to manage instance-family in compartment
production
Allow group A-Admins to manage instance-family in compartment Project-A
Allow group A-Admins to manage volume-family in compartment Project-A
```

Rationale:

Creating service-level administrators helps in tightly controlling access to Oracle Cloud Infrastructure (OCI) services to implement the least-privileged security principle.

Audit:

From CLI:

1. [Set up OCI CLI](#) with an IAM administrator user who has read access to IAM resources such as groups and policies.
2. Run OCI CLI command providing the root_compartment_OCID
Get the list of groups in a tenancy

```
oci iam group list --compartment-id <root_compartment_OCID> | grep name
```

3. Ensure distinct administrative groups are created as per your organization's definition of service-administrators.
4. Verify the appropriate policies are created for the service-administrators groups to have the right access to the corresponding services. Retrieve the policy statements scoped at the tenancy level and/or per compartment.

```
oci iam policy list --compartment-id <root_compartment_OCID> | grep "in  
tenancy"
```

```
oci iam policy list --compartment-id <root_compartment_OCID> | grep "in  
compartment"
```

The --compartment-id parameter can be changed to a child compartment to get policies associated with child compartments.

```
oci iam policy list --compartment-id <child_compartment_OCID> | grep "in  
compartment"
```

Verify the results to ensure the right policies are created for service-administrators to have the necessary access.

Remediation:

Refer to the [policy syntax document](#) and create new policies if the audit results indicate that the required policies are missing.

This can be done via OCI console or OCI CLI/SDK or API.

Creating a new policy:

From CLI:

```
oci iam policy create [OPTIONS]
```

Creates a new policy in the specified compartment (either the tenancy or another of your compartments). If you're new to policies, see

[Getting Started with Policies](<https://docs.cloud.oracle.com/Content/Identity/Concepts/policygetstarted.htm>)

You must specify a name for the policy, which must be unique across all policies in your tenancy and cannot be changed.

You must also specify a description for the policy (although it can be an empty string). It does not have to be unique, and you can change it anytime with UpdatePolicy.

You must specify one or more policy statements in the statements array.

For information about writing policies, see How [Policies Work](<https://docs.cloud.oracle.com/Content/Identity/Concepts/policies.htm>) and [Common Policies](<https://docs.cloud.oracle.com/Content/Identity/Concepts/commonpolicies.htm>).

CIS Controls:

Version 7

4.3 Ensure the Use of Dedicated Administrative Accounts

Ensure that all users with administrative account access use a dedicated or secondary account for elevated activities. This account should only be used for administrative activities and not internet browsing, email, or similar activities.

1.2 Ensure permissions on all resources are given only to the tenancy administrator group (Manual)

Profile Applicability:

- Level 1

Description:

There is a built-in OCI IAM policy enabling the Administrators group to perform any action within a tenancy. In the OCI IAM console, this policy reads:

```
Allow group Administrators to manage all-resources in tenancy
```

Administrators create more users, groups, and policies to provide appropriate access to other groups.

Administrators should not allow any-other-group full access to the tenancy by writing a policy like this -

```
Allow group any-other-group to manage all-resources in tenancy
```

The access should be narrowed down to ensure the least-privileged principle is applied.

Rationale:

Permission to manage all resources in a tenancy should be limited to a small number of users in the `Administrators` group for break-glass situations and to set up users/groups/policies when a tenancy is created.

No group other than `Administrators` in a tenancy should need access to all resources in a tenancy, as this violates the enforcement of the least privilege principle.

Audit:

From CLI:

1. Run OCI CLI command providing the root compartment OCID to get the list of groups having access to manage all resources in your tenancy.

```
oci iam policy list --compartment-id <root_compartment_OCID> | grep -i "to manage all-resources in tenancy"
```

2. Verify the results to ensure only the `Administrators` group has access to manage all resources in tenancy.

"Allow group Administrators to manage all-resources in tenancy"

Remediation:

From Console:

1. Login to OCI console.
2. Go to `Identity -> Policies`, In the compartment dropdown, choose the root compartment. Open each policy to view the policy statements.
3. Remove any policy statement that allows any group other than `Administrators` or any service access to manage all resources in the tenancy.

The policies can also be updated via OCI CLI/SDK/API.

Note: You should generally **not** delete the policy that allows the `Administrators` group the ability to manage all resources in the tenancy.

CIS Controls:

Version 7

4 Controlled Use of Administrative Privileges

Controlled Use of Administrative Privileges

1.3 Ensure IAM administrators cannot update tenancy Administrators group (Manual)

Profile Applicability:

- Level 1

Description:

Tenancy administrators can create more users, groups, and policies to provide other service administrators access to OCI resources.

For example, an IAM administrator will need to have access to manage resources like compartments, users, groups, dynamic-groups, policies, identity-providers, tenancy tag-namespaces, tag-definitions in the tenancy.

The policy that gives IAM-Administrators or any other group full access to 'groups' resources should not allow access to the tenancy 'Administrators' group.

The policy statements would look like -

```
Allow group IAMAdmins to inspect users in tenancy

Allow group IAMAdmins to use users in tenancy where target.group.name !=
'Administrators'

Allow group IAMAdmins to inspect groups in tenancy

Allow group IAMAdmins to use groups in tenancy where target.group.name !=
'Administrators'
```

Note: You must include separate statements for 'inspect' access, because the target.group.name variable is not used by the ListUsers and ListGroups operations

Rationale:

These policy statements ensure that no other group can manage tenancy administrator users or the membership to the 'Administrators' group thereby gain or remove tenancy administrator access.

Audit:

From CLI:

1. Run the following OCI CLI commands providing the root_compartment_OCID


```
oci iam policy list --compartment-id <root_compartment_OCID> | grep -i " to  
use users in tenancy"  
oci iam policy list --compartment-id <root_compartment_OCID> | grep -i " to  
use groups in tenancy"
```

2. Verify the results to ensure that the policy statements that grant access to use or manage users or groups in the tenancy have a condition that excludes access to Administrators group or to users in the Administrators group.

Remediation:

From Console:

1. Login to OCI Console.
2. Select Identity from Services Menu.
3. Select Policies from Identity Menu.
4. Click on an individual policy under the Name heading.
5. Ensure Policy statements look like this -

```
Allow group IAMAdmins to use users in tenancy **where target.group.name !=  
'Administrators'  
Allow group IAMAdmins to use groups in tenancy **where target.group.name !=  
'Administrators'
```

CIS Controls:

Version 7

4 Controlled Use of Administrative Privileges

Controlled Use of Administrative Privileges

14.6 Protect Information through Access Control Lists

Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

1.4 Ensure IAM password policy requires minimum length of 14 or greater (Manual)

Profile Applicability:

- Level 1

Description:

Password policies are used to enforce password complexity requirements. IAM password policies can be used to ensure password are at least a certain length. It is recommended the password policy require a minimum password length 14 characters.

Rationale:

Setting a password complexity policy increases account resiliency against brute force login attempts.

Audit:

OCI Native IAM

From Console:

1. Login to the OCI Console
2. Go to `Identity` in the Services menu.
3. Select `Authentication Settings` from the Identity menu.
4. Click `Edit` in the middle of the page.
5. Ensure the number `14` is entered into the box below the text: `MINIMUM PASSWORD LENGTH (IN CHARACTERS)`.

OCI Identity Cloud Service (IDCS)

If you use IDCS to login to OCI, verify the password policy settings in IDCS

1. Login to IDCS Admin Console
2. Expand the Navigation Drawer, click `Settings`, and then click `Password Policy`.
3. Verify the `Password length min size` setting.

Remediation:

OCI Native IAM

From Console:

1. Login to the OCI Console
2. Go to `Identity` in the Services menu.

3. Select `Authentication Settings` from the Identity menu.
4. Click `Edit` in the middle of the page.
5. Type the number `14` into the box below the text: `MINIMUM PASSWORD LENGTH (IN CHARACTERS)`.

OCI Identity Cloud Service (IDCS)

1. Login to IDCS Admin Console
2. Expand the Navigation Drawer, click `Settings`, and then click `Password Policy`.
3. Click on `Change Your Password Policy` button.
4. Update the `Password length min size` setting to `14` and Save.

CIS Controls:

Version 7

16 Account Monitoring and Control

Account Monitoring and Control

1.5 Ensure IAM password policy requires at least one uppercase letter (Manual)

Profile Applicability:

- Level 1

Description:

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

Rationale:

Setting a password complexity policy increases account resiliency against brute force login attempts.

Audit:

OCI Native IAM

From Console:

1. Login to the OCI Console
2. Go to **Identity** in the Services menu.
3. Select **Authentication Settings** from the Identity menu.
4. Click **Edit** in the middle of the page.
5. Ensure the checkbox is selected next to **MUST CONTAIN AT LEAST 1 UPPERCASE CHARACTER**.

From CLI:

Set up the OCI CLI with an IAM administrator user who has access to read IAM policies. Run OCI CLI command providing the root compartment OCID

```
oci iam authentication-policy get --compartment-id <root_compartment_OCID> --query 'data."password-policy"."is-uppercase-characters-required"'
```

Verify that the return value from the command is `true`.

OCI Identity Cloud Service (IDCS)

If you use IDCS to login to OCI, verify the password policy settings in IDCS

1. Login to IDCS Admin Console
2. Expand the Navigation Drawer, click **Settings**, and then click **Password Policy**.

3. Under `The password must contain these characters` section, ensure that the number given in `Uppercase min` setting is `> 0`.

Remediation:

OCI Native IAM

From Console:

1. Login to the OCI Console
2. Go to `Identity` in the `Services` menu.
3. Select `Authentication Settings` from the `Identity` menu.
4. Click `Edit` in the middle of the page.
5. Select checkbox next to `MUST CONTAIN AT LEAST 1 UPPERCASE CHARACTER`.
6. Click the `Save` button.

OCI Identity Cloud Service (IDCS)

If you use IDCS to login to OCI, verify the password policy settings in IDCS

1. Login to IDCS Admin Console
2. Expand the `Navigation Drawer`, click `Settings`, and then click `Password Policy`.
3. Click on `Change Your Password Policy` button.
4. Under "The password must contain these characters" section, update the number given in `Uppercase min` setting to 1. Click `Save`.

CIS Controls:

Version 7

16 Account Monitoring and Control

Account Monitoring and Control

1.6 Ensure IAM password policy requires at least one lowercase letter (Manual)

Profile Applicability:

- Level 1

Description:

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

Rationale:

Setting a password complexity policy increases account resiliency against brute force login attempts.

Audit:

OCI Native IAM

From Console:

Go to `Identity` in the Services menu.

Select `Authentication Settings` from the Identity menu.

Click `Edit` in the middle of the page.

Ensure checkbox is selected next to `MUST CONTAIN AT LEAST 1 LOWERCASE CHARACTER`.

From CLI:

Set up the OCI CLI with an IAM administrator user who has access to read IAM policies.

Run OCI CLI command providing the root compartment OCID

```
oci iam authentication-policy get --compartment-id <root_compartment_OCID> --query 'data."password-policy"."is-lowercase-characters-required"'
```

Verify that the return value from the command is true.

OCI Identity Cloud Service (IDCS)

If you use IDCS to login to OCI, verify the password policy settings in IDCS

1. Login to IDCS Admin Console
2. Expand the Navigation Drawer, click `Settings`, and then click `Password Policy`.
3. Under `The password must contain these characters` section, ensure that the number given in `Lowercase min` setting is `> 0`.

Remediation:

****OCI Native IAM ****

From Console:

1. Login to the OCI Console
2. Go to `Identity` in the Services menu.
3. Select `Authentication Settings` from the Identity menu.
4. Click `Edit` in the middle of the page.
5. Select checkbox next to `MUST CONTAIN AT LEAST 1 LOWERCASE CHARACTER`.
6. Click the `Save` button.

OCI Identity Cloud Service (IDCS)

If you use IDCS to login to OCI, verify the password policy settings in IDCS

1. Login to IDCS Admin Console
2. Expand the Navigation Drawer, click `Settings`, and then click `Password Policy`.
3. Click on `Change Your Password Policy` button.
4. Under `The password must contain these characters` section, update the number given in `Lowercase min` setting to 1. Click `Save`.

CIS Controls:

Version 7

16 Account Monitoring and Control

Account Monitoring and Control

1.7 Ensure IAM password policy requires at least one symbol (Manual)

Profile Applicability:

- Level 1

Description:

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

Rationale:

Setting a password complexity policy increases account resiliency against brute force login attempts.

Audit:

OCI Native IAM

From Console:

1. Login to the OCI Console
2. Go to `Identity` in the Services menu.
3. Select `Authentication Settings` from the Identity menu.
4. Click `Edit` in the middle of the page.
5. Ensure checkbox is selected next to `MUST CONTAIN AT LEAST 1 SPECIAL CHARACTER`.

OCI Identity Cloud Service (IDCS)

If you use IDCS to login to OCI, verify the password policy settings in IDCS

1. Login to IDCS Admin Console
2. Expand the Navigation Drawer, click `Settings`, and then click `Password Policy`.
3. Under `The password must contain these characters` section, ensure that the number given in `Special min` setting is `> 0`.

Remediation:

OCI Native IAM

From Console:

1. Login to the OCI Console
2. Go to `Identity` in the Services menu.
3. Select `Authentication Settings` from the Identity menu.

4. Click `Edit` in the middle of the page.
5. Select checkbox next to `MUST CONTAIN AT LEAST 1 SPECIAL CHARACTER`.
6. Click the `Save` button.

From CLI:

Set up the OCI CLI with an IAM administrator user who has access to read IAM policies.

Run OCI CLI command providing the root compartment OCID

```
oci iam authentication-policy get --compartment-id <root_compartment_OCID> --query 'data."password-policy"."is-special-characters-required"'
```

Verify that the return value from the command is `true`.

OCI Identity Cloud Service

If you use IDCS to login to OCI, verify the password policy settings in IDCS

1. Login to IDCS Admin Console
2. Expand the Navigation Drawer, click `Settings`, and then click `Password Policy`.
3. Click on `Change Your Password Policy` button.
4. Under `The password must contain these characters` section, update the number given in `Special min` setting to 1. Click `Save`.

CIS Controls:

Version 7

16 Account Monitoring and Control

Account Monitoring and Control

1.8 Ensure IAM password policy requires at least one number (Manual)

Profile Applicability:

- Level 1

Description:

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

Rationale:

Setting a password complexity policy increases account resiliency against brute force login attempts.

Audit:

OCI Native IAM

From Console:

1. Login to the OCI Console
2. Go to `Identity` in the Services menu.
3. Select `Authentication Settings` from the Identity menu.
4. Click `Edit` in the middle of the page.
5. Ensure the checkbox is selected next to `MUST CONTAIN AT LEAST 1 NUMERIC CHARACTER`.

OCI Identity Cloud Service

If you use IDCS to login to OCI, verify the password policy settings in IDCS

1. Login to IDCS Admin Console
2. Expand the Navigation Drawer, click `Settings`, and then click `Password Policy`.
3. Under `The password must contain these characters` section, ensure that the number given in `Numeric min` setting is `> 0`.

Remediation:

OCI Native IAM

From Console:

1. Login to the OCI Console
2. Go to `Identity` in the Services menu.

3. Select `Authentication Settings` from the `Identity` menu.
4. Click `Edit` in the middle of the page.
5. Select checkbox next to `MUST CONTAIN AT LEAST 1 NUMERIC CHARACTER`.
6. Click the `Save` button.

From CLI:

Set up the OCI CLI with an IAM administrator user who has access to read IAM policies.
Run OCI CLI command providing the root compartment OCID

```
oci iam authentication-policy get --compartment-id <root_compartment_OCID> --  
query 'data."password-policy"."is-numeric-characters-required"'
```

Verify that the return value from the command is `true`.

OCI Identity Cloud Service

If you use IDCS to login to OCI, verify the password policy settings in IDCS

1. Login to IDCS Admin Console
2. Expand the `Navigation Drawer`, click `Settings`, and then click `Password Policy`.
3. Click on `Change Your Password Policy` button.
4. Under `The password must contain these characters` section, update the number given in `Numeric min` setting to 1. Click `Save`.

CIS Controls:

Version 7

16 Account Monitoring and Control

Account Monitoring and Control

1.9 Ensure IAM password policy expires passwords within 90 days or less (Manual)

Profile Applicability:

- Level 1

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.

Rationale:

Reducing the life time of a password adds account resiliency. Passwords can be stolen by attackers in various ways and enforcing users to change their password periodically reduces that risk window.

Audit:

OCI Identity Cloud Service (IDCS)

1. Login to IDCS Admin Console
2. Expand the Navigation Drawer, click `Settings`, and then click `Password Policy`.
3. Ensure that the number of days configured in `Expires after` setting is `<=90`.

Remediation:

OCI Identity Cloud Service (IDCS)

1. Login to IDCS Admin Console
2. Expand the Navigation Drawer, click `Settings`, and then click `Password Policy`.
3. Click on `Change Your Password Policy` button.
4. Update the number of days configured in `Expires after` setting to 90.

CIS Controls:

Version 7

16 Account Monitoring and Control
Account Monitoring and Control

1.10 Ensure IAM password policy prevents password reuse (Manual)

Profile Applicability:

- Level 1

Description:

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

Rationale:

Enforcing password history ensures that passwords are not reused in a certain period of time. If a user is not allowed to use last 24 passwords, that window of time is greater. This helps maintain the effectiveness of password security.

Audit:

OCI Identity Cloud Service (IDCS)

1. Login to IDCS Admin Console
2. Expand the Navigation Drawer, click `Settings`, and then click `Password Policy`.
3. Ensure that the number of remembered passwords in `Previous passwords remembered` setting is set to 24.

Remediation:

OCI Identity Cloud Service (IDCS)

1. Login to IDCS Admin Console
2. Expand the Navigation Drawer, click `Settings`, and then click `Password Policy`.
3. Click on "Change Your Password Policy" button.
4. Update the number of remembered passwords in `Previous passwords remembered` setting to 24.

CIS Controls:

Version 7

16 Account Monitoring and Control
Account Monitoring and Control

1.11 Ensure MFA is enabled for all users with a console password (Automated)

Profile Applicability:

- Level 1

Description:

Multi-factor authentication is a method of authentication that requires the use of more than one factor to verify a user's identity.

With MFA enabled in the IAM service, when a user signs in to Oracle Cloud Infrastructure, they are prompted for their user name and password, which is the first factor (something that they know). The user is then prompted to provide a second verification code from a registered MFA device, which is the second factor (something that they have). The two factors work together, requiring an extra layer of security to verify the user's identity and complete the sign-in process.

OCI IAM supports two-factor authentication using a password (first factor) and a device that can generate a time-based one-time password (TOTP) (second factor).

See [OCI documentation](#) for more details.

Rationale:

Multi factor authentication adds an extra layer of security during the login process and makes it harder unauthorized users to gain access to OCI resources.

Audit:

From Console:

1. Login into OCI Console.
2. Select `Identity` from Services menu
3. Select `Users` from Identity menu.
4. Click on an individual user.
5. Ensure the word `Enabled` is next to `Multi-factor authentication`.

From CLI:

Set up the OCI CLI with an IAM administrator user who has access to read IAM policies. Run OCI CLI command providing the root compartment OCID

```
oci iam user list --query 'data[][\"id\", \"name\", \"is-mfa-activated\"]' --output table
```

Verify that the table column named Column2 has not values of `false`

Remediation:

Each user must enable MFA for themselves using a device they will have access to every time they sign in. An administrator cannot enable MFA for another user but can enforce MFA by identifying the list of non-complaint users, notifying them or disabling access by resetting password for non-complaint accounts.

Disabling access from Console:

1. Login into OCI Console.
2. Select `Identity` from Services menu
3. Select `Users` from Identity menu.
4. Click on each non-complaint user.
5. Click on `Create/Reset Password`.

From CLI:

```
oci iam user ui-password create-or-reset --user-id <OCID of the non-compliant user>
```

CIS Controls:

Version 7

4.5 Use Multifactor Authentication For All Administrative Access

Use multi-factor authentication and encrypted channels for all administrative account access.

16.3 Require Multi-factor Authentication

Require multi-factor authentication for all user accounts, on all systems, whether managed onsite or by a third-party provider.

1.12 Ensure user API keys rotate within 90 days or less (Automated)

Profile Applicability:

- Level 1

Description:

API keys are used by administrators, developers, services and scripts for accessing OCI APIs directly or via SDKs/OCI CLI to search, create, update or delete OCI resources.

The API key is an RSA key pair. The private key is used for signing the API requests and the public key is associated with a local or synchronized user's profile.

Rationale:

It is important to secure and rotate an API key every 90 days or less as it provides the same level of access that a user it is associated with has.

In addition to a security engineering best practice, this is also a compliance requirement. For example, PCI-DSS Section 3.6.4 states, "Verify that key-management procedures include a defined cryptoperiod for each key type in use and define a process for key changes at the end of the defined cryptoperiod(s)."

Audit:

OCI Native IAM

From Console:

1. Login to OCI Console.
2. Select **Identity** from the Services menu.
3. Select **Users** from the Identity menu.
4. Click on an individual user under the Name heading.
5. Click on **API Keys** in the lower left hand corner of the page.
6. Ensure the date of the API key under the **Created** column of the API Key is no more than 90 days old.

From CLI:

```
oci iam user api-key list --user-id <user_ocid> --query data[*].["time-created","fingerprint"]
```


Remediation:

OCI Native IAM

From Console:

1. Login to OCI Console.
2. Select `Identity` from the Services menu.
3. Select `Users` from the Identity menu.
4. Click on an individual user under the Name heading.
5. Click on `API Keys` in the lower left hand corner of the page.
6. Delete any API Keys with a date of 90 days or older under the `Created` column of the API Key table.

From CLI:

```
oci iam user api-key delete --user-id _<user_OCID>_ --fingerprint  
<fingerprint_of_the_key_to_be_deleted>
```

CIS Controls:

Version 7

16 Account Monitoring and Control

Account Monitoring and Control

1.13 Ensure API keys are not created for tenancy administrator users (Automated)

Profile Applicability:

- Level 1

Description:

Tenancy administrator users have full access to the organization's OCI tenancy. API keys associated with user accounts are used for invoking the OCI APIs via custom programs or clients like CLI/SDKs. The clients are typically used for performing day-to-day operations and should never require full tenancy access. Service-level administrative users with API keys should be used instead.

Rationale:

For performing day-to-day operations tenancy administrator access is not needed. Service-level administrative users with API keys should be used to apply privileged security principle.

Audit:

OCI Native IAM

From Console:

1. Login to OCI Console.
2. Verify user profile of each user who is member of the `Administrators` group directly or via federation group mapping.
3. Go to `Identity > Users` and click on each local or synchronized `Administrators` member profile.
4. Click on API Keys to verify if a user has an API key associated.

Remediation:

OCI Native IAM

From Console:

1. Login to OCI console
2. Select `Identity` from Services menu.
3. Select `Users` from Identity menu.
4. For each tenancy administrator user who has an API key, select `API Keys` from the menu in the lower left hand corner.
5. Delete any associated keys from the `API Keys` table.

CIS Controls:

Version 7

4 Controlled Use of Administrative Privileges

Controlled Use of Administrative Privileges

16 Account Monitoring and Control

Account Monitoring and Control

2 Networking

This section contains recommendations for configuring network security related options.

2.1 Ensure no security lists allow ingress from 0.0.0.0/0 to port 22 (Automated)

Profile Applicability:

- Level 1

Description:

Security lists provide stateful or stateless filtering of ingress/egress network traffic to OCI resources on a subnet level. It is recommended that no security group allows unrestricted ingress access to port 22.

Rationale:

Removing unfettered connectivity to remote console services, such as Secure Shell (SSH), reduces a server's exposure to risk.

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 network security group or security list.

Audit:

From Console:

1. Login into the OCI Console
2. Click in the search bar, top of the screen.
3. Type `Advanced Resource Query` and hit `enter`.
4. Click the `Advanced Resource Query` button in the upper right of the screen.
5. Enter the following query in the query box:

```
query SecurityList resources where
(IngressSecurityRules.source = '0.0.0.0/0' &&
IngressSecurityRules.protocol = 6 &&
IngressSecurityRules.tcpOptions.destinationPortRange.max = 22 &&
IngressSecurityRules.tcpOptions.destinationPortRange.min = 22)
```

6. Ensure query returns no results.

From CLI:

1. Execute the following command

```
oci search resource structured-search --query-text "query SecurityList
resources where
(IngressSecurityRules.source = '0.0.0.0/0' &&
IngressSecurityRules.protocol = 6 &&
IngressSecurityRules.tcpOptions.destinationPortRange.max = 22 &&
IngressSecurityRules.tcpOptions.destinationPortRange.min = 22)
"
```

2. Ensure query returns no results.

Remediation:

From Console:

1. Follow the audit procedure above.
2. For each security list in the returned results, click the security list name
3. Either edit the `ingress rule` to be more restrictive, delete the `ingress rule` or click on the `VCN` and terminate the `security list` as appropriate.

From CLI:

1. Follow the audit procedure.
2. For each of the `security lists` identified get the its details

```
oci network security-list get --security-list-id <security list id>
```

3. Then either:

- Update the `security list`, copy the `ingress-security-rules` element from the JSON returned by the above get call, edit it appropriately and use it in the following command

```
oci network security-list update --security-list-id <security-list-id> --
ingress-security-rules '<ingress security rules JSON>'
```

or

- Delete the security list

```
oci network security-list delete --security-list-id <security list id>
```

CIS Controls:

Version 7

9.2 Ensure Only Approved Ports, Protocols and Services Are Running

Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system.

2.2 Ensure no security lists allow ingress from 0.0.0.0/0 to port 3389 (Automated)

Profile Applicability:

- Level 1

Description:

Security lists provide stateful or stateless filtering of ingress/egress network traffic to OCI resources on a subnet level. It is recommended that no security group allows unrestricted ingress access to port 3389.

Rationale:

Removing unfettered connectivity to remote console services, such as Remote Desktop Protocol (RDP), reduces a server's exposure to risk.

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 network security group or security list.

Audit:

From Console:

1. Login into the OCI Console
2. Click in the search bar, top of the screen.
3. Type `Advanced Resource Query` and hit `enter`.
4. Click the `Advanced Resource Query` button in the upper right of the screen.
5. Enter the following query in the query box:

```
query SecurityList resources where
(IngressSecurityRules.source = '0.0.0.0/0' &&
IngressSecurityRules.protocol = 6 &&
IngressSecurityRules.tcpOptions.destinationPortRange.max = 3389 &&
IngressSecurityRules.tcpOptions.destinationPortRange.min = 3389)
```

6. Ensure query returns no results.

From CLI:

1. Execute the following command


```
oci search resource structured-search --query-text "query SecurityList
resources where
(IngressSecurityRules.source = '0.0.0.0/0' &&
IngressSecurityRules.protocol = 6 &&
IngressSecurityRules.tcpOptions.destinationPortRange.max = 3389 &&
IngressSecurityRules.tcpOptions.destinationPortRange.min = 3389)
"
```

2. Ensure query returns no results.

Remediation:

From Console:

1. Follow the audit procedure above.
2. For each security list in the returned results, click the security list name
3. Either edit the `ingress rule` to be more restrictive, delete the `ingress rule` or click on the `VCN` and terminate the `security list` as appropriate.

From CLI:

1. Follow the audit procedure.
2. For each of the `security lists` identified get the its details

```
oci network security-list get --security-list-id <security list id>
```

3. Then either:

- Update the `security list`, copy the `ingress-security-rules` element from the JSON returned by the above `get` call, edit it appropriately and use it in the following command

```
oci network security-list update --security-list-id <security-list-id> --
ingress-security-rules '<ingress security rules JSON>'
```

or

- Delete the security list

```
oci network security-list delete --security-list-id <security list id>
```

CIS Controls:

Version 7

9.2 Ensure Only Approved Ports, Protocols and Services Are Running

Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system.

2.3 Ensure no network security groups allow ingress from 0.0.0.0/0 to port 22 (Manual)

Profile Applicability:

- Level 1

Description:

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

Rationale:

Removing unfettered connectivity to remote console services, such as Secure Shell (SSH), reduces a server's exposure to risk.

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 network security group or security list.

Audit:

From CLI:

Issue the following command, it should not return anything.

```
for region in `oci iam region list | jq -r '.data[] | .name'`;
do
  for compid in `oci iam compartment list 2>/dev/null | jq -r '.data[] | .id'`;
  do
    for nsgid in `oci network nsg list --compartment-id $compid --
region $region --all 2>/dev/null | jq -r '.data[] | .id'`;
    do
      output=`oci network nsg rules list --nsg-id=$nsgid --all
2>/dev/null | jq -r '.data[] | select(.source == "0.0.0.0/0" and .direction
== "INGRESS" and ((."tcp-options"."destination-port-range".max >= 22 and
."tcp-options"."destination-port-range".min <= 22) or ."tcp-
options"."destination-port-range" == null))'`;
      if [ ! -z "$output" ]; then echo "NSGID=", $nsgid,
"Security Rules=", $output; fi
    done
  done
done
```

Remediation:

From CLI:

Using the details returned from the audit procedure either:

- Remove the security rules

```
oci network nsg rules remove --nsg-id=<NSGID from audit output>
```

or

- Update the security rules

```
oci network nsg rules update --nsg-id=<NSGID from audit output> --security-rules='[<updated security-rules JSON (without isValid and TimrCreated fields)>]'
```

eg:

```
oci network nsg rules update --nsg-id=ocidl.networksecuritygroup.oc1.iad.aaaaaaaaaaaaaaaaaaaaaaaaaaaa --security-rules='[{ "description": null, "destination": null, "destination-type": null, "direction": "INGRESS", "icmp-options": null, "id": "709001", "is-stateless": null, "protocol": "6", "source": "140.238.154.0/24", "source-type": "CIDR_BLOCK", "tcp-options": { "destination-port-range": { "max": 22, "min": 22 }, "source-port-range": null }, "udp-options": null }]'
```

CIS Controls:

Version 7

9.2 Ensure Only Approved Ports, Protocols and Services Are Running

Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system.

2.4 Ensure no network security groups allow ingress from 0.0.0.0/0 to port 3389 (Manual)

Profile Applicability:

- Level 1

Description:

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

Rationale:

Removing unfettered connectivity to remote console services, such as Remote Desktop Protocol (RDP), reduces a server's exposure to risk.

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 network security group or security list.

Audit:

From CLI:

Issue the following command, it should not return anything.

```
for region in `oci iam region list | jq -r '.data[] | .name'`;
do
  for compid in `oci iam compartment list 2>/dev/null | jq -r '.data[] | .id'`;
  do
    for nsgid in `oci network nsg list --compartment-id $compid --
region $region --all 2>/dev/null | jq -r '.data[] | .id'`;
    do
      output=`oci network nsg rules list --nsg-id=$nsgid --all
2>/dev/null | jq -r '.data[] | select(.source == "0.0.0.0/0" and .direction
== "INGRESS" and ((."tcp-options"."destination-port-range".max >= 3389 and
."tcp-options"."destination-port-range".min <= 3389) or ."tcp-
options"."destination-port-range" == null))'`;
      if [ ! -z "$output" ]; then echo "NSGID=", $nsgid,
"Security Rules=", $output; fi
    done
  done
done
```

Remediation:

From CLI:

Using the details returned from the audit procedure either:

- Remove the security rules

```
oci network nsg rules remove --nsg-id=<NSGID from audit output>
```

or

- Update the security rules

```
oci network nsg rules update --nsg-id=<NSGID from audit output> --security-rules=<updated security-rules JSON (without the isValid or TimeCreated fields)>
```

eg:

```
oci network nsg rules update --nsg-id=ocidl.networksecuritygroup.oc1.iad.aaaaaaaaaaaaaaaaaaaaaaaaaaaa --security-rules='[{ "description": null, "destination": null, "destination-type": null, "direction": "INGRESS", "icmp-options": null, "id": "709001", "is-stateless": null, "protocol": "6", "source": "140.238.154.0/24", "source-type": "CIDR_BLOCK", "tcp-options": { "destination-port-range": { "max": 3389, "min": 3389 }, "source-port-range": null }, "udp-options": null }]
```

CIS Controls:

Version 7

9.2 Ensure Only Approved Ports, Protocols and Services Are Running

Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system.

2.5 Ensure the default security list of every VCN restricts all traffic except ICMP (Automated)

Profile Applicability:

- Level 1

Description:

A default security list is created when a Virtual Cloud Network (VCN) is created. Security lists provide stateful filtering of ingress and egress network traffic to OCI resources. It is recommended no security list allows unrestricted ingress access to Secure Shell (SSH) via port 22.

Rationale:

Removing unfettered connectivity to remote console services, such as SSH on port 22, reduces a server's exposure to unauthorized access.

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.

Audit:

From Console:

1. Login into the OCI Console
2. Click on Networking -> Virtual Cloud Networks
3. For each VCN listed Click on Security Lists
4. Click on Default Security List for <VCN Name>
5. Verify that there is no Ingress rule with 'Source 0.0.0.0/0, IP Protocol 22 and Destination Port Range 22'

Remediation:

From Console:

1. Login into the OCI Console
2. Click on Networking -> Virtual Cloud Networks
3. For each VCN listed Click on Security Lists
4. Click on Default Security List for <VCN Name>

5. Select the Ingress Rule with 'Source 0.0.0.0/0, IP Protocol 22 and Destination Port Range 22'
6. Click `Remove`
7. Verify that you want to remove by clicking `Remove`

CIS Controls:

Version 7

9.2 Ensure Only Approved Ports, Protocols and Services Are Running

Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system.

3 Logging and Monitoring

This section contains recommendations for configuring logging and monitoring related options.

3.1 Ensure audit log retention period is set to 365 days (Automated)

Profile Applicability:

- Level 1

Description:

Ensuring audit logs are kept for 365 days.

Rationale:

Log retention controls how long activity logs should be retained. Studies have shown that The Mean Time to Detect(MTTD) a cyber breach is anywhere from 30 days in some sectors to up to 206 days in others. Retaining logs for at least 365 days or more will provide the ability to respond to incidents

Impact:

There is no performance impact when enabling the above described features but additional audit data will be retained.

Audit:

From Console:

1. Go to the Tenancy Details page: <https://console.us-ashburn-1.oraclecloud.com/a/tenancy>
2. View the Audit Retention Period and ensure it is set to 365 Days.

From CLI:

1. Retrieve the audit retention period from the command line

```
oci audit config get --compartment-id <compartment OCID>
```

2. Ensure the returned JSON contains `retention-period-days` of 365.

Remediation:

From Console:

1. Go to the Tenancy Details page: <https://console.us-ashburn-1.oraclecloud.com/a/tenancy>
2. Click Edit Audit Retention Policy

3. Set the `AUDIT RETENTION PERIOD` to 365
4. Click `Submit`

From CLI:

1. Update the `retention-period-days` to 365

```
oci audit config update --retention-period-days 365 --compartment-id  
<compartment OCID>
```

CIS Controls:

Version 7

6 Maintenance, Monitoring and Analysis of Audit Logs

Maintenance, Monitoring and Analysis of Audit Logs

3.2 Ensure default tags are used on resources (Manual)

Profile Applicability:

- Level 1

Description:

Using default tags is a way to ensure all resources that support tags are tagged during creation. Tags can be based on static values or based on computed values. It is recommended to setup default tags early on to ensure all created resources will get tagged. Tags are scoped to Compartments and are inherited by Child Compartments. The recommendation is to create default tags like “CreatedBy” at the Root Compartment level to ensure all resources get tagged. When using Tags it is important to ensure that Tag Namespaces are protected by IAM Policies otherwise this will allow users to change tags or tag values. Depending on the age of the OCI Tenancy there may already be Tag defaults setup at the Root Level and no need for further action to implement this action.

Rationale:

In the case of an incident having default tags like “CreatedBy” applied will provide info on who created the resource without having to search the Audit logs.

Impact:

There is no performance impact when enabling the above described features

Audit:

From Console:

1. Go to the `Compartments` page: <https://console.us-ashburn-1.oraclecloud.com>
2. Select the Root compartment
3. Select the `Tag Defaults` Link
4. In the Tag Defaults table verify that there is a Tag with a value of `${iam.principal.name}` and a Tag Key Status of `Active`

Note:

The name of the tag may be different then “CreatedBy” if the Tenancy Administrator has decided to use another tag.

From CLI:

1. List the active tag defaults defined at the Root compartment level by using the Tenancy OCID as compartment id.

Note: The Tenancy OCID can be found in the ~/.oci/config file used by the OCI Command Line Tool

```
oci iam tag-default list --compartment-id=<tenancy OCID> --query="data[?\"lifecycle-state\"=='ACTIVE']\". {\"name\": \"tag-definition-name\", \"value: value\"} --output table
```

2. Verify in the table returned that there is at least one row that contains the value of `${iam.principal.name}`

Remediation:

From Console:

1. Go to the Tag Namespaces page: <https://console.us-ashburn-1.oraclecloud.com/identity/tag-namespaces>
2. Select the Root compartment
3. Click Create Namespace Definition
4. Create the Namespace in the Root Compartment
5. Click on the newly created Namespace definition
6. Click Create Tag Key Definition
Create a Tag key definition by providing a Tag Key, Description and selecting "Static Value" for Tag Value Type
7. Go to the Compartments page:
<https://console.us-ashburn-1.oraclecloud.com/identity/compartments>
8. Select the Root compartment
9. Select the Tag Defaults Link
10. Click Create Tag Default
11. Create the Tag Default by providing the Tag namespace and Tag Key created previously and as the Default Value provide: `${iam.principal.name}`

From CLI:

1. Create a Tag Namespace in the Root Compartment

```
oci iam tag-namespace create --compartment-id=<tenancy OCID> --name=<name> --description=<description> --query data.{\"\\\"Tag Namespace OCID\\\":id\"} --output table
```

2. Note the Tag Namespace OCID and use it when creating the Tag Key Definition

```
oci iam tag create --tag-namespace-id=<tag namespace OCID> --name=<tag namespace OCID> --description=<description> --query data.{\"\\\"Tag Key Definition OCID\\\":id\"} --output table
```

3. Note the Tag Key Definition OCID and use it when creating the Tag Default in the Root compartment

```
oci iam tag-default create --compartment-id=<tenancy OCID> --tag-definition-id=<tag key definition id> --value="\${iam.principal.name}"
```

Default Value:

New OCI Tenancies will have Tag Defaults setup for CreatedBy and CreatedOn as default. If this is the case then there is no remediate action required in the Tenancy in order to meet this specific control.

Additional Information:

- The console URL shown is for the Ashburn region. Your tenancy might have a different home region and thus console URL.
- There is no requirement to use the “Oracle-Tags” namespace to implement this control. A Tag Namespace Administrator can create any namespace and use it for this control.

CIS Controls:

Version 7

1 Inventory and Control of Hardware Assets

Inventory and Control of Hardware Assets

1.4 Maintain Detailed Asset Inventory

Maintain an accurate and up-to-date inventory of all technology assets with the potential to store or process information. This inventory shall include all hardware assets, whether connected to the organization's network or not.

3.3 Create at least one notification topic and subscription to receive monitoring alerts (Manual)

Profile Applicability:

- Level 1

Description:

Notifications provide a multi-channel messaging service that allow users and applications to be notified of events of interest occurring within OCI. Messages can be sent via eMail, HTTPs, PagerDuty, Slack or the OCI Function service. Some channels, such as eMail require confirmation of the subscription before it becomes active.

Rationale:

Creating one or more notification topics allow administrators to be notified of relevant changes made to OCI infrastructure.

Impact:

There is no performance impact when enabling the above described features but depending on the amount of notifications sent per month there may be a cost associated.

Audit:

From Console:

1. Go to the Notifications Service page: <https://console.us-ashburn-1.oraclecloud.com/notification/topics>
2. Select the `Compartment` that hosts the notifications
3. Find and click the `Topic` relevant to your monitoring alerts.
4. Ensure a valid active subscription is shown.

From CLI:

1. List the topics in the `Compartment` that hosts the notifications

```
oci ons topic list --compartment-id <compartment OCID> --all
```

2. Note the `OCID` of the monitoring topic(s) using the `topic-id` field of the returned JSON and use it to list the subscriptions

```
oci ons subscription list --compartment-id <compartment OCID> --topic-id <topic OCID> --all
```

3. Ensure at least one active subscription is returned

Remediation:

From Console:

1. Go to the Notifications Service page: <https://console.us-ashburn-1.oraclecloud.com/notification/topics>
2. Select the `Compartment` that hosts the notifications
3. Click `Create Topic`
4. Set the `name` to something relevant
5. Set the `description` to describe the purpose of the topic
6. Click `Create`
7. Click the newly created topic
8. Click `Create Subscription`
9. Choose the correct `protocol`
10. Complete the correct parameter, for instance `email address`
11. Click `Create`

From CLI:

1. Create a topic in a compartment

```
oci ons topic create --name <topic name> --description <topic description> --compartment-id <compartment OCID>
```

2. Note the `OCID` of the `topic` using the `topic-id` field of the returned JSON and use it to create a new subscriptions

```
oci ons subscription create --compartment-id <compartment OCID> --topic-id <topic OCID> --protocol <protocol> --subscription-endpoint <subscription endpoint>
```

3. The returned JSON includes the id of the `subscription`.

Additional Information:

- The console URL shown is for the Ashburn region. Your tenancy might have a different home region and thus console URL.
- The same Notification topic can be reused by many Events. A single topic can have multiple subscriptions allowing the same topic to be published to multiple locations.
- The generated notification will include an `eventID` that can be used when querying the Audit Logs in case further investigation is required.

CIS Controls:

Version 7

6.2 Activate audit logging

Ensure that local logging has been enabled on all systems and networking devices.

6.7 Regularly Review Logs

On a regular basis, review logs to identify anomalies or abnormal events.

3.4 Ensure a notification is configured for Identity Provider changes (Manual)

Profile Applicability:

- Level 1

Description:

It is recommended to setup an Event Rule and Notification that gets triggered when Identity Providers are created, updated or deleted. Event Rules are compartment scoped and will detect events in child compartments. It is recommended to create the Event rule at the root compartment level.

Rationale:

OCI Identity Providers allow management of User ID / passwords in external systems and use of those credentials to access OCI resources. Identity Providers allow users to single sign-on to OCI console and have other OCI credentials like API Keys. Monitoring and alerting on changes to Identity Providers will help in identifying changes to the security posture.

Impact:

There is no performance impact when enabling the above described features but depending on the amount of notifications sent per month there may be a cost associated.

Audit:

From Console:

1. Go to the Events Service page:
<https://console.us-ashburn-1.oraclecloud.com/events/rules>
2. Select the `Compartment` that hosts the rules
3. Find and click the `Rule` that handles `Identity Provider Changes` (if any)
4. Click the `Edit Rule` button and verify that the `RuleConditions` section contains a condition for the `Service Identity` and `Event Types: Identity Provider - Create, Identity Provider - Delete and Identity Provider - Update`
5. Verify that in the `Actions` section the `Action Type` contains: `Notifications` and that a valid `Topic` is referenced.

From CLI:

1. Find the OCID of the specific Event Rule based on Display Name and Compartment OCID

```
oci events rule list --compartment-id=<compartment OCID> --query "data[?\"display-name\"=='<display name used>'].{\"id:id\"} --output table
```

2. List the details of a specific Event Rule based on the OCID of the rule.

```
oci events rule get --rule-id=<rule OCID>
```

3. In the JSON output locate the Conditions key value pair and verify that the following Conditions are present:

```
com.oraclecloud.identitycontrolplane.createidentityprovider  
com.oraclecloud.identitycontrolplane.deleteidentityprovider  
com.oraclecloud.identitycontrolplane.updateidentityprovider
```

4. Verify the value of the `is-enabled` attribute is `true`
5. In the JSON output verify that `actionType` is `ONS` and locate the `topic-id`
6. Verify the correct topic is used by checking the topic name

```
oci ons topic get --topic-id=<topic id> --query data.{\"name:name\"} --output table
```

Remediation:

From Console:

1. Go to the Events Service page: <https://console.us-ashburn-1.oraclecloud.com/events/rules>
2. Select the compartment that should host the rule
3. Click Create Rule
4. Provide a Display Name and Description
5. Create a Rule Condition by selecting Identity in the Service Name Drop-down and selecting Identity Provider - Create, Identity Provider - Delete and Identity Provider - Update
6. In the Actions section select Notifications as Action Type
7. Select the Compartment that hosts the Topic to be used.
8. Select the Topic to be used
9. Optionally add Tags to the Rule
10. Click Create Rule

From CLI:

1. Find the `topic-id` of the topic the Event Rule should use for sending Notifications by using the `topic` name and `Compartment OCID`

```
oci ons topic list --compartment-id=<compartment OCID> --all --query "data[?name=='<topic_name>']".{"name:name,topic_id:\"topic-id\""} --output table
```

2. Create a JSON file to be used when creating the Event Rule. Replace topic id, display name, description and compartment OCID.

```
{
  "actions":
  {
    "actions": [
      {
        "actionType": "ONS",
        "isEnabled": true,
        "topicId": "<topic id>"
      }
    ],
    "condition":
    "{\\"eventType\\":[\\"com.oraclecloud.identitycontrolplane.createidentityprovider\\",\\" com.oraclecloud.identitycontrolplane.deleteidentityprovider\\",\\" com.oraclecloud.identitycontrolplane.updateidentityprovider\\"],\\"data\\":{}}",
    "displayName": "<display name>",
    "description": "<description>",
    "isEnabled": true,
    "compartmentId": "compartment OCID"
  }
}
```

3. Create the actual event rule

```
oci events rule create --from-json file://event_rule.json
```

4. Note in the JSON returned that it lists the parameters specified in the JSON file provided and that there is an OCID provided for the Event Rule

Additional Information:

- The console URL shown is for the Ashburn region. Your tenancy might have a different home region and thus console URL.
- The same Notification topic can be reused by many Event Rules.
- The generated notification will include an `eventID` that can be used when querying the Audit Logs in case further investigation is required.

CIS Controls:

Version 7

16 Account Monitoring and Control

Account Monitoring and Control

3.5 Ensure a notification is configured for IdP group mapping changes (Manual)

Profile Applicability:

- Level 1

Description:

It is recommended to setup an Event Rule and Notification that gets triggered when Identity Provider Group Mappings are created, updated or deleted. Event Rules are compartment scoped and will detect events in child compartments. It is recommended to create the Event rule at the root compartment level

Rationale:

IAM Policies govern access to all resources within an OCI Tenancy. IAM Policies use OCI Groups for assigning the privileges. Identity Provider Groups could be mapped to OCI Groups to assign privileges to federated users in OCI. Monitoring and alerting on changes to Identity Provider Group mappings will help in identifying changes to the security posture.

Impact:

There is no performance impact when enabling the above described features but depending on the amount of notifications sent per month there may be a cost associated.

Audit:

From Console:

1. Go to the Events Service page:
<https://console.us-ashburn-1.oraclecloud.com/events/rules>
2. Select the `Compartment` that hosts the rules
3. Find and click the `Rule` that handles `Idp Group Mapping Changes` (if any)
4. Click the `Edit Rule` button and verify that the `RuleConditions` section contains a condition for the `Service Identity and Event Types: Idp Group Mapping - Create, Idp Group Mapping - Delete and Idp Group Mapping - Update`
5. Verify that in the `Actions` section the `Action Type` contains: `Notifications` and that a valid `Topic` is referenced.

From CLI:

1. Find the OCID of the specific Event Rule based on Display Name and Compartment OCID

```
oci events rule list --compartment-id=<compartment OCID> --query "data
[?\"display-name\"=='<display name used>'].{\"id:id\"} --output table
```

2. List the details of a specific Event Rule based on the OCID of the rule.

```
oci events rule get --rule-id=<rule OCID>
```

3. In the JSON output locate the Conditions key value pair and verify that the following Conditions are present:

```
com.oraclecloud.identitycontrolplane.createpolicy
com.oraclecloud.identitycontrolplane.deletepolicy
com.oraclecloud.identitycontrolplane.updatepolicy
```

4. Verify the value of the `is-enabled` attribute is `true`
5. In the JSON output verify that `actionType` is `ONS` and locate the `topic-id`
6. Verify the correct topic is used by checking the topic name

```
oci ons topic get --topic-id=<topic id> --query data.{\"name:name\"} --output
table
```

Remediation:

From Console:

1. Go to the Events Service page: <https://console.us-ashburn-1.oraclecloud.com/events/rules>
2. Select the compartment that should host the rule
3. Click Create Rule
4. Provide a Display Name and Description
5. Create a Rule Condition by selecting Identity in the Service Name Drop-down and selecting Idp Group Mapping - Create, Idp Group Mapping - Delete and Idp Group Mapping - Update
6. In the Actions section select Notifications as Action Type
7. Select the Compartment that hosts the Topic to be used.
8. Select the Topic to be used
9. Optionally add Tags to the Rule
10. Click Create Rule

From CLI:

1. Find the `topic-id` of the topic the Event Rule should use for sending Notifications by using the topic name and Compartment OCID

```
oci ons topic list --compartment-id=<compartment OCID> --all --query "data
[?name=='<topic_name>'].{\"name:name,topic_id:\"topic-id\"}\" --output table
```

2. Create a JSON file to be used when creating the Event Rule. Replace topic id, display name, description and compartment OCID.

```
{
  "actions": {
    "actions": [
      {
        "actionType": "ONS",
        "isEnabled": true,
        "topicId": "<topic id>"
      }
    ]
  },
  "condition": {
    "\\event\\": [\\"com.oraclecloud.identitycontrolplane.createpolicy\\", \\"com.oraclecloud.identitycontrolplane.deletepolicy\\", \\"com.oraclecloud.identitycontrolplane.updatepolicy\\"], \\"data\\": {}},
    "displayName": "<display name>",
    "description": "<description>",
    "isEnabled": true,
    "compartmentId": "compartment OCID"
  }
}
```

3. Create the actual event rule

```
oci events rule create --from-json file://event_rule.json
```

4. Note in the JSON returned that it lists the parameters specified in the JSON file provided and that there is an OCID provided for the Event Rule

Additional Information:

- The console URL shown is for the Ashburn region. Your tenancy might have a different home region and thus console URL.
- The same Notification topic can be reused by many Event Rules.
- The generated notification will include an eventID that can be used when querying the Audit Logs in case further investigation is required.

CIS Controls:

Version 7

16 Account Monitoring and Control

Account Monitoring and Control

3.6 Ensure a notification is configured for IAM group changes (Manual)

Profile Applicability:

- Level 1

Description:

It is recommended to setup an Event Rule and Notification that gets triggered when IAM Groups are created, updated or deleted. Event Rules are compartment scoped and will detect events in child compartments, it is recommended to create the Event rule at the root compartment level.

Rationale:

IAM Groups control access to all resources within an OCI Tenancy. Monitoring and alerting on changes to IAM Groups will help in identifying changes to satisfy least privilege principle.

Impact:

There is no performance impact when enabling the above described features but depending on the amount of notifications sent per month there may be a cost associated.

Audit:

From Console:

1. Go to the `Events Service` page:
<https://console.us-ashburn-1.oraclecloud.com/events/rules>
2. Select the `Compartment` that hosts the rules
3. Find and click the `Rule` that handles IAM Group Changes
4. Click the `Edit Rule` button and verify that the `Rule Conditions` section contains a condition for the `Service Identity` and `Event Types: Group - Create, Group - Delete and Group - Update`
5. Verify that in the `Actions` section the `Action Type` contains: `Notifications` and that a valid `Topic` is referenced.

From CLI:

1. Find the OCID of the specific Event Rule based on `Display Name` and `Compartment OCID`

```
oci events rule list --compartment-id=<compartment OCID> --query "data[?\"display-name\"=='<display name used>'].{\"id:id\"} --output table
```

2. List the details of a specific Event Rule based on the OCID of the rule.

```
oci events rule get --rule-id=<rule OCID>
```

3. In the JSON output locate the Conditions key value pair and verify that the following Conditions are present:

```
com.oraclecloud.identitycontrolplane.creategroup
com.oraclecloud.identitycontrolplane.deletegroup
com.oraclecloud.identitycontrolplane.updategroup
```

4. Verify the value of the `is-enabled` attribute is `true`
5. In the JSON output verify that `actionType` is `ONS` and locate the `topic-id`
6. Verify the correct topic is used by checking the topic name

```
oci ons topic get --topic-id=<topic id> --query data>{"name:name"} --output
table
```

Remediation:

From Console:

1. Go to the Events Service page: <https://console.us-ashburn-1.oraclecloud.com/events/rules>
2. Select the compartment that should host the rule
3. Click Create Rule
4. Provide a Display Name and Description
5. Create a Rule Condition by selecting Identity in the Service Name Drop-down and selecting Group - Create, Group - Delete and Group - Update
6. In the Actions section select Notifications as Action Type
7. Select the Compartment that hosts the Topic to be used.
8. Select the Topic to be used
9. Optionally add Tags to the Rule
10. Click Create Rule

From CLI:

1. Find the `topic-id` of the topic the Event Rule should use for sending Notifications by using the topic name and Compartment OCID

```
oci ons topic list --compartment-id=<compartment OCID> --all --query "data
[?name=='<topic_name>']".{"name:name,topic_id:\"topic-id\""} --output table
```

2. Create a JSON file to be used when creating the Event Rule. Replace topic id, display name, description and compartment OCID.

```
{
  "actions":
  {
    "actions": [
      {
        "actionType": "ONS",
        "isEnabled": true,
        "topicId": "<topic id>"
      }
    ],
    "condition":
    "{\\"eventType\\": [\\"com.oraclecloud.identitycontrolplane.creategroup\\", \\"com.oraclecloud.identitycontrolplane.deletegroup\\", \\"com.oraclecloud.identitycontrolplane.updategroup\\"], \\"data\\": {}}",
    "displayName": "<display name>",
    "description": "<description>",
    "isEnabled": true,
    "compartmentId": "compartment OCID"
  }
}
```

3. Create the actual event rule

```
oci events rule create --from-json file://event_rule.json
```

4. Note in the JSON returned that it lists the parameters specified in the JSON file provided and that there is an OCID provided for the Event Rule

Additional Information:

- The console URL shown is for the Ashburn region. Your tenancy might have a different home region and thus console URL.
- The same Notification topic can be reused by many Event Rules.
- The generated notification will include an eventId that can be used when querying the Audit Logs in case further investigation is required.

CIS Controls:

Version 7

16 Account Monitoring and Control

Account Monitoring and Control

3.7 Ensure a notification is configured for IAM policy changes (Manual)

Profile Applicability:

- Level 1

Description:

It is recommended to setup an Event Rule and Notification that gets triggered when IAM Policies are created, updated or deleted. Event Rules are compartment scoped and will detect events in child compartments, it is recommended to create the Event rule at the root compartment level.

Rationale:

IAM Policies govern access to all resources within an OCI Tenancy. Monitoring and alerting on changes to IAM policies will help in identifying changes to the security posture.

Impact:

There is no performance impact when enabling the above described features but depending on the amount of notifications sent per month there may be a cost associated.

Audit:

From Console:

1. Go to the Events Service page:
<https://console.us-ashburn-1.oraclecloud.com/events/rules>
2. Select the `Compartment` that hosts the rules
3. Find and click the `Rule` that handles `IAM Policy Changes` (if any)
4. Click the `Edit Rule` button and verify that the `RuleConditions` section contains a condition for the `Service Identity and Event Types: Policy - Create, Policy - Delete and Policy - Update`
5. Verify that in the `Actions` section the `Action Type` contains: `Notifications` and that a valid `Topic` is referenced.

From CLI:

1. Find the OCID of the specific Event Rule based on Display Name and Compartment OCID

```
oci events rule list --compartment-id=<compartment OCID> --query "data\n[?\"display-name\"=='<display name used>']\".\"id:id\"} --output table
```

2. List the details of a specific Event Rule based on the OCID of the rule.

```
oci events rule get --rule-id=<rule OCID>
```

3. In the JSON output locate the Conditions key value pair and verify that the following Conditions are present:

```
com.oraclecloud.identitycontrolplane.createpolicy
com.oraclecloud.identitycontrolplane.deletepolicy
com.oraclecloud.identitycontrolplane.updatepolicy
```

4. Verify the value of the `is-enabled` attribute is `true`
5. In the JSON output verify that `actionType` is `ONS` and locate the `topic-id`
6. Verify the correct topic is used by checking the topic name

```
oci ons topic get --topic-id=<topic id> --query data>{"name:name"} --output
table
```

Remediation:

From Console:

1. Go to the Events Service page: <https://console.us-ashburn-1.oraclecloud.com/events/rules>
2. Select the compartment that should host the rule
3. Click Create Rule
4. Provide a Display Name and Description
5. Create a Rule Condition by selecting Identity in the Service Name Drop-down and selecting Policy - Change Compartment, Policy - Create, Policy - Delete and Policy - Update
6. In the Actions section select Notifications as Action Type
7. Select the Compartment that hosts the Topic to be used.
8. Select the Topic to be used
9. Optionally add Tags to the Rule
10. Click Create Rule

From the Command Line:

1. Find the `topic-id` of the topic the Event Rule should use for sending Notifications by using the topic name and Compartment OCID

```
oci ons topic list --compartment-id=<compartment OCID> --all --query "data
[?name=='<topic_name>']".{"name:name,topic_id:\"topic-id\""} --output table
```

2. Create a JSON file to be used when creating the Event Rule. Replace topic id, display name, description and compartment OCID.

```
{
  "actions": {
    "actions": [
      {
        "actionType": "ONS",
        "isEnabled": true,
        "topicId": "<topic id>"
      }
    ]
  },
  "condition": {
    "\\event\\": [\\"com.oraclecloud.identitycontrolplane.createpolicy\\", \\"com.oraclecloud.identitycontrolplane.deletepolicy\\", \\"com.oraclecloud.identitycontrolplane.updatepolicy\\"], \\"data\\": {}",
    "displayName": "<display name>",
    "description": "<description>",
    "isEnabled": true,
    "compartmentId": "compartment OCID"
  }
}
```

3. Create the actual event rule

```
oci events rule create --from-json file://event_rule.json
```

4. Note in the JSON returned that it lists the parameters specified in the JSON file provided and that there is an OCID provided for the Event Rule

Additional Information:

- The console URL shown is for the Ashburn region. Your tenancy might have a different home region and thus console URL.
- The same Notification topic can be reused by many Event Rules.
- The generated notification will include an eventID that can be used when querying the Audit Logs in case further investigation is required.

CIS Controls:

Version 7

16 Account Monitoring and Control

Account Monitoring and Control

3.8 Ensure a notification is configured for user changes (Manual)

Profile Applicability:

- Level 1

Description:

It is recommended to setup an Event Rule and Notification that gets triggered when IAM Users are created, updated, deleted, capabilities updated, or state updated. Event Rules are compartment scoped and will detect events in child compartments, it is recommended to create the Event rule at the root compartment level.

Rationale:

Users use or manage Oracle Cloud Infrastructure resources. Monitoring and alerting on changes to Users will help in identifying changes to the security posture.

Impact:

There is no performance impact when enabling the above described features but depending on the amount of notifications sent per month there may be a cost associated.

Audit:

From Console:

1. Go to the `Events Service` page:
<https://console.us-ashburn-1.oraclecloud.com/events/rules>
2. Select the `Compartment` that hosts the rules
3. Find and click the `Rule` that handles `IAM User Changes`
4. Click the `Edit Rule` button and verify that the `Rule Conditions` section contains a condition for the `Service Identity` and `Event Types`:
User - Create,
User - Delete,
User - Update,
User Capabilities - Update,
User State - Update
5. Verify that in the `Actions` section the `Action Type` contains: `Notifications` and that a valid `Topic` is referenced.

From CLI:

1. Find the OCID of the specific Event Rule based on `Display Name` and `Compartment OCID`

```
oci events rule list --compartment-id=<compartment OCID> --query "data
[?\"display-name\"=='<display name used>'].{\"id:id\"} --output table
```

2. List the details of a specific Event Rule based on the OCID of the rule.

```
oci events rule get --rule-id=<rule OCID>
```

3. In the JSON output locate the Conditions key value pair and verify that the following Conditions are present:

```
com.oraclecloud.identityControlPlane.CreateUser
com.oraclecloud.identityControlPlane.DeleteUser
com.oraclecloud.identityControlPlane.UpdateUser
com.oraclecloud.identityControlPlane.UpdateUserCapabilities
com.oraclecloud.identityControlPlane.UpdateUserCapabilities
```

4. Verify the value of the `is-enabled` attribute is `true`
5. In the JSON output verify that `actionType` is `ONS` and locate the `topic-id`
6. Verify the correct topic is used by checking the topic name

```
oci ons topic get --topic-id=<topic id> --query data.{\"name:name\"} --output
table
```

Remediation:

From the Console:

1. Go to the Events Service page: <https://console.us-ashburn-1.oraclecloud.com/events/rules>
2. Select the compartment that should host the rule
3. Click **Create Rule**
4. Provide a **Display Name** and **Description**
5. Create a Rule Condition by selecting **Identity** in the Service Name Drop-down and selecting:
 - User - Create,
 - User - Delete,
 - User - Update,
 - User Capabilities - Update,
 - User State - Update
6. In the **Actions** section select **Notifications** as **Action Type**
7. Select the **Compartment** that hosts the Topic to be used.
8. Select the **Topic** to be used
9. Optionally add **Tags** to the Rule
10. Click **Create Rule**

From CLI:

1. Find the `topic-id` of the topic the Event Rule should use for sending Notifications by using the `topic` name and `Compartment OCID`

```
oci ons topic list --compartment-id=<compartment OCID> --all --query "data[?name=='<topic_name>']".{"name:name,topic_id:\"topic-id\""} --output table
```

2. Create a JSON file to be used when creating the Event Rule. Replace topic id, display name, description and compartment OCID.

```
{
  "actions":
  {
    "actions": [
      {
        "actionType": "ONS",
        "isEnabled": true,
        "topicId": "<topic id>"
      }
    ],
    "condition":
    "[\"event\\\": [\"com.oraclecloud.identityControlPlane.CreateUser\\\", \"com.oraclecloud.identityControlPlane.DeleteUser\\\", \"com.oraclecloud.identityControlPlane.UpdateUser\\\", \"com.oraclecloud.identityControlPlane.UpdateUserCapabilities\\\", \"com.oraclecloud.identityControlPlane.UpdateUserState\\\", \"data\\\": {}]\", \"displayName\": \"<display name>\", \"description\": \"<description>\", \"isEnabled\": true, \"compartmentId\": \"compartment OCID\"
  }
}
```

3. Create the actual event rule

```
oci events rule create --from-json file://event_rule.json
```

4. Note in the JSON returned that it lists the parameters specified in the JSON file provided and that there is an OCID provided for the Event Rule

Additional Information:

- The console URL shown is for the Ashburn region. Your tenancy might have a different home region and thus console URL.
- The same Notification topic can be reused by many Event Rules.
- The generated notification will include an `eventID` that can be used when querying the Audit Logs in case further investigation is required.

CIS Controls:

Version 7

4.1 Maintain Inventory of Administrative Accounts

Use automated tools to inventory all administrative accounts, including domain and local accounts, to ensure that only authorized individuals have elevated privileges.

16 Account Monitoring and Control

Account Monitoring and Control

3.9 Ensure a notification is configured for VCN changes (Manual)

Profile Applicability:

- Level 1

Description:

It is recommended to setup an Event Rule and Notification that gets triggered when Virtual Cloud Networks are created, updated or deleted. Event Rules are compartment scoped and will detect events in child compartments, it is recommended to create the Event rule at the root compartment level.

Rationale:

Virtual Cloud Networks (VCNs) closely resembles a traditional network. Monitoring and alerting on changes to VCNs will help in identifying changes to the security posture.

Impact:

There is no performance impact when enabling the above described features but depending on the amount of notifications sent per month there may be a cost associated.

Audit:

From Console:

1. Go to the Events Service page:
<https://console.us-ashburn-1.oraclecloud.com/events/rules>
2. Select the `Compartment` that hosts the rules
3. Find and click the `Rule` that handles VCN Changes (if any)
4. Click the `Edit Rule` button and verify that the `RuleConditions` section contains a condition for the `Service Networking and Event Types: VCN - Create, VCN - Delete and VCN - Update`
5. Verify that in the `Actions` section the `Action Type` contains: `Notifications` and that a valid `Topic` is referenced.

From CLI:

1. Find the OCID of the specific Event Rule based on Display Name and Compartment OCID

```
oci events rule list --compartment-id=<compartment OCID> --query "data[?\"display-name\"=='<display name used>'].{\"id:id\"} --output table
```

2. List the details of a specific Event Rule based on the OCID of the rule.

```
oci events rule get --rule-id=<rule OCID>
```

3. In the JSON output locate the Conditions key value pair and verify that the following Conditions are present:

```
com.oraclecloud.virtualnetwork.createvcn  
com.oraclecloud.virtualnetwork.deletevcn  
com.oraclecloud.virtualnetwork.updatevcn
```

4. Verify the value of the `is-enabled` attribute is `true`
5. In the JSON output verify that `actionType` is `ONS` and locate the `topic-id`
6. Verify the correct topic is used by checking the topic name

```
oci ons topic get --topic-id=<topic id> --query data>{"name:name"} --output  
table
```

Remediation:

From Console:

1. Go to the Events Service page: <https://console.us-ashburn-1.oraclecloud.com/events/rules>
2. Select the compartment that should host the rule
3. Click Create Rule
4. Provide a Display Name and Description
5. Create a Rule Condition by selecting Networking in the Service Name Drop-down and selecting VCN - Create, VCN - Delete and VCN - Update
6. In the Actions section select Notifications as Action Type
7. Select the Compartment that hosts the Topic to be used.
8. Select the Topic to be used
9. Optionally add Tags to the Rule
10. Click Create Rule

From CLI:

1. Find the `topic-id` of the topic the Event Rule should use for sending Notifications by using the topic name and Compartment OCID

```
oci ons topic list --compartment-id=<compartment OCID> --all --query "data  
[?name=='<topic_name>']".{"name:name,topic_id:\"topic-id\""} --output table
```

2. Create a JSON file to be used when creating the Event Rule. Replace topic id, display name, description and compartment OCID.

```
{
  "actions": {
    "actions": [
      {
        "actionType": "ONS",
        "isEnabled": true,
        "topicId": "<topic id>"
      }
    ]
  },
  "condition": {
    "eventTypes": [
      "com.oraclecloud.virtualnetwork.createvcn",
      "com.oraclecloud.virtualnetwork.deletevcn",
      "com.oraclecloud.virtualnetwork.updatevcn"
    ],
    "data": {},
    "displayName": "<display name>",
    "description": "<description>",
    "isEnabled": true,
    "compartmentId": "compartment OCID"
  }
}
```

3. Create the actual event rule

```
oci events rule create --from-json file://event_rule.json
```

4. Note in the JSON returned that it lists the parameters specified in the JSON file provided and that there is an OCID provided for the Event Rule

Additional Information:

- The console URL shown is for the Ashburn region. Your tenancy might have a different home region and thus console URL.
- The same Notification topic can be reused by many Event Rules.
- The generated notification will include an eventID that can be used when querying the Audit Logs in case further investigation is required.

CIS Controls:

Version 7

11.3 Use Automated Tools to Verify Standard Device Configurations and Detect Changes

Compare all network device configuration against approved security configurations defined for each network device in use and alert when any deviations are discovered.

3.10 Ensure a notification is configured for changes to route tables (Manual)

Profile Applicability:

- Level 1

Description:

It is recommended to setup an Event Rule and Notification that gets triggered when route tables are created, updated or deleted. Event Rules are compartment scoped and will detect events in child compartments, it is recommended to create the Event rule at the root compartment level.

Rationale:

Route tables control traffic flowing to or from Virtual Cloud Networks and Subnets. Monitoring and alerting on changes to route tables will help in identifying changes these traffic flows.

Impact:

There is no performance impact when enabling the above described features but depending on the amount of notifications sent per month there may be a cost associated.

Audit:

From Console:

1. Go to the Events Service page:
<https://console.us-ashburn-1.oraclecloud.com/events/rules>
2. Select the `Compartment` that hosts the rules
3. Find and click the `Rule` that handles `Route Table Changes` (if any)
4. Click the `Edit Rule` button and verify that the `RuleConditions` section contains a condition for the `Service Networking and Event Types: Route Table - Change` `Compartment, Route Table - Create, Route Table - Delete and Route Table - Update`
5. Verify that in the `Actions` section the `Action Type` contains: `Notifications` and that a valid `Topic` is referenced.

From CLI:

1. Find the OCID of the specific Event Rule based on Display Name and Compartment OCID

```
oci events rule list --compartment-id=<compartment OCID> --query "data
[?\"display-name\"=='<display name used>'].{\"id:id\"} --output table
```

2. List the details of a specific Event Rule based on the OCID of the rule.

```
oci events rule get --rule-id=<rule OCID>
```

3. In the JSON output locate the Conditions key value pair and verify that the following Conditions are present:

```
com.oraclecloud.virtualnetwork.changeroutetablecompartment
com.oraclecloud.virtualnetwork.createroutetable
com.oraclecloud.virtualnetwork.deleteroutetable
com.oraclecloud.virtualnetwork.updateroutetable
```

4. Verify the value of the `is-enabled` attribute is `true`
5. In the JSON output verify that `actionType` is `ONS` and locate the `topic-id`
6. Verify the correct topic is used by checking the topic name

```
oci ons topic get --topic-id=<topic id> --query data.{\"name:name\"} --output
table
```

Remediation:

From the Console:

1. Go to the Events Service page: <https://console.us-ashburn-1.oraclecloud.com/events/rules>
2. Select the compartment that should host the rule
3. Click Create Rule
4. Provide a Display Name and Description
5. Create a Rule Condition by selecting Networking in the Service Name Drop-down and selecting Route Table - Change Compartment, Route Table - Create, Route Table - Delete and Route Table - Update
6. In the Actions section select Notifications as Action Type
7. Select the Compartment that hosts the Topic to be used.
8. Select the Topic to be used
9. Optionally add Tags to the Rule
10. Click Create Rule

From CLI:

1. Find the `topic-id` of the topic the Event Rule should use for sending Notifications by using the topic name and Compartment OCID

```
oci ons topic list --compartment-id=<compartment OCID> --all --query "data
[?name=='<topic_name>'].{\"name:name,topic_id:\"topic-id\"}\" --output table
```

2. Create a JSON file to be used when creating the Event Rule. Replace topic id, display name, description and compartment OCID.

```
{
  "actions":
  {
    "actions": [
      {
        "actionType": "ONS",
        "isEnabled": true,
        "topicId": "<topic id>"
      }
    ]
  },
  "condition":
  "{\\"eventType\\":[\\\"
com.oraclecloud.virtualnetwork.changeroutetablecompartment\\\",\\\"
com.oraclecloud.virtualnetwork.createroutetable\\\",\\\"
com.oraclecloud.virtualnetwork.deleteroutetable\\\",\\\"
com.oraclecloud.virtualnetwork.updateroutetable\\\"],\\\"data\\\":{}}\",
  \"displayName\": \"<display name>\",
  \"description\": \"<description>\",
  \"isEnabled\": true,
  \"compartmentId\": \"compartment OCID\"
}
```

3. Create the actual event rule

```
oci events rule create --from-json file://event_rule.json
```

4. Note in the JSON returned that it lists the parameters specified in the JSON file provided and that there is an OCID provided for the Event Rule

Additional Information:

- The console URL shown is for the Ashburn region. Your tenancy might have a different home region and thus console URL.
- The same Notification topic can be reused by many Event Rules.
- The generated notification will include an eventId that can be used when querying the Audit Logs in case further investigation is required.

CIS Controls:

Version 7

11.3 Use Automated Tools to Verify Standard Device Configurations and Detect Changes

Compare all network device configuration against approved security configurations defined for each network device in use and alert when any deviations are discovered.

3.11 Ensure a notification is configured for security list changes (Manual)

Profile Applicability:

- Level 1

Description:

It is recommended to setup an Event Rule and Notification that gets triggered when security lists are created, updated or deleted. Event Rules are compartment scoped and will detect events in child compartments, it is recommended to create the Event rule at the root compartment level.

Rationale:

Security Lists control traffic flowing into and out of Subnets within a Virtual Cloud Network. Monitoring and alerting on changes to Security Lists will help in identifying changes to these security controls.

Impact:

There is no performance impact when enabling the above described features but depending on the amount of notifications sent per month there may be a cost associated.

Audit:

From Console:

1. Go to the Events Service page:
<https://console.us-ashburn-1.oraclecloud.com/events/rules>
2. Select the `Compartment` that hosts the rules
3. Find and click the `Rule` that handles `Security List Changes` (if any)
4. Click the `Edit Rule` button and verify that the `RuleConditions` section contains a condition for the `Service Networking and Event Types: Security List - Change Compartment, Security List - Create, Security List - Delete and Security List - Update`
5. Verify that in the `Actions` section the `Action Type` contains: `Notifications` and that a valid `Topic` is referenced.

From CLI:

1. Find the OCID of the specific Event Rule based on Display Name and Compartment OCID

```
oci events rule list --compartment-id=<compartment OCID> --query "data
[?\"display-name\"=='<display name used>'].{\"id:id\"} --output table
```

2. List the details of a specific Event Rule based on the OCID of the rule.

```
oci events rule get --rule-id=<rule OCID>
```

3. In the JSON output locate the Conditions key value pair and verify that the following Conditions are present:

```
com.oraclecloud.virtualnetwork.changesecuritylistcompartment
com.oraclecloud.virtualnetwork.createsecuritylist
com.oraclecloud.virtualnetwork.deletesecuritylist
com.oraclecloud.virtualnetwork.updatesecuritylist
```

4. Verify the value of the `is-enabled` attribute is `true`
5. In the JSON output verify that `actionType` is `ONS` and locate the `topic-id`
6. Verify the correct topic is used by checking the topic name

```
oci ons topic get --topic-id=<topic id> --query data.{\"name:name\"} --output
table
```

Remediation:

From Console:

1. Go to the Events Service page: <https://console.us-ashburn-1.oraclecloud.com/events/rules>
2. Select the compartment that should host the rule
3. Click Create Rule
4. Provide a Display Name and Description
5. Create a Rule Condition by selecting Networking in the Service Name Drop-down and selecting Network Security List - Change Compartment, Security List - Create, Security List - Delete and Security List - Update
6. In the Actions section select Notifications as Action Type
7. Select the Compartment that hosts the Topic to be used.
8. Select the Topic to be used
9. Optionally add Tags to the Rule
10. Click Create Rule

From CLI:

1. Find the `topic-id` of the topic the Event Rule should use for sending Notifications by using the topic name and Compartment OCID

```
oci ons topic list --compartment-id=<compartment OCID> --all --query "data
[?name=='<topic_name>'].{\"name:name,topic_id:\"topic-id\"}\" --output table
```

2. Create a JSON file to be used when creating the Event Rule. Replace topic id, display name, description and compartment OCID.

```
{
  "actions": {
    "actions": [
      {
        "actionType": "ONS",
        "isEnabled": true,
        "topicId": "<topic id>"
      }
    ]
  },
  "condition": {
    "\"eventType\": [\"com.oraclecloud.virtualnetwork.changesecuritylistcompartment\", \"com.oraclecloud.virtualnetwork.createsecuritylist\", \"com.oraclecloud.virtualnetwork.deletesecuritylist\", \"com.oraclecloud.virtualnetwork.updatesecuritylist\"]\", \"data\": {}\", \"displayName\": \"<display name>\", \"description\": \"<description>\", \"isEnabled\": true, \"compartmentId\": \"compartment OCID\"
  }
}
```

3. Create the actual event rule

```
oci events rule create --from-json file://event_rule.json
```

4. Note in the JSON returned that it lists the parameters specified in the JSON file provided and that there is an OCID provided for the Event Rule

Additional Information:

- The console URL shown is for the Ashburn region. Your tenancy might have a different home region and thus console URL.
- The same Notification topic can be reused by many Event Rules.
- The generated notification will include an eventID that can be used when querying the Audit Logs in case further investigation is required.

CIS Controls:

Version 7

11.3 Use Automated Tools to Verify Standard Device Configurations and Detect Changes

Compare all network device configuration against approved security configurations defined for each network device in use and alert when any deviations are discovered.

3.12 Ensure a notification is configured for network security group changes (Manual)

Profile Applicability:

- Level 1

Description:

It is recommended to setup an Event Rule and Notification that gets triggered when network security groups are created, updated or deleted. Event Rules are compartment scoped and will detect events in child compartments, it is recommended to create the Event rule at the root compartment level.

Rationale:

Network Security Groups control traffic flowing between Virtual Network Cards attached to Compute instances. Monitoring and alerting on changes to Network Security Groups will help in identifying changes these security controls.

Impact:

There is no performance impact when enabling the above described features but depending on the amount of notifications sent per month there may be a cost associated.

Audit:

From Console:

1. Go to the Events Service page:
<https://console.us-ashburn-1.oraclecloud.com/events/rules>
2. Select the `Compartment` that hosts the rules
3. Find and click the `Rule` that handles `Network Security Group Changes` (if any)
4. Click the `Edit Rule` button and verify that the `RuleConditions` section contains a condition for the `Service Networking and Event Types: Network Security Group - Change Compartment, Network Security Group - Create, Network Security Group - Delete and Network Security Group - Update`
5. Verify that in the `Actions` section the `Action Type` contains: `Notifications` and that a valid `Topic` is referenced.

From CLI:

1. Find the OCID of the specific Event Rule based on Display Name and Compartment OCID

```
oci events rule list --compartment-id=<compartment OCID> --query "data
[?\"display-name\"=='<display name used>'].{\"id:id\"} --output table
```

2. List the details of a specific Event Rule based on the OCID of the rule.

```
oci events rule get --rule-id=<rule OCID>
```

3. In the JSON output locate the Conditions key value pair and verify that the following Conditions are present:

```
com.oraclecloud.virtualnetwork.changenetworksecuritygroupcompartment
com.oraclecloud.virtualnetwork.createnetworksecuritygroup
com.oraclecloud.virtualnetwork.deletenetworksecuritygroup
com.oraclecloud.virtualnetwork.updatenetworksecuritygroup
```

4. Verify the value of the `is-enabled` attribute is `true`
5. In the JSON output verify that `actionType` is `ONS` and locate the `topic-id`
6. Verify the correct topic is used by checking the topic name

```
oci ons topic get --topic-id=<topic id> --query data.{\"name:name\"} --output
table
```

Remediation:

From Console:

1. Go to the Events Service page: <https://console.us-ashburn-1.oraclecloud.com/events/rules>
2. Select the compartment that should host the rule
3. Click **Create Rule**
4. Provide a Display Name and Description
5. Create a Rule Condition by selecting **Networking** in the Service Name Drop-down and selecting **Network Security Group - Change Compartment, Network Security Group - Create, Network Security Group - Delete** and **Network Security Group - Update**
6. In the Actions section select **Notifications** as Action Type
7. Select the Compartment that hosts the Topic to be used.
8. Select the Topic to be used
9. Optionally add Tags to the Rule
10. Click **Create Rule**

From CLI:

1. Find the `topic-id` of the topic the Event Rule should use for sending Notifications by using the `topic` name and `Compartment OCID`

```
oci ons topic list --compartment-id=<compartment OCID> --all --query "data[?name=='<topic_name>']".{"name:name,topic_id:\"topic-id\""} --output table
```

2. Create a JSON file to be used when creating the Event Rule. Replace topic id, display name, description and compartment OCID.

```
{
  "actions":
  {
    "actions": [
      {
        "actionType": "ONS",
        "isEnabled": true,
        "topicId": "<topic id>"
      }
    ],
    "condition":
    "{ \"eventType\": [ \"com.oraclecloud.virtualnetwork.changenetworksecuritygroupcompartment\", \"com.oraclecloud.virtualnetwork.createnetworksecuritygroup\", \"com.oraclecloud.virtualnetwork.deletenetworksecuritygroup\", \"com.oraclecloud.virtualnetwork.updatenetworksecuritygroup\" ], \"data\": { } }",
    "displayName": "<display name>",
    "description": "<description>",
    "isEnabled": true,
    "compartmentId": "compartment OCID"
  }
}
```

3. Create the actual event rule

```
oci events rule create --from-json file://event_rule.json
```

4. Note in the JSON returned that it lists the parameters specified in the JSON file provided and that there is an OCID provided for the Event Rule

Additional Information:

- The console URL shown is for the Ashburn region. Your tenancy might have a different home region and thus console URL.
- The same Notification topic can be reused by many Event Rules.
- The generated notification will include an eventID that can be used when querying the Audit Logs in case further investigation is required.

CIS Controls:

Version 7

11.3 Use Automated Tools to Verify Standard Device Configurations and Detect Changes

Compare all network device configuration against approved security configurations defined for each network device in use and alert when any deviations are discovered.

3.13 Ensure a notification is configured for changes to network gateways (Manual)

Profile Applicability:

- Level 1

Description:

It is recommended to setup an Event Rule and Notification that gets triggered when Network Gateways are created, updated, deleted, attached, detached, or moved. This recommendation includes Internet Gateways, Dynamic Routing Gateways, Service Gateways, Local Peering Gateways, and NAT Gateways. Event Rules are compartment scoped and will detect events in child compartments, it is recommended to create the Event rule at the root compartment level.

Rationale:

Network Gateways act as routers between VCNs and the Internet, Oracle Services Networks, other VCNS, and on-premise networks. Monitoring and alerting on changes to Network Gateways will help in identifying changes to the security posture.

Impact:

There is no performance impact when enabling the above described features but depending on the amount of notifications sent per month there may be a cost associated.

Audit:

From Console:

1. Go to the Events Service page:
<https://console.us-ashburn-1.oraclecloud.com/events/rules>
2. Select the `Compartment` that hosts the rules
3. Find and click the `Rule` that handles `Network Gateways Changes` (if any)
4. Click the `Edit Rule` button and verify that the `RuleConditions` section contains a condition for the `Service Networking` and `Event Types`: `DRG - Create`,
`DRG - Delete`,
`DRG - Update`,
`DRG Attachment - Create`,
`DRG Attachment - Delete`,
`DRG Attachment - Update`,
`Internet Gateway - Create`,
`Internet Gateway - Delete`,

Internet Gateway - Update,
 Internet Gateway - Change Compartment,
 Local Peering Gateway - Create,
 Local Peering Gateway - Delete,
 Local Peering Gateway - Update,
 Local Peering Gateway - Change Compartment,
 NAT Gateway - Create,
 NAT Gateway - Delete,
 NAT Gateway - Update,
 NAT Gateway - Change Compartment,
 Service Gateway - Create,
 Service Gateway - Delete Begin,
 Service Gateway - Delete End,
 Service Gateway - Update,
 Service Gateway - Attach Service,
 Service Gateway - Detach Service,
 Service Gateway - Change Compartment

5. Verify that in the **Actions** section the **Action Type** contains: **Notifications** and that a valid **Topic** is referenced.

From CLI:

1. Find the OCID of the specific Event Rule based on Display Name and Compartment OCID

```
oci events rule list --compartment-id=<compartment OCID> --query "data
[?\"display-name\"=='<display name used>'].{\"id:id\"} --output table
```

2. List the details of a specific Event Rule based on the OCID of the rule.

```
oci events rule get --rule-id=<rule OCID>
```

3. In the JSON output locate the **Conditions** key value pair and verify that the following **Conditions** are present:

```

com.oraclecloud.virtualnetwork.createdrg,
com.oraclecloud.virtualnetwork.deletedrg,
com.oraclecloud.virtualnetwork.updatedrg,
com.oraclecloud.virtualnetwork.createdrgattachment,
com.oraclecloud.virtualnetwork.deletedrgattachment,
com.oraclecloud.virtualnetwork.updatedrgattachment,
com.oraclecloud.virtualnetwork.changeinternetgatewaycompartment,
com.oraclecloud.virtualnetwork.createinternetgateway,
com.oraclecloud.virtualnetwork.deleteinternetgateway,
com.oraclecloud.virtualnetwork.updateinternetgateway,
com.oraclecloud.virtualnetwork.changelocalpeeringgatewaycompartment,
com.oraclecloud.virtualnetwork.createlocalpeeringgateway,
com.oraclecloud.virtualnetwork.deletelocalpeeringgateway,
```



```
com.oraclecloud.virtualnetwork.updatelocalpeeringgateway,  
com.oraclecloud.natgateway.changenatgatewaycompartment,  
com.oraclecloud.natgateway.createnatgateway,  
com.oraclecloud.natgateway.deletenatgateway,  
com.oraclecloud.natgateway.updatenatgateway,  
com.oraclecloud.servicegateway.attachserviceid,  
com.oraclecloud.servicegateway.changeservicegatewaycompartment,  
com.oraclecloud.servicegateway.createservicegateway,  
com.oraclecloud.servicegateway.deleteservicegateway.begin,  
com.oraclecloud.servicegateway.deleteservicegateway.end,  
com.oraclecloud.servicegateway.detachserviceid,  
com.oraclecloud.servicegateway.updateservicegateway
```

4. Verify the value of the `is-enabled` attribute is `true`
5. In the JSON output verify that `actionType` is `ONS` and locate the `topic-id`
6. Verify the correct topic is used by checking the topic name

```
oci ons topic get --topic-id=<topic id> --query data>{"name:name"} --output  
table
```

Remediation:

From Console:

1. Go to the `Events Service` page: <https://console.us-ashburn-1.oraclecloud.com/events/rules>
2. Select the `compartment` that should host the rule
3. Click `Create Rule`
4. Provide a `Display Name` and `Description`
5. Create a Rule Condition by selecting `Networking` in the `Service Name Drop-down` and selecting `DRG - Create`,
`DRG - Delete`,
`DRG - Update`,
`DRG Attachment - Create`,
`DRG Attachment - Delete`,
`DRG Attachment - Update`,
`Internet Gateway - Create`,
`Internet Gateway - Delete`,
`Internet Gateway - Update`,
`Internet Gateway - Change Compartment`,
`Local Peering Gateway - Create`,
`Local Peering Gateway - Delete`,
`Local Peering Gateway - Update`,
`Local Peering Gateway - Change Compartment`,
`NAT Gateway - Create`,
`NAT Gateway - Delete`,
`NAT Gateway - Update`,
`NAT Gateway - Change Compartment`,

- Service Gateway - Create,
- Service Gateway - Delete Begin,
- Service Gateway - Delete End,
- Service Gateway - Update,
- Service Gateway - Attach Service,
- Service Gateway - Detach Service,
- Service Gateway - Change Compartment

6. In the **Actions** section select **Notifications** as **Action Type**
7. Select the **Compartment** that hosts the **Topic** to be used.
8. Select the **Topic** to be used
9. Optionally add **Tags** to the Rule
10. Click **Create Rule**

From CLI:

1. Find the **topic-id** of the topic the Event Rule should use for sending Notifications by using the **topic** name and **Compartment OCID**

```
oci ons topic list --compartment-id=<compartment OCID> --all --query "data[?name=='<topic_name>']".{"name:name,topic_id:\"topic-id\""} --output table
```

2. Create a JSON file to be used when creating the Event Rule. Replace topic id, display name, description and compartment OCID.

```
{
  "actions":
  {
    "actions": [
      {
        "actionType": "ONS",
        "isEnabled": true,
        "topicId": "<topic id>"
      }
    ]
  },
  "condition": "{\"eventType\": [\"com.oraclecloud.virtualnetwork.createdrg\", \"com.oraclecloud.virtualnetwork.deletedrg\", \"com.oraclecloud.virtualnetwork.updatedrg\", \"com.oraclecloud.virtualnetwork.createdrgattachment\", \"com.oraclecloud.virtualnetwork.deletedrgattachment\", \"com.oraclecloud.virtualnetwork.updatedrgattachment\", \"com.oraclecloud.virtualnetwork.changeinternetgatewaycompartment\", \"com.oraclecloud.virtualnetwork.createinternetgateway\", \"com.oraclecloud.virtualnetwork.deleteinternetgateway\", \"com.oraclecloud.virtualnetwork.updateinternetgateway\", \"com.oraclecloud.virtualnetwork.changelocalpeeringleadinggatewaycompartment\", \"com.oraclecloud.virtualnetwork.createlocalpeeringleadinggateway\", \"com.oraclecloud.virtualnetwork.deletelocalpeeringleadinggateway\", \"com.oraclecloud.virtualnetwork.updatelocalpeeringleadinggateway\", \"com.oraclecloud.natgateway.changenatgatewaycompartment\", \"com.oraclecloud.natgateway.createnatgateway\", \"com.oraclecloud.natgateway.deletenatgateway\", \"com.oraclecloud.natgateway.updatenatgateway\", \"com.oraclecloud.servicegateway.attachserviceid\", \"com.oraclecloud.servicegateway.changeservicegatewaycompartment\", \"com.oraclecloud.servicegateway.createservicegateway\", \"com.oraclecloud.servicegateway
```

```
.deleteservicegateway.begin\","com.oraclecloud.servicegateway.deleteservicegateway.end\","com.oraclecloud.servicegateway.detachserviceid\","com.oraclecloud.servicegateway.updateservicegateway\" ], \"data\":{} }",
  "displayName": "<display name>",
  "description": "<description>",
  "isEnabled": true,
  "compartmentId": "compartment OCID"
}
```

3. Create the actual event rule

```
oci events rule create --from-json file://event_rule.json
```

4. Note in the JSON returned that it lists the parameters specified in the JSON file provided and that there is an OCID provided for the Event Rule

Additional Information:

- The console URL shown is for the Ashburn region. Your tenancy might have a different home region and thus console URL.
- The same Notification topic can be reused by many Event Rules.
- The generated notification will include an eventID that can be used when querying the Audit Logs in case further investigation is required.

CIS Controls:

Version 7

11.3 Use Automated Tools to Verify Standard Device Configurations and Detect Changes

Compare all network device configuration against approved security configurations defined for each network device in use and alert when any deviations are discovered.

Appendix: Summary Table

| Control | | Set Correctly | |
|----------|--|--------------------------|--------------------------|
| | | Yes | No |
| 1 | Identity and Access Management | | |
| 1.1 | Ensure service level admins are created to manage resources of particular service (Manual) | <input type="checkbox"/> | <input type="checkbox"/> |
| 1.2 | Ensure permissions on all resources are given only to the tenancy administrator group (Manual) | <input type="checkbox"/> | <input type="checkbox"/> |
| 1.3 | Ensure IAM administrators cannot update tenancy Administrators group (Manual) | <input type="checkbox"/> | <input type="checkbox"/> |
| 1.4 | Ensure IAM password policy requires minimum length of 14 or greater (Manual) | <input type="checkbox"/> | <input type="checkbox"/> |
| 1.5 | Ensure IAM password policy requires at least one uppercase letter (Manual) | <input type="checkbox"/> | <input type="checkbox"/> |
| 1.6 | Ensure IAM password policy requires at least one lowercase letter (Manual) | <input type="checkbox"/> | <input type="checkbox"/> |
| 1.7 | Ensure IAM password policy requires at least one symbol (Manual) | <input type="checkbox"/> | <input type="checkbox"/> |
| 1.8 | Ensure IAM password policy requires at least one number (Manual) | <input type="checkbox"/> | <input type="checkbox"/> |
| 1.9 | Ensure IAM password policy expires passwords within 90 days or less (Manual) | <input type="checkbox"/> | <input type="checkbox"/> |
| 1.10 | Ensure IAM password policy prevents password reuse (Manual) | <input type="checkbox"/> | <input type="checkbox"/> |
| 1.11 | Ensure MFA is enabled for all users with a console password (Automated) | <input type="checkbox"/> | <input type="checkbox"/> |
| 1.12 | Ensure user API keys rotate within 90 days or less (Automated) | <input type="checkbox"/> | <input type="checkbox"/> |
| 1.13 | Ensure API keys are not created for tenancy administrator users (Automated) | <input type="checkbox"/> | <input type="checkbox"/> |
| 2 | Networking | | |
| 2.1 | Ensure no security lists allow ingress from 0.0.0.0/0 to port 22 (Automated) | <input type="checkbox"/> | <input type="checkbox"/> |
| 2.2 | Ensure no security lists allow ingress from 0.0.0.0/0 to port 3389 (Automated) | <input type="checkbox"/> | <input type="checkbox"/> |
| 2.3 | Ensure no network security groups allow ingress from 0.0.0.0/0 to port 22 (Manual) | <input type="checkbox"/> | <input type="checkbox"/> |
| 2.4 | Ensure no network security groups allow ingress from 0.0.0.0/0 to port 3389 (Manual) | <input type="checkbox"/> | <input type="checkbox"/> |
| 2.5 | Ensure the default security list of every VCN restricts all traffic except ICMP (Automated) | <input type="checkbox"/> | <input type="checkbox"/> |

| | | | |
|----------|---|--------------------------|--------------------------|
| 3 | Logging and Monitoring | | |
| 3.1 | Ensure audit log retention period is set to 365 days (Automated) | <input type="checkbox"/> | <input type="checkbox"/> |
| 3.2 | Ensure default tags are used on resources (Manual) | <input type="checkbox"/> | <input type="checkbox"/> |
| 3.3 | Create at least one notification topic and subscription to receive monitoring alerts (Manual) | <input type="checkbox"/> | <input type="checkbox"/> |
| 3.4 | Ensure a notification is configured for Identity Provider changes (Manual) | <input type="checkbox"/> | <input type="checkbox"/> |
| 3.5 | Ensure a notification is configured for IdP group mapping changes (Manual) | <input type="checkbox"/> | <input type="checkbox"/> |
| 3.6 | Ensure a notification is configured for IAM group changes (Manual) | <input type="checkbox"/> | <input type="checkbox"/> |
| 3.7 | Ensure a notification is configured for IAM policy changes (Manual) | <input type="checkbox"/> | <input type="checkbox"/> |
| 3.8 | Ensure a notification is configured for user changes (Manual) | <input type="checkbox"/> | <input type="checkbox"/> |
| 3.9 | Ensure a notification is configured for VCN changes (Manual) | <input type="checkbox"/> | <input type="checkbox"/> |
| 3.10 | Ensure a notification is configured for changes to route tables (Manual) | <input type="checkbox"/> | <input type="checkbox"/> |
| 3.11 | Ensure a notification is configured for security list changes (Manual) | <input type="checkbox"/> | <input type="checkbox"/> |
| 3.12 | Ensure a notification is configured for network security group changes (Manual) | <input type="checkbox"/> | <input type="checkbox"/> |
| 3.13 | Ensure a notification is configured for changes to network gateways (Manual) | <input type="checkbox"/> | <input type="checkbox"/> |

Appendix: Change History

| Date | Version | Changes for this version |
|-------------|---------|--------------------------|
| Aug 6, 2020 | 1.0.0 | Document Created |