CIS Benchmarks

CIS Apache Cassandra 3.11 Benchmark

v1.0.0 - 03-29-2019

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

Overview

This document, CIS Apache Cassandra Benchmark, provides prescriptive guidance for establishing a secure configuration posture for Apache Cassandra version 3.11. This guide was tested against Apache Cassandra running on CentOS Linux 7, but applies to other Linux distributions as well. To obtain the latest version of this guide, please visit http://benchmarks.cisecurity.org. If you have questions, comments, or have identified ways to improve this guide, please write us at feedback@cisecurity.org.

Intended Audience

This document is intended for system and application administrators, security specialists, auditors, help desk, and platform deployment personnel who plan to develop, deploy, assess, or secure solutions that incorporate Apache Cassandra.

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
`Stylized Monospace font`	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.
<italic font in brackets>	Italic texts set in angle brackets denote a variable requiring substitution for a real value.
Italic font	Used to denote the title of a book, article, or other publication.
Note	Additional information or caveats

Scoring Information

A scoring status indicates whether compliance with the given recommendation impacts the assessed target's benchmark score. The following scoring statuses are used in this benchmark:

Scored

Failure to comply with "Scored" recommendations will decrease the final benchmark score. Compliance with "Scored" recommendations will increase the final benchmark score.

Not Scored

Failure to comply with "Not Scored" recommendations will not decrease the final benchmark score. Compliance with "Not Scored" recommendations will not increase the final benchmark score.

Profile Definitions

The following configuration profiles are defined by this Benchmark:

- **Level 1 - Cassandra**

  Items in this profile apply to Apache Cassandra and intend to:

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

  Note: The intent of this profile is to include checks that can be assessed by remotely connecting to PostgreSQL. Therefore, file system-related checks are not contained in this profile.

- **Level 2 - Cassandra**

  This profile extends the "Level 1 - Cassandra" profile. Items in this profile apply to Apache Cassandra and 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.

  Note: The intent of this profile is to include checks that can be assessed by remotely connecting to PostgreSQL. Therefore, file system-related checks are not contained in this profile.

- **Level 1 - Cassandra on Linux**

  This profile extends the "Level 1 - Cassandra" profile. Items in this profile apply to Apache Cassandra running on Linux and intend to:

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

- **Level 2 - Cassandra on Linux**

  This profile extends the “Level 1 - Cassandra on Linux” profile. Items in this profile apply to Apache Cassandra running on Linux and 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:

Author
Joseph Testa

Editor
Tim Harrison CISSP, ICP, Center for Internet Security

Recommendations

1 Installation and Updates

This section contains recommendations related to installing and patching Cassandra.

1.1 Ensure a separate user and group exist for Cassandra (Not Scored)

Profile Applicability:

- Level 1 - Cassandra on Linux
- Level 2 - Cassandra on Linux

Description:

Create separate userid and group for Cassandra.

Rationale:

All processes need to run as a user with least privilege. This mitigates the potential impact of malware to the system.

Audit:

Logon to the server where Cassandra is installed.
To confirm existence of the group, execute the following command:

```
$ getent group | grep cassandra
```

To confirm existence of the user, execute the following command:

```
$ getent passwd | grep cassandra
```

If either the group or user do not exist, or if the user is not a member of the group, this is a finding.

Remediation:

Create a group for cassandra(if it does not already exist)

```
sudo groupadd cassandra
```

Create a user which is only used for running Cassandra and its related processes.

```
sudo useradd -m -d /home/cassandra -s /bin/bash -g cassandra -u
<USERID_NUMBER> cassandra
```

Replacing *<USERID_NUMBER>* with a number not already used on the server

References:

1.

CIS Controls:

Version 6

5.1 Minimize And Sparingly Use Administrative Privileges
Minimize administrative privileges and only use administrative accounts when they are required. Implement focused auditing on the use of administrative privileged functions and monitor for anomalous behavior.

Version 7

4 Controlled Use of Administrative Privileges
Controlled Use of Administrative Privileges

1.2 Ensure the latest version of Java is installed (Scored)

Profile Applicability:

- Level 1 - Cassandra on Linux
- Level 2 - Cassandra on Linux

Description:

A prerequisite to installing Cassandra is the installation of Java. The version of Java installed should be the most recent that is compatible with the organization's operational needs.

Rationale:

Using the most recent Java SDK version can help limit the possibilities for vulnerabilities in the software, the installation version applied during setup should be established according to the needs of the organization. Ensure you are using a release that is covered by a level of support which includes regular updates to address vulnerabilities.

Audit:

To verify that you have the correct version of java installed:

```
# java -version
java version "1.8.0_172"
Java(TM) SE Runtime Environment (build 1.8.0_172-b11)
```

If an old/unsupported version of Java is installed this is a finding.

Remediation:

1. Uninstall the old/unsupported version of Java, if present.
2. Download the latest compatible release of the Java JDK, or OpenJDK.
3. Follow the provided installation instructions to complete the install.

References:

1. http://www.oracle.com/technetwork/java/javase/downloads/index-jsp-138363.html#javasejdk
2. http://openjdk.java.net/
3. http://openjdk.java.net/install/index.html
4. http://cassandra.apache.org/doc/latest/getting_started/installing.html#prerequisites

5. https://www.java.com/en/download/help/index_installing.xml?os=All+Platforms&j=8&n=20

CIS Controls:

Version 6

2 Inventory of Authorized and Unauthorized Software
Inventory of Authorized and Unauthorized Software

Version 7

18.4 Only Use Up-to-date And Trusted Third-Party Components
Only use up-to-date and trusted third-party components for the software developed by the organization.

1.3 Ensure the latest version of Python is installed (Scored)

Profile Applicability:

- Level 1 - Cassandra on Linux

- Level 2 - Cassandra on Linux

Description:

A prerequisite to installing Cassandra is the installation of Python. The version of Python installed should be the most recent that is compatible with the organizations' operational needs.

Rationale:

Using the most recent Python can help limit the possibilities for vulnerabilities in the software, the installation version applied during setup should be established according to the needs of the organization. Ensure you are using a release that is covered by a level of support which includes regular updates to address vulnerabilities.

Audit:

To verify that you have the correct version of python installed:

```
# python -V
```

If an old/unsupported version of Python is installed this is a finding.

Remediation:

1. Uninstall the old/unsupported version of Python, if present.
2. Download the latest compatible release of the Python: https://www.python.org/downloads/
3. Follow the provided installation instructions to complete the install.

References:

1. https://www.python.org/downloads/
2. http://cassandra.apache.org/doc/latest/getting_started/installing.html#prerequisites

CIS Controls:

Version 6

2 Inventory of Authorized and Unauthorized Software
Inventory of Authorized and Unauthorized Software

Version 7

18.4 Only Use Up-to-date And Trusted Third-Party Components
Only use up-to-date and trusted third-party components for the software developed by the organization.

1.4 Ensure latest version of Cassandra is installed (Scored)

Profile Applicability:

- Level 1 - Cassandra on Linux

- Level 2 - Cassandra on Linux

Description:

The Cassandra installation version, along with the patches, should be the most recent that is compatible with organization's operational needs. When obtaining and installing software packages (typically via apt-get or you can compile the source code), it's imperative that packages (or the source code, tarball) are sourced only from valid and authorized repositories.

For Cassandra, a short list of valid repositories may include:

- The official apache cassandra website: http://cassandra.apache.org/
- DataStax Enterprise: https://www.datastax.com/

Rationale:

Using the most recent version of Cassandra can help limit the possibilities for vulnerabilities in the software, the installation version applied during setup should be established according to the needs of the organization. Ensure you are using a release that is covered by a level of support which includes regular updates to address vulnerabilities.

Audit:

To verify the version of Cassandra you have installed:

```
cassandra -v

3.11.2 (as of 6/8/2018)
```

If an old/unsupported version of Cassandra is installed this is a finding.

Remediation:

Upgrade to the latest version of the Cassandra software:
For each node in the cluster:

1. Using the nodetool drain command to push all memtables data to SSTables.
2. Stop Cassandra services.

3. Backup the data set and all of your Cassandra configuration files.
4. Download/Update Java if needed.
5. Download/Update Python if needed.
6. Download the binaries for the latest Cassandra revision from the Cassandra Download Page.
7. Install new version of Cassandra.
8. Configure new version of Cassandra, taking into account all of your previous settings in your config files(`cassandra.yml`, `cassandrea-env.sh`, etc).
9. Start Cassandra services.
10. Check logs for warnings, errors.
11. Using the nodetool to upgrade your SSTables.
12. Using the nodetool command to check status of cluster.

References:

1. http://cassandra.apache.org/doc/latest/getting_started/installing.html#prerequisites

CIS Controls:

Version 6

2 Inventory of Authorized and Unauthorized Software
Inventory of Authorized and Unauthorized Software

Version 7

18.4 Only Use Up-to-date And Trusted Third-Party Components
Only use up-to-date and trusted third-party components for the software developed by the organization.

1.5 Ensure the Cassandra service is run as a non-root user (Scored)

Profile Applicability:

- Level 1 - Cassandra on Linux
- Level 2 - Cassandra on Linux

Description:

Though Cassandra database may be run as root, it should run as another non-root user.

Rationale:

One of the best ways to reduce your exposure to attack is to create a unique, unprivileged user and group for the server application. A best practice is to follow is ensuring processes run with a user with least privilege.

Audit:

Logon to the server where Cassandra is running and run the following command

```
ps -aef | grep cassandra | grep java | cut -d' ' -f1
```

This will show who is running the Cassandra binary.
If the user is root or has excessive privileges then this is a finding.

Remediation:

Create a group for cassandra (if it does not already exist)

```
sudo groupadd cassandra
```

Create a user which is only used for running Cassandra and its related processes.

```
sudo useradd -m -d <DIRECTORY_WHERE_CASSANDRA_INSTALLED> -s /bin/bash -g
cassandra -u <USERID_NUMBER> cassandra
```

Replacing *`<DIRECTORY_WHERE_CASSANDRA_INSTALLED>`* with the full path of where Cassandra binaries are installed.

Replacing *`<USERID_NUMBER>`* with a number not already used on the server

CIS Controls:

Version 6

5.1 Minimize And Sparingly Use Administrative Privileges
Minimize administrative privileges and only use administrative accounts when they are required. Implement focused auditing on the use of administrative privileged functions and monitor for anomalous behavior.

Version 7

4 Controlled Use of Administrative Privileges
Controlled Use of Administrative Privileges

1.6 Ensure clocks are synchronized on all nodes (Not Scored)

Profile Applicability:

- Level 1 - Cassandra on Linux
- Level 2 - Cassandra on Linux

Description:

Enabling Network Time Protocol (NTP), or some equivalent way, to keep clocks on all nodes in sync is critical.

Rationale:

Cassandra decides which data is most current between all of the nodes in the cluster based on timestamps. It is paramount to ensure all clocks are in-sync, otherwise the most current data may not be returned or worse, marked for deletion.

Audit:

Depending on the Linux installation this may be checked by executing the following command on each node:

```
ps -aef | grep ntp

OR

ps -aef | grep chronyd
```

If NTP is not configured or clocks are out-of-sync then this is a finding.

Remediation:

Install and start the time protocol on every node in the Cassandra cluster.

CIS Controls:

Version 6

6.1 Use At Least Two Synchronized Time Sources For All Servers And Network Equipment
Include at least two synchronized time sources from which all servers and network equipment retrieve time information on a regular basis so that timestamps in logs are consistent.

Version 7

6.1 Utilize Three Synchronized Time Sources

Use at least three synchronized time sources from which all servers and network devices retrieve time information on a regular basis so that timestamps in logs are consistent.

2 Authentication and Authorization

This section contains recommendations related to Cassandra's authentication and authorization mechanisms.

2.1 Ensure that authentication is enabled for Cassandra databases (Scored)

Profile Applicability:

- Level 1 - Cassandra on Linux
- Level 2 - Cassandra on Linux

Description:

Authentication is pluggable in Cassandra and is configured using the `authenticator` setting in `cassandra.yaml`. Cassandra ships with two options included in the default distribution, `AllowAllAuthenticator` and `PasswordAuthenticator`. The default, `AllowAllAuthenticator`, performs no authentication checks and therefore requires no credentials. It is used to disable authentication completely. The second option, `PasswordAuthenticator`, stores encrypted credentials in a system table. This can be used to enable simple username/password authentication.

Rationale:

Authentication is a necessary condition of Cassandra's permissions subsystem, so if authentication is disabled then so are permissions. Failure to authenticate clients, users, and/or servers can allow unauthorized access to the Cassandra database and can prevent tracing actions back to their sources. The authentication mechanism should be implemented before anyone accesses the Cassandra server.

Audit:

Run the following command to verify whether authentication is enabled (authenticator values set to `PasswordAuthenticator`) on the Cassandra server.

The Cassandra configuration files can be found in the conf directory of tarballs. For packages, the configuration files will be located in `/etc/cassandra`.

```
cat cassandra.yaml | grep -in "authenticator:"
```

If `authenticator` is set to `AllowAllAuthenticator`, then this is a finding.

Remediation:

To enable the authentication mechanism:

1. Stop the Cassandra database.
2. Modify `cassandra.yaml` file to modify/add entry for authenticator: set it to `PasswordAuthenticator`
3. Start the Cassandra database.

Default Value:

```
authenticator: AllowAllAuthenticator
```

References:

1. http://cassandra.apache.org/doc/latest/getting_started/configuring.html
2. http://cassandra.apache.org/doc/latest/operating/security.html

CIS Controls:

Version 6

16 Account Monitoring and Control
Account Monitoring and Control

Version 7

14.7 Enforce Access Control to Data through Automated Tools
Use an automated tool, such as host-based Data Loss Prevention, to enforce access controls to data even when data is copied off a system.

2.2 Ensure that authorization is enabled for Cassandra databases (Scored)

Profile Applicability:

- Level 1 - Cassandra on Linux

- Level 2 - Cassandra on Linux

Description:

Authorization is pluggable in Cassandra and is configured using the `authorizer` setting in `cassandra.yaml`. Cassandra ships with two options included in the default distribution, `AllowAllAuthenticator` and `CassandraAuthorizer`. The default, `AllowAllAuthenticator` performs no checking which grants all permissions to all roles. The second option, `CassandraAuthorizer`, implements full permissions management functionality and stores its data in Cassandra system tables.

Rationale:

Authorizing roles is an important step towards ensuring only authorized access to the Cassandra database tables is permitted. It also provides the requisite means of implementing least privilege best practices. The authorization mechanism should be implemented before anyone accesses the Cassandra database.

Audit:

Run the following command to verify whether authorization is enabled (authorization values set to `CassandraAuthorizer`) on the Cassandra server.

The Cassandra configuration files can be found in the conf directory of tarballs. For packages, the configuration files will be located in /etc/cassandra.

```
cat cassandra.yaml | grep -in "authorizer:"
```

If `authorizer` is set to `AllowAllAuthorizer`, then this is a finding.

Remediation:

To enable the authorization mechanism:

1. Stop the Cassandra database.
2. Modify cassandra.yaml file to modify/add entry for authorization: set it to CassandraAuthorizer

3. Start the Cassandra database.

Default Value:

```
authorizer: AllowAllAuthorizer
```

References:

1. http://cassandra.apache.org/doc/latest/getting_started/configuring.html
2. http://cassandra.apache.org/doc/latest/operating/security.html

Notes:

The `authorizer` must be configured to `AllowAllAuthorizer` if `AllowAllAuthenticator` is the configured authenticator.

CIS Controls:

Version 6

16 Account Monitoring and Control
Account Monitoring and Control

Version 7

14.7 Enforce Access Control to Data through Automated Tools
Use an automated tool, such as host-based Data Loss Prevention, to enforce access controls to data even when data is copied off a system.

3 Access Control / Password Policies

This section contains recommendations related to Cassandra's password policies.

3.1 Ensure the cassandra and superuser roles are separate (Scored)

Profile Applicability:

- Level 1 - Cassandra
- Level 2 - Cassandra
- Level 1 - Cassandra on Linux
- Level 2 - Cassandra on Linux

Description:

The default installation of cassandra includes a superuser role named `cassandra`. This necessitates the creation of a separate role to be the superuser role.

Rationale:

Superuser permissions allow for the creation, deletion, and permission management of other users. Considering the Cassandra role is well known it should not be a superuser or one which is used for any administrative tasks.

Audit:

To verify the configuration, run the following query:

```
SELECT role FROM system_auth.roles WHERE is_superuser = True;
```

If `cassandra` or any unapproved role is returned, this is a finding.

Remediation:

To remediate a misconfiguration, perform the following steps:

1. Execute the following command:

```
create role '<NEW_ROLE_HERE>' with password='<NEW_PASSWORD_HERE>' and
login=TRUE and superuser=TRUE ;

grant all permissions on all keyspaces to <NEW_ROLE_HERE>;
```

Note: Replace *<NEW_ROLE_HERE>* with the desired role and *<NEW_PASSWORD_HERE>* with a password.

2. Verify the new role is working.
3. Remove the superuser role from the `cassandra` account by executing the following command:

```
UPDATE system_auth.roles SET is_superuser=null WHERE role='cassandra'
```

Impact:

The separate account must be created, assigned the superuser role, and tested for correct functionality prior to removing the superuser role from the `cassandra` account. Otherwise,

CIS Controls:

Version 6

16 Account Monitoring and Control
Account Monitoring and Control

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.

3.2 Ensure that the default password changed for the cassandra role (Scored)

Profile Applicability:

- Level 1 - Cassandra
- Level 2 - Cassandra
- Level 1 - Cassandra on Linux
- Level 2 - Cassandra on Linux

Description:

The cassandra role has a default password which must be changed.

Rationale:

Failure to change the default password for the cassandra role may pose a risk to the database in the form of unauthorized access.

Audit:

Connect to Cassandra database to verify whether the cassandra role has default password.

```
cqlsh -u cassandra -p cassandra
```

If the connection is successful this is a finding.

Remediation:

Change the password for the casssandra role by issuing the following command:

```
cqlsh -u cassandra -p cassandra

alter role 'cassandra' with password '<NEWPASSWORD_HERE>';
```

Where `<NEWPASSWORD_HERE>` is replaced with the password of your choosing.

Default Value:

cassandra

References:

1. http://cassandra.apache.org/doc/latest/operating/security.html

CIS Controls:

Version 6

16 Account Monitoring and Control
Account Monitoring and Control

Version 7

4.4 Use Unique Passwords
Where multi-factor authentication is not supported (such as local administrator, root, or service accounts), accounts will use passwords that are unique to that system.

3.3 Ensure there are no unnecessary roles or excessive privileges (Not Scored)

Profile Applicability:

- Level 1 - Cassandra

- Level 2 - Cassandra

- Level 1 - Cassandra on Linux

- Level 2 - Cassandra on Linux

Description:

Verify each role is require and has only the privileges needed to do its job.

Rationale:

Roles which are unneeded, have super user or other potentially excessive privileges may be an avenue for a hacker to gain access to or modify data in the database.

Audit:

As a superuser, retrieve all roles:

```
list roles;
```

Retrieve all permissions for all roles

```
select * from system_auth.role_permissions;
```

If there are any unnecessary roles or roles with excessive privileges this is a finding.

Remediation:

Remove any unnecessary roles and/or permissions in accordance with organizational needs.

References:

1. http://cassandra.apache.org/doc/latest/cql/security.html

CIS Controls:

Version 6

5 Controlled Use of Administration Privileges
Controlled Use of Administration Privileges

16.1 Perform Regular Account Reviews
Review all system accounts and disable any account that cannot be associated with a business process and owner.

Version 7

16.8 Disable Any Unassociated Accounts
Disable any account that cannot be associated with a business process or business owner.

3.4 Ensure that Cassandra is run using a non-privileged, dedicated service account (Scored)

Profile Applicability:

- Level 1 - Cassandra on Linux
- Level 2 - Cassandra on Linux

Description:

As with any service installed on a host, it can be provided with its own user context. Providing a dedicated user to the service provides the ability to precisely constrain the service within the larger host context.

Rationale:

Utilizing a non-privileged account for Cassandra to execute as may reduce the impact of a Cassandra-born vulnerability. A restricted account will be unable to access resources unrelated to Cassandra, such as operating system configurations.

Audit:

Execute the following command at a terminal prompt to assess this recommendation:

```
ps -ef | egrep "^cassandra.*$"
```

If no lines are returned, then this is a finding.

NOTE: It is assumed that the Cassandra user is `cassandra`. Additionally, you may consider running `sudo -l` as the Cassandra user or to check the `sudoers` file.

Remediation:

Create a user which is only used for running Cassandra and directly related processes. This user must not have administrative rights to the system.

CIS Controls:

Version 7

4 Controlled Use of Administrative Privileges
Controlled Use of Administrative Privileges

14 Controlled Access Based on the Need to Know
Controlled Access Based on the Need to Know

3.5 Ensure that Cassandra only listens for network connections on authorized interfaces (Not Scored)

Profile Applicability:

- Level 1 - Cassandra on Linux
- Level 2 - Cassandra on Linux

Description:

When `listen_address` is blank and `listen_interface` is commented out, this will be set automatically by `InetAddress.getLocalHost()`. Presuming the node is configured correctly, e.g. hostname, name resolution, etc., this will configure the node to use the address associated with the hostname. The `listen_address` must not be set to `0.0.0.0`.

Rationale:

Setting the address or interface to bind to will tell other Cassandra nodes to which address or interface to connect. This must be changed from the default in order for multiple nodes to be able to communicate.

Audit:

Check the value of `listen_address` or `listen_interface` in the `cassandra.yaml`. If `listen_address` is set `0.0.0.0` or a non-authorized address or interface is specified, this is a finding.

Remediation:

Set the `listen_address` or `listen_interface`, not both, in the `cassandra.yaml` to an authorized address or interface.

Default Value:

`listen_address`: `localhost`

`listen_interface`: `eth0`, but is commented out by default.

References:

1. http://cassandra.apache.org/doc/3.11/configuration/cassandra_config_file.html#listen-address

2. http://cassandra.apache.org/doc/3.11/configuration/cassandra_config_file.html#listen-interface

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.

3.6 Review User-Defined Roles (Not Scored)

Profile Applicability:

- Level 1 - Cassandra
- Level 2 - Cassandra
- Level 1 - Cassandra on Linux
- Level 2 - Cassandra on Linux

Description:

The `MEMBER_OF` column found in the `system_auth.roles` table shows roles granted to roles.

Rationale:

The `MEMBER_OF` column shows whoever has roles granted to roles and depending on the role and the privileges grant to the role should be limited . Limiting the accounts that have the certain roles reduces the chances that an attacker can exploit these capabilities.

Audit:

Execute the following SQL statement to audit this setting:

```
select role, can_login, member_of from system_auth.roles;
```

Looking for `can_login` which tells you that role can log into cassandra and `member_of` is when roles are granted to roles.

Remediation:

Looking at those users from the query that have member_of that is NOT null, decide if that user truly needs that role, if not, for each user, issue the following SQL statement (replace *`<is_member>`* with the value of `member_of` returned by the query in the audit procedure)

```
revoke <is_member> from role;
```

CIS Controls:

Version 7

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.

3.7 Review Superuser/Admin Roles (Not Scored)

Profile Applicability:

- Level 1 - Cassandra
- Level 2 - Cassandra
- Level 1 - Cassandra on Linux
- Level 2 - Cassandra on Linux

Description:

The `IS_SUPERUSER` privilege found in the `system_auth.roles` table governs who can control the entire Cassandra database and all of its data contained within.

Rationale:

The `IS_SUPERUSER` privilege allows whoever has it to do anything to the data and full administrator rights to the database, including changing passwords, creating, dropping roles. Limiting the accounts that have the `IS_SUPERUSER` role reduces the chances that an attacker can exploit these capabilities.

Audit:

Execute the following SQL statement to audit this setting:

```
select role, is_superuser from system_auth.roles;
```

Looking for `is_superuser = True`

Remediation:

Perform the following steps to remediate this setting:

```
alter role <role> with superuser=false;
```

Looking at those users from the query that have `is_superuser = True`, decide if that user truly needs that role, if not, for each user, issue the following SQL statement (replace *`<role>`* with the role name from the query):

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.

4 Auditing and Logging

This section contains recommendations related to Cassandra's audit and logging mechanisms.

4.1 Ensure that logging is enabled. (Scored)

Profile Applicability:

- Level 1 - Cassandra

- Level 2 - Cassandra

- Level 1 - Cassandra on Linux

- Level 2 - Cassandra on Linux

Description:

Apache Cassandra uses Logback for logging functionality. While this can be set using `nodetool setlogginglevel` changes made using this method will be reverted to the level specified in the `logback.xml` file the next time the process restarts.

The configurable logging levels are:

- `OFF`
- `TRACE`
- `DEBUG`
- `INFO` (Default)
- `WARN`
- `ERROR`

Rationale:

If logging is not enabled, issues may go undiscovered, and compromises and other incidents may occur without being quickly detected. It may also not be possible to provide evidence of compliance with security laws, regulations, and other requirements.

Audit:

Execute the following command to confirm the setting is correct:

```
$ nodetool getlogginglevels
Logger Name                                        Log Level
ROOT                                                    INFO
org.cisecurity.workbench                                WARN
```

If set to `OFF` then this is a finding.

Remediation:

To remediate this setting:

1. Edit the `logback-test.xml` if present; otherwise, edit the `logback.xml`

```
<configuration scan="true">

    <appender name="STDOUT"
class="ch.qos.logback.core.ConsoleAppender">
        <filter class="ch.qos.logback.classic.filter.ThresholdFilter">
            <level>INFO</level>
        </filter>
        <encoder>
            <pattern>%-5level [%thread] %date{ISO8601} %F:%L -
%msg%n</pattern>
        </encoder>
    </appender>

    <root level="INFO">
        <appender-ref ref="STDOUT" />
    </root>

    <logger name="org.cisecurity.workbench" level="WARN"/>
</configuration>
```

2. Restart the Apache Cassandra

Default Value:

`INFO`

References:

1. http://cassandra.apache.org/doc/latest/troubleshooting/reading_logs.html?highlight=logging
2. https://logback.qos.ch/manual/configuration.html

CIS Controls:

Version 7

6.3 Enable Detailed Logging
Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

4.2 Ensure that auditing is enabled (Not Scored)

Profile Applicability:

- Level 1 - Cassandra on Linux
- Level 2 - Cassandra on Linux

Description:

Audit logging in Cassandra logs every incoming CQL command request, Authentication (successful as well as unsuccessful login) to C* node. Currently, there are two implementations provided, the custom logger can be implemented and injected with the class name as a parameter in cassandra.yaml.

Rationale:

Unauthorized attempts to create, drop or alter users or data should be a concern.

Audit:

Open Source Version
Apache Cassandra versions up to 3.11.4 does not have auditing capabilities, it will be in version 4.x but that has not been released yet according to apache Cassandra website.
http://cassandra.apache.org/download/

Commercial Version
Allows via DataStax allows logging to filesystem log files using `logback`, or to a Cassandra table. When you turn on audit logging, the default is to write to `logback` filesystem log files. If using DataStax version you can verify auditing is turned on.

```
cat dse.yaml | grep "audit_logging_options"
```

If failure is enabled: `true` means success
Anything else is a finding

Remediation:

Open Source Version
Apache Cassandra versions up to 3.11.4 does not have auditing capabilities, it will be in version 4.x but that has not been released yet according to apache Cassandra website.
http://cassandra.apache.org/download/

Commercial Version

Open the `dse.yaml` file in a text editor

In the `audit_logging_options` section, set `enabled` to `true`.

```
# Audit logging options
audit_logging_options:
    enabled: true
```

You must also define where you want logging to go, add either of the following lines:
Set the logger option to either `CassandraAuditWriter`, which logs to a table, or `SLF4JAuditWriter`, which logs to the `SLF4J` logger.

References:

1. https://docs.datastax.com/en/datastax_enterprise/4.8/datastax_enterprise/sec/secAudit.html#secAudit

CIS Controls:

Version 7

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

5 Encryption

These recommendations pertain to encryption-related aspects of Cassandra.

5.1 Inter-node Encryption (Scored)

Profile Applicability:

- Level 1 - Cassandra on Linux
- Level 2 - Cassandra on Linux

Description:

Cassandra offers the option to encrypt data in transit between nodes on the cluster. By default, inter-node encryption is turned off.

Rationale:

Data being transferred on the wire should be encrypted to avoid network snooping, whether legitimate or not.

Audit:

Run the following command to verify whether inter-node encryption is enabled.

```
cat cassandra.yaml | grep -in "internode_encryption:"
```

Acceptable values are `all`, `dc` or `rack`. If the `internode_encryption` is set to `none`, this is a finding.

Note: The Cassandra configuration files can be found in the conf directory of tarballs. For packages, the configuration files will be located in `/etc/cassandra`.

Remediation:

The inter-node encryption should be implemented before anyone accesses the Cassandra server.
To enable the inter-node encryption mechanism:

1. Stop the Cassandra database.
2. If not done so already, build out your keystore and truststore.
3. Modify `cassandra.yaml` file to modify/add entry for `internode_encryption:` set it to `all`
4. Start the Cassandra database.

Default Value:

```
internode_encryption: none
```

References:

1. http://cassandra.apache.org/doc/latest/operating/security.html

CIS Controls:

Version 7

14.4 Encrypt All Sensitive Information in Transit
Encrypt all sensitive information in transit.

5.2 Client Encryption (Scored)

Profile Applicability:

- Level 1 - Cassandra on Linux
- Level 2 - Cassandra on Linux

Description:

Cassandra offers the option to encrypt data in transit between the client and nodes on the cluster. By default client encryption is turned off.

Rationale:

Data in transit between the client and node on the cluster should be encrypted to avoid network snooping, whether legitimate or not.

Audit:

The Cassandra configuration files can be found in the conf directory of tarballs. For packages, the configuration files will be located in `/etc/cassandra`.
Open up the `cassandra.yaml` file, look for client_encryption_options section.
Look for `enabled:` and `optional:`

```
enabled: true

optional: false
```

If neither is true, then all client connections are unencrypted which makes this a finding.

If enabled is true and optional is false, then all client connections must be encrypted which makes this not a finding.

If enabled is false and optional is true, then enabled wins and all client connections are unencrypted which makes this a finding.

If both are set to true, then both unencrypted and encrypted connections are allowed on the same port which makes this not a finding.

Remediation:

The client encryption should be implemented before anyone accesses the Cassandra server. To enable the client encryption mechanism:

1. Stop the Cassandra database.
2. If not done so already, build out your keystore and truststore.
3. Modify `cassandra.yaml` file to modify/add entries under `client_encryption_options:`

   ```
   set enabled: true

   set optional: false
   ```

   This will force all connections to be encrypted between client and node on the cluster.

4. Start the Cassandra database.

Default Value:

```
enabled: false

optional: false
```

References:

1. http://cassandra.apache.org/doc/latest/operating/security.html

CIS Controls:

Version 7

14.4 Encrypt All Sensitive Information in Transit
Encrypt all sensitive information in transit.

Appendix: Summary Table

Control		**Set Correctly**	
		Yes	No
1	**Installation and Updates**		
1.1	Ensure a separate user and group exist for Cassandra (Not Scored)	□	□
1.2	Ensure the latest version of Java is installed (Scored)	□	□
1.3	Ensure the latest version of Python is installed (Scored)	□	□
1.4	Ensure latest version of Cassandra is installed (Scored)	□	□
1.5	Ensure the Cassandra service is run as a non-root user (Scored)	□	□
1.6	Ensure clocks are synchronized on all nodes (Not Scored)	□	□
2	**Authentication and Authorization**		
2.1	Ensure that authentication is enabled for Cassandra databases (Scored)	□	□
2.2	Ensure that authorization is enabled for Cassandra databases (Scored)	□	□
3	**Access Control / Password Policies**		
3.1	Ensure the cassandra and superuser roles are separate (Scored)	□	□
3.2	Ensure that the default password changed for the cassandra role (Scored)	□	□
3.3	Ensure there are no unnecessary roles or excessive privileges (Not Scored)	□	□
3.4	Ensure that Cassandra is run using a non-privileged, dedicated service account (Scored)	□	□
3.5	Ensure that Cassandra only listens for network connections on authorized interfaces (Not Scored)	□	□
3.6	Review User-Defined Roles (Not Scored)	□	□
3.7	Review Superuser/Admin Roles (Not Scored)	□	□
4	**Auditing and Logging**		
4.1	Ensure that logging is enabled. (Scored)	□	□
4.2	Ensure that auditing is enabled (Not Scored)	□	□
5	**Encryption**		
5.1	Inter-node Encryption (Scored)	□	□
5.2	Client Encryption (Scored)	□	□

Appendix: Change History

Date	Version	Changes for this version
	1.0.0	Initial Release