CIS YugabyteDB 2.x

v1.0.0 - 12-28-2023

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

All CIS Benchmarks focus on technical configuration settings used to maintain and/or increase the security of the addressed technology, and they should be used in **conjunction** with other essential cyber hygiene tasks like:

- Monitoring the base operating system for vulnerabilities and quickly updating with the latest security patches
- Monitoring applications and libraries for vulnerabilities and quickly updating with the latest security patches

In the end, the CIS Benchmarks are designed as a key **component** of a comprehensive cybersecurity program.

This document, YugabyteDB 2.x Benchmark, provides prescriptive guidance for establishing a secure configuration posture for YugabyteDB 2.x. This guide was tested against YugabyteDB 2.x running on Ubuntu Linux, 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 YugabyteDB 2.x

Consensus Guidance

This CIS Benchmark was created using a consensus review process comprised of a global community of subject matter experts. The process combines real world experience with data-based information to create technology specific guidance to assist users to secure their environments. 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

Recommendation Definitions

The following defines the various components included in a CIS recommendation as applicable. If any of the components are not applicable it will be noted or the component will not be included in the recommendation.

Title

Concise description for the recommendation's intended configuration.

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

A collection of recommendations for securing a technology or a supporting platform. Most benchmarks include at least a Level 1 and Level 2 Profile. Level 2 extends Level 1 recommendations and is not a standalone profile. The Profile Definitions section in the benchmark provides the definitions as they pertain to the recommendations included for the technology.

Description

Detailed information pertaining to the setting with which the recommendation is concerned. In some cases, the description will include the recommended value.

Rationale Statement

Detailed reasoning for the recommendation to provide the user a clear and concise understanding on the importance of the recommendation.

Impact Statement

Any security, functionality, or operational consequences that can result from following the recommendation.

Audit Procedure

Systematic instructions for determining if the target system complies with the recommendation.

Remediation Procedure

Systematic instructions for applying recommendations to the target system to bring it into compliance according to the recommendation.

Default Value

Default value for the given setting in this recommendation, if known. If not known, either not configured or not defined will be applied.

References

Additional documentation relative to the recommendation.

CIS Critical Security Controls® (CIS Controls®)

The mapping between a recommendation and the CIS Controls is organized by CIS Controls version, Safeguard, and Implementation Group (IG). The Benchmark in its entirety addresses the CIS Controls safeguards of (v7) “5.1 - Establish Secure Configurations” and (v8) '4.1 - Establish and Maintain a Secure Configuration Process” so individual recommendations will not be mapped to these safeguards.

Additional Information

Supplementary information that does not correspond to any other field but may be useful to the user.

Profile Definitions

The following configuration profiles are defined by this Benchmark:

- **Level 1 - Yugabyte**

  Items in this profile apply to YugabyteDB running on Linux-based OS 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 - Yugabyte**

  Items in this profile apply to YugabyteDB running on Linux-based OS and intend to:

  - 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

- Davis Dinh, Yugabyte

Contributors

- Sergey Stelmakh , Yugabyte
- Aliaksandr Sinkevich, Yugabyte
- Sushant Mishra, Yugabyte
- Suranjan Kumar, Yugabyte
- Oleg Loginov, Yugabyte
- Tim Harrison CISSP, ICP, KMP, Center for Internet Security, New York
- Phil White , Center for Internet Security, New York

Recommendations

1 1 Installation and Patches

One of the best ways to ensure YugabyteDB security is to implement security updates as they come out, along with any applicable OS patches that will not interfere with system operations. It is additionally prudent to ensure the installed version has not reached end-of-life.

1.1 Ensure packages are obtained from authorized repositories (Manual)

Profile Applicability:

- Level 1 - Yugabyte

Description:

Standard Linux distributions, although possessing the requisite packages, often do not have YugabyteDB pre-installed. The installation process includes installing the binaries and the means to generate a data cluster. Package installation should include both the server and client packages. Contribution modules are optional depending upon one's architectural requirements (they are recommended though).

When obtaining and installing software packages (typically via tar.gz), it's imperative that packages are sourced only from valid and authorized repositories. For YugabyteDB, the canonical repositories for the official YugabyteDB packages is download.yugabyte.com.

Rationale:

Being open-source, YugabyteDB is available across the internet through package aggregators and providers. However, using invalid or unauthorized sources for packages can lead to implementing untested, defective, or malicious software.

Many organizations choose to implement a local software repository within their organization. Care must be taken to ensure that only valid and authorized packages are downloaded and installed into such local repositories.

From a security perspective, it's imperative to verify the YugabyteDB binary packages are sourced from a valid software repository.

Audit:

Identify and inspect configured repositories to ensure they are all valid and authorized sources of packages. The following is an example of a simple Linux command verifying integrity of YugabyteDB package by validating the checksum:

```
#!/bin/bash

# File name and URLs
FILE_NAME="yugabyte-2.20.0.0-b1-linux-x86_64.tar.gz"
FILE_URL="https://downloads.yugabyte.com/releases/2.20.0.0/$FILE_NAME"
SIGNATURE_URL="$FILE_URL.sha"


# Check if file exists locally
if [ ! -f "$FILE_NAME" ]; then
    echo "File $FILE_NAME does not exist locally. Please download
installation package to proceed. Exiting."
    exit 1
fi

# Compare checksums
if [ "$(wget -qO- $SIGNATURE_URL)" == "$(sha1sum $FILE_NAME | awk '{print
$1}')" ]; then
    echo "Checksums match. File is verified."
else
    echo "Checksums do not match. File might be corrupted or tampered with."
```

If the checksum of the binary packages does not match the checksum from download.yugabyte.com, this is a fail.

Remediation:

Alter the configured repositories so they only include valid and authorized sources of packages.

```
#!/bin/bash

# URL of the file to download
FILE_URL="https://downloads.yugabyte.com/releases/2.20.0.0/yugabyte-2.20.0.0-
b1-linux-x86_64.tar.gz"

# Download the file
wget $FILE_URL
```

Default Value:

N/A

References:

1. https://download.yugabyte.com/#linux
2. https://en.wikipedia.org/wiki/Checksum
3. https://en.wikipedia.org/wiki/SHA-1

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	16.5 Use Up-to-Date and Trusted Third-Party Software Components Use up-to-date and trusted third-party software components. When possible, choose established and proven frameworks and libraries that provide adequate security. Acquire these components from trusted sources or evaluate the software for vulnerabilities before use.		●	●
v7	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.		●	●

1.2 Ensure systemd Service Files Are Enabled (Manual)

Profile Applicability:

- Level 1 - Yugabyte

Description:

Confirm, and correct if necessary, the YugabyteDB `systemd` service is enabled.

Rationale:

Enabling the `systemd` service on the OS ensures the database service is active when a change of state occurs as in the case of a system startup or reboot.

Audit:

The default operating target on `systemd`-powered operating systems is typically "multi-user". One confirms the default target by executing the following:

```
$ whoami root
$ systemctl get-default multi-user.target
$ systemctl list-dependencies multi-user.target | grep -i postgres
```

If the intended YugabyteDB service is not registered as a dependency (or "want") of the default target (no output for the 3rd command above), this is a fail.

```
/etc/systemd/system/yugabyted.service

[Unit]
Description=YugabyteDB Server
After=network.target

[Service]
User=<USERNAME>
Group=<GROUPNAME>
ExecStart=/path/to/yugabyted start --daemon=false
ExecStop=/path/to/yugabyted stop
Restart=on-failure
RestartSec=10
LimitNOFILE=4096

[Install]
WantedBy=multi-user.target
```

Remediation:

Irrespective of package source, YugabyteDB services can be identified because it typically includes the text string "yugabyted". PGDG installs do not automatically register the service as a "want" of the default `systemd` target. Multiple instances of YugabyteDB services often distinguish themselves using a version number.

```
# whoami root
# systemctl enable yugabyted
Created symlink /etc/systemd/system/multi-user.target.wants/yugabyted.service
→ /usr/lib/systemd/system/yugabyted.service.
# systemctl list-dependencies multi-user.target | grep -i yugabyted
├─yugabyted.service
```

Default Value:

N/A

References:

1. https://man7.org/linux/man-pages/man1/systemctl.1.html
2. https://www.freedesktop.org/software/systemd/man/systemd.special.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	0.0 Explicitly Not Mapped Explicitly Not Mapped			
v7	0.0 Explicitly Not Mapped Explicitly Not Mapped			

1.3 Ensure Data Cluster Initialized Successfully (Automated)

Profile Applicability:

- Level 1 - Yugabyte

Description:

First-time installs of YugabyteDB require the instantiation of the database cluster. A database cluster is a collection of databases that are managed by a single server instance.

Rationale:

For the purposes of security, YugabyteDB enforces ownership and permissions of the data cluster such that:

- An initialized data cluster is owned by the UNIX account that created it.
- The data cluster cannot be accessed by other UNIX user accounts.
- The data-cluster cannot be created or owned by `root`
- The YugabyteDB process cannot be invoked by `root` nor any UNIX user account other than the owner of the data cluster.

Incorrectly instantiating the data cluster will result in a failed installation.

Audit:

Use the `yb-admin` binary to check the cluster status for all YB-Master servers.

```
.bin/yb-admin -init_master_addrs <master-ip-address:port> list_all_masters
```

Next, check the cluster status for all YB-TServer.

```
.bin/yb-admin -init_master_addrs <master ip address> list_all_tablet_servers
```

Remediation:

In the case of a cluster initialization failure, please visit YugabyteDB Troubleshooting guide: https://docs.yugabyte.com/preview/troubleshoot/

Default Value:

N/A

References:

1. https://docs.yugabyte.com/preview/deploy/
2. https://docs.yugabyte.com/preview/deploy/manual-deployment/
3. https://docs.yugabyte.com/preview/troubleshoot/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user’s need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.	●	●	●
v7	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 a separate user and group exist for YugabyteDB (Manual)

Profile Applicability:

- Level 2 - Yugabyte

Description:

Create separate userid and group for YugabyteDB.

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 YugabyteDB is installed.
To confirm existence of the group, execute the following command:

```
getent growd | grep yugabyte
```

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 YugabyteDB (if it does not already exist)

```
sudo groupadd yugabyte
```

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

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

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

Default Value:

N/A

References:

1. https://docs.yugabyte.com/preview/secure/authorization/create-roles/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user’s need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.	●	●	●
v7	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.	●	●	●

1.5 Ensure the latest version of Python is installed (Automated)

Profile Applicability:

- Level 2 - Yugabyte

Description:

A prerequisite to installing YugabyteDB 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 YugabyteDB 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:

```
#!/bin/bash

# Get the Python version
PYTHON_VERSION=$(python --version 2>&1 | awk '{print $2}')

# Use sort to compare versions
if [[ $(echo -e "3.11\n$PYTHON_VERSION" | sort -V | head -n1) ==
"$PYTHON_VERSION" && "$PYTHON_VERSION" != "3.11" ]]; then
    echo "This version of python ($PYTHON_VERSION) is not supported by
YugabyteDB."
    exit 1
```

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: www.python.org/downloads
3. Follow the provided installation instructions to complete the install.

Default Value:

N/A

References:

1. https://www.python.org/downloads/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	16.5 Use Up-to-Date and Trusted Third-Party Software Components Use up-to-date and trusted third-party software components. When possible, choose established and proven frameworks and libraries that provide adequate security. Acquire these components from trusted sources or evaluate the software for vulnerabilities before use.		●	●
v7	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.		●	●

1.6 Ensure latest version of YugabyteDB is installed (Automated)

Profile Applicability:

- Level 2 - Yugabyte

Description:

The YugabyteDB installation version, along with the patches, should be the most recent that is compatible with the 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 repository download.yugabyte.com.

Rationale:

Using the most recent version of YugabyteDB 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 YugabyteDB you have installed:

```
./bin/yugabyted version
/home/root/yugabyte-2.20.0.0/version_metadata.json
----------------------------------------------------------------------
|                              Version                               |
----------------------------------------------------------------------
| Version        : 2.20.0.0-b76                                      |
| Build Time     : 04 Nov 2023 02:23:30 UTC                          |
| Build Hash     : 0026607ed49516b4d5770f5479dd5d60d44710af          |
----------------------------------------------------------------------
```

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

Remediation:

Upgrade to the latest stable version of the YugabyteDB software:

Upgrade YB-Master

Use the following procedure to upgrade a YB-Master:

1. Stop the older version of the YB-Master process, as follows:
   `pkill yb-master`
2. Verify that you are on the directory of the new version, as follows:
   `cd /home/yugabyte/softwareyb-$VER/`
3. Start the newer version of the YB-Master process. For more information, see Start YB-Masters.
4. Verify in `http://<any-yb-master>:7000/` that all YB-Masters are alive.
5. Pause for approximately 60 seconds before upgrading the next YB-Master.

Upgrade YB-TServer

Use the following procedure to upgrade a YB-TServer:

1. Stop the older version of the yb-tserver process, as follows: `pkill yb-tserver`
2. Verify that you're on the directory of the new version, as follows: `cd /home/yugabyte/softwareyb-$VER/`
3. Start the newer version of the YB-TServer process.
4. Verify in `http://<any-yb-master>:7000/tablet-servers` to see if the new YB-TServer is alive and heart beating.
5. Pause for approximately 60 seconds before upgrading the next YB-TServer.

Promote AutoFlags

Use the yb-admin utility to promote the new AutoFlags, as follows:

```
./bin/yb-admin \
    -master_addresses <master-addresses> \
    promote_auto_flags
```

Upgrade the YSQL system catalog

After completing the YugabyteDB upgrade process, use the yb-admin utility to upgrade the YSQL system catalog, as follows:

```
./bin/yb-admin \
    -master_addresses <master-addresses> \
    upgrade_ysql
```

Expect to see the following output:

```
YSQL successfully upgraded to the latest version
```

In certain scenarios, a YSQL upgrade can take longer than 60 seconds, which is the default timeout value for `yb-admin`. If this happens, run the following command with a greater timeout value:

```
./bin/yb-admin \
    -master_addresses ip1:7100,ip2:7100,ip3:7100 \
    -timeout_ms 180000 \
    upgrade_ysql
```

Default Value:

N/A

References:

1. https://docs.yugabyte.com/preview/manage/upgrade-deployment/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	16.5 Use Up-to-Date and Trusted Third-Party Software Components Use up-to-date and trusted third-party software components. When possible, choose established and proven frameworks and libraries that provide adequate security. Acquire these components from trusted sources or evaluate the software for vulnerabilities before use.		●	●
v7	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.		●	●

1.7 Ensure the YugabyteDB service is run as a non-root user (Automated)

Profile Applicability:

- Level 2 - Yugabyte

Description:

Though YugabyteDB 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 YugabyteDB master service is running and run the following command

```
ps -aef | grep master| cut -d' ' -f1
```

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

Remediation:

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

```
sudo groupadd yugabyte
```

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

```
sudo groupadd -m -d <DIRECTORY_WHERE_YUGABYTEDB_INSTALLED> -s /bin/bash -g
yugabyte -u <USERID_NUMBER> yugabyte
```

Replacing `<DIRECTORY_WHERE_YUGABYTEDB_INSTALLED` with the full path of where YugabyteDB binaries are installed.
Replacing `<USERID_NUMBER>` with a number not already used on the server

Default Value:

N/A

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.	●	●	●
v7	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.	●	●	●

1.8 Ensure clocks are synchronized on all node (Manual)

Profile Applicability:

- Level 2 - Yugabyte

Description:

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

Rationale:

YugabyteDB 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 YugabyteDB cluster.

Default Value:

N/A

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.4 Standardize Time Synchronization Standardize time synchronization. Configure at least two synchronized time sources across enterprise assets, where supported.		●	●
v7	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 Directory and File Permissions

This section provides guidance on securing all operating system specific objects for YugabyteDB.

2.1 Ensure the file permissions mask is correct (Manual)

Profile Applicability:

- Level 1 - Yugabyte

Description:

Files are always created using a default set of permissions. File permissions can be restricted by applying a permissions mask called the umask. The yugabyte user account should use a umask of 077 to deny file access to all user accounts except the owner.

Rationale:

The Linux OS defaults the umask to `002`, which means the owner and primary group can read and write the file, and other accounts are permitted to read the file. Not explicitly setting the umask to a value as restrictive as `077` allows other users to read, write, or even execute files and scripts created by the `yugabyte` user account. The alternative to using a umask is explicitly updating file permissions after file creation using the command line utility `chmod` (a manual and error-prone process that is not advised).

Audit:

To view the mask's current setting, execute the following commands:

```
# whoami
root
# su - yugabyte
# whoami
yugabyte
# umask 0022
```

The umask must be `077` or more restrictive for the `yugabyte` user, otherwise, this is a fail.

Remediation:

Depending upon the yugabyte user's environment, the umask is typically set in the initialization file .bash_profile, but may also be set in .profile or .bashrc. To set the umask, add the following to the appropriate profile file:

```
# whoami yugabyte
# cd ~
# ls -ld .{bash_profile,profile,bashrc}
ls: cannot access '.bash_profile': No such file or directory
ls: cannot access '.profile': No such file or directory
ls: cannot access '.bashrc': No such file or directory
# echo "umask 077" >> .bash_profile
# source .bash_profile
# umask 0077
```

Default Value:

```
0022
```

References:

1. https://man7.org/linux/man-pages/man2/umask.2.html
2. https://man7.org/linux/man-pages/man1/umask.1p.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.	●	●	●
v7	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 Logging Monitoring and Auditing

This section provides guidance with respect to Yugabyte's auditing and logging behavior.

3.1 Yugabyte Structured Query Language (YSQL) Logging

This section provides guidance with respect to YugabyteDB’s logging behavior as it applies to security and auditing. YugabyteDB contains significantly more logging options that are not audit and/or security related (and as such, are not covered herein).

3.1.1 Logging Rationale (Manual)

Profile Applicability:

- Level 1 - Yugabyte

Description:

Having an audit trail is an important feature of any relational database system. You want enough detail to describe when an event of interest has started and stopped, what the event is/was, the event's cause, and what the event did/is doing to the system.

Ideally, the logged information is in a format permitting further analysis giving us new perspectives and insight.

The YugabyteSQL configuration file `ysql_pg.conf` is where all adjustable parameters can be set. A configuration file is created as part of the data cluster's creation i.e. initdb. The configuration file enumerates all tunable parameters and even though most of them are commented out it is understood that they are in fact active and at those very same documented values. The reason that they are commented out is to signify their default values. Uncommenting them will force the server to read these values instead of using the default values.

Rationale:

Audit:

Remediation:

References:

1. https://docs.yugabyte.com/preview/reference/configuration/yb-tserver/#ysql-pg-co

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.2 Collect Audit Logs Collect audit logs. Ensure that logging, per the enterprise's audit log management process, has been enabled across enterprise assets.	●	●	●
v8	8.5 Collect Detailed Audit Logs Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.		●	●
v7	6.2 Activate audit logging Ensure that local logging has been enabled on all systems and networking devices.	●	●	●
v7	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.		●	●

3.1.2 Ensure the log destinations are set correctly (Automated)

Profile Applicability:

- Level 1 - Yugabyte

Description:

YugabyteDB by default supports logging server messages to stderr.

Rationale:

If `log_destination` is not set, then any log messages generated by the core YugabyteSQL processes will be lost.

Audit:

Execute the following YSQL statement to confirm that the expected logging directory is specified:

```
yugabyte=# show log_destination;
 log_destination
-----------------
 stderr
(1 row)
```

The log destinations should comply with your organization's policies on logging. If all the expected log destinations are not set, this is a fail.

Remediation:

Use the `yb-tserver` binary and its flags to configure the YB-TServer server. Add the following GFLAG to YB-TServer:

```
./bin/yb-tserver --ysql_pg_conf_csv=“log_destination='stderr'”
```

Default Value:

`stderr`

References:

1. https://docs.yugabyte.com/preview/reference/configuration/
2. https://docs.yugabyte.com/preview/reference/configuration/yb-tserver/#ysql
3. https://docs.yugabyte.com/preview/secure/audit-logging/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.2 Collect Audit Logs Collect audit logs. Ensure that logging, per the enterprise's audit log management process, has been enabled across enterprise assets.	●	●	●
v8	8.5 Collect Detailed Audit Logs Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.		●	●
v7	6.2 Activate audit logging Ensure that local logging has been enabled on all systems and networking devices.	●	●	●
v7	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.		●	●

3.1.3 Ensure the filename pattern for log files is set correctly (Automated)

Profile Applicability:

- Level 1 - Yugabyte

Description:

The log_filename setting specifies the filename pattern for log files. The value for log_filename should match your organization's logging policy.

The value is treated as a strftime pattern, so %-escapes can be used to specify time-varying file names. The supported %-escapes are similar to those listed in the Open Group's strftime specification. If you specify a file name without escapes, you should plan to use a log rotation utility to avoid eventually filling the partition that contains log_directory. If there are any time-zone-dependent %-escapes, the computation is done in the zone specified by log_timezone. Also, the system's strftime is not used directly, so platform- specific (nonstandard) extensions do not work.

Rationale:

If log_filename is not set, then the value of log_directory is set to the default.

Audit:

Execute the following YSQL statement to confirm that the desired pattern is set:

```
yugabyte=# show log_filename;
          log_filename
--------------------------------
 postgresql-%Y-%m-%d_%H%M%S.log
(1 row)
```

Note: This example shows the use of the strftime %a escape. This creates seven log files, one for each day of the week (e.g. postgresql-Mon.log, postgresql-Tue.log, et al).

Remediation:

Use the `yb-tserver` binary and its flags to configure the YB-TServer server. Add the following GFLAG to YB-TServer:

```
./bin/yb-tserver --ysql_pg_conf_csv=“log_filename='postgresql-%a.log'”
```

Default Value:

`postgresql-%Y-%m-%d_%H%M%S.log`

References:

1. https://man7.org/linux/man-pages/man3/strftime.3.html
2. https://docs.yugabyte.com/preview/secure/audit-logging/
3. https://docs.yugabyte.com/preview/reference/configuration/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.2 Collect Audit Logs Collect audit logs. Ensure that logging, per the enterprise's audit log management process, has been enabled across enterprise assets.	●	●	●
v8	8.5 Collect Detailed Audit Logs Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.		●	●
v7	6.2 Activate audit logging Ensure that local logging has been enabled on all systems and networking devices.	●	●	●
v7	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.		●	●

3.1.4 Ensure the log file permissions are set correctly (Automated)

Profile Applicability:

- Level 1 - Yugabyte

Description:

The `log_file_mode` setting determines the file permissions for log files when `logging_collector` is enabled. The parameter value is expected to be a numeric mode specification in the form accepted by the `chmod` and `umask` system calls. (To use the customary octal format, the number must start with a `0`)

The permissions should be set to allow only the necessary access to authorized personnel. In most cases, the best setting is `0600`, so that only the server owner can read or write the log files. The other commonly useful setting is `0640`, allowing members of the owner's group to read the files, although to make use of that, you will need to alter the `log_directory` setting to store the log files outside the cluster data directory.

Rationale:

Log files often contain sensitive data. Allowing unnecessary access to log files may inadvertently expose sensitive data to unauthorized personnel.

Audit:

Execute the following YSQL statement to verify that the setting is consistent with organizational logging policy:

```
yugabyte=# show log_file_mode;
 log_file_mode
---------------
 0600
(1 row)
```

Remediation:

Use the `yb-tserver` binary and its flags to configure the YB-TServer server (in this example, setting it to `0660`). Add the following GFLAG to YB-TServer:

```
./bin/yb-tserver --ysql_pg_conf_csv=“log_file_mode='0660'”
```

Default Value:

`0600`

References:

1. https://docs.yugabyte.com/preview/reference/configuration/
2. https://docs.yugabyte.com/preview/secure/audit-logging/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.	●	●	●
v7	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.	●	●	●

3.1.5 Ensure 'log_truncate_on_rotation' is enabled (Automated)

Profile Applicability:

- Level 1 - Yugabyte

Description:

Enabling the `log_truncate_on_rotation` setting when `logging_collector` is enabled causes YugabyteDB to truncate (overwrite) existing log files with the same name during log rotation instead of appending to them.

Note: Truncation will occur only when a new file is being opened due to time-based rotation, not during server startup or size-based rotation (see later in this benchmark for size-based rotation details).

Rationale:

If this setting is disabled, pre-existing log files will be appended to if `log_filename` is configured in such a way that static names are generated. Enabling or disabling the truncation should only be decided when also considering the value of `log_filename` and `log_rotation_age/log_rotation_size`.

Audit:

Execute the following YSQL statement to verify how log_truncate_on_rotation is set:

```
yugabyte=# show log_truncate_on_rotation;
 log_truncate_on_rotation
--------------------------
 off
(1 row)
```

If it is not set to `on`, this is a fail (depending on your organization's logging policy).

Remediation:

Use the `yb-tserver` binary and its flags to configure the YB-TServer server. Add the following GFLAG to YB-TServer:

```
./bin/yb-tserver --ysql_pg_conf_csv=“log_truncate_on_rotation='on'”
```

Default Value:

`off`

References:

1. https://docs.yugabyte.com/preview/reference/configuration/
2. https://docs.yugabyte.com/preview/secure/audit-logging/

Additional Information:

Be sure to consider your organization's logging retention policies and the use of any external log consumption tools before deciding if truncation should be enabled or disabled.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.3 Ensure Adequate Audit Log Storage Ensure that logging destinations maintain adequate storage to comply with the enterprise’s audit log management process.	●	●	●
v7	6.4 Ensure adequate storage for logs Ensure that all systems that store logs have adequate storage space for the logs generated.		●	●

3.1.6 Ensure the maximum log file lifetime is set correctly (Automated)

Profile Applicability:

- Level 1 - Yugabyte

Description:

When `logging_collector` is enabled, the `log_rotation_age` parameter determines the maximum lifetime of an individual log file (depending on the value of `log_filename`). After this many minutes have elapsed, a new log file will be created via automatic log file rotation. Current best practices advise log rotation at least daily, but your organization's logging policy should dictate your rotation schedule.

Rationale:

Log rotation is a standard best practice for log management.

Audit:

Execute the following YSQL statement to verify the log rotation age is set to an acceptable value:

```
yugabyte=# show log_rotation_age;
 log_rotation_age
------------------
 1d
(1 row)
```

Remediation:

Use the `yb-tserver` binary and its flags to configure the YB-TServer server (in this example, setting it to `10d`). Add the following GFLAG to YB-TServer:

```
./bin/yb-tserver --ysql_pg_conf_csv=“log_rotation_age='10d'”
```

Default Value:

`1d` (one day)

References:

1. https://docs.yugabyte.com/preview/reference/configuration/
2. https://docs.yugabyte.com/preview/secure/audit-logging/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.3 Ensure Adequate Audit Log Storage Ensure that logging destinations maintain adequate storage to comply with the enterprise’s audit log management process.	●	●	●
v7	6.4 Ensure adequate storage for logs Ensure that all systems that store logs have adequate storage space for the logs generated.		●	●

3.1.7 Ensure the maximum log file size is set correctly (Automated)

Profile Applicability:

- Level 1 - Yugabyte

Description:

The `log_rotation_size` setting determines the maximum size of an individual log file. Once the maximum size is reached, automatic log file rotation will occur.

Rationale:

If this is set to zero, the size-triggered creation of new log files is disabled. This will prevent automatic log file rotation when files become too large, which could put log data at increased risk of loss (unless age-based rotation is configured).

Audit:

Execute the following YSQL statement to verify that `log_rotation_size` is set in compliance with the organization's logging policy:

```
yugabyte=# show log_rotation_size;
 log_rotation_size
-------------------
 10MB
(1 row)
```

Remediation:

Use the yb-tserver binary and its flags to configure the YB-TServer server (in this example, setting it to `1GB`). Add the following GFLAG to YB-TServer:

```
./bin/yb-tserver --ysql_pg_conf_csv=“log_rotation_size='1GB'”
```

Default Value:

`10 MB`

References:

1. https://docs.yugabyte.com/preview/reference/configuration/
2. https://docs.yugabyte.com/preview/secure/audit-logging/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.3 Ensure Adequate Audit Log Storage Ensure that logging destinations maintain adequate storage to comply with the enterprise's audit log management process.	●	●	●
v7	6.4 Ensure adequate storage for logs Ensure that all systems that store logs have adequate storage space for the logs generated.		●	●

3.1.8 Ensure the correct syslog facility is selected (Manual)

Profile Applicability:

- Level 1 - Yugabyte

Description:

The `syslog_facility` setting specifies the syslog "facility" to be used when logging to `syslog` is enabled. You can choose from any of the 'local' facilities:

- `LOCAL0`
- `LOCAL1`
- `LOCAL2`
- `LOCAL3`
- `LOCAL4`
- `LOCAL5`
- `LOCAL6`
- `LOCAL7`

Your organization's logging policy should dictate which facility to use based on the syslog daemon in use.

Rationale:

If not set to the appropriate facility, the YugabyteDB log messages may be intermingled with other applications' log messages, incorrectly routed, or potentially dropped (depending on your syslog configuration).

Audit:

Execute the following YSQL statement and verify that the correct facility is selected:

```
yugabyte=# show syslog_facility;
 syslog_facility
-----------------
 local0
(1 row)
```

Remediation:

Use the yb-tserver binary and its flags to configure the YB-TServer server (in this example, setting it to the `LOCAL1` facility). Add the following GFLAG to YB-TServer:

```
./bin/yb-tserver --ysql_pg_conf_csv=“syslog_facility='local1'”
```

Default Value:

`LOCAL0`

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.2 Collect Audit Logs Collect audit logs. Ensure that logging, per the enterprise's audit log management process, has been enabled across enterprise assets.	●	●	●
v7	6.2 Activate audit logging Ensure that local logging has been enabled on all systems and networking devices.	●	●	●

3.1.9 Ensure syslog messages are not suppressed (Manual)

Profile Applicability:

- Level 1 - Yugabyte

Description:

When logging to Syslog and this is on, then each message will be prefixed by an increasing sequence number.

Rationale:

Many modern Syslog implementations perform a log optimization and suppress repeated log entries while emitting “--- last message repeated N times ---”. In more modern Syslog implementations, repeated message suppression can be configured (for example, $RepeatedMsgReduction in rsyslog).

Impact:

If disabled, messages sent to Syslog could be suppressed and not logged. While a message is emitted stating that a given message was repeated and suppressed, the timestamp associated with these suppressed messages are lost, potentially damaging the recreation of an incident timeline.

Audit:

Execute the following YSQL statement and confirm that the syslog_sequence_numbers is enabled (on):

```
yugabyte=# show syslog_sequence_numbers;
 syslog_sequence_numbers
-------------------------
 on
(1 row)
```

Remediation:

Use the `yb-tserver` binary and its flags to configure the YB-TServer server. Add the following GFLAG to YB-TServer:

```
./bin/yb-tserver --ysql_pg_conf_csv=“syslog_sequence_numbers='off'”
```

Default Value:

`on`

References:

1. https://docs.yugabyte.com/preview/reference/configuration/
2. https://docs.yugabyte.com/preview/secure/audit-logging/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.5 Collect Detailed Audit Logs Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.		●	●
v7	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.		●	●

3.1.10 Ensure syslog messages are not lost due to size (Manual)

Profile Applicability:

- Level 1 - Yugabyte

Description:

YugabyteDB log messages can exceed 1024 bytes, which is a typical size limit for traditional Syslog implementations. When syslog_split_messages is off, YugabyteDB server log messages are delivered to the Syslog service as is, and it is up to the Syslog service to cope with the potentially bulky messages. When syslog_split_messages is on, messages are split by lines, and long lines are split so that they will fit into 1024 bytes.

If syslog is ultimately logging to a text file, then the effect will be the same either way, and it is best to leave the setting on, since most syslog implementations either cannot handle large messages or would need to be specially configured to handle them. But if syslog is ultimately writing into some other medium, it might be necessary or more useful to keep messages logically together.

Rationale:

Impact:

Depending on the Syslog server in use, log messages exceeding 1024 bytes may be lost or, potentially, cause the Syslog server processes to abort.

Audit:

Execute the following YSQL statement to confirm that long log messages are split when logging to Syslog:

```
yugabyte=# show syslog_split_messages;
 syslog_split_messages
-----------------------
 on
(1 row)
```

Remediation:

Use the `yb-tserver` binary and its flags to configure the YB-TServer server. Add the following GFLAG to YB-TServer:

```
./bin/yb-tserver --ysql_pg_conf_csv=“syslog_split_messages='off'”
```

Default Value:

`on`

References:

1. https://docs.yugabyte.com/preview/reference/configuration/
2. https://docs.yugabyte.com/preview/secure/audit-logging/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.5 Collect Detailed Audit Logs Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.		●	●
v7	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.		●	●

3.1.11 Ensure the program name for YugabyteDB syslog messages is correct (Automated)

Profile Applicability:

- Level 1 - Yugabyte

Description:

The `syslog_ident` setting specifies the program name used to identify YugabyteDB messages in syslog logs. An example of a possible program name is `yugabyte`.

Rationale:

If this is not set correctly, it may be difficult or impossible to distinguish YugabyteDB messages from other messages in Syslog logs.

Audit:

Execute the following YSQL statement to verify the program name is set correctly:

```
yugabyte=# show syslog_ident;
 syslog_ident
--------------
 postgres
(1 row)
```

Remediation:

Use the `yb-tserver` binary and its flags to configure the YB-TServer server (in this example, assuming a program name of `yugabyte`). Add the following GFLAG to YB-TServer:

```
./bin/yb-tserver --ysql_pg_conf_csv="syslog_ident='yugabyte'"
```

Default Value:

`postgres`

References:

1. https://tools.ietf.org/html/rfc3164#section-4.1.3
2. https://docs.yugabyte.com/preview/reference/configuration/
3. https://docs.yugabyte.com/preview/secure/audit-logging/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.5 Collect Detailed Audit Logs Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.		●	●
v7	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.		●	●

3.1.12 Ensure the correct messages are written to the server log (Automated)

Profile Applicability:

- Level 1 - Yugabyte

Description:

The `log_min_messages` setting specifies the message levels that are written to the server log. Each level includes all the levels that follow it. The lower the level (vertically, below), the fewer messages are sent.

Valid values are:

- `DEBUG5`
- `DEBUG4`
- `DEBUG3`
- `DEBUG2`
- `DEBUG1`
- `INFO`
- `NOTICE`
- `WARNING`
- `ERROR`
- `LOG`
- `FATAL`
- `PANIC`

`WARNING` is considered the best practice unless indicated otherwise by your organization's logging policy.

Rationale:

If this is not set to the correct value, too many messages or too few messages may be written to the server log.

Audit:

Execute the following YSQL statement to confirm the setting is correct:

```
yugabyte=# show log_min_messages;
 log_min_messages
------------------
 warning
(1 row)
```

Remediation:

Use the `yb-tserver` binary and its flags to configure the YB-TServer server (in this example, to set it to `warning`). Add the following GFLAG to YB-TServer:

```
./bin/yb-tserver --ysql_pg_conf_csv="log_min_messages='warning'"
```

Default Value:

`WARNING`

References:

1. https://docs.yugabyte.com/preview/reference/configuration/
2. https://docs.yugabyte.com/preview/secure/audit-logging/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.5 Collect Detailed Audit Logs Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.		●	●
v7	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.		●	●

3.1.13 Ensure the correct YSQL statements generating errors are recorded (Automated)

Profile Applicability:

- Level 1 - Yugabyte

Description:

The `log_min_error_statement` setting causes all YSQL statements generating errors at or above the specified severity level to be recorded in the server log. Each level includes all the levels that follow it. The lower the level (vertically, below), the fewer messages are recorded. Valid values are:

- `DEBUG5`
- `DEBUG4`
- `DEBUG3`
- `DEBUG2`
- `DEBUG1`
- `INFO`
- `NOTICE`
- `WARNING`
- `ERROR`
- `LOG`
- `FATAL`
- `PANIC`

`ERROR` is considered the best practice setting. Changes should only be made in accordance with your organization's logging policy. Note: To effectively turn off logging of failing statements, set this parameter to `PANIC`.

Rationale:

If this is not set to the correct value, too many erring YSQL statements or too few erring YSQL statements may be written to the server log.

Audit:

Execute the following YSQL statement to verify the setting is correct:

```
yugabyte=# show log_min_error_statement;
 log_min_error_statement
-------------------------
 error
(1 row)
```

If not configured to at least `ERROR`, this is a fail.

Remediation:

Use the `yb-tserver` binary and its flags to configure the YB-TServer server (in the example, to `warning`). Add the following GFLAG to YB-TServer:

```
./bin/yb-tserver --ysql_pg_conf_csv="log_min_error_statement='warning'"
```

Default Value:

`ERROR`

References:

1. https://docs.yugabyte.com/preview/reference/configuration/
2. https://docs.yugabyte.com/preview/secure/audit-logging/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.5 Collect Detailed Audit Logs Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.		●	●
v7	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.		●	●

3.1.14 Ensure 'debug_print_parse' is disabled (Automated)

Profile Applicability:

- Level 1 - Yugabyte

Description:

The `debug_print_parse` setting enables printing the resulting parse tree for each executed query. These messages are emitted at the `LOG` message level. Unless directed otherwise by your organization's logging policy, it is recommended this setting be disabled by setting it to `off`.

Rationale:

Enabling any of the `DEBUG` printing variables may cause the logging of sensitive information that would otherwise be omitted based on the configuration of the other logging settings.

Audit:

Execute the following YSQL statement to confirm the setting is correct:

```
yugabyte=# show debug_print_parse;
 debug_print_parse
--------------------
 off
(1 row)
```

If not configured to `off`, this is a fail.

Remediation:

Use the `yb-tserver` binary and its flags to configure the YB-TServer server. Add the following GFLAG to YB-TServer:

```
./bin/yb-tserver --ysql_pg_conf_csv="debug_print_parse='off'"
```

Default Value:

`off`

References:

1. https://docs.yugabyte.com/preview/reference/configuration/
2. https://docs.yugabyte.com/preview/secure/audit-logging/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.5 Collect Detailed Audit Logs Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.		●	●
v7	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.		●	●

3.1.15 Ensure 'debug_print_rewritten' is disabled (Automated)

Profile Applicability:

- Level 1 - Yugabyte

Description:

The `debug_print_rewritten` setting enables printing the query rewriter output for each executed query. These messages are emitted at the `LOG` message level. Unless directed otherwise by your organization's logging policy, it is recommended this setting be disabled by setting it to `off`.

Rationale:

Enabling any of the `DEBUG` printing variables may cause the logging of sensitive information that would otherwise be omitted based on the configuration of the other logging settings.

Audit:

Execute the following YSQL statement to confirm the setting is disabled:

```
yugabyte=# show debug_print_rewritten;
 debug_print_rewritten
-----------------------
 off
(1 row)
```

If not configured to `off`, this is a fail.

Remediation:

Use the `yb-tserver` binary and its flags to configure the YB-TServer server. Add the following GFLAG to YB-TServer:

```
./bin/yb-tserver --ysql_pg_conf_csv="debug_print_rewritten='off'"
```

Default Value:

`off`

References:

1. https://docs.yugabyte.com/preview/reference/configuration/
2. https://docs.yugabyte.com/preview/secure/audit-logging/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.5 Collect Detailed Audit Logs Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.		●	●
v7	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.		●	●

3.1.16 Ensure 'debug_print_plan' is disabled (Automated)

Profile Applicability:

- Level 1 - Yugabyte

Description:

The `debug_print_plan` setting enables printing the execution plan for each executed query. These messages are emitted at the `LOG` message level. Unless directed otherwise by your organization's logging policy, it is recommended this setting be disabled by setting it to `off`.

Rationale:

Enabling any of the `DEBUG` printing variables may cause the logging of sensitive information that would otherwise be omitted based on the configuration of the other logging settings.

Audit:

Execute the following YSQL statement to verify the setting is disabled:

```
yugabyte=# show debug_print_plan;
 debug_print_plan
------------------
 off
(1 row)
```

If not configured to `off`, this is a fail.

Remediation:

Use the `yb-tserver` binary and its flags to configure the YB-TServer server. Add the following GFLAG to YB-TServer:

```
./bin/yb-tserver --ysql_pg_conf_csv="debug_print_plan='off'"
```

Default Value:

`off`

References:

1. https://docs.yugabyte.com/preview/reference/configuration/
2. https://docs.yugabyte.com/preview/secure/audit-logging/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.5 Collect Detailed Audit Logs Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.		●	●
v7	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.		●	●

3.1.17 Ensure 'debug_pretty_print' is enabled (Automated)

Profile Applicability:

- Level 1 - Yugabyte

Description:

Enabling `debug_pretty_print` indents the messages produced by `debug_print_parse`, `debug_print_rewritten`, or `debug_print_plan` making them significantly easier to read.

Rationale:

If this setting is disabled, the "compact" format is used instead, significantly reducing the readability of the `DEBUG` statement log messages.

Impact:

Be advised that the aforementioned `DEBUG` printing options are **disabled**, but if your organizational logging policy requires them to be `on` then this option comes into play.

Audit:

Execute the following YSQL statement to confirm the setting is enabled:

```
yugabyte=# show debug_pretty_print;
 debug_pretty_print
--------------------
 on
(1 row)
```

If not configured to `on`, this is a fail.

Remediation:

Use the `yb-tserver` binary and its flags to configure the YB-TServer server. Add the following GFLAG to YB-TServer:

```
./bin/yb-tserver --ysql_pg_conf_csv="debug_pretty_print='on'"
```

Default Value:

`on`

References:

1. https://docs.yugabyte.com/preview/reference/configuration/
2. https://docs.yugabyte.com/preview/secure/audit-logging/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.5 Collect Detailed Audit Logs Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.		●	●
v7	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.		●	●

3.1.18 Ensure 'log_connections' is enabled (Automated)

Profile Applicability:

- Level 1 - Yugabyte

Description:

Enabling the `log_connections` setting causes each attempted connection to the server to be logged, as well as successful completion of client authentication. This parameter cannot be changed after the session start.

Rationale:

YugabyteDB does not maintain an internal record of attempted connections to the database for later auditing. It is only by enabling the logging of these attempts that one can determine if unexpected attempts are being made.

Note that enabling this without also enabling `log_disconnections` provides little value. Generally, you would enable/disable the pair together.

Audit:

Execute the following YSQL statement to verify the setting is enabled:

```
yugabyte=# show log_connections;
 log_connections
-----------------
 off
(1 row)
```

If not configured to `on`, this is a fail.

Remediation:

Use the `yb-tserver` binary and its flags to configure the YB-TServer server. Add the following GFLAG to YB-TServer:

```
./bin/yb-tserver --ysql_pg_conf_csv="log_filename="log_connections='on'"
```

Default Value:

`off`

References:

1. https://docs.yugabyte.com/preview/reference/configuration/
2. https://docs.yugabyte.com/preview/secure/audit-logging/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.5 Collect Detailed Audit Logs Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.		●	●
v7	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.		●	●

3.1.19 Ensure 'log_disconnections' is enabled (Automated)

Profile Applicability:

- Level 1 - Yugabyte

Description:

Enabling the `log_disconnections` setting logs the end of each session, including session duration. This parameter cannot be changed after the session start.

Rationale:

YugabyteDB does not maintain the beginning or ending of a connection internally for later review. It is only by enabling the logging of these that one can examine connections for failed attempts, 'over long' duration, or other anomalies.

Note that enabling this without also enabling `log_connections` provides little value. Generally, you would enable/disable the pair together.

Audit:

Execute the following YSQL statement to verify the setting is enabled:

```
yugabyte=# show log_disconnections;
 log_disconnections
--------------------
 off
(1 row)
```

If not configured to `on`, this is a fail.

Remediation:

Use the `yb-tserver` binary and its flags to configure the YB-TServer server. Add the following GFLAG to YB-TServer:

```
./bin/yb-tserver --ysql_pg_conf_csv="log_disconnections='on'"
```

Default Value:

`off`

References:

1. https://docs.yugabyte.com/preview/reference/configuration/
2. https://docs.yugabyte.com/preview/secure/audit-logging/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.5 Collect Detailed Audit Logs Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.		●	●
v7	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.		●	●

3.1.20 Ensure 'log_error_verbosity' is set correctly (Automated)

Profile Applicability:

- Level 1 - Yugabyte

Description:

The `log_error_verbosity` setting specifies the verbosity (amount of detail) of logged messages. Valid values are:

- `TERSE`
- `DEFAULT`
- `VERBOSE`

with each containing the fields of the level above it as well as additional fields.

`TERSE` excludes the logging of `DETAIL`, `HINT`, `QUERY`, and `CONTEXT` error information.

`VERBOSE` output includes the `SQLSTATE`, error code, and the source code file name, function name, and line number that generated the error.

The appropriate value should be set based on your organization's logging policy.

Rationale:

If this is not set to the correct value, too many details or too few details may be logged.

Audit:

Execute the following YSQL statement to verify the setting is correct:

```
yugabyte=# show log_error_verbosity;
 log_error_verbosity
---------------------
 default
(1 row)
```

If not configured to `verbose`, this is a fail.

Remediation:

Use the `yb-tserver` binary and its flags to configure the YB-TServer server (in this example, to `verbose`). Add the following GFLAG to YB-TServer:.

```
./bin/yb-tserver --ysql_pg_conf_csv="log_error_verbosity=’verbose'"
```

Default Value:

`DEFAULT`

References:

1. https://docs.yugabyte.com/preview/reference/configuration/
2. https://docs.yugabyte.com/preview/secure/audit-logging/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.5 Collect Detailed Audit Logs Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.		●	●
v7	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.		●	●

3.1.21 Ensure 'log_hostname' is set correctly (Automated)

Profile Applicability:

- Level 1 - Yugabyte

Description:

Enabling the `log_hostname` setting causes the hostname of the connecting host to be logged in addition to the host's IP address for connection log messages. Disabling the setting causes only the connecting host's IP address to be logged, and not the hostname. Unless your organization's logging policy requires hostname logging, it is best to disable this setting so as not to incur the overhead of DNS resolution for each statement that is logged.

Rationale:

Depending on your hostname resolution setup, enabling this setting might impose a non- negligible performance penalty. Additionally, the IP addresses that are logged can be resolved to their DNS names when reviewing the logs (unless dynamic hostnames are being used as part of your DHCP setup).

Audit:

Execute the following YSQL statement to verify the setting is correct:

```
yugabyte=# show log_hostname;
 log_hostname
--------------
 off
(1 row)
```

If not configured to `off`, this is a fail.

Remediation:

Use the `yb-tserver` binary and its flags to configure the YB-TServer server (in this example, to `verbose`). Add the following GFLAG to YB-TServer:

```
./bin/yb-tserver --ysql_pg_conf_csv="log_hostname=’verbose'"
```

Default Value:

`off`

References:

1. https://docs.yugabyte.com/preview/reference/configuration/
2. https://docs.yugabyte.com/preview/secure/audit-logging/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.5 Collect Detailed Audit Logs Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.		●	●
v7	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.		●	●

3.1.22 Ensure 'log_line_prefix' is set correctly (Automated)

Profile Applicability:

- Level 1 - Yugabyte

Description:

The `log_line_prefix` setting specifies a `printf`-style string that is prefixed to each logline. If blank, no prefix is used. You should configure this as recommended by the pgBadger development team unless directed otherwise by your organization's logging policy.

`%` characters begin "escape sequences" that are replaced with status information as outlined below. Unrecognized escapes are ignored. Other characters are copied straight to the logline. Some escapes are only recognized by session processes and will be treated as empty by background processes such as the main server process. Status information may be aligned either left or right by specifying a numeric literal after the `%` and before the option. A negative value will cause the status information to be padded on the right with spaces to give it a minimum width, whereas a positive value will pad on the left. Padding can be useful to aid human readability in log files.

Any of the following escape sequences can be used:

```
%a = application name
%u = user name
%d = database name
%r = remote host and port
%h = remote host
%b = backend type
%p = process ID
%t = timestamp without milliseconds
%m = timestamp with milliseconds
%n = timestamp with milliseconds (as a Unix epoch)
%Q = query ID (0 if none or not computed)
%i = command tag
%e = SQL state
%c = session ID
%l = session line number
%s = session start timestamp
%v = virtual transaction ID
%x = transaction ID (0 if none)
%q = stop here in non-session processes
%% = '%'
```

Rationale:

Properly setting `log_line_prefix` allows for adding additional information to each log entry (such as the user, or the database). Said information may then be of use in auditing or security reviews.

Audit:

Execute the following YSQL statement to verify the setting is correct:

```
yugabyte=# show log_line_prefix;
 log_line_prefix
-----------------
 %m [%p]
(1 row)
```

If the prefix does not at a minimum include `%m [%p]: [%l-1] db=%d`,`user=%u`,`app=%a`,`client=%h`, this is a fail.

Remediation:

Use the `yb-tserver` binary and its flags to configure the YB-TServer server. Add the following GFLAG to YB-TServer:

```
./bin/yb-tserver --ysql_pg_conf_csv="log_line_prefix=’%m [%p]: [%l-1]’"
```

Default Value:

`%m [%p]`

References:

1. https://pgbadger.darold.net/
2. https://docs.yugabyte.com/preview/reference/configuration/
3. https://docs.yugabyte.com/preview/secure/audit-logging/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.5 Collect Detailed Audit Logs Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.		●	●
v7	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.		●	●

3.1.23 Ensure 'log_statement' is set correctly (Automated)

Profile Applicability:

- Level 1 - Yugabyte

Description:

The `log_statement` setting specifies the types of YSQL statements that are logged. Valid values are:

- none
- ddl
- mod
- all

It is recommended this be set to ddl unless otherwise directed by your organization's logging policy.

`ddl` logs all data definition statements:

- `CREATE`
- `ALTER`
- `DROP`

`mod` logs all ddl statements, plus data-modifying statements:

- `INSERT`
- `UPDATE`
- `DELETE`
- `TRUNCATE`
- `COPY FROM`

(`PREPARE`, `EXECUTE`, and `EXPLAIN ANALYZE` statements are also logged if their contained command is of an appropriate type.)

For clients using extended query protocol, logging occurs when an Execute message is received, and values of the Bind parameters are included (with any embedded single-quote marks doubled).

Rationale:

Setting `log_statement` to align with your organization's security and logging policies facilitates later auditing and review of database activities.

Audit:

Execute the following YSQL statement to verify the setting is correct:

```
yugabyte=# show log_statement;
 log_statement
---------------
 none
(1 row)
```

If `log_statement` is set to `none` then this is a fail.

Remediation:

Use the `yb-tserver` binary and its flags to configure the YB-TServer server. Add the following GFLAG to YB-TServer:

```
./bin/yb-tserver --ysql_pg_conf_csv="log_line_prefix=’%m [%p]: [%l-1]’"
```

Default Value:

`none`

References:

1. https://docs.yugabyte.com/preview/reference/configuration/
2. https://docs.yugabyte.com/preview/secure/audit-logging/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.5 Collect Detailed Audit Logs Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.		●	●
v7	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.		●	●

3.1.24 Ensure 'log_timezone' is set correctly (Automated)

Profile Applicability:

- Level 1 - Yugabyte

Description:

The `log_timezone` setting specifies the time zone to use in timestamps within log messages. This value is cluster-wide, so that all sessions will report timestamps consistently. Unless directed otherwise by your organization's logging policy, set this to either `GMT` or `UTC`.

Rationale:

Log entry timestamps should be configured for an appropriate time zone as defined by your organization's logging policy to ensure a lack of confusion around when a logged event occurred.

Note that this setting affects only the timestamps present in the logs. It does not affect the time zone in use by the database itself (for example, `select now()`), nor does it affect the host's time zone.

Audit:

Execute the following YSQL statement:

```
yugabyte=# show log_timezone;
 log_timezone
--------------
 UTC
(1 row)
```

If `log_timezone` is not set to `GMT`, `UTC`, or as defined by your organization's logging policy this is a fail.

Remediation:

Use the `yb-tserver` binary and its flags to configure the YB-TServer server (in this example, setting it to `UTC`). Add the following GFLAG to YB-TServer:

```
./bin/yb-tserver --ysql_pg_conf_csv=“log_timezone='UTC'”
```

Default Value:

By default, the YugabyteDB packages will set this to match the server's timezone in the Operating System.

References:

1. https://docs.yugabyte.com/preview/reference/configuration/
2. https://docs.yugabyte.com/preview/secure/audit-logging/
3. https://en.wikipedia.org/wiki/Time_zone

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.5 Collect Detailed Audit Logs Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.		●	●
v7	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.		●	●

3.2 Ensure the YugbayteDB Audit Extension (pgAudit) is enabled (Automated)

Profile Applicability:

- Level 1 - Yugabyte

Description:

The YugabyteDB Audit Extension (pgAudit) provides detailed session and/or object audit logging via the standard YugabyteDB logging facility. The goal of pgAudit is to provide YugabyteDB users with the capability to produce audit logs often required to comply with government, financial, or ISO certifications.

Rationale:

Basic statement logging can be provided by the standard logging facility with `log_statement = all`. This is acceptable for monitoring and other uses but does not provide the level of detail generally required for an audit. It is not enough to have a list of all the operations performed against the database, it must also be possible to find particular statements that are of interest to an auditor. The standard logging facility shows what the user requested, while pgAudit focuses on the details of what happened while the database was satisfying the request.

When logging `SELECT` and `DML` statements, pgAudit can be configured to log a separate entry for each relation referenced in a statement. No parsing is required to find all statements that touch a particular table. In fact, the goal is that the statement text is provided primarily for deep forensics and should not be required for an audit.

Impact:

Depending on settings, it is possible for pgAudit to generate an enormous volume of logging. Be careful to determine exactly what needs to be audit logged in your environment to avoid logging too much.

Audit:

Execute the following YSQL statement to verify pgAudit is enabled:

```
yugabyte=# show shared_preload_libraries ;
               shared_preload_libraries
-------------------------------------------------------
 pg_stat_statements,yb_pg_metrics,pgaudit,pg_hint_plan
(1 row)
```

If the output does not contain `pgaudit`, this is a fail.
Next, verify that desired auditing components are enabled:

```
yugabyte=# show pgaudit.log;
 pgaudit.log
-------------
 none
(1 row)
```

If the output does not contain the desired auditing components, this is a fail.
The list below summarizes `pgAudit.log` components:
READ: `SELECT` and `COPY` when the source is a relation or a query.
WRITE: `INSERT`, `UPDATE`, `DELETE`, `TRUNCATE`, and `COPY` when the destination is a relation.
FUNCTION: Function calls and `DO` blocks.
ROLE: Statements related to roles and privileges: `GRANT`, `REVOKE`, `CREATE/ALTER/DROP ROLE`.
DDL: All `DDL` that is not included in the `ROLE` class.
MISC: Miscellaneous commands, e.g. `DISCARD`, `FETCH`, `CHECKPOINT`, `VACUUM`.

Remediation:

After configuring the YB-TServer and starting the cluster, create the pgAudit extension by executing the following YSQL statement:

```
yugabyte=# CREATE EXTENSION IF NOT EXISTS pgaudit;
CREATE EXTENSION
```

You only need to run this statement on a single node, and it will apply across your cluster.
Use the following steps to configure audit logging in a YugabyteDB cluster with bare minimum configurations.
Use the `yb-tserver` binary and its flags to configure the YB-TServer server. Add the following GFLAG to YB-TServer:

```
./bin/yb-tserver --
ysql_pg_conf_csv="pgaudit.log='DDL',pgaudit.log_level=notice,pgaudit.log_clie
nt=ON"
```

Default Value:

`pgAudit` is pre-bundled with standard YugabyteDB distribution but requires installation.

References:

1. https://docs.yugabyte.com/preview/secure/audit-logging/audit-logging-ysql/#enable-audit-logging
2. https://www.pgaudit.org/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.2 Collect Audit Logs Collect audit logs. Ensure that logging, per the enterprise's audit log management process, has been enabled across enterprise assets.	●	●	●
v7	6.2 Activate audit logging Ensure that local logging has been enabled on all systems and networking devices.	●	●	●

3.3 Ensure that auditing is enabled for YCQL (Manual)

Profile Applicability:

- Level 2 - Yugabyte

Description:

Audit logging can be used to record information about YCQL statements or events (such as login events) and log the records on a per-node basis into the YB-Tserver logs. Audit logging can be enabled on YugabyteDB cluster by setting the `ycql_enable_audit_log` TServer flag to `true`. By default, each TServer records all login events and YCQL commands issued to the server.

Audit record is logged before an operation attempts to be executed, and failures are audited as well. If an operation fails to execute, both operation execution and failure are logged. However, an error that happens during parsing or analysis of YCQL statement results only in an error audit record to be logged.

YCQL audit logging can be further customized using additional YB-TServer flags.

Rationale:

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

Audit:

Run the following command to verify whether auditing is enabled on the YCQL interface of the YugabyteDB server.

```
cat ts.config | grep ycql_enable_audit_log
```

If not configured to `true`, this is a fail.

Remediation:

Use the `yb-tserver` binary and its flags to configure the YB-TServer server. Add the following GFLAG to YB-TServer:

```
./bin/yb-tserver <options> --ycql_enable_audit_log=true
```

Default Value:

`false`

References:

1. https://docs.yugabyte.com/preview/secure/audit-logging/audit-logging-ycql/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.2 Collect Audit Logs Collect audit logs. Ensure that logging, per the enterprise’s audit log management process, has been enabled across enterprise assets.	●	●	●
v7	6.2 Activate audit logging Ensure that local logging has been enabled on all systems and networking devices.	●	●	●

4 User Access and Authorization

The capability to use database resources at a given level, or user authorization rules, allows for user manipulation of the various parts of the Yugabyte Structured Query Language (YSQL) database. These authorizations must be structured to block unauthorized use and/or corruption of vital data and services by setting restrictions on user capabilities.

4.1 Ensure sudo is configured correctly (Manual)

Profile Applicability:

- Level 1 - Yugabyte

Description:

It is common to have more than one authorized individual administering the YugabyteDB service at the Operating System level. It is also quite common to permit login privileges to individuals on a YugabyteDB host who otherwise are not authorized to access the server's data cluster and files. Administering the YugabyteDB data cluster, as opposed to its data, is to be accomplished via a localhost login of a regular UNIX user account. Access to the `yugabyte` superuser account is restricted in such a manner as to interdict unauthorized access. `sudo` satisfies the requirements by escalating ordinary user account privileges as the YugabyteDB RDBMS superuser.

Rationale:

Without `sudo`, there would be no capabilities to strictly control access to the superuser account nor to securely and authoritatively audit its use.

Audit:

Log in as an Operating System user authorized to escalate privileges and test the sudo invocation by executing the following:

```
whoami user1
# groups user1
# sudo su - yugabyte
[sudo] password for user1:
user1 is not in the sudoers file. This incident will be reported.
```

As shown above, user1 has not been added to the `/etc/sudoers` file or made a member of any group listed in the `/etc/sudoers` file. Whereas:

```
# whoami user2
# groups user2 dba
# sudo su - yugabyte [sudo] password for user2:
# whoami
yugabyte
```

This shows that the `user2` user is configured properly for `sudo` access by being a member of the `dba` group.

Remediation:

As superuser `root`, execute the following commands:

```
# echo '%dba ALL= /bin/su - yugabyte' > /etc/sudoers.d/yugabyte
# chmod 600 /etc/sudoers.d/yugabyte
```

This grants any Operating System user that is a member of the `dba` group to `use sudo su - yugabyte` to become the `yugabyte` user.
Ensure that all Operating System user's that need such access are members of the group.

Default Value:

N/A

References:

1. https://www.sudo.ws/man/1.8.15/sudo.man.html
2. https://www.sudo.ws/man/1.8.17/visudo.man.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.4 Restrict Administrator Privileges to Dedicated Administrator Accounts Restrict administrator privileges to dedicated administrator accounts on enterprise assets. Conduct general computing activities, such as internet browsing, email, and productivity suite use, from the user's primary, non-privileged account.	●	●	●
v7	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.	●	●	●

4.2 Ensure excessive administrative privileges are revoked (Manual)

Profile Applicability:

- Level 1 - Yugabyte

Description:

With respect to YugabyteDB administrative YSQL commands, only superusers should have elevated privileges. YugabyteDB regular, or application, users should not possess the ability to create roles, create new databases, manage replication, or perform any other action deemed privileged. Typically, regular users should only be granted the minimal set of privileges commensurate with managing the application:

- DDL (`create table, create view, create index, etc.`)
- DML (`select, insert, update, delete`)

Further, it has become best practice to create separate roles for DDL and DML. Given an application called 'payroll', one would create the following users:

- `payroll_owner`
- `payroll_user`

Any DDL privileges would be granted to the `payroll_owner` account only, while DML privileges would be given to the `payroll_user` account only. This prevents accidental creation/altering/dropping of database objects by application code that runs as the `payroll_user` account.

Rationale:

By not restricting global administrative commands to superusers only, regular users granted excessive privileges may execute administrative commands with unintended and undesirable results.

Audit:

First, inspect the privileges granted to the database superuser (identified here as `yugabyte`) using the display command ysqlsh -c "\du yugabyte" to establish a baseline for granted administrative privileges. Based on the output below, the yugabyte superuser can create roles, create databases, manage replication, and bypass row-level security (RLS):

```
# whoami
yugabyte
# ysqlsh -c "\du yugabyte"
```

Now, let's inspect the same information for a mock regular user called appuser using the display command ysqlsh -c "\du appuser". The output confirms that regular user appuser has the same elevated privileges as system administrator user yugabyte. This is a fail.
While this example demonstrated excessive administrative privileges granted to a single user, a comprehensive audit should be conducted to inspect all database users for excessive administrative privileges. This can be accomplished via either of the commands below.

```
# whoami
yugabyte
# ysqlsh -c "\du *"
# ysqlsh -c "select * from pg_user order by usename"
```

NOTE Using `\du *` will show all the default YugabyteDB roles (e.g. `pg_monitor`) as well as any 'normal' roles. This is expected, and should not be cause for alarm.

Remediation:

If any regular or application users have been granted excessive administrative rights, those privileges should be removed immediately via the Yugabyte ALTER ROLE YSQL command. Using the same example above, the following YSQL statements revoke all unnecessary elevated administrative privileges from the regular user appuser:

```
# whoami
yugabyte
# ysqlsh -c "ALTER ROLE appuser NOSUPERUSER;" ALTER ROLE
# ysqlsh -c "ALTER ROLE appuser NOCREATEROLE;" ALTER ROLE
# ysqlsh -c "ALTER ROLE appuser NOCREATEDB;" ALTER ROLE
# ysqlsh -c "ALTER ROLE appuser NOREPLICATION;" ALTER ROLE
# ysqlsh -c "ALTER ROLE appuser NOBYPASSRLS;" ALTER ROLE
# ysqlsh -c "ALTER ROLE appuser NOINHERIT;" ALTER ROLE
```

Verify the `appuser` now passes your check by having no defined Attributes:

```
# whoami
yugabyte
# ysqlsh -c "\du appuser"
List of roles
Role name | Attributes | Member of
        +       +        appuser        |        | {}
```

Default Value:

N/A

References:

1. https://www.postgresql.org/docs/current/static/sql-revoke.html
2. https://www.postgresql.org/docs/current/static/sql-createrole.html
3. https://www.postgresql.org/docs/current/static/sql-alterrole.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.4 Restrict Administrator Privileges to Dedicated Administrator Accounts Restrict administrator privileges to dedicated administrator accounts on enterprise assets. Conduct general computing activities, such as internet browsing, email, and productivity suite use, from the user's primary, non-privileged account.	●	●	●
v7	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.	●	●	●

4.3 Ensure excessive function privileges are revoked (Automated)

Profile Applicability:

- Level 1 - Yugabyte

Description:

In certain situations, to provide the required functionality, YugabyteDB needs to execute internal logic (stored procedures, functions, triggers, etc.) and/or external code modules with elevated privileges. However, if the privileges required for execution are at a higher level than the privileges assigned to organizational users invoking the functionality applications/programs, those users are indirectly provided with greater privileges than assigned by their organization. This is known as privilege elevation. Privilege elevation must be utilized only where necessary. Execute privileges for application functions should be restricted to authorized users only.

Rationale:

Ideally, all application source code should be vetted to validate interactions between the application and the logic in the database, but this is usually not possible or feasible with available resources even if the source code is available. The DBA should attempt to obtain assurances from the development organization that this issue has been addressed and should document what has been discovered. The DBA should also inspect all application logic stored in the database (in the form of functions, rules, and triggers) for excessive privileges.

Audit:

Functions in YugabyteDB can be created with the `SECURITY DEFINER` option. When `SECURITY DEFINER` functions are executed by a user, said function is run with the privileges of the user who **created it**, not the user who is *running* it.
To list all functions that have `SECURITY DEFINER`, run the following YSQL:

```
# whoami root
# sudo su - yugabyte
# ysqlsh -c "SELECT nspname, proname, proargtypes, prosecdef, rolname,
proconfig FROM pg_proc p JOIN pg_namespace n ON p.pronamespace = n.oid JOIN
pg_authid a ON a.oid = p.proowner WHERE prosecdef OR NOT proconfig IS NULL;"
```

In the query results, a `prosecdef` value of '`t`' on a row indicates that that function uses privilege elevation.
If elevation privileges are utilized which are not required or are expressly forbidden by organizational guidance, this is a fail.

Remediation:

Where possible, revoke `SECURITY DEFINER` on YugabyteDB functions. To change a `SECURITY DEFINER` function to `SECURITY INVOKER`, run the following YSQL:

```
# whoami root
# sudo su - yugabyte
# ysqlsh -c "ALTER FUNCTION [functionname] SECURITY INVOKER;"
```

If it is not possible to revoke `SECURITY DEFINER`, ensure the function can be executed by only the accounts that absolutely need such functionality:

```
yugabyte=# SELECT proname, proacl FROM pg_proc WHERE proname =
'delete_customer';
proname |       proacl
        +        delete_customer |
{=X/postgres,postgres=X/postgres,appwriter=X/postgres}
(1 row)
yugabyte=# REVOKE EXECUTE ON FUNCTION delete_customer(integer,boolean) FROM
appreader;
REVOKE
yugabyte=# SELECT proname, proacl FROM pg_proc WHERE proname =
'delete_customer';
proname |       proacl
        +        delete_customer | {=X/postgres,postgres=X/postgres}
(1 row)
```

Based on the output above, `appreader=X/postgres` no longer exists in the `proacl` column results returned from the query and confirms `appreader` is no longer granted execute privilege on the function.

Default Value:

N/A

References:

1. https://docs.yugabyte.com/preview/secure/authorization/
2. https://docs.yugabyte.com/preview/secure/authorization/ysql-grant-permissions/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.	●	●	●
v7	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.	●	●	●

4.4 Ensure excessive DML privileges are revoked (Manual)

Profile Applicability:

- Level 1 - Yugabyte

Description:

DML (insert, update, delete) operations at the table level should be restricted to only authorized users. YugabyteDB manages table-level DML permissions via the GRANT statement.

Rationale:

Excessive DML grants can lead to unprivileged users changing or deleting information without proper authorization.

Audit:

To audit excessive DML privileges, take an inventory of all users defined in the cluster using the `\du+ *` YSQL command, as well as all tables defined in the database using the `\dt *.*` YSQL command. Furthermore, the intersection matrix of tables and user grants can be obtained by querying system catalogs `pg_tables` and `pg_user`. Note that in YugabyteDB, users are defined cluster-wide across all databases, while schemas and tables are specific to a particular database. Therefore, the commands below should be executed for each defined database in the cluster. With this information, inspect database table grants and determine if any are excessive for defined database users.

```
yugabyte=# -- display all users defined in the cluster
yugabyte=# \x Expanded display is on.
yugabyte=# \du+ *

List of roles
-[ RECORD 1 ]
Role name       | pg_database_owner Attributes | Cannot login Member of      |
{}
Description |
-[ RECORD 2 ]
Role name       | pg_execute_server_program Attributes | Cannot login
Member of       | {} Description |
-[ RECORD 3 ]
Role name       | pg_monitor Attributes | Cannot login
Member of       | {pg_read_all_settings,pg_read_all_stats,pg_stat_scan_tables}
Description |
-[ RECORD 4 ]
Role name       | pg_read_all_data Attributes | Cannot login Member of       |
{}
Description |
-[ RECORD 5 ]
Role name       | pg_read_all_settings Attributes | Cannot login
Member of       | {} Description |
-[ RECORD 6 ]
Role name       | pg_read_all_stats Attributes | Cannot login Member of      |
{}
Description |
-[ RECORD 7 ]
Role name       | pg_read_server_files Attributes | Cannot login
Member of       | {} Description |
-[ RECORD 8 ]
Role name       | pg_signal_backend Attributes | Cannot login Member of      |
{}
Description |
-[ RECORD 9 ]
Role name       | pg_stat_scan_tables Attributes | Cannot login
Member of       | {} Description |
-[ RECORD 10 ]
Role name       | pg_write_all_data Attributes | Cannot login Member of      |
{}
Description |
-[ RECORD 11 ]
Role name       | pg_write_server_files Attributes | Cannot login
Member of       | {} Description |
-[ RECORD 12 ]
Role name       | postgres
Attributes | Superuser, Create role, Create DB, Replication, Bypass RLS
Member of       | {}
Description |
yugabyte=# -- display all schema.tables created in current database
yugabyte=# \x
Expanded display is off. postgres=# \dt+ *.*
List of relations
|              |       |       |       |       |Descr- Schema |       Name
        |Type | Owner |Persistence| Size |iption
        +       +       +       +       +       +
information_schema|sql_features       |table|postgres| permanent |104 kB|
information_schema|sql_        |table|postgres| permanent |48 kB |
yugabyte=# -- query all tables and user grants in current database yugabyte=#
-- the system catalogs 'information_schema' and 'pg_catalog' are excluded
yugabyte=# select t.schemaname, t.tablename, u.usename,
has_table_privilege(u.usename, t.tablename, 'select') as select,
has_table_privilege(u.usename, t.tablename, 'insert') as insert,
has_table_privilege(u.usename, t.tablename, 'update') as update,
has_table_privilege(u.usename, t.tablename, 'delete') as delete
from pg_tables t, pg_user u
where t.schemaname not in ('information_schema','pg_catalog');

schemaname | tablename | usename | select | insert | update | delete
        +       +       +       +       +       +        (0 rows)
```

For the example below, we illustrate using a single `table customer` and two application users `appwriter` and `appreader`. The intention is for `appwriter` to have full select, insert, update, and delete rights and for `appreader` to only have select rights. We can query these privileges with the example below using the `has_table_privilege` function and filtering for just the table and roles in question.

```
yugabyte=# select t.tablename, u.usename, has_table_privilege(u.usename,
t.tablename, 'select') as select, has_table_privilege(u.usename, t.tablename,
'insert') as insert, has_table_privilege(u.usename, t.tablename, 'update') as
update, has_table_privilege(u.usename, t.tablename, 'delete') as delete
from    pg_tables t, pg_user u where t.tablename = 'customer'
and     u.usename in ('appwriter','appreader');

tablename | usename | select | insert | update | delete
        +       +       +       +       +
```

As depicted, both users have full privileges for the customer table. This is a fail. When inspecting database-wide results for all users and all table grants, employ a comprehensive approach. Collaboration with application developers is paramount to collectively determine only those database users that require specific DML privileges and on which tables.

Remediation:

If a given database user has been granted excessive DML privileges for a given database table, those privileges should be revoked immediately using the REVOKE YSQL command.
Continuing with the example above, remove unauthorized grants for appreader user using the REVOKE statement and verify the Boolean values are now false.

```
yugabyte=# REVOKE INSERT, UPDATE, DELETE ON TABLE customer FROM appreader;
REVOKE

yugabyte=# select t.tablename, u.usename, has_table_privilege(u.usename,
t.tablename, 'select') as select, has_table_privilege(u.usename, t.tablename,
'insert') as insert, has_table_privilege(u.usename, t.tablename, 'update') as
update, has_table_privilege(u.usename, t.tablename, 'delete') as delete
from    pg_tables t, pg_user u where t.tablename = 'customer'
and     u.usename in ('appwriter','appreader');

tablename | usename | select | insert | update | delete
        +       +       +       +       +
```

Default Value:

The table owner/creator has full privileges; all other users must be explicitly granted access.

References:

1. https://docs.yugabyte.com/preview/secure/authorization/
2. https://docs.yugabyte.com/preview/secure/authorization/ysql-grant-permissions/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.	●	●	●
v7	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.	●	●	●

4.5 Ensure Row Level Security (RLS) is configured correctly (Manual)

Profile Applicability:

- Level 1 - Yugabyte

Description:

In addition to the YSQL-standard privilege system available through `GRANT`, tables can have row security policies that restrict, on a per-user basis, which individual rows can be returned by normal queries or inserted, updated, or deleted by data modification commands. This feature is also known as Row Level Security (RLS). By default, tables do not have any policies, so if a user has access privileges to a table according to the YSQL privilege system, all rows within it are equally available for querying or updating. Row security policies can be specific to commands, to roles, or to both. A policy can be specified to apply to `ALL` commands, or to any combination of `SELECT`, `INSERT`, `UPDATE`, or `DELETE`. Multiple roles can be assigned to a given policy, and normal role membership and inheritance rules apply. If you use RLS and apply restrictive policies to certain users, it is important that the `Bypass RLS` privilege not be granted to any unauthorized users. This privilege overrides RLS- enabled tables and associated policies. Generally, only superusers and elevated users should possess this privilege.

Rationale:

If RLS policies and privileges are not configured correctly, users could perform actions on tables that they are not authorized to perform, such as inserting, updating, or deleting rows.

Audit:

The first step for an organization is to determine which, if any, database tables require RLS. This decision is a matter of business processes and is unique to each organization. To discover which, if any, database tables have RLS enabled, execute the following query. If any table(s) should have RLS policies applied, but do not appear in query results, then this is a fail.

```
yugabyte=# SELECT oid, relname, relrowsecurity FROM pg_class WHERE
relrowsecurity IS TRUE;
```

Based on the results below we can see RLS is not enabled. Assuming this table should be RLS enabled but is not, this is a fail.

```
yugabyte=# SELECT oid, relname, relrowsecurity FROM pg_class WHERE relname =
'passwd';
oid | relname | relrowsecurity
       +       +        24679 | passwd | f
(1 row)
```

Further inspection of RLS policies is provided via the system catalog `pg_policy`, which records policy details including table OID, policy name, applicable commands, the roles assigned a policy, and the `USING` and `WITH CHECK` clauses. Finally, RLS and associated policies (if implemented) may also be viewed using the standard `ysqlsh` display command `\d+ schema.table` which lists RLS information as part of the table description.

Should you implement Row Level Security and apply restrictive policies to certain users, it's imperative that you check each user's role definition via the `ysqlsh` display command `\du` and ensure unauthorized users have not been granted `Bypass RLS` privilege as this would override any RLS enabled tables and associated policies. If unauthorized users do have `Bypass RLS` granted then resolve this using the `ALTER ROLE <user> NOBYPASSRLS;` command.

Remediation:

In this example, we are using the using the example `passwd` table. We will create three database roles to illustrate the workings of RLS:

```
yugabyte=# CREATE ROLE admin; CREATE ROLE
yugabyte=# CREATE ROLE bob; CREATE ROLE
yugabyte=# CREATE ROLE alice; CREATE ROLE
```

Now, we will insert known data into the `passwd` table:

```
yugabyte=# INSERT INTO passwd VALUES
('admin','xxx',0,0,'Admin','111-222-3333',null,'/root','/bin/dash');
INSERT 0 1
yugabyte=# INSERT INTO passwd VALUES
('bob','xxx',1,1,'Bob','123-456-7890',null,'/home/bob','/bin/zsh');
INSERT 0 1
yugabyte=# INSERT INTO passwd VALUES
('alice','xxx',2,1,'Alice','098-765-4321',null,'/home/alice','/bin/zsh');
INSERT 0 1
```

And we will enable RLS on the table:

```
yugabyte=# ALTER TABLE passwd ENABLE ROW LEVEL SECURITY; ALTER TABLE
```

Now that RLS is enabled, we need to define one or more policies. Create the administrator policy and allow it access to all rows:

```
yugabyte=# CREATE POLICY admin_all ON passwd TO admin USING (true) WITH CHECK
(true);
CREATE POLICY
```

Create a policy for normal users to *view* all rows:

```
yugabyte=# CREATE POLICY all_view ON passwd FOR SELECT USING (true); CREATE
POLICY
```

Create a policy for normal users that allows them to update only their own rows and to limit what values can be set for their login shell:

```
yugabyte=# CREATE POLICY user_mod ON passwd FOR UPDATE USING (current_user =
user_name)
WITH CHECK (
current_user = user_name AND
shell IN ('/bin/bash','/bin/sh','/bin/dash','/bin/zsh','/bin/tcsh')
);
CREATE POLICY
```

Grant all the normal rights on the table to the `admin` user:

```
yugabyte=# GRANT SELECT, INSERT, UPDATE, DELETE ON passwd TO admin; GRANT
```

Grant only select access on non-sensitive columns to everyone:

```
yugabyte=# GRANT SELECT
(user_name, uid, gid, real_name, home_phone, extra_info, home_dir, shell) ON
passwd TO public;
GRANT
```

Grant update to only the sensitive columns:

```
yugabyte=# GRANT UPDATE
(pwhash, real_name, home_phone, extra_info, shell) ON passwd TO public;
GRANT
```

Ensure that no one has been granted `Bypass RLS` inadvertently, by running the `psql` display command `\du+`. If unauthorized users do have `Bypass RLS` granted then resolve this using the `ALTER ROLE <user> NOBYPASSRLS;` command.
You can now verify that 'admin', 'bob', and 'alice' are properly restricted by querying the `passwd` table as each of these roles.

Default Value:

N/A

References:

1. https://www.postgresql.org/docs/current/static/ddl-rowsecurity.html
2. https://www.postgresql.org/docs/current/static/sql-alterrole.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.	●	●	●
v7	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.	●	●	●

4.6 Make use of predefined roles (Manual)

Profile Applicability:

- Level 1 - Yugabyte

Description:

YugabyteDB provides a set of predefined roles that provide access to certain, commonly needed, privileged capabilities and information. Administrators can GRANT these roles to users and/or other roles in their environment, providing those users with access to the specified capabilities and information.

Rationale:

In keeping with the principle of least privilege, judicious use of the YugabyteDB predefined roles can greatly limit the access to privileged, or superuser, access.

Audit:

Review the list of all database roles that have `superuser` access and determine if one or more of the predefined roles would suffice for the needs of that role:

```
yugabyte=# select rolname from pg_roles where rolsuper is true;
 rolname
----------
 postgres
 yugabyte
(2 rows)
```

Remediation:

If you've determined that one or more of the predefined roles can be used, simply `GRANT` it:

```
yugabyte=# GRANT pg_monitor TO yugabyte; GRANT ROLE
```

And then remove superuser from the account:

```
yugabyte=# ALTER ROLE yugabyte NOSUPERUSER; ALTER ROLE
yugabyte=# select rolname from pg_roles where rolsuper is true; rolname
yugabyte (1 row)
```

Default Value:

The following predefined roles exist in YugabyteDB:

- pg_read_all_settings Read all configuration variables, even those normally visible only to superusers.
- pg_read_all_stats Read all pg_stat_* views and use various statistics related extensions, even those normally visible only to superusers.
- pg_stat_scan_tables Execute monitoring functions that may take ACCESS SHARE locks on tables, potentially for a long time.
- pg_monitor Read/execute various monitoring views and functions. This role is a member of pg_read_all_settings, pg_read_all_stats and pg_stat_scan_tables.
- pg_signal_backend Signal another backend to cancel a query or terminate its session.
- pg_read_server_files Allow reading files from any location the database can access on the server with COPY and other file-access functions.
- pg_write_server_files Allow writing to files in any location the database can access on the server with COPY and other file-access functions.
- pg_execute_server_program Allow executing programs on the database server as the user the database runs as with COPY and other functions which allow executing a server-side program.
- yb_fdw Role that allows non-superuser users to CREATE, ALTER, and DROP foreign data wrappers.
- yb_extension Role that allows non-superuser users to create YugabyteDB extensions.

Administrators can grant access to these roles to users using the GRANT command.

References:

1. https://docs.yugabyte.com/preview/yugabyte-cloud/cloud-secure-clusters/cloud-users/#ysql-default-roles-and-users
2. https://www.postgresql.org/docs/current/predefined-roles.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user’s need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.	●	●	●
v7	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.	●	●	●

5 Access Control / Password Policies

This section contains recommendations related to Yugabyte Cloud Query Language (YCQL) password policies.

5.1 Ensure that authentication is enabled for YCQL interface of the YugabyteDB (Automated)

Profile Applicability:

- Level 2 - Yugabyte

Description:

YCQL authentication is based on roles. Roles can be created with superuser, non-superuser, and login privileges. New roles can be created, and existing ones altered or dropped by administrators using YCQL commands.

Rationale:

Authentication is a necessary condition of YugabyteDB'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 YugabyteDB database and can prevent tracing actions back to their sources. The authentication mechanism should be implemented before anyone accesses the YCQL interface of the YugabyteDB server.

Audit:

Run the following command to verify whether authentication is enabled on the YCQL interface of the YugabyteDB server.
The YugabyteDB configuration files can be found in the conf directory of tarballs. For packages, the configuration files will be located in `/home/user/var/conf`.

```
cat ts.config | grep use_cassandra_authentication
```

If `use_cassandra_authentication` is set to `false`, then this is a finding.

Remediation:

Use the yb-tserver binary and its flags to configure the YB-TServer server. Add the following GFLAG to YB-TServer:

```
./bin/yb-tserver --use_cassandra_authentication=true
```

Default Value:

`false`

References:

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

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user’s need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.	●	●	●
v7	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.	●	●	●

5.2 Ensure that the default password changed for the cassandra role (Manual)

Profile Applicability:

- Level 2 - Yugabyte

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 YugabyteDB database via YCQL to verify whether the cassandra role has default password.

```
./bin/ycqlsh -u cassandra -p cassandra
```

If the connection is successful this is a finding.

Remediation:

Change the password for the `cassandra` role by issuing the following command:

```
cassandra@ycqlsh> ALTER ROLE cassandra WITH PASSWORD = 'new_password';
```

Where 'new_password' is replaced with the password of your choosing.
Verify the default password no longer works:

```
./bin/ycqlsh -u cassandra -p cassandra

Connection error:
  ... Provided username 'cassandra' and/or password are incorrect ...
```

You can now connect to the cluster using the new password:

```
./bin/ycqlsh -u cassandra -p new_password
```

Default Value:

`cassandra`

References:

1. https://docs.yugabyte.com/preview/secure/enable-authentication/ycql/#change-default-admin-credentials

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	4.7 Manage Default Accounts on Enterprise Assets and Software Manage default accounts on enterprise assets and software, such as root, administrator, and other pre-configured vendor accounts. Example implementations can include: disabling default accounts or making them unusable.	●	●	●
v8	5.2 Use Unique Passwords Use unique passwords for all enterprise assets. Best practice implementation includes, at a minimum, an 8-character password for accounts using MFA and a 14-character password for accounts not using MFA.	●	●	●
v7	4.2 Change Default Passwords Before deploying any new asset, change all default passwords to have values consistent with administrative level accounts.	●	●	●
v7	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.		●	●

5.3 Ensure the cassandra and superuser roles are separate (Manual)

Profile Applicability:

- Level 2 - Yugabyte

Description:

The default installation of YugabyteDB 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.

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.

Audit:

To verify the configuration, run the following query:

```
cassandra@ycqlsh> SELECT role FROM system_auth.roles WHERE is_superuser = 
True;


 role
-----------
 cassandra


(1 rows)
```

If `cassandara` 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.

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

```
ALTER ROLE cassandra WITH SUPERUSER=FALSE;
```

To enhance security further, disable login ability for this role by issuing the following command:

```
ALTER ROLE cassandra WITH LOGIN=FALSE;
```

Default Value:

N/A

References:

1. https://docs.yugabyte.com/preview/secure/enable-authentication/ycql/#create-users

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	4.7 Manage Default Accounts on Enterprise Assets and Software Manage default accounts on enterprise assets and software, such as root, administrator, and other pre-configured vendor accounts. Example implementations can include: disabling default accounts or making them unusable.	●	●	●
v8	5.4 Restrict Administrator Privileges to Dedicated Administrator Accounts Restrict administrator privileges to dedicated administrator accounts on enterprise assets. Conduct general computing activities, such as internet browsing, email, and productivity suite use, from the user's primary, non-privileged account.	●	●	●
v7	4.2 Change Default Passwords Before deploying any new asset, change all default passwords to have values consistent with administrative level accounts.	●	●	●

5.4 Ensure there are no unnecessary roles or excessive privileges (Manual)

Profile Applicability:

- Level 2 - Yugabyte

Description:

Verify each role is required 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:

```
cassandra@ycqlsh> SELECT * FROM system_auth.roles;


 role      | can_login | is_superuser | member_of | salted_hash
-----------+-----------+--------------+-----------+--------------------------
----------------------------------------------------
 cassandra |      True |          True |        [] |
$2a$12$vRLGwscF9sTH4MBbwhw22.VAVHtfTo/n327eNfB7VBcZVomNOIcQu\x00\x7f\x00\x00


(1 rows)
```

Retrieve all permissions for all roles

```
cassandra@ycqlsh> select * from system_auth.role_permissions;


 role | resource | permissions
------+----------+-------------


(0 rows)
```

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.
Drop unnecessary roles by issuing the following command:

```
cassandra@ycqlsh> DROP ROLE IF EXISTS developer
```

Alternatively, you can revoke permissions. For example, you can revoke the `engineering` role from the user `john` as follows:

```
cassandra@ycqlsh> REVOKE engineering FROM john;
```

Default Value:

N/A

References:

1. https://docs.yugabyte.com/preview/secure/authorization/create-roles-ycql/#list-roles
2. https://docs.yugabyte.com/preview/api/ycql/ddl_drop_role/
3. https://docs.yugabyte.com/preview/api/ycql/ddl_revoke_role/
4. https://docs.yugabyte.com/preview/api/ycql/ddl_revoke_permission/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.3 Disable Dormant Accounts Delete or disable any dormant accounts after a period of 45 days of inactivity, where supported.	●	●	●
v8	5.4 Restrict Administrator Privileges to Dedicated Administrator Accounts Restrict administrator privileges to dedicated administrator accounts on enterprise assets. Conduct general computing activities, such as internet browsing, email, and productivity suite use, from the user's primary, non-privileged account.	●	●	●
v7	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.	●	●	●
v7	16.8 Disable Any Unassociated Accounts Disable any account that cannot be associated with a business process or business owner.	●	●	●

5.5 Ensure that YCQL only listens for network connections on authorized interfaces (Manual)

Profile Applicability:

- Level 2 - Yugabyte

Description:

The `cql_proxy_bind_address` GFLAG setting specifies the IP address and port for YCQL interface. `cql_proxy_bind_address` should be set according to your organization's approved IP address and ports policy.

Rationale:

Setting the address or interface to bind to will tell other YCQL applications (e.g. 'ycqlsh' tool) to which address or interface to connect.

Audit:

Check the value of `cql_proxy_bind_address` in the `ts.config`. If `cql_proxy_bind_address` is set to a non-authorized IP address or port, this is a finding.

```
cat ts.config | grep cql_proxy_bind_address
```

Remediation:

Use the `yb-tserver` binary and its flags to configure the YB-TServer server. Add the following GFLAG to YB-TServer:

```
./bin/yb-tserver --cql_proxy_bind_address <ip address>:<port>
```

Default Value:

`0.0.0.0:9042 (127.0.0.1:9042)`

References:

1. https://docs.yugabyte.com/preview/reference/configuration/default-ports/
2. https://docs.yugabyte.com/preview/reference/configuration/yb-tserver/#cql-proxy-bind-address

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	4.8 Uninstall or Disable Unnecessary Services on Enterprise Assets and Software Uninstall or disable unnecessary services on enterprise assets and software, such as an unused file sharing service, web application module, or service function.		●	●
v7	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.		●	●

6 Connection and Login

The restrictions on client/user connections to the YugabyteDB database blocks unauthorized access to data and services by setting access rules. These security measures help to ensure that successful logins cannot be easily made through brute-force password attacks, replaying the password hash, or intuited by clever social engineering exploits.

Settings are generally recommended to be applied to all defined profiles. The following presents standalone examples of logins for particular use cases. The authentication rules are read from the YugabyteDB host-based authentication file, `pg_hba.conf`, from top to bottom. The first rule conforming to the condition of the request executes the METHOD and stops further processing of the file. Incorrectly applied rules, as defined by a single line instruction, can substantially alter the intended behavior resulting in either allowing or denying login attempts.

It is strongly recommended that authentication configurations be constructed incrementally with rigid testing for each newly applied rule. Because of the large number of different variations, this benchmark limits itself to a small number of authentication methods that can be successfully applied under most circumstances. Further analysis, using the other authentication methods available in YugabyteDB, is encouraged.

6.1 Ensure login via "local" UNIX Domain Socket is configured correctly (Manual)

Profile Applicability:

- Level 1 - Yugabyte

Description:

A remote host login, via SSH, is arguably the most secure means of remotely accessing and administering the YugabyteDB server. Once connected to the YugabyteDB server, using the `ysqlsh` client, via UNIX DOMAIN SOCKETS, while using the `peer` authentication method is the most secure mechanism available for local database connections. Provided a database user account of the same name of the UNIX account has already been defined in the database, even ordinary user accounts can access the cluster in a similarly highly secure manner.

Rationale:

Audit:

Newly created data clusters are empty of data and have two user accounts, the superusers `yugabyte` and `postgres`. By default, the data cluster superuser is named after the UNIX account `yugabyte`. Login authentication is tested via UNIX DOMAIN SOCKETS by the UNIX user account `yugabyte`, the default account, and `set_user` has not yet been configured:

```
# ./bin/yugabyte
yugabyte=#
```

Login attempts by another UNIX user account as the superuser should be denied:

```
# su - user1 # whoami user1
# ./bin/ysqlsh -U yugabyte -d yugabyte
ysqlsh: FATAL: Peer authentication failed for user "yugabyte"
# exit
```

This test demonstrates that not only is logging in as the superuser blocked, but so is logging in as another user:

```
# su - user2 # whoami user2
# ./bin/ysqlsh -U yugabyte -d yugabyte
ysqlsh: FATAL: Peer authentication failed for user "yugabyte"
# ./bin/ysqlsh -U user1 -d yugabyte
ysqlsh: FATAL: Peer authentication failed for user "user1"
# ./bin/ysqlsh -U user2 -d yugabyte
yugabyte=#
```

Remediation:

Creation of a database account that matches the local account allows PEER authentication:

```
# ./bin/ysqlsh -c "CREATE ROLE user1 WITH LOGIN;" CREATE ROLE
```

Execute the following as the UNIX user account, the default authentication rules should now permit the login:

```
# su - user1
# whoami user1
# ./bin/ysqlsh -d yugabyte
yugabyte=#
```

As per the host-based authentication rules in $PGDATA/pg_hba.conf, all login attempts via UNIX DOMAIN SOCKETS are processed on the line beginning with local.
Once edited, the server process must reload the authentication file before it can take effect. Improperly configured rules cannot update i.e. the old rules remain in place. The YugabyteDB logs will report the outcome of the SIGHUP:

```
yugabyte=# select pg_reload_conf();
 pg_reload_conf
----------------
 t
(1 row)
```

Use the `yb-tserver` binary and its flags to configure the YB-TServer server. Add the following GFLAG to YB-TServer:

```
./bin/yb-tserver --ysql_hba_conf_csv
```

Default Value:

N/A

References:

1. https://www.postgresql.org/docs/current/static/client-authentication.html
2. https://www.postgresql.org/docs/current/static/auth-pg-hba-conf.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	6.5 Require MFA for Administrative Access Require MFA for all administrative access accounts, where supported, on all enterprise assets, whether managed on-site or through a third-party provider.	●	●	●
v7	4.5 Use Multifactor Authentication For All Administrative Access Use multi-factor authentication and encrypted channels for all administrative account access.		●	●

6.2 Ensure login via "host" TCP/IP Socket is configured correctly (Manual)

Profile Applicability:

- Level 1 - Yugabyte

Description:

A large number of authentication METHODs are available for hosts connecting using TCP/IP sockets, including:

- trust
- reject
- md5
- scram-sha-256
- password
- gss
- sspi
- ident
- pam
- ldap
- radius
- cert

METHODs `trust`, `password`, and `ident` are **not** to be used for remote logins.

METHOD `md5` is the most popular and can be used in both encrypted and unencrypted sessions, however, it is vulnerable to packet replay attacks. **It is recommended that scram- sha-256 be used instead of md5.**

Use of the `gss`, `sspi`, `pam`, `ldap`, `radius`, and `cert` METHODs are dependent upon the availability of external authenticating processes/services and thus are not covered in this benchmark.

Rationale:

Audit:

Newly created data clusters are empty of data and have two user accounts, the superusers `yugabyte` and `postgres`. By default, the data cluster superuser is named after the UNIX account `yugabyte`. Login authentication can be tested via TCP/IP SOCKETS by any UNIX user account from the local host. A password must be assigned to each login ROLE:

```
yugabyte=# ALTER ROLE yugabyte WITH PASSWORD 'secret_password'; ALTER ROLE
```

Test an unencrypted session:

```
# ./bin/ysqlsh 'host=localhost user=yugabyte sslmode=disable' Password:
```

Test an encrypted session:

```
# ./bin/ysqlsh 'host=localhost user=yugabyte sslmode=require' Password:
```

Remote logins repeat the previous invocations but, of course, from the remote host:
Test unencrypted session:
Test encrypted sessions:

```
# ./bin/ysqlsh 'host=server-name-or-IP user=yugabyte sslmode=disable'
Password:
```

Test encrypted sessions:

```
# ./bin/ysqlsh 'host=server-name-or-IP user=yugabyte sslmode=require'
Password:
```

Remediation:

Confirm a login attempt has been made by looking for a logged error message detailing the nature of the authenticating failure. In the case of failed login attempts, whether encrypted or unencrypted, check the following:

- The server should be sitting on a port exposed to the remote connecting host i.e. NOT ip address 127.0.0.1

```
listen_addresses = '*'
```

- An authenticating rule must exist in the file `ysql_hba.conf`

This example permits only encrypted sessions for the `yugabyte` role and denies all unencrypted sessions for the `yugabyte` role:

```
# TYPE  DATABASE
USER
ADDRESS
METHOD
hostssl all
yugabyte
0.0.0.0/0
scram-sha-256
hostnossl all
yugabyte
0.0.0.0/0
reject
```

The following examples illustrate other possible configurations. The resultant "rule" of success/failure depends upon the **first matching line.**

```
# allow 'yugabyte' user only from 'localhost/loopback' connections # and only
if you know the password
# (accepts both SSL and non-SSL connections)
# TYPE  DATABASE       USER    ADDRESS METHOD
host    all     yugabyte       127.0.0.1/32   scram-sha- 256

# allow users to connect remotely only to the database named after them, #
with the correct user password:
# (accepts both SSL and non-SSL connections)
# TYPE  DATABASE       USER    ADDRESS METHOD
host    samerole       all     0.0.0.0/0      scram-sha- 256

# allow only those users who are a member of the 'rw' role to connect
# only to the database named after them, with the correct user password:
# (accepts both SSL and non-SSL connections)
# TYPE
DATABASE
USER
ADDRESS
METHOD
host
samerole
+rw
0.0.0.0/0
scram-sha-
256
```

Default Value:

The availability of the different password-based authentication methods depends on how a user's password on the server is encrypted (or hashed, more accurately). This is controlled by the configuration parameter `password_encryption` at the time the password is set.

If a password was encrypted using the `scram-sha-256` setting, then it can be used for the authentication methods `scram-sha-256`, `md5`, and `password` (but password transmission will be in plain text in the latter case).

If a password was encrypted using the `md5` setting, then it can be used only for the `md5` and `password` authentication method specifications (again, with the password transmitted in plain text in the latter case).

To check the currently stored password hashes, see the system catalog `pg_authid`.

To upgrade an existing installation from `md5` to `scram-sha-256`, after having ensured that all client libraries in use are new enough to support SCRAM, set `password_encryption = 'scram-sha-256'` in `ysql_pg.conf`, reload the `postmaster`, make all users set new passwords, and change the authentication method specifications in `ysql_hba.conf` to `scram-sha-256`.

References:

1. https://docs.yugabyte.com/preview/secure/authentication/password-authentication/#scram-sha-256

Additional Information:

1. Use TYPE `hostssl` when administrating the database cluster as a superuser.
2. Use TYPE `hostnossl` for performance purposes and when DML operations are deemed safe without SSL connections.
3. No examples have been given for ADDRESS, i.e., CIDR, hostname, domain names, etc.
4. Only three (3) types of METHOD have been documented; there are many more.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.10 Encrypt Sensitive Data in Transit Encrypt sensitive data in transit. Example implementations can include: Transport Layer Security (TLS) and Open Secure Shell (OpenSSH).		●	●
v7	14.4 Encrypt All Sensitive Information in Transit Encrypt all sensitive information in transit.		●	●

7 YugabyteDB Settings

As YugabyteDB evolves with each new iteration, configuration parameters are constantly being added, deprecated, or removed. These configuration parameters define not only server function but how well it performs.

Many routine activities, combined with a specific set of configuration parameter values, can sometimes result in degraded performance and, under a specific set of conditions, even comprise the security of the RDBMS. The fact of the matter is that any parameter has the potential to affect the accessibility and performance of a running server.

Rather than describing all the possible combinations of events, this benchmark describes how a parameter can be compromised. Examples reflect the most common, and easiest to understand, exploits. Although by no means exhaustive, it is hoped that you will be able to understand the attack vectors in the context of your environment.

7.1 Understanding attack vectors and runtime GFLAGs (Manual)

Profile Applicability:

- Level 1 - Yugabyte

Description:

Understanding the vulnerability of YugabyteDB runtime parameters by the particular delivery method, or attack vector.

Rationale:

There are as many ways of compromising a server as there are runtime parameters. A combination of any one or more of them executed at the right time under the right conditions has the potential to compromise the RDBMS. Mitigating risk is dependent upon one's understanding of the attack vectors and includes:

1. Via user session: includes those runtime parameters that can be set by a ROLE that persists for the life of a server-client session.
2. Via attribute: includes those runtime parameters that can be set by a ROLE during a server-client session that can be assigned as an attribute for an entity such as a table, index, database, or role.
3. Via server reload: includes those runtime parameters that can be set by the superuser using runtime GFLAGs affects the entire cluster.
4. Via server restart: includes those runtime parameters that can be set and effected by restarting the server process and affects the entire cluster.

Impact:

It can be difficult to totally eliminate risk. Once changed, detecting a miscreant parameter can become problematic.

Audit:

Review all configuration settings. Configure YugabyteDB logging to record all modifications and changes to the RDBMS.

Remediation:

In the case of a changed parameter, the value is returned back to its default value. In the case of a successful exploit of an already set runtime parameter then an analysis must be carried out to determine the best approach in mitigating the risk to prevent future exploitation.

Default Value:

N/A

References:

1. https://docs.yugabyte.com/preview/reference/configuration/yb-tserver/
2. https://docs.yugabyte.com/preview/reference/configuration/yb-master/
3. https://www.postgresql.org/docs/current/static/runtime-config.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	16.7 Use Standard Hardening Configuration Templates for Application Infrastructure Use standard, industry-recommended hardening configuration templates for application infrastructure components. This includes underlying servers, databases, and web servers, and applies to cloud containers, Platform as a Service (PaaS) components, and SaaS components. Do not allow in-house developed software to weaken configuration hardening.		●	●
v7	18.11 Use Standard Hardening Configuration Templates for Databases For applications that rely on a database, use standard hardening configuration templates. All systems that are part of critical business processes should also be tested.		●	●

7.2 Ensure 'backend' runtime parameters are configured correctly (Automated)

Profile Applicability:

- Level 1 - Yugabyte

Description:

In order to serve multiple clients efficiently, the YugabyteDB server server launches a new "backend" process for each client. The runtime parameters in this benchmark section are controlled by the backend process. The server's performance, in the form of slow queries causing a denial of service, and the RDBM's auditing abilities for determining root cause analysis can be compromised via these parameters.

Rationale:

A denial of service is possible by denying the use of indexes and by slowing down client access to an unreasonable level. Unsanctioned behavior can be introduced by introducing rogue libraries which can then be called in a database session. Logging can be altered and obfuscated inhibiting root cause analysis.

Impact:

All changes made on this level will affect the overall behavior of the server. These changes can only be affected by a server restart after the parameters have been altered in the configuration files.

Audit:

Issue the following command to verify the backend runtime parameters are configured correctly:

```
yugabyte=# SELECT name, setting FROM pg_settings WHERE context IN
('backend','superuser-backend') ORDER BY 1;
         name           | setting
-----------------------+---------
 ignore_system_indexes | off
 jit_debugging_support | off
 jit_profiling_support | off
 log_connections       | off
 log_disconnections    | off
 post_auth_delay       | 0
(6 rows)
```

Note: Effecting changes to these parameters can only be made at server start. Therefore, a successful exploit may not be detected until after a server restart, e.g., during a maintenance window.

Remediation:

Once detected, the unauthorized/undesired change can be corrected by altering the configuration file and executing a server restart. In the case where the parameter has been specified on the command-line invocation of yb_ctl the restart invocation is insufficient and an explicit stop and start must instead be made.

1. Query the view pg_settings and compare with previous query outputs for any changes.
2. Review configuration files `ysql_pg.conf` and `ysql_hba.conf` and compare them with previously archived file copies for any changes.
3. Examine the process output and look for parameters that were used at server startup:

```
ps -few | grep -E '[p]ost' | grep -- '-[D]'
```

1. Examine the contents of $PGDATA/postmaster.opts

Default Value:

N/A

References:

1. https://www.postgresql.org/docs/current/static/view-pg-settings.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	16.7 Use Standard Hardening Configuration Templates for Application Infrastructure Use standard, industry-recommended hardening configuration templates for application infrastructure components. This includes underlying servers, databases, and web servers, and applies to cloud containers, Platform as a Service (PaaS) components, and SaaS components. Do not allow in-house developed software to weaken configuration hardening.		●	●
v7	18.11 Use Standard Hardening Configuration Templates for Databases For applications that rely on a database, use standard hardening configuration templates. All systems that are part of critical business processes should also be tested.		●	●

7.3 Ensure 'Postmaster' Runtime Parameters are Configured (Manual)

Profile Applicability:

- Level 1 - Yugabyte

Description:

YugabyteDB runtime parameters that are executed by the postmaster process.

Rationale:

The `postmaster` process is the supervisory process that assigns a backend process to an incoming client connection. The `postmaster` manages key runtime parameters that are either shared by all backend connections or needed by the `postmaster` process itself to run.

Impact:

All changes made on this level will affect the overall behavior of the server. These changes can be affected by adding and modifying configuration flags for your YB-Master and YB-TServer nodes. A denial of service is possible by the over-allocating of limited resources, such as RAM. Client messages can be altered in such a way as to interfere with the application logic. Logging can be altered and obfuscated inhibiting root cause analysis.

Audit:

The following parameters can only be set at server start by the owner of the YugabyteDB server process and cluster, typically the UNIX user account `yugabyte`. Therefore, all exploits require the successful compromise of either that UNIX account or the `postgres` superuser account itself.

```
yugabyte=# SELECT name, setting FROM pg_settings WHERE context = 'postmaster'
ORDER BY 1;
                name                |                          setting
-------------------------------------+---------------------------------------
------------------
 allow_system_table_mods             | off
 archive_mode                        | off
 autovacuum_freeze_max_age           | 200000000
 autovacuum_max_workers              | 3
 autovacuum_multixact_freeze_max_age | 400000000
 bonjour                             | off
 bonjour_name                        |
 cluster_name                        |
 config_file                         |
/home/user/var/data/pg_data/ysql_pg.conf
 data_directory                      | /home/user/var/data/pg_data
 data_sync_retry                     | off
 dynamic_shared_memory_type          | posix
 event_source                        | PostgreSQL
 external_pid_file                   |
 hba_file                            |
/home/user/var/data/pg_data/ysql_hba.conf
 hot_standby                         | on
 huge_pages                          | try
 ident_file                          |
/home/user/var/data/pg_data/pg_ident.conf
 jit_provider                        | llvmjit
 listen_addresses                    | 10.169.1.4
 logging_collector                   | on
 max_connections                     | 300
 max_files_per_process               | 1000
 max_locks_per_transaction           | 64
 max_logical_replication_workers     | 4
 max_pred_locks_per_transaction      | 64
 max_prepared_transactions           | 0
 max_replication_slots               | 10
 max_wal_senders                     | 10
 max_worker_processes                | 8
 old_snapshot_threshold              | -1
 pg_stat_statements.max              | 5000
 port                                | 5433
 shared_buffers                      | 16384
 shared_memory_type                  | mmap
 shared_preload_libraries            |
pg_stat_statements,yb_pg_metrics,pgaudit,pg_hint_plan
 superuser_reserved_connections      | 3
 track_activity_query_size           | 1024
 track_commit_timestamp              | off
 unix_socket_directories             | /tmp/.yb.10.169.1.4:5433
 unix_socket_group                   |
 unix_socket_permissions             | 0700
 wal_buffers                         | 512
 wal_level                           | replica
 wal_log_hints                       | off
(45 rows)
```

Remediation:

Once detected, the unauthorized/undesired change can be corrected by editing the altered configuration file and executing a server restart. In the case where the parameter has been specified on the command-line invocation of pg_ctl the restart invocation is insufficient and an explicit stop and start must instead be made. Detecting a change is possible by one of the following methods:

1. Query the view pg_settings and compare with previous query outputs for any changes
2. Review the configuration files `ysql_pg.conf` and `ysql_hba.conf` and compare with previously archived file copies for any changes
3. Examine the process output and look for parameters that were used at server startup:

```
ps -few | grep -E 'postgres' | grep -- '-[D]'
```

1. Examine the contents of $PGDATA/postmaster.opts

Default Value:

N/A

References:

1. https://www.postgresql.org/docs/current/static/view-pg-settings.html
2. https://www.postgresql.org/docs/current/static/runtime-config.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	16.7 Use Standard Hardening Configuration Templates for Application Infrastructure Use standard, industry-recommended hardening configuration templates for application infrastructure components. This includes underlying servers, databases, and web servers, and applies to cloud containers, Platform as a Service (PaaS) components, and SaaS components. Do not allow in-house developed software to weaken configuration hardening.		●	●
v7	18.11 Use Standard Hardening Configuration Templates for Databases For applications that rely on a database, use standard hardening configuration templates. All systems that are part of critical business processes should also be tested.		●	●

7.4 Ensure 'SIGHUP' Runtime Parameters are Configured (Manual)

Profile Applicability:

- Level 1 - Yugabyte

Description:

YugabyteDB runtime parameters that are executed by the SIGHUP signal.

Rationale:

In order to define server behavior and optimize server performance, the server's superuser has the privilege of setting these parameters which are found in the configuration files `ysql_pg.conf` and `ysql_hba.conf`.

Impact:

All changes made on this level will affect the overall behavior of the server. These changes can be effected by editing the YugabyteDB configuration files and by either executing a server SIGHUP from the command line or, as superuser `yugabyte`, executing the YSQL command `select pg_reload_conf()`. A denial of service is possible by the over-allocating of limited resources, such as RAM. Data can be corrupted by allowing damaged pages to load or by changing parameters to reinterpret values in an unexpected fashion, e.g. changing the time zone. Client messages can be altered in such a way as to interfere with the application logic. Logging can be altered and obfuscated inhibiting root cause analysis.

Audit:

The following parameters can be set at any time, without interrupting the server, by the owner of the postmaster server process and cluster (typically UNIX user account yugabyte).

```
yugabyte=# SELECT name, setting FROM pg_settings WHERE context = 'sighup'
ORDER BY 1;
                  name                  |
                    setting

----------------------------------------+------------------------------------
------------------------------------------
-----------------------------------------------------------------------------
------------------------------------------
-------
 archive_command                        | (disabled)
 archive_timeout                        | 0
 authentication_timeout                 | 60
 autovacuum                             | on
 autovacuum_analyze_scale_factor        | 0.1
 autovacuum_analyze_threshold           | 50
 autovacuum_naptime                     | 60
 autovacuum_vacuum_cost_delay           | 20
 autovacuum_vacuum_cost_limit           | -1
 autovacuum_vacuum_scale_factor         | 0.2
 autovacuum_vacuum_threshold            | 50
 autovacuum_work_mem                    | -1
 bgwriter_delay                         | 200
 bgwriter_flush_after                   | 64
 bgwriter_lru_maxpages                  | 100
 bgwriter_lru_multiplier                | 2
 checkpoint_completion_target           | 0.5
 checkpoint_flush_after                 | 32
 checkpoint_timeout                     | 300
 checkpoint_warning                     | 30
 db_user_namespace                      | off
 fsync                                  | on
 full_page_writes                       | on
 hot_standby_feedback                   | off
 krb_caseins_users                      | off
 krb_server_keyfile                     | FILE:/nfusr/alma8-gcp-
cloud/jenkins-worker-bwwdaq/jenkins/jenkins-github-yuga
byte-db-alma8-master-clang15-release-1667/yugabyte-db/build/release-clang15-
linuxbrew-full-lto-ninja/postgres/etc/krb5.
keytab
 log_autovacuum_min_duration            | -1
 log_checkpoints                        | off
 log_destination                        | stderr
 log_directory                          | /home/ddinh_yugabyte_com
user/var/data/yb-data/tserver/logs
 log_file_mode                          | 0600
 log_filename                           | postgresql-%Y-%m-%d_%H%M%S.log
 log_hostname                           | off
 log_line_prefix                        | %m [%p]
 log_rotation_age                       | 1440
 log_rotation_size                      | 10240
 log_timezone                           | UTC
 log_truncate_on_rotation               | off
 max_pred_locks_per_page                | 2
 max_pred_locks_per_relation            | -2
 max_standby_archive_delay              | 30000
 max_standby_streaming_delay            | 30000
 max_sync_workers_per_subscription      | 2
 max_wal_size                           | 1024
 min_wal_size                           | 80
 pg_stat_statements.save                | on
 pre_auth_delay                         | 0
 restart_after_crash                    | on
 ssl                                    | off
 ssl_ca_file                            |
 ssl_cert_file                          | server.crt
 ssl_ciphers                            | HIGH:MEDIUM:+3DES:!aNULL
 ssl_crl_file                           |
 ssl_dh_params_file                     |
 ssl_ecdh_curve                         | prime256v1
 ssl_key_file                           | server.key
 ssl_max_protocol_version               |
 ssl_min_protocol_version               | TLSv1
 ssl_passphrase_command                 |
 ssl_passphrase_command_supports_reload | off
 ssl_prefer_server_ciphers              | on
 stats_temp_directory                   | pg_stat_tmp
 suppress_nonpg_logs                    | off
 synchronous_standby_names              |
 syslog_facility                        | local0
 syslog_ident                           | postgres
 syslog_sequence_numbers                | on
 syslog_split_messages                  | on
 trace_recovery_messages                | log
 vacuum_defer_cleanup_age               | 0
 wal_keep_segments                      | 0
 wal_receiver_status_interval           | 10
 wal_receiver_timeout                   | 60000
 wal_retrieve_retry_interval            | 5000
 wal_sender_timeout                     | 60000
 wal_sync_method                        | fdatasync
 wal_writer_delay                       | 200
 wal_writer_flush_after                 | 128
 yb_test_block_index_phase              |
(79 rows)
```

Remediation:

Restore all values in the YugabyteDB configuration files and invoke the server to restart.

Default Value:

N/A

References:

1. https://www.postgresql.org/docs/current/static/view-pg-settings.html
2. https://www.postgresql.org/docs/current/static/runtime-config.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	16.7 Use Standard Hardening Configuration Templates for Application Infrastructure Use standard, industry-recommended hardening configuration templates for application infrastructure components. This includes underlying servers, databases, and web servers, and applies to cloud containers, Platform as a Service (PaaS) components, and SaaS components. Do not allow in-house developed software to weaken configuration hardening.		●	●
v7	18.11 Use Standard Hardening Configuration Templates for Databases For applications that rely on a database, use standard hardening configuration templates. All systems that are part of critical business processes should also be tested.		●	●

7.5 Ensure 'Superuser' Runtime Parameters are Configured (Manual)

Profile Applicability:

- Level 1 - Yugabyte

Description:

YugabyteDB runtime parameters that can only be executed by the server's superuser,`yugabyte`.

Rationale:

In order to improve and optimize server performance, the server's superuser has the privilege of setting these parameters which are found in the configuration file `ysql_pg_conf_csv`.

Impact:

All changes made on this level will affect the overall behavior of the server. These changes can only be affected by a server restart after the parameters have been altered in the configuration files. A denial of service is possible by the over-allocating of limited resources, such as RAM. Data can be corrupted by allowing damaged pages to load or by changing parameters to reinterpret values in an unexpected fashion, e.g. changing the time zone. Client messages can be altered in such a way as to interfere with the application logic. Logging can be altered and obfuscated inhibiting root cause analysis.

Audit:

The following parameters can only be set at server start by the owner of the YugabyteDB server process and cluster i.e. typically UNIX user account `yugabyte`. Therefore, all exploits require the successful compromise of either that UNIX account or the `yugabyte` superuser account itself.

```
yugabyte=# SELECT name, setting FROM pg_settings WHERE context = 'superuser'
ORDER BY 1;
                     name                     |   setting
----------------------------------------------+-------------
 commit_delay                                 | 0
 deadlock_timeout                             | 1000
 dynamic_library_path                         | $libdir
 ignore_checksum_failure                      | off
 jit_dump_bitcode                             | off
 lc_messages                                  | en_US.UTF-8
 lo_compat_privileges                         | off
 log_duration                                 | off
 log_error_verbosity                          | default
 log_executor_stats                           | off
 log_lock_waits                               | off
 log_min_duration_sample                      | -1
 log_min_duration_statement                   | -1
 log_min_error_statement                      | error
 log_min_messages                             | warning
 log_parser_stats                             | off
 log_planner_stats                            | off
 log_replication_commands                     | off
 log_statement                                | none
 log_statement_sample_rate                    | 1
 log_statement_stats                          | off
 log_temp_files                               | -1
 log_transaction_sample_rate                  | 0
 max_stack_depth                              | 2048
 pg_stat_statements.track                     | top
 pg_stat_statements.track_utility             | on
 pgaudit.log                                  | none
 pgaudit.log_catalog                          | on
 pgaudit.log_client                           | off
 pgaudit.log_level                            | log
 pgaudit.log_parameter                        | off
 pgaudit.log_relation                         | off
 pgaudit.log_statement_once                   | off
 pgaudit.role                                 |
 session_preload_libraries                    |
 session_replication_role                     | origin
 temp_file_limit                              | 1048576
 track_activities                             | on
 track_counts                                 | on
 track_functions                              | none
 track_io_timing                              | off
 update_process_title                         | on
 wal_compression                              | off
 wal_consistency_checking                     |
 yb_binary_restore                            | off
 yb_disable_wait_for_backends_catalog_version | off
 yb_make_next_ddl_statement_nonbreaking       | off
 yb_test_system_catalogs_creation             | off
 ysql_upgrade_mode                            | off
 zero_damaged_pages                           | off
(50 rows)
```

Remediation:

The exploit is made in the configuration files. These changes are effected upon server restart. Once detected, the unauthorized/undesired change can be made by editing the altered configuration file and executing a server restart. In the case where the parameter has been set on the command-line invocation of pg_ctl the restart invocation is insufficient and an explicit stop and start must instead be made.
Detecting a change is possible by one of the following methods:

1. Query the view pg_settings and compare with previous query outputs for any changes.
2. Review the configuration files `ysql_pg.conf` and `ysql_hba.conf` and compare with previously archived file copies for any changes.
3. Examine the process output and look for parameters that were used at server startup:

```
ps aux | grep -E 'post' | grep -- '-[D]'
```

1. Examine the contents of `$PGDATA/postmaster.opts`

Default Value:

N/A

References:

1. https://www.postgresql.org/docs/current/static/view-pg-settings.html
2. https://www.postgresql.org/docs/current/static/runtime-config.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	16.7 Use Standard Hardening Configuration Templates for Application Infrastructure Use standard, industry-recommended hardening configuration templates for application infrastructure components. This includes underlying servers, databases, and web servers, and applies to cloud containers, Platform as a Service (PaaS) components, and SaaS components. Do not allow in-house developed software to weaken configuration hardening.		●	●
v7	18.11 Use Standard Hardening Configuration Templates for Databases For applications that rely on a database, use standard hardening configuration templates. All systems that are part of critical business processes should also be tested.		●	●

7.6 Ensure 'User' Runtime Parameters are Configured (Manual)

Profile Applicability:

- Level 1 - Yugabyte

Description:

These YugabyteDB runtime parameters are managed at the user account (ROLE) level.

Rationale:

In order to improve performance and optimize features, a `ROLE` has the privilege of setting numerous parameters in a transaction, session, or entity attribute. Any `ROLE` can alter any of these parameters.

Impact:

A denial of service is possible by the over-allocating of limited resources, such as RAM. Data can be corrupted by changing parameters to reinterpret values in an unexpected fashion, e.g. changing the time zone. Logging can be altered and obfuscated to inhibit root cause analysis.

Audit:

The method used to analyze the state of ROLE runtime parameters and to determine if they have been compromised is to inspect all catalogs and list attributes for database entities such as ROLEs and databases:

```
yugabyte=# SELECT name, setting FROM pg_settings WHERE context = 'user' ORDER
BY 1;
                     name                     |      setting
----------------------------------------------+--------------------
 DateStyle                                    | ISO, MDY
 IntervalStyle                                | postgres
 TimeZone                                     | UTC
 application_name                             | ysqlsh
 array_nulls                                  | on
 backend_flush_after                          | 0
 backslash_quote                              | safe_encoding
 bytea_output                                 | hex
 check_function_bodies                        | on
 client_encoding                              | UTF8
 client_min_messages                          | notice
 commit_siblings                              | 5
 constraint_exclusion                         | partition
 cpu_index_tuple_cost                         | 0.005
 cpu_operator_cost                            | 0.0025
 cpu_tuple_cost                               | 0.01
 cursor_tuple_fraction                        | 0.1
 debug_pretty_print                           | on
 debug_print_parse                            | off
 debug_print_plan                             | off
 debug_print_rewritten                        | off
 default_statistics_target                    | 100
 default_tablespace                           |
 default_text_search_config                   | pg_catalog.english
 default_transaction_deferrable               | off
 default_transaction_isolation                | read committed
 default_transaction_read_only                | off
 default_with_oids                            | off
 effective_cache_size                         | 524288
 effective_io_concurrency                     | 1
 enable_bitmapscan                            | on
 enable_gathermerge                           | on
 enable_hashagg                               | on
 enable_hashjoin                              | on
 enable_indexonlyscan                         | on
 enable_indexscan                             | on
 enable_material                              | on
 enable_mergejoin                             | on
 enable_nestloop                              | on
 enable_parallel_append                       | on
 enable_parallel_hash                         | on
 enable_partition_pruning                     | on
 enable_partitionwise_aggregate               | off
 enable_partitionwise_join                    | off
 enable_seqscan                               | on
 enable_sort                                  | on
 enable_tidscan                               | on
 escape_string_warning                        | on
 exit_on_error                                | off
 extra_float_digits                           | 0
 force_global_transaction                     | off
 force_parallel_mode                          | off
 from_collapse_limit                          | 8
 geqo                                         | on
 geqo_effort                                  | 5
 geqo_generations                             | 0
 geqo_pool_size                               | 0
 geqo_seed                                    | 0
 geqo_selection_bias                          | 2
 geqo_threshold                               | 12
 gin_fuzzy_search_limit                       | 0
 gin_pending_list_limit                       | 4096
 idle_in_transaction_session_timeout          | 0
 jit                                          | off
 jit_above_cost                               | 100000
 jit_expressions                              | on
 jit_inline_above_cost                        | 500000
 jit_optimize_above_cost                      | 500000
 jit_tuple_deforming                          | on
 join_collapse_limit                          | 8
 lc_monetary                                  | en_US.UTF-8
 lc_numeric                                   | en_US.UTF-8
 lc_time                                      | en_US.UTF-8
 local_preload_libraries                      |
 lock_timeout                                 | 0
 maintenance_work_mem                         | 65536
 max_parallel_maintenance_workers             | 2
 max_parallel_workers                         | 8
 max_parallel_workers_per_gather              | 2
 min_parallel_index_scan_size                 | 64
 min_parallel_table_scan_size                 | 1024
 operator_precedence_warning                  | off
 parallel_leader_participation                | on
 parallel_setup_cost                          | 1000
 parallel_tuple_cost                          | 0.1
 password_encryption                          | md5
 pg_hint_plan.debug_print                     | off
 pg_hint_plan.enable_hint                     | on
 pg_hint_plan.enable_hint_table               | off
 pg_hint_plan.message_level                   | log
 pg_hint_plan.parse_messages                  | info
 quote_all_identifiers                        | off
 random_page_cost                             | 4
 retry_backoff_multiplier                     | 2
 retry_max_backoff                            | 1000
 retry_min_backoff                            | 100
 row_security                                 | on
 search_path                                  | "$user", public
 seq_page_cost                                | 1
 standard_conforming_strings                  | on
 statement_timeout                            | 0
 synchronize_seqscans                         | on
 synchronous_commit                           | on
 tcp_keepalives_count                         | 9
 tcp_keepalives_idle                          | 7200
 tcp_keepalives_interval                      | 75
 temp_buffers                                 | 1024
 temp_tablespaces                             |
 timezone_abbreviations                       | Default
 trace_notify                                 | off
 trace_sort                                   | off
 transaction_deferrable                       | off
 transaction_isolation                        | read committed
 transaction_read_only                        | off
 transform_null_equals                        | off
 vacuum_cleanup_index_scale_factor            | 0.1
 vacuum_cost_delay                            | 0
 vacuum_cost_limit                            | 200
 vacuum_cost_page_dirty                       | 20
 vacuum_cost_page_hit                         | 1
 vacuum_cost_page_miss                        | 10
 vacuum_freeze_min_age                        | 50000000
 vacuum_freeze_table_age                      | 150000000
 vacuum_multixact_freeze_min_age              | 5000000
 vacuum_multixact_freeze_table_age            | 150000000
 work_mem                                     | 4096
 xmlbinary                                    | base64
 xmloption                                    | content
 yb_bnl_batch_size                            | 1
 yb_bnl_enable_hashing                        | on
 yb_bypass_cond_recheck                       | on
 yb_debug_log_catcache_events                 | off
 yb_debug_log_docdb_requests                  | off
 yb_debug_log_internal_restarts               | off
 yb_debug_report_error_stacktrace             | off
 yb_default_copy_from_rows_per_transaction    | 20000
 yb_disable_transactional_writes              | off
 yb_enable_create_with_table_oid              | off
 yb_enable_docdb_tracing                      | off
 yb_enable_expression_pushdown                | on
 yb_enable_geolocation_costing                | on
 yb_enable_hash_batch_in                      | on
 yb_enable_memory_tracking                    | on
 yb_enable_optimizer_statistics               | off
 yb_enable_sequence_pushdown                  | on
 yb_enable_upsert_mode                        | off
 yb_fetch_row_limit                           | 1024
 yb_fetch_size_limit                          | 0
 yb_follower_read_staleness_ms                | 30000
 yb_format_funcs_include_yb_metadata          | off
 yb_index_state_flags_update_delay            | 0
 yb_non_ddl_txn_for_sys_tables_allowed        | off
 yb_planner_custom_plan_for_partition_pruning | on
 yb_plpgsql_disable_prefetch_in_for_query     | off
 yb_pushdown_strict_inequality                | on
 yb_read_from_followers                       | off
 yb_test_fail_next_ddl                        | off
 yb_test_planner_custom_plan_threshold        | 5
 yb_transaction_priority_lower_bound          | 0
 yb_transaction_priority_upper_bound          | 1
 yb_wait_for_backends_catalog_version_timeout | 300000
 yb_xcluster_consistency_level                | database
 ysql_max_in_flight_ops                       | 10000
 ysql_session_max_batch_size                  | 0
(164 rows)
```

Remediation:

In the matter of a user session, the login sessions must be validated that it is not executing undesired parameter changes. In the matter of attributes that have been changed in entities, they must be manually reverted to their default value(s).

Default Value:

N/A

References:

1. https://www.postgresql.org/docs/current/static/view-pg-settings.html
2. https://www.postgresql.org/docs/current/static/runtime-config.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	16.7 Use Standard Hardening Configuration Templates for Application Infrastructure Use standard, industry-recommended hardening configuration templates for application infrastructure components. This includes underlying servers, databases, and web servers, and applies to cloud containers, Platform as a Service (PaaS) components, and SaaS components. Do not allow in-house developed software to weaken configuration hardening.		●	●
v7	18.11 Use Standard Hardening Configuration Templates for Databases For applications that rely on a database, use standard hardening configuration templates. All systems that are part of critical business processes should also be tested.		●	●

7.7 Ensure TLS is enabled and configured correctly: server to server (Automated)

Profile Applicability:

- Level 1 - Yugabyte

Description:

YugabyteDB clusters can be configured to use server-to-server encryption to protect data in transit between YugabyteDB servers. When enabled, Transport Layer Security (TLS), the successor to the deprecated Secure Sockets Layer (SSL), is used to ensure data protection for YSQL and YCQL only. Note that there is no planned support for YEDIS.

Rationale:

If TLS is not enabled and configured correctly, this increases the risk of data being compromised in transit.

Impact:

A self-signed certificate can be used for testing, but a certificate signed by a certificate authority (CA) (either one of the global CAs or a local one) should be used in production so that clients can verify the server's identity. If all the database clients are local to the organization, using a local CA is recommended.

To ultimately enable and enforce TLS authentication for the server, appropriate `hostssl` records must be added to the `pg_hba.conf` file. Be sure to `reload` YugabyteDB after any changes (restart not required).

Note: The `hostssl` record matches connection attempts made using TCP/IP, but only when the connection is made with TLS encryption. The host record matches attempts made using TCP/IP, but allows both TLS and non-TLS connections. The `hostnossl` record matches attempts made using TCP/IP, but only those without TLS. *Care should be taken to enforce TLS as appropriate.*

Audit:

To determine whether TLS is enabled, simply query the parameter value while logged into the database using either the `SHOW ssl` command or `SELECT` from system catalog view `pg_settings` as illustrated below. In both cases, `ssl` is `off`; this is a fail.

```
yugabyte=# SHOW ssl;
 ssl
-----
 off
(1 row)
```

Remediation:

Before you can enable and use server-to-server encryption, you need to create and configure server certificates for each node of your YugabyteDB cluster. For information, see: https://docs.yugabyte.com/preview/secure/tls-encryption/server-certificates/
To enable server-to-server encryption using TLS, start your YB-Master and YB-TServer nodes using the following flags.

- `--use_node_to_node_encryption` Set to `true` to enable encryption between YugabyteDB nodes. Default value is `false`.
- `--allow_insecure_connections` Set to `false` to disallow any service with unencrypted communication from joining this cluster. Default value is `true`. Note that this flag requires `--use_node_to_node_encryption` to be enabled.
- `--certs_for_client_dir` Optional. Directory containing the certificates created for this node to perform encrypted communication with the other nodes. Default for YB-Masters is `<data drive>/yb-data/master/data/certs` and for YB-TServers is `<data drive>/yb-data/tserver/data/certs`

You can enable access control by starting the yb-master services with the `--use_node_to_node_encryption=true` flag as described above.
Your command should look similar to this:

```
./bin/yb-master                                \
    --fs_data_dirs=<data directories>        \
    --master_addresses=<master addresses>    \
    --certs_dir=/home/centos/tls/$NODE_IP    \
    --allow_insecure_connections=false       \
    --use_node_to_node_encryption=true
```

Next, you can enable access control by starting the `yb-tserver` services using the `--use_node_to_node_encryption=true` flag described above.
Your command should look similar to this:

```
./bin/yb-tserver                                   \
    --fs_data_dirs=<data directories>            \
    --tserver_master_addrs=<master addresses>    \
    --certs_dir /home/centos/tls/$NODE_IP        \
    --use_node_to_node_encryption=true &
```

Finally, restart YugabyteDB and confirm `ssl` using commands outlined in Audit Procedures:

```
yugabyte=# SHOW ssl;
 ssl
-----
 on
(1 row)
```

Default Value:

Note that `server.crt` and `server.key` are the default names used by YugabyteDB. These files can be named otherwise, just ensure you update the `ysql_pg_conf_csv` to use these new names. The current names can be found via YSQL:

```
yugabyte=# Select name, setting from pg_settings where name like 'ssl%file';
        name        |  setting
--------------------+------------
 ssl_ca_file        |
 ssl_cert_file      | server.crt
 ssl_crl_file       |
 ssl_dh_params_file |
 ssl_key_file       | server.key
(5 rows)
```

References:

1. https://docs.yugabyte.com/preview/secure/tls-encryption/server-to-server/
2. https://docs.yugabyte.com/preview/secure/tls-encryption/server-certificates/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.10 Encrypt Sensitive Data in Transit Encrypt sensitive data in transit. Example implementations can include: Transport Layer Security (TLS) and Open Secure Shell (OpenSSH).		●	●
v7	14.4 Encrypt All Sensitive Information in Transit Encrypt all sensitive information in transit.		●	●

7.8 Ensure TLS is enabled and configured correctly: client to server (Automated)

Profile Applicability:

- Level 1 - Yugabyte

Description:

YugabyteDB clusters can be configured to use client-to-server encryption to protect data in transit between YugabyteDB servers and clients, tools, and APIs. When enabled, Transport Layer Security (TLS), the successor to the deprecated Secure Sockets Layer (SSL), is used to ensure data protection for YSQL and YCQL only. Note that there is no planned support for YEDIS.

Before you can enable client-to-server encryption, you first must enable server-to-server encryption.

Rationale:

If TLS is not enabled and configured correctly, this increases the risk of data being compromised in transit.

Impact:

A self-signed certificate can be used for testing, but a certificate signed by a certificate authority (CA) (either one of the global CAs or a local one) should be used in production so that clients can verify the server's identity. If all the database clients are local to the organization, using a local CA is recommended.

To ultimately enable and enforce TLS authentication for the server, appropriate `hostssl` records must be added to the `ysql_hba.conf` file. Be sure to restart YB-TServer after any changes.

Note: The `hostssl` record matches connection attempts made using TCP/IP, but only when the connection is made with TLS encryption. The host record matches attempts made using TCP/IP, but allows both TLS and non-TLS connections. The `hostnossl` record matches attempts made using TCP/IP, but only those without TLS. _Care should be taken to enforce TLS as appropriate.

Audit:

To determine whether TLS is enabled, simply query the parameter value while logged into the database using either the `SHOW ssl` command or `SELECT` from system catalog view `pg_settings` as illustrated below. In both cases, `ssl` is `off`; this is a fail.

```
yugabyte=# SHOW ssl;
 ssl
-----
 off
(1 row)
```

Remediation:

To enable client-to-server encryption for YSQL and YCQL, start your YB-TServer services with the required flags described below. Your YB-Master services do not require additional configuration.

- `--use_client_to_server_encryption` Set to `true` to enable encryption between the various YugabyteDB clients and the database cluster. Default value is `false`.
- `--allow_insecure_connections` Set to `false` to disallow any client with unencrypted communication from joining this cluster. Default value is `true`. Note that this flag requires `--use_client_to_server_encryption` to be enabled.
- `--certs_for_client_dir` Optional. Defaults to the same directory as the server-to-server encryption. This directory should contain the configuration for the client to perform TLS communication with the cluster. Default value for YB-TServer is `<data drive>/yb-data/tserver/data/certs`.

To enable access control, follow these steps, start the `yb-tserver` services with the following flag (described above):

```
--use_client_to_server_encryption=true
```

This flag enables both encrypted and unencrypted clients to connect to the cluster.
To prevent clients without the appropriate encryption from connecting, you must add the following flag:

```
--allow_insecure_connections=false
```

Your command should look similar to this:

```
./bin/yb-tserver                                        \
    --fs_data_dirs=<data directories>                \
    --tserver_master_addrs=<master addresses>        \
    --certs_for_client_dir /home/centos/tls/$NODE_IP \
    --allow_insecure_connections=false               \
    --use_client_to_server_encryption=true &
```

Default Value:

Note that server.crt and server.key are the default names used by YugabyteDB. These files can be named otherwise, just ensure you update the ysql_pg_conf_csv to use these new names. The current names can be found via YSQL:

```
yugabyte=# Select name, setting from pg_settings where name like 'ssl%file';
        name        |  setting
--------------------+------------
 ssl_ca_file        |
 ssl_cert_file      | server.crt
 ssl_crl_file       |
 ssl_dh_params_file |
 ssl_key_file       | server.key
(5 rows)
```

References:

1. https://docs.yugabyte.com/preview/secure/tls-encryption/client-to-server/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.10 Encrypt Sensitive Data in Transit Encrypt sensitive data in transit. Example implementations can include: Transport Layer Security (TLS) and Open Secure Shell (OpenSSH).		●	●
v7	14.4 Encrypt All Sensitive Information in Transit Encrypt all sensitive information in transit.		●	●

7.9 Ensure the pgcrypto extension is installed and configured correctly (Manual)

Profile Applicability:

- Level 1 - Yugabyte

Description:

YugabyteDB must implement cryptographic mechanisms to prevent unauthorized disclosure or modification of organization-defined information at rest (to include, at a minimum, PII and classified information) on organization-defined information system components.

Rationale:

YugabyteDB instances handling data that requires "data at rest" protections must employ cryptographic mechanisms to prevent unauthorized disclosure and modification of the information at rest. These cryptographic mechanisms may be native to YugabyteDB or implemented via additional software or operating system/file system settings, as appropriate to the situation. Information at rest refers to the state of information when it is located on a secondary storage device (e.g. disk drive, tape drive) within an organizational information system.

The selection of a cryptographic mechanism is based on the need to protect the integrity of organizational information. The strength of the mechanism is commensurate with the security category and/or classification of the information. Organizations have the flexibility to either encrypt all information on storage devices (i.e. full disk encryption) or encrypt specific data structures (e.g. files, records, or fields). Organizations may also optionally choose to implement both to implement layered security.

The decision of whether, and what, to encrypt rests with the data owner and is also influenced by the physical measures taken to secure the equipment and media on which the information resides. Organizations may choose to employ different mechanisms to achieve confidentiality and integrity protection, as appropriate. If the confidentiality and integrity of application data are not protected, the data will be open to compromise and unauthorized modification.

The YugabyteDB pgcrypto extension provides cryptographic functions for YugabyteDB and is intended to address the confidentiality and integrity of user and system information at rest in non-mobile devices.

Impact:

When considering or undertaking any form of encryption, it is critical to understand the state of the encrypted data at all stages of the data lifecycle. The use of `pgcrypto` ensures that the data at rest in the tables (and therefore on disk) is encrypted, but for the data to be accessed by any users or applications, said users/applications will, by necessity, have access to the encrypt and decrypt keys and the data in question will be encrypted/decrypted in memory and then transferred to/from the user/application in that form.

Audit:

One possible way to encrypt data within YugabyeDB is to use the `pgcrypto` extension. To check if `pgcrypto` is installed on YugabyteDB, as a database administrator run the following commands:

```
yugabyte=# SELECT * FROM pg_available_extensions WHERE name='pgcrypto';
   name   | default_version | installed_version |         comment
----------+-----------------+-------------------+-------------------------
 pgcrypto | 1.3             |                   | cryptographic functions
(1 row)
```

If data in the database requires encryption and pgcrypto is not available, this is a fail. If disk or filesystem requires encryption, ask the system owner, DBA, and SA to demonstrate the use of disk-level encryption. If this is required and is not found, this is a fail. If controls do not exist or are not enabled, this is also a fail.

Remediation:

The pgcrypto extension is included with the YugabyteDB 'contrib' package. Although included, it needs to be created in the database.
As the database administrator, run the following:

```
yugabyte=# CREATE EXTENSION pgcrypto;
```

Verify `pgcrypto` is installed:

```
yugabyte=# SELECT * FROM pg_available_extensions WHERE name='pgcrypto';
   name   | default_version | installed_version |         comment
----------+-----------------+-------------------+-------------------------
 pgcrypto | 1.3             | 1.3               | cryptographic functions
(1 row)
```

Default Value:

`pgcrypto` is pre-bundled with standard YugabyteDB distribution but requires installation.

References:

1. https://docs.yugabyte.com/preview/explore/ysql-language-features/pg-extensions/extension-pgcrypto/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.11 Encrypt Sensitive Data at Rest Encrypt sensitive data at rest on servers, applications, and databases containing sensitive data. Storage-layer encryption, also known as server-side encryption, meets the minimum requirement of this Safeguard. Additional encryption methods may include application-layer encryption, also known as client-side encryption, where access to the data storage device(s) does not permit access to the plain-text data.		●	●
v7	14.8 Encrypt Sensitive Information at Rest Encrypt all sensitive information at rest using a tool that requires a secondary authentication mechanism not integrated into the operating system, in order to access the information.			●

8 Special Configuration Considerations

The recommendations proposed here try to address some of the less common use cases which may warrant additional configuration guidance/consideration.

8.1 Ensure base backups are configured and functional (Manual)

Profile Applicability:

- Level 1 - Yugabyte

Description:

Backup and restoration is the process of creating and storing copies of your data for protection against data loss. With a proper backup strategy, you can always restore your data to a most-recent known working state and minimize application downtime. This in turn guarantees business and application continuity.

Rationale:

Unlike traditional single-instance databases, YugabyteDB is designed for fault tolerance. By maintaining at least three copies of your data across multiple data regions or multiple clouds, it makes sure no losses occur if a single node or single data region becomes unavailable. Thus, with YugabyteDB, you would mainly use backups to:

- Recover from a user or software error, such as accidental table removal.
- Recover from a disaster scenario, like a full cluster failure or a simultaneous outage of multiple data regions. Even though such scenarios are extremely unlikely, it's still a best practice to maintain a way to recover from them.
- Maintain a remote copy of data, as required by data protection regulations.

Audit:

Verify snapshots exist by executing the `list_snapshots` command, as follows:

```
./bin/yb-admin -master_addresses <ip1:7100,ip2:7100,ip3:7100> list_snapshots
```

All the snapshots in the cluster are listed, along with their statuses.

```
Snapshot UUID                           State       Creation Time
0d4b4935-2c95-4523-95ab-9ead1e95e794    COMPLETE    2023-04-20
00:20:38.214201
```

Remediation:

Using distributed snapshots allows you to back up a database and then restore it in case of a software or operational error, with minimal recovery time objectives (RTO) and overhead.
To back up a database, create a snapshot using the `create_database_snapshot` command, as follows:

```
./bin/yb-admin -master_addresses <ip1:7100,ip2:7100,ip3:7100>
create_database_snapshot ysql.<database_name>
```

A unique ID for the snapshot is returned, as shown in the following sample output:

```
Started snapshot creation: 0d4b4935-2c95-4523-95ab-9ead1e95e794
```

You can then use this ID to check the status of the snapshot, delete it, or use it to restore the database.
The `create_database_snapshot` command exits immediately, but the snapshot may take some time to complete. Before using the snapshot, verify its status by executing the `list_snapshots` command, as follows:

```
./bin/yb-admin -master_addresses <ip1:7100,ip2:7100,ip3:7100> list_snapshots
```

All the snapshots in the cluster are listed, along with their statuses. You can find the ID of the new snapshot and make sure it has been completed, as shown in the following sample output:

```
Snapshot UUID                           State       Creation Time
0d4b4935-2c95-4523-95ab-9ead1e95e794    COMPLETE    2023-04-20
00:20:38.214201
```

Default Value:

N/A

References:

1. https://docs.yugabyte.com/preview/manage/backup-restore/snapshot-ysql/
2. https://docs.yugabyte.com/preview/manage/backup-restore/snapshot-ysql/#restore-a-snapshot
3. https://docs.yugabyte.com/preview/manage/backup-restore/snapshot-ysql/#delete-a-snapshot

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	11.2 Perform Automated Backups Perform automated backups of in-scope enterprise assets. Run backups weekly, or more frequently, based on the sensitivity of the data.	●	●	●
v8	11.5 Test Data Recovery Test backup recovery quarterly, or more frequently, for a sampling of in-scope enterprise assets.		●	●
v7	10.1 Ensure Regular Automated Back Ups Ensure that all system data is automatically backed up on regular basis.	●	●	●
v7	10.3 Test Data on Backup Media Test data integrity on backup media on a regular basis by performing a data restoration process to ensure that the backup is properly working.		●	●

8.2 Ensure YugabyteDB configuration files are outside the data cluster (Manual)

Profile Applicability:

- Level 1 - Yugabyte

Description:

YugabyteDB configuration files within the data cluster's directory tree can be changed by anyone logging into the data cluster as the superuser, i.e. `yugabyte`. As a matter of default policy, configuration files such as `ysql_pg.conf`, `ysql_hba.conf`, and `pg_ident`, are placed in the data cluster's directory, `$PGDATA`. YugabyteDB can be configured to relocate these files to locations outside the data cluster which cannot then be altered by an ordinary superuser login session.

Consideration should also be given to "include directives"; these are cluster subdirectories where one can locate files containing additional configuration parameters. Include directives are meant to add more flexibility for unique installs or large network environments while maintaining order and consistent architectural design.

Rationale:

Leaving YugabyteDB configuration files within the data cluster's directory tree increases the chances that they will be inadvertently or intentionally altered.

Audit:

Execute the following commands to verify the configuration is correct:

```
yugabyte=# select name, setting from pg_settings where name ~ '.*_file$';
        name        |                          setting
--------------------+---------------------------------------------------------
-
 config_file        | /home/user/var/data/pg_data/ysql_pg.conf
 external_pid_file  |
 hba_file           | /home/user/var/data/pg_data/ysql_hba.conf
 ident_file         | /home/user/var/data/pg_data/pg_ident.conf
 ssl_ca_file        |
 ssl_cert_file      | server.crt
 ssl_crl_file       |
 ssl_dh_params_file |
 ssl_key_file       | server.key
(9 rows)
```

Execute the following command to see any active include settings:

```
$ grep ^include $PGDATA/postgresql.{auto.,}conf
```

Inspect the file directories and permissions for all returned values. Only superusers and authorized users should have access control rights for these files. If permissions are not highly restricted, this is a fail.

Remediation:

Follow these steps to remediate the configuration file locations and permissions:

- Determine appropriate locations for relocatable configuration files based on your organization's security policies. If necessary, relocate and/or rename configuration files outside of the data cluster.
- Ensure their file permissions are restricted as much as possible, i.e. only superuser read access.
- Change the settings accordingly in the `ysql_pg.conf` configuration file via GFLAGs.
- Restart the database cluster for the changes to take effect.

Default Value:

The defaults for YugabyteDB configuration files are listed below.

```
         name          |                setting
-----------------------+----------------------------------------
 config_file           | /var/lib/pgsql/14/data/ysql_pg.conf
 external_pid_file     |
 hba_file              | /var/lib/pgsql/14/data/ysql_hba.conf
 ident_file            | /var/lib/pgsql/14/data/pg_ident.conf
 promote_trigger_file  |
 ssl_ca_file           |
 ssl_cert_file         | server.crt
 ssl_crl_file          |
 ssl_dh_params_file    |
 ssl_key_file          | server.key
(10 rows)
```

References:

1. https://www.postgresql.org/docs/current/static/runtime-config-file-locations.html
2. https://www.postgresql.org/docs/current/static/runtime-config-connection.html
3. https://www.postgresql.org/docs/current/static/config-setting.html#CONFIG-INCLUDES

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	16.7 Use Standard Hardening Configuration Templates for Application Infrastructure Use standard, industry-recommended hardening configuration templates for application infrastructure components. This includes underlying servers, databases, and web servers, and applies to cloud containers, Platform as a Service (PaaS) components, and SaaS components. Do not allow in-house developed software to weaken configuration hardening.		●	●
v7	18.11 Use Standard Hardening Configuration Templates for Databases For applications that rely on a database, use standard hardening configuration templates. All systems that are part of critical business processes should also be tested.		●	●

8.3 Ensure YugabyteDB subdirectory locations are outside the data cluster (Manual)

Profile Applicability:

- Level 1 - Yugabyte

Description:

The YugbayteDB cluster is organized to carry out specific tasks in subdirectories. For the purposes of performance, reliability, and security some of these subdirectories should be relocated outside the data cluster.

Rationale:

Some subdirectories contain information, such as logs, which can be of value to others such as developers. Other subdirectories can gain a performance benefit when placed on fast storage devices. Finally, relocating a subdirectory to a separate and distinct partition mitigates denial of service and involuntary server shutdown when excessive writes fill the data cluster's partition, e.g. `fs_data_dir`, `fs_wal_dir` and `log_dir`.

Audit:

Execute the following YSQL statement to verify the configuration is correct. Alternatively, inspect the parameter settings in the `ysql_pg.conf` configuration file.

```
yugabyte=# select name, setting from pg_settings where (name ~
'_directory$');
         name         |                        setting
----------------------+------------------------------------------------------
--
 data_directory       | /home/user/var/data/pg_data
 default_tablespace   |
 log_directory        | /home/user/var/data/yb-data/tserver/logs
 stats_temp_directory | pg_stat_tmp
 temp_tablespaces     |
(5 rows)
```

Inspect the file and directory permissions for all returned values. Only superusers and authorized users should have access control rights for these files and directories. If permissions are not highly restrictive, this is a fail.

Remediation:

Perform the following steps to remediate the subdirectory locations and permissions:

- Determine appropriate data, log, and tablespace directories and locations based on your organization's security policies. If necessary, relocate all listed directories outside the data cluster.
- Ensure file permissions are restricted as much as possible, i.e. only superuser read access.
- When directories are relocated to other partitions, ensure that they are of sufficient size to mitigate against excessive space utilization.
- Lastly, change the settings accordingly in the `ysql_pg.conf` configuration file via GFLAGs and restart the database cluster for changes to take effect.

Default Value:

The default for `data_directory` is `ConfigDir` and the default for `log_directory` is `log` (based on absolute path of `data_directory`).

References:

1. https://docs.yugabyte.com/preview/troubleshoot/nodes/check-logs/
2. https://docs.yugabyte.com/preview/reference/configuration/yb-tserver/#ysql-pg-conf-csv
3. https://docs.yugabyte.com/preview/reference/configuration/yb-tserver/#log-dir

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	16.7 Use Standard Hardening Configuration Templates for Application Infrastructure Use standard, industry-recommended hardening configuration templates for application infrastructure components. This includes underlying servers, databases, and web servers, and applies to cloud containers, Platform as a Service (PaaS) components, and SaaS components. Do not allow in-house developed software to weaken configuration hardening.		●	●
v7	18.11 Use Standard Hardening Configuration Templates for Databases For applications that rely on a database, use standard hardening configuration templates. All systems that are part of critical business processes should also be tested.		●	●

8.4 Ensure miscellaneous configuration settings are correct (Manual)

Profile Applicability:

- Level 1 - Yugabyte

Description:

This recommendation covers non-regular, special files, and dynamic libraries.

YugabyteDB permits local logins via the UNIX DOMAIN SOCKET and, for the most part, anyone with a legitimate Unix login account can make the attempt. Limiting YugabyteDB login attempts can be made by relocating the UNIX DOMAIN SOCKET to a subdirectory with restricted permissions.

The creation and implementation of user-defined dynamic libraries is an extraordinary powerful capability. In the hands of an experienced DBA/programmer, it can significantly enhance the power and flexibility of the RDBMS; but new and unexpected behavior can also be assigned to the RDBMS, resulting in a very dangerous environment in what should otherwise be trusted.

Rationale:

Audit:

Execute the following YSQL statement to verify the configuration is correct. Alternatively, inspect the parameter settings in the `ysql_pg.conf` configuration file.

```
yugabyte=# select name, setting from pg_settings where name in
('external_pid_file',
'unix_socket_directories','shared_preload_libraries','dynamic_library_path','
local_preload_libraries','session_preload_libraries');
           name            |                        setting
---------------------------+-------------------------------------------------
------
 dynamic_library_path      | $libdir
 external_pid_file         |
 local_preload_libraries   |
 session_preload_libraries |
 shared_preload_libraries  |
pg_stat_statements,yb_pg_metrics,pgaudit,pg_hint_plan
 unix_socket_directories   | /tmp/
(6 rows)
```

Inspect the file and directory permissions for all returned values. Only superusers should have access control rights for these files and directories. If permissions are not highly restricted, this is a fail.

Remediation:

Follow these steps to remediate the configuration:

- Determine permissions based on your organization's security policies.
- Relocate all files and ensure their permissions are restricted as much as possible, i.e. only superuser read access.
- Ensure all directories where these files are located have restricted permissions such that the superuser can read but not write.
- Lastly, change the settings accordingly in the `ysql_pg.conf` via GFLAGs and restart the database cluster for changes to take effect.

Default Value:

The `dynamic_library_path` default is `$libdir` and unix_socket_directories default is `/tmp`. The default for `external_pid_file` and all library parameters are initially null, or not set, upon cluster creation.

References:

1. https://www.postgresql.org/docs/current/static/runtime-config-file-locations.html
2. https://www.postgresql.org/docs/current/static/runtime-config-connection.html
3. https://www.postgresql.org/docs/current/static/runtime-config-client.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	16.7 Use Standard Hardening Configuration Templates for Application Infrastructure Use standard, industry-recommended hardening configuration templates for application infrastructure components. This includes underlying servers, databases, and web servers, and applies to cloud containers, Platform as a Service (PaaS) components, and SaaS components. Do not allow in-house developed software to weaken configuration hardening.		●	●
v7	18.11 Use Standard Hardening Configuration Templates for Databases For applications that rely on a database, use standard hardening configuration templates. All systems that are part of critical business processes should also be tested.		●	●

Appendix: Summary Table

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
1	**1 Installation and Patches**		
1.1	Ensure packages are obtained from authorized repositories (Manual)	□	□
1.2	Ensure systemd Service Files Are Enabled (Manual)	□	□
1.3	Ensure Data Cluster Initialized Successfully (Automated)	□	□
1.4	Ensure a separate user and group exist for YugabyteDB (Manual)	□	□
1.5	Ensure the latest version of Python is installed (Automated)	□	□
1.6	Ensure latest version of YugabyteDB is installed (Automated)	□	□
1.7	Ensure the YugabyteDB service is run as a non-root user (Automated)	□	□
1.8	Ensure clocks are synchronized on all node (Manual)	□	□
2	**Directory and File Permissions**		
2.1	Ensure the file permissions mask is correct (Manual)	□	□
3	**Logging Monitoring And Auditing**		
3.1	**Yugabyte Structured Query Language (YSQL) Logging**		
3.1.1	Logging Rationale (Manual)	□	□
3.1.2	Ensure the log destinations are set correctly (Automated)	□	□
3.1.3	Ensure the filename pattern for log files is set correctly (Automated)	□	□

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
3.1.4	Ensure the log file permissions are set correctly (Automated)	□	□
3.1.5	Ensure 'log_truncate_on_rotation' is enabled (Automated)	□	□
3.1.6	Ensure the maximum log file lifetime is set correctly (Automated)	□	□
3.1.7	Ensure the maximum log file size is set correctly (Automated)	□	□
3.1.8	Ensure the correct syslog facility is selected (Manual)	□	□
3.1.9	Ensure syslog messages are not suppressed (Manual)	□	□
3.1.10	Ensure syslog messages are not lost due to size (Manual)	□	□
3.1.11	Ensure the program name for YugabyteDB syslog messages is correct (Automated)	□	□
3.1.12	Ensure the correct messages are written to the server log (Automated)	□	□
3.1.13	Ensure the correct YSQL statements generating errors are recorded (Automated)	□	□
3.1.14	Ensure 'debug_print_parse' is disabled (Automated)	□	□
3.1.15	Ensure 'debug_print_rewritten' is disabled (Automated)	□	□
3.1.16	Ensure 'debug_print_plan' is disabled (Automated)	□	□
3.1.17	Ensure 'debug_pretty_print' is enabled (Automated)	□	□
3.1.18	Ensure 'log_connections' is enabled (Automated)	□	□
3.1.19	Ensure 'log_disconnections' is enabled (Automated)	□	□
3.1.20	Ensure 'log_error_verbosity' is set correctly (Automated)	□	□

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
3.1.21	Ensure 'log_hostname' is set correctly (Automated)	□	□
3.1.22	Ensure 'log_line_prefix' is set correctly (Automated)	□	□
3.1.23	Ensure 'log_statement' is set correctly (Automated)	□	□
3.1.24	Ensure 'log_timezone' is set correctly (Automated)	□	□
3.2	Ensure the YugbayteDB Audit Extension (pgAudit) is enabled (Automated)	□	□
3.3	Ensure that auditing is enabled for YCQL (Manual)	□	□
4	**User Access and Authorization**		
4.1	Ensure sudo is configured correctly (Manual)	□	□
4.2	Ensure excessive administrative privileges are revoked (Manual)	□	□
4.3	Ensure excessive function privileges are revoked (Automated)	□	□
4.4	Ensure excessive DML privileges are revoked (Manual)	□	□
4.5	Ensure Row Level Security (RLS) is configured correctly (Manual)	□	□
4.6	Make use of predefined roles (Manual)	□	□
5	**Access Control / Password Policies**		
5.1	Ensure that authentication is enabled for YCQL interface of the YugabyteDB (Automated)	□	□
5.2	Ensure that the default password changed for the cassandra role (Manual)	□	□
5.3	Ensure the cassandra and superuser roles are separate (Manual)	□	□

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
5.4	Ensure there are no unnecessary roles or excessive privileges (Manual)	□	□
5.5	Ensure that YCQL only listens for network connections on authorized interfaces (Manual)	□	□
6	**Connection and Login**		
6.1	Ensure login via "local" UNIX Domain Socket is configured correctly (Manual)	□	□
6.2	Ensure login via "host" TCP/IP Socket is configured correctly (Manual)	□	□
7	**YugabyteDB Settings**		
7.1	Understanding attack vectors and runtime GFLAGs (Manual)	□	□
7.2	Ensure 'backend' runtime parameters are configured correctly (Automated)	□	□
7.3	Ensure 'Postmaster' Runtime Parameters are Configured (Manual)	□	□
7.4	Ensure 'SIGHUP' Runtime Parameters are Configured (Manual)	□	□
7.5	Ensure 'Superuser' Runtime Parameters are Configured (Manual)	□	□
7.6	Ensure 'User' Runtime Parameters are Configured (Manual)	□	□
7.7	Ensure TLS is enabled and configured correctly: server to server (Automated)	□	□
7.8	Ensure TLS is enabled and configured correctly: client to server (Automated)	□	□
7.9	Ensure the pgcrypto extension is installed and configured correctly (Manual)	□	□

CIS Benchmark Recommendation		**Set Correctly**	
		Yes	No
8	**Special Configuration Considerations**		
8.1	Ensure base backups are configured and functional (Manual)	□	□
8.2	Ensure YugabyteDB configuration files are outside the data cluster (Manual)	□	□
8.3	Ensure YugabyteDB subdirectory locations are outside the data cluster (Manual)	□	□
8.4	Ensure miscellaneous configuration settings are correct (Manual)	□	□

Appendix: CIS Controls v7 IG 1 Mapped Recommendations

Recommendation		Set Correctly	
		Yes	No
1.3	Ensure Data Cluster Initialized Successfully	□	□
1.4	Ensure a separate user and group exist for YugabyteDB	□	□
1.7	Ensure the YugabyteDB service is run as a non-root user	□	□
2.1	Ensure the file permissions mask is correct	□	□
3.1.1	Logging Rationale	□	□
3.1.2	Ensure the log destinations are set correctly	□	□
3.1.3	Ensure the filename pattern for log files is set correctly	□	□
3.1.4	Ensure the log file permissions are set correctly	□	□
3.1.8	Ensure the correct syslog facility is selected	□	□
3.2	Ensure the YugbayteDB Audit Extension (pgAudit) is enabled	□	□
3.3	Ensure that auditing is enabled for YCQL	□	□
4.1	Ensure sudo is configured correctly	□	□
4.2	Ensure excessive administrative privileges are revoked	□	□
4.3	Ensure excessive function privileges are revoked	□	□
4.4	Ensure excessive DML privileges are revoked	□	□
4.5	Ensure Row Level Security (RLS) is configured correctly	□	□
4.6	Make use of predefined roles	□	□
5.1	Ensure that authentication is enabled for YCQL interface of the YugabyteDB	□	□
5.2	Ensure that the default password changed for the cassandra role	□	□
5.3	Ensure the cassandra and superuser roles are separate	□	□
5.4	Ensure there are no unnecessary roles or excessive privileges	□	□
8.1	Ensure base backups are configured and functional	□	□

Appendix: CIS Controls v7 IG 2 Mapped Recommendations

Recommendation		Set Correctly	
		Yes	No
1.1	Ensure packages are obtained from authorized repositories	□	□
1.3	Ensure Data Cluster Initialized Successfully	□	□
1.4	Ensure a separate user and group exist for YugabyteDB	□	□
1.5	Ensure the latest version of Python is installed	□	□
1.6	Ensure latest version of YugabyteDB is installed	□	□
1.7	Ensure the YugabyteDB service is run as a non-root user	□	□
1.8	Ensure clocks are synchronized on all node	□	□
2.1	Ensure the file permissions mask is correct	□	□
3.1.1	Logging Rationale	□	□
3.1.2	Ensure the log destinations are set correctly	□	□
3.1.3	Ensure the filename pattern for log files is set correctly	□	□
3.1.4	Ensure the log file permissions are set correctly	□	□
3.1.5	Ensure 'log_truncate_on_rotation' is enabled	□	□
3.1.6	Ensure the maximum log file lifetime is set correctly	□	□
3.1.7	Ensure the maximum log file size is set correctly	□	□
3.1.8	Ensure the correct syslog facility is selected	□	□
3.1.9	Ensure syslog messages are not suppressed	□	□
3.1.10	Ensure syslog messages are not lost due to size	□	□
3.1.11	Ensure the program name for YugabyteDB syslog messages is correct	□	□
3.1.12	Ensure the correct messages are written to the server log	□	□
3.1.13	Ensure the correct YSQL statements generating errors are recorded	□	□
3.1.14	Ensure 'debug_print_parse' is disabled	□	□
3.1.15	Ensure 'debug_print_rewritten' is disabled	□	□
3.1.16	Ensure 'debug_print_plan' is disabled	□	□
3.1.17	Ensure 'debug_pretty_print' is enabled	□	□

Recommendation		Set Correctly	
		Yes	No
3.1.18	Ensure 'log_connections' is enabled	□	□
3.1.19	Ensure 'log_disconnections' is enabled	□	□
3.1.20	Ensure 'log_error_verbosity' is set correctly	□	□
3.1.21	Ensure 'log_hostname' is set correctly	□	□
3.1.22	Ensure 'log_line_prefix' is set correctly	□	□
3.1.23	Ensure 'log_statement' is set correctly	□	□
3.1.24	Ensure 'log_timezone' is set correctly	□	□
3.2	Ensure the YugbayteDB Audit Extension (pgAudit) is enabled	□	□
3.3	Ensure that auditing is enabled for YCQL	□	□
4.1	Ensure sudo is configured correctly	□	□
4.2	Ensure excessive administrative privileges are revoked	□	□
4.3	Ensure excessive function privileges are revoked	□	□
4.4	Ensure excessive DML privileges are revoked	□	□
4.5	Ensure Row Level Security (RLS) is configured correctly	□	□
4.6	Make use of predefined roles	□	□
5.1	Ensure that authentication is enabled for YCQL interface of the YugabyteDB	□	□
5.2	Ensure that the default password changed for the cassandra role	□	□
5.3	Ensure the cassandra and superuser roles are separate	□	□
5.4	Ensure there are no unnecessary roles or excessive privileges	□	□
5.5	Ensure that YCQL only listens for network connections on authorized interfaces	□	□
6.1	Ensure login via "local" UNIX Domain Socket is configured correctly	□	□
6.2	Ensure login via "host" TCP/IP Socket is configured correctly	□	□
7.1	Understanding attack vectors and runtime GFLAGs	□	□
7.2	Ensure 'backend' runtime parameters are configured correctly	□	□
7.3	Ensure 'Postmaster' Runtime Parameters are Configured	□	□

Recommendation		Set Correctly	
		Yes	No
7.4	Ensure 'SIGHUP' Runtime Parameters are Configured	□	□
7.5	Ensure 'Superuser' Runtime Parameters are Configured	□	□
7.6	Ensure 'User' Runtime Parameters are Configured	□	□
7.7	Ensure TLS is enabled and configured correctly: server to server	□	□
7.8	Ensure TLS is enabled and configured correctly: client to server	□	□
8.1	Ensure base backups are configured and functional	□	□
8.2	Ensure YugabyteDB configuration files are outside the data cluster	□	□
8.3	Ensure YugabyteDB subdirectory locations are outside the data cluster	□	□
8.4	Ensure miscellaneous configuration settings are correct	□	□

Appendix: CIS Controls v7 IG 3 Mapped Recommendations

Recommendation		Set Correctly	
		Yes	No
1.1	Ensure packages are obtained from authorized repositories	□	□
1.3	Ensure Data Cluster Initialized Successfully	□	□
1.4	Ensure a separate user and group exist for YugabyteDB	□	□
1.5	Ensure the latest version of Python is installed	□	□
1.6	Ensure latest version of YugabyteDB is installed	□	□
1.7	Ensure the YugabyteDB service is run as a non-root user	□	□
1.8	Ensure clocks are synchronized on all node	□	□
2.1	Ensure the file permissions mask is correct	□	□
3.1.1	Logging Rationale	□	□
3.1.2	Ensure the log destinations are set correctly	□	□
3.1.3	Ensure the filename pattern for log files is set correctly	□	□
3.1.4	Ensure the log file permissions are set correctly	□	□
3.1.5	Ensure 'log_truncate_on_rotation' is enabled	□	□
3.1.6	Ensure the maximum log file lifetime is set correctly	□	□
3.1.7	Ensure the maximum log file size is set correctly	□	□
3.1.8	Ensure the correct syslog facility is selected	□	□
3.1.9	Ensure syslog messages are not suppressed	□	□
3.1.10	Ensure syslog messages are not lost due to size	□	□
3.1.11	Ensure the program name for YugabyteDB syslog messages is correct	□	□
3.1.12	Ensure the correct messages are written to the server log	□	□
3.1.13	Ensure the correct YSQL statements generating errors are recorded	□	□
3.1.14	Ensure 'debug_print_parse' is disabled	□	□
3.1.15	Ensure 'debug_print_rewritten' is disabled	□	□
3.1.16	Ensure 'debug_print_plan' is disabled	□	□
3.1.17	Ensure 'debug_pretty_print' is enabled	□	□

Recommendation		Set Correctly	
		Yes	No
3.1.18	Ensure 'log_connections' is enabled	□	□
3.1.19	Ensure 'log_disconnections' is enabled	□	□
3.1.20	Ensure 'log_error_verbosity' is set correctly	□	□
3.1.21	Ensure 'log_hostname' is set correctly	□	□
3.1.22	Ensure 'log_line_prefix' is set correctly	□	□
3.1.23	Ensure 'log_statement' is set correctly	□	□
3.1.24	Ensure 'log_timezone' is set correctly	□	□
3.2	Ensure the YugbayteDB Audit Extension (pgAudit) is enabled	□	□
3.3	Ensure that auditing is enabled for YCQL	□	□
4.1	Ensure sudo is configured correctly	□	□
4.2	Ensure excessive administrative privileges are revoked	□	□
4.3	Ensure excessive function privileges are revoked	□	□
4.4	Ensure excessive DML privileges are revoked	□	□
4.5	Ensure Row Level Security (RLS) is configured correctly	□	□
4.6	Make use of predefined roles	□	□
5.1	Ensure that authentication is enabled for YCQL interface of the YugabyteDB	□	□
5.2	Ensure that the default password changed for the cassandra role	□	□
5.3	Ensure the cassandra and superuser roles are separate	□	□
5.4	Ensure there are no unnecessary roles or excessive privileges	□	□
5.5	Ensure that YCQL only listens for network connections on authorized interfaces	□	□
6.1	Ensure login via "local" UNIX Domain Socket is configured correctly	□	□
6.2	Ensure login via "host" TCP/IP Socket is configured correctly	□	□
7.1	Understanding attack vectors and runtime GFLAGs	□	□
7.2	Ensure 'backend' runtime parameters are configured correctly	□	□
7.3	Ensure 'Postmaster' Runtime Parameters are Configured	□	□

Recommendation		Set Correctly	
		Yes	No
7.4	Ensure 'SIGHUP' Runtime Parameters are Configured	□	□
7.5	Ensure 'Superuser' Runtime Parameters are Configured	□	□
7.6	Ensure 'User' Runtime Parameters are Configured	□	□
7.7	Ensure TLS is enabled and configured correctly: server to server	□	□
7.8	Ensure TLS is enabled and configured correctly: client to server	□	□
7.9	Ensure the pgcrypto extension is installed and configured correctly	□	□
8.1	Ensure base backups are configured and functional	□	□
8.2	Ensure YugabyteDB configuration files are outside the data cluster	□	□
8.3	Ensure YugabyteDB subdirectory locations are outside the data cluster	□	□
8.4	Ensure miscellaneous configuration settings are correct	□	□

Appendix: CIS Controls v7 Unmapped Recommendations

Recommendation		Set Correctly	
		Yes	No
	No unmapped recommendations to CIS Controls v7.0	□	□

Appendix: CIS Controls v8 IG 1 Mapped Recommendations

Recommendation		Set Correctly	
		Yes	No
1.3	Ensure Data Cluster Initialized Successfully	□	□
1.4	Ensure a separate user and group exist for YugabyteDB	□	□
1.7	Ensure the YugabyteDB service is run as a non-root user	□	□
2.1	Ensure the file permissions mask is correct	□	□
3.1.1	Logging Rationale	□	□
3.1.2	Ensure the log destinations are set correctly	□	□
3.1.3	Ensure the filename pattern for log files is set correctly	□	□
3.1.4	Ensure the log file permissions are set correctly	□	□
3.1.5	Ensure 'log_truncate_on_rotation' is enabled	□	□
3.1.6	Ensure the maximum log file lifetime is set correctly	□	□
3.1.7	Ensure the maximum log file size is set correctly	□	□
3.1.8	Ensure the correct syslog facility is selected	□	□
3.2	Ensure the YugbayteDB Audit Extension (pgAudit) is enabled	□	□
3.3	Ensure that auditing is enabled for YCQL	□	□
4.1	Ensure sudo is configured correctly	□	□
4.2	Ensure excessive administrative privileges are revoked	□	□
4.3	Ensure excessive function privileges are revoked	□	□
4.4	Ensure excessive DML privileges are revoked	□	□
4.5	Ensure Row Level Security (RLS) is configured correctly	□	□
4.6	Make use of predefined roles	□	□
5.1	Ensure that authentication is enabled for YCQL interface of the YugabyteDB	□	□
5.2	Ensure that the default password changed for the cassandra role	□	□
5.3	Ensure the cassandra and superuser roles are separate	□	□
5.4	Ensure there are no unnecessary roles or excessive privileges	□	□

Recommendation		Set Correctly	
		Yes	No
6.1	Ensure login via "local" UNIX Domain Socket is configured correctly	□	□
8.1	Ensure base backups are configured and functional	□	□

Appendix: CIS Controls v8 IG 2 Mapped Recommendations

Recommendation		Set Correctly	
		Yes	No
1.1	Ensure packages are obtained from authorized repositories	□	□
1.3	Ensure Data Cluster Initialized Successfully	□	□
1.4	Ensure a separate user and group exist for YugabyteDB	□	□
1.5	Ensure the latest version of Python is installed	□	□
1.6	Ensure latest version of YugabyteDB is installed	□	□
1.7	Ensure the YugabyteDB service is run as a non-root user	□	□
1.8	Ensure clocks are synchronized on all node	□	□
2.1	Ensure the file permissions mask is correct	□	□
3.1.1	Logging Rationale	□	□
3.1.2	Ensure the log destinations are set correctly	□	□
3.1.3	Ensure the filename pattern for log files is set correctly	□	□
3.1.4	Ensure the log file permissions are set correctly	□	□
3.1.5	Ensure 'log_truncate_on_rotation' is enabled	□	□
3.1.6	Ensure the maximum log file lifetime is set correctly	□	□
3.1.7	Ensure the maximum log file size is set correctly	□	□
3.1.8	Ensure the correct syslog facility is selected	□	□
3.1.9	Ensure syslog messages are not suppressed	□	□
3.1.10	Ensure syslog messages are not lost due to size	□	□
3.1.11	Ensure the program name for YugabyteDB syslog messages is correct	□	□
3.1.12	Ensure the correct messages are written to the server log	□	□
3.1.13	Ensure the correct YSQL statements generating errors are recorded	□	□
3.1.14	Ensure 'debug_print_parse' is disabled	□	□
3.1.15	Ensure 'debug_print_rewritten' is disabled	□	□
3.1.16	Ensure 'debug_print_plan' is disabled	□	□
3.1.17	Ensure 'debug_pretty_print' is enabled	□	□

Recommendation		Set Correctly	
		Yes	No
3.1.18	Ensure 'log_connections' is enabled	□	□
3.1.19	Ensure 'log_disconnections' is enabled	□	□
3.1.20	Ensure 'log_error_verbosity' is set correctly	□	□
3.1.21	Ensure 'log_hostname' is set correctly	□	□
3.1.22	Ensure 'log_line_prefix' is set correctly	□	□
3.1.23	Ensure 'log_statement' is set correctly	□	□
3.1.24	Ensure 'log_timezone' is set correctly	□	□
3.2	Ensure the YugbayteDB Audit Extension (pgAudit) is enabled	□	□
3.3	Ensure that auditing is enabled for YCQL	□	□
4.1	Ensure sudo is configured correctly	□	□
4.2	Ensure excessive administrative privileges are revoked	□	□
4.3	Ensure excessive function privileges are revoked	□	□
4.4	Ensure excessive DML privileges are revoked	□	□
4.5	Ensure Row Level Security (RLS) is configured correctly	□	□
4.6	Make use of predefined roles	□	□
5.1	Ensure that authentication is enabled for YCQL interface of the YugabyteDB	□	□
5.2	Ensure that the default password changed for the cassandra role	□	□
5.3	Ensure the cassandra and superuser roles are separate	□	□
5.4	Ensure there are no unnecessary roles or excessive privileges	□	□
5.5	Ensure that YCQL only listens for network connections on authorized interfaces	□	□
6.1	Ensure login via "local" UNIX Domain Socket is configured correctly	□	□
6.2	Ensure login via "host" TCP/IP Socket is configured correctly	□	□
7.1	Understanding attack vectors and runtime GFLAGs	□	□
7.2	Ensure 'backend' runtime parameters are configured correctly	□	□
7.3	Ensure 'Postmaster' Runtime Parameters are Configured	□	□

Recommendation		Set Correctly	
		Yes	No
7.4	Ensure 'SIGHUP' Runtime Parameters are Configured	□	□
7.5	Ensure 'Superuser' Runtime Parameters are Configured	□	□
7.6	Ensure 'User' Runtime Parameters are Configured	□	□
7.7	Ensure TLS is enabled and configured correctly: server to server	□	□
7.8	Ensure TLS is enabled and configured correctly: client to server	□	□
7.9	Ensure the pgcrypto extension is installed and configured correctly	□	□
8.1	Ensure base backups are configured and functional	□	□
8.2	Ensure YugabyteDB configuration files are outside the data cluster	□	□
8.3	Ensure YugabyteDB subdirectory locations are outside the data cluster	□	□
8.4	Ensure miscellaneous configuration settings are correct	□	□

Appendix: CIS Controls v8 IG 3 Mapped Recommendations

Recommendation		Set Correctly	
		Yes	No
1.1	Ensure packages are obtained from authorized repositories	□	□
1.3	Ensure Data Cluster Initialized Successfully	□	□
1.4	Ensure a separate user and group exist for YugabyteDB	□	□
1.5	Ensure the latest version of Python is installed	□	□
1.6	Ensure latest version of YugabyteDB is installed	□	□
1.7	Ensure the YugabyteDB service is run as a non-root user	□	□
1.8	Ensure clocks are synchronized on all node	□	□
2.1	Ensure the file permissions mask is correct	□	□
3.1.1	Logging Rationale	□	□
3.1.2	Ensure the log destinations are set correctly	□	□
3.1.3	Ensure the filename pattern for log files is set correctly	□	□
3.1.4	Ensure the log file permissions are set correctly	□	□
3.1.5	Ensure 'log_truncate_on_rotation' is enabled	□	□
3.1.6	Ensure the maximum log file lifetime is set correctly	□	□
3.1.7	Ensure the maximum log file size is set correctly	□	□
3.1.8	Ensure the correct syslog facility is selected	□	□
3.1.9	Ensure syslog messages are not suppressed	□	□
3.1.10	Ensure syslog messages are not lost due to size	□	□
3.1.11	Ensure the program name for YugabyteDB syslog messages is correct	□	□
3.1.12	Ensure the correct messages are written to the server log	□	□
3.1.13	Ensure the correct YSQL statements generating errors are recorded	□	□
3.1.14	Ensure 'debug_print_parse' is disabled	□	□
3.1.15	Ensure 'debug_print_rewritten' is disabled	□	□
3.1.16	Ensure 'debug_print_plan' is disabled	□	□
3.1.17	Ensure 'debug_pretty_print' is enabled	□	□

Recommendation		Set Correctly	
		Yes	No
3.1.18	Ensure 'log_connections' is enabled	□	□
3.1.19	Ensure 'log_disconnections' is enabled	□	□
3.1.20	Ensure 'log_error_verbosity' is set correctly	□	□
3.1.21	Ensure 'log_hostname' is set correctly	□	□
3.1.22	Ensure 'log_line_prefix' is set correctly	□	□
3.1.23	Ensure 'log_statement' is set correctly	□	□
3.1.24	Ensure 'log_timezone' is set correctly	□	□
3.2	Ensure the YugbayteDB Audit Extension (pgAudit) is enabled	□	□
3.3	Ensure that auditing is enabled for YCQL	□	□
4.1	Ensure sudo is configured correctly	□	□
4.2	Ensure excessive administrative privileges are revoked	□	□
4.3	Ensure excessive function privileges are revoked	□	□
4.4	Ensure excessive DML privileges are revoked	□	□
4.5	Ensure Row Level Security (RLS) is configured correctly	□	□
4.6	Make use of predefined roles	□	□
5.1	Ensure that authentication is enabled for YCQL interface of the YugabyteDB	□	□
5.2	Ensure that the default password changed for the cassandra role	□	□
5.3	Ensure the cassandra and superuser roles are separate	□	□
5.4	Ensure there are no unnecessary roles or excessive privileges	□	□
5.5	Ensure that YCQL only listens for network connections on authorized interfaces	□	□
6.1	Ensure login via "local" UNIX Domain Socket is configured correctly	□	□
6.2	Ensure login via "host" TCP/IP Socket is configured correctly	□	□
7.1	Understanding attack vectors and runtime GFLAGs	□	□
7.2	Ensure 'backend' runtime parameters are configured correctly	□	□
7.3	Ensure 'Postmaster' Runtime Parameters are Configured	□	□

Recommendation		Set Correctly	
		Yes	No
7.4	Ensure 'SIGHUP' Runtime Parameters are Configured	□	□
7.5	Ensure 'Superuser' Runtime Parameters are Configured	□	□
7.6	Ensure 'User' Runtime Parameters are Configured	□	□
7.7	Ensure TLS is enabled and configured correctly: server to server	□	□
7.8	Ensure TLS is enabled and configured correctly: client to server	□	□
7.9	Ensure the pgcrypto extension is installed and configured correctly	□	□
8.1	Ensure base backups are configured and functional	□	□
8.2	Ensure YugabyteDB configuration files are outside the data cluster	□	□
8.3	Ensure YugabyteDB subdirectory locations are outside the data cluster	□	□
8.4	Ensure miscellaneous configuration settings are correct	□	□

Appendix: CIS Controls v8 Unmapped Recommendations

Recommendation		Set Correctly	
		Yes	No
	No unmapped recommendations to CIS Controls v8.0	□	□

Appendix: Change History

Date	Version	Changes for this version
12/28/2023	1.0.0	Initial release