

6 Logging and Auditing

The items in this section describe how to configure logging, log monitoring, and auditing, using tools included in most distributions.

It is recommended that `rsyslog` be used for logging (with `logwatch` providing summarization) and `auditd` be used for auditing (with `aureport` providing summarization) to automatically monitor logs for intrusion attempts and other suspicious system behavior.

In addition to the local log files created by the steps in this section, it is also recommended that sites collect copies of their system logs on a secure, centralized log server via an encrypted connection. Not only does centralized logging help sites correlate events that may be occurring on multiple systems, but having a second copy of the system log information may be critical after a system compromise where the attacker has modified the local log files on the affected system(s). If a log correlation system is deployed, configure it to process the logs described in this section.

Because it is often necessary to correlate log information from many different systems (particularly after a security incident) it is recommended that the time be synchronized among systems and devices connected to the local network. The standard Internet protocol for time synchronization is the Network Time Protocol (NTP), which is supported by most network-ready devices. Reference <<http://chrony.tuxfamily.org/>> manual page for more information on configuring chrony.

It is important that all logs described in this section be monitored on a regular basis and correlated to determine trends. A seemingly innocuous entry in one log could be more significant when compared to an entry in another log.

Note on log file permissions: There really isn't a "one size fits all" solution to the permissions on log files. Many sites utilize group permissions so that administrators who are in a defined security group, such as "wheel" do not have to elevate privileges to root in order to read log files. Also, if a third party log aggregation tool is used, it may need to have group permissions to read the log files, which is preferable to having it run setuid to root. Therefore, there are two remediation and audit steps for log file permissions. One is for systems that do not have a secured group method implemented that only permits root to read the log files (`root:root 600`). The other is for sites that do have such a setup and are designated as `root:securegrp 640` where `securegrp` is the defined security group (in some cases `wheel`).

6.1 System Logging

Logging services should be configured to prevent information leaks and to aggregate logs on a remote server so that they can be reviewed in the event of a system compromise. A centralized log server provides a single point of entry for further analysis, monitoring and filtering.

Security principals for logging

- Ensure transport layer security is implemented between the client and the log server.
- Ensure that logs are rotated as per the environment requirements.
- Ensure all locally generated logs have the appropriate permissions.
- Ensure all security logs are sent to a remote log server.
- Ensure the required events are logged.

What is covered

This section will cover the minimum best practices for the usage of either **rsyslog** - **OR** - **journald**. The recommendations are written such that each is wholly independent of each other and **only one is implemented**.

- If your organization makes use of an enterprise wide logging system completely outside of **rsyslog** or **journald**, then the following recommendations do not directly apply. However, the principals of the recommendations should be followed regardless of what solution is implemented. If the enterprise solution incorporates either of these tools, careful consideration should be given to the following recommendations to determine exactly what applies.
- Should your organization make use of both **rsyslog** and **journald**, take care how the recommendations may or may not apply to you.

What is not covered

- Enterprise logging systems not utilizing **rsyslog** or **journald**. As logging is very situational and dependent on the local environment, not everything can be covered here.
- Transport layer security should be applied to all remote logging functionality. Both **rsyslog** and **journald** supports secure transport and should be configured as such.
- The log server. There are a multitude of reasons for a centralized log server (and keeping a short period logging on the local system), but the log server is out of scope for these recommendations.

6.1.1 Configure `systemd-journald` service

`systemd-journald` is a system service that collects and stores logging data. It creates and maintains structured, indexed journals based on logging information that is received from a variety of sources:

- Kernel log messages, via `kmsg`
- Simple system log messages, via the `libc` `syslog` call
- Structured system log messages via the native Journal API
- Standard output and standard error of service units
- Audit records, originating from the kernel audit subsystem

The daemon will implicitly collect numerous metadata fields for each log messages in a secure and unfakeable way. See `systemd.journal-fields` man page for more information about the collected metadata.

The journal service stores log data either persistently below `/var/log/journal` or in a volatile way below `/run/log/journal/`. By default, log data is stored persistently if `/var/log/journal/` exists during boot, with an implicit fallback to volatile storage. Use `Storage=` in `journald.conf` to configure where log data is placed, independently of the existence of `/var/log/journal/`.

On systems where `/var/log/journal/` does not exist but where persistent logging is desired, and the default `journald.conf` is used, it is sufficient to create the directory and ensure it has the correct access modes and ownership.

Note: `systemd-journald.service` must be configured appropriately for either `journald` - **OR** - `rsyslog` to operate effectively.

6.1.1.1 Ensure journald service is enabled and active (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

Ensure that the **systemd-journald** service is enabled to allow capturing of logging events.

Rationale:

If the **systemd-journald** service is not enabled to start on boot, the system will not capture logging events.

Audit:

Run the following command to verify **systemd-journald** is enabled:

```
# systemctl is-enabled systemd-journald.service  
static
```

Note: By default the **systemd-journald** service does not have an **[Install]** section and thus cannot be enabled / disabled. It is meant to be referenced as **Requires** or **Wants** by other unit files. As such, if the status of **systemd-journald** is not **static**, investigate why

Run the following command to verify **systemd-journald** is active:

```
# systemctl is-active systemd-journald.service  
active
```

Remediation:

Run the following commands to unmask and start **systemd-journald.service**

```
# systemctl unmask systemd-journald.service  
# systemctl start systemd-journald.service
```

References:

1. NIST SP 800-53 Rev. 5: AU-2, AU-7 AU-12

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.	●	●	●
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.		●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1070, T1070.002, T1562, T1562.001	TA0005	M1029

6.1.1.2 Ensure journald log file access is configured (Manual)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

Journald will create logfiles that do not already exist on the system. This setting controls what permissions will be applied to these newly created files.

Rationale:

It is important to ensure that log files have the correct permissions to ensure that sensitive data is archived and protected.

Audit:

Run the following script to verify:

- systemd-journald logfiles are mode **0640** or more restrictive
- Directories /run/ and /var/lib/systemd/ are mode **0755** or more restrictive
- All other configured directories are mode **2755, 0750**, or more restrictive

```

#!/usr/bin/env bash

{
    a_output=() a_output2=()
    l_systemd_config_file="/etc/tmpfiles.d/systemd.conf"
    l_analyze_cmd=$(readlink -f /bin/systemd-analyze)
    f_file_chk()
    {
        l_maxperm=$(printf '%o' $(( 0777 & ~$l_perm_mask )))
        if [ $(( $l_mode & $l_perm_mask )) -le 0 ] || [[ "$l_type" =
"Directory" && "$l_mode" =~ 275(0|5) ]]; then
            a_out+=(" - $l_type \"$l_logfile\" access is:" \
                    " mode: \"$l_mode\", owned by: \"$l_user\", and group owned by:
 \"$l_group\"")
        else
            a_out2+=(" - $l_type \"$l_logfile\" access is:" \
                    " mode: \"$l_mode\", owned by: \"$l_user\", and group owned by:
 \"$l_group\" \
                    " should be mode: \"$l_maxperm\" or more restrictive")
        fi
    }
    while IFS= read -r l_file; do
        l_file=$(tr -d '# ' <<< "$l_file") a_out=() a_out2=()
        l_logfile_perms_line=$(awk '$1~/^(f|d)$/ && $2~/\/$/ && $3~/[0-
9]{3,}/ {print $2 ":" $3 ":" $4 ":" $5}' "$l_file")
        while IFS=: read -r l_logfile l_mode l_user l_group; do
            if [ -d "$l_logfile" ]; then
                l_perm_mask="0027" l_type="Directory"
                grep -Psq '^(/run|/var/lib/systemd)\b' <<< "$l_logfile" &&
                l_perm_mask="0022"
            else
                l_perm_mask="0137" l_type="File"
            fi
            grep -Psq '^(/run|/var/lib/systemd)\b' <<< "$l_logfile" &&
            l_perm_mask="0022"
            f_file_chk
            done <<< "$l_logfile_perms_line"
            [ "${#a_out[@]}" -gt "0" ] && a_output+=(" - File: \"$l_file\" sets:" \
                "${a_out[@]}")
            [ "${#a_out2[@]}" -gt "0" ] && a_out2+=(" - File: \"$l_file\" sets:" \
                "${a_out2[@]}")
            done <<($l_analyze_cmd cat-config "$l_systemd_config_file" | tac | grep -
            Pio '^h*\#\h*/[^#\n\r\h]+\.\conf\b')
            if [ "${#a_output2[@]}" -le 0 ]; then
                printf '%s\n' "" "- Audit Result:" " ** PASS **" "${a_output[@]}" ""
            else
                printf '%s\n' "" "- Audit Result:" " ** REVIEW **" \
                    " - Review file access to ensure they are set IAW site policy:" \
                    "${a_output2[@]}"
                [ "${#a_output[@]}" -gt 0 ] && printf '%s\n' "" "- Correctly set:" \
                    "${a_output[@]}" ""
            fi
        }
}

```

Review the output

Remediation:

If the default configuration is not appropriate for the site specific requirements, copy `/usr/lib/tmpfiles.d/systemd.conf` to `/etc/tmpfiles.d/systemd.conf` and modify as required. Recommended mode for logfiles is `0640` or more restrictive.

References:

1. NIST SP 800-53 Rev. 5: AC-3, AU-2, AU-12, MP-2, SI-5

Additional Information:

See `man 5 tmpfiles.d` for detailed information on the permission sets for the relevant log files. Further information with examples can be found at <https://www.freedesktop.org/software/systemd/man/tmpfiles.d.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.	●	●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1070, T1070.002, T1083, T1083.000	TA0007	M1022

6.1.1.3 Ensure journald log file rotation is configured (Manual)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

Journald includes the capability of rotating log files regularly to avoid filling up the system with logs or making the logs unmanageably large. The file **/etc/systemd/journald.conf** is the configuration file used to specify how logs generated by Journald should be rotated.

Rationale:

By keeping the log files smaller and more manageable, a system administrator can easily archive these files to another system and spend less time looking through inordinately large log files.

Audit:

Review the **systemd-journald** configuration. Verify logs are rotated according to site policy. The specific parameters for log rotation are:

Run the following script and review the output to ensure logs are rotated according to site policy:

```

#!/usr/bin/env bash

{
    a_output=() a_output2=() l_analyze_cmd=$(readlink -f /bin/systemd-analyze)
l_systemd_config_file="systemd/journald.conf"
    a_parameters=("SystemMaxUse=^.+$" "SystemKeepFree=^.+$" "RuntimeMaxUse=^.+${}" "RuntimeKeepFree=^.+${}" "MaxFileSec=^.+${}")
    f_config_file_parameter_chk()
    {
        l_used_parameter_setting=""
        while IFS= read -r l_file; do
            l_file=$(tr -d '# ' <<< "$l_file")
            l_used_parameter_setting=$(grep -PHs -- '^\\h*' "$l_parameter_name" '\b' "$l_file" | tail -n 1)
            [ -n "$l_used_parameter_setting" ] && break
        done <<< ($l_analyze_cmd cat-config "$l_systemd_config_file" | tac | grep -Pio '^\\h*#\\h*/[^#\\n\\r\\h]+\\.conf\\b')
        if [ -n "$l_used_parameter_setting" ]; then
            while IFS=: read -r l_file_name l_file_parameter; do
                while IFS== read -r l_file_parameter_name l_file_parameter_value; do
                    if grep -Pq -- "$l_parameter_value" <<< "$l_file_parameter_value"; then
                        a_output+=(" - Parameter: \"${l_file_parameter_name// /}\\""
                        " set to: \"${l_file_parameter_value// /}\\""
                        " in the file: \"$l_file_name\"")
                    fi
                done <<< "$l_file_parameter"
            done <<< "$l_used_parameter_setting"
        else
            a_output2+=(" - Parameter: \"$l_parameter_name\" is not set in an included
file"
            " *** Note: *** \" \"$l_parameter_name\" May be set in a file that's
ignored by load procedure")
        fi
    }
    for l_input_parameter in "${a_parameters[@]}"; do
        while IFS== read -r l_parameter_name l_parameter_value; do # Assess and check
parameters
            l_parameter_name="${l_parameter_name// /}";
            l_parameter_value="${l_parameter_value// /}"
            l_value_out="${l_parameter_value// /- through }";
            l_value_out="${l_value_out// / or }"
            l_value_out=$(tr -d '()' <<< "$l_value_out")
            f_config_file_parameter_chk
            done <<< "$l_input_parameter"
        done
        if [ "${#a_output2[@]}" -le 0 ]; then
            printf '%s\n' "" "- Audit Result:" " ** PASS **" "${a_output[@]}" ""
        else
            printf '%s\n' "" "- Audit Result:" " ** FAIL **" " - Reason(s) for audit
failure:" "${a_output2[@]}"
            [ "${#a_output[@]}" -gt 0 ] && printf '%s\n' "" "- Correctly set:"
"${a_output[@]}" ""
        fi
    }
}

```

Remediation:

Edit `/etc/systemd/journald.conf` or a file ending in `.conf` in the `/etc/systemd/journald.conf.d/` directory. Set the following parameters in the **[Journal]** section to ensure logs are rotated according to site policy. The settings should be carefully understood as there are specific edge cases and prioritization of parameters.

Example Configuration:

```
[Journal]
SystemMaxUse=1G
SystemKeepFree=500M
RuntimeMaxUse=200M
RuntimeKeepFree=50M
MaxFileSec=1month
```

Example script to create systemd drop-in configuration file:

```
{
    a_settings=("SystemMaxUse=1G" "SystemKeepFree=500M" "RuntimeMaxUse=200M"
"RuntimeKeepFree=50M" "MaxFileSec=1month")
    [ ! -d /etc/systemd/journald.conf.d/ ] && mkdir
/etc/systemd/journald.conf.d/
    if grep -Psq -- '^h*\[Journal\]' /etc/systemd/journald.conf.d/60-
journald.conf; then
        printf '%s\n' "" "${a_settings[@]}" >> /etc/systemd/journald.conf.d/60-
journald.conf
    else
        printf '%s\n' "" "[Journal]" "${a_settings[@]}" >>
/etc/systemd/journald.conf.d/60-journald.conf
    fi
}
```

Note:

- If these settings appear in a canonically later file, or later in the same file, the setting will be overwritten
- Logfile size and configuration to move logfiles to a remote log server should be accounted for when configuring these settings

Run the following command to update the parameters in the service:

```
# systemctl reload-or-restart systemd-journald
```

References:

1. NIST SP 800-53 Rev. 5: AU-2, AU-7, AU-12

Additional Information:

See `man 5 journald.conf` for detailed information regarding the parameters in use.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<p>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.</p>	●	●	●
v7	<p>6.2 Activate audit logging Ensure that local logging has been enabled on all systems and networking devices.</p>	●	●	●
v7	<p>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.</p>		●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1070, T1070.002	TA0040	M1022

6.1.1.4 Ensure only one logging system is in use (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

Best practices recommend that a single centralized logging system be used for log management, choose a single service either **rsyslog** - **OR** - **journald** to be used as a single centralized logging system.

Rationale:

Configuring only one logging service either **rsyslog** - **OR** - **journald** avoids redundancy, optimizes resources, simplifies configuration and management, and ensures consistency.

Impact:

Transitioning from one logging service to another can be complex and time consuming, it involves reconfiguration and may result in data loss if not managed and reconfigured correctly.

Audit:

Run the following script to ensure only one logging system is in use:

```
#!/usr/bin/env bash

{
    l_output="" l_output2="" # Check the status of rsyslog and journald
    if systemctl is-active --quiet rsyslog; then
        l_output="$l_output\n - rsyslog is in use\n- follow the
recommendations in Configure rsyslog subsection only"
    elif systemctl is-active --quiet systemd-journald; then
        l_output="$l_output\n - journald is in use\n- follow the
recommendations in Configure journald subsection only"
    else
        echo -e "unable to determine system logging"
        l_output2="$l_output2\n - unable to determine system logging\n-
Configure only ONE system logging: rsyslog OR journald"
    fi
    if [ -z "$l_output2" ]; then # Provide audit results
        echo -e "\n- Audit Result:\n  ** PASS **\n$l_output\n"
    else
        echo -e "\n- Audit Result:\n  ** FAIL **\n - Reason(s) for audit
failure:$l_output2"
    fi
}
```

Remediation:

1. Determine whether to use **journald - OR - rsyslog** depending on site needs
2. Configure **systemd-jounald.service**
3. Configure only **ONE** either **journald - OR - rsyslog** and complete the recommendations in that subsection
4. Return to this recommendation to ensure only one logging system is in use

6.1.2 Configure journald

Included in the systemd suite is a journaling service called `systemd-journald.service` for the collection and storage of logging data. It creates and maintains structured, indexed journals based on logging information that is received from a variety of sources such as:

Classic RFC3164 BSD syslog via the `/dev/log` socket `STDOUT/STDERR` of programs via `StandardOutput=journal + StandardError=journal` in service files (both of which are default settings) Kernel log messages via the `/dev/kmsg` device node Audit records via the kernel's audit subsystem Structured log messages via journald's native protocol Any changes made to the `systemd-journald` configuration will require a re-start of `systemd-journald`

Note:

- IF - `rsyslog` will be used for remote logging on the system this subsection can be skipped

6.1.2.1 Configure `systemd-journal-remote`

The `systemd-journal-remote` package includes `systemd-journal-upload`.

`systemd-journal-upload` will upload journal entries to the URL specified with `--url=`. This program reads journal entries from one or more journal files, similarly to `journalctl`.

`systemd-journal-upload` transfers the raw content of journal file and uses HTTP as a transport protocol.

`systemd-journal-upload.service` is a system service that uses `systemd-journal-upload` to upload journal entries to a server. It uses the configuration in `journal-upload.conf`.

Note:

- - IF - `rsyslog` is in use this subsection can be skipped.
- `systemd-journal-remote` package is part of the `universe` component, this may impact support and update frequency which should be considered when assessing organizational risk.

6.1.2.1.1 Ensure systemd-journal-remote is installed (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

Journald **systemd-journal-remote** supports the ability to send log events it gathers to a remote log host or to receive messages from remote hosts, thus enabling centralized log management.

Rationale:

Storing log data on a remote host protects log integrity from local attacks. If an attacker gains root access on the local system, they could tamper with or remove log data that is stored on the local system.

Note: This recommendation **only applies if journald is the chosen method for client side logging**. Do not apply this recommendation if **rsyslog** is used.

Audit:

- IF - **journald** will be used for logging on the system:

Run the following command to verify **systemd-journal-remote** is installed.

```
# dpkg-query -s systemd-journal-remote &>/dev/null && echo "systemd-journal-remote is installed"
```

Verify the output matches:

```
systemd-journal-remote is installed
```

Remediation:

Run the following command to install **systemd-journal-remote**:

```
# apt install systemd-journal-remote
```

References:

1. NIST SP 800-53 Rev. 5: AU-2, AU-7 AU-12

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.	●	●	●
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.		●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1070, T1070.002, T1562, T1562.006	TA0040	M1029

6.1.2.1.2 Ensure systemd-journal-upload authentication is configured (Manual)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

Journald **systemd-journal-upload** supports the ability to send log events it gathers to a remote log host.

Rationale:

Storing log data on a remote host protects log integrity from local attacks. If an attacker gains root access on the local system, they could tamper with or remove log data that is stored on the local system.

Note: This recommendation **only applies if journald is the chosen method for client side logging**. Do not apply this recommendation if **rsyslog** is used.

Audit:

Run the following script to verify **systemd-journal-upload** authentication is configured:

```

#!/usr/bin/env bash

{
    a_output=() a_output2=() l_analyze_cmd=$(readlink -f /bin/systemd-analyze)
l_systemd_config_file="systemd/journal-upload.conf"
    a_parameters=( "URL=.+" "ServerKeyFile=.+" "ServerCertificateFile=.+"
"TrustedCertificateFile=.+")
    f_config_file_parameter_chk()
    {
        l_used_parameter_setting=""
        while IFS= read -r l_file; do
            l_file=$(tr -d '# ' <<< "$l_file")
            l_used_parameter_setting=$(grep -PHs -- '^\\h*' "$l_parameter_name" '\b'
"$l_file" | tail -n 1)
            [ -n "$l_used_parameter_setting" ] && break
        done <<<($l_analyze_cmd cat-config "$l_systemd_config_file" | tac | grep -Pio
'^\\h*#\\h*/[^#\\n\\r\\h]+\\.conf\\b')
        if [ -n "$l_used_parameter_setting" ]; then
            while IFS=: read -r l_file_name l_file_parameter; do
                while IFS== read -r l_file_parameter_name l_file_parameter_value; do
                    if grep -Pq -- "$l_parameter_value" <<< "$l_file_parameter_value"; then
                        a_output+=(" - Parameter: \"${l_file_parameter_name// /}\"
" " set to: \"${l_file_parameter_value// /}\"
" " in the file: \"$l_file_name\"")
                    fi
                done <<< "$l_file_parameter"
            done <<< "$l_used_parameter_setting"
        else
            a_output2+=(" - Parameter: \"$l_parameter_name\" is not set in an included
file"
            " *** Note: ***" " \"$l_parameter_name\" May be set in a file that's
ignored by load procedure")
        fi
    }
    for l_input_parameter in "${a_parameters[@]}"; do
        while IFS== read -r l_parameter_name l_parameter_value; do # Assess and check
parameters
            l_parameter_name="${l_parameter_name// /}";
            l_parameter_value="${l_parameter_value// /}"
            l_value_out="${l_parameter_value// /- through }";
            l_value_out="${l_value_out// / or }"
            l_value_out=$(tr -d '()' <<< "$l_value_out")
            f_config_file_parameter_chk
            done <<< "$l_input_parameter"
        done
        if [ "${#a_output2[@]}" -le 0 ]; then
            printf '%s\n' "- Audit Result:" " ** PASS **" "${a_output[@]}"
        else
            printf '%s\n' "- Audit Result:" " ** FAIL **" " - Reason(s) for audit
failure:" "${a_output2[@]}"
            [ "${#a_output[@]}" -gt 0 ] && printf '%s\n' "- Correctly set:"
"${a_output[@]}"
        fi
    }
}

```

Review the output to ensure it matches your environments' certificate locations and the URL of the log server:

Example output:

```

- Audit Result:
  ** PASS **

- Parameter: "URL"
  set to: "192.168.50.42"
  in the file: "/etc/systemd/journal-upload.conf.d/60-journald_upload.conf"
- Parameter: "ServerKeyFile"
  set to: "/etc/ssl/private/journal-upload.pem"
  in the file: "/etc/systemd/journal-upload.conf.d/60-journald_upload.conf"
- Parameter: "ServerCertificateFile"
  set to: "/etc/ssl/certs/journal-upload.pem"
  in the file: "/etc/systemd/journal-upload.conf.d/60-journald_upload.conf"
- Parameter: "TrustedCertificateFile"
  set to: "/etc/ssl/ca/trusted.pem"
  in the file: "/etc/systemd/journal-upload.conf.d/60-journald_upload.conf"

```

Remediation:

Edit the `/etc/systemd/journal-upload.conf` file or a file in `/etc/systemd/journal-upload.conf.d` ending in `.conf` and ensure the following lines are set in the **[Upload]** section per your environment:

Example settings:

```

[Upload]
URL=192.168.50.42
ServerKeyFile=/etc/ssl/private/journal-upload.pem
ServerCertificateFile=/etc/ssl/certs/journal-upload.pem
TrustedCertificateFile=/etc/ssl/ca/trusted.pem

```

Example script to create systemd drop-in configuration file:

```

#!/usr/bin/env bash

{
    a_settings=("URL=192.168.50.42" "ServerKeyFile=/etc/ssl/private/journal-
upload.pem" \
    "ServerCertificateFile=/etc/ssl/certs/journal-upload.pem"
    "TrustedCertificateFile=/etc/ssl/ca/trusted.pem")
    [ ! -d /etc/systemd/journal-upload.conf.d/ ] && mkdir
    /etc/systemd/journal-upload.conf.d/
    if grep -Psq -- '^h*\[Upload\]' /etc/systemd/journal-upload.conf.d/60-
journald_upload.conf; then
        printf '%s\n' "" "${a_settings[@]}" >> /etc/systemd/journal-
upload.conf.d/60-journald_upload.conf
    else
        printf '%s\n' "" "[Journal]" "${a_settings[@]}" >>
    /etc/systemd/journal-upload.conf.d/60-journald_upload.conf
    fi
}

```

Run the following command to update the parameters in the service:

```
# systemctl reload-or-restart systemd-journal-upload
```

References:

1. NIST SP 800-53 Rev. 5: AU-2, AU-12

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.	●	●	●
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.		●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1070, T1070.002, T1562, T1562.006	TA0040	M1029

6.1.2.1.3 Ensure `systemd-journal-upload` is enabled and active (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

Journald `systemd-journal-upload` supports the ability to send log events it gathers to a remote log host.

Rationale:

Storing log data on a remote host protects log integrity from local attacks. If an attacker gains root access on the local system, they could tamper with or remove log data that is stored on the local system.

Note: This recommendation **only applies if `journald` is the chosen method for client side logging**. Do not apply this recommendation if `rsyslog` is used.

Audit:

Run the following command to verify `systemd-journal-upload` is enabled.

```
# systemctl is-enabled systemd-journal-upload.service  
enabled
```

Run the following command to verify `systemd-journal-upload` is active:

```
# systemctl is-active systemd-journal-upload.service  
active
```

Remediation:

Run the following commands to unmask, enable and start `systemd-journal-upload`:

```
# systemctl unmask systemd-journal-upload.service  
# systemctl --now enable systemd-journal-upload.service
```

References:

1. NIST SP 800-53 Rev. 5: AU-2, AU-12

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.	●	●	●
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.		●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1070, T1070.002, T1562, T1562.006	TA0040	M1029

6.1.2.1.4 Ensure `systemd-journal-remote` service is not in use (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

Journald `systemd-journal-remote` supports the ability to receive messages from remote hosts, thus acting as a log server. Clients should not receive data from other hosts.

Note:

- The same package, `systemd-journal-remote`, is used for both sending logs to remote hosts and receiving incoming logs.
- With regards to receiving logs, there are two services; `systemd-journal-remote.socket` and `systemd-journal-remote.service`.

Rationale:

If a client is configured to also receive data, thus turning it into a server, the client system is acting outside its operational boundary.

Note: This recommendation **only applies if `journald` is the chosen method for client side logging**. Do not apply this recommendation if `rsyslog` is used.

Audit:

Run the following command to verify `systemd-journal-remote.socket` and `systemd-journal-remote.service` are not enabled:

```
# systemctl is-enabled systemd-journal-remote.socket systemd-journal-remote.service | grep -P -- '^enabled'
```

Nothing should be returned

Run the following command to verify `systemd-journal-remote.socket` and `systemd-journal-remote.service` are not active:

```
# systemctl is-active systemd-journal-remote.socket systemd-journal-remote.service | grep -P -- '^active'
```

Nothing should be returned

Remediation:

Run the following commands to stop and mask `systemd-journal-remote.socket` and `systemd-journal-remote.service`:

```
# systemctl stop systemd-journal-remote.socket systemd-journal-remote.service  
# systemctl mask systemd-journal-remote.socket systemd-journal-remote.service
```

References:

1. NIST SP 800-53 Rev. 5: AU-2, AU-7 AU-12

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<u>4.8 Uninstall or Disable Unnecessary Services on Enterprise Assets and Software</u> Uninstall or disable unnecessary services on enterprise assets and software, such as an unused file sharing service, web application module, or service function.		●	●
v7	<u>9.2 Ensure Only Approved Ports, Protocols and Services Are Running</u> Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system.		●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1070, T1070.002, T1562, T1562.006	TA0040	M1029

6.1.2.2 Ensure journald ForwardToSyslog is disabled (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

Data from **journald** should be kept in the confines of the service and not forwarded to other services.

Rationale:

- IF - **journald** is the method for capturing logs, all logs of the system should be handled by **journald** and not forwarded to other logging mechanisms.

Note: This recommendation **only applies if journald is the chosen method for client side logging**. Do not apply this recommendation if **rsyslog** is used.

Audit:

- IF - **journald** is the method for capturing logs

Run the following script to verify **ForwardToSyslog** is set to **no**:

```

#!/usr/bin/env bash

{
    a_output=() a_output2=() l_analyze_cmd=$(readlink -f /bin/systemd-analyze)
l_systemd_config_file="systemd/journald.conf"
a_parameters=("ForwardToSyslog=no")
f_config_file_parameter_chk()
{
    l_used_parameter_setting=""
    while IFS= read -r l_file; do
        l_file=$(tr -d '# ' <<< "$l_file")
        l_used_parameter_setting=$(grep -PHs -- '^\\h*' "$l_parameter_name"'\b'$l_file" | tail -n 1)"
        [ -n "$l_used_parameter_setting" ] && break
        done <<($l_analyze_cmd cat-config "$l_systemd_config_file" | tac | grep -Pio
'^\\h*#[\\h*/[^#\\n\\r\\h]+\\.conf\\b')
        if [ -n "$l_used_parameter_setting" ]; then
            while IFS=: read -r l_file_name l_file_parameter; do
                while IFS== read -r l_file_parameter_name l_file_parameter_value; do
                    if grep -Pq -- "$l_parameter_value" <<< "$l_file_parameter_value"; then
                        a_output+=(" - Parameter: \"${l_file_parameter_name// /}\"
                            " correctly set to: \"${l_file_parameter_value// /}\"
                            " in the file: \"${l_file_name}\"")
                    else
                        a_output2+=(" - Parameter: \"${l_file_parameter_name// /}\"
                            " incorrectly set to: \"${l_file_parameter_value// /}\"
                            " in the file: \"${l_file_name}\""
                            " Should be set to: \"${l_value_out}\"")
                    fi
                    done <<< "$l_file_parameter"
                    done <<< "$l_used_parameter_setting"
                else
                    a_output2+=(" - Parameter: \"${l_parameter_name}\" is not set in an included
file" \
                    " *** Note: \"${l_parameter_name}\" May be set in a file that's ignored by
load procedure ***")
                fi
            }
            for l_input_parameter in "${a_parameters[@]}"; do
                while IFS==" read -r l_parameter_name l_parameter_value; do # Assess and check
parameters
                    l_parameter_name="${l_parameter_name// /}";
                    l_parameter_value="${l_parameter_value// /}"
                    l_value_out="${l_value_out// /- through }";
                    l_value_out=$(tr -d '()' <<< "$l_value_out")
                    f_config_file_parameter_chk
                    done <<< "$l_input_parameter"
                done
                if [ "${#a_output2[@]}" -le 0 ]; then
                    printf '%s\n' "" "- Audit Result:" " ** PASS **" "${a_output[@]}" ""
                else
                    printf '%s\n' "" "- Audit Result:" " ** FAIL **" " - Reason(s) for audit
failure:" "${a_output2[@]}"
                    [ "${#a_output[@]}" -gt 0 ] && printf '%s\n' "" "- Correctly set:""
"${a_output[@]}" ""
                fi
            }
}

```

Remediation:

- **IF** - **rsyslog** is the preferred method for capturing logs, this section and Recommendation should be skipped and the "Configure rsyslog" section followed.
- **IF** - **journald** is the preferred method for capturing logs:
Set the following parameter in the **[Journal]** section in **/etc/systemd/journald.conf** or a file in **/etc/systemd/journald.conf.d** ending in **.conf**:

```
ForwardToSyslog=no
```

Example script to create systemd drop-in configuration file:

```
#!/usr/bin/env bash

{
    a_settings=("ForwardToSyslog=no")
    [ ! -d /etc/systemd/journald.conf.d/ ] && mkdir
/etc/systemd/journald.conf.d/
    if grep -Psq -- '^h*\[Journal\]' /etc/systemd/journald.conf.d/60-
journald.conf; then
        printf '%s\n' "" "${a_settings[@]}" >> /etc/systemd/journald.conf.d/60-
journald.conf
    else
        printf '%s\n' "" "[Journal]" "${a_settings[@]}" >>
/etc/systemd/journald.conf.d/60-journald.conf
    fi
}
```

Note: If this setting appears in a canonically later file, or later in the same file, the setting will be overwritten

Run to following command to update the parameters in the service:

```
# systemctl reload-or-restart systemd-journald
```

Default Value:

ForwardToSyslog=no

References:

1. NIST SP 800-53 Rev. 5: AU-2, AU-6, AU-7, AU-12

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<p>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.</p>	●	●	●
v7	<p>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.</p>		●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1070, T1070.002, T1562, T1562.006	TA0040	M1029

6.1.2.3 Ensure journald Compress is configured (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The journald system includes the capability of compressing overly large files to avoid filling up the system with logs or making the logs unmanageably large.

Rationale:

Uncompressed large files may unexpectedly fill a filesystem leading to resource unavailability. Compressing logs prior to write can prevent sudden, unexpected filesystem impacts.

Note: This recommendation **only applies if journald is the chosen method for client side logging**. Do not apply this recommendation if rsyslog is used.

Audit:

- IF - **journald** is the method for capturing logs

Run the following script to verify **Compress** is set to **yes**:

```

#!/usr/bin/env bash

{
    a_output=() a_output2=() l_analyze_cmd=$(readlink -f /bin/systemd-analyze)
l_systemd_config_file="systemd/journald.conf"
a_parameters=("Compress=yes")
f_config_file_parameter_chk()
{
    l_used_parameter_setting=""
    while IFS= read -r l_file; do
        l_file=$(tr -d '# ' <<< "$l_file")
        l_used_parameter_setting=$(grep -PHs -- '^\\h*' "$l_parameter_name"'\b'$l_file" | tail -n 1)"
        [ -n "$l_used_parameter_setting" ] && break
        done <<($l_analyze_cmd cat-config "$l_systemd_config_file" | tac | grep -Pio
'^\\h*#[\\h*/[^#\\n\\r\\h]+\\.conf\\b')
        if [ -n "$l_used_parameter_setting" ]; then
            while IFS=: read -r l_file_name l_file_parameter; do
                while IFS== read -r l_file_parameter_name l_file_parameter_value; do
                    if grep -Pq -- "$l_parameter_value" <<< "$l_file_parameter_value"; then
                        a_output+=(" - Parameter: \"${l_file_parameter_name// /}\"
                            " correctly set to: \"${l_file_parameter_value// /}\"
                            " in the file: \"${l_file_name}\"")
                    else
                        a_output2+=(" - Parameter: \"${l_file_parameter_name// /}\"
                            " incorrectly set to: \"${l_file_parameter_value// /}\"
                            " in the file: \"${l_file_name}\""
                            " Should be set to: \"${l_value_out}\"")
                    fi
                    done <<< "$l_file_parameter"
                    done <<< "$l_used_parameter_setting"
                else
                    a_output2+=(" - Parameter: \"${l_parameter_name}\" is not set in an included
file" \
                    " *** Note: \"${l_parameter_name}\" May be set in a file that's ignored by
load procedure ***")
                fi
            }
            for l_input_parameter in "${a_parameters[@]}"; do
                while IFS==" read -r l_parameter_name l_parameter_value; do # Assess and check
parameters
                    l_parameter_name="${l_parameter_name// /}";
                    l_parameter_value="${l_parameter_value// /}"
                    l_value_out="${l_value_out// /- through }";
                    l_value_out=$(tr -d '()' <<< "$l_value_out")
                    f_config_file_parameter_chk
                    done <<< "$l_input_parameter"
                done
                if [ ${#a_output2[@]} -le 0 ]; then
                    printf '%s\n' "" "- Audit Result:" " ** PASS **" "${a_output[@]}" ""
                else
                    printf '%s\n' "" "- Audit Result:" " ** FAIL **" " - Reason(s) for audit
failure:" "${a_output2[@]}"
                    [ ${#a_output[@]} -gt 0 ] && printf '%s\n' "" "- Correctly set:""
                    "${a_output[@]}" ""
                fi
            }
}

```

Remediation:

- IF - **rsyslog** is the preferred method for capturing logs, this section and Recommendation should be skipped and the "Configure rsyslog" section followed.
- IF - **journald** is the preferred method for capturing logs:
Set the following parameter in the **[Journal]** section in **/etc/systemd/journald.conf** or a file in **/etc/systemd/journald.conf.d/** ending in **.conf**:

```
Compress=yes
```

Example script to create systemd drop-in configuration file:

```
#!/usr/bin/env bash

{
    a_settings=("Compress=yes")
    [ ! -d /etc/systemd/journald.conf.d/ ] && mkdir
/etc/systemd/journald.conf.d/
    if grep -Psq -- '^h*\[Journal\]' /etc/systemd/journald.conf.d/60-
journald.conf; then
        printf '%s\n' "" "${a_settings[@]}" >> /etc/systemd/journald.conf.d/60-
journald.conf
    else
        printf '%s\n' "" "[Journal]" "${a_settings[@]}" >>
/etc/systemd/journald.conf.d/60-journald.conf
    fi
}
```

Note: If this setting appears in a canonically later file, or later in the same file, the setting will be overwritten

Run to following command to update the parameters in the service:

```
# systemctl reload-or-restart systemd-journald
```

Default Value:

Compress=yes

References:

1. NIST SP 800-53 Rev. 5: AU-4

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	8.3 Ensure Adequate Audit Log Storage Ensure that logging destinations maintain adequate storage to comply with the enterprise's audit log management process.	●	●	●
v7	6.2 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.		●	●
v7	6.4 Ensure adequate storage for logs Ensure that all systems that store logs have adequate storage space for the logs generated.		●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562, T1562.001	TA0040	M1053

6.1.2.4 Ensure journald Storage is configured (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

Data from journald may be stored in volatile memory or persisted locally on the server. Logs in memory will be lost upon a system reboot. By persisting logs to local disk on the server they are protected from loss due to a reboot.

Rationale:

Writing log data to disk will provide the ability to forensically reconstruct events which may have impacted the operations or security of a system even after a system crash or reboot.

Note: This recommendation **only applies if journald is the chosen method for client side logging**. Do not apply this recommendation if rsyslog is used.

Audit:

- IF - **journald** is the method for capturing logs

Run the following script to verify **Storage** is set to **persistent**:

```

#!/usr/bin/env bash

{
    a_output=() a_output2=() l_analyze_cmd=$(readlink -f /bin/systemd-analyze)
l_systemd_config_file="systemd/journald.conf"
a_parameters=("Storage=persistent")
f_config_file_parameter_chk()
{
    l_used_parameter_setting=""
    while IFS= read -r l_file; do
        l_file=$(tr -d '# ' <<< "$l_file")
        l_used_parameter_setting=$(grep -PHs -- '^\\h*' "$l_parameter_name"'\b'$"$l_file" | tail -n 1)
        [ -n "$l_used_parameter_setting" ] && break
        done <<($l_analyze_cmd cat-config "$l_systemd_config_file" | tac | grep -Pio
'^\\h*#[\\h*/[^#\\n\\r\\h]+\\.conf\\b')
        if [ -n "$l_used_parameter_setting" ]; then
            while IFS=: read -r l_file_name l_file_parameter; do
                while IFS== read -r l_file_parameter_name l_file_parameter_value; do
                    if grep -Pq -- "$l_parameter_value" <<< "$l_file_parameter_value"; then
                        a_output+=(" - Parameter: \"${l_file_parameter_name// /}\\""
                        " correctly set to: \"${l_file_parameter_value// /}\\""
                        " in the file: \"${l_file_name}\"")
                    else
                        a_output2+=(" - Parameter: \"${l_file_parameter_name// /}\\""
                        " incorrectly set to: \"${l_file_parameter_value// /}\\""
                        " in the file: \"${l_file_name}\""
                        " Should be set to: \"${l_value_out}\\"")
                    fi
                    done <<< "$l_file_parameter"
                    done <<< "$l_used_parameter_setting"
                else
                    a_output2+=(" - Parameter: \"${l_parameter_name}\\" is not set in an included
file" \
                    " *** Note: \"${l_parameter_name}\\" May be set in a file that's ignored by
load procedure ***")
                fi
            }
            for l_input_parameter in "${a_parameters[@]}"; do
                while IFS==" read -r l_parameter_name l_parameter_value; do # Assess and check
parameters
                    l_parameter_name="${l_parameter_name// /}";
                    l_parameter_value="${l_parameter_value// /}"
                    l_value_out="${l_value_out// /- through }";
                    l_value_out=$(tr -d '()' <<< "$l_value_out")
                    f_config_file_parameter_chk
                    done <<< "$l_input_parameter"
                done
                if [ "${#a_output2[@]}" -le 0 ]; then
                    printf '%s\n' "" "- Audit Result:" " ** PASS **" "${a_output[@]}" ""
                else
                    printf '%s\n' "" "- Audit Result:" " ** FAIL **" " - Reason(s) for audit
failure:" "${a_output2[@]}"
                    [ "${#a_output[@]}" -gt 0 ] && printf '%s\n' "" "- Correctly set:""
"${a_output[@]}" ""
                fi
            }
}

```

Remediation:

- IF - **rsyslog** is the preferred method for capturing logs, this section and Recommendation should be skipped and the "Configure rsyslog" section followed.
- IF - **journald** is the preferred method for capturing logs:
Set the following parameter in the **[Journal]** section in **/etc/systemd/journald.conf** or a file in **/etc/systemd/journald.conf.d** ending in **.conf**:

```
Storage=persistent
```

Example script to create systemd drop-in configuration file:

```
#!/usr/bin/env bash

{
    a_settings=("Storage=persistent")
    [ ! -d /etc/systemd/journald.conf.d/ ] && mkdir
/etc/systemd/journald.conf.d/
    if grep -Psq -- '^h*\[Journal\]' /etc/systemd/journald.conf.d/60-
journald.conf; then
        printf '%s\n' "" "${a_settings[@]}" >> /etc/systemd/journald.conf.d/60-
journald.conf
    else
        printf '%s\n' "" "[Journal]" "${a_settings[@]}" >>
/etc/systemd/journald.conf.d/60-journald.conf
    fi
}
```

Note: If this setting appears in a canonically later file, or later in the same file, the setting will be overwritten

Run to following command to update the parameters in the service:

```
# systemctl reload-or-restart systemd-journald
```

Default Value:

Storage=persistent

References:

1. NIST SP 800-53 Rev. 5: AU-3, AU-12

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.	●	●	●
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.		●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1070, T1070.002, T1562, T1562.006	TA0005	M1022

6.1.3 Configure rsyslog

The **rsyslog** software package may be used instead of the default **journald** logging mechanism.

Rsyslog has evolved over several decades. For this reason it supports three different configuration formats (“languages”):

- **basic** - previously known as the **sysklogd** format, this is the format best used to express basic things, such as where the statement fits on a single line.
 - It stems back to the original syslog.conf format, in use now for several decades.
 - The most common use case is matching on facility/severity and writing matching messages to a log file.
- **advanced** - previously known as the **RainerScript** format, this format was first available in rsyslog v6 and is the current, best and most precise format for non-trivial use cases where more than one line is needed.
 - Prior to v7, there was a performance impact when using this format that encouraged use of the basic format for best results. Current versions of rsyslog do not suffer from this (historical) performance impact.
 - This new style format is specifically targeted towards more advanced use cases like forwarding to remote hosts that might be partially offline.
- **obsolete legacy** - previously known simply as the **legacy** format, this format is exactly what its name implies: it is obsolete and should not be used when writing new configurations. It was created in the early days (up to rsyslog version 5) where we expected that rsyslog would extend sysklogd just mildly. Consequently, it was primarily aimed at small additions to the original sysklogd format.
 - Practice has shown that it was notoriously hard to use for more advanced use cases, and thus we replaced it with the advanced format.
 - In essence, everything that needs to be written on a single line that starts with a dollar sign is legacy format. Users of this format are encouraged to migrate to the basic or advanced formats.

Note: This section only applies if **rsyslog** is the chosen method for client side logging. Do not apply this section if **journald** is used.

6.1.3.1 Ensure rsyslog is installed (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The **rsyslog** software is recommended in environments where **journald** does not meet operation requirements.

Rationale:

The security enhancements of **rsyslog** such as connection-oriented (i.e. TCP) transmission of logs, the option to log to database formats, and the encryption of log data en route to a central logging server) justify installing and configuring the package.

Audit:

- **IF** - **rsyslog** is being used for logging on the system:

Run the following command to verify **rsyslog** is installed:

```
# dpkg-query -s rsyslog &>/dev/null && echo "rsyslog is installed"
```

Verify the output matches:

```
rsyslog is installed
```

Remediation:

Run the following command to install **rsyslog**:

```
# apt install rsyslog
```

Default Value:

Installed

References:

1. NIST SP 800-53 Rev. 5: AU-2, AU-3, AU-12

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.	●	●	●
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.		●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1005, T1005.000, T1070, T1070.002	TA0005	M1029, M1057

6.1.3.2 Ensure rsyslog service is enabled and active (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

Once the `rsyslog` package is installed, ensure that the service is enabled.

Rationale:

If the `rsyslog` service is not enabled to start on boot, the system will not capture logging events.

Audit:

- IF - `rsyslog` is being used for logging on the system:

Run the following command to verify `rsyslog.service` is enabled:

```
# systemctl is-enabled rsyslog  
enabled
```

Run the following command to verify `rsyslog.service` is active:

```
# systemctl is-active rsyslog.service  
active
```

Remediation:

- IF - `rsyslog` is being used for logging on the system:

Run the following commands to unmask, enable, and start `rsyslog.service`:

```
# systemctl unmask rsyslog.service  
# systemctl enable rsyslog.service  
# systemctl start rsyslog.service
```

References:

1. NIST SP 800-53 Rev. 5: AU-2, AU-3, AU-12

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.	●	●	●
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.		●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1070, T1070.002, T1211, T1562, T1562.001	TA0005	M1029

6.1.3.3 Ensure journald is configured to send logs to rsyslog (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

Data from `systemd-journald` may be stored in volatile memory or persisted locally on the server. Utilities exist to accept remote export of `systemd-journald` logs, however, use of the `rsyslog` service provides a consistent means of log collection and export.

Rationale:

- IF - `rsyslog` is the preferred method for capturing logs, all logs of the system should be sent to it for further processing.

Note: This recommendation **only applies if rsyslog is the chosen method for client side logging**. Do not apply this recommendation if `systemd-journald` is used.

Audit:

- IF - `rsyslog` is the preferred method for capturing logs

Run the following script to verify that logs are forwarded to `rsyslog` by setting `ForwardToSyslog` to `yes` in the `systemd-journald` configuration:

```

#!/usr/bin/env bash

{
    a_output=() a_output2=() l_analyze_cmd=$(readlink -f /bin/systemd-analyze)
l_systemd_config_file="systemd/journald.conf"
a_parameters=("ForwardToSyslog=yes")
f_config_file_parameter_chk()
{
    l_used_parameter_setting=""
    while IFS= read -r l_file; do
        l_file=$(tr -d '# ' <<< "$l_file")
        l_used_parameter_setting=$(grep -PHs -- '^\\h*' "$l_parameter_name"'\b'$"$l_file" | tail -n 1)
        [ -n "$l_used_parameter_setting" ] && break
        done <<<($l_analyze_cmd cat-config "$l_systemd_config_file" | tac | grep -Pio
'^\\h*#[\\h*/[^#\\n\\r\\h]+\\.conf\\b')
        if [ -n "$l_used_parameter_setting" ]; then
            while IFS=: read -r l_file_name l_file_parameter; do
                while IFS== read -r l_file_parameter_name l_file_parameter_value; do
                    if grep -Pq -- "$l_parameter_value" <<< "$l_file_parameter_value"; then
                        a_output+=(" - Parameter: \"${l_file_parameter_name// /}\\""
                        " correctly set to: \"${l_file_parameter_value// /}\\""
                        " in the file: \"${l_file_name}\"")
                    else
                        a_output2+=(" - Parameter: \"${l_file_parameter_name// /}\\""
                        " incorrectly set to: \"${l_file_parameter_value// /}\\""
                        " in the file: \"${l_file_name}\""
                        " Should be set to: \"${l_value_out}\\"")
                    fi
                    done <<< "$l_file_parameter"
                    done <<< "$l_used_parameter_setting"
                else
                    a_output2+=(" - Parameter: \"${l_parameter_name}\\" is not set in an included
file" \
                    " *** Note: \"${l_parameter_name}\\" May be set in a file that's ignored by
load procedure ***")
                fi
            }
            for l_input_parameter in "${a_parameters[@]}"; do
                while IFS== read -r l_parameter_name l_parameter_value; do # Assess and check
parameters
                    l_parameter_name="${l_parameter_name// /}";
                    l_parameter_value="${l_parameter_value// /}"
                    l_value_out="${l_value_out// /- through }";
                    l_value_out=$(tr -d '()' <<< "$l_value_out")
                    f_config_file_parameter_chk
                    done <<< "$l_input_parameter"
                done
                if [ "${#a_output2[@]}" -le 0 ]; then
                    printf '%s\n' "" "- Audit Result:" " ** PASS **" "${a_output[@]}" ""
                else
                    printf '%s\n' "" "- Audit Result:" " ** FAIL **" " - Reason(s) for audit
failure:" "${a_output2[@]}"
                    [ "${#a_output[@]}" -gt 0 ] && printf '%s\n' "" "- Correctly set:""
"${a_output[@]}" ""
                fi
            }
}

```

Run the following command to verify **systemd-journald.service** and **rsyslog.service** are loaded and active:

```
# systemctl list-units --type service | grep -P -- '(journald|rsyslog)'
```

Output should be similar to:

rsyslog.service	loaded active running
System Logging Service	
systemd-journald.service	loaded active running
Journal Service	

Remediation:

- IF - **Journald** is the preferred method for capturing logs, this section and Recommendation should be skipped and the "Configure Journald" section followed.

- IF - **rsyslog** is the preferred method for capturing logs:

Set the following parameter in the **[Journal]** section in **/etc/systemd/journald.conf** or a file in **/etc/systemd/journald.conf.d/** ending in **.conf**:

```
ForwardToSyslog=yes
```

Example script to create systemd drop-in configuration file:

```
#!/usr/bin/env bash

{
    a_settings=("ForwardToSyslog=yes")
    [ ! -d /etc/systemd/journald.conf.d/ ] && mkdir /etc/systemd/journald.conf.d/
    if grep -Psq -- '^h*\[Journal\]' /etc/systemd/journald.conf.d/60-journald.conf; then
        printf '%s\n' "" "${a_settings[@]}" >> /etc/systemd/journald.conf.d/60-journald.conf
    else
        printf '%s\n' "" "[Journal]" "${a_settings[@]}" >> /etc/systemd/journald.conf.d/60-journald.conf
    fi
}
```

Note: If this setting appears in a canonically later file, or later in the same file, the setting will be overwritten

Run the following command to update the parameters in the service:

Restart **systemd-journald.service**:

```
# systemctl reload-or-restart systemd-journald.service
```

References:

1. NIST SP 800-53 Rev. 5: AC-3, AU-2, AU-4, AU-12, MP-2
2. SYSTEMD-JOURNALD.SERVICE(8)
3. JOURNALD.CONF(5)

Additional Information:

As noted in the `systemd-journald` man pages, `systemd-journald` logs may be exported to `rsyslog` either through the process mentioned here, or through a facility like `systemd-journald.service`. There are trade-offs involved in each implementation, where `ForwardToSyslog` will immediately capture all events (and forward to an external log server, if properly configured), but may not capture all boot-up activities. Mechanisms such as `systemd-journald.service`, on the other hand, will record bootup events, but may delay sending the information to `rsyslog`, leading to the potential for log manipulation prior to export. Be aware of the limitations of all tools employed to secure a system.

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	8.9 Centralize Audit Logs Centralize, to the extent possible, audit log collection and retention across enterprise assets.		●	●
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.		●	●
v7	6.5 Central Log Management Ensure that appropriate logs are being aggregated to a central log management system for analysis and review.		●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1070, T1070.002, T1562, T1562.006, T1565	TA0040	M1029

6.1.3.4 Ensure rsyslog log file creation mode is configured (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

rsyslog will create logfiles that do not already exist on the system.

The **global()** configuration object **umask**, available in rsyslog 8.26.0+, sets the rsyslogd process' umask. If not specified, the system-provided default is used. The value given must always be a 4-digit octal number, with the initial digit being zero.

The legacy **\$umask** parameter sets the **rsyslogd** process' umask. If not specified, the system-provided default is used. The value given must always be a 4-digit octal number, with the initial digit being zero.

The legacy **\$FileCreateMode** parameter allows the setting of the mode with which **rsyslogd** creates new files. If not specified, the value **0644** is used. The value given must always be a 4-digit octal number, with the initial digit being zero. Please note that the actual permission depend on **rsyslogd** process **umask**. If in doubt, use **\$umask 0000** right at the beginning of the configuration file to remove any restrictions.

The legacy **\$FileCreateMode** may be specified multiple times. If so, it specifies the creation mode for all selector lines that follow until the next **\$FileCreateMode** parameter. Order of lines is vitally important.

Rationale:

It is important to ensure that log files have the correct permissions to ensure that sensitive data is archived and protected.

Audit:

Run the following command

Run the following script to verify **\$FileCreateMode** to set to mode **0640** or more restrictive:

```

#!/usr/bin/env bash

{
    a_output=() a_output2=() l_analyze_cmd=$(readlink -f /bin/systemd-analyze)
    l_include='\$IncludeConfig' a_config_files=("rsyslog.conf")
    l_parameter_name='\$FileCreateMode'
    f_parameter_chk()
    {
        l_perm_mask="0137"; l_maxperm=$(( 0777 & ~$l_perm_mask )) "
        l_mode=$(awk '{print $2}' <<< "$l_used_parameter_setting" | xargs)"
        if [ $(( $l_mode & $l_perm_mask )) -gt 0 ]; then
            a_output2+=(" - Parameter: \"${l_parameter_name}///\}\\" is incorrectly set
to mode: \"\$l_file_mode\" \
                    " in the file: \"\$l_file\" " Should be mode: \"\$l_maxperm\" or more
restrictive")
        else
            a_output+=(" - Parameter: \"${l_parameter_name}///\}\\" is correctly set to
mode: \"\$l_file_mode\" \
                    " in the file: \"\$l_file\" " Should be mode: \"\$l_maxperm\" or more
restrictive")
        fi
    }
    while IFS= read -r l_file; do
        l_conf_loc=$(awk '$1~/^s*' "$l_include"'/ {print $2}' "$l_file" | tr -d '# ' <<<
"$l_file")" | tail -n 1"
        [ -n "$l_conf_loc" ] && break
        done < <($l_analyze_cmd cat-config "${a_config_files[@]}") | tac | grep -Pio
'^\h*\#\h*/[^#\n\r\h]+\.\conf\b'
        if [ -d "$l_conf_loc" ]; then
            l_dir="$l_conf_loc" l_ext="*"
        elif grep -Psq '\/\*\.\([^\#/n\r]+)\?\h*$' <<< "$l_conf_loc" || [ -f "$(readlink -f
"$l_conf_loc")" ]; then
            l_dir="$(dirname "$l_conf_loc")" l_ext="${basename "$l_conf_loc")"
        fi
        while read -r -d $'\0' l_file_name; do
            [ -f "$(readlink -f "$l_file_name")" ] && a_config_files+=("$(readlink -f
"$l_file_name")")
            done < <(find -L "$l_dir" -type f -name "$l_ext" -print0 2>/dev/null)
            while IFS= read -r l_file; do
                l_file=$(tr -d '# ' <<< "$l_file")
                l_used_parameter_setting=$(grep -PHs -- '^h*'"$l_parameter_name"'\\b' "$l_file"
| tail -n 1")
                [ -n "$l_used_parameter_setting" ] && break
                done < <($l_analyze_cmd cat-config "${a_config_files[@]}") | tac | grep -Pio
'^\h*\#\h*/[^#\n\r\h]+\.\conf\b'

                if [ -n "$l_used_parameter_setting" ]; then
                    f_parameter_chk
                else
                    a_output2+=(" - Parameter: \"${l_parameter_name}///\}\\" is not set in a
configuration file" \
                            " *** Note: \"${l_parameter_name}///\}\\" May be set in a file that's ignored
by load procedure ***")
                fi
                if [ "${#a_output2[@]}" -le 0 ]; then
                    printf '%s\n' "" "- Audit Result:" " ** PASS **" "${a_output[@]}" ""
                else
                    printf '%s\n' "" "- Audit Result:" " ** FAIL **" " - Reason(s) for audit
failure:" "${a_output2[@]}"
                    [ "${#a_output[@]}" -gt 0 ] && printf '%s\n' "" "- Correctly set:""
                    "${a_output[@]}" ""
                fi
            fi
        }
    }
}

```

Remediation:

Edit either `/etc/rsyslog.conf` or a dedicated `.conf` file in `/etc/rsyslog.d/` and set `$FileCreateMode` to `0640` or more restrictive:

```
$FileCreateMode 0640
```

Example script to create a drop-in configuration file in the default include location:

```
#!/usr/bin/env bash

{
    [ ! -d "/etc/rsyslog.d/" ] && mkdir /etc/rsyslog.d/
    printf '%s\n' "" "\$FileCreateMode 0640" >> /etc/rsyslog.d/60-rsyslog.conf
}
```

Reload the service:

```
# systemctl reload-or-restart rsyslog
```

References:

1. RSYSLOG.CONF(5)
2. NIST SP 800-53 Rev. 5: AC-3, AC-6, MP-2
3. <https://www.rsyslog.com/doc/>

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.	●	●	●
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	5.1 Establish Secure Configurations Maintain documented, standard security configuration standards for all authorized operating systems and software.	●	●	●
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.		●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1070, T1070.002, T1083, T1083.000	TA0007	M1022

6.1.3.5 Ensure rsyslog logging is configured (Manual)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The **rsyslog** and configuration files specifies rules for logging and which files are to be used to log certain classes of messages.

Rationale:

A great deal of important security-related information is sent via **rsyslog** (e.g., successful and failed su attempts, failed login attempts, root login attempts, etc.).

Audit:

Review the contents of **/etc/rsyslog.conf** and **/etc/rsyslog.d/*.conf** files to ensure appropriate logging is set. In addition, run the following command and verify that the log files are logging information as expected:

Run the following script and review the output from the **rsyslog** configuration to ensure appropriate logging is set an in accordance with local site policy.

```
#!/usr/bin/env bash

{
    l_analyze_cmd=$(readlink -f /bin/systemd-analyze)
    l_include='\$IncludeConfig' a_config_files=("rsyslog.conf")
    while IFS= read -r l_file; do
        l_conf_loc=$(awk '$1~/^s*'"$l_include"'$/ {print $2}' "$l_analyze_cmd" | tr -d '#'
        <<< "$l_file")" | tail -n 1)"
        [ -n "$l_conf_loc" ] && break
    done < <($l_analyze_cmd cat-config "${a_config_files[@]}") | tac | grep -
    Pio '^#\h*\#[\h*\v[^#\n\r\h]+\.\conf\b)'
    if [ -d "$l_conf_loc" ]; then
        l_dir="$l_conf_loc" l_ext="*"
    elif grep -Psq '\v\.\([^\#\n\r]+\)?\h*$' <<< "$l_conf_loc" || [ -f
    "$(readlink -f "$l_conf_loc")" ]; then
        l_dir="$(dirname "$l_conf_loc")" l_ext="${(basename "$l_conf_loc")}"
    fi
    while read -r -d $'\0' l_file_name; do
        [ -f "$(readlink -f "$l_file_name")" ] && a_config_files+=("$(readlink
        -f "$l_file_name")")
    done < <(find -L "$l_dir" -type f -name "$l_ext" -print0 2>/dev/null)
    for l_logfile in "${a_config_files[@]}"; do
        grep -PHs -- '^#\h*\v[^#\n\r\:/]+\var\log\.*\$' "$l_logfile"
    done
}
```

Example output:

/etc/rsyslog.d/60-rsyslog.conf:auth,authpriv.*	/var/log/secure
/etc/rsyslog.d/60-rsyslog.conf:mail.*	-/var/log/mail
/etc/rsyslog.d/60-rsyslog.conf:mail.info	-/var/log/mail.info
/etc/rsyslog.d/60-rsyslog.conf:mail.warning	-/var/log/mail.warn
/etc/rsyslog.d/60-rsyslog.conf:mail.err	/var/log/mail.err
/etc/rsyslog.d/60-rsyslog.conf:cron.*	/var/log/cron
/etc/rsyslog.d/60-rsyslog.conf:*.*=warning;*.=err	-/var/log/warn
/etc/rsyslog.d/60-rsyslog.conf:*.crit	/var/log/warn
/etc/rsyslog.d/60-rsyslog.conf:*.*.mail.none;news.none	-/var/log/messages
/etc/rsyslog.d/60-rsyslog.conf:local0,local1.*	-
/var/log/localmessages	
/etc/rsyslog.d/60-rsyslog.conf:local2,local3.*	-
/var/log/localmessages	
/etc/rsyslog.d/60-rsyslog.conf:local4,local5.*	-
/var/log/localmessages	
/etc/rsyslog.d/60-rsyslog.conf:local6,local7.*	-
/var/log/localmessages	
/etc/rsyslog.d/50-default.conf:auth,authpriv.*	/var/log/auth.log
#<- Will be ignored	
/etc/rsyslog.d/50-default.conf:*.*.auth,authpriv.none	-/var/log/syslog
/etc/rsyslog.d/50-default.conf:kern.*	-/var/log/kern.log
/etc/rsyslog.d/50-default.conf:mail.*	-/var/log/mail.log
#<- Will be ignored	
/etc/rsyslog.d/50-default.conf:mail.err	/var/log/mail.err
#<- Will be ignored	

Note:

- Output is generated as <CONFIGURATION_FILE>:<PARAMETER>
- Files are listed in order of precedence. If the same parameter is listed multiple times, only the first occurrence will be used by the **rsyslog** daemon

Remediation:

Edit the following lines in the configuration file(s) returned by the audit as appropriate for your environment.

Note: The below configuration is shown for example purposes only. Due care should be given to how the organization wishes to store log data.

. emerg	:omusrmsg:
auth,authpriv.*	/var/log/secure
mail.*	-/var/log/mail
mail.info	-/var/log/mail.info
mail.warning	-/var/log/mail.warn
mail.err	/var/log/mail.err
cron.*	/var/log/cron
.=warning;*.=err	-/var/log/warn
*.crit	/var/log/warn
..mail.none;news.none	-/var/log/messages
local0,local1.*	-/var/log/localmessages
local2,local3.*	-/var/log/localmessages
local4,local5.*	-/var/log/localmessages
local6,local7.*	-/var/log/localmessages

Run the following command to reload the **rsyslogd** configuration:

```
# systemctl reload-or-restart rsyslog
```

References:

1. See the rsyslog.conf(5) man page for more information.
2. NIST SP 800-53 Rev. 5: AU-2, AU-7, AU-12

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.	●	●	●
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.		●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1070, T1070.002	TA0005	M1047

6.1.3.6 Ensure rsyslog is configured to send logs to a remote log host (Manual)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

rsyslog supports the ability to send log events it gathers to a remote log host or to receive messages from remote hosts, thus enabling centralized log management.

Rationale:

Storing log data on a remote host protects log integrity from local attacks. If an attacker gains root access on the local system, they could tamper with or remove log data that is stored on the local system.

Audit:

Run the following script and review the output of **rsyslog** configuration. Verify that logs are sent to a central host used by your organization:

basic format

```
#!/usr/bin/env bash

{
    l_analyze_cmd=$(readlink -f /bin/systemd-analyze)
    l_include='\$IncludeConfig' a_config_files=("rsyslog.conf")
    while IFS= read -r l_file; do
        l_conf_loc=$(awk '$1~/^\\s*'"$l_include"'$/ {print $2}' "$tr -d '# '
<<< "$l_file")" | tail -n 1)
        [ -n "$l_conf_loc" ] && break
    done < <($l_analyze_cmd cat-config "${a_config_files[@]}") | tac | grep -
Pio '^\\h*#\\h*/[^#\\n\\r\\h]+\.\conf\b')
    if [ -d "$l_conf_loc" ]; then
        l_dir="$l_conf_loc" l_ext="*"
    elif grep -Psq '^/*\.([^\#\\n\\r]+)?\\h*$' <<< "$l_conf_loc" || [ -f
"$(readlink -f "$l_conf_loc")" ]; then
        l_dir="$(dirname "$l_conf_loc")" l_ext="$(basename "$l_conf_loc")"
    fi
    while read -r -d $'\0' l_file_name; do
        [ -f "$(readlink -f "$l_file_name")" ] && a_config_files+=("$(readlink
-f "$l_file_name")")
    done < <(find -L "$l_dir" -type f -name "$l_ext" -print0 2>/dev/null)
    for l_logfile in "${a_config_files[@]}"; do
        grep -Hs -- "^.+[^I][^I]*@" "$l_logfile"
    done
}
```

Output should include **@@<FQDN or IP of remote loghost>**:

Example output:

```
/etc/rsyslog.d/60-rsyslog.conf:.* @@loghost.example.com
```

- OR -

Run the following script and review the output of **rsyslog** configuration. Verify that logs are sent to a central host used by your organization:

advanced format

```

#!/usr/bin/env bash

{
    l_analyze_cmd=$(readlink -f /bin/systemd-analyze)
    l_include='\$IncludeConfig' a_config_files=("rsyslog.conf")
    while IFS= read -r l_file; do
        l_conf_loc=$(awk '$1~/^s*'"$l_include"'$/ {print $2}' "$tr -d '# ' <<< "$l_file") | tail -n 1"
        [ -n "$l_conf_loc" ] && break
    done < <($l_analyze_cmd cat-config "${a_config_files[@]}") | tac | grep -Pio '^h*\#h*/[^#\n\r\h]+\.\conf\b'
    if [ -d "$l_conf_loc" ]; then
        l_dir="$l_conf_loc" l_ext="*"
    elif grep -Psq '\/\*\*.([^\#\n\r]+)\?h*$' <<< "$l_conf_loc" || [ -f "$(readlink -f "$l_conf_loc")" ]; then
        l_dir=$(dirname "$l_conf_loc") l_ext=$(basename "$l_conf_loc")
    fi
    while read -r -d $'\0' l_file_name; do
        [ -f "$(readlink -f "$l_file_name")" ] && a_config_files+=("$(readlink -f "$l_file_name")")
    done < <(find -L "$l_dir" -type f -name "$l_ext" -print0 2>/dev/null)
    for l_logfile in "${a_config_files[@]}"; do
        grep -PHsi -- '^s*([^\#]+\s+)?action\(([^\#]+\s+)?bttarget=\\"?[^"]+\\"?\b' "$l_logfile"
    done
}

```

Output should include **target=<FQDN or IP of remote loghost>**:

Example output:

```
/etc/rsyslog.d/60-rsyslog.conf:.* action(type="omfwd"
target="loghost.example.com" port="514" protocol="tcp"
```

Remediation:

Edit the **rsyslog** configuration and add the following line (where **loghost.example.com** is the name of your central log host). The **target** directive may either be a fully qualified domain name or an IP address.

Example script to create a drop-in configuration file:

```

#!/usr/bin/env bash

{
    a_parameters='.* action(type="omfwd" target="loghost.example.com"
port="514" protocol="tcp" '
    '           action.resumeRetryCount="100" '
    queue.type="LinkedList" queue.size="1000")'
    [ ! -d "/etc/rsyslog.d/" ] && mkdir /etc/rsyslog.d/
    printf '%s\n' "" "${a_parameters[@]}" >> /etc/rsyslog.d/60-rsyslog.conf
}

```

Run the following command to reload **rsyslog.service**:

```
# systemctl reload-or-restart rsyslog.service
```

References:

1. See the `rsyslog.conf(5)` man page for more information.
2. NIST SP 800-53 Rev. 5: AU-6
3. <https://www.rsyslog.com/doc/>

Additional Information:

In addition, see the [rsyslog documentation](#) for implementation details of TLS.

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.	●	●	●
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.		●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1070, T1070.002, T1562, T1562.006	TA0040	M1029

6.1.3.7 Ensure rsyslog is not configured to receive logs from a remote client (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

rsyslog supports the ability to receive messages from remote hosts, thus acting as a log server. Clients should not receive data from other hosts.

Rationale:

If a client is configured to also receive data, thus turning it into a server, the client system is acting outside its operational boundary.

Audit:

Unless the system's primary function is to serve as a logfile server, run the following script to review the **rsyslog** configuration and verify that the system is not configured to accept incoming logs.

advanced format

```
#!/usr/bin/env bash

{
    a_output2=()
    l_analyze_cmd=$(readlink -f /bin/systemd-analyze)
    l_include='\$IncludeConfig' a_config_files=("rsyslog.conf")
    while IFS= read -r l_file; do
        l_conf_loc=$(awk '$1~/^s*'"$l_include"'$/ {print $2}' '$(tr -d '# '
<<< "$l_file")' | tail -n 1)"
        [ -n "$l_conf_loc" ] && break
    done <<($l_analyze_cmd cat-config "${a_config_files[@]}") | tac | grep -
Pio '^h#\h*/[^#\n\r\h]+\.\conf\b'
    if [ -d "$l_conf_loc" ]; then
        l_dir="$l_conf_loc" l_ext="*"
    elif grep -Psq '\/*\.([^#\n\r]+)\h*$' << "$l_conf_loc" || [ -f
"${readlink -f "$l_conf_loc")" ]; then
        l_dir=$(dirname "$l_conf_loc") l_ext=$(basename "$l_conf_loc")"
    fi
    while read -r -d $'\0' l_file_name; do
        [ -f "$(readlink -f "$l_file_name")" ] && a_config_files+=("$(readlink
-f "$l_file_name")")
    done <<(find -L "$l_dir" -type f -name "$l_ext" -print0 2>/dev/null)
    for l_logfile in "${a_config_files[@]}"; do
        l_fail=$(grep -Psi -- '^h*module\(load=?imtcp\?)"' "$l_logfile)"
        [ -n "$l_fail" ] && a_output2+=("- Advanced format entry to accept
incoming logs: \"$l_fail\" found in: \"$l_logfile\")"
        l_fail=$(grep -Psi -- '^h*input\(type=?imtcp\?\b' "$l_logfile)"
        [ -n "$l_fail" ] && a_output2+=("- Advanced format entry to accept
incoming logs: \"$l_fail\" found in: \"$l_logfile\")"
        l_fail=$(grep -Psi -- '^h*module\(load=?imtcp\?)"' "$l_logfile)"
        [ -n "$l_fail" ] && a_output2+=("- Obsolete format entry to accept
incoming logs: \"$l_fail\" found in: \"$l_logfile\")"
        l_fail=$(grep -Psi -- '^h*input\(type=?imtcp\?\b' "$l_logfile)"
        [ -n "$l_fail" ] && a_output2+=("- Obsolete format entry to accept
incoming logs: \"$l_fail\" found in: \"$l_logfile\")"
    done
    if [ "${#a_output2[@]}" -le "0" ]; then
        printf '%s\n' "" "- Audit Result:" " ** PASS **" " - No entries to
accept incoming logs found"
    else
        printf '%s\n' "" "- Audit Result:" " ** FAIL **" " - Reason(s) for
audit failure: "${a_output2[@]}"
    fi
}
```

Remediation:

Unless the system's primary function is to serve as a logfile server , modify the files returned by the Audit Procedure and remove the specific lines highlighted by the audit. Verify none of the following entries are present in the **rsyslog** configuration.

advanced format

```
module(load="imtcp")
input(type="imtcp" port="514")
```

deprecated legacy format

```
$ModLoad imtcp
$InputTCPServerRun
```

Reload the service:

```
# systemctl reload-or-restart rsyslog
```

References:

1. NIST SP 800-53 Rev. 5: AU-2, AU-7, AU-12, CM-6
2. <https://www.rsyslog.com/doc/index.html>

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.		●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1070, T1070.002, T1562, T1562.006	TA0005, TA0040	M1029

6.1.3.8 Ensure logrotate is configured (Manual)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The system includes the capability of rotating log files regularly to avoid filling up the system with logs or making the logs unmanageably large. The file `/etc/logrotate.d/rsyslog` is the configuration file used to rotate log files created by `rsyslog`.

Rationale:

By keeping the log files smaller and more manageable, a system administrator can easily archive these files to another system and spend less time looking through inordinately large log files.

Audit:

Run the following script to analyze the `logrotate` configuration:

```
#!/usr/bin/env bash

{
    l_analyze_cmd=$(readlink -f /bin/systemd-analyze)
    l_config_file="/etc/logrotate.conf"
    l_include=$(awk '$1~/^s*include$/ {print$2}' "$l_config_file"
2>/dev/null)"
    [ -d "$l_include" ] && l_include="$l_include/*"
    $l_analyze_cmd cat-config "$l_config_file" $l_include
}
```

Note: The last occurrence of a argument is the one used for the `logrotate` configuration

Remediation:

Edit `/etc/logrotate.conf`, or the appropriate configuration file provided by the script in the Audit Procedure, as necessary to ensure logs are rotated according to site policy.

References:

1. NIST SP 800-53 Rev. 5: AU-8

Additional Information:

If no `maxage` setting is set for `logrotate` a situation can occur where `logrotate` is interrupted and fails to delete rotated log files. It is recommended to set this to a value greater than the longest any log file should exist on your system to ensure that any such log file is removed but standard rotation settings are not overridden.

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.	●	●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1070, T1070.002	TA0040	M1022

6.1.4 Configure Logfiles

6.1.4.1 Ensure access to all logfiles has been configured (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

Log files contain information from many services on the local system, or in the event of a centralized log server, others systems logs as well.

In general log files are found in `/var/log/`, although application can be configured to store logs elsewhere. Should your application store logs in another, ensure to run the same test on that location.

Rationale:

It is important that log files have the correct permissions to ensure that sensitive data is protected and that only the appropriate users / groups have access to them.

Audit:

Run the following script to verify that files in `/var/log/` have appropriate permissions and ownership:

```
#!/usr/bin/env bash

{
    a_output=(); a_output2=()
    f_file_test_chk()
    {
        a_out2=()
        maxperm=$( printf '%o' $(( 0777 & ~$perm_mask)) )
        [ $(( $1_mode & $perm_mask )) -gt 0 ] && \
            a_out2+=(" o Mode: \"\$1_mode\" should be \"\$maxperm\" or more restrictive")
        [[ ! \"$1_user\" =~ \$1_auser ]] && \
            a_out2+=(" o Owned by: \"\$1_user\" and should be owned by \"\$1_auser//|/ or }\"")
        [[ ! \"$1_group\" =~ \$1_agroup ]] && \
            a_out2+=(" o Group owned by: \"\$1_group\" and should be group owned by \
\"\$1_agroup//|/ or }\"")
        [ "${#a_out2[@]}" -gt 0 ] && a_output2+=(" - File: \"\$1_fname\" is:" "${a_out2[@]}")
    }
    while IFS= read -r -d '$\0' l_file; do
        while IFS=: read -r l_fname l_mode l_user l_group; do
            if grep -Pq -- '/(apt)\h*$' <<< "$(\$l_fname)"; then
                perm_mask='0133' l_auser="root" l_agroup="(root|adm)"; f_file_test_chk
            else
                case "$(basename \$l_fname)" in
                    lastlog | lastlog.* | wtmp | wtmp.* | wtmp-* | btmp | btmp.* | btmp-* | README)
                        perm_mask='0113' l_auser="root" l_agroup="(root|utmp)"
                        f_file_test_chk ;;
                    cloud-init.log* | localmessages* | waagent.log*)
                        perm_mask='0133' l_auser="(root|syslog)" l_agroup="(root|adm)"
                        file test_chk ;;
                    secure{,.*,..*,-*} | auth.log | syslog | messages)
                        perm_mask='0137' l_auser="(root|syslog)" l_agroup="(root|adm)"
                        f_file_test_chk ;;
                    SSSD | sssd)
                        perm_mask='0117' l_auser="(root|SSSD)" l_agroup="(root|SSSD)"
                        f_file_test_chk ;;
                    gdm | gdm3)
                        perm_mask='0117' l_auser="root" l_agroup="(root|gdm|gdm3)"
                        f_file_test_chk ;;
                    *.journal | *.journal~)
                        perm_mask='0137' l_auser="root" l_agroup="(root|systemd-journal)"
                        f_file_test_chk ;;
                    *)
                        perm_mask='0137' l_auser="(root|syslog)" l_agroup="(root|adm)"
                        if [ "$l_user" = "root" ] || ! grep -Pq -- '^h*$(awk -F: '$1==""\$l_user'\""
{print \$7}' /etc/passwd)\b" /etc/shells; then
                            ! grep -Pq -- "\$l_auser" <<< "\$l_user" && l_auser="(root|syslog|\$l_user)"
                            ! grep -Pq -- "\$l_agroup" <<< "\$l_group" && l_agroup="(root|adm|\$l_group)"
                        fi
                        f_file_test_chk ;;
                    esac
                fi
            done < <(stat -Lc '%n:%#a:%U:%G' "\$l_file")
            done < <(find -L /var/log -type f \(\ -perm /0137 -o ! -user root -o ! -group root \) -print0)
            if [ "${#a_output2[@]}" -le 0 ]; then
                a_output+=(" - All files in \"/var/log/\" have appropriate permissions and ownership")
                printf '\n%s' "- Audit Result:" " ** PASS **" "${a_output[@]}" ""
            else
                printf '\n%s' "- Audit Result:" " ** FAIL **" " - Reason(s) for audit failure:"
                "${a_output2[@]}"
            fi
        }
    }
}
```

Remediation:

Run the following script to update permissions and ownership on files in `/var/log`. Although the script is not destructive, ensure that the output of the audit procedure is captured in the event that the remediation causes issues.

```

#!/usr/bin/env bash

{
    a_output2=()
    f_file_test_fix()
    {
        a_out2=()
        maxperm=$( printf '%o' $(( 0777 & ~$perm_mask)) )
        if [ $(( $l_mode & $perm_mask )) -gt 0 ]; then
            a_out2+=(" o Mode: \"$l_mode\" should be \"$maxperm\" or more
restrictive" " x Removing excess permissions")
            chmod "$l_rperms" "$l_fname"
        fi
        if [[ ! "$l_user" =~ $l_auser ]]; then
            a_out2+=(" o Owned by: \"$l_user\" and should be owned by
\"${l_auser//\// or }\" " " x Changing ownership to: \"$l_fix_account\"")
            chown "$l_fix_account" "$l_fname"
        fi
        if [[ ! "$l_group" =~ $l_agroup ]]; then
            a_out2+=(" o Group owned by: \"$l_group\" and should be group
owned by \"${l_agroup//\// or }\" " " x Changing group ownership to:
\"$l_fix_account\"")
            chgrp "$l_fix_account" "$l_fname"
        fi
        [ "${#a_out2[@]}" -gt 0 ] && a_output2+=(" - File: \"$l_fname\" is:"
"${a_out2[@]}")
    }
    l_fix_account='root'
    while IFS= read -r -d $'\0' l_file; do
        while IFS=: read -r l_fname l_mode l_user l_group; do
            if grep -Pq -- '\/(apt)\h*$' <<< "$dirname \"$l_fname\")"; then
                perm_mask='0133' l_rperms="u-x,go-wx" l_auser="root"
            l_agroup="(root|adm)"; f_file_test_fix
            else
                case "$(basename \"$l_fname\")" in
                    lastlog | lastlog.* | wtmp | wtmp.* | wtmp-* | btmp | btmp.* |
btmp-* | README)
                        perm_mask='0113' l_rperms="ug-x,o-wx" l_auser="root"
                l_agroup="(root|utmp)"
                        f_file_test_fix ;;
                    cloud-init.log* | localmessages* | waagent.log*)
                        perm_mask='0133' l_rperms="u-x,go-wx"
                l_auser="(root|syslog)" l_agroup="(root|adm)"
                        file_test_fix ;;
                    secure | auth.log | syslog | messages)
                        perm_mask='0137' l_rperms="u-x,g-wx,o-rwx"
                l_auser="(root|syslog)" l_agroup="(root|adm)"
                        f_file_test_fix ;;
                    SSSD | sssd)
                        perm_mask='0117' l_rperms="ug-x,o-rwx"
                l_auser="(root|SSSD)" l_agroup="(root|SSSD)"
                        f_file_test_fix ;;
                    gdm | gdm3)
                        perm_mask='0117' l_rperms="ug-x,o-rwx" l_auser="root"
                l_agroup="(root|gdm|gdm3)"
                        f_file_test_fix ;;
                    *.journal | *.journal~)
                        ;;
                esac
            fi
        done
    done
}

```

```

perm_mask='0137' l_rperms="u-x,g-wx,o-rwx" l_auser="root"
l_agroup="(root|systemd-journal)"
        f_file_test_fix ;;
*)
perm_mask='0137' l_rperms="u-x,g-wx,o-rwx"
l_auser="(root|syslog)" l_agroup="(root|adm)"
        if [ "$l_user" = "root" ] || ! grep -Pq -- "^h*(awk -F:
'$1==""$l_user'"' {print $7}' /etc/passwd)\b" /etc/shells; then
                ! grep -Pq -- "$l_auser" <<< "$l_user" &&
l_auser="(root|syslog|$l_user)"
                ! grep -Pq -- "$l_agroup" <<< "$l_group" &&
l_agroup="(root|adm|$l_group)"
                fi
                f_file_test_fix ;;
esac
fi
done < <(stat -Lc '%n:%#a:%U:%G' "$l_file")
done < <(find -L /var/log -type f \(\ -perm /0137 -o ! -user root -o ! -
group root \) -print0)
if [ "${#a_output2[@]}" -le 0 ]; then # If all files passed, then we
report no changes
        a_output+=(" - All files in \"/var/log/\" have appropriate permissions
and ownership")
        printf '\n%s' "- All files in \"/var/log/\" have appropriate
permissions and ownership" " o No changes required" ""
else
        printf '\n%s' "${a_output2[@]}" ""
fi
}

```

Note: You may also need to change the configuration for your logging software or services for any logs that had incorrect permissions.

If there are services that log to other locations, ensure that those log files have the appropriate permissions.

References:

1. NIST SP 800-53 Rev. 5: AC-3, MP-2

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<p>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.</p>	●	●	●
v7	<p>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.</p>	●	●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1070, T1070.002, T1083, T1083.000	TA0007	M1028

6.2 System Auditing

The Linux Auditing System operates on a set of rules that collects certain types of system activity to facilitate incident investigation, detect unauthorized access or modification of data. By default events will be logged to `/var/log/audit/audit.log`, which can be configured in `/etc/audit/auditd.conf`.

The following types of audit rules can be specified:

- Control rules: Configuration of the auditing system.
- File system rules: Allow the auditing of access to a particular file or a directory. Also known as file watches.
- System call rules: Allow logging of system calls that any specified program makes.

Audit rules can be set:

- On the command line using the `auditctl` utility. These rules are not persistent across reboots.
- In `/etc/audit/audit.rules`. These rules have to be merged and loaded before they are active.

Notes:

- For 64 bit systems that have `arch` as a rule parameter, you will need two rules: one for 64 bit and one for 32 bit systems calls.
- If the auditing system is configured to be locked (`-e 2`), a system reboot will be required in order to load any changes.
- Key names are optional on the rules and will not be used as a compliance auditing. The usage of key names is highly recommended as it facilitates organization and searching, as such, all remediation steps will have key names supplied.
- It is best practice to store the rules, in number prepended files, in `/etc/audit/rules.d/`. Rules must end in a `.rules` suffix. This then requires the use of `augenrules` to merge all the rules into `/etc/audit/audit.rules` based on their their alphabetical (lexical) sort order. All benchmark recommendations follow this best practice for remediation, specifically using the prefix of `50` which is centre weighed if all rule sets make use of the number prepending naming convention.
- Your system may have been customized to change the default `UID_MIN`. All samples output uses `1000`, but this value will not be used in compliance auditing. To confirm the `UID_MIN` for your system, run the following command: `awk '/^s*UID_MIN/{print $2}' /etc/login.defs`

Normalization

The Audit system normalizes some entries, so when you look at the sample output keep in mind that:

- With regards to users whose login UID is not set, the values **-1** / **unset** / **4294967295** are equivalent and normalized to **-1**.
- When comparing field types and both sides of the comparison is valid fields types, such as **euid!=uid**, then the auditing system may normalize such that the output is **uid!=euid**.
- Some parts of the rule may be rearranged whilst others are dependant on previous syntax. For example, the following two statements are the same:

```
-a always,exit -F arch=b64 -S execve -C uid!=euid -F auid==1 -F  
key=user_emulation
```

and

```
-a always,exit -F arch=b64 -C euid!=uid -F auid!=unset -S execve -k  
user_emulation
```

Capacity planning

The recommendations in this section implement auditing policies that not only produces large quantities of logged data, but may also negatively impact system performance. Capacity planning is critical in order not to adversely impact production environments.

- Disk space. If a significantly large set of events are captured, additional on system or off system storage may need to be allocated. If the logs are not sent to a remote log server, ensure that log rotation is implemented else the disk will fill up and the system will halt. Even when logs are sent to a log server, ensure sufficient disk space to allow caching of logs in the case of temporary network outages.
- Disk IO. It is not just the amount of data collected that should be considered, but the rate at which logs are generated.
- CPU overhead. System call rules might incur considerable CPU overhead. Test the systems open/close syscalls per second with and without the rules to gauge the impact of the rules.

6.2.1 Configure auditd Service

The capturing of system events provides system administrators with information to allow them to determine if unauthorized access to their system is occurring.

6.2.1.1 Ensure auditd packages are installed (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

auditd is the userspace component to the Linux Auditing System. It's responsible for writing audit records to the disk

Rationale:

The capturing of system events provides system administrators with information to allow them to determine if unauthorized access to their system is occurring.

Audit:

Run the following command and verify **auditd** is installed:

```
# dpkg-query -s auditd &>/dev/null && echo auditd is installed  
auditd is installed
```

Run the following command to verify **audispd-plugins** is installed:

```
# dpkg-query -s audispd-plugins &>/dev/null && echo audispd-plugins is  
installed  
audispd-plugins is installed
```

Remediation:

Run the following command to Install **auditd** and **audispd-plugins**

```
# apt install auditd audispd-plugins
```

References:

1. NIST SP 800-53 Rev. 5: AU-2, AU-3, AU-12, SI-5

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<p>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.</p>		●	●
v7	<p>6.2 Activate audit logging Ensure that local logging has been enabled on all systems and networking devices.</p>	●	●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562, T1562.001	TA0005	M1018

6.2.1.2 Ensure auditd service is enabled and active (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

Turn on the **auditd** daemon to record system events.

Rationale:

The capturing of system events provides system administrators with information to allow them to determine if unauthorized access to their system is occurring.

Audit:

Run the following command to verify **auditd** is enabled:

```
# systemctl is-enabled auditd | grep '^enabled'  
enabled
```

Verify result is "enabled".

Run the following command to verify **auditd** is active:

```
# systemctl is-active auditd | grep '^active'  
active
```

Verify result is active

Remediation:

Run the following commands to unmask, enable and start **auditd**:

```
# systemctl unmask auditd  
# systemctl enable auditd  
# systemctl start auditd
```

References:

1. NIST SP 800-53 Rev. 5: AU-2, AU-12, SI-5

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<p>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.</p>	●	●	●
v7	<p>6.2 Activate audit logging Ensure that local logging has been enabled on all systems and networking devices.</p>	●	●	●
v7	<p>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.</p>		●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562, T1562.001	TA0005	M1028

6.2.1.3 Ensure auditing for processes that start prior to auditd is enabled (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

Configure **grub2** so that processes that are capable of being audited can be audited even if they start up prior to **auditd** startup.

Rationale:

Audit events need to be captured on processes that start up prior to **auditd**, so that potential malicious activity cannot go undetected.

Audit:

Run the following command:

```
# find /boot -type f -name 'grub.cfg' -exec grep -Ph -- '^\\h*linux' {} + | grep -v 'audit=1'
```

Nothing should be returned.

Remediation:

Edit **/etc/default/grub** and add **audit=1** to **GRUB_CMDLINE_LINUX**:

Example:

```
GRUB_CMDLINE_LINUX="audit=1"
```

Run the following command to update the **grub2** configuration:

```
# update-grub
```

References:

1. NIST SP 800-53 Rev. 5: AU-2, AU-3, AU-12

Additional Information:

This recommendation is designed around the grub2 bootloader, if another bootloader is in use in your environment enact equivalent settings.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<p>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.</p>	●	●	●
v7	<p>6.2 Activate audit logging Ensure that local logging has been enabled on all systems and networking devices.</p>	●	●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562, T1562.001	TA0005	M1047

6.2.1.4 Ensure audit_backlog_limit is sufficient (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

In the kernel-level audit subsystem, a socket buffer queue is used to hold audit events. Whenever a new audit event is received, it is logged and prepared to be added to this queue.

The kernel boot parameter **audit_backlog_limit=N**, with **N** representing the amount of messages, will ensure that a queue cannot grow beyond a certain size. If an audit event is logged which would grow the queue beyond this limit, then a failure occurs and is handled according to the system configuration

Rationale:

If an audit event is logged which would grow the queue beyond the **audit_backlog_limit**, then a failure occurs, auditd records will be lost, and potential malicious activity could go undetected.

Audit:

Run the following command and verify the **audit_backlog_limit=** parameter is set:

```
# find /boot -type f -name 'grub.cfg' -exec grep -Ph -- '^h*linux' {} + | grep -Pv 'audit_backlog_limit=\d+\b'
```

Nothing should be returned.

Remediation:

Edit **/etc/default/grub** and add **audit_backlog_limit=N** to GRUB_CMDLINE_LINUX. The recommended size for **N** is **8192** or larger.

Example:

```
GRUB_CMDLINE_LINUX="audit_backlog_limit=8192"
```

Run the following command to update the **grub2** configuration:

```
# update-grub
```

Default Value:

if **audit_backlog_limit** is not set, the system defaults to **audit_backlog_limit=64**

References:

1. NIST SP 800-53 Rev. 5: AU-2, AU-3, AU-12

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.	●	●	●
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.		●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562, T1562.001	TA0005	M1028

6.2.2 Configure Data Retention

When auditing, it is important to carefully configure the storage requirements for audit logs. By default, auditd will max out the log files at 5MB and retain only 4 copies of them. Older versions will be deleted. It is possible on a system that the 20 MBs of audit logs may fill up the system causing loss of audit data. While the recommendations here provide guidance, check your site policy for audit storage requirements.

6.2.2.1 Ensure audit log storage size is configured (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

Configure the maximum size of the audit log file. Once the log reaches the maximum size, it will be rotated and a new log file will be started.

Rationale:

It is important that an appropriate size is determined for log files so that they do not impact the system and audit data is not lost.

Audit:

Run the following command and ensure output is in compliance with site policy:

```
# grep -Po -- '^h*max_log_file\h*=\h*\d+\b' /etc/audit/auditd.conf  
max_log_file = <MB>
```

Remediation:

Set the following parameter in [*/etc/audit/auditd.conf*](#) in accordance with site policy:

```
max_log_file = <MB>
```

Default Value:

`max_log_file = 8`

References:

1. NIST SP 800-53 Rev. 5: AU-8

Additional Information:

The `max_log_file` parameter is measured in megabytes.

Other methods of log rotation may be appropriate based on site policy. One example is time-based rotation strategies which don't have native support in auditd configurations. Manual audit of custom configurations should be evaluated for effectiveness and completeness.

CIS Controls:

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562, T1562.006	TA0040	M1053

6.2.2.2 Ensure audit logs are not automatically deleted (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

The `max_log_file_action` setting determines how to handle the audit log file reaching the max file size. A value of `keep_logs` will rotate the logs but never delete old logs.

Rationale:

In high security contexts, the benefits of maintaining a long audit history exceed the cost of storing the audit history.

Audit:

Run the following command and verify output matches:

```
# grep max_log_file_action /etc/audit/auditd.conf
max_log_file_action = keep_logs
```

Remediation:

Set the following parameter in `/etc/audit/auditd.conf`:

```
max_log_file_action = keep_logs
```

References:

1. NIST SP 800-53 Rev. 5: AU-8

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.		●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562, T1562.006	TA0005	M1053

6.2.2.3 Ensure system is disabled when audit logs are full (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

The **auditd** daemon can be configured to halt the system or put the system in single user mode, if no free space is available or an error is detected on the partition that holds the audit log files.

The **disk_full_action** parameter tells the system what action to take when no free space is available on the partition that holds the audit log files. Valid values are **ignore**, **syslog**, **rotate**, **exec**, **suspend**, **single**, and **halt**.

- **ignore**, the audit daemon will issue a syslog message but no other action is taken
- **syslog**, the audit daemon will issue a warning to syslog
- **rotate**, the audit daemon will rotate logs, losing the oldest to free up space
- **exec**, /path-to-script will execute the script. You cannot pass parameters to the script. The script is also responsible for telling the auditd daemon to resume logging once its completed its action
- **suspend**, the audit daemon will stop writing records to the disk
- **single**, the audit daemon will put the computer system in single user mode
- **halt**, the audit daemon will shut down the system

The **disk_error_action** parameter tells the system what action to take when an error is detected on the partition that holds the audit log files. Valid values are **ignore**, **syslog**, **exec**, **suspend**, **single**, and **halt**.

- **ignore**, the audit daemon will not take any action
- **syslog**, the audit daemon will issue no more than 5 consecutive warnings to syslog
- **exec**, /path-to-script will execute the script. You cannot pass parameters to the script
- **suspend**, the audit daemon will stop writing records to the disk
- **single**, the audit daemon will put the computer system in single user mode
- **halt**, the audit daemon will shut down the system

Rationale:

In high security contexts, the risk of detecting unauthorized access or nonrepudiation exceeds the benefit of the system's availability.

Impact:

disk_full_action parameter:

- Set to **halt** - the **auditd** daemon will shutdown the system when the disk partition containing the audit logs becomes full.
- Set to **single** - the **auditd** daemon will put the computer system in single user mode when the disk partition containing the audit logs becomes full.

disk_error_action parameter:

- Set to **halt** - the **auditd** daemon will shutdown the system when an error is detected on the partition that holds the audit log files.
- Set to **single** - the **auditd** daemon will put the computer system in single user mode when an error is detected on the partition that holds the audit log files.
- Set to **syslog** - the **auditd** daemon will issue no more than 5 consecutive warnings to syslog when an error is detected on the partition that holds the audit log files.

Audit:

Run the following command and verify the **disk_full_action** is set to either **halt** or **single**:

```
# grep -Pi -- '^h*disk_full_action\h*=\h*(halt|single)\b'  
/etc/audit/auditd.conf  
  
disk_full_action = <halt|single>
```

Run the following command and verify the **disk_error_action** is set to **syslog**, **single**, or **halt**:

```
# grep -Pi -- '^h*disk_error_action\h*=\h*(syslog|single|halt)\b'  
/etc/audit/auditd.conf  
  
disk_error_action = <syslog|single|halt>
```

Remediation:

Set one of the following parameters in `/etc/audit/auditd.conf` depending on your local security policies.

```
disk_full_action = <halt|single>
disk_error_action = <syslog|single|halt>
```

Example:

```
disk_full_action = halt
disk_error_action = halt
```

References:

1. NIST SP 800-53 Rev. 5: AU-2, AU-8, AU-12, SI-5
2. AUDITD.CONF(5)

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	8.3 Ensure Adequate Audit Log Storage Ensure that logging destinations maintain adequate storage to comply with the enterprise's audit log management process.	●	●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562, T1562.006	TA0005	M1028

6.2.2.4 Ensure system warns when audit logs are low on space (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

The **auditd** daemon can be configured to halt the system, put the system in single user mode or send a warning message, if the partition that holds the audit log files is low on space.

The **space_left_action** parameter tells the system what action to take when the system has detected that it is starting to get low on disk space. Valid values are **ignore**, **syslog**, **rotate**, **email**, **exec**, **suspend**, **single**, and **halt**.

- **ignore**, the audit daemon does nothing
- **syslog**, the audit daemon will issue a warning to syslog
- **rotate**, the audit daemon will rotate logs, losing the oldest to free up space
- **email**, the audit daemon will send a warning to the email account specified in **action_mail_acct** as well as sending the message to syslog
- **exec**, /path-to-script will execute the script. You cannot pass parameters to the script. The script is also responsible for telling the auditd daemon to resume logging once its completed its action
- **suspend**, the audit daemon will stop writing records to the disk
- **single**, the audit daemon will put the computer system in single user mode
- **halt**, the audit daemon will shut down the system

The **admin_space_left_action** parameter tells the system what action to take when the system has detected that it is low on disk space. Valid values are **ignore**, **syslog**, **rotate**, **email**, **exec**, **suspend**, **single**, and **halt**.

- **ignore**, the audit daemon does nothing
- **syslog**, the audit daemon will issue a warning to syslog
- **rotate**, the audit daemon will rotate logs, losing the oldest to free up space
- **email**, the audit daemon will send a warning to the email account specified in **action_mail_acct** as well as sending the message to syslog
- **exec**, /path-to-script will execute the script. You cannot pass parameters to the script. The script is also responsible for telling the auditd daemon to resume logging once its completed its action
- **suspend**, the audit daemon will stop writing records to the disk
- **single**, the audit daemon will put the computer system in single user mode
- **halt**, the audit daemon will shut down the system

Rationale:

In high security contexts, the risk of detecting unauthorized access or nonrepudiation exceeds the benefit of the system's availability.

Impact:

If the `admin_space_left_action` is set to `single` the audit daemon will put the computer system in single user mode.

Audit:

Run the following command and verify the `space_left_action` is set to `email`, `exec`, `single`, or `halt`:

```
# grep -P -- '^h*space_left_action\h*=h*(email|exec|single|halt)\b'  
/etc/audit/auditd.conf
```

Verify the output is `email`, `exec`, `single`, or `halt`

Example output

```
space_left_action = email
```

Run the following command and verify the `admin_space_left_action` is set to `single` - OR - `halt`:

```
# grep -P -- '^h*admin_space_left_action\h*=h*(single|halt)\b'  
/etc/audit/auditd.conf
```

Verify the output is `single` or `halt`

Example output:

```
admin_space_left_action = single
```

Note: A Mail Transfer Agent (MTA) must be installed and configured properly to set `space_left_action = email`

Remediation:

Set the `space_left_action` parameter in `/etc/audit/auditd.conf` to `email`, `exec`, `single`, or `halt`:

Example:

```
space_left_action = email
```

Set the `admin_space_left_action` parameter in `/etc/audit/auditd.conf` to `single` or `halt`:

Example:

```
admin_space_left_action = single
```

Note: A Mail Transfer Agent (MTA) must be installed and configured properly to set `space_left_action = email`

References:

1. NIST SP 800-53 Rev. 5: AU-2, AU-8, AU-12, SI-5
2. AUDITD.CONF(5)

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	8.3 Ensure Adequate Audit Log Storage Ensure that logging destinations maintain adequate storage to comply with the enterprise's audit log management process.	●	●	●
v7	6.2 Activate audit logging Ensure that local logging has been enabled on all systems and networking devices.	●	●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562, T1562.006	TA0005	

6.2.3 Configure auditd Rules

The Audit system operates on a set of rules that define what is to be captured in the log files.

The following types of Audit rules can be specified:

- Control rules: Allow the Audit system's behavior and some of its configuration to be modified.
- File system rules: Allow the auditing of access to a particular file or a directory. (Also known as file watches)
- System call rules: Allow logging of system calls that any specified program makes.

Audit rules can be set:

- on the command line using the auditctl utility. Note that these rules are not persistent across reboots.
- in a file ending in `.rules` in the `/etc/audit/rules.d/` directory.

Note: The Linux Benchmarks are written and tested against x86_64 processor architecture. If you are running a different processor type, please review and update the audit rules for the processor architecture of the system

6.2.3.1 Ensure changes to system administration scope (`sudoers`) is collected (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

Monitor scope changes for system administrators. If the system has been properly configured to force system administrators to log in as themselves first and then use the `sudo` command to execute privileged commands, it is possible to monitor changes in scope. The file `/etc/sudoers`, or files in `/etc/sudoers.d`, will be written to when the file(s) or related attributes have changed. The audit records will be tagged with the identifier "scope".

Rationale:

Changes in the `/etc/sudoers` and `/etc/sudoers.d` files can indicate that an unauthorized change has been made to the scope of system administrator activity.

Audit:

On disk configuration

Run the following command to check the on disk rules:

```
# awk '/^ *-w/ \
&&/etc/sudoers/ \
&& / +p *wa/ \
&&(/ key= *[!-~]* *$/ || / -k *[!-~]* *$/)' /etc/audit/rules.d/*.rules
```

Verify the output matches:

```
-w /etc/sudoers -p wa -k scope
-w /etc/sudoers.d -p wa -k scope
```

Running configuration

Run the following command to check loaded rules:

```
# auditctl -l | awk '/^ *-w/ \
&&/etc/sudoers/ \
&& / +p *wa/ \
&&(/ key= *[!-~]* *$/ || / -k *[!-~]* *$/)'
```

Verify the output matches:

```
-w /etc/sudoers -p wa -k scope
-w /etc/sudoers.d -p wa -k scope
```

Remediation:

Edit or create a file in the [/etc/audit/rules.d/](#) directory, ending in [.rules](#) extension, with the relevant rules to monitor scope changes for system administrators.

Example:

```
# printf "
-w /etc/sudoers -p wa -k scope
-w /etc/sudoers.d -p wa -k scope
" >> /etc/audit/rules.d/50-scope.rules
```

Merge and load the rules into active configuration:

```
# augenrules --load
```

Check if reboot is required.

```
# if [[ $(auditctl -s | grep "enabled") =~ "2" ]]; then printf "Reboot
required to load rules\n"; fi
```

References:

1. NIST SP 800-53 Rev. 5: AU-3

Additional Information:

Potential reboot required

If the auditing configuration is locked (`-e 2`), then `augenrules` will not warn in any way that rules could not be loaded into the running configuration. A system reboot will be required to load the rules into the running configuration.

System call structure

For performance (`man 7 audit.rules`) reasons it is preferable to have all the system calls on one line. However, your configuration may have them on one line each or some other combination. This is important to understand for both the auditing and remediation sections as the examples given are optimized for performance as per the man page.

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	4.8 Log and Alert on Changes to Administrative Group Membership Configure systems to issue a log entry and alert when an account is added to or removed from any group assigned administrative privileges.		●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562, T1562.006	TA0004	M1047

6.2.3.2 Ensure actions as another user are always logged (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

sudo provides users with temporary elevated privileges to perform operations, either as the superuser or another user.

Rationale:

Creating an audit log of users with temporary elevated privileges and the operation(s) they performed is essential to reporting. Administrators will want to correlate the events written to the audit trail with the records written to **sudo**'s logfile to verify if unauthorized commands have been executed.

Audit:

On disk configuration

Run the following command to check the on disk rules:

```
# awk '/^ *-a *always,exit/ \
&& / -F *arch=b(32|64)/ \
&& / -F *auid!=unset/||/ -F *auid!=-1/||/ -F *auid!=4294967295/ ) \
&& / -C *euid!=uid/||/ -C *uid!=euid/ ) \
&& / -S *execve/ \
&& / key= *[!-~]* *$/||/ -k *[!-~]* *$/)' /etc/audit/rules.d/*.rules
```

Verify the output matches:

```
-a always,exit -F arch=b64 -C euid!=uid -F auid!=unset -S execve -k
user_emulation
-a always,exit -F arch=b32 -C euid!=uid -F auid!=unset -S execve -k
user_emulation
```

Running configuration

Run the following command to check loaded rules:

```
# auditctl -l | awk '/^ *-a *always,exit/ \
&& / -F *arch=b(32|64)/ \
&& / -F *auid!=unset/||/ -F *auid!=-1/||/ -F *auid!=4294967295/ ) \
&& / -C *euid!=uid/||/ -C *uid!=euid/ ) \
&& / -S *execve/ \
&& / key= *[!-~]* *$/||/ -k *[!-~]* *$/)'
```

Verify the output matches:

```
-a always,exit -F arch=b64 -S execve -C uid!=euid -F auid!=-1 -F
key=user_emulation
-a always,exit -F arch=b32 -S execve -C uid!=euid -F auid!=-1 -F
key=user_emulation
```

Remediation:

Create audit rules

Edit or create a file in the `/etc/audit/rules.d/` directory, ending in `.rules` extension, with the relevant rules to monitor elevated privileges.

Example:

```
# printf "
-a always,exit -F arch=b64 -C euid!=uid -F auid!=unset -S execve -k
user_emulation
-a always,exit -F arch=b32 -C euid!=uid -F auid!=unset -S execve -k
user_emulation
" >> /etc/audit/rules.d/50-user_emulation.rules
```

Load audit rules

Merge and load the rules into active configuration:

```
# augenrules --load
```

Check if reboot is required.

```
# if [[ $(auditctl -s | grep "enabled") =~ "2" ]]; then printf "Reboot
required to load rules\n"; fi
```

References:

1. NIST SP 800-53 Rev. 5: AU-3

Additional Information:

Potential reboot required

If the auditing configuration is locked (`-e 2`), then `augenrules` will not warn in any way that rules could not be loaded into the running configuration. A system reboot will be required to load the rules into the running configuration.

System call structure

For performance (`man 7 audit.rules`) reasons it is preferable to have all the system calls on one line. However, your configuration may have them on one line each or some other combination. This is important to understand for both the auditing and remediation sections as the examples given are optimized for performance as per the man page.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<p>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.</p>	●	●	●
v7	<p>4.9 Log and Alert on Unsuccessful Administrative Account Login Configure systems to issue a log entry and alert on unsuccessful logins to an administrative account.</p>	●	●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562, T1562.006	TA0004	M1047

6.2.3.3 Ensure events that modify the sudo log file are collected (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

Monitor the `sudo` log file. If the system has been properly configured to disable the use of the `su` command and force all administrators to have to log in first and then use `sudo` to execute privileged commands, then all administrator commands will be logged to `/var/log/sudo.log`. Any time a command is executed, an audit event will be triggered as the `/var/log/sudo.log` file will be opened for write and the executed administration command will be written to the log.

Rationale:

Changes in `/var/log/sudo.log` indicate that an administrator has executed a command or the log file itself has been tampered with. Administrators will want to correlate the events written to the audit trail with the records written to `/var/log/sudo.log` to verify if unauthorized commands have been executed.

Audit:

Note: This recommendation requires that the sudo logfile is configured. See guidance provided in the recommendation "Ensure sudo log file exists"

On disk configuration

Run the following command to check the on disk rules:

```
# {
  SUDO_LOG_FILE=$(grep -r logfile /etc/sudoers* | sed -e 's/.*logfile=//;s/,?'
.*//' -e 's/"//g' -e 's|/|\\||g')
  [ -n "${SUDO_LOG_FILE}" ] && awk "/^ *-w/ \
&&/"${SUDO_LOG_FILE}"/ \
&&/ +-p *wa/ \
&&(/ key= *[!-~]* *$/||/ -k *[!-~]* *$/)" /etc/audit/rules.d/*.rules \
|| printf "ERROR: Variable 'SUDO_LOG_FILE' is unset.\n"
}
```

Verify output of matches:

```
-w /var/log/sudo.log -p wa -k sudo_log_file
```

Running configuration

Run the following command to check loaded rules:

```
# {
  SUDO_LOG_FILE=$(grep -r logfile /etc/sudoers* | sed -e 's/.*logfile=//;s/,?'
.*//' -e 's/"//g' -e 's|/|\\||g')
  [ -n "${SUDO_LOG_FILE}" ] && auditctl -l | awk "/^ *-w/ \
&&/"${SUDO_LOG_FILE}"/ \
&&/ +-p *wa/ \
&&(/ key= *[!-~]* *$/||/ -k *[!-~]* *$/)" \
|| printf "ERROR: Variable 'SUDO_LOG_FILE' is unset.\n"
}
```

Verify output matches:

```
-w /var/log/sudo.log -p wa -k sudo_log_file
```

Remediation:

Note: This recommendation requires that the sudo logfile is configured. See guidance provided in the recommendation "Ensure sudo log file exists"

Edit or create a file in the [`/etc/audit/rules.d/`](#) directory, ending in `.rules` extension, with the relevant rules to monitor events that modify the sudo log file.

Example:

```
# {
SUDO_LOG_FILE=$(grep -r logfile /etc/sudoers* | sed -e 's/.*/logfile=//;s/,?'
.*//' -e 's//g')
[ -n "${SUDO_LOG_FILE}" ] && printf "
-w ${SUDO_LOG_FILE} -p wa -k sudo_log_file
" >> /etc/audit/rules.d/50-sudo.rules || printf "ERROR: Variable
'SUDO_LOG_FILE' is unset.\n"
}
```

Merge and load the rules into active configuration:

```
# augenrules --load
```

Check if reboot is required.

```
# if [[ $(auditctl -s | grep "enabled") =~ "2" ]]; then printf "Reboot
required to load rules\n"; fi
```

Additional Information:

Potential reboot required

If the auditing configuration is locked ([`-e 2`](#)), then `augenrules` will not warn in any way that rules could not be loaded into the running configuration. A system reboot will be required to load the rules into the running configuration.

System call structure

For performance ([`man 7 audit.rules`](#)) reasons it is preferable to have all the system calls on one line. However, your configuration may have them on one line each or some other combination. This is important to understand for both the auditing and remediation sections as the examples given are optimized for performance as per the man page.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<p>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.</p>	●	●	●
v7	<p>4.9 Log and Alert on Unsuccessful Administrative Account Login Configure systems to issue a log entry and alert on unsuccessful logins to an administrative account.</p>	●	●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562, T1562.006	TA0004	

6.2.3.4 Ensure events that modify date and time information are collected (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

Capture events where the system date and/or time has been modified. The parameters in this section are set to determine if the;

- `adjtimex` - tune kernel clock
- `settimeofday` - set time using `timeval` and `timezone` structures
- `stime` - using seconds since 1/1/1970
- `clock_settime` - allows for the setting of several internal clocks and timers

system calls have been executed. Further, ensure to write an audit record to the configured audit log file upon exit, tagging the records with a unique identifier such as "time-change".

Rationale:

Unexpected changes in system date and/or time could be a sign of malicious activity on the system.

Audit:

On disk configuration

Run the following command to check the on disk rules:

```
# {
awk '/^ *-a *always,exit/ \
&& -F *arch=b(32|64)/ \
&& -S/ \
&&(/ adjtimex/ \
||/settimeofday/ \
||/clock_settime/ ) \
&&(/ key= *[!-~]* *$/||/ -k *[!-~]* *$/)' /etc/audit/rules.d/*.rules

awk '/^ *-w/ \
&&/etc/localtime/ \
&& +p *wa/ \
&&(/ key= *[!-~]* *$/||/ -k *[!-~]* *$/)' /etc/audit/rules.d/*.rules
}
```

Verify output of matches:

```
-a always,exit -F arch=b64 -S adjtimex,settimeofday -k time-change
-a always,exit -F arch=b32 -S adjtimex,settimeofday -k time-change
-a always,exit -F arch=b64 -S clock_settime -F a0=0x0 -k time-change
-a always,exit -F arch=b32 -S clock_settime -F a0=0x0 -k time-change
-w /etc/localtime -p wa -k time-change
```

Running configuration

Run the following command to check loaded rules:

```
# {
auditctl -l | awk '/^ *-a *always,exit/ \
&& -F *arch=b(32|64)/ \
&& -S/ \
&&(/ adjtimex/ \
||/settimeofday/ \
||/clock_settime/ ) \
&&(/ key= *[!-~]* *$/||/ -k *[!-~]* *$/)' 

auditctl -l | awk '/^ *-w/ \
&&/etc/localtime/ \
&& +p *wa/ \
&&(/ key= *[!-~]* *$/||/ -k *[!-~]* *$/)' 
}
```

Verify the output includes:

```
-a always,exit -F arch=b64 -S adjtimex,settimoofday -F key=time-change  
-a always,exit -F arch=b32 -S settimoofday,adjtimex -F key=time-change  
-a always,exit -F arch=b64 -S clock_settime -F a0=0x0 -F key=time-change  
-a always,exit -F arch=b32 -S clock_settime -F a0=0x0 -F key=time-change  
-w /etc/localtime -p wa -k time-change
```

Remediation:

Create audit rules

Edit or create a file in the `/etc/audit/rules.d/` directory, ending in `.rules` extension, with the relevant rules to monitor events that modify date and time information.

Example:

```
# printf "  
-a always,exit -F arch=b64 -S adjtimex,settimoofday -k time-change  
-a always,exit -F arch=b32 -S adjtimex,settimoofday -k time-change  
-a always,exit -F arch=b64 -S clock_settime -F a0=0x0 -k time-change  
-a always,exit -F arch=b32 -S clock_settime -F a0=0x0 -k time-change  
-w /etc/localtime -p wa -k time-change  
" >> /etc/audit/rules.d/50-time-change.rules
```

Load audit rules

Merge and load the rules into active configuration:

```
# augenrules --load
```

Check if reboot is required.

```
# if [[ $(auditctl -s | grep "enabled") =~ "2" ]]; then printf "Reboot  
required to load rules\n"; fi
```

References:

1. NIST SP 800-53 Rev. 5: AU-3, CM-6

Additional Information:

Potential reboot required

If the auditing configuration is locked (`-e 2`), then `augenrules` will not warn in any way that rules could not be loaded into the running configuration. A system reboot will be required to load the rules into the running configuration.

System call structure

For performance (`man 7 audit.rules`) reasons it is preferable to have all the system calls on one line. However, your configuration may have them on one line each or some other combination. This is important to understand for both the auditing and remediation sections as the examples given are optimized for performance as per the man page.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<p>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.</p>	●	●	
v7	<p>5.5 Implement Automated Configuration Monitoring Systems Utilize a Security Content Automation Protocol (SCAP) compliant configuration monitoring system to verify all security configuration elements, catalog approved exceptions, and alert when unauthorized changes occur.</p>	●	●	

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562, T1562.006	TA0005	M1047

6.2.3.5 Ensure events that modify the system's network environment are collected (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

Record changes to network environment files or system calls. The below parameters monitors the following system calls, and write an audit event on system call exit:

- `sethostname` - set the systems host name
- `setdomainname` - set the systems domain name

The files being monitored are:

- `/etc/issue` and `/etc/issue.net` - messages displayed pre-login
- `/etc/hosts` - file containing host names and associated IP addresses
- `/etc/networks` - symbolic names for networks
- `/etc/network/` - directory containing network interface scripts and configurations files
- `/etc/netplan/` - central location for YAML networking configurations files

Rationale:

Monitoring system events that change network environments, such as `sethostname` and `setdomainname`, helps identify unauthorized alterations to host and domain names, which could compromise security settings reliant on these names. Changes to `/etc/hosts` can signal unauthorized attempts to alter machine associations with IP addresses, potentially redirecting users and processes to unintended destinations. Surveillance of `/etc/issue` and `/etc/issue.net` is crucial to detect intruders inserting false information to deceive users. Monitoring `/etc/network/` reveals modifications to network interfaces or scripts that may jeopardize system availability or security. Additionally, tracking changes in the `/etc/netplan/` directory ensures swift detection of unauthorized adjustments to network configurations. All audit records should be appropriately tagged for relevance

Audit:

On disk configuration

Run the following commands to check the on disk rules:

```
# awk '/^ *-a *always,exit/ \
&& -F arch=b(32|64) / \
&& -S/ \
&&(/sethostname/ \
||/setdomainname/) \
&&(/ key= *[!-~]* *$/ ||/ -k *[!-~]* *$/)' /etc/audit/rules.d/*.rules

# awk '/^ *-w/ \
&&(/etc/issue/ \
||/etc/issue.net/ \
||/etc/hosts/ \
||/etc/network/ \
||/etc/netplan/) \
&& +p *wa/ \
&&(/ key= *[!-~]* *$/ ||/ -k *[!-~]* *$/)' /etc/audit/rules.d/*.rules
```

Verify the output matches:

```
-a always,exit -F arch=b64 -S sethostname,setdomainname -k system-locale
-a always,exit -F arch=b32 -S sethostname,setdomainname -k system-locale
-w /etc/issue -p wa -k system-locale
-w /etc/issue.net -p wa -k system-locale
-w /etc/hosts -p wa -k system-locale
-w /etc/networks -p wa -k system-locale
-w /etc/network -p wa -k system-locale
-w /etc/netplan -p wa -k system-locale
```

Running configuration

Run the following command to check loaded rules:

```
# auditctl -l | awk '/^ *-a *always,exit/ \
&& -F arch=b(32|64) / \
&& -S/ \
&&(/sethostname/ \
||/setdomainname/) \
&&(/ key= *[!-~]* *$/ ||/ -k *[!-~]* *$/)' 

# auditctl -l | awk '/^ *-w/ \
&&(/etc/issue/ \
||/etc/issue.net/ \
||/etc/hosts/ \
||/etc/network/ \
||/etc/netplan/) \
&& +p *wa/ \
&&(/ key= *[!-~]* *$/ ||/ -k *[!-~]* *$/)'
```

Verify the output includes:

```
-a always,exit -F arch=b64 -S sethostname,setdomainname -F key=system-locale  
-a always,exit -F arch=b32 -S sethostname,setdomainname -F key=system-locale  
-w /etc/issue -p wa -k system-locale  
-w /etc/issue.net -p wa -k system-locale  
-w /etc/hosts -p wa -k system-locale  
-w /etc/networks -p wa -k system-locale  
-w /etc/network -p wa -k system-locale  
-w /etc/netplan -p wa -k system-locale
```

Remediation:

Create audit rules

Edit or create a file in the `/etc/audit/rules.d/` directory, ending in `.rules` extension, with the relevant rules to monitor events that modify the system's network environment.

Example:

```
# printf "  
-a always,exit -F arch=b64 -S sethostname,setdomainname -k system-locale  
-a always,exit -F arch=b32 -S sethostname,setdomainname -k system-locale  
-w /etc/issue -p wa -k system-locale  
-w /etc/issue.net -p wa -k system-locale  
-w /etc/hosts -p wa -k system-locale  
-w /etc/networks -p wa -k system-locale  
-w /etc/network/ -p wa -k system-locale  
-w /etc/netplan/ -p wa -k system-locale  
" >> /etc/audit/rules.d/50-system_locale.rules
```

Load audit rules

Merge and load the rules into active configuration:

```
# augenrules --load
```

Check if reboot is required.

```
# if [[ $(auditctl -s | grep "enabled") =~ "2" ]]; then printf "Reboot  
required to load rules\n"; fi
```

References:

1. NIST SP 800-53 Rev. 5: AU-3, CM-6
2. <https://netplan.io/faq>

Additional Information:

Potential reboot required

If the auditing configuration is locked (`-e 2`), then `augenrules` will not warn in any way that rules could not be loaded into the running configuration. A system reboot will be required to load the rules into the running configuration.

System call structure

For performance (`man 7 audit.rules`) reasons it is preferable to have all the system calls on one line. However, your configuration may have them on one line each or some other combination. This is important to understand for both the auditing and remediation sections as the examples given are optimized for performance as per the man page.

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	5.5 Implement Automated Configuration Monitoring Systems Utilize a Security Content Automation Protocol (SCAP) compliant configuration monitoring system to verify all security configuration elements, catalog approved exceptions, and alert when unauthorized changes occur.		●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562, T1562.006	TA0003	M1047

6.2.3.6 Ensure use of privileged commands are collected (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

Monitor privileged programs, those that have the **setuid** and/or **setgid** bit set on execution, to determine if unprivileged users are running these commands.

Rationale:

Execution of privileged commands by non-privileged users could be an indication of someone trying to gain unauthorized access to the system.

Impact:

Both the audit and remediation section of this recommendation will traverse all mounted file systems that is not mounted with either **noexec** or **nosuid** mount options. If there are large file systems without these mount options, **such traversal will be significantly detrimental to the performance of the system.**

Before running either the audit or remediation section, inspect the output of the following command to determine exactly which file systems will be traversed:

```
# findmnt -n -l -k -it $(awk '/nodev/ { print $2 }' /proc/filesystems | paste -sd,) | grep -Pv "noexec|nosuid"
```

To exclude a particular file system due to adverse performance impacts, update the audit and remediation sections by adding a sufficiently unique string to the **grep** statement. The above command can be used to test the modified exclusions.

Audit:

On disk configuration

Run the following script to check on disk rules:

```
#!/usr/bin/env bash

{
    for PARTITION in $(findmnt -n -l -k -it $(awk '/nodev/ { print $2 }' /proc/filesystems | paste -sd,) | grep -Pv "noexec|nosuid" | awk '{print $1}'); do
        for PRIVILEGED in $(find "${PARTITION}" -xdev -perm /6000 -type f); do
            grep -qr "${PRIVILEGED}" /etc/audit/rules.d && printf "OK:
'${PRIVILEGED}' found in auditing rules.\n" || printf "Warning:
'${PRIVILEGED}' not found in on disk configuration.\n"
        done
    done
}
```

Verify that all output is **OK**.

Running configuration

Run the following script to check loaded rules:

```
#!/usr/bin/env bash

{
    RUNNING=$(auditctl -l)
    [ -n "${RUNNING}" ] && for PARTITION in $(findmnt -n -l -k -it $(awk '/nodev/ { print $2 }' /proc/filesystems | paste -sd,) | grep -Pv "noexec|nosuid" | awk '{print $1}'); do
        for PRIVILEGED in $(find "${PARTITION}" -xdev -perm /6000 -type f); do
            printf -- "${RUNNING}" | grep -q "${PRIVILEGED}" && printf "OK:
'${PRIVILEGED}' found in auditing rules.\n" || printf "Warning:
'${PRIVILEGED}' not found in running configuration.\n"
        done
    done \
    || printf "ERROR: Variable 'RUNNING' is unset.\n"
}
```

Verify that all output is **OK**.

Special mount points

If there are any special mount points that are not visible by default from **findmnt** as per the above audit, said file systems would have to be manually audited.

Remediation:

Edit or create a file in the `/etc/audit/rules.d/` directory, ending in `.rules` extension, with the relevant rules to monitor the use of privileged commands.

Example script:

```
#!/usr/bin/env bash

{
    UID_MIN=$(awk '/^s*UID_MIN/{print $2}' /etc/login.defs)
    AUDIT_RULE_FILE="/etc/audit/rules.d/50-privileged.rules"
    NEW_DATA=()
    for PARTITION in $(findmnt -n -l -k -it $(awk '/nodev/ { print $2 }' /proc/filesystems | paste -sd,) | grep -Pv "noexec|nosuid" | awk '{print $1}'); do
        readarray -t DATA < <(find "${PARTITION}" -xdev -perm /6000 -type f | awk -v UID_MIN=${UID_MIN} '{print "-a always,exit -F path=\"\$1\" -F perm=x -F auid>=\"UID_MIN\" -F auid!=unset -k privileged"}')
        for ENTRY in "${DATA[@]}"; do
            NEW_DATA+=("{$ENTRY}")
        done
    done
    readarray &> /dev/null -t OLD_DATA < "${AUDIT_RULE_FILE}"
    COMBINED_DATA=( "${OLD_DATA[@]}" "${NEW_DATA[@]}" )
    printf '%s\n' "${COMBINED_DATA[@]}" | sort -u > "${AUDIT_RULE_FILE}"
}
```

Merge and load the rules into active configuration:

```
# augenrules --load
```

Check if reboot is required.

```
# if [[ $(auditctl -s | grep "enabled") =~ "2" ]]; then printf "Reboot required to load rules\n"; fi
```

Special mount points

If there are any special mount points that are not visible by default from just scanning `/`, change the `PARTITION` variable to the appropriate partition and re-run the remediation.

References:

1. NIST SP 800-53 Rev. 5: AU-3, AU-3(1)

Additional Information:

Potential reboot required

If the auditing configuration is locked (`-e 2`), then `augenrules` will not warn in any way that rules could not be loaded into the running configuration. A system reboot will be required to load the rules into the running configuration.

System call structure

For performance (`man 7 audit.rules`) reasons it is preferable to have all the system calls on one line. However, your configuration may have them on one line each or some other combination. This is important to understand for both the auditing and remediation sections as the examples given are optimized for performance as per the man page.

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562, T1562.006	TA0002	M1026

6.2.3.7 Ensure unsuccessful file access attempts are collected (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

Monitor for unsuccessful attempts to access files. The following parameters are associated with system calls that control files:

- creation - `creat`
- opening - `open` , `openat`
- truncation - `truncate` , `ftruncate`

An audit log record will only be written if all of the following criteria is met for the user when trying to access a file:

- a non-privileged user (`auid>=UID_MIN`)
- is not a Daemon event (`auid=4294967295/unset/-1`)
- if the system call returned EACCES (permission denied) or EPERM (some other permanent error associated with the specific system call)

Rationale:

Failed attempts to open, create or truncate files could be an indication that an individual or process is trying to gain unauthorized access to the system.

Audit:

On disk configuration

Run the following command to check the on disk rules:

```
# {
  UID_MIN=$(awk '/^s*UID_MIN/{print $2}' /etc/login.defs)
  [ -n "${UID_MIN}" ] && awk "/^ *-a *always,exit/ \
  && / -F *arch=b(32|64)/ \
  && / -F *auid!=unset/ ||/ -F *auid==1/ ||/ -F *auid!=4294967295/ ) \
  && / -F *auid>=${UID_MIN}/ \
  && / -F *exit=-EACCES/ ||/ -F *exit=-EPERM/ ) \
  && / -S/ \
  && /creat/ \
  && /open/ \
  && /truncate/ \
  && / key= *[!-~]* *$/ ||/ -k *[!-~]* *$/" /etc/audit/rules.d/*.rules \
  || printf "ERROR: Variable 'UID_MIN' is unset.\n"
}
```

Verify the output includes:

```
-a always,exit -F arch=b64 -S creat,open,openat,truncate,ftruncate -F exit==EACCES -F auid>=1000 -F auid!=unset -k access
-a always,exit -F arch=b64 -S creat,open,openat,truncate,ftruncate -F exit==EPERM -F auid>=1000 -F auid!=unset -k access
-a always,exit -F arch=b32 -S creat,open,openat,truncate,ftruncate -F exit==EACCES -F auid>=1000 -F auid!=unset -k access
-a always,exit -F arch=b32 -S creat,open,openat,truncate,ftruncate -F exit==EPERM -F auid>=1000 -F auid!=unset -k access
```

Running configuration

Run the following command to check loaded rules:

```
# {
  UID_MIN=$(awk '/^s*UID_MIN/{print $2}' /etc/login.defs)
  [ -n "${UID_MIN}" ] && auditctl -l | awk "/^ *-a *always,exit/ \
  && / -F *arch=b(32|64)/ \
  && / -F *auid!=unset/ ||/ -F *auid==1/ ||/ -F *auid!=4294967295/ ) \
  && / -F *auid>=${UID_MIN}/ \
  && / -F *exit=-EACCES/ ||/ -F *exit=-EPERM/ ) \
  && / -S/ \
  && /creat/ \
  && /open/ \
  && /truncate/ \
  && / key= *[!-~]* *$/ ||/ -k *[!-~]* *$/" \
  || printf "ERROR: Variable 'UID_MIN' is unset.\n"
}
```

Verify the output includes:

```

-a always,exit -F arch=b64 -S open,truncate,ftruncate,creat,openat -F exit==EACCES -F auid>=1000 -F auid!=--1 -F key=access
-a always,exit -F arch=b64 -S open,truncate,ftruncate,creat,openat -F exit==EPERM -F auid>=1000 -F auid!=--1 -F key=access
-a always,exit -F arch=b32 -S open,truncate,ftruncate,creat,openat -F exit==EACCES -F auid>=1000 -F auid!=--1 -F key=access
-a always,exit -F arch=b32 -S open,truncate,ftruncate,creat,openat -F exit==EPERM -F auid>=1000 -F auid!=--1 -F key=access

```

Remediation:

Create audit rules

Edit or create a file in the `/etc/audit/rules.d/` directory, ending in `.rules` extension, with the relevant rules to monitor unsuccessful file access attempts.

Example:

```

# {
UID_MIN=$(awk '/^s*UID_MIN/{print $2}' /etc/login.defs)
[ -n "${UID_MIN}" ] && printf "
-a always,exit -F arch=b64 -S creat,open,openat,truncate,ftruncate -F exit==EACCES -F auid>=${UID_MIN} -F auid!=unset -k access
-a always,exit -F arch=b64 -S creat,open,openat,truncate,ftruncate -F exit==EPERM -F auid>=${UID_MIN} -F auid!=unset -k access
-a always,exit -F arch=b32 -S creat,open,openat,truncate,ftruncate -F exit==EACCES -F auid>=${UID_MIN} -F auid!=unset -k access
-a always,exit -F arch=b32 -S creat,open,openat,truncate,ftruncate -F exit==EPERM -F auid>=${UID_MIN} -F auid!=unset -k access
" >> /etc/audit/rules.d/50-access.rules || printf "ERROR: Variable 'UID_MIN' is unset.\n"
}

```

Load audit rules

Merge and load the rules into active configuration:

```
# augenrules --load
```

Check if reboot is required.

```
# if [[ $(auditctl -s | grep "enabled") =~ "2" ]]; then printf "Reboot required to load rules\n"; fi
```

References:

1. NIST SP 800-53 Rev. 5: AU-3

Additional Information:

Potential reboot required

If the auditing configuration is locked (`-e 2`), then `augenrules` will not warn in any way that rules could not be loaded into the running configuration. A system reboot will be required to load the rules into the running configuration.

System call structure

For performance (`man 7 audit.rules`) reasons it is preferable to have all the system calls on one line. However, your configuration may have them on one line each or some other combination. This is important to understand for both the auditing and remediation sections as the examples given are optimized for performance as per the man page.

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	14.9 Enforce Detail Logging for Access or Changes to Sensitive Data Enforce detailed audit logging for access to sensitive data or changes to sensitive data (utilizing tools such as File Integrity Monitoring or Security Information and Event Monitoring).			●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562, T1562.006	TA0007	M1047

6.2.3.8 Ensure events that modify user/group information are collected (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

Record events affecting the modification of user or group information, including that of passwords and old passwords if in use.

- `/etc/group` - system groups
- `/etc/passwd` - system users
- `/etc/gshadow` - encrypted password for each group
- `/etc/shadow` - system user passwords
- `/etc/security/opasswd` - storage of old passwords if the relevant PAM module is in use
- `/etc/nsswitch.conf` - file configures how the system uses various databases and name resolution mechanisms
- `/etc/pam.conf` - file determines the authentication services to be used, and the order in which the services are used.
- `/etc/pam.d` - directory contains the PAM configuration files for each PAM-aware application.

The parameters in this section will watch the files to see if they have been opened for write or have had attribute changes (e.g. permissions) and tag them with the identifier "identity" in the audit log file.

Rationale:

Unexpected changes to these files could be an indication that the system has been compromised and that an unauthorized user is attempting to hide their activities or compromise additional accounts.

Audit:

On disk configuration

Run the following command to check the on disk rules:

```
# awk '/^ *-w/ \
&&(/etc/group/ \
||/etc/passwd/ \
||/etc/gshadow/ \
||/etc/shadow/ \
||/etc/security/opasswd/ \
||/etc/nsswitch.conf/ \
||/etc/pam.conf/ \
||/etc/pam.d/) \
&&/ +-p *wa/ \
&&(/ key= *[^-~]* *$||/ -k *[^-~]* *$/) /etc/audit/rules.d/*.rules
```

Verify the output matches:

```
-w /etc/group -p wa -k identity
-w /etc/passwd -p wa -k identity
-w /etc/gshadow -p wa -k identity
-w /etc/shadow -p wa -k identity
-w /etc/security/opasswd -p wa -k identity
-w /etc/nsswitch.conf -p wa -k identity
-w /etc/pam.conf -p wa -k identity
-w /etc/pam.d -p wa -k identity
```

Running configuration

Run the following command to check loaded rules:

```
# auditctl -l | awk '/^ *-w/ \
&&(/etc/group/ \
||/etc/passwd/ \
||/etc/gshadow/ \
||/etc/shadow/ \
||/etc/security/opasswd/ \
||/etc/nsswitch.conf/ \
||/etc/pam.conf/ \
||/etc/pam.d/) \
&&/ +-p *wa/ \
&&(/ key= *[^-~]* *$||/ -k *[^-~]* *$/) '
```

Verify the output matches:

```
-w /etc/group -p wa -k identity  
-w /etc/passwd -p wa -k identity  
-w /etc/gshadow -p wa -k identity  
-w /etc/shadow -p wa -k identity  
-w /etc/security/opasswd -p wa -k identity  
-w /etc/nsswitch.conf -p wa -k identity  
-w /etc/pam.conf -p wa -k identity  
-w /etc/pam.d -p wa -k identity
```

Remediation:

Edit or create a file in the [`/etc/audit/rules.d/`](#) directory, ending in [`.rules`](#) extension, with the relevant rules to monitor events that modify user/group information.

Example:

```
# printf "  
-w /etc/group -p wa -k identity  
-w /etc/passwd -p wa -k identity  
-w /etc/gshadow -p wa -k identity  
-w /etc/shadow -p wa -k identity  
-w /etc/security/opasswd -p wa -k identity  
-w /etc/nsswitch.conf -p wa -k identity  
-w /etc/pam.conf -p wa -k identity  
-w /etc/pam.d -p wa -k identity  
" >> /etc/audit/rules.d/50-identity.rules
```

Merge and load the rules into active configuration:

```
# augenrules --load
```

Check if reboot is required.

```
# if [[ $(auditctl -s | grep "enabled") =~ "2" ]]; then printf "Reboot  
required to load rules\n"; fi
```

References:

1. NIST SP 800-53 Rev. 5: AU-3
2. <https://manpages.debian.org/bookworm/manpages/nsswitch.conf.5.en.html>
3. https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/6/html/managing_smart_cards/pam_configuration_files

Additional Information:

Potential reboot required

If the auditing configuration is locked (`-e 2`), then `augenrules` will not warn in any way that rules could not be loaded into the running configuration. A system reboot will be required to load the rules into the running configuration.

System call structure

For performance (`man 7 audit.rules`) reasons it is preferable to have all the system calls on one line. However, your configuration may have them on one line each or some other combination. This is important to understand for both the auditing and remediation sections as the examples given are optimized for performance as per the man page.

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	4.8 Log and Alert on Changes to Administrative Group Membership Configure systems to issue a log entry and alert when an account is added to or removed from any group assigned administrative privileges.		●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562, T1562.006	TA0004	M1047

6.2.3.9 Ensure discretionary access control permission modification events are collected (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

Monitor changes to file permissions, attributes, ownership and group. The parameters in this section track changes for system calls that affect file permissions and attributes.

The following commands and system calls effect the permissions, ownership and various attributes of files.

- `chmod`
- `fchmod`
- `fchmodat`
- `chown`
- `fchown`
- `fchownat`
- `lchown`
- `setxattr`
- `lsetxattr`
- `fsetxattr`
- `removexattr`
- `lremovexattr`
- `fremovexattr`

In all cases, an audit record will only be written for non-system user ids and will ignore Daemon events. All audit records will be tagged with the identifier "perm_mod."

Rationale:

Monitoring for changes in file attributes could alert a system administrator to activity that could indicate intruder activity or policy violation.

Audit:

Note: Output showing all audited syscalls, e.g. (-a always,exit -F arch=b64 -S chmod,fchmod,fchmodat,chmod,fchmod,fchmodat,setattr,lsetattr,fsetattr,removexattr,!removexattr,fremovexattr -F auid>=1000 -F auid!=unset -F key=perm_mod) is also acceptable. These have been separated by function on the displayed output for clarity.

On disk configuration

Run the following command to check the on disk rules:

```
# {
  UID_MIN=$(awk '/^s*UID_MIN/{print $2}' /etc/login.defs)
  [ -n "${UID_MIN}" ] && awk "/^ *-a *always,exit / \
    && / -F *arch=b(32|64) / \
    && (/ -F *auid!=unset/||/ -F *auid!=-1/||/ -F *auid!=4294967295/) \
    && / -S / \
    && / -F *auid>=${UID_MIN}/ \
    && (/chmod/||/fchmod/||/fchmodat/ \
      ||/chown/||/fchown/||/fchownat/||/lchown/ \
      ||/setattr/||/lsetattr/||/f setattr/ \
      ||/removexattr/||/lremovexattr/||/fremovexattr/) \
    && (/ key= *[!~-]* *$/||/ -k *[!~-]* *$/)" /etc/audit/rules.d/*.rules \
  || printf "ERROR: Variable 'UID_MIN' is unset.\n"
}
```

Verify the output matches:

```
-a always,exit -F arch=b64 -S chmod,fchmod,fchmodat -F auid>=1000 -F
auid!=unset -F key=perm_mod
-a always,exit -F arch=b64 -S chown,fchown,lchown,fchownat -F auid>=1000 -F
auid!=unset -F key=perm_mod
-a always,exit -F arch=b32 -S chmod,fchmod,fchmodat -F auid>=1000 -F
auid!=unset -F key=perm_mod
-a always,exit -F arch=b32 -S lchown,fchown,chown,fchownat -F auid>=1000 -F
auid!=unset -F key=perm_mod
-a always,exit -F arch=b64 -S
setattr,lsetattr,f setattr,removexattr,lremovexattr,fremovexattr -F
auid>=1000 -F auid!=unset -F key=perm_mod
-a always,exit -F arch=b32 -S
setattr,lsetattr,f setattr,removexattr,lremovexattr,fremovexattr -F
auid>=1000 -F auid!=unset -F key=perm_mod
```

Running configuration

Run the following command to check loaded rules:

```

# {
  UID_MIN=$(awk '/^s*UID_MIN/{print $2}' /etc/login.defs)
  [ -n "${UID_MIN}" ] && auditctl -l | awk "/^ *-a *always,exit/ \
  && /-F arch=b(32|64) / \
  && (/ -F auid!=unset/ ||/ -F auid!=-1/ ||/ -F auid!=4294967295/) \
  && -S/ \
  && -F auid>=${UID_MIN}/ \
  && (/chmod/ ||/fchmod/ ||/fchmodat/ \
    ||/chown/ ||/fchown/ ||/fchownat/ ||/lchown/ \
    ||/setxattr/ ||/lsetxattr/ ||/fsetxattr/ \
    ||/removexattr/ ||/lremovexattr/ ||/fremovexattr/) \
  && (/ key= *[!~]* *$/ ||/ -k *[!~]* *$/) " \
  || printf "ERROR: Variable 'UID_MIN' is unset.\n"
}

```

Verify the output matches:

```

-a always,exit -F arch=b64 -S chmod,fchmod,fchmodat -F auid>=1000 -F auid!=-1
-F key=perm_mod
-a always,exit -F arch=b64 -S chown,fchown,lchown,fchownat -F auid>=1000 -F
auid!=-1 -F key=perm_mod
-a always,exit -F arch=b32 -S chmod,fchmod,fchmodat -F auid>=1000 -F auid!=-1
-F key=perm_mod
-a always,exit -F arch=b32 -S lchown,fchown,chown,fchownat -F auid>=1000 -F
auid!=-1 -F key=perm_mod
-a always,exit -F arch=b64 -S
setxattr,lsetxattr,fsetxattr,removexattr,lremovexattr,fremovexattr -F
auid>=1000 -F auid!=-1 -F key=perm_mod
-a always,exit -F arch=b32 -S
setxattr,lsetxattr,fsetxattr,removexattr,lremovexattr,fremovexattr -F
auid>=1000 -F auid!=-1 -F key=perm_mod

```

Remediation:

Create audit rules

Edit or create a file in the `/etc/audit/rules.d/` directory, ending in `.rules` extension, with the relevant rules to monitor discretionary access control permission modification events.

Example:

```
# {
UID_MIN=$(awk '/^s*UID_MIN/{print $2}' /etc/login.defs)
[ -n "${UID_MIN}" ] && printf "
-a always,exit -F arch=b64 -S chmod,fchmod,fchmodat -F auid>=${UID_MIN} -F
auid!=unset -F key=perm_mod
-a always,exit -F arch=b64 -S chown,fchown,lchown,fchownat -F
auid>=${UID_MIN} -F auid!=unset -F key=perm_mod
-a always,exit -F arch=b32 -S chmod,fchmod,fchmodat -F auid>=${UID_MIN} -F
auid!=unset -F key=perm_mod
-a always,exit -F arch=b32 -S lchown,fchown,chown,fchownat -F
auid>=${UID_MIN} -F auid!=unset -F key=perm_mod
-a always,exit -F arch=b64 -S
setxattr,lsetxattr,fsetxattr,removexattr,lremovexattr,fremovexattr -F
auid>=${UID_MIN} -F auid!=unset -F key=perm_mod
-a always,exit -F arch=b32 -S
setxattr,lsetxattr,fsetxattr,removexattr,lremovexattr,fremovexattr -F
auid>=${UID_MIN} -F auid!=unset -F key=perm_mod
" >> /etc/audit/rules.d/50-perm_mod.rules || printf "ERROR: Variable
'UID_MIN' is unset.\n"
}
```

Load audit rules

Merge and load the rules into active configuration:

```
# augenrules --load
```

Check if reboot is required.

```
# if [[ $(auditctl -s | grep "enabled") =~ "2" ]]; then printf "Reboot
required to load rules\n"; fi
```

References:

1. NIST SP 800-53 Rev. 5: AU-3, CM-6

Additional Information:

Potential reboot required

If the auditing configuration is locked (`-e 2`), then `augenrules` will not warn in any way that rules could not be loaded into the running configuration. A system reboot will be required to load the rules into the running configuration.

System call structure

For performance (`man 7 audit.rules`) reasons it is preferable to have all the system calls on one line. However, your configuration may have them on one line each or some other combination. This is important to understand for both the auditing and remediation sections as the examples given are optimized for performance as per the man page.

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	5.5 Implement Automated Configuration Monitoring Systems Utilize a Security Content Automation Protocol (SCAP) compliant configuration monitoring system to verify all security configuration elements, catalog approved exceptions, and alert when unauthorized changes occur.		●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562, T1562.006	TA0005	M1022

6.2.3.10 Ensure successful file system mounts are collected (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

Monitor the use of the `mount` system call. The `mount` (and `umount`) system call controls the mounting and unmounting of file systems. The parameters below configure the system to create an audit record when the `mount` system call is used by a non-privileged user

Rationale:

It is highly unusual for a non privileged user to `mount` file systems to the system. While tracking `mount` commands gives the system administrator evidence that external media may have been mounted (based on a review of the source of the mount and confirming it's an external media type), it does not conclusively indicate that data was exported to the media. System administrators who wish to determine if data were exported, would also have to track successful `open`, `creat` and `truncate` system calls requiring write access to a file under the mount point of the external media file system. This could give a fair indication that a write occurred. The only way to truly prove it, would be to track successful writes to the external media. Tracking write system calls could quickly fill up the audit log and is not recommended. Recommendations on configuration options to track data export to media is beyond the scope of this document.

Audit:

On disk configuration

Run the following command to check the on disk rules:

```
# {
  UID_MIN=$(awk '/^s*UID_MIN/{print $2}' /etc/login.defs)
  [ -n "${UID_MIN}" ] && awk "/^ *-a *always,exit/ \
  && / -F *arch=b(32|64)/ \
  && / -F *auid!=unset/ || / -F *auid==1/ || / -F *auid!=4294967295/ ) \
  && / -F *auid>=${UID_MIN}/ \
  && / -S/ \
  &&/mount/ \
  &&(/ key= *[!~]* *$/|| / -k *[!~]* *$/)" /etc/audit/rules.d/*.rules \
  || printf "ERROR: Variable 'UID_MIN' is unset.\n"
}
```

Verify the output matches:

```
-a always,exit -F arch=b64 -S mount -F auid>=1000 -F auid!=unset -k mounts
-a always,exit -F arch=b32 -S mount -F auid>=1000 -F auid!=unset -k mounts
```

Running configuration

Run the following command to check loaded rules:

```
# {
  UID_MIN=$(awk '/^s*UID_MIN/{print $2}' /etc/login.defs)
  [ -n "${UID_MIN}" ] && auditctl -l | awk "/^ *-a *always,exit/ \
  && / -F *arch=b(32|64)/ \
  && / -F *auid!=unset/ || / -F *auid==1/ || / -F *auid!=4294967295/ ) \
  && / -F *auid>=${UID_MIN}/ \
  && / -S/ \
  &&/mount/ \
  &&(/ key= *[!~]* *$/|| / -k *[!~]* *$/)" \
  || printf "ERROR: Variable 'UID_MIN' is unset.\n"
}
```

Verify the output matches:

```
-a always,exit -F arch=b64 -S mount -F auid>=1000 -F auid==1 -F key=mounts
-a always,exit -F arch=b32 -S mount -F auid>=1000 -F auid==1 -F key=mounts
```

Remediation:

Create audit rules

Edit or create a file in the `/etc/audit/rules.d/` directory, ending in `.rules` extension, with the relevant rules to monitor successful file system mounts.

Example:

```
# {
UID_MIN=$(awk '/^s*UID_MIN/{print $2}' /etc/login.defs)
[ -n "${UID_MIN}" ] && printf "
-a always,exit -F arch=b32 -S mount -F auid>=$UID_MIN -F auid!=unset -k
mounts
-a always,exit -F arch=b64 -S mount -F auid>=$UID_MIN -F auid!=unset -k
mounts
" >> /etc/audit/rules.d/50-mounts.rules || printf "ERROR: Variable 'UID_MIN'
is unset.\n"
}
```

Load audit rules

Merge and load the rules into active configuration:

```
# augenrules --load
```

Check if reboot is required.

```
# if [[ $(auditctl -s | grep "enabled") =~ "2" ]]; then printf "Reboot
required to load rules\n"; fi
```

References:

1. NIST SP 800-53 Rev. 5: CM-6

Additional Information:

Potential reboot required

If the auditing configuration is locked (`-e 2`), then `augenrules` will not warn in any way that rules could not be loaded into the running configuration. A system reboot will be required to load the rules into the running configuration.

System call structure

For performance (`man 7 audit.rules`) reasons it is preferable to have all the system calls on one line. However, your configuration may have them on one line each or some other combination. This is important to understand for both the auditing and remediation sections as the examples given are optimized for performance as per the man page.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<p>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.</p>		●	●
v7	<p>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.</p>		●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562, T1562.006	TA0010	M1034

6.2.3.11 Ensure session initiation information is collected (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

Monitor session initiation events. The parameters in this section track changes to the files associated with session events.

- `/var/run/utmp` - tracks all currently logged in users.
- `/var/log/wtmp` - file tracks logins, logouts, shutdown, and reboot events.
- `/var/log/btmp` - keeps track of failed login attempts and can be read by entering the command `/usr/bin/last -f /var/log/btmp`.

All audit records will be tagged with the identifier "session."

Rationale:

Monitoring these files for changes could alert a system administrator to logins occurring at unusual hours, which could indicate intruder activity (i.e. a user logging in at a time when they do not normally log in).

Audit:

On disk configuration

Run the following command to check the on disk rules:

```
# awk '/^ *-w/ \
&&(/\/var\/run\/utmp/ \
||/\/var\/log\/wtmp/ \
||/\/var\/log\/btmp/) \
&&/ +-p *wa/ \
&&(/ key= *[!-~]* *$/ ||/ -k *[!-~]* *$/)' /etc/audit/rules.d/*.rules
```

Verify the output matches:

```
-w /var/run/utmp -p wa -k session
-w /var/log/wtmp -p wa -k session
-w /var/log/btmp -p wa -k session
```

Running configuration

Run the following command to check loaded rules:

```
# auditctl -l | awk '/^ *-w/ \
&&(/\/var\/run\/utmp/ \
||/\/var\/log\/wtmp/ \
||/\/var\/log\/btmp/) \
&&/ +-p *wa/ \
&&(/ key= *[!-~]* *$/ ||/ -k *[!-~]* *$/)'
```

Verify the output matches:

```
-w /var/run/utmp -p wa -k session
-w /var/log/wtmp -p wa -k session
-w /var/log/btmp -p wa -k session
```

Remediation:

Edit or create a file in the `/etc/audit/rules.d/` directory, ending in `.rules` extension, with the relevant rules to monitor session initiation information.

Example:

```
# printf "
-w /var/run/utmp -p wa -k session
-w /var/log/wtmp -p wa -k session
-w /var/log/btmp -p wa -k session
" >> /etc/audit/rules.d/50-session.rules
```

Merge and load the rules into active configuration:

```
# augenrules --load
```

Check if reboot is required.

```
# if [[ $(auditctl -s | grep "enabled") =~ "2" ]]; then printf "Reboot
required to load rules\n"; fi
```

References:

1. NIST SP 800-53 Rev. 5: AU-3

Additional Information:

Potential reboot required

If the auditing configuration is locked (`-e 2`), then `augenrules` will not warn in any way that rules could not be loaded into the running configuration. A system reboot will be required to load the rules into the running configuration.

System call structure

For performance (`man 7 audit.rules`) reasons it is preferable to have all the system calls on one line. However, your configuration may have them on one line each or some other combination. This is important to understand for both the auditing and remediation sections as the examples given are optimized for performance as per the man page.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<p>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.</p>		●	●
v7	<p>4.9 Log and Alert on Unsuccessful Administrative Account Login Configure systems to issue a log entry and alert on unsuccessful logins to an administrative account.</p>		●	●
v7	<p>16.13 Alert on Account Login Behavior Deviation Alert when users deviate from normal login behavior, such as time-of-day, workstation location and duration.</p>			●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562, T1562.006	TA0001	M1047

6.2.3.12 Ensure login and logout events are collected (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

Monitor login and logout events. The parameters below track changes to files associated with login/logout events.

- `/var/log/lastlog` - maintain records of the last time a user successfully logged in.
- `/var/run/faillock` - directory maintains records of login failures via the `pam_faillock` module.

Rationale:

Monitoring login/logout events could provide a system administrator with information associated with brute force attacks against user logins.

Audit:

On disk configuration

Run the following command to check the on disk rules:

```
# awk '/^ *-w/ \
&&(/\/var\/log\/lastlog/ \
||/\/var\/run\/faillock/) \
&&/ +-p *wa/ \
&&(/ key= * [!-~]* *$/ ||/ -k * [!-~]* *$/) ' /etc/audit/rules.d/*.rules
```

Verify the output matches:

```
-w /var/log/lastlog -p wa -k logins
-w /var/run/faillock -p wa -k logins
```

Running configuration

Run the following command to check loaded rules:

```
# auditctl -l | awk '/^ *-w/ \
&&(/\/var\/log\/lastlog/ \
||/\/var\/run\/faillock/) \
&&/ +-p *wa/ \
&&(/ key= * [!-~]* *$/ ||/ -k * [!-~]* *$/) '
```

Verify the output matches:

```
-w /var/log/lastlog -p wa -k logins  
-w /var/run/faillock -p wa -k logins
```

Remediation:

Edit or create a file in the `/etc/audit/rules.d/` directory, ending in `.rules` extension, with the relevant rules to monitor login and logout events.

Example:

```
# printf "  
-w /var/log/lastlog -p wa -k logins  
-w /var/run/faillock -p wa -k logins  
" >> /etc/audit/rules.d/50-login.rules
```

Merge and load the rules into active configuration:

```
# augenrules --load
```

Check if reboot is required.

```
# if [[ $(auditctl -s | grep "enabled") =~ "2" ]]; then printf "Reboot  
required to load rules\n"; fi
```

References:

1. NIST SP 800-53 Rev. 5: AU-3

Additional Information:

Potential reboot required

If the auditing configuration is locked (`-e 2`), then `augenrules` will not warn in any way that rules could not be loaded into the running configuration. A system reboot will be required to load the rules into the running configuration.

System call structure

For performance (`man 7 audit.rules`) reasons it is preferable to have all the system calls on one line. However, your configuration may have them on one line each or some other combination. This is important to understand for both the auditing and remediation sections as the examples given are optimized for performance as per the man page.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<p>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.</p>		●	●
v7	<p>4.9 Log and Alert on Unsuccessful Administrative Account Login Configure systems to issue a log entry and alert on unsuccessful logins to an administrative account.</p>		●	●
v7	<p>16.11 Lock Workstation Sessions After Inactivity Automatically lock workstation sessions after a standard period of inactivity.</p>	●	●	●
v7	<p>16.13 Alert on Account Login Behavior Deviation Alert when users deviate from normal login behavior, such as time-of-day, workstation location and duration.</p>			●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562, T1562.006	TA0001	M1047

6.2.3.13 Ensure file deletion events by users are collected (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

Monitor the use of system calls associated with the deletion or renaming of files and file attributes. This configuration statement sets up monitoring for:

- `unlink` - remove a file
- `unlinkat` - remove a file attribute
- `rename` - rename a file
- `renameat` rename a file attribute system calls and tags them with the identifier "delete".

Rationale:

Monitoring these calls from non-privileged users could provide a system administrator with evidence that inappropriate removal of files and file attributes associated with protected files is occurring. While this audit option will look at all events, system administrators will want to look for specific privileged files that are being deleted or altered.

Audit:

On disk configuration

Run the following command to check the on disk rules:

```
# {
  UID_MIN=$(awk '/^s*UID_MIN/{print $2}' /etc/login.defs)
  [ -n "${UID_MIN}" ] && awk "/^ *-a *always,exit/ \
  && / -F *arch=b(32|64)/ \
  && / -F *auid!=unset/ || / -F *auid!=-1/ || / -F *auid!=4294967295/ ) \
  && / -F *auid>=${UID_MIN}/ \
  && / -S/ \
  && (/unlink/ || /rename/ || /unlinkat/ || /renameat/) \
  && (/ key= *[!-~]* *$/ || / -k *[!-~]* *$/) /etc/audit/rules.d/*.rules \
  || printf "ERROR: Variable 'UID_MIN' is unset.\n"
}
```

Verify the output matches:

```
-a always,exit -F arch=b64 -S unlink,unlinkat,rename,renameat -F auid>=1000 -
F auid!=unset -k delete
-a always,exit -F arch=b32 -S unlink,unlinkat,rename,renameat -F auid>=1000 -
F auid!=unset -k delete
```

Running configuration

Run the following command to check loaded rules:

```
# {
  UID_MIN=$(awk '/^s*UID_MIN/{print $2}' /etc/login.defs)
  [ -n "${UID_MIN}" ] && auditctl -l | awk "/^ *-a *always,exit/ \
  && / -F *arch=b(32|64)/ \
  && / -F *auid!=unset/ || / -F *auid!=-1/ || / -F *auid!=4294967295/ ) \
  && / -F *auid>=${UID_MIN}/ \
  && / -S/ \
  && (/unlink/ || /rename/ || /unlinkat/ || /renameat/) \
  && (/ key= *[!-~]* *$/ || / -k *[!-~]* *$/) \
  || printf "ERROR: Variable 'UID_MIN' is unset.\n"
}
```

Verify the output matches:

```
-a always,exit -F arch=b64 -S rename,unlink,unlinkat,renameat -F auid>=1000 -
F auid!=-1 -F key=delete
-a always,exit -F arch=b32 -S unlink,rename,unlinkat,renameat -F auid>=1000 -
F auid!=-1 -F key=delete
```

Remediation:

Create audit rules

Edit or create a file in the `/etc/audit/rules.d/` directory, ending in `.rules` extension, with the relevant rules to monitor file deletion events by users.

Example:

```
# {
UID_MIN=$(awk '/^s*UID_MIN/{print $2}' /etc/login.defs)
[ -n "${UID_MIN}" ] && printf "
-a always,exit -F arch=b64 -S rename,unlink,unlinkat,renameat -F
auid>=${UID_MIN} -F auid!=unset -F key=delete
-a always,exit -F arch=b32 -S rename,unlink,unlinkat,renameat -F
auid>=${UID_MIN} -F auid!=unset -F key=delete
" >> /etc/audit/rules.d/50-delete.rules || printf "ERROR: Variable 'UID_MIN'
is unset.\n"
}
```

Load audit rules

Merge and load the rules into active configuration:

```
# augenrules --load
```

Check if reboot is required.

```
# if [[ $(auditctl -s | grep "enabled") =~ "2" ]]; then printf "Reboot
required to load rules\n"; fi
```

References:

1. NIST SP 800-53 Rev. 5: AU-12, SC-7

Additional Information:

Potential reboot required

If the auditing configuration is locked (`-e 2`), then `augenrules` will not warn in any way that rules could not be loaded into the running configuration. A system reboot will be required to load the rules into the running configuration.

System call structure

For performance (`man 7 audit.rules`) reasons it is preferable to have all the system calls on one line. However, your configuration may have them on one line each or some other combination. This is important to understand for both the auditing and remediation sections as the examples given are optimized for performance as per the man page.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<p>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.</p>		●	●
v7	<p>6.2 Activate audit logging Ensure that local logging has been enabled on all systems and networking devices.</p>	●	●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562, T1562.006	TA0005	M1047

6.2.3.14 Ensure events that modify the system's Mandatory Access Controls are collected (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

Monitor AppArmor, an implementation of mandatory access controls. The parameters below monitor any write access (potential additional, deletion or modification of files in the directory) or attribute changes to the `/etc/apparmor/` and `/etc/apparmor.d/` directories.

Note: If a different Mandatory Access Control method is used, changes to the corresponding directories should be audited.

Rationale:

Changes to files in the `/etc/apparmor/` and `/etc/apparmor.d/` directories could indicate that an unauthorized user is attempting to modify access controls and change security contexts, leading to a compromise of the system.

Audit:

On disk configuration

Run the following command to check the on disk rules:

```
# awk '/^ *-w/ \
&&(/etc/apparmor/ \
||/etc/apparmor.d/) \
&& / +p *wa/ \
&&(/ key= *[!~]* *$/ ||/ -k *[!~]* *$/) ' /etc/audit/rules.d/*.rules
```

Verify the output matches:

```
-w /etc/apparmor/ -p wa -k MAC-policy
-w /etc/apparmor.d/ -p wa -k MAC-policy
```

Running configuration

Run the following command to check loaded rules:

```
# auditctl -l | awk '/^ *-w/ \
&&(/etc/apparmor/ \
||/etc/apparmor.d/) \
&& / +p *wa/ \
&&(/ key= *[!~]* *$/ ||/ -k *[!~]* *$/) '
```

Verify the output matches:

```
-w /etc/apparmor/ -p wa -k MAC-policy  
-w /etc/apparmor.d/ -p wa -k MAC-policy
```

Remediation:

Edit or create a file in the `/etc/audit/rules.d/` directory, ending in `.rules` extension, with the relevant rules to monitor events that modify the system's Mandatory Access Controls.

Example:

```
# printf "  
-w /etc/apparmor/ -p wa -k MAC-policy  
-w /etc/apparmor.d/ -p wa -k MAC-policy  
" >> /etc/audit/rules.d/50-MAC-policy.rules
```

Merge and load the rules into active configuration:

```
# augenrules --load
```

Check if reboot is required.

```
# if [[ $(auditctl -s | grep "enabled") =~ "2" ]]; then printf "Reboot  
required to load rules\n"; fi
```

References:

1. NIST SP 800-53 Rev. 5: AU-3, CM-6

Additional Information:

Potential reboot required

If the auditing configuration is locked (`-e 2`), then `augenrules` will not warn in any way that rules could not be loaded into the running configuration. A system reboot will be required to load the rules into the running configuration.

System call structure

For performance (`man 7 audit.rules`) reasons it is preferable to have all the system calls on one line. However, your configuration may have them on one line each or some other combination. This is important to understand for both the auditing and remediation sections as the examples given are optimized for performance as per the man page.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<p>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.</p>	●	●	
v7	<p>5.5 Implement Automated Configuration Monitoring Systems Utilize a Security Content Automation Protocol (SCAP) compliant configuration monitoring system to verify all security configuration elements, catalog approved exceptions, and alert when unauthorized changes occur.</p>	●	●	

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562, T1562.006	TA0004	M1022

6.2.3.15 Ensure successful and unsuccessful attempts to use the chcon command are collected (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

The operating system must generate audit records for successful/unsuccessful uses of the **chcon** command.

Rationale:

The **chcon** command is used to change file security context. Without generating audit records that are specific to the security and mission needs of the organization, it would be difficult to establish, correlate, and investigate the events relating to an incident or identify those responsible for one.

Audit records can be generated from various components within the information system (e.g., module or policy filter).

Audit:

On disk configuration

Run the following command to check the on disk rules:

```
# {
  UID_MIN=$(awk '/^s*UID_MIN/{print $2}' /etc/login.defs)
  [ -n "${UID_MIN}" ] && awk "/^ *-a *always,exit/ \
    && (/ -F *auid!=unset/||/ -F *auid==1/||/ -F *auid!=4294967295/) \
    &&/ -F *auid>=${UID_MIN}/ \
    &&/ -F *perm=x/ \
    &&/ -F *path=/usr/bin/chcon/ \
    &&(/ key= *[!~]* *$/|/ -k *[!~]* *$/) " /etc/audit/rules.d/*.rules \
  || printf "ERROR: Variable 'UID_MIN' is unset.\n"
}
```

Verify the output matches:

```
-a always,exit -F path=/usr/bin/chcon -F perm=x -F auid>=1000 -F auid!=unset
-k perm_chng
```

Running configuration

Run the following command to check loaded rules:

```
# {
  UID_MIN=$(awk '/^s*UID_MIN/{print $2}' /etc/login.defs)
  [ -n "${UID_MIN}" ] && auditctl -l | awk "/^ *-a *always,exit/ \
    && (/ -F *auid!=unset/||/ -F *auid==1/||/ -F *auid!=4294967295/) \
    &&/ -F *auid>=${UID_MIN}/ \
    &&/ -F *perm=x/ \
    &&/ -F *path=/usr/bin/chcon/ \
    &&(/ key= *[!~]* *$/|/ -k *[!~]* *$/) " \
  || printf "ERROR: Variable 'UID_MIN' is unset.\n"
}
```

Verify the output matches:

```
-a always,exit -S all -F path=/usr/bin/chcon -F perm=x -F auid>=1000 -F
auid==1 -F key=perm_chng
```

Remediation:

Create audit rules

Edit or create a file in the `/etc/audit/rules.d/` directory, ending in `.rules` extension, with the relevant rules to monitor successful and unsuccessful attempts to use the `chcon` command.

Example:

```
# {
  UID_MIN=$(awk '/^s*UID_MIN/{print $2}' /etc/login.defs)
  [ -n "${UID_MIN}" ] && printf "
-a always,exit -F path=/usr/bin/chcon -F perm=x -F auid>=${UID_MIN} -F
auid!=unset -k perm_chng
" >> /etc/audit/rules.d/50-perm_chng.rules || printf "ERROR: Variable
'UID_MIN' is unset.\n"
}
```

Load audit rules

Merge and load the rules into active configuration:

```
# augenrules --load
```

Check if reboot is required.

```
# if [[ $(auditctl -s | grep "enabled") =~ "2" ]]; then printf "Reboot
required to load rules\n"; fi
```

References:

1. NIST SP 800-53 Rev. 5: AU-2, AU-12, SI-5

Additional Information:

Potential reboot required

If the auditing configuration is locked (`-e 2`), then `augenrules` will not warn in any way that rules could not be loaded into the running configuration. A system reboot will be required to load the rules into the running configuration.

System call structure

For performance (`man 7 audit.rules`) reasons it is preferable to have all the system calls on one line. However, your configuration may have them on one line each or some other combination. This is important to understand for both the auditing and remediation sections as the examples given are optimized for performance as per the man page.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<p>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.</p>	●	●	●
v7	<p>6.2 Activate audit logging Ensure that local logging has been enabled on all systems and networking devices.</p>	●	●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562, T1562.006	TA0005	M1022

6.2.3.16 Ensure successful and unsuccessful attempts to use the setfacl command are collected (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

The operating system must generate audit records for successful/unsuccessful uses of the **setfacl** command

Rationale:

This utility sets Access Control Lists (ACLs) of files and directories. Without generating audit records that are specific to the security and mission needs of the organization, it would be difficult to establish, correlate, and investigate the events relating to an incident or identify those responsible for one.

Audit records can be generated from various components within the information system (e.g., module or policy filter).

Audit:

On disk configuration

Run the following command to check the on disk rules:

```
# {
  UID_MIN=$(awk '/^s*UID_MIN/{print $2}' /etc/login.defs)
  [ -n "${UID_MIN}" ] && awk "/^ *-a *always,exit/ \
    &&(/ -F *auid!=unset/||/ -F *auid!=-1/||/ -F *auid!=4294967295/) \
    &&/ -F *auid>=${UID_MIN}/ \
    &&/ -F *perm=x/ \
    &&/ -F *path=/usr/bin/setfacl/ \
    &&(/ key= *[!~]* *$/|/ -k *[!~]* *$/) /etc/audit/rules.d/*.rules ||
  printf "ERROR: Variable 'UID_MIN' is unset.\n"
}
```

Verify the output matches:

```
-a always,exit -F path=/usr/bin/setfacl -F perm=x -F auid>=1000 -F
auid!=unset -k perm_chng
```

Running configuration

Run the following command to check loaded rules:

```
# {
  UID_MIN=$(awk '/^s*UID_MIN/{print $2}' /etc/login.defs)
  [ -n "${UID_MIN}" ] && auditctl -l | awk "/^ *-a *always,exit/ \
    &&(/ -F *auid!=unset/||/ -F *auid!=-1/||/ -F *auid!=4294967295/) \
    &&/ -F *auid>=${UID_MIN}/ \
    &&/ -F *perm=x/ \
    &&/ -F *path=/usr/bin/setfacl/ \
    &&(/ key= *[!~]* *$/|/ -k *[!~]* *$/) \
  || printf "ERROR: Variable 'UID_MIN' is unset.\n"
}
```

Verify the output matches:

```
-a always,exit -S all -F path=/usr/bin/setfacl -F perm=x -F auid>=1000 -F
auid!=-1 -F key=perm_chng
```

Remediation:

Create audit rules

Edit or create a file in the `/etc/audit/rules.d/` directory, ending in `.rules` extension, with the relevant rules to monitor successful and unsuccessful attempts to use the `setfacl` command.

Example:

```
# {
  UID_MIN=$(awk '/^s*UID_MIN/{print $2}' /etc/login.defs)
  [ -n "${UID_MIN}" ] && printf "
-a always,exit -F path=/usr/bin/setfacl -F perm=x -F auid>=${UID_MIN} -F
auid!=unset -k perm_chng
" >> /etc/audit/rules.d/50-perm_chng.rules || printf "ERROR: Variable
'UID_MIN' is unset.\n"
}
```

Load audit rules

Merge and load the rules into active configuration:

```
# augenrules --load
```

Check if reboot is required.

```
# if [[ $(auditctl -s | grep "enabled") =~ "2" ]]; then printf "Reboot
required to load rules\n"; fi
```

References:

1. NIST SP 800-53 Rev. 5: AU-2, AU-12, SI-5

Additional Information:

Potential reboot required

If the auditing configuration is locked (`-e 2`), then `augenrules` will not warn in any way that rules could not be loaded into the running configuration. A system reboot will be required to load the rules into the running configuration.

System call structure

For performance (`man 7 audit.rules`) reasons it is preferable to have all the system calls on one line. However, your configuration may have them on one line each or some other combination. This is important to understand for both the auditing and remediation sections as the examples given are optimized for performance as per the man page.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<p>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.</p>	●	●	●
v7	<p>6.2 Activate audit logging Ensure that local logging has been enabled on all systems and networking devices.</p>	●	●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562, T1562.006	TA0005	M1022

6.2.3.17 Ensure successful and unsuccessful attempts to use the `chacl` command are collected (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

The operating system must generate audit records for successful/unsuccessful uses of the `chacl` command.

`chacl` is an IRIX-compatibility command, and is maintained for those users who are familiar with its use from either XFS or IRIX.

Rationale:

`chacl` changes the ACL(s) for a file or directory. Without generating audit records that are specific to the security and mission needs of the organization, it would be difficult to establish, correlate, and investigate the events relating to an incident or identify those responsible for one.

Audit records can be generated from various components within the information system (e.g., module or policy filter).

Audit:

On disk configuration

Run the following command to check the on disk rules:

```
# {
  UID_MIN=$(awk '/^s*UID_MIN/{print $2}' /etc/login.defs)
  [ -n "${UID_MIN}" ] && awk "/^ *-a *always,exit/ \
    && (/ -F *auid!=unset/||/ -F *auid==1/||/ -F *auid!=4294967295/) \
    &&/ -F *auid>=${UID_MIN}/ \
    &&/ -F *perm=x/ \
    &&/ -F *path=/usr/bin/chacl/ \
    &&(/ key= *[!-~]* *$/|/ -k *[!-~]* *$/) " /etc/audit/rules.d/*.rules \
  || printf "ERROR: Variable 'UID_MIN' is unset.\n"
}
```

Verify the output matches:

```
-a always,exit -F path=/usr/bin/chacl -F perm=x -F auid>=1000 -F auid!=unset
-k perm_chng
```

Running configuration

Run the following command to check loaded rules:

```
# {
  UID_MIN=$(awk '/^s*UID_MIN/{print $2}' /etc/login.defs)
  [ -n "${UID_MIN}" ] && auditctl -l | awk "/^ *-a *always,exit/ \
    && (/ -F *auid!=unset/||/ -F *auid==1/||/ -F *auid!=4294967295/) \
    &&/ -F *auid>=${UID_MIN}/ \
    &&/ -F *perm=x/ \
    &&/ -F *path=/usr/bin/chacl/ \
    &&(/ key= *[!-~]* *$/|/ -k *[!-~]* *$/) " \
  || printf "ERROR: Variable 'UID_MIN' is unset.\n"
}
```

Verify the output matches:

```
-a always,exit -S all -F path=/usr/bin/chacl -F perm=x -F auid>=1000 -F
auid==1 -F key=perm_chng
```

Remediation:

Create audit rules

Edit or create a file in the `/etc/audit/rules.d/` directory, ending in `.rules` extension, with the relevant rules to monitor successful and unsuccessful attempts to use the `chacl` command.

Example:

```
# {
  UID_MIN=$(awk '/^s*UID_MIN/{print $2}' /etc/login.defs)
  [ -n "${UID_MIN}" ] && printf "
-a always,exit -F path=/usr/bin/chacl -F perm=x -F auid>=${UID_MIN} -F
auid!=unset -k perm_chng
" >> /etc/audit/rules.d/50-perm_chng.rules || printf "ERROR: Variable
'UID_MIN' is unset.\n"
}
```

Load audit rules

Merge and load the rules into active configuration:

```
# augenrules --load
```

Check if reboot is required.

```
# if [[ $(auditctl -s | grep "enabled") =~ "2" ]]; then printf "Reboot
required to load rules\n"; fi
```

References:

1. NIST SP 800-53 Rev. 5: AU-2, AU-12, SI-5

Additional Information:

Potential reboot required

If the auditing configuration is locked (`-e 2`), then `augenrules` will not warn in any way that rules could not be loaded into the running configuration. A system reboot will be required to load the rules into the running configuration.

System call structure

For performance (`man 7 audit.rules`) reasons it is preferable to have all the system calls on one line. However, your configuration may have them on one line each or some other combination. This is important to understand for both the auditing and remediation sections as the examples given are optimized for performance as per the man page.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<p>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.</p>	●	●	●
v7	<p>6.2 Activate audit logging Ensure that local logging has been enabled on all systems and networking devices.</p>	●	●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562, T1562.006	TA0005	M1022

6.2.3.18 Ensure successful and unsuccessful attempts to use the usermod command are collected (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

The operating system must generate audit records for successful/unsuccessful uses of the **usermod** command.

Rationale:

The **usermod** command modifies the system account files to reflect the changes that are specified on the command line. Without generating audit records that are specific to the security and mission needs of the organization, it would be difficult to establish, correlate, and investigate the events relating to an incident or identify those responsible for one.

Audit records can be generated from various components within the information system (e.g., module or policy filter).

Audit:

On disk configuration

Run the following command to check the on disk rules:

```
# {
  UID_MIN=$(awk '/^s*UID_MIN/{print $2}' /etc/login.defs)
  [ -n "${UID_MIN}" ] && awk "/^ *-a *always,exit/ \
    && (/ -F *auid!=unset/||/ -F *auid!=-1/||/ -F *auid!=4294967295/) \
    &&/ -F *auid>=${UID_MIN}/ \
    &&/ -F *perm=x/ \
    &&/ -F *path=/usr/sbin/usermod/ \
    &&(/ key= *[!-~]* *$/|/ -k *[!-~]* *$/) " /etc/audit/rules.d/*.rules \
  || printf "ERROR: Variable 'UID_MIN' is unset.\n"
}
```

Verify the output matches:

```
-a always,exit -F path=/usr/sbin/usermod -F perm=x -F auid>=1000 -F
auid!=unset -k usermod
```

Running configuration

Run the following command to check loaded rules:

```
# {
  UID_MIN=$(awk '/^s*UID_MIN/{print $2}' /etc/login.defs)
  [ -n "${UID_MIN}" ] && auditctl -l | awk "/^ *-a *always,exit/ \
    && (/ -F *auid!=unset/||/ -F *auid!=-1/||/ -F *auid!=4294967295/) \
    &&/ -F *auid>=${UID_MIN}/ \
    &&/ -F *perm=x/ \
    &&/ -F *path=/usr/sbin/usermod/ \
    &&(/ key= *[!-~]* *$/|/ -k *[!-~]* *$/) " \
  || printf "ERROR: Variable 'UID_MIN' is unset.\n"
}
```

Verify the output matches:

```
-a always,exit -S all -F path=/usr/sbin/usermod -F perm=x -F auid>=1000 -F
auid!=-1 -F key=usermod
```

Remediation:

Create audit rules

Edit or create a file in the `/etc/audit/rules.d/` directory, ending in `.rules` extension, with the relevant rules to monitor successful and unsuccessful attempts to use the `usermod` command.

Example:

```
# {
  UID_MIN=$(awk '/^s*UID_MIN/{print $2}' /etc/login.defs)
  [ -n "${UID_MIN}" ] && printf "
-a always,exit -F path=/usr/sbin/usermod -F perm=x -F auid>=${UID_MIN} -F
auid!=unset -k usermod
" >> /etc/audit/rules.d/50-usermod.rules || printf "ERROR: Variable 'UID_MIN'
is unset.\n"
}
```

Load audit rules

Merge and load the rules into active configuration:

```
# augenrules --load
```

Check if reboot is required.

```
# if [[ $(auditctl -s | grep "enabled") =~ "2" ]]; then printf "Reboot
required to load rules\n"; fi
```

References:

1. NIST SP 800-53 Rev. 5: AU-2, AU-12, SI-5

Additional Information:

Potential reboot required

If the auditing configuration is locked (`-e 2`), then `augenrules` will not warn in any way that rules could not be loaded into the running configuration. A system reboot will be required to load the rules into the running configuration.

System call structure

For performance (`man 7 audit.rules`) reasons it is preferable to have all the system calls on one line. However, your configuration may have them on one line each or some other combination. This is important to understand for both the auditing and remediation sections as the examples given are optimized for performance as per the man page.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<p>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.</p>	●	●	●
v7	<p>6.2 Activate audit logging Ensure that local logging has been enabled on all systems and networking devices.</p>	●	●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562, T1562.006	TA0005	M1022

6.2.3.19 Ensure kernel module loading unloading and modification is collected (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

Monitor the loading and unloading of kernel modules. All the loading / listing / dependency checking of modules is done by **kmod** via symbolic links.

The following system calls control loading and unloading of modules:

- **init_module** - load a module
- **finit_module** - load a module (used when the overhead of using cryptographically signed modules to determine the authenticity of a module can be avoided)
- **delete_module** - delete a module
- **create_module** - create a loadable module entry
- **query_module** - query the kernel for various bits pertaining to modules

Any execution of the loading and unloading module programs and system calls will trigger an audit record with an identifier of **modules**.

Rationale:

Monitoring the use of all the various ways to manipulate kernel modules could provide system administrators with evidence that an unauthorized change was made to a kernel module, possibly compromising the security of the system.

Audit:

On disk configuration

Run the following script to check the on disk rules:

```
#!/usr/bin/env bash

{
    awk '/^ *-a *always,exit/ \
&& -F *arch=b(32|64) / \
&&(/ -F auid!=unset/||/ -F auid!=-1/||/ -F auid!=4294967295/) \
&& -S/ \
&&(/init_module/ \
    ||/finit_module/ \
    ||/delete_module/ \
    ||/create_module/ \
    ||/query_module/) \
&&(/ key= *[!~]* *$/||/ -k *[!~]* *$/)' /etc/audit/rules.d/*.rules

    UID_MIN=$(awk '/^s*UID_MIN/{print $2}' /etc/login.defs)
    [ -n "${UID_MIN}" ] && awk "/^ *-a *always,exit/ \
&&(/ -F *auid!=unset/||/ -F *auid!=-1/||/ -F *auid!=4294967295/) \
&& -F *auid>=${UID_MIN}/ \
&& -F *perm=x/ \
&& -F *path=/usr/bin/kmod/ \
&&(/ key= *[!~]* *$/||/ -k *[!~]* *$/)" /etc/audit/rules.d/*.rules \
    || printf "ERROR: Variable 'UID_MIN' is unset.\n"
}
```

Verify the output matches:

```
-a always,exit -F arch=b64 -S
init_module,finit_module,delete_module,create_module,query_module -F
auid>=1000 -F auid!=unset -k kernel_modules
-a always,exit -F path=/usr/bin/kmod -F perm=x -F auid>=1000 -F auid!=unset -
k kernel_modules
```

Running configuration

Run the following script to check loaded rules:

```

#!/usr/bin/env bash

{
    auditctl -l | awk '/^ *-a *always,exit/ \
&& -F arch=b(32|64) / \
&&(/ -F auid!=unset/||/ -F auid!=-1/||/ -F auid!=4294967295/) \
&& -S/ \
&&(/init_module/ \
||/finit_module/ \
||/delete_module/ \
||/create_module/ \
||/query_module/) \
&&(/ key= *[!~]* *$/||/ -k *[!~]* *$/)' \
UID_MIN=$(awk '/^s*UID_MIN/{print $2}' /etc/login.defs)
[ -n "${UID_MIN}" ] && auditctl -l | awk "/^ *-a *always,exit/ \
&&(/ -F auid!=unset/||/ -F auid!=-1/||/ -F auid!=4294967295/) \
&& -F auid>=${UID_MIN}/ \
&& -F perm=x/ \
&& -F path=/usr/bin/kmod/ \
&&(/ key= *[!~]* *$/||/ -k *[!~]* *$/)" \
|| printf "ERROR: Variable 'UID_MIN' is unset.\n"
}

```

Verify the output includes:

```

-a always,exit -F arch=b64 -S
create_module,init_module,delete_module,query_module,finit_module -F
auid>=1000 -F auid!=-1 -F key=kernel_modules
-a always,exit -S all -F path=/usr/bin/kmod -F perm=x -F auid>=1000 -F
auid!=-1 -F key=kernel_modules

```

Symlink audit

Run the following script to audit if the symlinks **kmod** accepts are indeed pointing at it:

```

#!/usr/bin/env bash

{
    a_files=(/usr/sbin/lsmod "/usr/sbin/rmmod" "/usr/sbin/insmod"
"/usr/sbin/modinfo" "/usr/sbin/modprobe" "/usr/sbin/depmod")
    for l_file in "${a_files[@]}"; do
        if [ "$(readlink -f "$l_file")" = "$(readlink -f /bin/kmod)" ]; then
            printf "OK: \"$l_file\"\n"
        else
            printf "Issue with symlink for file: \"$l_file\"\n"
        fi
    done
}

```

Verify the output states **OK**. If there is a symlink pointing to a different location it should be investigated

Remediation:

Create audit rules

Edit or create a file in the `/etc/audit/rules.d/` directory, ending in `.rules` extension, with the relevant rules to monitor kernel module modification.

Example:

```
#!/usr/bin/env bash

{
    UID_MIN=$(awk '/^s*UID_MIN/{print $2}' /etc/login.defs)
    [ -n "$UID_MIN" ] && printf "
        -a always,exit -F arch=b64 -S
        init_module,finit_module,delete_module,create_module,query_module -F
        auid>=${UID_MIN} -F auid!=unset -k kernel_modules
        -a always,exit -F path=/usr/bin/kmod -F perm=x -F auid>=${UID_MIN} -F
        auid!=unset -k kernel_modules
        " >> /etc/audit/rules.d/50-kernel_modules.rules || printf "ERROR: Variable
        'UID_MIN' is unset.\n"
}
```

Load audit rules

Merge and load the rules into active configuration:

```
# augenrules --load
```

Check if reboot is required.

```
# if [[ $(auditctl -s | grep "enabled") =~ "2" ]]; then printf "Reboot
required to load rules\n"; fi
```

References:

1. NIST SP 800-53 Rev. 5: AU-3, CM-6

Additional Information:

Potential reboot required

If the auditing configuration is locked (`-e 2`), then `augenrules` will not warn in any way that rules could not be loaded into the running configuration. A system reboot will be required to load the rules into the running configuration.

System call structure

For performance (`man 7 audit.rules`) reasons it is preferable to have all the system calls on one line. However, your configuration may have them on one line each or some other combination. This is important to understand for both the auditing and remediation sections as the examples given are optimized for performance as per the man page.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<p>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.</p>		●	●
v7	<p>6.2 Activate audit logging Ensure that local logging has been enabled on all systems and networking devices.</p>	●	●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562, T1562.006	TA0004	M1047

6.2.3.20 Ensure the audit configuration is immutable (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

Set system audit so that audit rules cannot be modified with `auditctl`. Setting the flag `-e 2` forces audit to be put in immutable mode. Audit changes can only be made on system reboot.

Note: This setting will require the system to be rebooted to update the active `auditd` configuration settings.

Rationale:

In immutable mode, unauthorized users cannot execute changes to the audit system to potentially hide malicious activity and then put the audit rules back. Users would most likely notice a system reboot and that could alert administrators of an attempt to make unauthorized audit changes.

Audit:

Run the following command and verify output matches:

```
# grep -Ph -- '^h*-e\h+2\b' /etc/audit/rules.d/*.rules | tail -1  
-e 2
```

Remediation:

Edit or create the file `/etc/audit/rules.d/99-finalize.rules` and add the line `-e 2` at the end of the file:

Example:

```
# printf '\n%s' "-e 2" >> /etc/audit/rules.d/99-finalize.rules
```

Load audit rules

Merge and load the rules into active configuration:

```
# augenrules --load
```

Check if reboot is required.

```
# if [[ $(auditctl -s | grep "enabled") =~ "2" ]]; then printf "Reboot  
required to load rules\n"; fi
```

References:

1. NIST SP 800-53 Rev. 5: AC-3, AU-3, AU-12, MP-2

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<p>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.</p>	●	●	●
v8	<p>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.</p>		●	●
v7	<p>6.2 Activate audit logging Ensure that local logging has been enabled on all systems and networking devices.</p>	●	●	●
v7	<p>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.</p>		●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562, T1562.001	TA0005	M1022

6.2.3.21 Ensure the running and on disk configuration is the same (Manual)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

The Audit system have both on disk and running configuration. It is possible for these configuration settings to differ.

Note: Due to the limitations of `augenrules` and `auditctl`, it is not absolutely guaranteed that loading the rule sets via `augenrules --load` will result in all rules being loaded or even that the user will be informed if there was a problem loading the rules.

Rationale:

Configuration differences between what is currently running and what is on disk could cause unexpected problems or may give a false impression of compliance requirements.

Audit:

Merged rule sets

Ensure that all rules in `/etc/audit/rules.d` have been merged into `/etc/audit/audit.rules`:

```
# augenrules --check  
  
/usr/sbin/augenrules: No change
```

Should there be any drift, run `augenrules --load` to merge and load all rules.

Remediation:

If the rules are not aligned across all three () areas, run the following command to merge and load all rules:

```
# augenrules --load
```

Check if reboot is required.

```
if [[ $(auditctl -s | grep "enabled") =~ "2" ]]; then echo "Reboot required  
to load rules"; fi
```

References:

1. NIST SP 800-53 Rev. 5: AU-3

Additional Information:

Potential reboot required

If the auditing configuration is locked (`-e 2`), then `augenrules` will not warn in any way that rules could not be loaded into the running configuration. A system reboot will be required to load the rules into the running configuration.

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.		●	●

6.2.4 Configure auditd File Access

Without the capability to restrict which roles and individuals can select which events are audited, unauthorized personnel may be able to prevent the auditing of critical events.

6.2.4.1 Ensure audit log files mode is configured (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

Audit log files contain information about the system and system activity.

Rationale:

Access to audit records can reveal system and configuration data to attackers, potentially compromising its confidentiality.

Audit:

Run the following script to verify audit log files are mode **0640** or more restrictive:

```
#!/usr/bin/env bash

{
    l_perm_mask="0137"
    if [ -e "/etc/audit/auditd.conf" ]; then
        l_audit_log_directory=$(dirname $(awk -F= '/^s*log_file\s*/{print $2}' /etc/audit/auditd.conf | xargs))"
        if [ -d "$l_audit_log_directory" ]; then
            l_maxperm=$((printf '%o' $(( 0777 & ~$l_perm_mask )) )"
            a_files=()
            while IFS= read -r -d $'\0' l_file; do
                [ -e "$l_file" ] && a_files+=("$l_file")
            done < <(find "$l_audit_log_directory" -maxdepth 1 -type f -perm /"$l_perm_mask" -print0)
            if (( "${#a_files[@]}" > 0 )); then
                for l_file in "${a_files[@]}"; do
                    l_file_mode=$(stat -Lc '%#a' "$l_file")
                    echo -e "\n- Audit Result:\n  ** FAIL **\n - File:
\"$l_file\" is mode: \"$l_file_mode\"\n      (should be mode: \"$l_maxperm\" or more restrictive)\n"
                done
            else
                echo -e "\n- Audit Result:\n  ** PASS **\n - All files in
\"$l_audit_log_directory\" are mode: \"$l_maxperm\" or more restrictive"
            fi
        else
            echo -e "\n- Audit Result:\n  ** FAIL **\n - Log file directory not set in \"/etc/audit/auditd.conf\" please set log file directory"
            fi
        else
            echo -e "\n- Audit Result:\n  ** FAIL **\n - File:
\"/etc/audit/auditd.conf\" not found.\n      - ** Verify auditd is installed **"
            fi
    }
}
```

Remediation:

Run the following command to remove more permissive mode than **0640** from audit log files:

```
# [ -f /etc/audit/auditd.conf ] && find "$(dirname $(awk -F "=" '/^s*log_file/ {print $2}' /etc/audit/auditd.conf | xargs))" -type f -perm /0137 -exec chmod u-x,g-wx,o-rwx {} +
```

References:

1. NIST SP 800-53 Rev. 5: AU-3

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<p>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.</p>	●	●	●
v7	<p>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.</p>	●	●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1070, T1070.002, T1083, T1083.000	TA0007	M1022

6.2.4.2 Ensure audit log files owner is configured (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

Audit log files contain information about the system and system activity.

Rationale:

Access to audit records can reveal system and configuration data to attackers, potentially compromising its confidentiality.

Audit:

Run the following script to verify audit log files are owned by the **root** user:

```
#!/usr/bin/env bash

{
    l_output="" l_output2=""
    if [ -e "/etc/audit/auditd.conf" ]; then
        l_audit_log_directory=$(dirname $(awk -F= '/^s*log_file\s*/{print $2}' /etc/audit/auditd.conf | xargs))"
        if [ -d "$l_audit_log_directory" ]; then
            while IFS= read -r -d '$\0' l_file; do
                l_output2="$l_output2\n - File: \"$l_file\" is owned by user:
\"$(stat -Lc '%U' \"$l_file\")\"\n      (should be owned by user: \"root\")\n"
            done <<(find "$l_audit_log_directory" -maxdepth 1 -type f ! -user root -print0)
        else
            l_output2="$l_output2\n - Log file directory not set in
\"/etc/audit/auditd.conf\" please set log file directory"
        fi
    else
        l_output2="$l_output2\n - File: \"/etc/audit/auditd.conf\" not
found.\n - ** Verify auditd is installed **"
    fi
    if [ -z "$l_output2" ]; then
        l_output="$l_output\n - All files in \"$l_audit_log_directory\" are
owned by user: \"root\"\n"
        echo -e "\n- Audit Result:\n  ** PASS **\n  * Correctly configured *
:$l_output"
    else
        echo -e "\n- Audit Result:\n  ** FAIL **\n  * Reasons for audit
failure * :$l_output2\n"
    fi
}
```

Remediation:

Run the following command to configure the audit log files to be owned by the **root** user:

```
# [ -f /etc/audit/auditd.conf ] && find "$(dirname $(awk -F= "="
'/^s*log_file/ {print $2}' /etc/audit/auditd.conf | xargs))" -type f ! -user
root -exec chown root {} +
```

References:

1. NIST SP 800-53 Rev. 5: AU-3

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<p>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.</p>	●	●	●
v7	<p>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.</p>	●	●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1070, T1070.002, T1083, T1083.000	TA0007	M1022

6.2.4.3 Ensure audit log files group owner is configured (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

Audit log files contain information about the system and system activity.

Rationale:

Access to audit records can reveal system and configuration data to attackers, potentially compromising its confidentiality.

Audit:

Run the following command to verify **log_group** parameter is set to either **adm** or **root** in **/etc/audit/auditd.conf**:

```
# grep -Piws -- '^h*log_group\h*=\h*\H+\b' /etc/audit/auditd.conf | grep -Pvi -- '(adm)'
```

Nothing should be returned

Using the path of the directory containing the audit logs, verify audit log files are owned by the "root" or "adm" group by running the following script:

```
#!/usr/bin/env bash

{
    if [ -e /etc/audit/auditd.conf ]; then
        l_fpath=$(dirname "$(awk -F "=" '/^s*log_file/ {print $2}' /etc/audit/auditd.conf | xargs)""
        find -L "$l_fpath" -not -path "$l_fpath"/lost+found -type f \(! -group
root -a ! -group adm \) -exec ls -l {} +
    fi
}
```

Nothing should be returned

Remediation:

Run the following command to configure the audit log files to be group owned by **adm**:

```
# find $(dirname $(awk -F"="/^\\s*log_file/ {print $2}' /etc/audit/auditd.conf | xargs)) -type f \(! -group adm -a ! -group root \) -exec chgrp adm {} +
```

Run the following command to set the **log_group** parameter in the audit configuration file to **log_group = adm**:

```
# sed -ri 's/^\\s*#?\\s*log_group\\s*=\\s*\\S+\\s*(\\s*#.*)?.*$ /log_group = adm\\1/' /etc/audit/auditd.conf
```

Run the following command to restart the audit daemon to reload the configuration file:

```
# systemctl restart auditd
```

References:

1. NIST SP 800-53 Rev. 5: AU-3

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.	●	●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1070, T1070.002, T1083, T1083.000	TA0007	M1022

6.2.4.4 Ensure the audit log file directory mode is configured (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

The audit log directory contains audit log files.

Rationale:

Audit information includes all information including: audit records, audit settings and audit reports. This information is needed to successfully audit system activity. This information must be protected from unauthorized modification or deletion. If this information were to be compromised, forensic analysis and discovery of the true source of potentially malicious system activity is impossible to achieve.

Audit:

Run the following script to verify the audit log directory is mode 0750 or more restrictive:

```
#!/usr/bin/env bash

{
    l_perm_mask="0027"
    if [ -e "/etc/audit/auditd.conf" ]; then
        l_audit_log_directory=$(dirname $(awk -F= '/^s*log_file\s*/{print $2}' /etc/audit/auditd.conf | xargs))"
        if [ -d "$l_audit_log_directory" ]; then
            l_maxperm=$(printf '%o' $(( 0777 & ~$l_perm_mask )) )
            l_directory_mode=$(stat -Lc '%#a' "$l_audit_log_directory")
            if [ $(( $l_directory_mode & $l_perm_mask )) -gt 0 ]; then
                echo -e "\n- Audit Result:\n ** FAIL **\n - Directory:
\"$l_audit_log_directory\" is mode: \"$l_directory_mode\"\n      (should be
mode: \"$l_maxperm\" or more restrictive)\n"
            else
                echo -e "\n- Audit Result:\n ** PASS **\n - Directory:
\"$l_audit_log_directory\" is mode: \"$l_directory_mode\"\n      (should be
mode: \"$l_maxperm\" or more restrictive)\n"
            fi
        else
            echo -e "\n- Audit Result:\n ** FAIL **\n - Log file directory not
set in \"/etc/audit/auditd.conf\" please set log file directory"
            fi
        else
            echo -e "\n- Audit Result:\n ** FAIL **\n - File:
\"/etc/audit/auditd.conf\" not found\n - ** Verify auditd is installed **"
            fi
    }
}
```

Remediation:

Run the following command to configure the audit log directory to have a mode of "0750" or less permissive:

```
# chmod g-w,o-rwx "$(dirname $(awk -F= '/^s*log_file\s*/{print $2}' /etc/audit/auditd.conf | xargs))"
```

Default Value:

750

References:

1. NIST SP 800-53 Rev. 5: AU-3

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<p>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.</p>	●	●	●
v7	<p>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.</p>	●	●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1070, T1070.002, T1083, T1083.000	TA0007	M1022

6.2.4.5 Ensure audit configuration files mode is configured (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

Audit configuration files control auditd and what events are audited.

Rationale:

Access to the audit configuration files could allow unauthorized personnel to prevent the auditing of critical events.

Misconfigured audit configuration files may prevent the auditing of critical events or impact the system's performance by overwhelming the audit log. Misconfiguration of the audit configuration files may also make it more difficult to establish and investigate events relating to an incident.

Audit:

Run the following script to verify that the audit configuration files are mode **0640** or more restrictive:

```
#!/usr/bin/env bash

{
    l_output="" l_output2="" l_perm_mask="0137"
    l_maxperm=$( printf '%o' $(( 0777 & ~$l_perm_mask )) )
    while IFS= read -r -d $'\0' l_fname; do
        l_mode=$(stat -Lc '%#a' "$l_fname")
        if [ $(( "$l_mode" & "$l_perm_mask" )) -gt 0 ]; then
            l_output2="$l_output2\n- file: \"$l_fname\" is mode: \"$l_mode\""
            (should be mode: \"$l_maxperm\" or more restrictive)"
        fi
    done < <(find /etc/audit/ -type f \(
        -name "*.conf" -o -name "*.rules" \
        -print0)
        if [ -z "$l_output2" ]; then
            echo -e "\n- Audit Result:\n  ** PASS **\n- All audit configuration
files are mode: \"$l_maxperm\" or more restrictive"
        else
            echo -e "\n- Audit Result:\n  ** FAIL **\n$l_output2"
        fi
}
```

Remediation:

Run the following command to remove more permissive mode than **0640** from the audit configuration files:

```
# find /etc/audit/ -type f \(-name '*.conf' -o -name '*.rules'\) -exec chmod u-x,g-wx,o-rwx {} +
```

References:

1. NIST SP 800-53 Rev. 5: AU-3

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.	●	●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1070, T1070.002, T1083, T1083.000	TA0007	M1022

6.2.4.6 Ensure audit configuration files owner is configured (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

Audit configuration files control auditd and what events are audited.

Rationale:

Access to the audit configuration files could allow unauthorized personnel to prevent the auditing of critical events.

Misconfigured audit configuration files may prevent the auditing of critical events or impact the system's performance by overwhelming the audit log. Misconfiguration of the audit configuration files may also make it more difficult to establish and investigate events relating to an incident.

Audit:

Run the following command to verify that the audit configuration files are owned by the root user:

```
# find /etc/audit/ -type f \(-name '*.conf' -o -name '*.rules'\) ! -user root
```

Nothing should be returned

Remediation:

Run the following command to change ownership to **root** user:

```
# find /etc/audit/ -type f \(-name '*.conf' -o -name '*.rules'\) ! -user root -exec chown root {} +
```

References:

1. NIST SP 800-53 Rev. 5: AU-3

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<p>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.</p>	●	●	●
v7	<p>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.</p>	●	●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1070, T1070.002, T1083, T1083.000	TA0007	M1022

6.2.4.7 Ensure audit configuration files group owner is configured (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

Audit configuration files control auditd and what events are audited.

Rationale:

Access to the audit configuration files could allow unauthorized personnel to prevent the auditing of critical events.

Misconfigured audit configuration files may prevent the auditing of critical events or impact the system's performance by overwhelming the audit log. Misconfiguration of the audit configuration files may also make it more difficult to establish and investigate events relating to an incident.

Audit:

Run the following command to verify that the audit configuration files are owned by the group **root**:

```
# find /etc/audit/ -type f \(-name '*.conf' -o -name '*.rules'\) ! -group root
```

Nothing should be returned

Remediation:

Run the following command to change group to **root**:

```
# find /etc/audit/ -type f \(-name '*.conf' -o -name '*.rules'\) ! -group root -exec chgrp root {} +
```

References:

1. NIST SP 800-53 Rev. 5: AU-3

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<p>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.</p>	●	●	●
v7	<p>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.</p>	●	●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1070, T1070.002, T1083, T1083.000	TA0007	M1022

6.2.4.8 Ensure audit tools mode is configured (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

Audit tools include, but are not limited to, vendor-provided and open source audit tools needed to successfully view and manipulate audit information system activity and records. Audit tools include custom queries and report generators.

Rationale:

Protecting audit information includes identifying and protecting the tools used to view and manipulate log data. Protecting audit tools is necessary to prevent unauthorized operation on audit information.

Audit:

Run the following script to verify the audit tools are mode **0755** or more restrictive:

```
#!/usr/bin/env bash

{
    l_output="" l_output2="" l_perm_mask="0022"
    l_maxperm=$(printf '%o' $(( 0777 & ~$l_perm_mask )) )
    a_audit_tools=("/sbin/auditctl" "/sbin/aureport" "/sbin/ausearch"
    "/sbin/autrace" "/sbin/auditd" "/sbin/augenrules")
    for l_audit_tool in "${a_audit_tools[@]}"; do
        l_mode=$(stat -Lc '%#a' "$l_audit_tool")
        if [ $(( ${l_mode} & ${l_perm_mask} )) -gt 0 ]; then
            l_output2="$l_output2\n - Audit tool \"$l_audit_tool\" is mode:
\"$l_mode\" and should be mode: \"$l_maxperm\" or more restrictive"
        else
            l_output="$l_output\n - Audit tool \"$l_audit_tool\" is correctly
configured to mode: \"$l_mode\""
        fi
    done
    if [ -z "$l_output2" ]; then
        echo -e "\n- Audit Result:\n  ** PASS **\n - * Correctly configured *\n:$l_output"
    else
        echo -e "\n- Audit Result:\n  ** FAIL **\n - * Reasons for audit
failure *\n:$l_output2\n"
        [ -n "$l_output" ] && echo -e "\n - * Correctly configured *\n:$l_output\n"
    fi
    unset a_audit_tools
}
```

Remediation:

Run the following command to remove more permissive mode from the audit tools:

```
# chmod go-w /sbin/auditctl /sbin/aureport /sbin/ausearch /sbin/autrace
/sbin/auditd /sbin/augenrules
```

References:

1. NIST SP 800-53 Rev. 5: AU-3

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<p>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.</p>	●	●	●
v7	<p>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.</p>	●	●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1070, T1070.002, T1083, T1083.000	TA0007	M1022

6.2.4.9 Ensure audit tools owner is configured (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

Audit tools include, but are not limited to, vendor-provided and open source audit tools needed to successfully view and manipulate audit information system activity and records. Audit tools include custom queries and report generators.

Rationale:

Protecting audit information includes identifying and protecting the tools used to view and manipulate log data. Protecting audit tools is necessary to prevent unauthorized operation on audit information.

Audit:

Run the following command to verify the audit tools are owned by the **root** user:

```
# stat -Lc "%n %U" /sbin/auditctl /sbin/aureport /sbin/ausearch /sbin/autrace  
/sbin/auditd /sbin/augenrules | awk '$2 != "root" {print}'
```

Nothing should be returned

Remediation:

Run the following command to change the owner of the audit tools to the **root** user:

```
# chown root /sbin/auditctl /sbin/aureport /sbin/ausearch /sbin/autrace  
/sbin/auditd /sbin/augenrules
```

References:

1. NIST SP 800-53 Rev. 5: AU-3

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<p>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.</p>	●	●	●
v7	<p>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.</p>	●	●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1070, T1070.002, T1083, T1083.000	TA0007	M1022

6.2.4.10 Ensure audit tools group owner is configured (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

Audit tools include, but are not limited to, vendor-provided and open source audit tools needed to successfully view and manipulate audit information system activity and records. Audit tools include custom queries and report generators.

Rationale:

Protecting audit information includes identifying and protecting the tools used to view and manipulate log data. Protecting audit tools is necessary to prevent unauthorized operation on audit information.

Audit:

Run the following command to verify the audit tools are owned by the group **root**

```
# stat -Lc "%n %G" /sbin/auditctl /sbin/aureport /sbin/ausearch /sbin/autrace  
/sbin/auditd /sbin/augenrules | awk '$2 != "root" {print}'
```

Nothing should be returned

Remediation:

Run the following command to change group ownership to the group **root**:

```
# chgrp root /sbin/auditctl /sbin/aureport /sbin/ausearch /sbin/autrace  
/sbin/auditd /sbin/augenrules
```

References:

1. NIST SP 800-53 Rev. 5: AU-3

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<p>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.</p>	●	●	●
v7	<p>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.</p>	●	●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1070, T1070.002, T1083, T1083.000	TA0007	M1022