2.3 Configure Time Synchronization

It is recommended that systems be configured to synchronize their time using a service such as `systemd-timesyncd`, or `chrony`.

Virtual systems may be configured to receive their time synchronization from their host system.

The host system must be configured to synchronize its time from an authoritative source to be considered compliant with this section.

Any "physical" clock present on a system should be synchronized from an authoritative time source.

Only one time synchronization method should be in use on the system

Notes: Only the section related to the time synchronization method in use on the system should be followed, all other time synchronization recommendations should be skipped

2.3.1 Ensure time synchronization is in use

It is recommended that physical systems and virtual guests lacking direct access to the physical host's clock be configured to synchronize their time using a service such as `systemd-timesyncd`, or `chrony`.

2.3.1.1 Ensure a single time synchronization daemon is in use (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

System time should be synchronized between all systems in an environment. This is typically done by establishing an authoritative time server or set of servers and having all systems synchronize their clocks to them.

Note:

- **On virtual systems where host based time synchronization is available consult your virtualization software documentation and verify that host based synchronization is in use and follows local site policy. In this scenario, this section should be skipped**
- Only **one** time synchronization method should be in use on the system. Configuring multiple time synchronization methods could lead to unexpected or unreliable results

Rationale:

Time synchronization is important to support time sensitive security mechanisms and ensures log files have consistent time records across the enterprise, which aids in forensic investigations.

Audit:

On physical systems, and virtual systems where host based time synchronization is not available.
One of the two time synchronization daemons should be available; **`chrony`** or **`systemd-timesyncd`**
Run the following script to verify that a single time synchronization daemon is available on the system:

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

{
    l_output="" l_output2=""
    service_not_enabled_chk()
    {
        l_out2=""
        if systemctl is-enabled "$l_service_name" 2>/dev/null | grep -q 'enabled'; then
            l_out2="$l_out2\n  - Daemon: \"$l_service_name\" is enabled on the system"
        fi
        if systemctl is-active "$l_service_name" 2>/dev/null | grep -q '^active'; then
            l_out2="$l_out2\n  - Daemon: \"$l_service_name\" is active on the system"
        fi
    }
    l_service_name="systemd-timesyncd.service" # Check systemd-timesyncd daemon
    service_not_enabled_chk
    if [ -n "$l_out2" ]; then
        l_timesyncd="y"
        l_out_tsd="$l_out2"
    else
        l_timesyncd="n"
        l_out_tsd="\n  - Daemon: \"$l_service_name\" is not enabled and not active on the system"
    fi
    l_service_name="chrony.service" # Check chrony
    service_not_enabled_chk
    if [ -n "$l_out2" ]; then
        l_chrony="y"
        l_out_chrony="$l_out2"
    else
        l_chrony="n"
        l_out_chrony="\n  - Daemon: \"$l_service_name\" is not enabled and not active on the 
system"
    fi
    l_status="$l_timesyncd$l_chrony"
    case "$l_status" in
        yy)
            l_output2=" - More than one time sync daemon is in use on the 
system$l_out_tsd$l_out_chrony"
            ;;
        nn)
            l_output2=" - No time sync daemon is in use on the system$l_out_tsd$l_out_chrony"
            ;;
        yn|ny)
            l_output=" - Only one time sync daemon is in use on the 
system$l_out_tsd$l_out_chrony"
            ;;
        *)
            l_output2=" - Unable to determine time sync daemon(s) status"
            ;;
    esac

    if [ -z "$l_output2" ]; then
        echo -e "\n- Audit Result:\n  ** PASS **\n$l_output\n"
    else
        echo -e "\n- Audit Result:\n  ** FAIL **\n - * Reasons for audit failure * 
:\n$l_output2\n"
    fi
}
```

Note: Follow the guidance in the subsection for the time synchronization daemon available on the system and skip the other time synchronization daemon subsection.

Remediation:

On physical systems, and virtual systems where host based time synchronization is not available.
Select **one** of the two time synchronization daemons; **`chrony (1)`** or **`systemd-timesyncd (2)`** and following the remediation procedure for the selected daemon.
Note: enabling more than one synchronization daemon could lead to unexpected or unreliable results:

1. **`chrony`**

Run the following command to install `chrony`:

```
# apt install chrony
```

Run the following commands to stop and mask the `systemd-timesyncd` daemon:

```
# systemctl stop systemd-timesyncd.service

# systemctl mask systemd-timesyncd.service
```

Note:

- Subsection: ***Configure chrony*** should be followed
- Subsection: ***Configure systemd-timesyncd*** should be skipped

2. **`systemd-timesyncd`**

Run the following command to remove the chrony package:

```
# apt purge chrony
# apt autoremove chrony
```

Note:

- Subsection: ***Configure systemd-timesyncd*** should be followed
- Subsection: ***Configure chrony*** should be skipped

References:

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

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.4 Standardize Time Synchronization Standardize time synchronization. Configure at least two synchronized time sources across enterprise assets, where supported.		●	●
v7	6.1 Utilize Three Synchronized Time Sources Use at least three synchronized time sources from which all servers and network devices retrieve time information on a regular basis so that timestamps in logs are consistent.		●	●

MITRE ATT&CK Mappings:

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

2.3.2 Configure systemd-timesyncd

`systemd-timesyncd` is a daemon that has been added for synchronizing the system clock across the network. It implements an SNTP client. In contrast to NTP implementations such as chrony or the NTP reference server this only implements a client side, and does not bother with the full NTP complexity, focusing only on querying time from one remote server and synchronizing the local clock to it. The daemon runs with minimal privileges, and has been hooked up with networkd to only operate when network connectivity is available. The daemon saves the current clock to disk every time a new NTP sync has been acquired, and uses this to possibly correct the system clock early at bootup, in order to accommodate for systems that lack an RTC such as the Raspberry Pi and embedded devices, and make sure that time monotonically progresses on these systems, even if it is not always correct. To make use of this daemon a new system user and group "systemd-timesync" needs to be created on installation of systemd.

The default configuration is set during compilation, so configuration is only needed when it is necessary to deviate from those defaults. Initially, the main configuration file in /etc/systemd/ contains commented out entries showing the defaults as a guide to the administrator. Local overrides can be created by editing this file or by creating drop-ins, as described below. Using drop-ins for local configuration is recommended over modifications to the main configuration file.

In addition to the "main" configuration file, drop-in configuration snippets are read from `/usr/lib/systemd/*.conf.d/`, `/usr/local/lib/systemd/*.conf.d/`, and `/etc/systemd/*.conf.d/`. Those drop-ins have higher precedence and override the main configuration file. Files in the *.conf.d/ configuration subdirectories are sorted by their filename in lexicographic order, regardless of in which of the subdirectories they reside. When multiple files specify the same option, for options which accept just a single value, the entry in the file sorted last takes precedence, and for options which accept a list of values, entries are collected as they occur in the sorted files.

When packages need to customize the configuration, they can install drop-ins under /usr/. Files in /etc/ are reserved for the local administrator, who may use this logic to override the configuration files installed by vendor packages. Drop-ins have to be used to override package drop-ins, since the main configuration file has lower precedence. It is recommended to prefix all filenames in those subdirectories with a two-digit number and a dash, to simplify the ordering of the files.

To disable a configuration file supplied by the vendor, the recommended way is to place a symlink to /dev/null in the configuration directory in /etc/, with the same filename as the vendor configuration file.

Note:

- The recommendations in this section only apply if `timesyncd` is in use on the system
- The `systemd-timesyncd` service specifically implements only SNTP.
    - This minimalistic service will set the system clock for large offsets or slowly adjust it for smaller deltas
    - More complex use cases are not covered by `systemd-timesyncd`
- **If `chrony` is used, `systemd-timesyncd` should be stopped and masked, and this section skipped**
- **One, and only one, time synchronization method should be in use on the system**

2.3.2.1 Ensure systemd-timesyncd configured with authorized timeserver (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

`NTP=`

- A space-separated list of NTP server host names or IP addresses. During runtime this list is combined with any per-interface NTP servers acquired from systemd-networkd.service(8). systemd-timesyncd will contact all configured system or per-interface servers in turn, until one responds. When the empty string is assigned, the list of NTP servers is reset, and all prior assignments will have no effect. This setting defaults to an empty list.

`FallbackNTP=`

- A space-separated list of NTP server host names or IP addresses to be used as the fallback NTP servers. Any per-interface NTP servers obtained from systemd-networkd.service(8) take precedence over this setting, as do any servers set via NTP= above. This setting is hence only relevant if no other NTP server information is known. When the empty string is assigned, the list of NTP servers is reset, and all prior assignments will have no effect. If this option is not given, a compiled-in list of NTP servers is used.

Rationale:

Time synchronization is important to support time sensitive security mechanisms and to ensure log files have consistent time records across the enterprise to aid in forensic investigations

Audit:

Run the following command to verify the `NTP` **and/or** `FallbackNTP` option is set to local site approved authoritative time server(s):

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

{
   a_output=(); a_output2=(); a_parlist=("NTP=[^#\n\r]+" "FallbackNTP=[^#\n\r]+")
   l_systemd_config_file="/etc/systemd/timesyncd.conf" # Main systemd configuration file
   f_config_file_parameter_chk()
   {
      unset A_out; declare -A A_out # Check config file(s) setting
      while read -r l_out; do
         if [ -n "$l_out" ]; then
            if [[ $l_out =~ ^\s*# ]]; then
               l_file="${l_out//# /}"
            else
               l_systemd_parameter="$(awk -F= '{print $1}' <<< "$l_out" | xargs)"
               grep -Piq -- "^\h*$l_systemd_parameter_name\b" <<< "$l_systemd_parameter" &&
A_out+=(["$l_systemd_parameter"]="$l_file")
            fi
         fi
      done < <("$l_systemdanalyze" cat-config "$l_systemd_config_file" | grep -Pio
'^\h*([^#\n\r]+|#\h*\/[^#\n\r\h]+\.conf\b)')
      if (( ${#A_out[@]} > 0 )); then # Assess output from files and generate output
         while IFS="=" read -r l_systemd_file_parameter_name l_systemd_file_parameter_value; do
            l_systemd_file_parameter_name="${l_systemd_file_parameter_name// /}"
            l_systemd_file_parameter_value="${l_systemd_file_parameter_value// /}"
            if grep -Piq "\b$l_systemd_parameter_value\b" <<< "$l_systemd_file_parameter_value";
then
               a_output+=(" - \"$l_systemd_parameter_name\" is correctly set to
\"$l_systemd_file_parameter_value\"" \
               "    in \"$(printf '%s' "${A_out[@]}")\"")
            else
               a_output2+=(" - \"$l_systemd_parameter_name\" is incorrectly set to
\"$l_systemd_file_parameter_value\"" \
               "    in \"$(printf '%s' "${A_out[@]}")\" and should have a value matching:
\"$l_value_out\"")
            fi
         done < <(grep -Pio -- "^\h*$l_systemd_parameter_name\h*=\h*\H+" "${A_out[@]}")
      else
         a_output2+=(" - \"$l_systemd_parameter_name\" is not set in an included file" \
         "   *** Note: \"$l_systemd_parameter_name\" May be set in a file that's ignored by load
procedure ***")
      fi
   }
   l_systemdanalyze="$(readlink -f /bin/systemd-analyze)"
   while IFS="=" read -r l_systemd_parameter_name l_systemd_parameter_value; do # Assess and
check parameters
      l_systemd_parameter_name="${l_systemd_parameter_name// /}";
l_systemd_parameter_value="${l_systemd_parameter_value// /}"
      l_value_out="${l_systemd_parameter_value//-/ through }"; l_value_out="${l_value_out//|/ or
}"
      l_value_out="$(tr -d '(){}' <<< "$l_value_out")"
      f_config_file_parameter_chk
   done < <(printf '%s\n' "${a_parlist[@]}")
   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
}
```

Example output:

```
- Audit Result:
  ** PASS **
 - "NTP" is correctly set to "time.nist.gov"
    in "/etc/systemd/timesyncd.conf.d/60-timesyncd.conf"
 - "FallbackNTP" is correctly set to "time-a-g.nist.gov"
    in "/etc/systemd/timesyncd.conf.d/60-timesyncd.conf"
```

Note: Please ensure the output for `NTP` and/or `FallbackNTP` is in accordance with local site policy. The timeservers in the example output are provided as an example of possible timeservers and they may not follow local site policy.

Remediation:

Set `NTP` and/or `FallbackNPT` parameters to local site approved authoritative time server(s) in `/etc/systemd/timesyncd.conf` or a file in `/etc/systemd/timesyncd.conf.d/` ending in `.conf` in the `[Time]` section:
Example file:

```
[Time]
NTP=time.nist.gov # Uses the generic name for NIST's time servers
FallbackNTP=time-a-g.nist.gov time-b-g.nist.gov time-c-g.nist.gov # Space
separated list of NIST time servers
```

Example script to create systemd drop-in configuration file:

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

{
   a_settings=("NTP=time.nist.gov" "FallbackNTP=time-a-g.nist.gov time-b-
g.nist.gov time-c-g.nist.gov")
   [ ! -d /etc/systemd/timesyncd.conf.d/ ] && mkdir
/etc/systemd/timesyncd.conf.d/
   if grep -Psq -- '^\h*\[Time\]' /etc/systemd/timesyncd.conf.d/60-
timesyncd.conf; then
      printf '%s\n' "" "${a_settings[@]}" >>
/etc/systemd/timesyncd.conf.d/60-timesyncd.conf
   else
      printf '%s\n' "" "[Time]" "${a_settings[@]}" >>
/etc/systemd/timesyncd.conf.d/60-timesyncd.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:

#NTP=

#FallbackNTP=

References:

1. https://www.freedesktop.org/software/systemd/man/timesyncd.conf.html
2. https://tf.nist.gov/tf-cgi/servers.cgi
3. NIST SP 800-53 Rev. 5: AU-7, AU-8

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.4 Standardize Time Synchronization Standardize time synchronization. Configure at least two synchronized time sources across enterprise assets, where supported.		●	●
v7	6.1 Utilize Three Synchronized Time Sources Use at least three synchronized time sources from which all servers and network devices retrieve time information on a regular basis so that timestamps in logs are consistent.		●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1070, T1070.002, T1562, T1562.001	TA0002	M1022

2.3.2.2 Ensure systemd-timesyncd is enabled and running (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

systemd-timesyncd is a daemon that has been added for synchronizing the system clock across the network

Rationale:

systemd-timesyncd needs to be enabled and running in order to synchronize the system to a timeserver.

Time synchronization is important to support time sensitive security mechanisms and to ensure log files have consistent time records across the enterprise to aid in forensic investigations

Audit:

- IF - systemd-timesyncd is in use on the system, run the following commands:
Run the following command to verify that the `systemd-timesyncd` service is enabled:

```
# systemctl is-enabled systemd-timesyncd.service

enabled
```

Run the following command to verify that the `systemd-timesyncd` service is active:

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

active
```

Remediation:

- IF - systemd-timesyncd is in use on the system, run the following commands:
Run the following command to unmask systemd-timesyncd.service:

```
# systemctl unmask systemd-timesyncd.service
```

Run the following command to enable and start systemd-timesyncd.service:

```
# systemctl --now enable systemd-timesyncd.service
```

- OR -
If another time synchronization service is in use on the system, run the following command to stop and mask systemd-timesyncd:

```
# systemctl --now mask systemd-timesyncd.service
```

References:

1. NIST SP 800-53 Rev. 5: AU-7, AU-8

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.4 Standardize Time Synchronization Standardize time synchronization. Configure at least two synchronized time sources across enterprise assets, where supported.		●	●
v7	6.1 Utilize Three Synchronized Time Sources Use at least three synchronized time sources from which all servers and network devices retrieve time information on a regular basis so that timestamps in logs are consistent.		●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1070, T1070.002, T1562, T1562.001	TA0002	M1022

2.3.3 Configure chrony

`chrony` is a daemon which implements the Network Time Protocol (NTP) and is designed to synchronize system clocks across a variety of systems and use a source that is highly accurate.

`chrony` can be configured to be a client and/or a server.

More information on `chrony` can be found at: http://chrony.tuxfamily.org/.

Note:

- If `systemd-timesyncd` is being used, `chrony` should be removed and this section skipped
- Only one time synchronization method should be in use on the system

2.3.3.1 Ensure chrony is configured with authorized timeserver (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

- server
    - The server directive specifies an NTP server which can be used as a time source. The client-server relationship is strictly hierarchical: a client might synchronize its system time to that of the server, but the server’s system time will never be influenced by that of a client.
    - This directive can be used multiple times to specify multiple servers.
    - The directive is immediately followed by either the name of the server, or its IP address.
- pool
    - The syntax of this directive is similar to that for the server directive, except that it is used to specify a pool of NTP servers rather than a single NTP server. The pool name is expected to resolve to multiple addresses which might change over time.
    - This directive can be used multiple times to specify multiple pools.
    - All options valid in the server directive can be used in this directive too.

Rationale:

Time synchronization is important to support time sensitive security mechanisms and to ensure log files have consistent time records across the enterprise to aid in forensic investigations

Audit:

- IF - `chrony` is in use on the system, run the following script to ensure `chrony` is configured with an authorized timeserver:

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

{
   a_output=() a_output2=() a_config_files=("/etc/chrony/chrony.conf")
   l_include='(confdir|sourcedir)' l_parameter_name='(server|pool)'
l_parameter_value='.+'
   while IFS= read -r l_conf_loc; do
      l_dir="" l_ext=""
      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
      if [[ -n "$l_dir" && -n "$l_ext" ]]; then
         while IFS= 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)
      fi
   done < <(awk '$1~/^\s*'"$l_include"'$/{print $2}' "${a_config_files[*]}"
2>/dev/null)
   for l_file in "${a_config_files[@]}"; do
      l_parameter_line="$(grep -Psi
'^\h*'"$l_parameter_name"'(\h+|\h*:\h*)'"$l_parameter_value"'\b' "$l_file")"
      [ -n "$l_parameter_line" ] && a_output+=("  - Parameter: \"$(tr -d '()'
<<< ${l_parameter_name//|/ or })\"" \
      "    Exists in the file: \"$l_file\" as:" "$l_parameter_line")
   done
   [ "${#a_output[@]}" -le "0" ] && a_output2+=("  - Parameter: \"$(tr -d
'()' <<< ${l_parameter_name//|/ or })\"" \
   "    Does not exist in the chrony configuration")
   if [ "${#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[@]}"
   fi
}
```

Remediation:

Edit `/etc/chrony/chrony.conf` or a file ending in `.sources` in `/etc/chrony/sources.d/` and add or edit server or pool lines as appropriate according to local site policy:
Edit the `Chrony` configuration and add or edit the server and/or pool lines returned by the Audit Procedure as appropriate according to local site policy

```
<[server|pool]> <[remote-server|remote-pool]>
```

Example script to add a drop-in configuration for the `pool` directive:

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

{
   [ ! -d "/etc/chrony/sources.d/" ] && mkdir /etc/chrony/sources.d/
   printf '%s\n' "" "#The maxsources option is unique to the pool directive"
\
   "pool time.nist.gov iburst maxsources 4" >> /etc/chrony/sources.d/60-
sources.sources
   chronyc reload sources &>/dev/null
}
```

Example script to add a drop-in configuration for the `server` directive:

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

{
   [ ! -d "/etc/chrony/sources.d/" ] && mkdir /etc/chrony/sources.d/
   printf '%s\n' "" "server time-a-g.nist.gov iburst" "server 132.163.97.3
iburst" \
   "server time-d-b.nist.gov iburst" >> /etc/chrony/sources.d/60-
sources.sources
   chronyc reload sources &>/dev/null
}
```

Run the following command to reload the `chronyd` config:

```
# systemctl reload-or-restart chronyd
```

References:

1. chrony.conf(5) Manual Page
2. https://tf.nist.gov/tf-cgi/servers.cgi
3. NIST SP 800-53 Rev. 5: AU-3, AU-12

Additional Information:

If pool and/or server directive(s) are set in a sources file in `/etc/chrony/sources.d`, the line:

```
sourcedir /etc/chrony/sources.d
```

must be present in `/etc/chrony/chrony.conf`

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.4 Standardize Time Synchronization Standardize time synchronization. Configure at least two synchronized time sources across enterprise assets, where supported.		●	●
v7	6.1 Utilize Three Synchronized Time Sources Use at least three synchronized time sources from which all servers and network devices retrieve time information on a regular basis so that timestamps in logs are consistent.		●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1070, T1070.002, T1562, T1562.001	TA0002	M1022

2.3.3.2 Ensure chrony is running as user _chrony (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The chrony package is installed with a dedicated user account _chrony. This account is granted the access required by the chronyd service

Rationale:

The chronyd service should run with only the required privlidges

Audit:

- IF - chrony is in use on the system, run the following command to verify the chronyd service is being run as the _chrony user:

```
# ps -ef | awk '(/[c]hronyd/ && $1!="_chrony") { print $1 }'
```

Nothing should be returned

Remediation:

Add or edit the user line to /etc/chrony/chrony.conf or a file ending in .conf in /etc/chrony/conf.d/:

```
user _chrony
```

- OR -
If another time synchronization service is in use on the system, run the following command to remove chrony from the system:

```
# apt purge chrony
# apt autoremove chrony
```

Default Value:

user _chrony

References:

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

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.4 Standardize Time Synchronization Standardize time synchronization. Configure at least two synchronized time sources across enterprise assets, where supported.		●	●
v7	6.1 Utilize Three Synchronized Time Sources Use at least three synchronized time sources from which all servers and network devices retrieve time information on a regular basis so that timestamps in logs are consistent.		●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1070, T1070.002, T1562, T1562.001	TA0002	M1022

2.3.3.3 Ensure chrony is enabled and running (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

chrony is a daemon for synchronizing the system clock across the network

Rationale:

chrony needs to be enabled and running in order to synchronize the system to a timeserver.

Time synchronization is important to support time sensitive security mechanisms and to ensure log files have consistent time records across the enterprise to aid in forensic investigations

Audit:

- IF - chrony is in use on the system, run the following commands:
Run the following command to verify that the `chrony` service is enabled:

```
# systemctl is-enabled chrony.service

enabled
```

Run the following command to verify that the `chrony` service is active:

```
# systemctl is-active chrony.service

active
```

Remediation:

- IF - chrony is in use on the system, run the following commands:
Run the following command to unmask chrony.service:

```
# systemctl unmask chrony.service
```

Run the following command to enable and start chrony.service:

```
# systemctl --now enable chrony.service
```

- OR -
If another time synchronization service is in use on the system, run the following command to remove chrony:

```
# apt purge chrony
# apt autoremove chrony
```

References:

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

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.4 Standardize Time Synchronization Standardize time synchronization. Configure at least two synchronized time sources across enterprise assets, where supported.		●	●
v7	6.1 Utilize Three Synchronized Time Sources Use at least three synchronized time sources from which all servers and network devices retrieve time information on a regular basis so that timestamps in logs are consistent.		●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1070, T1070.002, T1562, T1562.001	TA0002	M1022

2.4 Job Schedulers

A job scheduler is used to execute jobs, commands, or shell scripts, at fixed times, dates, or intervals

2.4.1 Configure cron

`cron` is a time based job scheduler

- IF - `cron` is not installed on the system, this sub section can be skipped

Note: Other methods such as `systemd timers` exist for scheduling jobs. If another method is used `cron` should may be removed. The alternate method should be secured in accordance with local site policy

2.4.1.1 Ensure cron daemon is enabled and active (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The `cron` daemon is used to execute batch jobs on the system.

Rationale:

While there may not be user jobs that need to be run on the system, the system does have maintenance jobs that may include security monitoring that have to run, and `cron` is used to execute them.

Audit:

- IF - cron is installed on the system:
Run the following command to verify `cron` is enabled:

```
# systemctl list-unit-files | awk '$1~/^crond?\.service/{print $2}'

enabled
```

Run the following command to verify that `cron` is active:

```
# systemctl list-units | awk '$1~/^crond?\.service/{print $3}'

active
```

Remediation:

- IF - cron is installed on the system:
Run the following commands to unmask, enable, and start `cron`:

```
# systemctl unmask "$(systemctl list-unit-files | awk
'$1~/^crond?\.service/{print $1}')"
# systemctl --now enable "$(systemctl list-unit-files | awk
'$1~/^crond?\.service/{print $1}')"
```

References:

1. NIST SP 800-53 Rev. 5: CM-1, CM-2, CM-6, CM-7, IA-5

MITRE ATT&CK Mappings:

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

2.4.1.2 Ensure permissions on /etc/crontab are configured (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The `/etc/crontab` file is used by `cron` to control its own jobs. The commands in this item make sure that root is the user and group owner of the file and that only the owner can access the file.

Rationale:

This file contains information on what system jobs are run by cron. Write access to these files could provide unprivileged users with the ability to elevate their privileges. Read access to these files could provide users with the ability to gain insight on system jobs that run on the system and could provide them a way to gain unauthorized privileged access.

Audit:

- IF - cron is installed on the system:
Run the following command and verify `Uid` and `Gid` are both `0/root` and `Access` does not grant permissions to `group` or `other` :

```
# stat -Lc 'Access: (%a/%A) Uid: ( %u/ %U) Gid: ( %g/ %G)' /etc/crontab

Access: (600/-rw-------) Uid: ( 0/ root) Gid: ( 0/ root)
```

Remediation:

- IF - cron is installed on the system:
Run the following commands to set ownership and permissions on `/etc/crontab`:

```
# chown root:root /etc/crontab
# chmod og-rwx /etc/crontab
```

Default Value:

Access: (644/-rw-r--r--) Uid: (0/ root) Gid: (0/ root)

References:

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

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
T1053, T1053.003	TA0002, TA0007	M1018

2.4.1.3 Ensure permissions on /etc/cron.hourly are configured (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

This directory contains system `cron` jobs that need to run on an hourly basis. The files in this directory cannot be manipulated by the `crontab` command, but are instead edited by system administrators using a text editor. The commands below restrict read/write and search access to user and group root, preventing regular users from accessing this directory.

Rationale:

Granting write access to this directory for non-privileged users could provide them the means for gaining unauthorized elevated privileges. Granting read access to this directory could give an unprivileged user insight in how to gain elevated privileges or circumvent auditing controls.

Audit:

- IF - cron is installed on the system:
Run the following command and verify `Uid` and `Gid` are both `0/root` and `Access` does not grant permissions to `group` or `other`:

```
# stat -Lc 'Access: (%a/%A) Uid: ( %u/ %U) Gid: ( %g/ %G)' /etc/cron.hourly/

Access: (700/drwx------) Uid: ( 0/ root) Gid: ( 0/ root)
```

Remediation:

- IF - cron is installed on the system:
Run the following commands to set ownership and permissions on the `/etc/cron.hourly` directory:

```
# chown root:root /etc/cron.hourly/
# chmod og-rwx /etc/cron.hourly/
```

Default Value:

Access: (755/drwxr-xr-x) Uid: (0/ root) Gid: (0/ root)

References:

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

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
T1053, T1053.003	TA0002, TA0007	M1018

2.4.1.4 Ensure permissions on /etc/cron.daily are configured (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The `/etc/cron.daily` directory contains system cron jobs that need to run on a daily basis. The files in this directory cannot be manipulated by the `crontab` command, but are instead edited by system administrators using a text editor. The commands below restrict read/write and search access to user and group root, preventing regular users from accessing this directory.

Rationale:

Granting write access to this directory for non-privileged users could provide them the means for gaining unauthorized elevated privileges. Granting read access to this directory could give an unprivileged user insight in how to gain elevated privileges or circumvent auditing controls.

Audit:

- IF - cron is installed on the system:
Run the following command and verify `Uid` and `Gid` are both `0/root` and `Access` does not grant permissions to `group` or `other`:

```
# stat -Lc 'Access: (%a/%A) Uid: ( %u/ %U) Gid: ( %g/ %G)' /etc/cron.daily/

Access: (700/drwx------) Uid: ( 0/ root) Gid: ( 0/ root)
```

Remediation:

- IF - cron is installed on the system:
Run the following commands to set ownership and permissions on the `/etc/cron.daily` directory:

```
# chown root:root /etc/cron.daily/
# chmod og-rwx /etc/cron.daily/
```

Default Value:

Access: (755/drwxr-xr-x) Uid: (0/ root) Gid: (0/ root)

References:

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

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
T1053, T1053.003	TA0002, TA0007	M1018

2.4.1.5 Ensure permissions on /etc/cron.weekly are configured (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The `/etc/cron.weekly` directory contains system cron jobs that need to run on a weekly basis. The files in this directory cannot be manipulated by the `crontab` command but are instead edited by system administrators using a text editor. The commands below restrict read/write and search access to user and group root, preventing regular users from accessing this directory.

Rationale:

Granting write access to this directory for non-privileged users could provide them the means for gaining unauthorized elevated privileges. Granting read access to this directory could give an unprivileged user insight in how to gain elevated privileges or circumvent auditing controls.

Audit:

- IF - cron is installed on the system:
Run the following command and verify `Uid` and `Gid` are both `0/root` and `Access` does not grant permissions to `group` or `other`:

```
# stat -Lc 'Access: (%a/%A) Uid: ( %u/ %U) Gid: ( %g/ %G)' /etc/cron.weekly/

Access: (700/drwx------) Uid: ( 0/ root) Gid: ( 0/ root)
```

Remediation:

- IF - cron is installed on the system:
Run the following commands to set ownership and permissions on the `/etc/cron.weekly` directory:

```
# chown root:root /etc/cron.weekly/
# chmod og-rwx /etc/cron.weekly/
```

Default Value:

Access: (755/drwxr-xr-x) Uid: (0/ root) Gid: (0/ root)

References:

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

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
T1053, T1053.003	TA0002, TA0007	M1018

2.4.1.6 Ensure permissions on /etc/cron.monthly are configured (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The `/etc/cron.monthly` directory contains system cron jobs that need to run on a monthly basis. The files in this directory cannot be manipulated by the `crontab` command but are instead edited by system administrators using a text editor. The commands below restrict read/write and search access to user and group root, preventing regular users from accessing this directory.

Rationale:

Granting write access to this directory for non-privileged users could provide them the means for gaining unauthorized elevated privileges. Granting read access to this directory could give an unprivileged user insight in how to gain elevated privileges or circumvent auditing controls.

Audit:

- IF - cron is installed on the system:
Run the following command and verify `Uid` and `Gid` are both `0/root` and `Access` does not grant permissions to `group` or `other`:

```
# stat -Lc 'Access: (%a/%A) Uid: ( %u/ %U) Gid: ( %g/ %G)' /etc/cron.monthly/

Access: (700/drwx------) Uid: ( 0/ root) Gid: ( 0/ root)
```

Remediation:

- IF - cron is installed on the system:
Run the following commands to set ownership and permissions on the `/etc/cron.monthly` directory:

```
# chown root:root /etc/cron.monthly/
# chmod og-rwx /etc/cron.monthly/
```

Default Value:

Access: (755/drwxr-xr-x) Uid: (0/ root) Gid: (0/ root)

References:

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

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
T1053, T1053.003	TA0002, TA0007	M1018

2.4.1.7 Ensure permissions on /etc/cron.d are configured (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The `/etc/cron.d` directory contains system `cron` jobs that need to run in a similar manner to the hourly, daily weekly and monthly jobs from `/etc/crontab`, but require more granular control as to when they run. The files in this directory cannot be manipulated by the `crontab` command but are instead edited by system administrators using a text editor. The commands below restrict read/write and search access to user and group root, preventing regular users from accessing this directory.

Rationale:

Granting write access to this directory for non-privileged users could provide them the means for gaining unauthorized elevated privileges. Granting read access to this directory could give an unprivileged user insight in how to gain elevated privileges or circumvent auditing controls.

Audit:

- IF - cron is installed on the system:
Run the following command and verify `Uid` and `Gid` are both `0/root` and `Access` does not grant permissions to `group` or `other`:

```
# stat -Lc 'Access: (%a/%A) Uid: ( %u/ %U) Gid: ( %g/ %G)' /etc/cron.d/

Access: (700/drwx------) Uid: ( 0/ root) Gid: ( 0/ root)
```

Remediation:

- IF - cron is installed on the system:
Run the following commands to set ownership and permissions on the `/etc/cron.d` directory:

```
# chown root:root /etc/cron.d/
# chmod og-rwx /etc/cron.d/
```

Default Value:

Access: (755/drwxr-xr-x) Uid: (0/ root) Gid: (0/ root)

References:

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

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
T1053, T1053.003	TA0002, TA0007	M1018

2.4.1.8 Ensure crontab is restricted to authorized users (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

`crontab` is the program used to install, deinstall, or list the tables used to drive the cron daemon. Each user can have their own crontab, and though these are files in `/var/spool/cron/crontabs`, they are not intended to be edited directly.

If the `/etc/cron.allow` file exists, then you must be listed (one user per line) therein in order to be allowed to use this command. If the `/etc/cron.allow` file does not exist but the `/etc/cron.deny` file does exist, then you must not be listed in the `/etc/cron.deny` file in order to use this command.

If neither of these files exists, then depending on site-dependent configuration parameters, only the super user will be allowed to use this command, or all users will be able to use this command.

If both files exist then `/etc/cron.allow` takes precedence. Which means that `/etc/cron.deny` is not considered and your user must be listed in `/etc/cron.allow` in order to be able to use the crontab.

Regardless of the existence of any of these files, the root administrative user is always allowed to setup a crontab.

The files `/etc/cron.allow` and `/etc/cron.deny`, if they exist, must be either world-readable, or readable by group `crontab`. If they are not, then cron will deny access to all users until the permissions are fixed.

There is one file for each user's crontab under the `/var/spool/cron/crontabs` directory. Users are not allowed to edit the files under that directory directly to ensure that only users allowed by the system to run periodic tasks can add them, and only syntactically correct crontabs will be written there. This is enforced by having the directory writable only by the `crontab` group and configuring crontab command with the setgid bid set for that specific group.

Note:

- Even though a given user is not listed in `cron.allow`, cron jobs can still be run as that user
- The files `/etc/cron.allow` and `/etc/cron.deny`, if they exist, only controls administrative access to the crontab command for scheduling and modifying cron jobs

Rationale:

On many systems, only the system administrator is authorized to schedule `cron` jobs. Using the `cron.allow` file to control who can run `cron` jobs enforces this policy. It is easier to manage an allow list than a deny list. In a deny list, you could potentially add a user ID to the system and forget to add it to the deny files.

Audit:

- IF - cron is installed on the system:
Run the following command to verify `/etc/cron.allow`:

- Exists
- Is mode `0640` or more restrictive
- Is owned by the user `root`
- Is group owned by the group `root` **- OR -** the group `crontab`

```
# stat -Lc 'Access: (%a/%A) Owner: (%U) Group: (%G)' /etc/cron.allow
```

Verify the returned value is:

```
Access: (640/-rw-r-----) Owner: (root) Group: (root)
- OR -
Access: (640/-rw-r-----) Owner: (root) Group: (crontab)
```

Run the following command to verify either `cron.deny` doesn't exist or is:

- Mode `0640` or more restrictive
- Owned by the user `root`
- Is group owned by the group `root` **- OR -** the group `crontab`

```
# [ -e "/etc/cron.deny" ] && stat -Lc 'Access: (%a/%A) Owner: (%U) Group:
(%G)' /etc/cron.deny
```

Verify either nothing is returned **- OR -** returned value is one of the following:

```
Access: (640/-rw-r-----) Owner: (root) Group: (root)
- OR -
Access: (640/-rw-r-----) Owner: (root) Group: (crontab)
```

Note: On systems where cron is configured to use the group `crontab`, if the group `crontab` is not set as the owner of `cron.allow`, then cron will deny access to all users and you will see an error similar to:

```
You (<USERNAME>) are not allowed to use this program (crontab)
See crontab(1) for more information
```

Remediation:

- IF - cron is installed on the system:
Run the following script to:

- Create /etc/cron.allow if it doesn't exist
- Change owner to user root
- Change group owner to group root **- OR -** group crontab if it exists
- Change mode to 640 or more restrictive

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

{
   [ ! -e "/etc/cron.deny" ] && touch /etc/cron.allow
   chmod u-x,g-wx,o-rwx /etc/cron.allow
   if grep -Pq -- '^\h*crontab\:' /etc/group; then
      chown root:crontab /etc/cron.allow
   else
      chown root:root /etc/cron.allow
   fi
}
```

- IF - /etc/cron.deny exists, run the following script to:

- Change owner to user root
- Change group owner to group root **- OR -** group crontab if it exists
- Change mode to 640 or more restrictive

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

{
   if [ -e "/etc/cron.deny" ]; then
      chmod u-x,g-wx,o-rwx /etc/cron.deny
      if grep -Pq -- '^\h*crontab\:' /etc/group; then
         chown root:crontab /etc/cron.deny
      else
         chown root:root /etc/cron.deny
      fi
   fi
}
```

Note: On systems where cron is configured to use the group crontab, if the group crontab is not set as the owner of cron.allow, then cron will deny access to all users and you will see an error similar to:

```
You (<USERNAME>) are not allowed to use this program (crontab)
See crontab(1) for more information
```

References:

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

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
T1053, T1053.003	TA0002	M1018

2.4.2 Configure at

`at` is a command-line utility used to schedule a job for later execution

Note: if `at` is not installed on the system, this section can be skipped

2.4.2.1 Ensure at is restricted to authorized users (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

`at` allows fairly complex time specifications, extending the POSIX.2 standard. It accepts times of the form HH:MM to run a job at a specific time of day. (If that time is already past, the next day is assumed.) You may also specify midnight, noon, or teatime (4pm) and you can have a time-of-day suffixed with AM or PM for running in the morning or the evening. You can also say what day the job will be run, by giving a date in the form month-name day with an optional year, or giving a date of the form MMDD[CC]YY, MM/DD/[CC]YY, DD.MM.[CC]YY or [CC]YY-MM-DD. The specification of a date must follow the specification of the time of day. You can also give times like now + count time-units, where the time-units can be minutes, hours, days, or weeks and you can tell at to run the job today by suffixing the time with today and to run the job tomorrow by suffixing the time with tomorrow.

The `/etc/at.allow` and `/etc/at.deny` files determine which user can submit commands for later execution via at or batch. The format of the files is a list of usernames, one on each line. Whitespace is not permitted. If the file `/etc/at.allow` exists, only usernames mentioned in it are allowed to use at. If `/etc/at.allow` does not exist, `/etc/at.deny` is checked, every username not mentioned in it is then allowed to use at. An empty `/etc/at.deny` means that every user may use at. If neither file exists, only the superuser is allowed to use at.

Rationale:

On many systems, only the system administrator is authorized to schedule `at` jobs. Using the `at.allow` file to control who can run `at` jobs enforces this policy. It is easier to manage an allow list than a deny list. In a deny list, you could potentially add a user ID to the system and forget to add it to the deny files.

Audit:

- IF - at is installed on the system:
Run the following command to verify `/etc/at.allow`:

- Exists
- Is mode `0640` or more restrictive
- Is owned by the user `root`
- Is group owned by the group `daemon` or group `root`

```
# stat -Lc 'Access: (%a/%A) Owner: (%U) Group: (%G)' /etc/at.allow

Access: (640/-rw-r-----) Owner: (root) Group: (daemon)
-OR-
Access: (640/-rw-r-----) Owner: (root) Group: (root)
```

Verify mode is `640` or more restrictive, owner is `root`, and group is `daemon` or `root`
Run the following command to verify `at.deny` doesn't exist, **-OR-** is:

- Mode `0640` or more restrictive
- Owned by the user `root`
- Group owned by the group `daemon` or group `root`

```
# [ -e "/etc/at.deny" ] && stat -Lc 'Access: (%a/%A) Owner: (%U) Group: (%G)'
/etc/at.deny

Access: (640/-rw-r-----) Owner: (root) Group: (daemon)
-OR-
Access: (640/-rw-r-----) Owner: (root) Group: (root)
-OR-
Nothing is returned
```

If a value is returned, verify mode is 640 or more restrictive, owner is `root`, and group is `daemon` or `root`

Remediation:

- IF - at is installed on the system:
Run the following script to:

- `/etc/at.allow`:
  - Create the file if it doesn't exist
  - Change owner or user `root`
  - If group `daemon` exists, change to group `daemon`, else change group to `root`
  - Change mode to `640` or more restrictive
- **- IF -** `/etc/at.deny` exists:
  - Change owner or user `root`
  - If group `daemon` exists, change to group `daemon`, else change group to `root`
  - Change mode to `640` or more restrictive

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

{
   grep -Pq -- '^daemon\b' /etc/group && l_group="daemon" || l_group="root"
   [ ! -e "/etc/at.allow" ] && touch /etc/at.allow
   chown root:"$l_group" /etc/at.allow
   chmod u-x,g-wx,o-rwx /etc/at.allow
   [ -e "/etc/at.deny" ] && chown root:"$l_group" /etc/at.deny
   [ -e "/etc/at.deny" ] && chmod u-x,g-wx,o-rwx /etc/at.deny
}
```

References:

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

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
T1053, T1053.003	TA0002	M1018