

## 5 Access Control

## 5.1 Configure SSH Server

Secure Shell (SSH) is a secure, encrypted replacement for common login services such as `telnet`, `ftp`, `rlogin`, `rsh`, and `rcp`. It is strongly recommended that sites abandon older clear-text login protocols and use SSH to prevent session hijacking and sniffing of sensitive data off the network.

The recommendations in this section only apply if the SSH daemon is installed on the system, **if remote access is not required the SSH daemon can be removed and this section skipped.**

`sshd_config`:

- The openSSH daemon configuration directives, `Include` and `Match`, may cause the audits in this section's recommendations to report incorrectly. It is recommended that these options only be used if they're needed and fully understood. If these options are configured in accordance with local site policy, they should be accounted for when following the recommendations in this section.
- The default `Include` location is the `/etc/ssh/sshd_config.d` directory. This default has been accounted for in this section. If a file has an additional `Include` that isn't this default location, the files should be reviewed to verify that the recommended setting is not being over-ridden.
- The audits of the running configuration in this section are run in the context of the root user, the local host name, and the local host's IP address. If a `Match` block exists that matches one of these criteria, the output of the audit will be from the match block. The respective matched criteria should be replaced with a non-matching substitution.
- **Include:**
  - Include the specified configuration file(s).
  - Multiple pathnames may be specified and each pathname may contain glob(7) wildcards that will be expanded and processed in lexical order.
  - Files without absolute paths are assumed to be in `/etc/ssh/`.
  - An Include directive may appear inside a Match block to perform conditional inclusion.

- **Match:**
  - Introduces a conditional block. If all of the criteria on the Match line are satisfied, the keywords on the following lines override those set in the global section of the config file, until either another Match line or the end of the file. If a keyword appears in multiple Match blocks that are satisfied, only the first instance of the keyword is applied.
  - The arguments to Match are one or more criteria-pattern pairs or the single token All which matches all criteria. The available criteria are User, Group, Host, LocalAddress, LocalPort, and Address.
  - The match patterns may consist of single entries or comma-separated lists and may use the wildcard and negation operators described in the PATTERNS section of ssh\_config(5).
  - The patterns in an Address criteria may additionally contain addresses to match in CIDR address/masklen format, such as 192.0.2.0/24 or 2001:db8::/32. Note that the mask length provided must be consistent with the address - it is an error to specify a mask length that is too long for the address or one with bits set in this host portion of the address. For example, 192.0.2.0/33 and 192.0.2.0/8, respectively.
  - Only a subset of keywords may be used on the lines following a Match keyword. Available keywords are available in the ssh\_config man page.
- Once all configuration changes have been made to /etc/ssh/sshd\_config or any included configuration files, the sshd configuration must be reloaded

Command to re-load the SSH daemon configuration:

```
# systemctl reload-or-restart sshd
```

sshd command:

- **-T** - Extended test mode. Check the validity of the configuration file, output the effective configuration to stdout and then exit. Optionally, Match rules may be applied by specifying the connection parameters using one or more **-C** options.
- **-C** - connection\_spec. Specify the connection parameters to use for the -T extended test mode. If provided, any Match directives in the configuration file that would apply are applied before the configuration is written to standard output. The connection parameters are supplied as keyword=value pairs and may be supplied in any order, either with multiple **-C** options or as a comma-separated list. The keywords are addr, user, host, laddr, lport, and rdomain and correspond to source address, user, resolved source host name, local address, local port number and routing domain respectively.

### *5.1.1 Ensure permissions on /etc/ssh/sshd\_config are configured (Automated)*

#### **Profile Applicability:**

- Level 1 - Server
- Level 1 - Workstation

#### **Description:**

The file `/etc/ssh/sshd_config`, and files ending in `.conf` in the `/etc/ssh/sshd_config.d` directory, contain configuration specifications for `sshd`.

#### **Rationale:**

configuration specifications for `sshd` need to be protected from unauthorized changes by non-privileged users.

## Audit:

Run the following script and verify `/etc/ssh/sshd_config` and files ending in `.conf` in the `/etc/ssh/sshd_config.d` directory are:

- Mode `0600` or more restrictive
- Owned by the `root` user
- Group owned by the group `root`.

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

{
  a_output=(); a_output2=()
  perm_mask='0177' && maxperm="$( printf '%o' $(( 0777 & ~$perm_mask )) )"
  f_sshd_files_chk()
  {
    while IFS=: read -r l_mode l_user l_group; do
      a_out2=()
      [ "$l_mode & $perm_mask" ] -gt 0 ] && a_out2+=("    Is mode:
\"$l_mode\" \" \"
      "    should be mode: \"$maxperm\" or more restrictive")
      [ "$l_user" != "root" ] && a_out2+=("    Is owned by \"$l_user\"
should be owned by \"root\"")
      [ "$l_group" != "root" ] && a_out2+=("    Is group owned by
\"$l_user\" should be group owned by \"root\"")
      if [ "${#a_out2[@]}" -gt 0 ]; then
        a_output2+=(" - File: \"$l_file\":" "${a_out2[@]}")
      else
        a_output+=(" - File: \"$l_file\":" "    Correct: mode ($l_mode),
owner ($l_user) \" \"
        "    and group owner ($l_group) configured")
      fi
      done <<(stat -Lc '%a:%U:%G' "$l_file")
    }
    [ -e "/etc/ssh/sshd_config" ] && l_file="/etc/ssh/sshd_config" &&
f_sshd_files_chk
    while IFS= read -r -d $'\0' l_file; do
      [ -e "$l_file" ] && f_sshd_files_chk
      done <<(find /etc/ssh/sshd_config.d -type f -name '*.conf' \( -perm /077
-o ! -user root -o ! -group root \) -print0 2>/dev/null)
      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_output2[@]}" -gt 0 ] && printf '%s\n' "" "- Correctly set:"
"${a_output[@]}" ""
      fi
    }
  }
}
```

- IF - other locations are listed in an `Include` statement, `*.conf` files in these locations should also be checked.

## Remediation:

Run the following script to set ownership and permissions on `/etc/ssh/sshd_config` and files ending in `.conf` in the `/etc/ssh/sshd_config.d` directory:

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







{
  chmod u-x,og-rwx /etc/ssh/sshd_config
  chown root:root /etc/ssh/sshd_config
  while IFS= read -r -d $'\0' l_file; do
    if [ -e "$l_file" ]; then
      chmod u-x,og-rwx "$l_file"
      chown root:root "$l_file"
    fi
  done < <(find /etc/ssh/sshd_config.d -type f -print0 2>/dev/null)
}
```

- **IF** - other locations are listed in an **Include** statement, `*.conf` files in these locations access should also be modified.

## References:

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

## CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>3.3 <u>Configure Data Access Control Lists</u></b> 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	<b>14.6 <u>Protect Information through Access Control Lists</u></b> 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
T1098, T1098.004, T1543, T1543.002	TA0005	M1022

### *5.1.2 Ensure permissions on SSH private host key files are configured (Automated)*

#### **Profile Applicability:**

- Level 1 - Server
- Level 1 - Workstation

#### **Description:**

An SSH private key is one of two files used in SSH public key authentication. In this authentication method, the possession of the private key is proof of identity. Only a private key that corresponds to a public key will be able to authenticate successfully. The private keys need to be stored and handled carefully, and no copies of the private key should be distributed.

#### **Rationale:**

If an unauthorized user obtains the private SSH host key file, the host could be impersonated

## Audit:

Run the following script to verify SSH private host key files are owned by the root user and either:

- owned by the group root and mode **0600** or more restrictive

- OR -

- owned by the group designated to own openSSH private keys and mode **0640** or more restrictive

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

{
  a_output=(); a_output2=()
  l_ssh_group_name="$(awk -F: '($1 ~ /^(ssh_keys|_?ssh)$/) {print $1}' /etc/group)"
  f_file_chk()
  {
    while IFS= read -r l_file_mode l_file_owner l_file_group; do
      a_out2=()
      [ "$l_file_group" = "$l_ssh_group_name" ] && l_pmask="0137" || l_pmask="0177"
      l_maxperm="$( printf '%o' $(( 0777 & ~$l_pmask )) )"
      if [ $(( $l_file_mode & $l_pmask )) -gt 0 ]; then
        a_out2+=("    Mode: \"$l_file_mode\" should be mode: \"$l_maxperm\" or
more restrictive")
      fi
      if [ "$l_file_owner" != "root" ]; then
        a_out2+=("    Owned by: \"$l_file_owner\" should be owned by \"root\"")
      fi
      if [[ ! "$l_file_group" =~ ($l_ssh_group_name|root) ]]; then
        a_out2+=("    Owned by group \"$l_file_group\" should be group owned by:
\"$l_ssh_group_name\" or \"root\"")
      fi
      if [ "${#a_out2[@]}" -gt 0 ]; then
        a_output2+=(" - File: \"$l_file\"${a_out2[@]}")
      else
        a_output+=(" - File: \"$l_file\" \" \"
        \"    Correct: mode: \"$l_file_mode\", owner: \"$l_file_owner\" and group
owner: \"$l_file_group\" configured")
      fi
      done <<(stat -Lc '%a:%U:%G' "$l_file")
    }
    while IFS= read -r -d $'\0' l_file; do
      if ssh-keygen -lf &>/dev/null "$l_file"; then
        file "$l_file" | grep -Piq -- '\bopenssh\b+([\#\n\r]+\b+)?private\b+key\b' &&
f_file_chk
      fi
      done <<(find -L /etc/ssh -xdev -type f -print0 2>/dev/null)
      if [ "${#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[@]}"
        [ "${#a_output2[@]}" -gt 0 ] && printf '%s\n' "" "- Correctly set:"
"${a_output[@]}" ""
      fi
    }
  }
}
```



## Remediation:

Run the following script to set mode, ownership, and group on the private SSH host key files:







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

{
  a_output=(); a_output2=(); l_ssh_group_name="$(awk -F: '($1 ~ /^(ssh_keys|_?ssh)$/) {print $1}' /etc/group)"
  f_file_access_fix()
  {
    while IFS=: read -r l_file_mode l_file_owner l_file_group; do
      a_out2=()
      [ "$l_file_group" = "$l_ssh_group_name" ] && l_pmask="0137" || l_pmask="0177"
      l_maxperm="$( printf '%o' $(( 0777 & ~$l_pmask )) )"
      if [ $(( $l_file_mode & $l_pmask )) -gt 0 ]; then
        a_out2+=("      Mode: \"$l_file_mode\" should be mode: \"$l_maxperm\" or
more restrictive" \
"      updating to mode: \:$l_maxperm\"")
        if [ "l_file_group" = "$l_ssh_group_name" ]; then
          chmod u-x,g-wx,o-rwx "$l_file"
        else
          chmod u-x,go-rwx "$l_file"
        fi
      fi
      if [ "$l_file_owner" != "root" ]; then
        a_out2+=("      Owned by: \"$l_file_owner\" should be owned by \"root\" \" \" \
"      Changing ownership to \"root\" \" \" )
        chown root "$l_file"
      fi
      if [[ ! "$l_file_group" =~ ($l_ssh_group_name|root) ]]; then
        [ -n "$l_ssh_group_name" ] && l_new_group="$l_ssh_group_name" ||
l_new_group="root"
        a_out2+=("      Owned by group \"$l_file_group\" should be group owned by:
\"$l_ssh_group_name\" or \"root\" \" \" \
"      Changing group ownership to \"$l_new_group\" \" \" )
        chgrp "$l_new_group" "$l_file"
      fi
      if [ "${#a_out2[@]}" -gt 0 ]; then
        a_output2+=(" - File: \"$l_file\" \" \" \"${a_out2[@]}\"")
      else
        a_output+=(" - File: \"$l_file\" \" \" \
"Correct: mode: \"$l_file_mode\", owner: \"$l_file_owner\", and group
owner: \"$l_file_group\" configured")
      fi
      done <<(stat -Lc '%a:%U:%G' "$l_file")
    }
    while IFS= read -r -d $'\0' l_file; do
      if ssh-keygen -lf &>/dev/null "$l_file"; then
        file "$l_file" | grep -Piq -- '\bopenssh\b+([^\n\r]+\h+)?private\b+hkey\b' &&
f_file_access_fix
      fi
      done <<(find -L /etc/ssh -xdev -type f -print0 2>/dev/null)
      if [ "${#a_output2[@]}" -le 0 ]; then
        printf '%s\n' " " - No access changes required " "
      else
        printf '%s\n' " " - Remediation results: " "${a_output2[@]} " "
      fi
    fi
  }
}
```

## References:

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

## CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>3.3 <u>Configure Data Access Control Lists</u></b> 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	<b>14.6 <u>Protect Information through Access Control Lists</u></b> 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
T1552, T1552.004	TA0003, TA0006	M1022

### *5.1.3 Ensure permissions on SSH public host key files are configured (Automated)*

#### **Profile Applicability:**

- Level 1 - Server
- Level 1 - Workstation

#### **Description:**

An SSH public key is one of two files used in SSH public key authentication. In this authentication method, a public key is a key that can be used for verifying digital signatures generated using a corresponding private key. Only a public key that corresponds to a private key will be able to authenticate successfully.

#### **Rationale:**

If a public host key file is modified by an unauthorized user, the SSH service may be compromised.

#### **Audit:**

Run the following command and verify Access does not grant write or execute permissions to group or other for all returned files:

Run the following script to verify SSH public host key files are mode **0644** or more restrictive, owned by the **root** user, and owned by the **root** group:

```

#!/usr/bin/env bash

{
    a_output=(); a_output2=()
    l_pmask="0133"; l_maxperm="$( printf '%o' $(( 0777 & ~$l_pmask )) )"
    f_file_chk()
    {
        while IFS= read -r l_file_mode l_file_owner l_file_group; do
            a_out2=()
            if [ $(( $l_file_mode & $l_pmask )) -gt 0 ]; then
                a_out2+=("    Mode: \"$l_file_mode\" should be mode:
\"$l_maxperm\" or more restrictive")
            fi
            if [ "$l_file_owner" != "root" ]; then
                a_out2+=("    Owned by: \"$l_file_owner\" should be owned by:
\"root\"")
            fi
            if [ "$l_file_group" != "root" ]; then
                a_out2+=("    Owned by group \"$l_file_group\" should be group
owned by group: \"root\"")
            fi
            if [ "${#a_out2[@]}" -gt 0 ]; then
                a_output2+=(" - File: \"$l_file\" \"${a_out2[@]}")
            else
                a_output+=(" - File: \"$l_file\" \
                "    Correct: mode: \"$l_file_mode\", owner: \"$l_file_owner\"
and group owner: \"$l_file_group\" configured")
            fi
            done < <(stat -Lc '%a:%U:%G' "$l_file")
        }
        while IFS= read -r -d $'\0' l_file; do
            if ssh-keygen -lf &>/dev/null "$l_file"; then
                file "$l_file" | grep -Piq --
'\bopenssh\b+([\#\n\r]+\b+)?public\b+key\b' && f_file_chk
            fi
            done < <(find -L /etc/ssh -xdev -type f -print0 2>/dev/null)
            if [ "${#a_output2[@]}" -le 0 ]; then
                [ "${#a_output[@]}" -le 0 ] && a_output+=(" - No openSSH public keys
found")
                printf '%s\n' "" "- Audit Result:" " ** PASS **" "${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:

Run the following script to set mode, ownership, and group on the public SSH host key files:

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

{
  a_output=(); a_output2=()
  l_mask="0133"; l_maxperm="$( printf '%o' $(( 0777 & ~$l_mask )) )"
  f_file_access_fix()
  {
    while IFS=: read -r l_file_mode l_file_owner l_file_group; do
      a_out2=()
      [ $(( $l_file_mode & $l_mask )) -gt 0 ] && \
        a_out2+=("    Mode: \"$l_file_mode\" should be mode: \
\"$l_maxperm\" or more restrictive" \
"    updating to mode: \"$l_maxperm\"") && chmod u-x,go-wx
"$l_file"
      [ "$l_file_owner" != "root" ] && \
        a_out2+=("    Owned by: \"$l_file_owner\" should be owned by \
\"root\"" \
"    Changing ownership to \"root\"") && chown root "$l_file"
      [ "$l_file_group" != "root" ] && \
        a_out2+=("    Owned by group \"$l_file_group\" should be group \
owned by: \"root\"" \
"    Changing group ownership to \"root\"") && chgrp root
"$l_file"
      if [ "${#a_out2[@]}" -gt 0 ]; then
        a_output2+=(" - File: \"$l_file\" \"${a_out2[@]}")
      else
        a_output+=(" - File: \"$l_file\" \
"    Correct: mode: \"$l_file_mode\", owner: \"$l_file_owner\", \
and group owner: \"$l_file_group\" configured")
      fi
      done < <(stat -Lc '%a:%U:%G' "$l_file")
    }
    while IFS= read -r -d $'\0' l_file; do
      if ssh-keygen -lf &>/dev/null "$l_file"; then
        file "$l_file" | grep -Piq --
'\bopenssh\b+([\^#\n\r]+\b)?public\b+key\b' && f_file_access_fix
      fi
      done < <(find -L /etc/ssh -xdev -type f -print0 2>/dev/null)
      if [ "${#a_output2[@]}" -le 0 ]; then
        printf '%s\n' "" " - No access changes required" ""
      else
        printf '%s\n' " - Remediation results:" "${a_output2[@]}" ""
      fi
    fi
  }
}
```







## Default Value:

644 0/root 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	<b>3.3 <u>Configure Data Access Control Lists</u></b> 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	<b>14.6 <u>Protect Information through Access Control Lists</u></b> 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
T1557, T1557.000	TA0003, TA0006	M1022

### 5.1.4 Ensure sshd access is configured (Automated)

#### Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

#### Description:

There are several options available to limit which users and group can access the system via SSH. It is recommended that at least one of the following options be leveraged:

- **AllowUsers:**
  - The **AllowUsers** variable gives the system administrator the option of allowing specific users to **ssh** into the system. The list consists of space separated user names. Numeric user IDs are not recognized with this variable. If a system administrator wants to restrict user access further by only allowing the allowed users to log in from a particular host, the entry can be specified in the form of user@host.
- **AllowGroups:**
  - The **AllowGroups** variable gives the system administrator the option of allowing specific groups of users to **ssh** into the system. The list consists of space separated group names. Numeric group IDs are not recognized with this variable.
- **DenyUsers:**
  - The **DenyUsers** variable gives the system administrator the option of denying specific users to **ssh** into the system. The list consists of space separated user names. Numeric user IDs are not recognized with this variable. If a system administrator wants to restrict user access further by specifically denying a user's access from a particular host, the entry can be specified in the form of user@host.
- **DenyGroups:**
  - The **DenyGroups** variable gives the system administrator the option of denying specific groups of users to **ssh** into the system. The list consists of space separated group names. Numeric group IDs are not recognized with this variable.

#### Rationale:

Restricting which users can remotely access the system via SSH will help ensure that only authorized users access the system.

## Audit:

Run the following command and verify the output:

```
# sshd -T | grep -Pi -- '^h*(allow|deny) (users|groups) \h+\H+'
```

Verify that the output matches at least one of the following lines:

```
allowusers <userlist>
-OR-
allowgroups <grouplist>
-OR-
denyusers <userlist>
-OR-
denygroups <grouplist>
```

Review the list(s) to ensure included users and/or groups follow local site policy  
- **IF** - **Match** set statements are used in your environment, specify the connection parameters to use for the **-T** extended test mode and run the audit to verify the setting is not incorrectly configured in a match block

*Example additional audit needed for a match block for the user **sshuser**:*

```
# sshd -T -C user=sshuser | grep -Pi --
'^h*(allow|deny) (users|groups) \h+\H+'
```

**Note:** If provided, any Match directives in the configuration file that would apply are applied before the configuration is written to standard output. The connection parameters are supplied as keyword=value pairs and may be supplied in any order, either with multiple **-C** options or as a comma-separated list. The keywords are **addr** (source address), **user** (user), **host** (resolved source host name), **laddr** (local address), **lport** (local port number), and **rdomain** (routing domain).

## Remediation:

Edit the **/etc/ssh/sshd\_config** file to set one or more of the parameters above any **Include** and **Match** set statements as follows:

```
AllowUsers <userlist>
- AND/OR -
AllowGroups <grouplist>
```

## Note:

- First occurrence of a option takes precedence, **Match** set statements withstanding. If **Include** locations are enabled, used, and order of precedence is understood in your environment, the entry may be created in a **.conf** file in an Include directory.
- **Be advised** that these options are "ANDed" together. If both **AllowUsers** and **AllowGroups** are set, connections will be limited to the list of users that are also a member of an allowed group. It is recommended that only one be set for clarity and ease of administration.
- It is easier to manage an allow list than a deny list. In a deny list, you could potentially add a user or group and forget to add it to the deny list.









**Default Value:**

None

**References:**

1. SSHD\_CONFIG(5)
2. NIST SP 800-53 Rev. 5: AC-3. MP-2
3. SSHD(8)

**CIS Controls:**

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>3.3 <u>Configure Data Access Control Lists</u></b> 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	<b>4.3 <u>Ensure the Use of Dedicated Administrative Accounts</u></b> Ensure that all users with administrative account access use a dedicated or secondary account for elevated activities. This account should only be used for administrative activities and not internet browsing, email, or similar activities.			

**MITRE ATT&CK Mappings:**

Techniques / Sub-techniques	Tactics	Mitigations
T1021, T1021.004	TA0008	M1018

### *5.1.5 Ensure sshd Banner is configured (Automated)*

**Profile Applicability:**

- Level 1 - Server
- Level 1 - Workstation

**Description:**

The **Banner** parameter specifies a file whose contents must be sent to the remote user before authentication is permitted. By default, no banner is displayed.

**Rationale:**

Banners are used to warn connecting users of the particular site's policy regarding connection. Presenting a warning message prior to the normal user login may assist the prosecution of trespassers on the computer system.

## Audit:

Run the following command to verify **Banner** is set:

```
# sshd -T | grep -Pi -- '^banner\h+\//\H+'
```

*Example:*

```
banner /etc/issue.net
```

- **IF - Match** set statements are used in your environment, specify the connection parameters to use for the **-T** extended test mode and run the audit to verify the setting is not incorrectly configured in a match block

*Example additional audit needed for a match block for the user **sshuser**:*

```
# sshd -T -C user=sshuser | grep -Pi -- '^banner\h+\//\H+'
```

**Note:** If provided, any Match directives in the configuration file that would apply are applied before the configuration is written to standard output. The connection parameters are supplied as keyword=value pairs and may be supplied in any order, either with multiple **-C** options or as a comma-separated list. The keywords are **addr** (source address), **user** (user), **host** (resolved source host name), **laddr** (local address), **lport** (local port number), and **rdomain** (routing domain).

Run the following command and verify that the contents or the file being called by the **Banner** argument match site policy:

```
# [ -e "$(sshd -T | awk '$1 == "banner" {print $2}')" ] && cat "$(sshd -T | awk '$1 == "banner" {print $2}')"
```

Run the following command and verify no results are returned:

```
# grep -Psi -- "(\v|\r|\m|\s\b$(grep '^ID=' /etc/os-release | cut -d= -f2 | sed -e 's//g')\b)" "$(sshd -T | awk '$1 == "banner" {print $2}')"
```

## Remediation:

Edit the `/etc/ssh/sshd_config` file to set the **Banner** parameter above any **Include** and **Match** entries as follows:

```
Banner /etc/issue.net
```

**Note:** First occurrence of a option takes precedence, Match set statements withstanding. If Include locations are enabled, used, and order of precedence is understood in your environment, the entry may be created in a file in Include location. Edit the file being called by the **Banner** argument with the appropriate contents according to your site policy, remove any instances of `\m`, `\r`, `\s`, `\v` or references to the **OS platform**

*Example:*

```
# printf '%s\n' "Authorized users only. All activity may be monitored and  
reported." > "$(sshd -T | awk '$1 == "banner" {print $2}')
```

## 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
	TA0001, TA0007	M1035

## 5.1.6 Ensure sshd Ciphers are configured (Automated)

### Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

### Description:

This variable limits the ciphers that SSH can use during communication.

### Notes:

- Some organizations may have stricter requirements for approved ciphers.
- Ensure that ciphers used are in compliance with site policy.
- The only "strong" ciphers currently FIPS 140 compliant are:
  - [aes256-gcm@openssh.com](#)
  - [aes128-gcm@openssh.com](#)
  - aes256-ctr
  - aes192-ctr
  - aes128-ctr

### Rationale:

Weak ciphers that are used for authentication to the cryptographic module cannot be relied upon to provide confidentiality or integrity, and system data may be compromised.

- The Triple DES ciphers, as used in SSH, have a birthday bound of approximately four billion blocks, which makes it easier for remote attackers to obtain clear text data via a birthday attack against a long-duration encrypted session, aka a "Sweet32" attack.
- Error handling in the SSH protocol; Client and Server, when using a block cipher algorithm in Cipher Block Chaining (CBC) mode, makes it easier for remote attackers to recover certain plain text data from an arbitrary block of cipher text in an SSH session via unknown vectors.

## Audit:

Run the following command to verify none of the "weak" ciphers are being used:

```
# sshd -T | grep -Pi --  
'^ciphers\h+\\"?([^\n\r]+,)?((3des|blowfish|cast128|aes(128|192|256))-  
cbc|arcfour(128|256)?|rijndael-cbc@lysator.liu.se|chacha20-  
poly1305@openssh.com)\b'
```

- **IF** - a line is returned, review the list of ciphers. If the line includes **chacha20-poly1305@openssh.com**, review **CVE-2023-48795** and verify the system has been patched. No ciphers in the list below should be returned as they're considered "weak":

```
3des-cbc  
aes128-cbc  
aes192-cbc  
aes256-cbc
```

## Remediation:

Edit the `/etc/ssh/sshd_config` file and add/modify the **Ciphers** line to contain a comma separated list of the site unapproved (weak) Ciphers preceded with a **-** above any **Include** entries:

*Example:*

```
Ciphers -3des-cbc,aes128-cbc,aes192-cbc,aes256-cbc,chacha20-  
poly1305@openssh.com
```

- **IF** - **CVE-2023-48795** has been addressed, and it meets local site policy, **chacha20-poly1305@openssh.com** may be removed from the list of excluded ciphers.

**Note:** First occurrence of an option takes precedence. If Include locations are enabled, used, and order of precedence is understood in your environment, the entry may be created in a file in Include location.





## Default Value:

Ciphers [chacha20-poly1305@openssh.com](#),aes128-ctr,aes192-ctr,aes256-ctr,[aes128-gcm@openssh.com](#),[aes256-gcm@openssh.com](#)

## References:

1. <https://nvd.nist.gov/vuln/detail/CVE-2023-48795>
2. <https://nvd.nist.gov/vuln/detail/CVE-2019-1543>
3. <https://nvd.nist.gov/vuln/detail/CVE-2016-2183>
4. <https://nvd.nist.gov/vuln/detail/CVE-2008-5161>
5. <https://www.openssh.com/txt/cbc.adv>
6. <https://www.openssh.com/txt/cbc.adv>
7. SSHD\_CONFIG(5)
8. NIST SP 800-53 Rev. 5: SC-8

**CIS Controls:**

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

**MITRE ATT&CK Mappings:**

Techniques / Sub-techniques	Tactics	Mitigations
T1040, T1040.000, T1557	TA0006	M1041

### 5.1.7 Ensure sshd ClientAliveInterval and ClientAliveCountMax are configured (Automated)

#### Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

#### Description:

**Note:** To clarify, the two settings described below are only meant for idle connections from a protocol perspective and are not meant to check if the user is active or not. An idle user does not mean an idle connection. SSH does not and never had, intentionally, the capability to drop idle users. In SSH versions before 8.2p1 there was a bug that caused these values to behave in such a manner that they were abused to disconnect idle users. This bug has been resolved in 8.2p1 and thus it can no longer be abused to disconnect idle users.

The two options `ClientAliveInterval` and `ClientAliveCountMax` control the timeout of SSH sessions. Taken directly from `man 5 sshd_config`:

- `ClientAliveInterval` Sets a timeout interval in seconds after which if no data has been received from the client, sshd(8) will send a message through the encrypted channel to request a response from the client. The default is 0, indicating that these messages will not be sent to the client.
- `ClientAliveCountMax` Sets the number of client alive messages which may be sent without sshd(8) receiving any messages back from the client. If this threshold is reached while client alive messages are being sent, sshd will disconnect the client, terminating the session. It is important to note that the use of client alive messages is very different from TCPKeepAlive. The client alive messages are sent through the encrypted channel and therefore will not be spoofable. The TCP keepalive option enabled by TCPKeepAlive is spoofable. The client alive mechanism is valuable when the client or server depend on knowing when a connection has become unresponsive. The default value is 3. If `ClientAliveInterval` is set to 15, and `ClientAliveCountMax` is left at the default, unresponsive SSH clients will be disconnected after approximately 45 seconds. Setting a zero `ClientAliveCountMax` disables connection termination.



## Rationale:

In order to prevent resource exhaustion, appropriate values should be set for both `ClientAliveInterval` and `ClientAliveCountMax`. Specifically, looking at the source code, `ClientAliveCountMax` must be greater than zero in order to utilize the ability of SSH to drop idle connections. If connections are allowed to stay open indefinitely, this can potentially be used as a DDOS attack or simple resource exhaustion could occur over unreliable networks.

The example set here is a 45 second timeout. Consult your site policy for network timeouts and apply as appropriate.

## Audit:

Run the following command and verify `ClientAliveInterval` and `ClientAliveCountMax` are greater than zero:

```
# sshd -T | grep -Pi -- '(clientaliveinterval|clientalivecountmax)'
```

*Example Output:*

```
clientaliveinterval 15
clientalivecountmax 3
```

- IF - **Match** set statements are used in your environment, specify the connection parameters to use for the **-T** extended test mode and run the audit to verify the setting is not incorrectly configured in a match block

*Example additional audit needed for a match block for the user `sshuser`:*

```
# sshd -T -C user=sshuser | grep -Pi --
'(clientaliveinterval|clientalivecountmax)'
```

**Note:** If provided, any Match directives in the configuration file that would apply are applied before the configuration is written to standard output. The connection parameters are supplied as keyword=value pairs and may be supplied in any order, either with multiple **-C** options or as a comma-separated list. The keywords are **addr** (source address), **user** (user), **host** (resolved source host name), **laddr** (local address), **lport** (local port number), and **rdomain** (routing domain).

## Remediation:

Edit the `/etc/ssh/sshd_config` file to set the `ClientAliveInterval` and `ClientAliveCountMax` parameters above any **Include** and **Match** entries according to site policy.

*Example:*

```
ClientAliveInterval 15
ClientAliveCountMax 3
```

**Note:** First occurrence of an option takes precedence, Match set statements withstanding. If Include locations are enabled, used, and order of precedence is understood in your environment, the entry may be created in a file in Include location.

**Default Value:**

ClientAliveInterval 0

ClientAliveCountMax 3

**References:**

1. SSHD\_CONFIG(5)
2. SSHD(8)
3. NIST SP 800-53 Rev. 5: CM-1, CM-2, CM-6, CM-7, IA-5

**Additional Information:**

[https://bugzilla.redhat.com/show\\_bug.cgi?id=1873547](https://bugzilla.redhat.com/show_bug.cgi?id=1873547)

[https://github.com/openssh/openssh-portable/blob/V\\_8\\_9/serverloop.c#L137](https://github.com/openssh/openssh-portable/blob/V_8_9/serverloop.c#L137)

**MITRE ATT&CK Mappings:**

Techniques / Sub-techniques	Tactics	Mitigations
T1078, T1078.001, T1078.002, T1078.003	TA0001	M1026

## 5.1.8 Ensure sshd DisableForwarding is enabled (Automated)

### Profile Applicability:

- Level 1 - Workstation
- Level 2 - Server

### Description:

The **DisableForwarding** parameter disables all forwarding features, including X11, ssh-agent(1), TCP and StreamLocal. This option overrides all other forwarding-related options and may simplify restricted configurations.

- X11Forwarding provides the ability to tunnel X11 traffic through the connection to enable remote graphic connections.
- ssh-agent is a program to hold private keys used for public key authentication. Through use of environment variables the agent can be located and automatically used for authentication when logging in to other machines using ssh.
- SSH port forwarding is a mechanism in SSH for tunneling application ports from the client to the server, or servers to clients. It can be used for adding encryption to legacy applications, going through firewalls, and some system administrators and IT professionals use it for opening backdoors into the internal network from their home machines.

### Rationale:

Disable X11 forwarding unless there is an operational requirement to use X11 applications directly. There is a small risk that the remote X11 servers of users who are logged in via SSH with X11 forwarding could be compromised by other users on the X11 server. Note that even if X11 forwarding is disabled, users can always install their own forwarders.

anyone with root privilege on the the intermediate server can make free use of ssh-agent to authenticate them to other servers

Leaving port forwarding enabled can expose the organization to security risks and backdoors. SSH connections are protected with strong encryption. This makes their contents invisible to most deployed network monitoring and traffic filtering solutions. This invisibility carries considerable risk potential if it is used for malicious purposes such as data exfiltration. Cybercriminals or malware could exploit SSH to hide their unauthorized communications, or to exfiltrate stolen data from the target network.

## Impact:

SSH tunnels are widely used in many corporate environments. In some environments the applications themselves may have very limited native support for security. By utilizing tunneling, compliance with SOX, HIPAA, PCI-DSS, and other standards can be achieved without having to modify the applications.

## Audit:

Run the following command to verify **DisableForwarding** is set to **yes**:

```
# sshd -T | grep -i disableforwarding  
disableforwarding yes
```

## Remediation:

Edit the **/etc/ssh/sshd\_config** file to set the **DisableForwarding** parameter to **yes** above any **Include** entry as follows:





```
DisableForwarding yes
```

**Note:** First occurrence of a option takes precedence. If Include locations are enabled, used, and order of precedence is understood in your environment, the entry may be created in a file in Include location.

## References:

1. sshd\_config(5)
2. SSHD(8)
3. NIST SP 800-53 Rev. 5: CM-7

## CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>4.8 <u>Uninstall or Disable Unnecessary Services on Enterprise Assets and Software</u></b> Uninstall or disable unnecessary services on enterprise assets and software, such as an unused file sharing service, web application module, or service function.			
v7	<b>9.2 <u>Ensure Only Approved Ports, Protocols and Services Are Running</u></b> 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:**

<b>Techniques / Sub-techniques</b>	<b>Tactics</b>	<b>Mitigations</b>
T1210, T1210.000	TA0008	M1042

### 5.1.9 Ensure sshd GSSAPIAuthentication is disabled (Automated)

#### Profile Applicability:

- Level 1 - Workstation
- Level 2 - Server

#### Description:

The **GSSAPIAuthentication** parameter specifies whether user authentication based on GSSAPI is allowed

#### Rationale:

Allowing GSSAPI authentication through SSH exposes the system's GSSAPI to remote hosts, and should be disabled to reduce the attack surface of the system

#### Audit:

Run the following command to verify **GSSAPIAuthentication** is set to **no**:

```
# sshd -T | grep gssapiauthentication  
gssapiauthentication no
```

- **IF** - **Match** set statements are used in your environment, specify the connection parameters to use for the **-T** extended test mode and run the audit to verify the setting is not incorrectly configured in a match block

*Example additional audit needed for a match block for the user **sshuser**:*

```
# sshd -T -C user=sshuser | grep gssapiauthentication
```

**Note:** If provided, any Match directives in the configuration file that would apply are applied before the configuration is written to standard output. The connection parameters are supplied as keyword=value pairs and may be supplied in any order, either with multiple **-C** options or as a comma-separated list. The keywords are **addr** (source address), **user** (user), **host** (resolved source host name), **laddr** (local address), **lport** (local port number), and **rdomain** (routing domain)

#### Remediation:

Edit the **/etc/ssh/sshd\_config** file to set the **GSSAPIAuthentication** parameter to **no** above any **Include** and **Match** entries as follows:

```
GSSAPIAuthentication no
```

**Note:** First occurrence of an option takes precedence, **Match** set statements withstanding. If **Include** locations are enabled, used, and order of precedence is understood in your environment, the entry may be created in a file in **Include** location.






**Default Value:**

GSSAPIAuthentication no

**References:**

1. SSHD\_CONFIG(5)
2. SSHD(8)
3. NIST SP 800-53 Rev. 5: CM-1, CM-2, CM-6, CM-7, IA-5

**CIS Controls:**

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>5.2 Use Unique Passwords</b> Use unique passwords for all enterprise assets. Best practice implementation includes, at a minimum, an 8-character password for accounts using MFA and a 14-character password for accounts not using MFA.			
v7	<b>4.4 Use Unique Passwords</b> Where multi-factor authentication is not supported (such as local administrator, root, or service accounts), accounts will use passwords that are unique to that system.			

**MITRE ATT&CK Mappings:**

Techniques / Sub-techniques	Tactics	Mitigations
T1078, T1078.001, T1078.003	TA0001	M1042

## 5.1.10 Ensure sshd HostbasedAuthentication is disabled (Automated)

### Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

### Description:

The **HostbasedAuthentication** parameter specifies if authentication is allowed through trusted hosts via the user of **.rhosts**, or **/etc/hosts.equiv**, along with successful public key client host authentication.

### Rationale:

Even though the **.rhosts** files are ineffective if support is disabled in **/etc/pam.conf**, disabling the ability to use **.rhosts** files in SSH provides an additional layer of protection.

### Audit:

Run the following command to verify **HostbasedAuthentication** is set to **no**:

```
# sshd -T | grep hostbasedauthentication  
hostbasedauthentication no
```

- **IF - Match** set statements are used in your environment, specify the connection parameters to use for the **-T** extended test mode and run the audit to verify the setting is not incorrectly configured in a match block

*Example additional audit needed for a match block for the user **sshuser**:*

```
# sshd -T -C user=sshuser | grep hostbasedauthentication
```

**Note:** If provided, any Match directives in the configuration file that would apply are applied before the configuration is written to standard output. The connection parameters are supplied as keyword=value pairs and may be supplied in any order, either with multiple **-C** options or as a comma-separated list. The keywords are **addr** (source address), **user** (user), **host** (resolved source host name), **laddr** (local address), **lport** (local port number), and **rdomain** (routing domain)



## Remediation:

Edit the `/etc/ssh/sshd_config` file to set the `HostbasedAuthentication` parameter to `no` above any `Include` and `Match` entries as follows:

```
HostbasedAuthentication no
```

**Note:** First occurrence of a option takes precedence, `Match` set statements withstanding. If `Include` locations are enabled, used, and order of precedence is understood in your environment, the entry may be created in a file in `Include` location.

## Default Value:

HostbasedAuthentication no

## References:

1. SSHD\_CONFIG(5)
2. SSHD(8)
3. 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
T1078, T1078.001, T1078.003	TA0001	M1042

### 5.1.11 Ensure sshd IgnoreRhosts is enabled (Automated)

#### Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

#### Description:

The **IgnoreRhosts** parameter specifies that **.rhosts** and **.shosts** files will not be used in **RhostsRSAAuthentication** or **HostbasedAuthentication**.

#### Rationale:

Setting this parameter forces users to enter a password when authenticating with SSH.

#### Audit:

Run the following command to verify **IgnoreRhosts** is set to **yes**:

```
# sshd -T | grep ignorerhosts  
ignorerhosts yes
```

- **IF** - **Match** set statements are used in your environment, specify the connection parameters to use for the **-T** extended test mode and run the audit to verify the setting is not incorrectly configured in a match block

*Example additional audit needed for a match block for the user **sshuser**:*

```
# sshd -T -C user=sshuser | grep ignorerhosts
```

**Note:** If provided, any Match directives in the configuration file that would apply are applied before the configuration is written to standard output. The connection parameters are supplied as keyword=value pairs and may be supplied in any order, either with multiple **-C** options or as a comma-separated list. The keywords are **addr** (source address), **user** (user), **host** (resolved source host name), **laddr** (local address), **lport** (local port number), and **rdomain** (routing domain)

#### Remediation:

Edit the **/etc/ssh/sshd\_config** file to set the **IgnoreRhosts** parameter to **yes** above any **Include** and **Match** entries as follows:

```
IgnoreRhosts yes
```

**Note:** First occurrence of a option takes precedence, **Match** set statements withstanding. If **Include** locations are enabled, used, and order of precedence is understood in your environment, the entry may be created in a file in **Include** location.






#### Default Value:

IgnoreRhosts yes

## References:

1. SSHD\_CONFIG(5)
2. SSHD(8)
3. NIST SP 800-53 Rev. 5: CM-1, CM-2, CM-6, CM-7, IA-5

## CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>5.2 Use Unique Passwords</b> Use unique passwords for all enterprise assets. Best practice implementation includes, at a minimum, an 8-character password for accounts using MFA and a 14-character password for accounts not using MFA.			
v7	<b>4.4 Use Unique Passwords</b> Where multi-factor authentication is not supported (such as local administrator, root, or service accounts), accounts will use passwords that are unique to that system.			

## MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1078, T1078.001, T1078.003	TA0001	M1027

## 5.1.12 Ensure sshd KexAlgorithms is configured (Automated)

### Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

### Description:

Key exchange is any method in cryptography by which cryptographic keys are exchanged between two parties, allowing use of a cryptographic algorithm. If the sender and receiver wish to exchange encrypted messages, each must be equipped to encrypt messages to be sent and decrypt messages received

### Notes:

- Kex algorithms have a higher preference the earlier they appear in the list
- Some organizations may have stricter requirements for approved Key exchange algorithms
- Ensure that Key exchange algorithms used are in compliance with site policy
- The only Key Exchange Algorithms currently FIPS 140 approved are:
  - ecdh-sha2-nistp256
  - ecdh-sha2-nistp384
  - ecdh-sha2-nistp521
  - diffie-hellman-group-exchange-sha256
  - diffie-hellman-group16-sha512
  - diffie-hellman-group18-sha512
  - diffie-hellman-group14-sha256

### Rationale:

Key exchange methods that are considered weak should be removed. A key exchange method may be weak because too few bits are used, or the hashing algorithm is considered too weak. Using weak algorithms could expose connections to man-in-the-middle attacks

## Audit:

Run the following command to verify none of the "weak" Key Exchange algorithms are being used:

```
# sshd -T | grep -Pi -- 'kexalgorithms\h+([^\n\r]+,)?(diffie-hellman-group1-sha1|diffie-hellman-group14-sha1|diffie-hellman-group-exchange-sha1)\b'
```

Nothing should be returned

The following are considered "weak" Key Exchange Algorithms, and should not be used:

```
diffie-hellman-group1-sha1
diffie-hellman-group14-sha1
diffie-hellman-group-exchange-sha1
```

## Remediation:

Edit the `/etc/ssh/sshd_config` file and add/modify the `KexAlgorithms` line to contain a comma separated list of the site unapproved (weak) KexAlgorithms preceded with a `-` above any `Include` entries:

*Example:*

```
KexAlgorithms -diffie-hellman-group1-sha1,diffie-hellman-group14-sha1,diffie-hellman-group-exchange-sha1
```

**Note:** First occurrence of an option takes precedence. If Include locations are enabled, used, and order of precedence is understood in your environment, the entry may be created in a file in Include location.

## Default Value:

KexAlgorithms [sntrup761x25519-sha512@openssh.com](https://cweb.mit.edu/sshd/openssh.com),curve25519-sha256,[curve25519-sha256@libssh.org](https://cweb.mit.edu/sshd/libssh.org),ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group-exchange-sha256,diffie-hellman-group16-sha512,diffie-hellman-group18-sha512,diffie-hellman-group14-sha256

## References:





1. <https://ubuntu.com/server/docs/openssh-crypto-configuration>
2. NIST SP 800-53 Rev. 5: SC-8
3. SSHD(8)
4. SSHD\_CONFIG(5)

## Additional Information:

The supported algorithms are:

```
curve25519-sha256
curve25519-sha256@libssh.org
diffie-hellman-group1-sha1
diffie-hellman-group14-sha1
diffie-hellman-group14-sha256
diffie-hellman-group16-sha512
diffie-hellman-group18-sha512
diffie-hellman-group-exchange-sha1
diffie-hellman-group-exchange-sha256
ecdh-sha2-nistp256
ecdh-sha2-nistp384
ecdh-sha2-nistp521
sntrup4591761x25519-sha512@tinyssh.org
```

## CIS Controls:

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

## MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1040, T1040.000, T1557, T1557.000	TA0006	M1041

### 5.1.13 Ensure sshd LoginGraceTime is configured (Automated)

#### Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

#### Description:

The **LoginGraceTime** parameter specifies the time allowed for successful authentication to the SSH server. The longer the Grace period is the more open unauthenticated connections can exist. Like other session controls in this session the Grace Period should be limited to appropriate organizational limits to ensure the service is available for needed access.

#### Rationale:

Setting the **LoginGraceTime** parameter to a low number will minimize the risk of successful brute force attacks to the SSH server. It will also limit the number of concurrent unauthenticated connections While the recommended setting is 60 seconds (1 Minute), set the number based on site policy.

#### Audit:

Run the following command and verify that output **LoginGraceTime** is between **1** and **60** seconds:

```
# sshd -T | grep logingracetime  
logingracetime 60
```

#### Remediation:

Edit the **/etc/ssh/sshd\_config** file to set the **LoginGraceTime** parameter to **60** seconds or less above any **Include** entry as follows:

```
LoginGraceTime 60
```

**Note:** First occurrence of a option takes precedence. If Include locations are enabled, used, and order of precedence is understood in your environment, the entry may be created in a file in Include location.

#### Default Value:

LoginGraceTime 120

**References:**

1. SSHD\_CONFIG(5)
2. NIST SP 800-53 Rev. 5: CM-6
3. SSHD(8)

**MITRE ATT&CK Mappings:**

Techniques / Sub-techniques	Tactics	Mitigations
T1110, T1110.001, T1110.003, T1110.004	TA0006	M1036



### 5.1.14 Ensure sshd LogLevel is configured (Automated)

#### Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

#### Description:

SSH provides several logging levels with varying amounts of verbosity. The **DEBUG** options are specifically not recommended other than strictly for debugging SSH communications. These levels provide so much data that it is difficult to identify important security information, and may violate the privacy of users.

#### Rationale:

The **INFO** level is the basic level that only records login activity of SSH users. In many situations, such as Incident Response, it is important to determine when a particular user was active on a system. The logout record can eliminate those users who disconnected, which helps narrow the field.

The **VERBOSE** level specifies that login and logout activity as well as the key fingerprint for any SSH key used for login will be logged. This information is important for SSH key management, especially in legacy environments.

#### Audit:

Run the following command and verify that output matches **loglevel VERBOSE** or **loglevel INFO**:

```
# sshd -T | grep loglevel  
  
loglevel VERBOSE  
- OR -  
loglevel INFO
```

- **IF - Match** set statements are used in your environment, specify the connection parameters to use for the **-T** extended test mode and run the audit to verify the setting is not incorrectly configured in a match block

*Example additional audit needed for a match block for the user **sshuser**:*

```
# sshd -T -C user=sshuser | grep loglevel
```

**Note:** If provided, any Match directives in the configuration file that would apply are applied before the configuration is written to standard output. The connection parameters are supplied as keyword=value pairs and may be supplied in any order, either with multiple **-C** options or as a comma-separated list. The keywords are **addr** (source address), **user** (user), **host** (resolved source host name), **laddr** (local address), **lport** (local port number), and **rdomain** (routing domain)

## Remediation:

Edit the `/etc/ssh/sshd_config` file to set the `LogLevel` parameter to `VERBOSE` or `INFO` above any `Include` and `Match` entries as follows:

```
LogLevel VERBOSE
- OR -
LogLevel INFO
```

**Note:** First occurrence of an option takes precedence, `Match` set statements withstanding. If `Include` locations are enabled, used, and order of precedence is understood in your environment, the entry may be created in a file in `Include` location.









## Default Value:

LogLevel INFO

## References:

1. [https://www.ssh.com/ssh/sshd\\_config/](https://www.ssh.com/ssh/sshd_config/)
2. NIST SP 800-53 Rev. 5: AU-3, AU-12, SI-5

## CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>8.2 Collect Audit Logs</b> Collect audit logs. Ensure that logging, per the enterprise's audit log management process, has been enabled across enterprise assets.			
v7	<b>6.2 Activate audit logging</b> Ensure that local logging has been enabled on all systems and networking devices.			
v7	<b>6.3 Enable Detailed Logging</b> 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, T1562.006	TA0005	

### *5.1.15 Ensure sshd MACs are configured (Automated)*

#### **Profile Applicability:**

- Level 1 - Server
- Level 1 - Workstation

#### **Description:**

This variable limits the types of MAC algorithms that SSH can use during communication.

#### **Notes:**

- Some organizations may have stricter requirements for approved MACs.
- Ensure that MACs used are in compliance with site policy.
- The only "strong" MACs currently FIPS 140 approved are:
  - HMAC-SHA1
  - HMAC-SHA2-256
  - HMAC-SHA2-384
  - HMAC-SHA2-512

#### **Rationale:**

MD5 and 96-bit MAC algorithms are considered weak and have been shown to increase exploitability in SSH downgrade attacks. Weak algorithms continue to have a great deal of attention as a weak spot that can be exploited with expanded computing power. An attacker that breaks the algorithm could take advantage of a MiTM position to decrypt the SSH tunnel and capture credentials and information.

## Audit:

Run the following command to verify none of the "weak" MACs are being used:

```
# sshd -T | grep -Pi -- 'macs\h+([^\n\r]+,)?(hmac-md5|hmac-md5-96|hmac-ripemd160|hmac-sha1-96|umac-64@openssh.com|hmac-md5-etm@openssh.com|hmac-md5-96-etm@openssh.com|hmac-ripemd160-etm@openssh.com|hmac-sha1-96-etm@openssh.com|umac-64-etm@openssh.com|umac-128-etm@openssh.com)\b'
```

Nothing should be returned

**Note:** Review [CVE-2023-48795](#) and verify the system has been patched. If the system has not been patched, review the use of the Encrypt Then Mac (etm) MACs. The following are considered "weak" MACs, and should not be used:

```
hmac-md5
hmac-md5-96
hmac-ripemd160
hmac-sha1-96
umac-64@openssh.com
hmac-md5-etm@openssh.com
hmac-md5-96-etm@openssh.com
hmac-ripemd160-etm@openssh.com
hmac-sha1-96-etm@openssh.com
umac-64-etm@openssh.com
umac-128-etm@openssh.com
```

## Remediation:

Edit the `/etc/ssh/sshd_config` file and add/modify the **MACs** line to contain a comma separated list of the site unapproved (weak) MACs preceded with a **-** above any **Include** entries:

*Example:*

```
MACs -hmac-md5,hmac-md5-96,hmac-ripemd160,hmac-sha1-96,umac-64@openssh.com,hmac-md5-etm@openssh.com,hmac-md5-96-etm@openssh.com,hmac-ripemd160-etm@openssh.com,hmac-sha1-96-etm@openssh.com,umac-64-etm@openssh.com,umac-128-etm@openssh.com
```

- **IF** - [CVE-2023-48795](#) has not been reviewed and addressed, the following **etm** MACs should be added to the exclude list: [hmac-sha1-etm@openssh.com](#),[hmac-sha2-256-etm@openssh.com](#),[hmac-sha2-512-etm@openssh.com](#)

**Note:** First occurrence of an option takes precedence. If Include locations are enabled, used, and order of precedence is understood in your environment, the entry may be created in a file in Include location.







## Default Value:

MACs [umac-64-etm@openssh.com](#),[umac-128-etm@openssh.com](#),[hmac-sha2-256-etm@openssh.com](#),[hmac-sha2-512-etm@openssh.com](#),[hmac-sha1-etm@openssh.com](#),[umac-64@openssh.com](#),[umac-128@openssh.com](#),hmac-sha2-256,hmac-sha2-512,hmac-sha1

## References:

1. <https://nvd.nist.gov/vuln/detail/CVE-2023-48795>
2. More information on SSH downgrade attacks can be found here:  
<http://www.mitsl.org/pages/attacks/SLOTH>
3. SSHD\_CONFIG(5)
4. NIST SP 800-53 Rev. 5: CM-1, CM-2, CM-6, CM-7, IA-5

## CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>3.10 <u>Encrypt Sensitive Data in Transit</u></b> Encrypt sensitive data in transit. Example implementations can include: Transport Layer Security (TLS) and Open Secure Shell (OpenSSH).			
v7	<b>14.4 <u>Encrypt All Sensitive Information in Transit</u></b> Encrypt all sensitive information in transit.			
v7	<b>16.5 <u>Encrypt Transmittal of Username and Authentication Credentials</u></b> Ensure that all account usernames and authentication credentials are transmitted across networks using encrypted channels.			

## MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1040, T1040.000, T1557, T1557.000	TA0006	M1041

## 5.1.16 Ensure sshd MaxAuthTries is configured (Automated)

### Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

### Description:

The **MaxAuthTries** parameter specifies the maximum number of authentication attempts permitted per connection. When the login failure count reaches half the number, error messages will be written to the **syslog** file detailing the login failure.

### Rationale:

Setting the **MaxAuthTries** parameter to a low number will minimize the risk of successful brute force attacks to the SSH server. While the recommended setting is 4, set the number based on site policy.

### Audit:

Run the following command and verify that **MaxAuthTries** is 4 or less:

```
# sshd -T | grep maxauthtries  
maxauthtries 4
```

- **IF - Match** set statements are used in your environment, specify the connection parameters to use for the **-T** extended test mode and run the audit to verify the setting is not incorrectly configured in a match block

*Example additional audit needed for a match block for the user **ssuser**:*

```
# sshd -T -C user=ssuser | grep maxauthtries
```

**Note:** If provided, any Match directives in the configuration file that would apply are applied before the configuration is written to standard output. The connection parameters are supplied as keyword=value pairs and may be supplied in any order, either with multiple **-C** options or as a comma-separated list. The keywords are **addr** (source address), **user** (user), **host** (resolved source host name), **laddr** (local address), **lport** (local port number), and **rdomain** (routing domain)

### Remediation:

Edit the **/etc/ssh/sshd\_config** file to set the **MaxAuthTries** parameter to 4 or less above any **Include** and **Match** entries as follows:

```
MaxAuthTries 4
```

**Note:** First occurrence of an option takes precedence, **Match** set statements withstanding. If **Include** locations are enabled, used, and order of precedence is understood in your environment, the entry may be created in a file in **Include** location.

**Default Value:**

MaxAuthTries 6

**References:**

1. SSHD\_CONFIG(5)
2. NIST SP 800-53 Rev. 5: AU-3

**CIS Controls:**

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>8.5 <u>Collect Detailed Audit Logs</u></b> 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	<b>16.13 <u>Alert on Account Login Behavior Deviation</u></b> Alert when users deviate from normal login behavior, such as time-of-day, workstation location and duration.			●

**MITRE ATT&CK Mappings:**

Techniques / Sub-techniques	Tactics	Mitigations
T1110, T1110.001, T1110.003	TA0006	M1036

## 5.1.17 Ensure sshd MaxSessions is configured (Automated)

### Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

### Description:

The **MaxSessions** parameter specifies the maximum number of open sessions permitted from a given connection.

### Rationale:

To protect a system from denial of service due to a large number of concurrent sessions, use the rate limiting function of MaxSessions to protect availability of sshd logins and prevent overwhelming the daemon.

### Audit:

Run the following command and verify that **MaxSessions** is **10** or less:

```
# sshd -T | grep -i maxsessions  
  
maxsessions 10
```

Run the following command and verify the output:

```
grep -Psi -- '^h*MaxSessions\h+\\"?(1[1-9]|[2-9][0-9]|[1-9][0-9][0-9]+)\b'  
/etc/ssh/sshd_config /etc/ssh/sshd_config.d/*.conf  
  
Nothing should be returned
```

- **IF - Match** set statements are used in your environment, specify the connection parameters to use for the **-T** extended test mode and run the audit to verify the setting is not incorrectly configured in a match block

*Example additional audit needed for a match block for the user **sshuser**:*

```
# sshd -T -C user=sshuser | grep maxsessions
```

**Note:** If provided, any Match directives in the configuration file that would apply are applied before the configuration is written to standard output. The connection parameters are supplied as keyword=value pairs and may be supplied in any order, either with multiple **-C** options or as a comma-separated list. The keywords are **addr** (source address), **user** (user), **host** (resolved source host name), **laddr** (local address), **lport** (local port number), and **rdomain** (routing domain)



## Remediation:

Edit the `/etc/ssh/sshd_config` file to set the `MaxSessions` parameter to **10** or less above any `Include` and `Match` entries as follows:

```
MaxSessions 10
```

**Note:** First occurrence of an option takes precedence, `Match` set statements withstanding. If `Include` locations are enabled, used, and order of precedence is understood in your environment, the entry may be created in a file in `Include` location.

## Default Value:

MaxSessions 10

## References:

1. SSHD\_CONFIG(5)
2. 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
T1499, T1499.002	TA0040	

## 5.1.18 Ensure sshd MaxStartups is configured (Automated)

### Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

### Description:

The **MaxStartups** parameter specifies the maximum number of concurrent unauthenticated connections to the SSH daemon.

### Rationale:

To protect a system from denial of service due to a large number of pending authentication connection attempts, use the rate limiting function of MaxStartups to protect availability of sshd logins and prevent overwhelming the daemon.

### Audit:

Run the following command to verify **MaxStartups** is **10:30:60** or more restrictive:

```
# sshd -T | awk '$1 ~ /^s*maxstartups/{split($2, a, ":");if(a[1] > 10 || a[2] > 30 || a[3] > 60) print $0}{'
```

Nothing should be returned

### Remediation:

Edit the **/etc/ssh/sshd\_config** file to set the **MaxStartups** parameter to **10:30:60** or more restrictive above any **Include** entries as follows:

```
MaxStartups 10:30:60
```

**Note:** First occurrence of a option takes precedence. If Include locations are enabled, used, and order of precedence is understood in your environment, the entry may be created in a file in Include location.

### Default Value:

MaxStartups 10:30:100

### References:

1. SSHD\_CONFIG(5)
2. NIST SP 800-53 Rev. 5: CM-1, CM-2, CM-6, CM-7, IA-5

**MITRE ATT&CK Mappings:**

<b>Techniques / Sub-techniques</b>	<b>Tactics</b>	<b>Mitigations</b>
T1499, T1499.002	TA0040	

## 5.1.19 Ensure sshd PermitEmptyPasswords is disabled (Automated)

### Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

### Description:

The **PermitEmptyPasswords** parameter specifies if the SSH server allows login to accounts with empty password strings.

### Rationale:

Disallowing remote shell access to accounts that have an empty password reduces the probability of unauthorized access to the system.

### Audit:

Run the following command to verify **PermitEmptyPasswords** is set to **no**:

```
# sshd -T | grep permitemptypasswords  
  
permitemptypasswords no
```

- **IF - Match** set statements are used in your environment, specify the connection parameters to use for the **-T** extended test mode and run the audit to verify the setting is not incorrectly configured in a match block

*Example additional audit needed for a match block for the user **sshuser**:*

```
# sshd -T -C user=sshuser | grep permitemptypasswords
```

**Note:** If provided, any Match directives in the configuration file that would apply are applied before the configuration is written to standard output. The connection parameters are supplied as keyword=value pairs and may be supplied in any order, either with multiple **-C** options or as a comma-separated list. The keywords are **addr** (source address), **user** (user), **host** (resolved source host name), **laddr** (local address), **lport** (local port number), and **rdomain** (routing domain)

### Remediation:

Edit **/etc/ssh/sshd\_config** and set the **PermitEmptyPasswords** parameter to **no** above any **Include** and **Match** entries as follows:

```
PermitEmptyPasswords no
```

**Note:** First occurrence of an option takes precedence, **Match** set statements withstanding. If **Include** locations are enabled, used, and order of precedence is understood in your environment, the entry may be created in a file in **Include** location.






**Default Value:**

PermitEmptyPasswords no

**References:**

1. SSHD\_CONFIG(5)
2. NIST SP 800-53 Rev. 5: CM-1,CM-2, CM-6, CM-7, IA-5

**CIS Controls:**

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>5.2 Use Unique Passwords</b> Use unique passwords for all enterprise assets. Best practice implementation includes, at a minimum, an 8-character password for accounts using MFA and a 14-character password for accounts not using MFA.			
v7	<b>4.4 Use Unique Passwords</b> Where multi-factor authentication is not supported (such as local administrator, root, or service accounts), accounts will use passwords that are unique to that system.			

**MITRE ATT&CK Mappings:**

Techniques / Sub-techniques	Tactics	Mitigations
T1021	TA0008	M1042

## 5.1.20 Ensure sshd PermitRootLogin is disabled (Automated)

### Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

### Description:

The **PermitRootLogin** parameter specifies if the root user can log in using SSH. The default is **prohibit-password**.

### Rationale:

Disallowing **root** logins over SSH requires system admins to authenticate using their own individual account, then escalating to **root**. This limits opportunity for non-repudiation and provides a clear audit trail in the event of a security incident.

### Audit:

Run the following command to verify **PermitRootLogin** is set to **no**:

```
# sshd -T | grep permitrootlogin  
permitrootlogin no
```

- **IF - Match** set statements are used in your environment, specify the connection parameters to use for the **-T** extended test mode and run the audit to verify the setting is not incorrectly configured in a match block

*Example additional audit needed for a match block for the user **sshuser**:*

```
# sshd -T -C user=sshuser | grep permitrootlogin
```

**Note:** If provided, any Match directives in the configuration file that would apply are applied before the configuration is written to standard output. The connection parameters are supplied as keyword=value pairs and may be supplied in any order, either with multiple **-C** options or as a comma-separated list. The keywords are **addr** (source address), **user** (user), **host** (resolved source host name), **laddr** (local address), **lport** (local port number), and **rdomain** (routing domain)

### Remediation:

Edit the **/etc/ssh/sshd\_config** file to set the **PermitRootLogin** parameter to **no** above any **Include** and **Match** entries as follows:

```
PermitRootLogin no
```

**Note:** First occurrence of an option takes precedence, **Match** set statements withstanding. If **Include** locations are enabled, used, and order of precedence is understood in your environment, the entry may be created in a file in **Include** location.







**Default Value:**

PermitRootLogin without-password

**References:**

1. SSHD\_CONFIG(5)
2. NIST SP 800-53 Rev. 5:AC-6

**CIS Controls:**

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>5.4 <u>Restrict Administrator Privileges to Dedicated Administrator Accounts</u></b> Restrict administrator privileges to dedicated administrator accounts on enterprise assets. Conduct general computing activities, such as internet browsing, email, and productivity suite use, from the user's primary, non-privileged account.			
v7	<b>4.3 <u>Ensure the Use of Dedicated Administrative Accounts</u></b> Ensure that all users with administrative account access use a dedicated or secondary account for elevated activities. This account should only be used for administrative activities and not internet browsing, email, or similar activities.			

**MITRE ATT&CK Mappings:**

Techniques / Sub-techniques	Tactics	Mitigations
T1021	TA0008	M1042

## 5.1.21 Ensure sshd PermitUserEnvironment is disabled (Automated)

### Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

### Description:

The **PermitUserEnvironment** option allows users to present environment options to the SSH daemon.

### Rationale:

Permitting users the ability to set environment variables through the SSH daemon could potentially allow users to bypass security controls (e.g. setting an execution path that has SSH executing trojan'd programs)

### Audit:

Run the following command to verify **PermitUserEnvironment** is set to **no**:

```
# sshd -T | grep permituserenvironment  
permituserenvironment no
```

### Remediation:

Edit the **/etc/ssh/sshd\_config** file to set the **PermitUserEnvironment** parameter to **no** above any **Include** entries as follows:

```
PermitUserEnvironment no
```

**Note:** First occurrence of an option takes precedence. If Include locations are enabled, used, and order of precedence is understood in your environment, the entry may be created in a file in Include location.

### Default Value:

PermitUserEnvironment no

### References:

1. SSHD\_CONFIG(5)
2. NIST SP 800-53 Rev. 5: CM-1, CM-2, CM-6, CM-7, IA-5
3. SSHD(8)



**MITRE ATT&CK Mappings:**

<b>Techniques / Sub-techniques</b>	<b>Tactics</b>	<b>Mitigations</b>
T1021	TA0008	M1042

## 5.1.22 Ensure sshd UsePAM is enabled (Automated)

### Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

### Description:

The **UsePAM** directive enables the Pluggable Authentication Module (PAM) interface. If set to **yes** this will enable PAM authentication using **ChallengeResponseAuthentication** and **PasswordAuthentication** directives in addition to PAM account and session module processing for all authentication types.

### Rationale:

When **usePAM** is set to **yes**, PAM runs through account and session types properly. This is important if you want to restrict access to services based off of IP, time or other factors of the account. Additionally, you can make sure users inherit certain environment variables on login or disallow access to the server

### Audit:

Run the following command to verify **UsePAM** is set to **yes**:

```
# sshd -T | grep -i usepam  
usepam yes
```

### Remediation:

Edit the **/etc/ssh/sshd\_config** file to set the **UsePAM** parameter to **yes** above any **Include** entries as follows:

```
UsePAM yes
```

**Note:** First occurrence of an option takes precedence. If Include locations are enabled, used, and order of precedence is understood in your environment, the entry may be created in a file in Include location.






### Default Value:

UsePAM yes

### References:

1. SSHD\_CONFIG(5)
2. NIST SP 800-53 Rev. 5: CM-1, CM-2, CM-6, CM-7, IA-5
3. SSHD(8)

## CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>5.2 Use Unique Passwords</b> Use unique passwords for all enterprise assets. Best practice implementation includes, at a minimum, an 8-character password for accounts using MFA and a 14-character password for accounts not using MFA.			
v7	<b>4.4 Use Unique Passwords</b> Where multi-factor authentication is not supported (such as local administrator, root, or service accounts), accounts will use passwords that are unique to that system.			

## MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1021, T1021.004	TA0001	M1035

## 5.2 Configure privilege escalation

There are various tools which allows a permitted user to execute a command as the superuser or another user, as specified by the security policy.

### **sudo**

[sudo documentation](#)

The invoking user's real (not effective) user ID is used to determine the user name with which to query the security policy.

**sudo** supports a plug-in architecture for security policies and input/output logging. Third parties can develop and distribute their own policy and I/O logging plug-ins to work seamlessly with the **sudo** front end. The default security policy is **sudoers**, which is configured via the file **/etc/sudoers** and any entries in **/etc/sudoers.d**.

### **pkexec**

[pkexec documentation](#)

**pkexec** allows an authorized user to execute *PROGRAM* as another user. If *username* is not specified, then the program will be executed as the administrative super user, **root**.

## 5.2.1 Ensure sudo is installed (Automated)

### Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

### Description:

**sudo** allows a permitted user to execute a command as the superuser or another user, as specified by the security policy. The invoking user's real (not effective) user ID is used to determine the user name with which to query the security policy.

### Rationale:

**sudo** supports a plug-in architecture for security policies and input/output logging. Third parties can develop and distribute their own policy and I/O logging plug-ins to work seamlessly with the **sudo** front end. The default security policy is **sudoers**, which is configured via the file **/etc/sudoers** and any entries in **/etc/sudoers.d**.

The security policy determines what privileges, if any, a user has to run **sudo**. The policy may require that users authenticate themselves with a password or another authentication mechanism. If authentication is required, **sudo** will exit if the user's password is not entered within a configurable time limit. This limit is policy-specific.

### Audit:

Run the following command to verify that either **sudo** is installed:

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

### - OR -

Run the following command to verify that either **sudo-ldap** is installed:

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

### Remediation:

First determine if LDAP functionality is required. If so, then install **sudo-ldap**, else install **sudo**.







*Example:*

```
# apt install sudo
```

## References:

1. SUDO(8)
2. NIST SP 800-53 Rev. 5: AC-6

## CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>5.4 <u>Restrict Administrator Privileges to Dedicated Administrator Accounts</u></b> Restrict administrator privileges to dedicated administrator accounts on enterprise assets. Conduct general computing activities, such as internet browsing, email, and productivity suite use, from the user's primary, non-privileged account.			
v7	<b>4.3 <u>Ensure the Use of Dedicated Administrative Accounts</u></b> Ensure that all users with administrative account access use a dedicated or secondary account for elevated activities. This account should only be used for administrative activities and not internet browsing, email, or similar activities.			

## MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1548	TA0004	M1026

## 5.2.2 Ensure sudo commands use pty (Automated)

### Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

### Description:

**sudo** can be configured to run only from a pseudo terminal (**pseudo-pty**).

### Rationale:

Attackers can run a malicious program using **sudo** which would fork a background process that remains even when the main program has finished executing.

### Impact:

**WARNING:** Editing the **sudo** configuration incorrectly can cause **sudo** to stop functioning. Always use **visudo** to modify **sudo** configuration files.

### Audit:

Verify that **sudo** can only run other commands from a pseudo terminal.

Run the following command to verify **Defaults use\_pty** is set:

```
# grep -rPi -- '^\\h*Defaults\\h+([\\^#\\n\\r]+,\\h*)?use_pty\\b' /etc/sudoers*
```

Verify the output matches:

```
/etc/sudoers:Defaults use_pty
```

Run the follow command to to verify **Defaults !use\_pty** is not set:

```
# grep -rPi -- '^\\h*Defaults\\h+([\\^#\\n\\r]+,\\h*)?!use_pty\\b' /etc/sudoers*
```

Nothing should be returned

## Remediation:

Edit the file `/etc/sudoers` with `visudo` or a file in `/etc/sudoers.d/` with `visudo -f <PATH TO FILE>` and add the following line:

```
Defaults use_pty
```

Edit the file `/etc/sudoers` with `visudo` and any files in `/etc/sudoers.d/` with `visudo -f <PATH TO FILE>` and remove any occurrence of `!use_pty`







### Note:

- sudo will read each file in `/etc/sudoers.d`, skipping file names that end in `~` or contain a `.` character to avoid causing problems with package manager or editor temporary/backup files.
- Files are parsed in sorted lexical order. That is, `/etc/sudoers.d/01_first` will be parsed before `/etc/sudoers.d/10_second`.
- Be aware that because the sorting is lexical, not numeric, `/etc/sudoers.d/1_whoops` would be loaded after `/etc/sudoers.d/10_second`.
- Using a consistent number of leading zeroes in the file names can be used to avoid such problems.

## References:

1. SUDO(8)
2. VISUDO(8)
3. sudoers(5)
4. NIST SP 800-53 Rev. 5: AC-6

## CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b><u>5.4 Restrict Administrator Privileges to Dedicated Administrator Accounts</u></b> Restrict administrator privileges to dedicated administrator accounts on enterprise assets. Conduct general computing activities, such as internet browsing, email, and productivity suite use, from the user's primary, non-privileged account.			
v7	<b><u>5.1 Establish Secure Configurations</u></b> Maintain documented, standard security configuration standards for all authorized operating systems and software.			



**MITRE ATT&CK Mappings:**

Techniques / Sub-techniques	Tactics	Mitigations
T1078, T1078.003, T1548, T1548.003	TA0001, TA0003	M1026, M1028

### 5.2.3 Ensure sudo log file exists (Automated)

#### Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

#### Description:

sudo can use a custom log file

#### Rationale:

A sudo log file simplifies auditing of sudo commands

#### Impact:

**WARNING:** Editing the **sudo** configuration incorrectly can cause **sudo** to stop functioning. Always use **visudo** to modify **sudo** configuration files.

#### Audit:

Run the following command to verify that sudo has a custom log file configured:

```
# grep -rPsi  
"^\\h*Defaults\\h+([^#]+,\\h*)?logfile\\h*=\\h*(\"|\\')?\\H+(\"|\\')?(,\\h*\\H+\\h*)*\\h*  
(#\\.*)?$" /etc/sudoers*
```

Verify the output matches:

```
Defaults logfile="/var/log/sudo.log"
```

## Remediation:

Edit the file `/etc/sudoers` or a file in `/etc/sudoers.d/` with `visudo` or `visudo -f <PATH TO FILE>` and add the following line:

Example:

```
Defaults logfile="/var/log/sudo.log"
```

## Note:

- `sudo` will read each file in `/etc/sudoers.d`, skipping file names that end in `~` or contain a `.` character to avoid causing problems with package manager or editor temporary/backup files.
- Files are parsed in sorted lexical order. That is, `/etc/sudoers.d/01_first` will be parsed before `/etc/sudoers.d/10_second`.
- Be aware that because the sorting is lexical, not numeric, `/etc/sudoers.d/1_whoops` would be loaded after `/etc/sudoers.d/10_second`.
- Using a consistent number of leading zeroes in the file names can be used to avoid such problems.





## References:

1. SUDO(8)
2. VISUDO(8)
3. sudoers(5)
4. NIST SP 800-53 Rev. 5: AU-3, AU-12

## Additional Information:

`visudo` edits the `sudoers` file in a safe fashion, analogous to `vipw(8)`. `visudo` locks the `sudoers` file against multiple simultaneous edits, provides basic sanity checks, and checks for parse errors. If the `sudoers` file is currently being edited you will receive a message to try again later.

## CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>8.5 Collect Detailed Audit Logs</b> 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	<b>6.3 Enable Detailed Logging</b> 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:**

<b>Techniques / Sub-techniques</b>	<b>Tactics</b>	<b>Mitigations</b>
T1562, T1562.006	TA0004	M1026

## 5.2.4 Ensure users must provide password for privilege escalation (Automated)

### Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

### Description:

The operating system must be configured so that users must provide a password for privilege escalation.

### Rationale:

Without (re-)authentication, users may access resources or perform tasks for which they do not have authorization.

When operating systems provide the capability to escalate a functional capability, it is critical the user (re-)authenticate.

### Impact:

This will prevent automated processes from being able to elevate privileges.

### Audit:

**Note:** If passwords are not being used for authentication, this is not applicable. Verify the operating system requires users to supply a password for privilege escalation. Check the configuration of the `/etc/sudoers` and `/etc/sudoers.d/*` files with the following command:

```
# grep -r "^[^#]*NOPASSWD" /etc/sudoers*
```

If any line is found refer to the remediation procedure below.

### Remediation:







Based on the outcome of the audit procedure, use `visudo -f <PATH TO FILE>` to edit the relevant sudoers file.

Remove any line with occurrences of `NOPASSWD` tags in the file.

### References:

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

### CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>5.4 <u>Restrict Administrator Privileges to Dedicated Administrator Accounts</u></b> Restrict administrator privileges to dedicated administrator accounts on enterprise assets. Conduct general computing activities, such as internet browsing, email, and productivity suite use, from the user's primary, non-privileged account.			
v7	<b>4.3 <u>Ensure the Use of Dedicated Administrative Accounts</u></b> Ensure that all users with administrative account access use a dedicated or secondary account for elevated activities. This account should only be used for administrative activities and not internet browsing, email, or similar activities.			

### MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1548	TA0004	M1026

## 5.2.5 Ensure re-authentication for privilege escalation is not disabled globally (Automated)

### Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

### Description:

The operating system must be configured so that users must re-authenticate for privilege escalation.

### Rationale:

Without re-authentication, users may access resources or perform tasks for which they do not have authorization.

When operating systems provide the capability to escalate a functional capability, it is critical the user re-authenticate.

### Audit:

Verify the operating system requires users to re-authenticate for privilege escalation. Check the configuration of the `/etc/sudoers` and `/etc/sudoers.d/*` files with the following command:

```
# grep -r "^[^#].*\!authenticate" /etc/sudoers*
```

If any line is found with a `!authenticate` tag, refer to the remediation procedure below.

### Remediation:

Configure the operating system to require users to reauthenticate for privilege escalation.







Based on the outcome of the audit procedure, use `visudo -f <PATH TO FILE>` to edit the relevant sudoers file.

Remove any occurrences of `!authenticate` tags in the file(s).

### References:

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

### CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b><u>5.4 Restrict Administrator Privileges to Dedicated Administrator Accounts</u></b> Restrict administrator privileges to dedicated administrator accounts on enterprise assets. Conduct general computing activities, such as internet browsing, email, and productivity suite use, from the user's primary, non-privileged account.			
v7	<b><u>4.3 Ensure the Use of Dedicated Administrative Accounts</u></b> Ensure that all users with administrative account access use a dedicated or secondary account for elevated activities. This account should only be used for administrative activities and not internet browsing, email, or similar activities.			

### MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1548	TA0004	M1026



## 5.2.6 Ensure sudo authentication timeout is configured correctly (Automated)

### Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

### Description:

**sudo** caches used credentials for a default of 15 minutes. This is for ease of use when there are multiple administrative tasks to perform. The timeout can be modified to suit local security policies.

This default is distribution specific. See audit section for further information.

### Rationale:

Setting a timeout value reduces the window of opportunity for unauthorized privileged access to another user.

### Audit:

Ensure that the caching timeout is no more than 15 minutes.

Example:

```
# grep -roP "timestamp_timeout=\K[0-9]*" /etc/sudoers*
```

If there is no **timestamp\_timeout** configured in **/etc/sudoers\*** then the default is 15 minutes. This default can be checked with:

```
# sudo -V | grep "Authentication timestamp timeout:"
```

**Note:** A value of **-1** means that the timeout is disabled. Depending on the configuration of the **timestamp\_type**, this could mean for all terminals / processes of that user and not just that one single terminal session.

## Remediation:







If the currently configured timeout is larger than 15 minutes, edit the file listed in the audit section with `visudo -f <PATH TO FILE>` and modify the entry `timestamp_timeout=` to 15 minutes or less as per your site policy. The value is in minutes. This particular entry may appear on it's own, or on the same line as `env_reset`. See the following two examples:

```
Defaults    env_reset, timestamp_timeout=15
Defaults    timestamp_timeout=15
Defaults    env_reset
```

## References:

1. <https://www.sudo.ws/man/1.9.0/sudoers.man.html>
2. NIST SP 800-53 Rev. 5: AC-6

## CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b><u>5.4 Restrict Administrator Privileges to Dedicated Administrator Accounts</u></b> Restrict administrator privileges to dedicated administrator accounts on enterprise assets. Conduct general computing activities, such as internet browsing, email, and productivity suite use, from the user's primary, non-privileged account.			
v7	<b><u>4.3 Ensure the Use of Dedicated Administrative Accounts</u></b> Ensure that all users with administrative account access use a dedicated or secondary account for elevated activities. This account should only be used for administrative activities and not internet browsing, email, or similar activities.			

## MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1548	TA0004	M1026

## 5.2.7 Ensure access to the su command is restricted (Automated)

### Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

### Description:

The **su** command allows a user to run a command or shell as another user. The program has been superseded by **sudo**, which allows for more granular control over privileged access. Normally, the **su** command can be executed by any user. By uncommenting the **pam\_wheel.so** statement in **/etc/pam.d/su**, the **su** command will only allow users in a specific groups to execute **su**. This group should be empty to reinforce the use of **sudo** for privileged access.

### Rationale:

Restricting the use of **su**, and using **sudo** in its place, provides system administrators better control of the escalation of user privileges to execute privileged commands. The **sudo** utility also provides a better logging and audit mechanism, as it can log each command executed via **sudo**, whereas **su** can only record that a user executed the **su** program.

### Audit:

Run the following command:

```
# grep -Pi
'^\h*auth\h+(?:required|requisite)\h+pam_wheel\.so\h+(?:[^\n\r]+\h+)?((?!2)
(use_uid\b|group=\H+\b))\h+(?:[^\n\r]+\h+)?((?!1)(use_uid\b|group=\H+\b))(\
h+.*?)?$' /etc/pam.d/su
```

Verify the output matches:

```
auth required pam_wheel.so use_uid group=<group_name>
```

Run the following command and verify that the group specified in **<group\_name>** contains no users:

```
# grep <group_name> /etc/group
```

Verify the output does not contain any users in the relevant group:

```
<group_name>:x:<GID>:
```

## Remediation:

Create an empty group that will be specified for use of the **su** command. The group should be named according to site policy.

Example:

```
# groupadd sugroup
```







Add the following line to the **/etc/pam.d/su** file, specifying the empty group:

```
auth required pam_wheel.so use_uid group=sugroup
```

## References:

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

## CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>3.3 <u>Configure Data Access Control Lists</u></b> 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	<b>14.6 <u>Protect Information through Access Control Lists</u></b> 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
T1548	TA0005	M1026