

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_output2[@]}" ""
      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	3.3 <u>Configure Data Access Control Lists</u> Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	14.6 <u>Protect Information through Access Control Lists</u> 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	3.3 <u>Configure Data Access Control Lists</u> Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	14.6 <u>Protect Information through Access Control Lists</u> 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	3.3 <u>Configure Data Access Control Lists</u> Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	14.6 <u>Protect Information through Access Control Lists</u> 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	3.3 <u>Configure Data Access Control Lists</u> 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	4.3 <u>Ensure the Use of Dedicated Administrative Accounts</u> 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://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	4.8 <u>Uninstall or Disable Unnecessary Services on Enterprise Assets and Software</u> Uninstall or disable unnecessary services on enterprise assets and software, such as an unused file sharing service, web application module, or service function.			
v7	9.2 <u>Ensure Only Approved Ports, Protocols and Services Are Running</u> Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
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	5.2 Use Unique Passwords Use unique passwords for all enterprise assets. Best practice implementation includes, at a minimum, an 8-character password for accounts using MFA and a 14-character password for accounts not using MFA.			
v7	4.4 Use Unique Passwords Where multi-factor authentication is not supported (such as local administrator, root, or service accounts), accounts will use passwords that are unique to that system.			

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

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://cvsweb.openbsd.org/cvsweb/src/etc/ssh/ssh_config),curve25519-sha256,[curve25519-sha256@libssh.org](https://cvsweb.openbsd.org/cvsweb/src/etc/ssh/ssh_config),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	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, 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/
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	8.2 Collect Audit Logs Collect audit logs. Ensure that logging, per the enterprise's audit log management process, has been enabled across enterprise assets.			
v7	6.2 Activate audit logging Ensure that local logging has been enabled on all systems and networking devices.			
v7	6.3 Enable Detailed Logging Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1562, T1562.006	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.mitls.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	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.			
v7	16.5 <u>Encrypt Transmittal of Username and Authentication Credentials</u> 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	8.5 <u>Collect Detailed Audit Logs</u> 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	16.13 <u>Alert on Account Login Behavior Deviation</u> 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:

Techniques / Sub-techniques	Tactics	Mitigations
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	5.2 Use Unique Passwords Use unique passwords for all enterprise assets. Best practice implementation includes, at a minimum, an 8-character password for accounts using MFA and a 14-character password for accounts not using MFA.			
v7	4.4 Use Unique Passwords Where multi-factor authentication is not supported (such as local administrator, root, or service accounts), accounts will use passwords that are unique to that system.			

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	5.4 <u>Restrict Administrator Privileges to Dedicated Administrator Accounts</u> Restrict administrator privileges to dedicated administrator accounts on enterprise assets. Conduct general computing activities, such as internet browsing, email, and productivity suite use, from the user's primary, non-privileged account.			
v7	4.3 <u>Ensure the Use of Dedicated Administrative Accounts</u> 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:

Techniques / Sub-techniques	Tactics	Mitigations
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	5.2 Use Unique Passwords Use unique passwords for all enterprise assets. Best practice implementation includes, at a minimum, an 8-character password for accounts using MFA and a 14-character password for accounts not using MFA.			
v7	4.4 Use Unique Passwords Where multi-factor authentication is not supported (such as local administrator, root, or service accounts), accounts will use passwords that are unique to that system.			

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	5.4 <u>Restrict Administrator Privileges to Dedicated Administrator Accounts</u> Restrict administrator privileges to dedicated administrator accounts on enterprise assets. Conduct general computing activities, such as internet browsing, email, and productivity suite use, from the user's primary, non-privileged account.			
v7	4.3 <u>Ensure the Use of Dedicated Administrative Accounts</u> 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	<u>5.4 Restrict Administrator Privileges to Dedicated Administrator Accounts</u> 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	<u>5.1 Establish Secure Configurations</u> 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	8.5 Collect Detailed Audit Logs Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.			
v7	6.3 Enable Detailed Logging Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.			

MITRE ATT&CK Mappings:

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

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	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1548	TA0005	M1026

5.3 Pluggable Authentication Modules

5.3.1 Configure PAM software packages

Updated versions of PAM include additional functionality

5.3.1.1 Ensure latest version of pam is installed (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

Updated versions of PAM include additional functionality

Rationale:

To ensure the system has full functionality and access to the options covered by this Benchmark the latest version of **libpam-runtime** should be installed on the system

Audit:

Run the following command to verify the version of **libpam-runtime** on the system:

```
# dpkg-query -s libpam-runtime | grep -P -- '^(Status|Version)\b'
```

The output should be similar to:

```
Status: install ok installed
Version: 1.5.3-5
```

Remediation:

- **IF** - the version of **libpam-runtime** on the system is less than version **1.5.3-5**:
Run the following command to update to the latest version of **PAM**:

```
# apt upgrade libpam-runtime
```

5.3.1.2 Ensure *libpam-modules* is installed (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

Pluggable Authentication Modules for PAM

Rationale:

To ensure the system has full functionality and access to the PAM options covered by this Benchmark

Audit:

Run the following command to verify **libpam-modules** is installed and version **1.5.3-5** or later:

```
# dpkg-query -s libpam-modules | grep -P -- '^(Status|Version)\b'
```

The output should be similar to:

```
Status: install ok installed
Version: 1.5.3-5
```

Remediation:

- **IF** - the version of **libpam-modules** on the system is less than version **1.5.3-5**:
Run the following command to update to the latest version of **PAM**:

```
# apt upgrade libpam-modules
```


5.3.1.3 Ensure libpam-pwquality is installed (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

libpwquality provides common functions for password quality checking and scoring them based on their apparent randomness. The library also provides a function for generating random passwords with good pronounceability.

This module can be plugged into the password stack of a given service to provide some plug-in strength-checking for passwords. The code was originally based on **pam_cracklib** module and the module is backwards compatible with its options.

Rationale:

Strong passwords reduce the risk of systems being hacked through brute force methods.

Audit:

Run the following command to verify **libpam-pwquality** is installed:

```
# dpkg-query -s libpam-pwquality | grep -P -- '^(Status|Version)\b'
```

The output should be similar to:

```
Status: install ok installed
Version: 1.4.5-3+build1
```

Remediation:

Run the following command to install **libpam-pwquality**:

```
# apt install libpam-pwquality
```

References:

1. <https://packages.debian.org/buster/libpam-pwquality>

5.3.2 Configure pam-auth-update profiles

pam-auth-update is a utility that permits configuring the central authentication policy for the system using pre-defined profiles as supplied by PAM module packages.

Profiles - Shipped in the **/usr/share/pam-configs/** directory specify the modules, with options, to enable; the preferred ordering with respect to other profiles; and whether a profile should be enabled by default. Packages providing PAM modules register their profiles at install time by calling **pam-auth-update --package**.

Selection of profiles is done using the standard **debconf** interface. The profile selection question will be asked at **medium** priority when packages are added or removed, so no user interaction is required by default. Users may invoke **pam-auth-update** directly to change their authentication configuration.

The **pam-auth-update** script makes every effort to respect local changes to **/etc/pam.d/common-***. Local modifications to the list of module options will be preserved, and additions of modules within the managed portion of the stack will cause **pam-auth-update** to treat the config files as locally modified and not make further changes to the config files unless given the **--force** option.

If the user specifies that **pam-auth-update** should override local configuration changes, the locally-modified files will be saved in **/etc/pam.d/** with a suffix of **.pam-old**.

5.3.2.1 Ensure pam_unix module is enabled (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

pam_unix is the standard Unix authentication module. It uses standard calls from the system's libraries to retrieve and set account information as well as authentication. Usually this is obtained from the **/etc/passwd** and if shadow is enabled, the **/etc/shadow** file as well.

The account component performs the task of establishing the status of the user's account and password based on the following shadow elements: **expire**, **last_change**, **max_change**, **min_change**, **warn_change**. In the case of the latter, it may offer advice to the user on changing their password or, through the **PAM_AUTHTOKEN_REQD** return, delay giving service to the user until they have established a new password. The entries listed above are documented in the shadow(5) manual page. Should the user's record not contain one or more of these entries, the corresponding shadow check is not performed.

The authentication component performs the task of checking the users credentials (password). The default action of this module is to not permit the user access to a service if their official password is blank.

Rationale:

The system should only provide access after performing authentication of a user.

Audit:

Run the following command to verify that **pam_unix** is enabled:

```
# grep -P -- '\bpam_unix\.so\b' /etc/pam.d/common-{account,session,auth,password}
```

Output should be similar to:

```
/etc/pam.d/common-account:account    [success=1 new_authtok_reqd=done
default=ignore]    pam_unix.so
/etc/pam.d/common-session:session    required    pam_unix.so
/etc/pam.d/common-auth:auth    [success=2 default=ignore]    pam_unix.so
try_first_pass
/etc/pam.d/common-password:password    [success=1 default=ignore]
pam_unix.so obscure use_authtok try_first_pass yescrypt
```

Remediation:

Run the following command to enable the `pam_unix` module:







```
# pam-auth-update --enable unix
```

Note: If a site specific custom profile is being used in your environment to configure PAM that includes the configuration for the `pam_faillock` module, enable that module instead

References:

1. NIST SP 800-53 Rev. 5: IA-5(1)

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1110, T1110.001, T1110.002, T1110.003, T1178.001, T1178.002, T1178.003, T1178.004	TA0006	M1027

5.3.2.2 Ensure pam_faillock module is enabled (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The `pam_faillock.so` module maintains a list of failed authentication attempts per user during a specified interval and locks the account in case there were more than the configured number of consecutive failed authentications (this is defined by the `deny` parameter in the faillock configuration). It stores the failure records into per-user files in the tally directory.

Rationale:

Locking out user IDs after n unsuccessful consecutive login attempts mitigates brute force password attacks against your systems.

Audit:

Run the following commands to verify that `pam_faillock` is enabled:

```
# grep -P -- '\bpam_faillock\.so\b' /etc/pam.d/common-{auth,account}
```

Output should be similar to:

```
/etc/pam.d/common-auth:auth      requisite
pam_faillock.so preauth
/etc/pam.d/common-auth:auth      [default=die]
pam_faillock.so authfail
/etc/pam.d/common-account:account required
pam_faillock.so
```

Remediation:

Create two pam-auth-update profiles in `/usr/share/pam-configs/`:

1. Create the **faillock** profile in `/usr/share/pam-configs/` with the following lines:

```
Name: Enable pam_faillock to deny access
Default: yes
Priority: 0
Auth-Type: Primary
Auth:
    [default=die]                                pam_faillock.so authfail
```

Example Script:

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

{
    arr=('Name: Enable pam_faillock to deny access' 'Default: yes' 'Priority:
0' 'Auth-Type: Primary' 'Auth:' ' ' [default=die]
pam_faillock.so authfail')
    printf '%s\n' "${arr[@]}" > /usr/share/pam-configs/faillock
}
```

2. Create the **faillock_notify** profile in `/usr/share/pam-configs/` with the following lines:

```
Name: Notify of failed login attempts and reset count upon success
Default: yes
Priority: 1024
Auth-Type: Primary
Auth:
    requisite                                pam_faillock.so preauth
Account-Type: Primary
Account:
    required                                pam_faillock.so
```

Example Script:

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

{
    arr=('Name: Notify of failed login attempts and reset count upon success'
'Default: yes' 'Priority: 1024' 'Auth-Type: Primary' 'Auth:' ' '
requisite                                pam_faillock.so preauth' 'Account-Type:
Primary' 'Account:' ' ' required
pam_faillock.so')
    printf '%s\n' "${arr[@]}" > /usr/share/pam-configs/faillock_notify
}
```

Run the following command to update the **common-auth** and **common-account** PAM files with the new profiles:

```
# pam-auth-update --enable <profile_filename>
```






Example:

```
# pam-auth-update --enable faillock
# pam-auth-update --enable faillock_notify
```

Note:

- The name used for the file must be used in the **pam-auth-update --enable** command
- The **Name:** line should be easily recognizable and understood
- The **Priority:** Line is important as it effects the order of the lines in the **/etc/pam.d/** files
- If a site specific custom profile is being used in your environment to configure PAM that includes the configuration for the **pam_faillock** module, enable that module instead

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	6.2 Establish an Access Revoking Process Establish and follow a process, preferably automated, for revoking access to enterprise assets, through disabling accounts immediately upon termination, rights revocation, or role change of a user. Disabling accounts, instead of deleting accounts, may be necessary to preserve audit trails.			
v7	16.7 Establish Process for Revoking Access Establish and follow an automated process for revoking system access by disabling accounts immediately upon termination or change of responsibilities of an employee or contractor . Disabling these accounts, instead of deleting accounts, allows preservation of audit trails.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1110, T1110.001, T1110.003	TA0006	M1027

5.3.2.3 Ensure pam_pwquality module is enabled (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The **pam_pwquality.so** module performs password quality checking. This module can be plugged into the password stack of a given service to provide strength-checking for passwords. The code was originally based on pam_cracklib module and the module is backwards compatible with its options.

The action of this module is to prompt the user for a password and check its strength against a system dictionary and a set of rules for identifying poor choices.

The first action is to prompt for a single password, check its strength and then, if it is considered strong, prompt for the password a second time (to verify that it was typed correctly on the first occasion). All being well, the password is passed on to subsequent modules to be installed as the new authentication token.

Rationale:

Use of a unique, complex passwords helps to increase the time and resources required to compromise the password.

Audit:

Run the following command to verify that pam_pwhistory is enabled:

```
# grep -P -- '\bpam_pwquality\.so\b' /etc/pam.d/common-password
```

Output should be similar to:

```
password    requisite    pam_pwquality.so  retry=3
```

Remediation:

Run the following script to verify the **pam_pwquality.so** line exists in a **pam-auth-update** profile:

```
# grep -P -- '\bpam_pwquality\.so\b' /usr/share/pam-configs/*
```

Output should be similar to:


```
/usr/share/pam-configs/pwquality:      requisite
pam_pwquality.so retry=3
/usr/share/pam-configs/pwquality:      requisite
pam_pwquality.so retry=3
```

- **IF** - similar output is returned:

Run the following command to update `/etc/pam.d/common-password` with the returned profile:

```
# pam-auth-update --enable {PROFILE_NAME}
```

Example:

```
# pam-auth-update pwquality
```

- **IF** - similar output is **NOT** returned:

Create a pam-auth-update profile in `/usr/share/pam-configs/` with the following lines:

```
Name: Pwquality password strength checking
Default: yes
Priority: 1024
Conflicts: cracklib
Password-Type: Primary
Password:
    requisite                                pam_pwquality.so retry=3
```

Example:

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

{
    arr=('Name: Pwquality password strength checking' 'Default: yes'
'Priority: 1024' 'Conflicts: cracklib' 'Password-Type: Primary' 'Password:' '
requisite                                pam_pwquality.so retry=3')
    printf '%s\n' "${arr[@]}" > /usr/share/pam-configs/pwquality
}
```






Run the following command to update `/etc/pam.d/common-password` with the `pwquality` profile:

```
# pam-auth-update --enable pwquality
```

Note:

- The name used for the file must be used in the `pam-auth-update --enable` command
- The **Name:** line should be easily recognizable and understood
- The **Priority:** Line is important as it effects the order of the lines in the `/etc/pam.d/` files
- If a site specific custom profile is being used in your environment to configure PAM that includes the configuration for the `pam_pwquality` module, enable that module instead

CIS Controls:

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1110, T1110.001, T1110.002, T1110.003, T1178.001, T1178.002, T1178.003, T1178.004	TA0006	M1027

5.3.2.4 Ensure pam_pwhistory module is enabled (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The **pam_pwhistory.so** module saves the last passwords for each user in order to force password change history and keep the user from alternating between the same password too frequently.

This module does not work together with kerberos. In general, it does not make much sense to use this module in conjunction with **NIS** or **LDAP**, since the old passwords are stored on the local machine and are not available on another machine for password history checking.

Rationale:

Use of a unique, complex passwords helps to increase the time and resources required to compromise the password.

Audit:

Run the following command to verify that pam_pwhistory is enabled:

```
# grep -P -- '\bpam_pwhistory\.so\b' /etc/pam.d/common-password
```

Output should be similar to:

```
password    requisite    pam_pwhistory.so remember=24 enforce_for_root
try_first_pass use_authok
```

Remediation:

Run the following script to verify the **pam_pwquality.so** line exists in a **pam-auth-update** profile:

```
# grep -P -- '\bpam_pwhistory\.so\b' /usr/share/pam-configs/*
```

Output should be similar to:

```
/usr/share/pam-configs/pwhistory:    requisite    pam_pwhistory.so remember=24
enforce_for_root try_first_pass use_authok
```

- **IF** - similar output is returned:

Run the following command to update **/etc/pam.d/common-password** with the returned profile:

```
# pam-auth-update --enable {PROFILE_NAME}
```

Example:

```
# pam-auth-update pwhistory
```

- **IF** - similar output is **NOT** returned:

Create a **pwhistory** profile in **/usr/share/pam-configs/** with the following lines:

```
Name: pwhistory password history checking
Default: yes
Priority: 1024
Password-Type: Primary
Password: requisite pam_pwhistory.so remember=24 enforce_for_root
try_first_pass use_authok
```

Example Script:

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

{
    arr=('Name: pwhistory password history checking' 'Default: yes' 'Priority:
1024' 'Password-Type: Primary' 'Password:' '      requisite
pam_pwhistory.so remember=24 enforce_for_root try_first_pass use_authok')
    printf '%s\n' "${arr[@]}" > /usr/share/pam-configs/pwhistory
}
```

Run the following command to update **/etc/pam.d/common-password** with the **pwhistory** profile:

```
# pam-auth-update --enable pwhistory
```

Note:

- The name used for the file must be used in the **pam-auth-update --enable** command
- The **Name:** line should be easily recognizable and understood
- The **Priority:** Line is important as it effects the order of the lines in the **/etc/pam.d/** files
- If a site specific custom profile is being used in your environment to configure PAM that includes the configuration for the **pam_pwhistory** module, enable that module instead

CIS Controls:

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1110, T1110.001, T1110.002, T1110.003, T1178.001, T1178.002, T1178.003, T1178.004	TA0006	M1027

5.3.3 Configure PAM Arguments

Pluggable Authentication Modules (PAM) uses arguments to pass information to a pluggable module during authentication for a particular module type. These arguments allow the PAM configuration files for particular programs to use a common PAM module but in different ways.

Invalid arguments are ignored and do not otherwise affect the success or failure of the PAM module. When an invalid argument is passed, an error is usually written to `/var/log/messages` file. However, since the reporting method is controlled by the PAM module, the module must be written correctly to log the error to this file.

Note: If custom PAM files are being used, for this section's remediation, the corresponding files in `/etc/pam.d/` would need to be edited directly, and the `pam-auth-update --enable <EDITED_PROFILE_NAME>` command skipped

5.3.3.1 Configure pam_faillock module

`pam_faillock.so` provides a way to configure the default settings for locking the user after multiple failed authentication attempts.

Options:

- `<dir=/path/to/tally-directory>` - The directory where the user files with the failure records are kept. The default is `/var/run/faillock`. Note: These files will disappear after reboot on systems configured with directory `/var/run/faillock` mounted on virtual memory.
- `audit` - Will log the user name into the system log if the user is not found.
- `silent` - Don't print informative messages to the user. Please note that when this option is not used there will be difference in the authentication behavior for users which exist on the system and non-existing users.
- `no_log_info` - Don't log informative messages via syslog(3).
- `local_users_only` - Only track failed user authentications attempts for local users in `/etc/passwd` and ignore centralized (AD, IdM, LDAP, etc.) users. The `faillock(8)` command will also no longer track user failed authentication attempts. Enabling this option will prevent a double-lockout scenario where a user is locked out locally and in the centralized mechanism.
- `nodelay` - Don't enforce a delay after authentication failures.
- `deny=<n>` - Deny access if the number of consecutive authentication failures for this user during the recent interval exceeds `n`. The default is 3.
- `fail_interval=n` - The length of the interval during which the consecutive authentication failures must happen for the user account lock out is `n` seconds. The default is 900 (15 minutes).
- `unlock_time=n` - The access will be re-enabled after `n` seconds after the lock out. The value 0 has the same meaning as value never - the access will not be re-enabled without resetting the faillock entries by the `faillock(8)` command. The default is 600 (10 minutes). Note that the default directory that `pam_faillock` uses is usually cleared on system boot so the access will be also re-enabled after system reboot. If that is undesirable a different tally directory must be set with the `dir` option. Also note that it is usually undesirable to permanently lock out users as they can become easily a target of denial of service attack unless the usernames are random and kept secret to potential attackers.
- `even_deny_root` - Root account can become locked as well as regular accounts.
- `root_unlock_time=n` - This option implies `even_deny_root` option. Allow access after `n` seconds to root account after the account is locked. In case the option is not specified the value is the same as of the `unlock_time` option.
- `admin_group=name` - If a group name is specified with this option, members of the group will be handled by this module the same as the root account (the options `even_deny_root` and `root_unlock_time` will apply to them. By default the option is not set.

5.3.3.1.1 Ensure password failed attempts lockout is configured (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The **deny=<n>** option will deny access if the number of consecutive authentication failures for this user during the recent interval exceeds .

Rationale:

Locking out user IDs after *n* unsuccessful consecutive login attempts mitigates brute force password attacks against your systems.

Audit:

Run the following command to verify that Number of failed logon attempts before the account is locked is no greater than **5** and meets local site policy:

```
# grep -Pi -- '^\\h*deny\\h*=\\h*[1-5]\\b' /etc/security/faillock.conf
deny = 5
```

Run the following command to verify that the **deny** argument has not been set, or **5** or less and meets local site policy:

```
# grep -Pi -- '^\\h*auth\\h+(requisite|required|sufficient)\\h+pam_faillock\\.so\\h+([\\^#\\n\\r]+\\h+)?deny\\h*=\\h*(0|[6-9]|[1-9][0-9]+)\\b' /etc/pam.d/common-auth
```

Nothing should be returned

Remediation:

Create or edit the following line in **/etc/security/faillock.conf** setting the **deny** option to **5** or less:

```
deny = 5
```

Run the following command:

```
# grep -Pl -- '\\bpam_faillock\\.so\\h+([\\^#\\n\\r]+\\h+)?deny\\b' /usr/share/pam-configs/*
```

Edit any returned files and remove the **deny=<N>** arguments from the **pam_faillock.so** line(s):






Default Value:

deny = 3

Additional Information:

If a user has been locked out because they have reached the maximum consecutive failure count defined by **deny=** in the **pam_faillock.so** module, the user can be unlocked by issuing the command **faillock --user <USERNAME> --reset**. This command sets the failed count to 0, effectively unlocking the user.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	6.2 Establish an Access Revoking Process Establish and follow a process, preferably automated, for revoking access to enterprise assets, through disabling accounts immediately upon termination, rights revocation, or role change of a user. Disabling accounts, instead of deleting accounts, may be necessary to preserve audit trails.			
v7	16.7 Establish Process for Revoking Access Establish and follow an automated process for revoking system access by disabling accounts immediately upon termination or change of responsibilities of an employee or contractor . Disabling these accounts, instead of deleting accounts, allows preservation of audit trails.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1110, T1110.001, T1110.003	TA0006	M1027

5.3.3.1.2 Ensure password unlock time is configured (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

`unlock_time=<n>` - The access will be re-enabled after seconds after the lock out. The value `0` has the same meaning as value never - the access will not be re-enabled without resetting the faillock entries by the `faillock(8)` command.

Note:

- The default directory that `pam_faillock` uses is usually cleared on system boot so the access will be also re-enabled after system reboot. If that is undesirable a different tally directory must be set with the `dir` option.
- It is usually undesirable to permanently lock out users as they can become easily a target of denial of service attack unless the usernames are random and kept secret to potential attackers.
- The maximum configurable value for `unlock_time` is `604800`

Rationale:

Locking out user IDs after *n* unsuccessful consecutive login attempts mitigates brute force password attacks against your systems.

Impact:

Use of `unlock_time=0` may allow an attacker to cause denial of service to legitimate users. This will also require a systems administrator with elevated privileges to unlock the account.

Audit:

Run the following command to verify that the time in seconds before the account is unlocked is either 0 (never) or 900 (15 minutes) or more and meets local site policy:

```
# grep -Pi -- '^h*unlock_timeh*=\h*(0|9[0-9][0-9]|[1-9][0-9]{3,})\b'
/etc/security/faillock.conf

unlock_time = 900
```

Run the following command to verify that the `unlock_time` argument has not been set, or is either 0 (never) or 900 (15 minutes) or more and meets local site policy:

```
# grep -Pi --
'^h*authh+(requisite|required|sufficient)\h+pam_faillock\.so\h+([\#\n\r]+\h
+)?unlock_timeh*=\h*([1-9]|[1-9][0-9]|[1-8][0-9][0-9])\b' /etc/pam.d/common-
auth
```

Nothing should be returned

Remediation:

Set password unlock time to conform to site policy. `unlock_time` should be 0 (never), or 900 seconds or greater.

Edit `/etc/security/faillock.conf` and update or add the following line:

```
unlock_time = 900
```

Run the following command: remove the `unlock_time` argument from the `pam_faillock.so` module in the PAM files:

```
# grep -Pl -- '\bpam_faillock\.so\h+([\#\n\r]+\h+)?unlock_time\b'
/usr/share/pam-configs/*
```

Edit any returned files and remove the `unlock_time=<N>` argument from the `pam_faillock.so` line(s):






Default Value:

```
unlock_time = 600
```

Additional Information:

If a user has been locked out because they have reached the maximum consecutive failure count defined by `deny=` in the `pam_faillock.so` module, the user can be unlocked by issuing the command `faillock --user <USERNAME> --reset`. This command sets the failed count to 0, effectively unlocking the user.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	6.2 <u>Establish an Access Revoking Process</u> Establish and follow a process, preferably automated, for revoking access to enterprise assets, through disabling accounts immediately upon termination, rights revocation, or role change of a user. Disabling accounts, instead of deleting accounts, may be necessary to preserve audit trails.			
v7	16.7 <u>Establish Process for Revoking Access</u> Establish and follow an automated process for revoking system access by disabling accounts immediately upon termination or change of responsibilities of an employee or contractor . Disabling these accounts, instead of deleting accounts, allows preservation of audit trails.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1110, T1110.001, T1110.003	TA0006	M1027

5.3.3.1.3 Ensure password failed attempts lockout includes root account (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

`even_deny_root` - Root account can become locked as well as regular accounts

`root_unlock_time=n` - This option implies `even_deny_root` option. Allow access after n seconds to root account after the account is locked. In case the option is not specified the value is the same as of the `unlock_time` option.

Rationale:

Locking out user IDs after n unsuccessful consecutive login attempts mitigates brute force password attacks against your systems.

Impact:

Use of `unlock_time=0` or `root_unlock_time=0` may allow an attacker to cause denial of service to legitimate users.

Audit:

Run the following command to verify that **even_deny_root** and/or **root_unlock_time** is enabled:

```
# grep -Pi -- '^\\h*(even_deny_root|root_unlock_time\\h*=\\h*\\d+)\\b' /etc/security/faillock.conf
```

Example output:

```
even_deny_root

--AND/OR--

root_unlock_time = 60
```

Run the following command to verify that - **IF** - **root_unlock_time** is set, it is set to **60** (One minute) or more:

```
# grep -Pi -- '^\\h*root_unlock_time\\h*=\\h*([1-9]|[1-5][0-9])\\b' /etc/security/faillock.conf
```

Nothing should be returned

Run the following command to check the **pam_faillock.so** module for the **root_unlock_time** argument. Verify -**IF**- **root_unlock_time** is set, it is set to **60** (One minute) or more:

```
# grep -Pi -- '^\\h*auth\\h+([\\^#\\n\\r]+\\h+)pam_faillock\\.so\\h+([\\^#\\n\\r]+\\h+)?root_unlock_time\\h*=\\h*([1-9]|[1-5][0-9])\\b' /etc/pam.d/common-auth
```

Nothing should be returned

Remediation:

Edit **/etc/security/faillock.conf**:

- Remove or update any line containing **root_unlock_time**, - **OR** - set it to a value of **60** or more
- Update or add the following line:

```
even_deny_root
```

Run the following command:

```
# grep -Pl -- '\\bpam_faillock\\.so\\h+([\\^#\\n\\r]+\\h+)?(even_deny_root|root_unlock_time)' /usr/share/pam-configs/*
```

Edit any returned files and remove the **even_deny_root** and **root_unlock_time** arguments from the **pam_faillock.so** line(s):






Default Value:

disabled

Additional Information:

If a user has been locked out because they have reached the maximum consecutive failure count defined by `deny=` in the `pam_faillock.so` module, the user can be unlocked by issuing the command `faillock --user <USERNAME> --reset`. This command sets the failed count to 0, effectively unlocking the user.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	6.2 Establish an Access Revoking Process Establish and follow a process, preferably automated, for revoking access to enterprise assets, through disabling accounts immediately upon termination, rights revocation, or role change of a user. Disabling accounts, instead of deleting accounts, may be necessary to preserve audit trails.			
v7	16.7 Establish Process for Revoking Access Establish and follow an automated process for revoking system access by disabling accounts immediately upon termination or change of responsibilities of an employee or contractor. Disabling these accounts, instead of deleting accounts, allows preservation of audit trails.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1110, T1110.001, T1110.003	TA0006	M1027

5.3.3.2 Configure pam_pwquality module

The `pam_pwquality.so` module checks the strength of passwords. It performs checks such as making sure a password is not a dictionary word, it is a certain length, contains a mix of characters (e.g. alphabet, numeric, other) and more.

These checks are configurable by either:

- use of the module arguments
- modifying the `/etc/security/pwquality.conf` configuration file
- creating a `.conf` file in the `/etc/security/pwquality.conf.d/` directory.

Note: The module arguments override the settings in the `/etc/security/pwquality.conf` configuration file. Settings in the `/etc/security/pwquality.conf` configuration file override settings in a `.conf` file in the `/etc/security/pwquality.conf.d/` directory.

The possible options in the file are:

- `difok` - Number of characters in the new password that must not be present in the old password. (default 1). The special value of 0 disables all checks of similarity of the new password with the old password except the new password being exactly the same as the old one.
- `minlen` - Minimum acceptable size for the new password (plus one if credits are not disabled which is the default). (See `pam_pwquality(8)`.) Cannot be set to lower value than 6. (default 8)
- `dcredit` - The maximum credit for having digits in the new password. If less than 0 it is the minimum number of digits in the new password. (default 0)
- `uccredit` - The maximum credit for having uppercase characters in the new password. If less than 0 it is the minimum number of uppercase characters in the new password. (default 0)
- `lcredit` - The maximum credit for having lowercase characters in the new password. If less than 0 it is the minimum number of lowercase characters in the new password. (default 0)
- `ocredit` - The maximum credit for having other characters in the new password. If less than 0 it is the minimum number of other characters in the new password. (default 0)
- `minclass` - The minimum number of required classes of characters for the new password (digits, uppercase, lowercase, others). (default 0)
- `maxrepeat` - The maximum number of allowed same consecutive characters in the new password. The check is disabled if the value is 0. (default 0)
- `maxsequence` - The maximum length of monotonic character sequences in the new password. Examples of such sequence are '12345' or 'fedcb'. Note that most such passwords will not pass the simplicity check unless the sequence is only a minor part of the password. The check is disabled if the value is 0. (default 0)

- **maxclassrepeat** - The maximum number of allowed consecutive characters of the same class in the new password. The check is disabled if the value is 0. (default 0)
- **gecoscheck** - If nonzero, check whether the words longer than 3 characters from the GECOS field of the user's passwd(5) entry are contained in the new password. The check is disabled if the value is 0. (default 0)
- **dictcheck** - If nonzero, check whether the password (with possible modifications) matches a word in a dictionary. Currently the dictionary check is performed using the cracklib library. (default 1)
- **usercheck=<N>** - If nonzero, check whether the password (with possible modifications) contains the user name in some form. It is not performed for user names shorter than 3 characters. (default 1)
- **usersubstr=<N>** - If greater than 3 (due to the minimum length in usercheck), check whether the password contains a substring of at least N length in some form. (default 0)
- **enforcing=<N>** - If nonzero, reject the password if it fails the checks, otherwise only print the warning. This setting applies only to the pam_pwquality module and possibly other applications that explicitly change their behavior based on it. It does not affect pwmake(1) and pwscore(1). (default 1)
- **badwords** - Space separated list of words that must not be contained in the password. These are additional words to the cracklib dictionary check. This setting can be also used by applications to emulate the gecos check for user accounts that are not created yet.
- **dictpath** - Path to the cracklib dictionaries. Default is to use the cracklib default.
- **retry=<N>** - Prompt user at most N times before returning with error. The default is 1.
- **enforce_for_root** - The module will return error on failed check even if the user changing the password is root. This option is off by default which means that just the message about the failed check is printed but root can change the password anyway. Note that root is not asked for an old password so the checks that compare the old and new password are not performed.
- **local_users_only** - The module will not test the password quality for users that are not present in the /etc/passwd file. The module still asks for the password so the following modules in the stack can use the use_authok option. This option is off by default.

5.3.3.2.1 Ensure password number of changed characters is configured (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The `pwquality difok` option sets the number of characters in a password that must not be present in the old password.

Rationale:

Use of a complex password helps to increase the time and resources required to compromise the password. Password complexity, or strength, is a measure of the effectiveness of a password in resisting attempts at guessing and brute-force attacks.

Password complexity is one factor of several that determines how long it takes to crack a password. The more complex the password, the greater the number of possible combinations that need to be tested before the password is compromised.

Audit:

Run the following command to verify that the **difok** option is set to **2** or more and follows local site policy:

```
# grep -Psi -- '^\\h*difok\\h*==\\h*([2-9]|[1-9][0-9]+)\\b'
/etc/security/pwquality.conf /etc/security/pwquality.conf.d/*.conf
```

Example output:

```
/etc/security/pwquality.conf.d/50-pwdifok.conf:difok = 2
```

Verify returned value(s) are **2** or more and meet local site policy

Run the following command to verify that **difok** is not set, is **2** or more, and conforms to local site policy:

```
# grep -Psi --
'^\\h*password\\h+(requisite|required|sufficient)\\h+pam_pwquality\\.so\\h+([^\n\r]+\\h+)?difok\\h*==\\h*([0-1])\\b' /etc/pam.d/common-password
```

Nothing should be returned

Note:

- settings should be configured in only **one** location for clarity
- Settings observe an order of precedence:
 - module arguments override the settings in the **/etc/security/pwquality.conf** configuration file
 - settings in the **/etc/security/pwquality.conf** configuration file override settings in a **.conf** file in the **/etc/security/pwquality.conf.d/** directory
 - settings in a **.conf** file in the **/etc/security/pwquality.conf.d/** directory are read in canonical order, with last read file containing the setting taking precedence
- It is recommended that settings be configured in a **.conf** file in the **/etc/security/pwquality.conf.d/** directory for clarity, convenience, and durability.

Remediation:

Create or modify a file ending in **.conf** in the **/etc/security/pwquality.conf.d/** directory or the file **/etc/security/pwquality.conf** and add or modify the following line to set **difok** to **2** or more. Ensure setting conforms to local site policy:

Example:

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

{
    sed -ri 's/^\s*difok\s*=/# &/' /etc/security/pwquality.conf
    [ ! -d /etc/security/pwquality.conf.d/ ] && mkdir
    /etc/security/pwquality.conf.d/
    printf '\n%s' "difok = 2" > /etc/security/pwquality.conf.d/50-pwdifok.conf
}
```

Run the following command:

```
# grep -Pl -- '\bpam_pwquality\.so\h+([\^#\n\r]+\h+)?difok\b' /usr/share/pam-
configs/*
```

Edit any returned files and remove the **difok** argument from the **pam_pwquality.so** line(s):






Default Value:

difok = 1

References:

1. NIST SP 800-53 Rev. 5: IA-5

CIS Controls:

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1110, T1110.001, T1110.002, T1110.003, T1178.001, T1178.002, T1178.003, T1178.004	TA0006	M1027

5.3.3.2.2 Ensure minimum password length is configured (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The minimum password length setting determines the lowest number of characters that make up a password for a user account. There are many different theories about how to determine the best password length for an organization, but perhaps "passphrase" is a better term than "password".

The **minlen** option sets the minimum acceptable size for the new password (plus one if credits are not disabled which is the default). Cannot be set to lower value than 6.

Rationale:

Strong passwords help protect systems from password attacks. Types of password attacks include dictionary attacks, which attempt to use common words and phrases, and brute force attacks, which try every possible combination of characters. Also attackers may try to obtain the account database so they can use tools to discover the accounts and passwords.

Impact:

In general, it is true that longer passwords are better (harder to crack), but it is also true that forced password length requirements can cause user behavior that is predictable and undesirable. For example, requiring users to have a minimum 16-character password may cause them to choose repeating patterns like fourfourfourfour or passwordpassword that meet the requirement but aren't hard to guess. Additionally, length requirements increase the chances that users will adopt other insecure practices, like writing them down, re-using them or storing them unencrypted in their documents.

Having a reasonable minimum length with no maximum character limit increases the resulting average password length used (and therefore the strength).⁶

Audit:

Run the following command to verify that password length is **14** or more characters, and conforms to local site policy:

```
# grep -Psi -- '^h*minlen\h*=\h*(1[4-9]|[2-9][0-9]|[1-9][0-9]{2,})\b'
/etc/security/pwquality.conf /etc/security/pwquality.conf.d/*.conf
```

Example output:

```
/etc/security/pwquality.conf.d/50-pwlength.conf:minlen = 14
```

Verify returned value(s) are no less than **14** characters and meet local site policy

Run the following command to verify that **minlen** is not set, or is **14** or more characters, and conforms to local site policy:

```
# grep -Psi --
'^h*password\h+(requisite|required|sufficient)\h+pam_pwquality\.so\h+([\^#\n\
r]+\h+)?minlen\h*=\h*([0-9]|1[0-3])\b' /etc/pam.d/system-auth
/etc/pam.d/common-password
```

Nothing should be returned

Note:

- settings should be configured in only **one** location for clarity
- Settings observe an order of precedence:
 - module arguments override the settings in the **/etc/security/pwquality.conf** configuration file
 - settings in the **/etc/security/pwquality.conf** configuration file override settings in a **.conf** file in the **/etc/security/pwquality.conf.d/** directory
 - settings in a **.conf** file in the **/etc/security/pwquality.conf.d/** directory are read in canonical order, with last read file containing the setting taking precedence
- It is recommended that settings be configured in a **.conf** file in the **/etc/security/pwquality.conf.d/** directory for clarity, convenience, and durability.

Remediation:

Create or modify a file ending in **.conf** in the **/etc/security/pwquality.conf.d/** directory or the file **/etc/security/pwquality.conf** and add or modify the following line to set password length of **14** or more characters. Ensure that password length conforms to local site policy:

Example:

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

{
    sed -ri 's/^\s*minlen\s*=/# &/' /etc/security/pwquality.conf
    [ ! -d /etc/security/pwquality.conf.d/ ] && mkdir
    /etc/security/pwquality.conf.d/
    printf '\n%s' "minlen = 14" > /etc/security/pwquality.conf.d/50-
    pwlength.conf
}
```

Run the following command:

```
# grep -Pl -- '\bpam_pwquality\.so\h+([\#\n\r]+\h+)?minlen\b' /usr/share/pam-
configs/*
```

Edit any returned files and remove the **minlen** argument from the **pam_pwquality.so** line(s):






Default Value:

minlen = 8

References:

1. pam_pwquality(8)
2. CIS Password Policy Guide
3. NIST SP 800-53 Rev. 5: IA-5(1)

CIS Controls:

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1110, T1110.001, T1110.002, T1110.003, T1178.001, T1178.002, T1178.003, T1178.004	TA0006	M1027

5.3.3.2.3 Ensure password complexity is configured (Manual)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

Password complexity can be set through:

- **minclass** - The minimum number of classes of characters required in a new password. (digits, uppercase, lowercase, others). e.g. **minclass = 4** requires digits, uppercase, lower case, and special characters.
- **dcredit** - The maximum credit for having digits in the new password. If less than 0 it is the minimum number of digits in the new password. e.g. **dcredit = -1** requires at least one digit
- **ucredit** - The maximum credit for having uppercase characters in the new password. If less than 0 it is the minimum number of uppercase characters in the new password. e.g. **ucredit = -1** requires at least one uppercase character
- **ocredit** - The maximum credit for having other characters in the new password. If less than 0 it is the minimum number of other characters in the new password. e.g. **ocredit = -1** requires at least one special character
- **lcredit** - The maximum credit for having lowercase characters in the new password. If less than 0 it is the minimum number of lowercase characters in the new password. e.g. **lcredit = -1** requires at least one lowercase character

Rationale:

Strong passwords protect systems from being hacked through brute force methods.

Requiring at least one non-alphabetic character increases the search space beyond pure dictionary words, which makes the resulting password harder to crack.

Forcing users to choose an excessively complex password, e.g. some combination of upper-case, lower-case, numbers, and special characters, has a negative impact. It places an extra burden on users and many will use predictable patterns (for example, a capital letter in the first position, followed by lowercase letters, then one or two numbers, and a “special character” at the end). Attackers know this, so dictionary attacks will often contain these common patterns and use the most common substitutions like, \$ for s, @ for a, 1 for l, 0 for o.

Impact:

Passwords that are too complex in nature make it harder for users to remember, leading to bad practices. In addition, composition requirements provide no defense against common attack types such as social engineering or insecure storage of passwords

Audit:

Run the following command to verify:

- **dcredit**, **ucredit**, **lcredit**, and **ocredit** are not set to a value greater than 0
- Complexity conforms to local site policy:

```
# grep -Psi -- '^\\h*(minclass|[dulo]credit)\\b' /etc/security/pwquality.conf /etc/security/pwquality.conf.d/*.conf
```

Example output:

```
/etc/security/pwquality.conf.d/50-pwcomplexity.conf:minclass = 3
/etc/security/pwquality.conf.d/50-pwcomplexity.conf:ucredit = -2
/etc/security/pwquality.conf.d/50-pwcomplexity.conf:lcredit = -2
/etc/security/pwquality.conf.d/50-pwcomplexity.conf:dcredit = -1
/etc/security/pwquality.conf.d/50-pwcomplexity.conf:ocredit = 0
```

The example represents a requirement of three character classes, with passwords requiring two upper case, two lower case, and one numeric character.

Run the following command to verify that module arguments in the configuration file(s) are not being overridden by arguments in **/etc/pam.d/common-password**:

```
# grep -Psi -- '^\\h*password\\h+(requisite|required|sufficient)\\h+pam_pwquality\\.so\\h+([^#\\n\\r]+\\h+)?(minclass=\\d*|[dulo]credit=-?\\d*)\\b' /etc/pam.d/common-password
```

Nothing should be returned

Note:

- settings should be configured in only **one** location for clarity
- Settings observe an order of precedence:
 - module arguments override the settings in the **/etc/security/pwquality.conf** configuration file
 - settings in the **/etc/security/pwquality.conf** configuration file override settings in a **.conf** file in the **/etc/security/pwquality.conf.d/** directory
 - settings in a **.conf** file in the **/etc/security/pwquality.conf.d/** directory are read in canonical order, with last read file containing the setting taking precedence
- It is recommended that settings be configured in a **.conf** file in the **/etc/security/pwquality.conf.d/** directory for clarity, convenience, and durability.

Remediation:

Run the following command:

```
# grep -Pl --
'\bpam_pwquality\.so\h+([\^#\n\r]+\h+)?(minclass|[dulo]credit)\b'
/usr/share/pam-configs/*
```

Edit any returned files and remove the **minclass**, **dcredit**, **ucredit**, **lcredit**, and **ocredit** arguments from the **pam_pwquality.so** line(s)

Create or modify a file ending in **.conf** in the **/etc/security/pwquality.conf.d/** directory or the file **/etc/security/pwquality.conf** and add or modify the following line(s) to set complexity according to local site policy:

- **minclass** = **_N_**
- **dcredit** = **_N_** # Value should be either **0** or a number proceeded by a minus (-) symbol
- **ucredit** = **-1** # Value should be either **0** or a number proceeded by a minus (-) symbol
- **ocredit** = **-1** # Value should be either **0** or a number proceeded by a minus (-) symbol
- **lcredit** = **-1** # Value should be either **0** or a number proceeded by a minus (-) symbol

Example 1 - Set **minclass** = 3:

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

{
    sed -ri 's/^\s*minclass\s*=/# &/' /etc/security/pwquality.conf
    sed -ri 's/^\s*[dulo]credit\s*=/# &/' /etc/security/pwquality.conf
    [ ! -d /etc/security/pwquality.conf.d/ ] && mkdir
    /etc/security/pwquality.conf.d/
    printf '\n%s' "minclass = 3" > /etc/security/pwquality.conf.d/50-
    pwcomplexity.conf
}
```

Example 2 - set **dcredit** = -1, **ucredit** = -1, and **lcredit** = -1:

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

{
    sed -ri 's/^\s*minclass\s*=/# &/' /etc/security/pwquality.conf
    sed -ri 's/^\s*[dulo]credit\s*=/# &/' /etc/security/pwquality.conf
    [ ! -d /etc/security/pwquality.conf.d/ ] && mkdir
    /etc/security/pwquality.conf.d/
    printf '%s\n' "dcredit = -1" "ucredit = -1" "lcredit = -1" >
    /etc/security/pwquality.conf.d/50-pwcomplexity.conf
}
```

Default Value:

minclass = 0

dcredit = 0

ucredit = 0






ocredit = 0

lcredit = 0

References:

1. pam_pwquality(8)
2. PWQUALITY.CONF(5)
3. <https://www.cisecurity.org/insights/white-papers/cis-password-policy-guide>
4. NIST SP 800-53 Rev. 5: IA-5

CIS Controls:

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1110, T1110.001, T1110.002, T1110.003, T1178.001, T1178.002, T1178.003, T1178.004	TA0006	M1027

5.3.3.2.4 Ensure password same consecutive characters is configured (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The `pwquality maxrepeat` option sets the maximum number of allowed same consecutive characters in a new password.

Rationale:

Use of a complex password helps to increase the time and resources required to compromise the password. Password complexity, or strength, is a measure of the effectiveness of a password in resisting attempts at guessing and brute-force attacks.

Password complexity is one factor of several that determines how long it takes to crack a password. The more complex the password, the greater the number of possible combinations that need to be tested before the password is compromised.

Audit:

Run the following command to verify that the **maxrepeat** option is set to **3** or less, not **0**, and follows local site policy:

```
# grep -Psi -- '^h*maxrepeat\h*=\h*[1-3]\b' /etc/security/pwquality.conf
/etc/security/pwquality.conf.d/*.conf
```

Example output:

```
/etc/security/pwquality.conf.d/50-pwrepeat.conf:maxrepeat = 3
```

Verify returned value(s) are **3** or less, not **0**, and meet local site policy

Run the following command to verify that **maxrepeat** is not set, is **3** or less, not **0**, and conforms to local site policy:

```
# grep -Psi --
'^h*password\h+(requisite|required|sufficient)\h+pam_pwquality\.so\h+([\^#\n\
r]+\h+)?maxrepeat\h*=\h*(0|[4-9]|[1-9][0-9]+)\b' /etc/pam.d/common-password
```

Nothing should be returned

Note:

- settings should be configured in only **one** location for clarity
- Settings observe an order of precedence:
 - module arguments override the settings in the **/etc/security/pwquality.conf** configuration file
 - settings in the **/etc/security/pwquality.conf** configuration file override settings in a **.conf** file in the **/etc/security/pwquality.conf.d/** directory
 - settings in a **.conf** file in the **/etc/security/pwquality.conf.d/** directory are read in canonical order, with last read file containing the setting taking precedence
- It is recommended that settings be configured in a **.conf** file in the **/etc/security/pwquality.conf.d/** directory for clarity, convenience, and durability.

Remediation:

Create or modify a file ending in **.conf** in the **/etc/security/pwquality.conf.d/** directory or the file **/etc/security/pwquality.conf** and add or modify the following line to set **maxrepeat** to **3** or less and not **0**. Ensure setting conforms to local site policy:

Example:

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

{
    sed -ri 's/^\s*maxrepeat\s*=/# &/' /etc/security/pwquality.conf
    [ ! -d /etc/security/pwquality.conf.d/ ] && mkdir
    /etc/security/pwquality.conf.d/
    printf '\n%s' "maxrepeat = 3" > /etc/security/pwquality.conf.d/50-
    pwrepeat.conf
}
```

Run the following command:

```
# grep -Pl -- '\bpam_pwquality\.so\h+([\^#\n\r]+\h+)?maxrepeat\b'
/usr/share/pam-configs/*
```

Edit any returned files and remove the **maxrepeat** argument from the **pam_pwquality.so** line(s):






Default Value:

maxrepeat = 0

References:

1. NIST SP 800-53 Rev. 5: IA-5

CIS Controls:

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1110, T1110.001, T1110.002, T1110.003, T1178.001, T1178.002, T1178.003, T1178.004	TA0006	M1027

5.3.3.2.5 Ensure password maximum sequential characters is configured (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The `pwquality maxsequence` option sets the maximum length of monotonic character sequences in the new password. Examples of such sequence are `12345` or `fedcb`. The check is disabled if the value is `0`.

Note: Most such passwords will not pass the simplicity check unless the sequence is only a minor part of the password.

Rationale:

Use of a complex password helps to increase the time and resources required to compromise the password. Password complexity, or strength, is a measure of the effectiveness of a password in resisting attempts at guessing and brute-force attacks.

Password complexity is one factor of several that determines how long it takes to crack a password. The more complex the password, the greater the number of possible combinations that need to be tested before the password is compromised.

Audit:

Run the following command to verify that the **maxsequence** option is set to **3** or less, not **0**, and follows local site policy:

```
# grep -Psi -- '^h*maxsequenceh*=\h*[1-3]\b' /etc/security/pwquality.conf
/etc/security/pwquality.conf.d/*.conf
```

Example output:

```
/etc/security/pwquality.conf.d/50-pwmaxsequence.conf:maxsequence = 3
```

Verify returned value(s) are **3** or less, not **0**, and meet local site policy

Run the following command to verify that **maxsequence** is not set, is **3** or less, not **0**, and conforms to local site policy:

```
# grep -Psi --
'^h*passwordh+(requisite|required|sufficient)\h+pam_pwquality\.so\h+([\^#\n\r]+\h+)?maxsequenceh*=\h*(0|[4-9]|[1-9][0-9]+)\b' /etc/pam.d/common-password
```

Nothing should be returned

Note:

- settings should be configured in only **one** location for clarity
- Settings observe an order of precedence:
 - module arguments override the settings in the **/etc/security/pwquality.conf** configuration file
 - settings in the **/etc/security/pwquality.conf** configuration file override settings in a **.conf** file in the **/etc/security/pwquality.conf.d/** directory
 - settings in a **.conf** file in the **/etc/security/pwquality.conf.d/** directory are read in canonical order, with last read file containing the setting taking precedence
- It is recommended that settings be configured in a **.conf** file in the **/etc/security/pwquality.conf.d/** directory for clarity, convenience, and durability.

Remediation:

Create or modify a file ending in **.conf** in the **/etc/security/pwquality.conf.d/** directory or the file **/etc/security/pwquality.conf** and add or modify the following line to set **maxsequence** to **3** or less and not **0**. Ensure setting conforms to local site policy:

Example:

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

{
    sed -ri 's/^\s*maxsequence\s*=/# &/' /etc/security/pwquality.conf
    [ ! -d /etc/security/pwquality.conf.d/ ] && mkdir
    /etc/security/pwquality.conf.d/
    printf '\n%s' "maxsequence = 3" > /etc/security/pwquality.conf.d/50-
    pwmaxsequence.conf
}
```

Run the following command:

```
# grep -Pl -- '\bpam_pwquality\.so\h+([\^#\n\r]+\h+)?maxsequence\b'
/usr/share/pam-configs/*
```

Edit any returned files and remove the **maxsequence** argument from the **pam_pwquality.so** line(s):






Default Value:

maxsequence = 0

References:

1. NIST SP 800-53 Rev. 5: IA-5

CIS Controls:

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1110, T1110.001, T1110.002, T1110.003, T1178.001, T1178.002, T1178.003, T1178.004	TA0006	M1027

5.3.3.2.6 Ensure password dictionary check is enabled (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The `pwquality dictcheck` option sets whether to check for the words from the `cracklib` dictionary.

Rationale:

If the operating system allows the user to select passwords based on dictionary words, this increases the chances of password compromise by increasing the opportunity for successful guesses, and brute-force attacks.

Audit:

Run the following command to verify that the **dictcheck** option is not set to **0** (disabled) in a pwquality configuration file:

```
# grep -Psi -- '^h*dictcheck\h*=\h*0\b' /etc/security/pwquality.conf
/etc/security/pwquality.conf.d/*.conf
```

Nothing should be returned

Run the following command to verify that the **dictcheck** option is not set to **0** (disabled) as a module argument in a PAM file:

```
# grep -Psi --
'^h*password\h+(requisite|required|sufficient)\h+pam_pwquality\.so\h+([\^#\n\r]+\h+)?dictcheck\h*=\h*0\b' /etc/pam.d/common-password
```

Nothing should be returned

Note:

- Settings observe an order of precedence:
 - module arguments override the settings in the **/etc/security/pwquality.conf** configuration file
 - settings in the **/etc/security/pwquality.conf** configuration file override settings in a **.conf** file in the **/etc/security/pwquality.conf.d/** directory
 - settings in a **.conf** file in the **/etc/security/pwquality.conf.d/** directory are read in canonical order, with last read file containing the setting taking precedence
- It is recommended that settings be configured in a **.conf** file in the **/etc/security/pwquality.conf.d/** directory for clarity, convenience, and durability.

Remediation:

Edit any file ending in **.conf** in the **/etc/security/pwquality.conf.d/** directory and/or the file **/etc/security/pwquality.conf** and comment out or remove any instance of **dictcheck = 0**:

Example:

```
# sed -ri 's/^\s*dictcheck\s*=/# &/' /etc/security/pwquality.conf
/etc/security/pwquality.conf.d/*.conf
```

Run the following command:

```
# grep -Pl -- '\bpam_pwquality\.so\h+([\^#\n\r]+\h+)?dictcheck\b'
/usr/share/pam-configs/*
```

Edit any returned files and remove the **dictcheck** argument from the **pam_pwquality.so** line(s)

Default Value:

dictcheck = 1

References:

1. NIST SP 800-53 Rev. 5: IA-5

CIS Controls:

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1110, T1110.001, T1110.002, T1110.003, T1178.001, T1178.002, T1178.003, T1178.004	TA0006	M1027

5.3.3.2.7 Ensure password quality checking is enforced (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The **pam_pwquality** module can be configured to either reject a password if it fails the checks, or only print a warning.

This is configured by setting the **enforcing=<N>** argument. If nonzero, a password will be rejected if it fails the checks, otherwise only a warning message will be provided.

This setting applies only to the **pam_pwquality** module and possibly other applications that explicitly change their behavior based on it. It does not affect **pwmake(1)** and **pwscore(1)**.

Rationale:

Strong passwords help protect systems from password attacks. Types of password attacks include dictionary attacks, which attempt to use common words and phrases, and brute force attacks, which try every possible combination of characters. Also attackers may try to obtain the account database so they can use tools to discover the accounts and passwords.

Audit:

Run the following command to verify that **enforcing=0** has not been set in a **pwquality** configuration file:

```
# grep -PHsi -- '^h*enforcing\h*=\h*0\b' /etc/security/pwquality.conf
/etc/security/pwquality.conf.d/*.conf
```

Nothing should be returned

Run the following command to verify that the **enforcing=0** argument has not been set on the **pam_pwquality** module:

```
# grep -PHsi --
'^h*password\h+([^\n\r]+\h+pam_pwquality\.so\h+([^\n\r]+\h+)?enforcing=0\b'
/etc/pam.d/common-password
```

Nothing should be returned

Remediation:

Run the following command:

```
# grep -Pl -- '\bpam_pwquality\.so\h+([\#\n\r]+\h+)?enforcing=0\b' /usr/share/pam-configs/*
```

Edit any returned files and remove the **enforcing=0** argument from the **pam_pwquality.so** line(s)

Edit **/etc/security/pwquality.conf** and all files ending in **.conf** in the **/etc/security/pwquality.conf.d/** directory and remove or comment out any line containing the **enforcing = 0** argument:

Example:

```
# sed -ri 's/^\s*enforcing\s*=\s*0/# &/' /etc/security/pwquality.conf /etc/security/pwquality.conf.d/*.conf
```






Default Value:

enforcing=1

References:

1. pam_pwquality(8)
2. PWQUALITY.CONF(5)
3. NIST SP 800-53 Rev. 5: IA-5

CIS Controls:

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

5.3.3.2.8 Ensure password quality is enforced for the root user (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

If the `pwquality enforce_for_root` option is enabled, the module will return error on failed check even if the user changing the password is root.

This option is off by default which means that just the message about the failed check is printed but root can change the password anyway.

Note: The root is not asked for an old password so the checks that compare the old and new password are not performed.

Rationale:

Use of a complex password helps to increase the time and resources required to compromise the password. Password complexity, or strength, is a measure of the effectiveness of a password in resisting attempts at guessing and brute-force attacks.

Password complexity is one factor of several that determines how long it takes to crack a password. The more complex the password, the greater the number of possible combinations that need to be tested before the password is compromised.

Audit:

Run the following command to verify that the `enforce_for_root` option is enabled in a pwquality configuration file:

```
# grep -Psi -- '^h*enforce_for_root\b' /etc/security/pwquality.conf
/etc/security/pwquality.conf.d/*.conf
```

Example output:

```
/etc/security/pwquality.conf.d/50-pwroot.conf:enforce_for_root
```

Note:

- Settings observe an order of precedence:
 - module arguments override the settings in the `/etc/security/pwquality.conf` configuration file
 - settings in the `/etc/security/pwquality.conf` configuration file override settings in a `.conf` file in the `/etc/security/pwquality.conf.d/` directory
 - settings in a `.conf` file in the `/etc/security/pwquality.conf.d/` directory are read in canonical order, with last read file containing the setting taking precedence
- It is recommended that settings be configured in a `.conf` file in the `/etc/security/pwquality.conf.d/` directory for clarity, convenience, and durability.

Remediation:

Edit or add the following line in a `*.conf` file in `/etc/security/pwquality.conf.d` or in `/etc/security/pwquality.conf`:

Example:

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

{
    [ ! -d /etc/security/pwquality.conf.d/ ] && mkdir
    /etc/security/pwquality.conf.d/
    printf '\n%s\n' "enforce_for_root" > /etc/security/pwquality.conf.d/50-
    pwroot.conf
}
```






Default Value:

disabled

References:

1. NIST SP 800-53 Rev. 5: IA-5

CIS Controls:

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

5.3.3.3 Configure pam_pwhistory module

pam_pwhistory - PAM module to remember last passwords

pam_history.so module - This module saves the last passwords for each user in order to force password change history and keep the user from alternating between the same password too frequently.

This module does not work together with kerberos. In general, it does not make much sense to use this module in conjunction with **NIS** or **LDAP**, since the old passwords are stored on the local machine and are not available on another machine for password history checking.

Options:

- **debug** - Turns on debugging via syslog(3).
- **use_authtok** - When password changing enforce the module to use the new password provided by a previously stacked password module (this is used in the example of the stacking of the **pam_passwdqc** module documented below).
- **enforce_for_root** - If this option is set, the check is enforced for root, too.
- **remember=<N>** - The last <N> passwords for each user are saved. The default is **10**. Value of **0** makes the module to keep the existing contents of the opasswd file unchanged.
- **retry=<N>** - Prompt user at most <N> times before returning with error. The default is **1**.
- **authtok_type=<STRING>** - See pam_get_authtok(3) for more details.

Examples:

An example password section would be:

```
##PAM-1.0
password      required      pam_pwhistory.so
password      required      pam_unix.so          use_authtok
```

In combination with pam_passwdqc:

```
##PAM-1.0
password      required      pam_passwdqc.so    config=/etc/passwdqc.conf
password      required      pam_pwhistory.so    use_authtok
password      required      pam_unix.so          use_authtok
```

5.3.3.3.1 Ensure password history remember is configured (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The `/etc/security/opasswd` file stores the users' old passwords and can be checked to ensure that users are not recycling recent passwords. The number of passwords remembered is set via the `remember` argument value in `set` for the `pam_pwhistory` module.

- `remember=<N>` - `<N>` is the number of old passwords to remember

Rationale:

Requiring users not to reuse their passwords make it less likely that an attacker will be able to guess the password or use a compromised password.

Note: These change only apply to accounts configured on the local system.

Audit:

Run the following command and verify:

- The `pwhistory` line in `/etc/pam.d/common-password` includes `remember=<N>`
- The value of `<N>` is `24` or more
- The value meets local site policy

```
# grep -Psi --
'^\h*password\h+([\#\n\r]+\h+pam_pwhistory\.so\h+([\#\n\r]+\h+)?remember=\d+\b
' /etc/pam.d/common-password
```

Output should be similar to:

```
password    requisite    pam_pwhistory.so remember=24 enforce_for_root
try_first_pass use_authok
```

Remediation:

Run the following command:

```
# awk '/Password-Type:/{ f = 1;next } /-Type:/{ f = 0 } f {if (/pam_pwhistory\.so/) print FILENAME}' /usr/share/pam-configs/*
```

Edit any returned files and edit or add the **remember=** argument, with a value of **24** or more, that meets local site policy to the **pam_pwhistory** line in the **Password** section:

Example File:

```
Name: pwhistory password history checking
Default: yes
Priority: 1024
Password-Type: Primary
Password:
    requisite    pam_pwhistory.so remember=24 enforce_for_root try_first_pass
use_authok # <- **ensure line includes remember=<N>**
```

Run the following command to update the files in the **/etc/pam.d/** directory:

```
# pam-auth-update --enable <MODIFIED_PROFILE_NAME>
```






Example:

```
# pam-auth-update --enable pwhistory
```

References:

1. NIST SP 800-53 Rev. 5: IA-5(1)

CIS Controls:

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1078, T1078.001, T1078.002, T1078.003, T1078.004, T1110, T1110.004		

5.3.3.3.2 Ensure password history is enforced for the root user (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

If the **pwhistory enforce_for_root** option is enabled, the module will enforce password history for the root user as well

Rationale:

Requiring users not to reuse their passwords make it less likely that an attacker will be able to guess the password or use a compromised password

Note: These change only apply to accounts configured on the local system.

Audit:

Run the following command to verify that the **enforce_for_root** argument is exists on the **pwhistory** line in **/etc/pam.d/common-password**:

```
# grep -Psi --  
'^\h*password\h+[\^#\n\r]+\h+pam_pwhistory\.so\h+([\^#\n\r]+\h+)?enforce_for_ro  
ot\b' /etc/pam.d/common-password
```

Output should be similar to:

```
password    requisite    pam_pwhistory.so remember=24 enforce_for_root  
try_first_pass use_authok
```

Remediation:

Run the following command:

```
# awk '/Password-Type:/{ f = 1;next } /-Type:/{ f = 0 } f {if (/pam_pwhistory\.so/) print FILENAME}' /usr/share/pam-configs/*
```

Edit any returned files and add the **enforce_for_root** argument to the **pam_pwhistory** line in the **Password** section:

Example File:

```
Name: pwhistory password history checking
Default: yes
Priority: 1024
Password-Type: Primary
Password:
    requisite pam_pwhistory.so remember=24 enforce_for_root try_first_pass
use_authok # <- **ensure line includes enforce_for_root**
```

Run the following command to update the files in the **/etc/pam.d/** directory:

```
# pam-auth-update --enable <MODIFIED_PROFILE_NAME>
```

Example:

```
# pam-auth-update --enable pwhistory
```

Default Value:

disabled

References:

1. NIST SP 800-53 Rev. 5: IA-5

CIS Controls:

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1110, T1110.001, T1110.002, T1110.003, T1178.001, T1178.002, T1178.003, T1178.004	TA0006	M1027

5.3.3.3.3 *Ensure pam_pwhistory includes use_authtok (Automated)*

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

use_authtok - When password changing enforce the module to set the new password to the one provided by a previously stacked password module

Rationale:

use_authtok allows multiple pam modules to confirm a new password before it is accepted.

Audit:

Run the following command to verify that the **use_authtok** argument exists on the **pwhistory** line in **/etc/pam.d/common-password**:

```
# grep -Psi --  
'^\h*password\h+([\#\n\r]+\h+pam_pwhistory\.so\h+([\#\n\r]+\h+)?use_authtok\b'  
/etc/pam.d/common-password
```

Output should be similar to:

```
password    requisite    pam_pwhistory.so remember=24 enforce_for_root  
try_first_pass use_authtok
```

Remediation:

Run the following command:

```
# awk '/Password-Type:/{ f = 1;next } /-Type:/{ f = 0 } f {if (/pam_pwhistory\.so/) print FILENAME}' /usr/share/pam-configs/*
```

Edit any returned files and add the **use_authok** argument to the **pam_pwhistory** line in the **Password** section:

Example File:

```
Name: pwhistory password history checking
Default: yes
Priority: 1024
Password-Type: Primary
Password:
    requisite    pam_pwhistory.so remember=24 enforce_for_root try_first_pass
use_authok # <- **ensure line includes use_authok**
```

Run the following command to update the files in the **/etc/pam.d/** directory:

```
# pam-auth-update --enable <MODIFIED_PROFILE_NAME>
```

Example:

```
# pam-auth-update --enable pwhistory
```

References:

1. NIST SP 800-53 Rev. 5: IA-5

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.11 <u>Encrypt Sensitive Data at Rest</u> Encrypt sensitive data at rest on servers, applications, and databases containing sensitive data. Storage-layer encryption, also known as server-side encryption, meets the minimum requirement of this Safeguard. Additional encryption methods may include application-layer encryption, also known as client-side encryption, where access to the data storage device(s) does not permit access to the plain-text data.		●	●
v7	16.4 <u>Encrypt or Hash all Authentication Credentials</u> Encrypt or hash with a salt all authentication credentials when stored.		●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1003, T1003.008, T1110, T1110.002	TA0006	M1041

5.3.3.4 Configure pam_unix module

The `pam_unix.so` module is the standard Unix authentication module. It uses standard calls from the system's libraries to retrieve and set account information as well as authentication. Usually this is obtained from the `/etc/passwd` and the `/etc/shadow` file as well if shadow is enabled.

5.3.3.4.1 Ensure pam_unix does not include nullok (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The **nullok** argument overrides the default action of **pam_unix.so** to not permit the user access to a service if their official password is blank.

Rationale:

Using a strong password is essential to helping protect personal and sensitive information from unauthorized access

Audit:

Run the following command to verify that the **nullok** argument is not set on the **pam_unix.so** module:

```
# grep -PH -- '^\\h*^\\h*[\\^#\\n\\r]+\\h+pam_unix\\.so\\b' /etc/pam.d/common-  
{password,auth,account,session,session-noninteractive} | grep -Pv --  
'\\bnullok\\b'
```

Output should be similar to:

```
/etc/pam.d/common-password:password [success=1 default=ignore]  
pam_unix.so obscure use_authtok try_first_pass yescrypt  
/etc/pam.d/common-auth:auth [success=2 default=ignore] pam_unix.so  
try_first_pass  
/etc/pam.d/common-account:account [success=1 new_authtok_reqd=done  
default=ignore] pam_unix.so  
/etc/pam.d/common-session:session required pam_unix.so  
/etc/pam.d/common-session-noninteractive:session required pam_unix.so
```

Remediation:

Run the following command:

```
# grep -PH -- '^h*([^\n\r]+\h+)?pam_unix\.so\h+([^\n\r]+\h+)?nullok\b' /usr/share/pam-configs/*
```

Edit any files returned and remove the **nullok** argument for the **pam_unix** lines

Example File:

```
Name: Unix authentication
Default: yes
Priority: 256
Auth-Type: Primary
Auth:
    [success=end default=ignore]    pam_unix.so try_first_pass # <-
**ensure line does not include nullok nullok**
Auth-Initial:
    [success=end default=ignore]    pam_unix.so # <- **ensure line does
not include nullok nullok**
Account-Type: Primary
Account:
    [success=end new_authtok_reqd=done default=ignore]    pam_unix.so
Account-Initial:
    [success=end new_authtok_reqd=done default=ignore]    pam_unix.so
Session-Type: Additional
Session:
    required    pam_unix.so
Session-Initial:
    required    pam_unix.so
Password-Type: Primary
Password:
    [success=end default=ignore]    pam_unix.so obscure use_authtok
try_first_pass yescrypt
Password-Initial:
    [success=end default=ignore]    pam_unix.so obscure yescrypt
```

Run the following command to update the files in the **/etc/pam.d/** directory:






```
# pam-auth-update --enable <EDITED_PROFILE_NAME>
```

Example:

```
# pam-auth-update --enable unix
```

Note: If custom files are being used, the corresponding files in **/etc/pam.d/** would need to be edited directly, and the **pam-auth-update --enable <EDITED_PROFILE_NAME>** command skipped

CIS Controls:

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1003, T1003.008, T1110, T1110.002	TA0006	M1041

5.3.3.4.2 Ensure pam_unix does not include remember (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The `remember=n` argument saves the last n passwords for each user in `/etc/security/opasswd` in order to force password change history and keep the user from alternating between the same password too frequently. The MD5 password hash algorithm is used for storing the old passwords. Instead of this option the `pam_pwhistory` module should be used. The `pam_pwhistory` module saves the last n passwords for each user in `/etc/security/opasswd` using the password hash algorithm set on the `pam_unix` module. This allows for the `yescrypt` or `sha512` hash algorithm to be used.

Rationale:

The `remember=n` argument should be removed to ensure a strong password hashing algorithm is being used. A stronger hash provides additional protection to the system by increasing the level of effort needed for an attacker to successfully determine local user's old passwords stored in `/etc/security/opasswd`.

Audit:

Run the following command to verify that the `remember` argument is not set on the `pam_unix.so` module:

```
# grep -PH -- '^h*\^h*[^#\n\r]+\h+pam_unix\.so\b' /etc/pam.d/common-  
{password,auth,account,session,session-noninteractive} | grep -Pv --  
'\bremember=\d+\b'
```

Output should be similar to:

```
/etc/pam.d/common-password:password [success=1 default=ignore]  
pam_unix.so obscure yescrypt  
/etc/pam.d/common-auth:auth [success=1 default=ignore] pam_unix.so  
/etc/pam.d/common-account:account [success=1 new_authtok_reqd=done  
default=ignore] pam_unix.so  
/etc/pam.d/common-session:session required pam_unix.so  
/etc/pam.d/common-session-noninteractive:session required pam_unix.so
```

Remediation:

Run the following command:

```
# grep -PH -- '^h*([\#\n\r]+\h+)?pam_unix\.so\h+([\#\n\r]+\h+)?remember\b' /usr/share/pam-configs/*
```

Edit any files returned and remove the **remember=<N>** argument for the **pam_unix** lines

Example output:

```
[success=end default=ignore] pam_unix.so obscure use_authtok try_first_pass yescrypt remember=5 # **<- remove remember=<N>**  
[success=end default=ignore] pam_unix.so obscure yescrypt remember=5 # **<- remove remember=<N>**
```

Run the following command to update the files in the **/etc/pam.d/** directory:






```
# pam-auth-update --enable <EDITED_PROFILE_NAME>
```

Example:

```
# pam-auth-update --enable unix
```

Note: If custom files are being used, the corresponding files in **/etc/pam.d/** would need to be edited directly, and the **pam-auth-update --enable <EDITED_PROFILE_NAME>** command skipped

CIS Controls:

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1003, T1003.008, T1110, T1110.002	TA0006	M1041

5.3.3.4.3 Ensure pam_unix includes a strong password hashing algorithm (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

A cryptographic hash function converts an arbitrary-length input into a fixed length output. Password hashing performs a one-way transformation of a password, turning the password into another string, called the hashed password.

The **pam_unix** module can be configured to use one of the following hashing algorithms for user's passwords:

- **md5** - When a user changes their password next, encrypt it with the **MD5** algorithm.
- **bigcrypt** - When a user changes their password next, encrypt it with the **DEC C2** algorithm.
- **sha256** - When a user changes their password next, encrypt it with the **SHA256** algorithm. The **SHA256** algorithm must be supported by the crypt(3) function.
- **sha512** - When a user changes their password next, encrypt it with the **SHA512** algorithm. The **SHA512** algorithm must be supported by the crypt(3) function.
- **blowfish** - When a user changes their password next, encrypt it with the **blowfish** algorithm. The **blowfish** algorithm must be supported by the crypt(3) function.
- **gost_ycrypt** - When a user changes their password next, encrypt it with the **gost-ycrypt** algorithm. The **gost-ycrypt** algorithm must be supported by the crypt(3) function.
- **ycrypt** - When a user changes their password next, encrypt it with the **ycrypt** algorithm. The **ycrypt** algorithm must be supported by the crypt(3) function.

Rationale:

The **SHA-512** and **ycrypt** algorithms provide a stronger hash than other algorithms used by Linux for password hash generation. A stronger hash provides additional protection to the system by increasing the level of effort needed for an attacker to successfully determine local user passwords.

Note: These changes only apply to the local system.

Audit:

Run the following command to verify that a strong password hashing algorithm is set on the `pam_unix.so` module:

```
# grep -PH --
'^\h*password\h+([\#\n\r]+)\h+pam_unix\.so\h+([\#\n\r]+\h+)?(sha512|yescrypt)
\b' /etc/pam.d/common-password
```

Output should be similar to:

```
/etc/pam.d/common-password:password [success=1 default=ignore]
pam_unix.so obscure use_authtok try_first_pass yescrypt
```

Verify that the line(s) include either **sha512** - OR - **yescrypt**

Remediation:

Run the following command:

```
# awk '/Password-Type:/{ f = 1;next } /-Type:/{ f = 0 } f {if
(/pam_unix\.so/) print FILENAME}' /usr/share/pam-configs/*
```

Edit any returned files and edit or add a strong hashing algorithm, either `sha512` or `yescrypt`, that meets local site policy to the `pam_unix` lines in the **Password** section:

Example File:

```
Name: Unix authentication
Default: yes
Priority: 256
Auth-Type: Primary # <- Start of "Auth" section
Auth:
    [success=end default=ignore]    pam_unix.so try_first_pass
Auth-Initial:
    [success=end default=ignore]    pam_unix.so
Account-Type: Primary # <- Start of "Account" section
Account:
    [success=end new_authtok_reqd=done default=ignore]    pam_unix.so
Account-Initial:
    [success=end new_authtok_reqd=done default=ignore]    pam_unix.so
Session-Type: Additional # <- Start of "Session" section
Session:
    required    pam_unix.so
Session-Initial:
    required    pam_unix.so
Password-Type: Primary # <- Start of "Password" section
Password:
    [success=end default=ignore]    pam_unix.so obscure use_authtok
try_first_pass yescrypt # <- **ensure hashing algorithm is either sha512 or
yescrypt**
Password-Initial:
    [success=end default=ignore]    pam_unix.so obscure yescrypt # <-
**ensure hashing algorithm is either sha512 or yescrypt**
```

Run the following command to update the files in the `/etc/pam.d/` directory:

```
# pam-auth-update --enable <MODIFIED_PROFILE_NAME>
```

Example:

```
# pam-auth-update --enable unix
```

References:





1. NIST SP 800-53 Rev. 5: IA-5

Additional Information:

The following command may be used to expire all non-system user ID's immediately and force them to change their passwords on next login. Any system accounts that need to be expired should be carefully done separately by the system administrator to prevent any potential problems.

```
# awk -F: ' ( $3<"$(awk '/^\s*UID_MIN/{print $2}' /etc/login.defs)"' && $1 != "nfsnobody" ) { print $1 }' /etc/passwd | xargs -n 1 chage -d 0
```

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.11 <u>Encrypt Sensitive Data at Rest</u> Encrypt sensitive data at rest on servers, applications, and databases containing sensitive data. Storage-layer encryption, also known as server-side encryption, meets the minimum requirement of this Safeguard. Additional encryption methods may include application-layer encryption, also known as client-side encryption, where access to the data storage device(s) does not permit access to the plain-text data.			
v7	16.4 <u>Encrypt or Hash all Authentication Credentials</u> Encrypt or hash with a salt all authentication credentials when stored.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1003, T1003.008, T1110, T1110.002	TA0006	M1041

5.3.3.4.4 Ensure pam_unix includes use_authtok (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

use_authtok - When password changing enforce the module to set the new password to the one provided by a previously stacked password module

Rationale:

use_authtok allows multiple pam modules to confirm a new password before it is accepted.

Audit:

Run the following command to verify that **use_authtok** is set on the pam_unix.so module lines in the password stack:

```
# grep -PH --  
'^\h*password\h+([\^#\n\r]+)\h+pam_unix\.so\h+([\^#\n\r]+\h+)?use_authtok\b'  
/etc/pam.d/common-password
```

Output should be similar to:

```
/etc/pam.d/common-password:password    [success=1 default=ignore]  
pam_unix.so obscure use_authtok try_first_pass yescrypt
```

Verify that the line(s) include **use_authtok**

Remediation:

Run the following command:

```
# awk '/Password-Type:/{ f = 1;next } /-Type:/{ f = 0 } f {if (/pam_unix\.so/) print FILENAME}' /usr/share/pam-configs/*
```

Edit any returned files add **use_authtok** to the **pam_unix** line in the **Password** section under **Password:** subsection:

Note: The if the file's **Password** section includes a **Password-Initial:** subsection, **use_authtok** should not be added to the **pam_unix** line in the **Password-Initial:** subsection

Example File:

```
Name: Unix authentication
Default: yes
Priority: 256
Auth-Type: Primary # <- Start of "Auth" section
Auth:
    [success=end default=ignore]    pam_unix.so try_first_pass
Auth-Initial:
    [success=end default=ignore]    pam_unix.so
Account-Type: Primary # <- Start of "Account" section
Account:
    [success=end new_authtok_reqd=done default=ignore]    pam_unix.so
Account-Initial:
    [success=end new_authtok_reqd=done default=ignore]    pam_unix.so
Session-Type: Additional # <- Start of "Session" section
Session:
    required    pam_unix.so
Session-Initial:
    required    pam_unix.so
Password-Type: Primary # <- Start of "Password" section
Password:
    [success=end default=ignore]    pam_unix.so obscure use_authtok
try_first_pass yescrypt # <- **ensure line includes use_authtok**
Password-Initial:
    [success=end default=ignore]    pam_unix.so obscure yescrypt # <-
**Password-Initial: subsection does not include use_authtok
```

Run the following command to update the files in the **/etc/pam.d/** directory:

```
# pam-auth-update --enable <MODIFIED_PROFILE_NAME>
```





Example:

```
# pam-auth-update --enable unix
```

References:

1. NIST SP 800-53 Rev. 5: IA-5

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.11 <u>Encrypt Sensitive Data at Rest</u> Encrypt sensitive data at rest on servers, applications, and databases containing sensitive data. Storage-layer encryption, also known as server-side encryption, meets the minimum requirement of this Safeguard. Additional encryption methods may include application-layer encryption, also known as client-side encryption, where access to the data storage device(s) does not permit access to the plain-text data.			
v7	16.4 <u>Encrypt or Hash all Authentication Credentials</u> Encrypt or hash with a salt all authentication credentials when stored.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1003, T1003.008, T1110, T1110.002	TA0006	M1041

5.4 User Accounts and Environment

This section provides guidance on setting up secure defaults for system and user accounts and their environment.

5.4.1 Configure shadow password suite parameters

While a majority of the password control parameters have been moved to PAM, some parameters are still available through the shadow password suite. Any changes made to `/etc/login.defs` will only be applied if the `usermod` command is used. If user IDs are added a different way, use the `chage` command to effect changes to individual user IDs.

5.4.1.1 Ensure password expiration is configured (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The `PASS_MAX_DAYS` parameter in `/etc/login.defs` allows an administrator to force passwords to expire once they reach a defined age.

`PASS_MAX_DAYS <N>` - The maximum number of days a password may be used. If the password is older than this, a password change will be forced. If not specified, -1 will be assumed (which disables the restriction).

Rationale:

The window of opportunity for an attacker to leverage compromised credentials or successfully compromise credentials via an online brute force attack is limited by the age of the password. Therefore, reducing the maximum age of a password also reduces an attacker's window of opportunity.

We recommend a yearly password change. This is primarily because for all their good intentions users will share credentials across accounts. Therefore, even if a breach is publicly identified, the user may not see this notification, or forget they have an account on that site. This could leave a shared credential vulnerable indefinitely. Having an organizational policy of a 1-year (annual) password expiration is a reasonable compromise to mitigate this with minimal user burden.

Impact:

The password expiration must be greater than the minimum days between password changes or users will be unable to change their password.

Excessive password expiration requirements do more harm than good, because these requirements make users select predictable passwords, composed of sequential words and numbers that are closely related to each other. In these cases, the next password can be predicted based on the previous one (incrementing a number used in the password for example). Also, password expiration requirements offer no containment benefits because attackers will often use credentials as soon as they compromise them. Instead, immediate password changes should be based on key events including, but not limited to:

- Indication of compromise
- Change of user roles
- When a user leaves the organization.

Not only does changing passwords every few weeks or months frustrate the user, but it's also been suggested that it does more harm than good, because it could lead to bad practices by the user such as adding a character to the end of their existing password.

Audit:

Run the following command and verify **PASS_MAX_DAYS** is set to 365 days or less and conforms to local site policy:

```
# grep -Pi -- '^h*PASS_MAX_DAYS\h+\d+\b' /etc/login.defs
```

Example output:

```
PASS_MAX_DAYS 365
```

Run the following command to verify all **/etc/shadow** passwords **PASS_MAX_DAYS**:

- is greater than **0** days
- is less than or equal to **365** days
- conforms to local site policy

```
# awk -F: '($2~/^\$.+\$/ ) {if($5 > 365 || $5 < 1)print "User: " $1 " " "PASS_MAX_DAYS: " $5}' /etc/shadow
```

Nothing should be returned

Remediation:

Set the **PASS_MAX_DAYS** parameter to conform to site policy in **/etc/login.defs** :

```
PASS_MAX_DAYS 365
```

Modify user parameters for all users with a password set to match:

```
# chage --maxdays 365 <user>
```

Edit **/etc/login.defs** and set **PASS_MAX_DAYS** to a value greater than **0** that follows local site policy:

Example:

```
PASS_MAX_DAYS 365
```

Run the following command to modify user parameters for all users with a password set to a maximum age no greater than **365** or less than **1** that follows local site policy:

```
# chage --maxdays <N> <user>
```

Example:

```
# awk -F: '($2~/^\$.+\/) {if($5 > 365 || $5 < 1)system ("chage --maxdays 365 " $1)}' /etc/shadow
```

Warning: If a password has been set at system install or kickstart, the **last change date** field is not set, In this case, setting **PASS_MAX_DAYS** will immediately expire the password. One possible solution is to populate the **last change date** field through a command like: **chage -d "\$(date +%Y-%m-%d)" root**

Default Value:

PASS_MAX_DAYS 99999






References:

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

Additional Information:

A value of -1 will disable password expiration.

CIS Controls:

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1078, T1078.001, T1078.002, T1078.003, T1078.004, T1110, T1110.001, T1110.002, T1110.003, T1110.004		

5.4.1.2 Ensure minimum password days is configured (Manual)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

PASS_MIN_DAYS <N> - The minimum number of days allowed between password changes. Any password changes attempted sooner than this will be rejected. If not specified, 0 will be assumed (which disables the restriction).

Rationale:

Users may have favorite passwords that they like to use because they are easy to remember, and they believe that their password choice is secure from compromise. Unfortunately, passwords are compromised and if an attacker is targeting a specific individual user account, with foreknowledge of data about that user, reuse of old, potentially compromised passwords, may cause a security breach.

By restricting the frequency of password changes, an administrator can prevent users from repeatedly changing their password in an attempt to circumvent password reuse controls

Impact:

If a user's password is set by other personnel as a procedure in dealing with a lost or expired password, the user should be forced to update this "set" password with their own password. e.g. force "change at next logon".

If it is not possible to have a user set their own password immediately, and this recommendation or local site procedure may cause a user to continue using a third party generated password, **PASS_MIN_DAYS** for the effected user should be temporally changed to 0, to allow a user to change their password immediately.

For applications where the user is not using the password at console, the ability to "change at next logon" may be limited. This may cause a user to continue to use a password created by other personnel.

Audit:

Run the following command to verify that **PASS_MIN_DAYS** is set to a value greater than 0 and follows local site policy:

```
# grep -Pi -- '^\\h*PASS_MIN_DAYS\\h+\\d+\\b' /etc/login.defs
```

Example output:

```
PASS_MIN_DAYS 1
```

Run the following command to verify all passwords have a **PASS_MIN_DAYS** greater than 0:

```
# awk -F: '($2~/^\\$.+\\$/) {if($4 < 1)print "User: " $1 " PASS_MIN_DAYS: " $4}' /etc/shadow
```

Nothing should be returned

Remediation:

Edit **/etc/login.defs** and set **PASS_MIN_DAYS** to a value greater than 0 that follows local site policy:

Example:

```
PASS_MIN_DAYS 1
```

Run the following command to modify user parameters for all users with a password set to a minimum days greater than zero that follows local site policy:

```
# chage --mindays <N> <user>
```

Example:

```
# awk -F: '($2~/^\\$.+\\$/) {if($4 < 1)system ("chage --mindays 1 " $1)}' /etc/shadow
```






Default Value:

PASS_MIN_DAYS 0

References:

1. CIS Password Policy Guide

CIS Controls:

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1078, T1078.001, T1078.002, T1078.003, T1078.004, T1110, T1110.004	TA0006	M1027

5.4.1.3 Ensure password expiration warning days is configured (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The **PASS_WARN_AGE** parameter in **/etc/login.defs** allows an administrator to notify users that their password will expire in a defined number of days.

PASS_WARN_AGE <N> - The number of days warning given before a password expires. A zero means warning is given only upon the day of expiration, a negative value means no warning is given. If not specified, no warning will be provided.

Rationale:

Providing an advance warning that a password will be expiring gives users time to think of a secure password. Users caught unaware may choose a simple password or write it down where it may be discovered.

Audit:

Run the following command and verify **PASS_WARN_AGE** is 7 or more and follows local site policy:

```
# grep -Pi -- '^h*PASS_WARN_AGE\h+\d+\b' /etc/login.defs
```

Example output:

```
PASS_WARN_AGE 7
```

Run the following command to verify all passwords have a **PASS_WARN_AGE** of 7 or more:

```
# awk -F: '($2~/^\$.+\$/ ) {if($6 < 7)print "User: " $1 " PASS_WARN_AGE: " $6}' /etc/shadow
```

Nothing should be returned

Remediation:

Edit `/etc/login.defs` and set `PASS_WARN_AGE` to a value of **7** or more that follows local site policy:

Example:

```
PASS_WARN_AGE 7
```

Run the following command to modify user parameters for all users with a password set to a minimum warning to **7** or more days that follows local site policy:

```
# chage --warndays <N> <user>
```






Example:

```
# awk -F: '($2~/^\$.+\/) {if($6 < 7)system ("chage --warndays 7 " $1)}' /etc/shadow
```

Default Value:

`PASS_WARN_AGE 7`

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	4.1 Establish and Maintain a Secure Configuration Process Establish and maintain a secure configuration process for enterprise assets (end-user devices, including portable and mobile, non-computing/IoT devices, and servers) and software (operating systems and applications). Review and update documentation annually, or when significant enterprise changes occur that could impact this Safeguard.			
v7	4.4 Use Unique Passwords Where multi-factor authentication is not supported (such as local administrator, root, or service accounts), accounts will use passwords that are unique to that system.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1078	TA0006	M1027

5.4.1.4 Ensure strong password hashing algorithm is configured (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

A cryptographic hash function converts an arbitrary-length input into a fixed length output. Password hashing performs a one-way transformation of a password, turning the password into another string, called the hashed password.

ENCRYPT_METHOD (string) - This defines the system default encryption algorithm for encrypting passwords (if no algorithm are specified on the command line). It can take one of these values:

- **MD5** - MD5-based algorithm will be used for encrypting password
- **SHA256** - SHA256-based algorithm will be used for encrypting password
- **SHA512** - SHA512-based algorithm will be used for encrypting password
- **BCRYPT** - BCRYPT-based algorithm will be used for encrypting password
- **YESCRYPT** - YESCRYPT-based algorithm will be used for encrypting password
- **DES** - DES-based algorithm will be used for encrypting password (default)

Note:

- This parameter overrides the deprecated **MD5_CRYPT_ENAB** variable.
- This parameter will only affect the generation of group passwords.
- The generation of user passwords is done by PAM and subject to the PAM configuration.
- It is recommended to set this variable consistently with the PAM configuration.

Rationale:

The **SHA-512** and **yescrypt** algorithms provide a stronger hash than other algorithms used by Linux for password hash generation. A stronger hash provides additional protection to the system by increasing the level of effort needed for an attacker to successfully determine local group passwords.

Audit:

Run the following command to verify the hashing algorithm is **sha512** or **yescrypt** in **/etc/login.defs**:

```
# grep -Pi -- '^\\h*ENCRYPT_METHOD\\h+(SHA512|yescrypt)\\b' /etc/login.defs
```

Example output:

```
ENCRYPT_METHOD SHA512
- OR -
ENCRYPT_METHOD YESCRYPT
```

Remediation:

Edit **/etc/login.defs** and set the **ENCRYPT_METHOD** to **SHA512** or **YESCRYPT**:

```
ENCRYPT_METHOD <HASHING_ALGORITHM>
```

Example:

```
ENCRYPT_METHOD YESCRYPT
```

Note:

- This only effects local groups' passwords created after updating the file to use **sha512** or **yescrypt**.
- If it is determined that the password algorithm being used is not **sha512** or **yescrypt**, once it is changed, it is recommended that all group passwords be updated to use the stronger hashing algorithm.
- It is recommended that the chosen hashing algorithm is consistent across **/etc/login.defs** and the PAM configuration





Default Value:

```
ENCRYPT_METHOD SHA512
```

References:

1. NIST SP 800-53 Rev. 5: IA-5

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.11 <u>Encrypt Sensitive Data at Rest</u> Encrypt sensitive data at rest on servers, applications, and databases containing sensitive data. Storage-layer encryption, also known as server-side encryption, meets the minimum requirement of this Safeguard. Additional encryption methods may include application-layer encryption, also known as client-side encryption, where access to the data storage device(s) does not permit access to the plain-text data.			
v7	16.4 <u>Encrypt or Hash all Authentication Credentials</u> Encrypt or hash with a salt all authentication credentials when stored.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1003, T1003.008, T1110, T1110.002	TA0006	M1041

5.4.1.5 Ensure inactive password lock is configured (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

User accounts that have been inactive for over a given period of time can be automatically disabled.

INACTIVE - Defines the number of days after the password exceeded its maximum age where the user is expected to replace this password.

The value is stored in the shadow password file. An input of **0** will disable an expired password with no delay. An input of **-1** will blank the respective field in the shadow password file.

Rationale:

Inactive accounts pose a threat to system security since the users are not logging in to notice failed login attempts or other anomalies.

Audit:

Run the following command and verify **INACTIVE** conforms to site policy (no more than 45 days):

```
# useradd -D | grep INACTIVE  
  
INACTIVE=45
```

Verify all users with a password have Password inactive no more than 45 days after password expires

Verify all users with a password have Password inactive no more than 45 days after password expires: Run the following command and Review list of users and **INACTIVE** to verify that all users **INACTIVE** conforms to site policy (no more than 45 days):

```
# awk -F: '($2~/^\$.+\$/ ) {if($7 > 45 || $7 < 0)print "User: " $1 " INACTIVE:  
" $7}' /etc/shadow
```

Nothing should be returned

Remediation:

Run the following command to set the default password inactivity period to 45 days or less that meets local site policy:

```
# useradd -D -f <N>
```

Example:

```
# useradd -D -f 45
```

Run the following command to modify user parameters for all users with a password set to a inactive age of 45 days or less that follows local site policy:

```
# chage --inactive <N> <user>
```

Example:

```
# awk -F: '($2~/^\$.+\/) {if($7 > 45 || $7 < 0)system ("chage --inactive 45 " $1)}' /etc/shadow
```

Default Value:

INACTIVE=-1






References:

1. CIS Password Policy Guide

Additional Information:

A value of -1 would disable this setting.

CIS Controls:

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1078, T1078.002, T1078.003	TA0001	M1027

5.4.1.6 Ensure all users last password change date is in the past (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

All users should have a password change date in the past.

Rationale:

If a user's recorded password change date is in the future, then they could bypass any set password expiration.

Audit:






Run the following command and verify nothing is returned

```
{
  while IFS= read -r l_user; do
    l_change=$(date -d "$(chage --list $l_user | grep '^Last password
change' | cut -d: -f2 | grep -v 'never$')" +%s)
    if [[ "$l_change" -gt "$(date +%s)" ]]; then
      echo "User: \"$l_user\" last password change was \"$(chage --list
$l_user | grep '^Last password change' | cut -d: -f2)\""
    fi
  done <<(awk -F: '$2~/^\$.+\/{{print $1}}' /etc/shadow)
}
```

Remediation:

Investigate any users with a password change date in the future and correct them. Locking the account, expiring the password, or resetting the password manually may be appropriate.

CIS Controls:

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1078, T1078.001, T1078.002, T1078.003, T1078.004, T1110, T1110.001, T1110.002, T1110.003, T1110.004		

5.4.2 Configure root and system accounts and environment

5.4.2.1 Ensure root is the only UID 0 account (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

Any account with UID 0 has superuser privileges on the system.

Rationale:

This access must be limited to only the default **root** account and only from the system console. Administrative access must be through an unprivileged account using an approved mechanism as noted in Item 5.6 Ensure access to the su command is restricted.

Audit:

Run the following command and verify that only "root" is returned:

```
# awk -F: '($3 == 0) { print $1 }' /etc/passwd  
root
```

Remediation:

Run the following command to change the **root** account UID to **0**:

```
# usermod -u 0 root
```

Modify any users other than **root** with UID **0** and assign them a new UID.

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
T1548, T1548.000	TA0001	M1026

5.4.2.2 Ensure root is the only GID 0 account (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The **usermod** command can be used to specify which group the **root** account belongs to. This affects permissions of files that are created by the **root** account.

Rationale:

Using GID 0 for the **root** account helps prevent **root** -owned files from accidentally becoming accessible to non-privileged users.

Audit:

Run the following command to verify the **root** user's primary GID is 0, and no other user's have GID 0 as their primary GID:

```
# awk -F: '($1 !~ /^(sync|shutdown|halt|operator)/ && $4=="0") {print $1":"$4}' /etc/passwd

root:0
```

Note: User's: sync, shutdown, halt, and operator are excluded from the check for other user's with GID 0

Remediation:

Run the following command to set the **root** user's GID to 0:

```
# usermod -g 0 root
```

Run the following command to set the **root** group's GID to 0:







```
# groupmod -g 0 root
```

Remove any users other than the **root** user with GID 0 or assign them a new GID if appropriate.

References:

1. 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	3.3 <u>Configure Data Access Control Lists</u> Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	14.6 <u>Protect Information through Access Control Lists</u> 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, T1548.000	TA0005	M1026

5.4.2.3 Ensure group root is the only GID 0 group (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The **groupmod** command can be used to specify which group the **root** group belongs to. This affects permissions of files that are group owned by the **root** group.

Rationale:

Using GID 0 for the **root** group helps prevent **root** group owned files from accidentally becoming accessible to non-privileged users.

Audit:

Run the following command to verify no group other than **root** is assigned GID 0:

```
# awk -F: '$3=="0"{print $1":"$3}' /etc/group  
root:0
```

Remediation:

Run the following command to set the **root** group's GID to 0:







```
# groupmod -g 0 root
```

Remove any groups other than the **root** group with GID 0 or assign them a new GID if appropriate.

References:

1. 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	3.3 <u>Configure Data Access Control Lists</u> Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	14.6 <u>Protect Information through Access Control Lists</u> 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, T1548.000	TA0005	M1026

5.4.2.4 Ensure root account access is controlled (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

There are a number of methods to access the root account directly. Without a password set any user would be able to gain access and thus control over the entire system.

Rationale:

Access to **root** should be secured at all times.

Impact:

If there are any automated processes that relies on access to the root account without authentication, they will fail after remediation.

Audit:

Run the following command to verify that either the root user's password is set or the root user's account is locked:

```
# passwd -S root | awk '$2 ~ /^(P|L)/ {print "User: \"" $1 "\"" Password is status: " $2}'
```

Verify the output is either:

```
User: "root" Password is status: P
- OR -
User: "root" Password is status: L
```

Note:

- **P** - Password is set
- **L** - Password is locked

Remediation:

Run the following command to set a password for the **root** user:







```
# passwd root
```

- OR -

Run the following command to lock the **root** user account:

```
# usermod -L root
```

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 <u>Configure Data Access Control Lists</u> Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	14.6 <u>Protect Information through Access Control Lists</u> 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
T1078	TA0005	M1026

5.4.2.5 Ensure root path integrity (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The **root** user can execute any command on the system and could be fooled into executing programs unintentionally if the **PATH** is not set correctly.

Rationale:

Including the current working directory (.) or other writable directory in **root**'s executable path makes it likely that an attacker can gain superuser access by forcing an administrator operating as **root** to execute a Trojan horse program.

Audit:

Run the following script to verify root's path does not include:

- Locations that are not directories
- An empty directory (:::)
- A trailing (:)
- Current working directory (.)
- Non **root** owned directories
- Directories that less restrictive than mode **0755**

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

{
    l_output2=""
    l_pmask="0022"
    l_maxperm="$( printf '%o' $(( 0777 & ~$l_pmask )) )"
    l_root_path="$(sudo -Hiu root env | grep '^PATH' | cut -d= -f2)"
    unset a_path_loc && IFS=":" read -ra a_path_loc <<< "$l_root_path"
    grep -q "::-" <<< "$l_root_path" && l_output2="$l_output2\n - root's path
contains a empty directory (:::)"
    grep -Pq ":\h*$" <<< "$l_root_path" && l_output2="$l_output2\n - root's
path contains a trailing (:)"
    grep -Pq '(\h+|:)\.(:|\h*$)' <<< "$l_root_path" && l_output2="$l_output2\n
- root's path contains current working directory (.)"
    while read -r l_path; do
        if [ -d "$l_path" ]; then
            while read -r l_fmode l_fown; do
                [ "$l_fown" != "root" ] && l_output2="$l_output2\n - Directory:
\"$l_path\" is owned by: \"$l_fown\" should be owned by \"root\""
                [ $(( $l_fmode & $l_pmask )) -gt 0 ] && l_output2="$l_output2\n -
Directory: \"$l_path\" is mode: \"$l_fmode\" and should be mode:
\"$l_maxperm\" or more restrictive"
                done <<< "$(stat -Lc '%#a %U' "$l_path")"
            else
                l_output2="$l_output2\n - \"$l_path\" is not a directory"
            fi
        done <<< "$(printf "%s\n" "${a_path_loc[@]}")"
        if [ -z "$l_output2" ]; then
            echo -e "\n- Audit Result:\n  *** PASS ***\n - Root's path is correctly
configured\n"
        else
            echo -e "\n- Audit Result:\n  ** FAIL **\n - * Reasons for audit
failure * :\n$l_output2\n"
        fi
    }
}
```


Remediation:

Correct or justify any:

- Locations that are not directories
- Empty directories (::)
- Trailing (:)
- Current working directory (.)
- Non root owned directories
- Directories that less restrictive than mode 0755

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
T1204, T1204.002	TA0006	M1022

5.4.2.6 Ensure root user umask is configured (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The user file-creation mode mask (**umask**) is used to determine the file permission for newly created directories and files. In Linux, the default permissions for any newly created directory is 0777 (**rxwxrwxrwx**), and for any newly created file it is 0666 (**rw-rw-rw-**). The **umask** modifies the default Linux permissions by restricting (masking) these permissions. The **umask** is not simply subtracted, but is processed bitwise. Bits set in the **umask** are cleared in the resulting file mode.

umask can be set with either **Octal** or **Symbolic** values:

- **Octal** (Numeric) Value - Represented by either three or four digits. ie **umask 0027** or **umask 027**. If a four digit umask is used, the first digit is ignored. The remaining three digits effect the resulting permissions for user, group, and world/other respectively.
- **Symbolic** Value - Represented by a comma separated list for User **u**, group **g**, and world/other **o**. The permissions listed are not masked by **umask**. ie a **umask** set by **umask u=rwx,g=rx,o=** is the **Symbolic** equivalent of the **Octal umask 027**. This **umask** would set a newly created directory with file mode **drwxr-x---** and a newly created file with file mode **rw-r-----**.

root user Shell Configuration Files:

- **/root/.bash_profile** - Is executed to configure the root users' shell before the initial command prompt. **Is only read by login shells.**
- **/root/.bashrc** - Is executed for interactive shells. **only read by a shell that's both interactive and non-login**

umask is set by order of precedence. If **umask** is set in multiple locations, this order of precedence will determine the system's default **umask**.

Order of precedence:

1. **/root/.bash_profile**
2. **/root/.bashrc**
3. The system default umask

Rationale:

Setting a secure value for **umask** ensures that users make a conscious choice about their file permissions. A permissive **umask** value could result in directories or files with excessive permissions that can be read and/or written to by unauthorized users.

Audit:

Run the following to verify the root user **umask** is set to enforce a newly created directories' permissions to be **750 (drwxr-x---)**, and a newly created file's permissions be **640 (rw-r-----)**, or more restrictive:

```
# grep -Psi -- '^h*umask\h+([0-7][0-7][01][0-7]\b|[0-7][0-7][0-7][0-6]\b)|([0-7][01][0-7]\b|[0-7][0-7][0-6]\b)|(u=[rx]{1,3},)?((g=[rx]?[rx]?w[rx]?[rx]?b)(,o=[rx]{1,3})?)|((g=[rx]{1,3},)?o=[rx]{1,3}\b)))' /root/.bash_profile /root/.bashrc
```

Nothing should be returned.

Remediation:

Edit **/root/.bash_profile** and **/root/.bashrc** and remove, comment out, or update any line with **umask** to be **0027** or more restrictive.







Default Value:

System default **umask**

References:

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

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1083	TA0007	

5.4.2.7 Ensure system accounts do not have a valid login shell (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

There are a number of accounts provided with most distributions that are used to manage applications and are not intended to provide an interactive shell. Furthermore, a user may add special accounts that are not intended to provide an interactive shell.

Rationale:

It is important to make sure that accounts that are not being used by regular users are prevented from being used to provide an interactive shell. By default, most distributions set the password field for these accounts to an invalid string, but it is also recommended that the shell field in the password file be set to the **nologin** shell. This prevents the account from potentially being used to run any commands.

Audit:

Run the following command to verify system accounts, except for **root**, **halt**, **sync**, **shutdown** or **nfsnobody**, do not have a valid login shell:

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

{
  l_valid_shells="^($(awk -F\| '$NF != "nologin" {print}' /etc/shells | sed
-rn '/^\|/{s/,/,\\|/,g;p}' | paste -s -d '|' - ))$"
  awk -v pat="$l_valid_shells" -F:
'($1!~/^(root|halt|sync|shutdown|nfsnobody)$/ && ($3<'$(awk
'/^\s*UID_MIN/{print $2}' /etc/login.defs)'" || $3 == 65534) && $(NF) ~ pat)
{print "Service account: \" " $1 "\" has a valid shell: \" $7}' /etc/passwd
}
```

Nothing should be returned

Remediation:

Run the following command to set the shell for any service accounts returned by the audit to **nologin**:

```
# usermod -s $(command -v nologin) <user>
```

Example script:

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

{
  l_valid_shells="^($( awk -F\| ' $NF != "nologin" {print}' /etc/shells | sed
-rn '/^\|/{s/,/,\\|/,g;p}' | paste -s -d '|' - ))$"
  awk -v pat="$l_valid_shells" -F:
' ($1!~/^(root|halt|sync|shutdown|nfsnobody)$/ && ($3<'$(awk
'/^s*UID_MIN/{print $2}' /etc/login.defs)' || $3 == 65534) && $(NF) ~ pat)
{system ("usermod -s '$(command -v nologin)' " $1)}' /etc/passwd
}
```







References:

1. NIST SP 800-53 Rev. 5: AC-2(5), AC-3, AC-11, MP-2

Additional Information:

The **root**, **sync**, **shutdown**, and **halt** users are exempted from requiring a non-login shell.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1078, T1078.001, T1078.003	TA0005	M1026

5.4.2.8 Ensure accounts without a valid login shell are locked (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

There are a number of accounts provided with most distributions that are used to manage applications and are not intended to provide an interactive shell. Furthermore, a user may add special accounts that are not intended to provide an interactive shell.

Rationale:

It is important to make sure that accounts that are not being used by regular users are prevented from being used to provide an interactive shell. By default, most distributions set the password field for these accounts to an invalid string, but it is also recommended that the shell field in the password file be set to the **nologin** shell. This prevents the account from potentially being used to run any commands.

Audit:

Run the following script to verify all non-root accounts without a valid login shell are locked.

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

{
  l_valid_shells="^($(awk -F\ / '$NF != "nologin" {print}' /etc/shells | sed
-rn '/^\/\{s,/,,\|\/,g;p}' | paste -s -d '|' - ))$"
  while IFS= read -r l_user; do
    passwd -S "$l_user" | awk '$2 !~ /^L/ {print "Account: \"" $1 "\" does
not have a valid login shell and is not locked"}'
    done <<(awk -v pat="$l_valid_shells" -F: '($1 != "root" && $(NF) !~ pat)
{print $1}' /etc/passwd)
  }
}
```

Nothing should be returned

Remediation:

Run the following command to lock any non-root accounts without a valid login shell returned by the audit:

```
# usermod -L <user>
```

Example script::







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

{
  l_valid_shells="^($(awk -F\| ' $NF != "nologin" {print}' /etc/shells | sed
-rn '/^\|/{s/,/,\\|/,g;p}' | paste -s -d '|' - ))$"
  while IFS= read -r l_user; do
    passwd -S "$l_user" | awk '$2 !~ /^L/ {system ("usermod -L " $1)}'
  done < <$(awk -v pat="$l_valid_shells" -F: '($1 != "root" && $(NF) !~ pat)
{print $1}' /etc/passwd)
}
```

References:

1. NIST SP 800-53 Rev. 5: AC-2(5), AC-3, AC-11, MP-2

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1078, T1078.001, T1078.003	TA0005	M1026

5.4.3 Configure user default environment

5.4.3.1 Ensure *nologin* is not listed in */etc/shells* (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

/etc/shells is a text file which contains the full pathnames of valid login shells. This file is consulted by *chsh* and available to be queried by other programs.

Be aware that there are programs which consult this file to find out if a user is a normal user; for example, FTP daemons traditionally disallow access to users with shells not included in this file.

Rationale:

A user can use *chsh* to change their configured shell.

If a user has a shell configured that isn't in */etc/shells*, then the system assumes that they're somehow restricted. In the case of *chsh* it means that the user cannot change that value.

Other programs might query that list and apply similar restrictions.

By putting *nologin* in */etc/shells*, any user that has *nologin* as its shell is considered a full, unrestricted user. This is not the expected behavior for *nologin*.

Audit:

Run the following command to verify that *nologin* is not listed in the */etc/shells* file:

```
# grep -Ps '^h*([^\n\r]+)?\/nologin\b' /etc/shells
```

Nothing should be returned

Remediation:

Edit */etc/shells* and remove any lines that include *nologin*

References:

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

5.4.3.2 Ensure default user shell timeout is configured (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

TMOUT is an environmental setting that determines the timeout of a shell in seconds.

- **TMOUT=*n*** - Sets the shell timeout to *n* seconds. A setting of **TMOUT=0** disables timeout.
- **readonly TMOUT** - Sets the TMOUT environmental variable as readonly, preventing unwanted modification during run-time.
- **export TMOUT** - exports the TMOUT variable

System Wide Shell Configuration Files:

- **/etc/profile** - used to set system wide environmental variables on users shells. The variables are sometimes the same ones that are in the **.bash_profile**, however this file is used to set an initial PATH or PS1 for all shell users of the system. **is only executed for interactive login shells, or shells executed with the --login parameter.**
- **/etc/profile.d** - **/etc/profile** will execute the scripts within **/etc/profile.d/*.sh**. It is recommended to place your configuration in a shell script within **/etc/profile.d** to set your own system wide environmental variables.
- **/etc/bashrc** - System wide version of **.bashrc**. In Fedora derived distributions, **/etc/bashrc** also invokes **/etc/profile.d/*.sh** if *non-login* shell, but redirects output to **/dev/null** if *non-interactive*. **Is only executed for interactive shells or if BASH_ENV is set to /etc/bashrc.**

Rationale:

Setting a timeout value reduces the window of opportunity for unauthorized user access to another user's shell session that has been left unattended. It also ends the inactive session and releases the resources associated with that session.

Audit:

Run the following script to verify that **TMOUT** is configured to: include a timeout of no more than **900** seconds, to be **readonly**, to be **exported**, and is not being changed to a longer timeout.

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

{
    output1="" output2=""
    [ -f /etc/bashrc ] && BRC="/etc/bashrc"
    for f in "$BRC" /etc/profile /etc/profile.d/*.sh ; do
        grep -Pq '^s*([^\#]+\s+)?TMOUT=(900|[1-8][0-9][0-9]|[1-9][0-9]|[1-9])\b' "$f" && grep -Pq '^s*([^\#]+\s+)?readonly\s+TMOUT(\s+|\s*;\s*$|=(900|[1-8][0-9][0-9]|[1-9][0-9]|[1-9]))\b' "$f" && grep -Pq '^s*([^\#]+\s+)?export\s+TMOUT(\s+|\s*;\s*$|=(900|[1-8][0-9][0-9]|[1-9][0-9]|[1-9]))\b' "$f" &&
        output1="$f"
    done
    grep -Pq '^s*([^\#]+\s+)?TMOUT=(9[0-9][1-9]|9[1-9][0-9]|0+|[1-9]\d{3,})\b' /etc/profile /etc/profile.d/*.sh "$BRC" && output2=$(grep -Ps '^s*([^\#]+\s+)?TMOUT=(9[0-9][1-9]|9[1-9][0-9]|0+|[1-9]\d{3,})\b' /etc/profile /etc/profile.d/*.sh $BRC)
    if [ -n "$output1" ] && [ -z "$output2" ]; then
        echo -e "\nPASSED\n\nTMOUT is configured in: \"$output1\"\n"
    else
        [ -z "$output1" ] && echo -e "\nFAILED\n\nTMOUT is not configured\n"
        [ -n "$output2" ] && echo -e "\nFAILED\n\nTMOUT is incorrectly configured in: \"$output2\"\n"
    fi
}
```

Remediation:

Review `/etc/bashrc`, `/etc/profile`, and all files ending in `*.sh` in the `/etc/profile.d/` directory and remove or edit all `TMOUT=_n_` entries to follow local site policy. `TMOUT` should not exceed 900 or be equal to 0.

Configure `TMOUT` in **one** of the following files:

- A file in the `/etc/profile.d/` directory ending in `.sh`
- `/etc/profile`
- `/etc/bashrc`

TMOUT configuration examples:

- As multiple lines:

```
TMOUT=900
readonly TMOUT
export TMOUT
```

- As a single line:







```
readonly TMOUT=900 ; export TMOUT
```

Additional Information:

The audit and remediation in this recommendation apply to bash and shell. If other shells are supported on the system, it is recommended that their configuration files also are checked. Other methods of setting a timeout exist for other shells not covered here.

Ensure that the timeout conforms to your local policy.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	4.3 <u>Configure Automatic Session Locking on Enterprise Assets</u> Configure automatic session locking on enterprise assets after a defined period of inactivity. For general purpose operating systems, the period must not exceed 15 minutes. For mobile end-user devices, the period must not exceed 2 minutes.			
v7	16.11 <u>Lock Workstation Sessions After Inactivity</u> Automatically lock workstation sessions after a standard period of inactivity.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1078	TA0005	M1026

5.4.3.3 Ensure default user umask is configured (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The user file-creation mode mask (**umask**) is used to determine the file permission for newly created directories and files. In Linux, the default permissions for any newly created directory is 0777 (**rxwxrwxrwx**), and for any newly created file it is 0666 (**rw-rw-rw-**). The **umask** modifies the default Linux permissions by restricting (masking) these permissions. The **umask** is not simply subtracted, but is processed bitwise. Bits set in the **umask** are cleared in the resulting file mode.

umask can be set with either **Octal** or **Symbolic** values:

- **Octal** (Numeric) Value - Represented by either three or four digits. ie **umask 0027** or **umask 027**. If a four digit umask is used, the first digit is ignored. The remaining three digits effect the resulting permissions for user, group, and world/other respectively.
- **Symbolic** Value - Represented by a comma separated list for User **u**, group **g**, and world/other **o**. The permissions listed are not masked by **umask**. ie a **umask** set by **umask u=rwx,g=rx,o=** is the **Symbolic** equivalent of the **Octal umask 027**. This **umask** would set a newly created directory with file mode **drwxr-x---** and a newly created file with file mode **rw-r-----**.

The default **umask** can be set to use the **pam_umask** module or in a **System Wide Shell Configuration File**. The user creating the directories or files has the discretion of changing the permissions via the **chmod** command, or choosing a different default **umask** by adding the **umask** command into a **User Shell Configuration File**, (**.bash_profile** or **.bashrc**), in their home directory.

Setting the default umask:

- pam_umask module:
 - will set the umask according to the system default in `/etc/login.defs` and user settings, solving the problem of different `umask` settings with different shells, display managers, remote sessions etc.
 - `umask=<mask>` value in the `/etc/login.defs` file is interpreted as Octal
 - Setting `USERGROUPS_ENAB` to yes in `/etc/login.defs` (default):
 - will enable setting of the `umask` group bits to be the same as owner bits. (examples: 022 -> 002, 077 -> 007) for non-root users, if the `uid` is the same as `gid`, and `username` is the same as the `<primary group name>`
 - `userdel` will remove the user's group if it contains no more members, and `useradd` will create by default a group with the name of the user
- System Wide Shell Configuration File:
 - `/etc/profile` - used to set system wide environmental variables on users shells. The variables are sometimes the same ones that are in the `.bash_profile`, however this file is used to set an initial PATH or PS1 for all shell users of the system. **is only executed for interactive login shells, or shells executed with the `--login` parameter.**
 - `/etc/profile.d` - `/etc/profile` will execute the scripts within `/etc/profile.d/*.sh`. It is recommended to place your configuration in a shell script within `/etc/profile.d` to set your own system wide environmental variables.
 - `/etc/bashrc` - System wide version of `.bashrc`. In Fedora derived distributions, `etc/bashrc` also invokes `/etc/profile.d/*.sh` if *non-login* shell, but redirects output to `/dev/null` if *non-interactive*. **Is only executed for interactive shells or if `BASH_ENV` is set to `/etc/bashrc`.**

User Shell Configuration Files:

- `~/.bash_profile` - Is executed to configure your shell before the initial command prompt. **Is only read by login shells.**
- `~/.bashrc` - Is executed for interactive shells. **only read by a shell that's both interactive and non-login**

`umask` is set by order of precedence. If `umask` is set in multiple locations, this order of precedence will determine the system's default `umask`.

Order of precedence:

1. A file in `/etc/profile.d/` ending in `.sh` - This will override any other system-wide `umask` setting
2. In the file `/etc/profile`
3. On the `pam_umask.so` module in `/etc/pam.d/postlogin`
4. In the file `/etc/login.defs`
5. In the file `/etc/default/login`

Rationale:

Setting a secure default value for `umask` ensures that users make a conscious choice about their file permissions. A permissive `umask` value could result in directories or files with excessive permissions that can be read and/or written to by unauthorized users.

Audit:

Run the following to verify the default user **umask** is set to **027**(octal) or **u=rwx,g=rx,o=** (Symbolic) to enforce newly created directories' permissions to be **750** (**drwxr-x---**), and newly created file's permissions be **640** (**rw-r-----**), or more restrictive:

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

{
    l_output="" l_output2=""
    file_umask_chk()
    {
        if grep -Psiq -- '^\\h*umask\\h+(0?[0-7][2-7]7|u(=[rwx]{0,3}),g(=[rx]{0,2}),o(=)(\\h*#.*)?\\$' "$l_file"; then
            l_output="$l_output\\n - umask is set correctly in \"$l_file\\n\"
            elif grep -Psiq -- '^\\h*umask\\h+(((0-7)[0-7][01][0-7]\\b|[0-7][0-7][0-7][0-6]\\b)|((0-7)[01][0-7]\\b|[0-7][0-7][0-6]\\b)|(u(=[rwx]{1,3}),)?((g(=[rx]?[rx]?w[rx]?[rx]?\\b)(,o(=[rwx]{1,3}))?)|((g(=[wrx]{1,3}),)?o(=[wrx]{1,3}\\b))))' "$l_file"; then
                l_output2="$l_output2\\n - umask is incorrectly set in \"$l_file\\n\"
            fi
        }
        while IFS= read -r -d $'\\0' l_file; do
            file_umask_chk
            done <<(find /etc/profile.d/ -type f -name '*.sh' -print0)
            [ -z "$l_output" ] && l_file="/etc/profile" && file_umask_chk
            [ -z "$l_output" ] && l_file="/etc/bashrc" && file_umask_chk
            [ -z "$l_output" ] && l_file="/etc/bash.bashrc" && file_umask_chk
            [ -z "$l_output" ] && l_file="/etc/pam.d/postlogin"
            if [ -z "$l_output" ]; then
                if grep -Psiq -- '^\\h*session\\h+[^#\\n\\r]+\\h+pam_umask\\.so\\h+([^#\\n\\r]+\\h+)?umask=(0?[0-7][2-7]7)\\b' "$l_file"; then
                    l_output1="$l_output1\\n - umask is set correctly in \"$l_file\\n\"
                    elif grep -Psiq '^\\h*session\\h+[^#\\n\\r]+\\h+pam_umask\\.so\\h+([^#\\n\\r]+\\h+)?umask=((0-7)[0-7][01][0-7]\\b|[0-7][0-7][0-7][0-6]\\b)|((0-7)[01][0-7]\\b))' "$l_file"; then
                        l_output2="$l_output2\\n - umask is incorrectly set in \"$l_file\\n\"
                    fi
                fi
                [ -z "$l_output" ] && l_file="/etc/login.defs" && file_umask_chk
                [ -z "$l_output" ] && l_file="/etc/default/login" && file_umask_chk
                [[ -z "$l_output" && -z "$l_output2" ]] && l_output2="$l_output2\\n - umask is not set"
                if [ -z "$l_output2" ]; then
                    echo -e "\\n- Audit Result:\\n  ** PASS **\\n - * Correctly configured *
                    :\\n$l_output\\n"
                else
                    echo -e "\\n- Audit Result:\\n  ** FAIL **\\n - * Reasons for audit failure *
                    :\\n$l_output2"
                    [ -n "$l_output" ] && echo -e "\\n- * Correctly configured *
                    :\\n$l_output\\n"
                fi
            }
}
```

Remediation:

Run the following script and perform the instructions in the output to set the default umask to **027** or more restrictive:

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

{
    l_output="" l_output2="" l_out=""
    file_umask_chk()
    {
        if grep -Psiq -- '^\\h*umask\\h+(0?[0-7][2-7]7|u(=[rx]{0,3}),g(=[rx]{0,2}),o=)(\\h*#\\.*)?$' "$l_file"; then
            l_out="$l_out\\n - umask is set correctly in \"$l_file\\n\"
            elif grep -Psiq -- '^\\h*umask\\h+((([0-7][0-7][01][0-7]\\b|[0-7][0-7][0-7][0-6]\\b)|([0-7][01][0-7]\\b|[0-7][0-7][0-6]\\b)|(u(=[rx]{1,3}),)?((g(=[rx]?[rx]?w[rx]?[rx]?\\b)(,o(=[rx]{1,3})?)|((g(=[rx]?[rx]?o(=[rx]{1,3})\\b)))' "$l_file"; then
                l_output2="$l_output2\\n    - \"$l_file\\n\"
            fi
        }
        while IFS= read -r -d $'\\0' l_file; do
            file_umask_chk
            done < <(find /etc/profile.d/ -type f -name '*.sh' -print0)
            [ -n "$l_out" ] && l_output="$l_out"
            l_file="/etc/profile" && file_umask_chk
            l_file="/etc/bashrc" && file_umask_chk
            l_file="/etc/bash.bashrc" && file_umask_chk
            l_file="/etc/pam.d/postlogin"
            if grep -Psiq '^\\h*session\\h+([\\#\\n\\r]+\\h+pam_umask\\.so\\h+([\\#\\n\\r]+\\h+)?umask=((([0-7][0-7][01][0-7]\\b|[0-7][0-7][0-7][0-6]\\b)|([0-7][01][0-7]\\b))' "$l_file"; then
                l_output2="$l_output2\\n    - \"$l_file\\n\"
            fi
            l_file="/etc/login.defs" && file_umask_chk
            l_file="/etc/default/login" && file_umask_chk
            if [ -z "$l_output2" ]; then
                echo -e " - No files contain a UMASK that is not restrictive enough\\n No UMASK updates required to existing files"
            else
                echo -e "\\n - UMASK is not restrictive enough in the following file(s):$l_output2\\n\\n- Remediation Procedure:\\n - Update these files and comment out the UMASK line\\n    or update umask to be \\\"0027\\\" or more restrictive"
            fi
            if [ -n "$l_output" ]; then
                echo -e "$l_output"
            else
                echo -e " - Configure UMASK in a file in the \\\"/etc/profile.d/\\\" directory ending in \\\".sh\\\"\\n\\n    Example Command (Hash to represent being run at a root prompt):\\n\\n# printf '%s\\\\\\n' \\\"umask 027\\\" > /etc/profile.d/50-systemwide_umask.sh\\n"
            fi
        }
    }
}
```

Notes:

- This method only applies to bash and shell. If other shells are supported on the system, it is recommended that their configuration files also are checked
- If the `pam_umask.so` module is going to be used to set `umask`, ensure that it's not being overridden by another setting. Refer to the `PAM_UMASK(8)` man page for more information

Default Value:

UMASK 022







References:

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

Additional Information:

- Other methods of setting a default user umask exist
- If other methods are in use in your environment they should be audited
- The default user umask can be overridden with a user specific umask
- The user creating the directories or files has the discretion of changing the permissions:
 - Using the `chmod` command
 - Setting a different default umask by adding the `umask` command into a User Shell Configuration File, (`.bashrc`), in their home directory
 - Manually changing the umask for the duration of a login session by running the `umask` command

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 <u>Configure Data Access Control Lists</u> Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	14.6 <u>Protect Information through Access Control Lists</u> 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
T1083	TA0007	