

## 7.1 System File Permissions

This section provides guidance on securing aspects of system files and directories.

### 7.1.1 Ensure permissions on /etc/passwd are configured (Automated)

#### Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

#### Description:

The `/etc/passwd` file contains user account information that is used by many system utilities and therefore must be readable for these utilities to operate.

#### Rationale:

It is critical to ensure that the `/etc/passwd` file is protected from unauthorized write access. Although it is protected by default, the file permissions could be changed either inadvertently or through malicious actions.

#### Audit:

Run the following command to verify `/etc/passwd` is mode 644 or more restrictive, `Uid` is `0/root` and `Gid` is `0/root`:

```
# stat -Lc 'Access: (%#a/%A)  Uid: ( %u/ %U) Gid: ( %g/ %G)' /etc/passwd
Access: (0644/-rw-r--r--)  Uid: ( 0/ root) Gid: ( 0/ root)
```

#### Remediation:

Run the following commands to remove excess permissions, set owner, and set group on `/etc/passwd`:

```
# chmod u-x,go-wx /etc/passwd
# chown root:root /etc/passwd
```

#### Default Value:

Access: (0644/-rw-r--r--) Uid: ( 0/ root) Gid: ( 0/ root)

#### References:

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

## CIS Controls:

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

## MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1003, T1003.008, T1222, T1222.002	TA0005	M1022

## 7.1.2 Ensure permissions on /etc/passwd- are configured (Automated)

### Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

### Description:

The `/etc/passwd-` file contains backup user account information.

### Rationale:

It is critical to ensure that the `/etc/passwd-` file is protected from unauthorized access. Although it is protected by default, the file permissions could be changed either inadvertently or through malicious actions.

### Audit:

Run the following command to verify `/etc/passwd-` is mode 644 or more restrictive, `Uid` is `0/root` and `Gid` is `0/root`:

```
# stat -Lc 'Access: (%#a/%A)  Uid: ( %u/ %U) Gid: { %g/ %G)' /etc/passwd-  
Access: (0644/-rw-r--r--)  Uid: ( 0/ root) Gid: { 0/ root)
```

### Remediation:

Run the following commands to remove excess permissions, set owner, and set group on `/etc/passwd-`:

```
# chmod u-x,go-wx /etc/passwd-  
# chown root:root /etc/passwd-
```







### Default Value:

Access: (0644/-rw-r--r--) Uid: ( 0/ root) Gid: { 0/ root)

### References:

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

## CIS Controls:

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

## MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1003, T1003.008, T1222, T1222.002	TA0005	M1022

### 7.1.3 Ensure permissions on /etc/group are configured (Automated)

#### Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

#### Description:

The `/etc/group` file contains a list of all the valid groups defined in the system. The command below allows read/write access for root and read access for everyone else.

#### Rationale:

The `/etc/group` file needs to be protected from unauthorized changes by non-privileged users, but needs to be readable as this information is used with many non-privileged programs.

#### Audit:

Run the following command to verify `/etc/group` is mode 644 or more restrictive, `Uid` is `0/root` and `Gid` is `0/root`:

```
# stat -Lc 'Access: (%a/%A)  Uid: ( %u/ %U) Gid: ( %g/ %G)' /etc/group
Access: (0644/-rw-r--r--)  Uid: ( 0/ root) Gid: ( 0/ root)
```

#### Remediation:

Run the following commands to remove excess permissions, set owner, and set group on `/etc/group`:

```
# chmod u-x,go-wx /etc/group
# chown root:root /etc/group
```







#### Default Value:

Access: (0644/-rw-r--r--) Uid: ( 0/ root) Gid: ( 0/ root)

#### References:

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

## CIS Controls:

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

## MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1003, T1003.008, T1222, T1222.002	TA0005	M1022

## 7.1.4 Ensure permissions on /etc/group- are configured (Automated)

### Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

### Description:

The `/etc/group-` file contains a backup list of all the valid groups defined in the system.

### Rationale:

It is critical to ensure that the `/etc/group-` file is protected from unauthorized access. Although it is protected by default, the file permissions could be changed either inadvertently or through malicious actions.

### Audit:

Run the following command to verify `/etc/group-` is mode 644 or more restrictive, `Uid` is `0/root` and `Gid` is `0/root`:

```
# stat -Lc 'Access: (%#a/%A)  Uid: ( %u/ %U) Gid: ( %g/ %G)' /etc/group-  
Access: (0644/-rw-r--r--)  Uid: ( 0/ root) Gid: ( 0/ root)
```

### Remediation:

Run the following commands to remove excess permissions, set owner, and set group on `/etc/group-`:

```
# chmod u-x,go-wx /etc/group-  
# chown root:root /etc/group-
```

### Default Value:







Access: (0644/-rw-r--r--) Uid: ( 0/ root) Gid: ( 0/ root)

### References:

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



## CIS Controls:

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

## MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1003, T1003.008, T1222, T1222.002	TA0005	M1022

## 7.1.5 Ensure permissions on /etc/shadow are configured (Automated)

### Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

### Description:

The `/etc/shadow` file is used to store the information about user accounts that is critical to the security of those accounts, such as the hashed password and other security information.

### Rationale:

If attackers can gain read access to the `/etc/shadow` file, they can easily run a password cracking program against the hashed password to break it. Other security information that is stored in the `/etc/shadow` file (such as expiration) could also be useful to subvert the user accounts.

### Audit:

Run the following command to verify `/etc/shadow` is mode 640 or more restrictive, `Uid` is `0/root` and `Gid` is `0/root` or `{GID}/shadow`:

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

#### Example:

```
Access: (0640/-rw-r-----)  Uid: ( 0/ root) Gid: ( 42/ shadow)
```

### Remediation:

Run **one** of the following commands to set ownership of `/etc/shadow` to `root` and group to either `root` or `shadow`:

```
# chown root:shadow /etc/shadow
-OR-
# chown root:root /etc/shadow
```

Run the following command to remove excess permissions from `/etc/shadow`:

```
# chmod u-x,g-wx,o-rwx /etc/shadow
```







### Default Value:

Access: (0640/-rw-r-----) Uid: ( 0/ root) Gid: ( 42/ shadow)

## References:

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

## CIS Controls:

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

## MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1003, T1003.008, T1222, T1222.002	TA0005	M1022

## 7.1.6 Ensure permissions on /etc/shadow- are configured (Automated)

### Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

### Description:

The `/etc/shadow-` file is used to store backup information about user accounts that is critical to the security of those accounts, such as the hashed password and other security information.

### Rationale:

It is critical to ensure that the `/etc/shadow-` file is protected from unauthorized access. Although it is protected by default, the file permissions could be changed either inadvertently or through malicious actions.

### Audit:

Run the following command to verify `/etc/shadow-` is mode 640 or more restrictive, `Uid` is `0/root` and `Gid` is `0/root` or `{GID}/shadow`:

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

### Example:

```
Access: (0640/-rw-r-----)  Uid: ( 0/ root) Gid: ( 42/ shadow)
```

### Remediation:

Run **one** of the following commands to set ownership of `/etc/shadow-` to `root` and group to either `root` or `shadow`:

```
# chown root:shadow /etc/shadow-  
-OR-  
# chown root:root /etc/shadow-
```

Run the following command to remove excess permissions from `/etc/shadow-`:

```
# chmod u-x,g-wx,o-rwx /etc/shadow-
```







### Default Value:

```
Access: (0640/-rw-r-----)  Uid: ( 0/ root) Gid: ( 42/ shadow)
```

### References:

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

## CIS Controls:

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

## MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1003, T1003.008, T1222, T1222.002	TA0005	M1022

## 7.1.7 Ensure permissions on /etc/gshadow are configured (Automated)

### Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

### Description:

The `/etc/gshadow` file is used to store the information about groups that is critical to the security of those accounts, such as the hashed password and other security information.

### Rationale:

If attackers can gain read access to the `/etc/gshadow` file, they can easily run a password cracking program against the hashed password to break it. Other security information that is stored in the `/etc/gshadow` file (such as group administrators) could also be useful to subvert the group.

### Audit:

Run the following command to verify `/etc/gshadow` is mode 640 or more restrictive, `Uid` is `0/root` and `Gid` is `0/root` or `{GID}shadow`:

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

#### Example:

```
Access: (0640/-rw-r-----)  Uid: ( 0/ root) Gid: ( 42/ shadow)
```

### Remediation:

Run **one** of the following commands to set ownership of `/etc/gshadow` to `root` and group to either `root` or `shadow`:

```
# chown root:shadow /etc/gshadow
-OR-
# chown root:root /etc/gshadow
```

Run the following command to remove excess permissions from `/etc/gshadow`:

```
# chmod u-x,g-wx,o-rwx /etc/gshadow
```







### Default Value:

Access: (0640/-rw-r-----) Uid: ( 0/ root) Gid: ( 42/ shadow)

## References:

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

## CIS Controls:

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

## MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1003, T1003.008, T1222, T1222.002	TA0005	M1022

## 7.1.8 Ensure permissions on /etc/gshadow- are configured (Automated)

### Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

### Description:

The `/etc/gshadow-` file is used to store backup information about groups that is critical to the security of those accounts, such as the hashed password and other security information.

### Rationale:

It is critical to ensure that the `/etc/gshadow-` file is protected from unauthorized access. Although it is protected by default, the file permissions could be changed either inadvertently or through malicious actions.

### Audit:

Run the following command to verify `/etc/gshadow-` is mode 640 or more restrictive, `Uid` is `0/root` and `Gid` is `0/root` or `{GID}/shadow`:

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

### Example:

```
Access: (0640/-rw-r-----)  Uid: ( 0/ root) Gid: ( 42/ shadow)
```

### Remediation:

Run **one** of the following commands to set ownership of `/etc/gshadow-` to `root` and group to either `root` or `shadow`:

```
# chown root:shadow /etc/gshadow-  
-OR-  
# chown root:root /etc/gshadow-
```

Run the following command to remove excess permissions from `/etc/gshadow-`:

```
# chmod u-x,g-wx,o-rwx /etc/gshadow-
```

### Default Value:







```
Access: (0640/-rw-r-----)  Uid: ( 0/ root) Gid: ( 42/ shadow)
```

### References:

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



## CIS Controls:

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

## MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1003, T1003.008, T1222, T1222.002	TA0005	M1022

## 7.1.9 Ensure permissions on /etc/shells are configured (Automated)

### Profile Applicability:

- Level 1 - Server
- Level 1 - 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.

### Rationale:

It is critical to ensure that the `/etc/shells` file is protected from unauthorized access. Although it is protected by default, the file permissions could be changed either inadvertently or through malicious actions.

### Audit:

Run the following command to verify `/etc/shells` is mode 644 or more restrictive, `Uid` is `0/root` and `Gid` is `0/root`:

```
# stat -Lc 'Access: (%#a/%A)  Uid: ( %u/ %U) Gid: ( %g/ %G)' /etc/shells
Access: (0644/-rw-r--r--)  Uid: ( 0/ root) Gid: ( 0/ root)
```

### Remediation:

Run the following commands to remove excess permissions, set owner, and set group on `/etc/shells`:

```
# chmod u-x,go-wx /etc/shells
# chown root:root /etc/shells
```

### Default Value:

Access: (0644/-rw-r--r--) Uid: ( 0/ root) Gid: ( 0/ root)

### References:

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

## CIS Controls:

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

## MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1003, T1003.008, T1222, T1222.002	TA0005	M1022

## 7.1.10 Ensure permissions on /etc/security/opasswd are configured (Automated)

### Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

### Description:

`/etc/security/opasswd` and its backup `/etc/security/opasswd.old` hold user's previous passwords if `pam_unix` or `pam_pwhistory` is in use on the system

### Rationale:

It is critical to ensure that `/etc/security/opasswd` is protected from unauthorized access. Although it is protected by default, the file permissions could be changed either inadvertently or through malicious actions.

### Audit:

Run the following commands to verify `/etc/security/opasswd` and `/etc/security/opasswd.old` are mode 600 or more restrictive, `Uid` is `0/root` and `Gid` is `0/root` if they exist:

```
# [ -e "/etc/security/opasswd" ] && stat -Lc '%n Access: (%#a/%A)  Uid: (
%u/ %U)  Gid: ( %g/ %G)' /etc/security/opasswd

/etc/security/opasswd Access: (0600/-rw-----)  Uid: ( 0/ root) Gid: ( 0/
root)
-OR-
Nothing is returned
# [ -e "/etc/security/opasswd.old" ] && stat -Lc '%n Access: (%#a/%A)  Uid:
( %u/ %U)  Gid: ( %g/ %G)' /etc/security/opasswd.old

/etc/security/opasswd.old Access: (0600/-rw-----)  Uid: ( 0/ root) Gid: (
0/ root)
-OR-
Nothing is returned
```

## Remediation:

Run the following commands to remove excess permissions, set owner, and set group on `/etc/security/opasswd` and `/etc/security/opasswd.old` if they exist:

```
# [ -e "/etc/security/opasswd" ] && chmod u-x,go-rwx /etc/security/opasswd
# [ -e "/etc/security/opasswd" ] && chown root:root /etc/security/opasswd
# [ -e "/etc/security/opasswd.old" ] && chmod u-x,go-rwx /etc/security/opasswd.old
# [ -e "/etc/security/opasswd.old" ] && chown root:root /etc/security/opasswd.old
```







## Default Value:

`/etc/security/opasswd` Access: (0600/-rw-----) Uid: ( 0/ root) Gid: ( 0/ root)

## References:

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

## CIS Controls:

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

## MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1003, T1003.008, T1222, T1222.002	TA0005	M1022

### *7.1.11 Ensure world writable files and directories are secured (Automated)*

#### **Profile Applicability:**

- Level 1 - Server
- Level 1 - Workstation

#### **Description:**

World writable files are the least secure. Data in world-writable files can be modified and compromised by any user on the system. World writable files may also indicate an incorrectly written script or program that could potentially be the cause of a larger compromise to the system's integrity. See the `chmod(2)` man page for more information.

Setting the sticky bit on world writable directories prevents users from deleting or renaming files in that directory that are not owned by them.

#### **Rationale:**

Data in world-writable files can be modified and compromised by any user on the system. World writable files may also indicate an incorrectly written script or program that could potentially be the cause of a larger compromise to the system's integrity.

This feature prevents the ability to delete or rename files in world writable directories (such as `/tmp`) that are owned by another user.

#### **Audit:**

Run the following script to verify:

- No world writable files exist
- No world writable directories without the sticky bit exist

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

{
    l_output="" l_output2=""
    l_smask='01000'
    a_file=(); a_dir=() # Initialize arrays
    a_path=(! -path "/run/user/*" -a ! -path "/proc/*" -a ! -path
    "*/containerd/*" -a ! -path "*/kubelet/pods/*" -a ! -path
    "*/kubelet/plugins/*" -a ! -path "/sys/*" -a ! -path "/snap/*")
    while IFS= read -r l_mount; do
        while IFS= read -r -d $'\0' l_file; do
            if [ -e "$l_file" ]; then
                [ -f "$l_file" ] && a_file+=("$l_file") # Add WR files
                if [ -d "$l_file" ]; then # Add directories w/o sticky bit
                    l_mode="$(stat -Lc '%#a' "$l_file")"
                    [ ! $(( $l_mode & $l_smask )) -gt 0 ] && a_dir+=("$l_file")
                fi
            fi
        done <<(find "$l_mount" -xdev \( "${a_path[@]}" \) \( -type f -o -type
d \) -perm -0002 -print0 2> /dev/null)
        done <<(findmnt -Dkerno fstype,target | awk '($1 !~
/^\\s*(nfs|proc|smb|vfat|iso9660|efivarfs|selinuxfs)/ && $2 !~
/^(\\/run\\/user\\/|\\/tmp\\/|\\/var\\/tmp\\/){print $2}')
        if ! (( ${#a_file[@]} > 0 )); then
            l_output="$l_output\n - No world writable files exist on the local
filesystem."
        else
            l_output2="$l_output2\n - There are \"$(printf '%s' "${#a_file[@]}")\"
World writable files on the system.\n - The following is a list of World
writable files:\n$(printf '%s\n' "${a_file[@]}")\n - end of list\n"
            fi
            if ! (( ${#a_dir[@]} > 0 )); then
                l_output="$l_output\n - Sticky bit is set on world writable
directories on the local filesystem."
            else
                l_output2="$l_output2\n - There are \"$(printf '%s' "${#a_dir[@]}")\"
World writable directories without the sticky bit on the system.\n - The
following is a list of World writable directories without the sticky
bit:\n$(printf '%s\n' "${a_dir[@]}")\n - end of list\n"
                fi
            unset a_path; unset a_arr; unset a_file; unset a_dir # Remove arrays
            # If l_output2 is empty, we pass
            if [ -z "$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 "$l_output" ] && echo -e "- * Correctly configured *
:\n$l_output\n"
            fi
        }
}
```

**Note:** On systems with a large number of files and/or directories, this audit may be a long running process

## Remediation:

- World Writable Files:
  - It is recommended that write access is removed from **other** with the command ( **chmod o-w <filename>** ), but always consult relevant vendor documentation to avoid breaking any application dependencies on a given file.
- World Writable Directories:
  - Set the sticky bit on all world writable directories with the command ( **chmod a+t <directory\_name>** )

Run the following script to:

- Remove other write permission from any world writable files
- Add the sticky bit to all world writable directories

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







{
  l_smask='01000'
  a_file=(); a_dir=() # Initialize arrays
  a_path=(! -path "/run/user/*" -a ! -path "/proc/*" -a ! -path
"/containerd/*" -a ! -path "*/kubelet/pods/*" -a ! -path
"/kubelet/plugins/*" -a ! -path "/sys/*" -a ! -path "/snap/*")
  while IFS= read -r l_mount; do
    while IFS= read -r -d $'\0' l_file; do
      if [ -e "$l_file" ]; then
        l_mode="$(stat -Lc '%#a' "$l_file")"
        if [ -f "$l_file" ]; then # Remove excess permissions from WW
files
          echo -e " - File: \"$l_file\" is mode: \"$l_mode\" \n -
removing write permission on \"$l_file\" from \"other\""
          chmod o-w "$l_file"
        fi
        if [ -d "$l_file" ]; then # Add sticky bit
          if [ ! $(( $l_mode & $l_smask )) -gt 0 ]; then
            echo -e " - Directory: \"$l_file\" is mode: \"$l_mode\" and
doesn't have the sticky bit set \n - Adding the sticky bit"
            chmod a+t "$l_file"
          fi
        fi
      fi
    done <<(find "$l_mount" -xdev \( "${a_path[@]}" \) \( -type f -o -type
d \) -perm -0002 -print0 2> /dev/null)
    done <<(findmnt -Dkerno fstype,target | awk '($1 !~
/^\s*(nfs|proc|smb|vfat|iso9660|efivarfs|selinuxfs)/ && $2 !~
/^(\/run\/user\/|\/tmp\/|\/var\/tmp\/){print $2}')
```

## References:

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



## CIS Controls:

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

## MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1222, T1222.002, T1548	TA0004, TA0005	M1022, M1028

### *7.1.12 Ensure no files or directories without an owner and a group exist (Automated)*

**Profile Applicability:**

- Level 1 - Server
- Level 1 - Workstation

**Description:**

Administrators may delete users or groups from the system and neglect to remove all files and/or directories owned by those users or groups.

**Rationale:**

A new user or group who is assigned a deleted user's user ID or group ID may then end up "owning" a deleted user or group's files, and thus have more access on the system than was intended.

**Audit:**

Run the following script to verify no unowned or ungrouped files or directories exist:

```

#!/usr/bin/env bash

{
    l_output="" l_output2=""
    a_nouser=(); a_nogroup=() # Initialize arrays
    a_path=(! -path "/run/user/*" -a ! -path "/proc/*" -a ! -path
"/containerd/*" -a ! -path "*/kubelet/pods/*" -a ! -path
"/kubelet/plugins/*" -a ! -path "/sys/fs/cgroup/memory/*" -a ! -path
"/var/*/private/*")
    while IFS= read -r l_mount; do
        while IFS= read -r -d $'\0' l_file; do
            if [ -e "$l_file" ]; then
                while IFS= read -r l_user l_group; do
                    [ "$l_user" = "UNKNOWN" ] && a_nouser+=("$l_file")
                    [ "$l_group" = "UNKNOWN" ] && a_nogroup+=("$l_file")
                    done <<(stat -Lc '%U:%G' "$l_file")
                fi
                done <<(find "$l_mount" -xdev \( "${a_path[@]}" \) \( -type f -o -type
d \) \( -nouser -o -nogroup \) -print0 2> /dev/null)
                done <<(findmnt -Dkerno fstype,target | awk '($1 !~
/^\/s*(nfs|proc|smb|vfat|iso9660|efivarfs|selinuxfs)/ && $2 !~
/^\/run\/user\/){print $2}')
                if ! (( ${#a_nouser[@]} > 0 )); then
                    l_output="$l_output\n - No files or directories without a owner exist
on the local filesystem."
                else
                    l_output2="$l_output2\n - There are \"$(printf '%s'
"${#a_nouser[@]}")\" unowned files or directories on the system.\n - The
following is a list of unowned files and/or directories:\n$(printf '%s\n'
"${a_nouser[@]}")\n - end of list"
                fi
                if ! (( ${#a_nogroup[@]} > 0 )); then
                    l_output="$l_output\n - No files or directories without a group exist
on the local filesystem."
                else
                    l_output2="$l_output2\n - There are \"$(printf '%s'
"${#a_nogroup[@]}")\" ungrouped files or directories on the system.\n - The
following is a list of ungrouped files and/or directories:\n$(printf '%s\n'
"${a_nogroup[@]}")\n - end of list"
                fi
                unset a_path; unset a_arr ; unset a_nouser; unset a_nogroup # Remove
arrays
                if [ -z "$l_output2" ]; then # If l_output2 is empty, we pass
                    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
            fi
        }
}

```

**Note:** On systems with a large number of files and/or directories, this audit may be a long running process







**Remediation:**

Remove or set ownership and group ownership of these files and/or directories to an active user on the system as appropriate.

**References:**

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

**CIS Controls:**

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

**MITRE ATT&CK Mappings:**

Techniques / Sub-techniques	Tactics	Mitigations
T1222, T1222.002	TA0007	M1022

### *7.1.13 Ensure SUID and SGID files are reviewed (Manual)*

**Profile Applicability:**

- Level 1 - Server
- Level 1 - Workstation

**Description:**

The owner of a file can set the file's permissions to run with the owner's or group's permissions, even if the user running the program is not the owner or a member of the group. The most common reason for a SUID or SGID program is to enable users to perform functions (such as changing their password) that require root privileges.

**Rationale:**

There are valid reasons for SUID and SGID programs, but it is important to identify and review such programs to ensure they are legitimate. Review the files returned by the action in the audit section and check to see if system binaries have a different checksum than what from the package. This is an indication that the binary may have been replaced.

## Audit:

Run the following script to generate a list of SUID and SGID files:

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

{
    l_output="" l_output2=""
    a_suid=(); a_sgid=() # initialize arrays
    while IFS= read -r l_mount; do
        while IFS= read -r -d $'\0' l_file; do
            if [ -e "$l_file" ]; then
                l_mode="$(stat -Lc '%#a' "$l_file")"
                [ $(( $l_mode & 04000 )) -gt 0 ] && a_suid+=("$l_file")
                [ $(( $l_mode & 02000 )) -gt 0 ] && a_sgid+=("$l_file")
            fi
        done << (find "$l_mount" -xdev -type f \( -perm -2000 -o -perm -4000 \)
        -print0 2>/dev/null)
        done << (findmnt -Dkerno fstype,target,options | awk '($1 !~
        /\^s*(nfs|proc|smb|vfat|iso9660|efivarfs|selinuxfs)/ && $2 !~
        /\^\/run\/user\/\// && $3 !~/noexec/ && $3 !~/nosuid/) {print $2}')
        if ! (( ${#a_suid[@]} > 0 )); then
            l_output="$l_output\n - No executable SUID files exist on the system"
        else
            l_output2="$l_output2\n - List of \"$(printf '%s' "${#a_suid[@]}")\"
            SUID executable files:\n$(printf '%s\n' "${a_suid[@]}")\n - end of list -\n"
        fi
        if ! (( ${#a_sgid[@]} > 0 )); then
            l_output="$l_output\n - No SGID files exist on the system"
        else
            l_output2="$l_output2\n - List of \"$(printf '%s' "${#a_sgid[@]}")\"
            SGID executable files:\n$(printf '%s\n' "${a_sgid[@]}")\n - end of list -\n"
        fi
        [ -n "$l_output2" ] && l_output2="$l_output2\n- Review the preceding
        list(s) of SUID and/or SGID files to\n- ensure that no rogue programs have
        been introduced onto the system.\n"
        unset a_arr; unset a_suid; unset a_sgid # Remove arrays
        # If l_output2 is empty, Nothing to report
        if [ -z "$l_output2" ]; then
            echo -e "\n- Audit Result:\n$l_output\n"
        else
            echo -e "\n- Audit Result:\n$l_output2\n"
            [ -n "$l_output" ] && echo -e "$l_output\n"
        fi
    fi
}
```

**Note:** on systems with a large number of files, this may be a long running process







## Remediation:

Ensure that no rogue SUID or SGID programs have been introduced into the system. Review the files returned by the action in the Audit section and confirm the integrity of these binaries.

## References:

1. NIST SP 800-53 Rev. 5: CM-1, CM-2, CM-6, CM-7, IA-5, AC-3, MP-2

## CIS Controls:

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

## MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1548, T1548.001	TA0004	M1028