

## 7 System Maintenance

Recommendations in this section are intended as maintenance and are intended to be checked on a frequent basis to ensure system stability. Many recommendations do not have quick remediations and require investigation into the cause and best fix available and may indicate an attempted breach of system security.

## 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><br>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><br>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><br>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><br>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><br>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><br>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><br>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><br>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><br>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><br>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><br>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><br>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><br>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><br>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><br>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><br>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><br>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><br>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><br>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><br>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><br>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><br>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><br>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><br>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><br>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><br>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       |



## 7.2 Local User and Group Settings

This section provides guidance on securing aspects of the local users and groups.

**Note:** The recommendations in this section check local users and groups. Any users or groups from other sources such as LDAP will not be audited. In a domain environment similar checks should be performed against domain users and groups.

## 7.2.1 Ensure accounts in /etc/passwd use shadowed passwords (Automated)

### Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

### Description:

Local accounts can use shadowed passwords. With shadowed passwords, the passwords are saved in the shadow password file, `/etc/shadow`, encrypted by a salted one-way hash. Accounts with a shadowed password have an `x` in the second field in `/etc/passwd`.

### Rationale:

The `/etc/passwd` file also contains information like user ID's and group ID's that are used by many system programs. Therefore, the `/etc/passwd` file must remain world readable. In spite of encoding the password with a randomly-generated one-way hash function, an attacker could still break the system if they got access to the `/etc/passwd` file. This can be mitigated by using shadowed passwords, thus moving the passwords in the `/etc/passwd` file to `/etc/shadow`. The `/etc/shadow` file is set so only root will be able to read and write. This helps mitigate the risk of an attacker gaining access to the encoded passwords with which to perform a dictionary attack.

### Note:

- All accounts must have passwords or be locked to prevent the account from being used by an unauthorized user.
- A user account with an empty second field in `/etc/passwd` allows the account to be logged into by providing only the username.

### Audit:

Run the following command and verify that no output is returned:

```
# awk -F: '($2 != "x" ) { print "User: \"" $1 "\" is not set to shadowed passwords "}' /etc/passwd
```

### Remediation:

Run the following command to set accounts to use shadowed passwords and migrate passwords in `/etc/passwd` to `/etc/shadow`:

```
# pwconv
```

Investigate to determine if the account is logged in and what it is being used for, to determine if it needs to be forced off.

## References:

1. NIST SP 800-53 Rev. 5: IA-5
2. PWCONV(8)

## Additional Information:

The `pwconv` command creates shadow from `passwd` and an optionally existing `shadow`.

- The `pwunconv` command creates `passwd` from `passwd` and `shadow` and then removes `shadow`.
- The `grpconv` command creates `gshadow` from `group` and an optionally existing `gshadow`.
- The `grpunconv` command creates `group` from `group` and `gshadow` and then removes `gshadow`.





These four programs all operate on the normal and shadow password and group files: `/etc/passwd`, `/etc/group`, `/etc/shadow`, and `/etc/gshadow`.

Each program acquires the necessary locks before conversion. `pwconv` and `grpconv` are similar. First, entries in the shadowed file which don't exist in the main file are removed. Then, shadowed entries which don't have `x'` as the password in the main file are updated. Any missing shadowed entries are added. Finally, passwords in the main file are replaced with `x'`. These programs can be used for initial conversion as well to update the shadowed file if the main file is edited by hand.

`pwconv` will use the values of `PASS_MIN_DAYS`, `PASS_MAX_DAYS`, and `PASS_WARN_AGE` from `/etc/login.defs` when adding new entries to `/etc/shadow`.

`pwunconv` and `grpunconv` are similar. Passwords in the main file are updated from the shadowed file. Entries which exist in the main file but not in the shadowed file are left alone. Finally, the shadowed file is removed. Some password aging information is lost by `pwunconv`. It will convert what it can.

## CIS Controls:

| Controls Version | Control                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | IG 1 | IG 2                                                                                | IG 3                                                                                |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
| v8               | <b>3.11 <u>Encrypt Sensitive Data at Rest</u></b><br>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               | <b>16.4 <u>Encrypt or Hash all Authentication Credentials</u></b><br>Encrypt or hash with a salt all authentication credentials when stored.                                                                                                                                                                                                                                                                                                                                    |      |  |  |

## MITRE ATT&CK Mappings:

| Techniques / Sub-techniques | Tactics | Mitigations |
|-----------------------------|---------|-------------|
| T1003, T1003.008            | TA0003  | M1027       |

## 7.2.2 Ensure /etc/shadow password fields are not empty (Automated)

### Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

### Description:

An account with an empty password field means that anybody may log in as that user without providing a password.

### Rationale:

All accounts must have passwords or be locked to prevent the account from being used by an unauthorized user.

### Audit:

Run the following command and verify that no output is returned:

```
# awk -F: '($2 == "" ) { print $1 " does not have a password "}' /etc/shadow
```

### Remediation:

If any accounts in the `/etc/shadow` file do not have a password, run the following command to lock the account until it can be determined why it does not have a password:






```
# passwd -l <username>
```

Also, check to see if the account is logged in and investigate what it is being used for to determine if it needs to be forced off.

### References:

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

## CIS Controls:

| Controls Version | Control                                                                                                                                                                                                                                        | IG 1                                                                                | IG 2                                                                                | IG 3                                                                                |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
| v8               | <b>5.2 Use Unique Passwords</b><br>Use unique passwords for all enterprise assets. Best practice implementation includes, at a minimum, an 8-character password for accounts using MFA and a 14-character password for accounts not using MFA. |  |  |  |
| v7               | <b>4.4 Use Unique Passwords</b><br>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 | TA0003  | M1027       |

### 7.2.3 Ensure all groups in /etc/passwd exist in /etc/group (Automated)

#### Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

#### Description:

Over time, system administration errors and changes can lead to groups being defined in **/etc/passwd** but not in **/etc/group**.

#### Rationale:

Groups defined in the **/etc/passwd** file but not in the **/etc/group** file pose a threat to system security since group permissions are not properly managed.

#### Audit:

Run the following script to verify all GIDs in **/etc/passwd** exist in **/etc/group**:

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

{
  a_passwd_group_gid=("${awk -F: '{print $4}' /etc/passwd | sort -u}")
  a_group_gid=("${awk -F: '{print $3}' /etc/group | sort -u}")
  a_passwd_group_diff=("${printf '%s\n' "${a_group_gid[@]}"
"${a_passwd_group_gid[@]}" | sort | uniq -u}")
  while IFS= read -r l_gid; do
    awk -F: '($4 == '"$l_gid"') {print "  - User: \"" $1 "\" has GID: \""
$4 "\" which does not exist in /etc/group" }' /etc/passwd
  done <<("${printf '%s\n' "${a_passwd_group_gid[@]}"
"${a_passwd_group_diff[@]}" | sort | uniq -D | uniq)
  unset a_passwd_group_gid; unset a_group_gid; unset a_passwd_group_diff
}
```

Nothing should be returned







#### Remediation:

Analyze the output of the Audit step above and perform the appropriate action to correct any discrepancies found.

#### 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               | <b>3.3 <u>Configure Data Access Control Lists</u></b><br>Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications. |  |  |  |
| v8               | <b>14.6 <u>Train Workforce Members on Recognizing and Reporting Security Incidents</u></b><br>Train workforce members to be able to recognize a potential incident and be able to report such an incident.                                                     |  |  |  |

## MITRE ATT&CK Mappings:

| Techniques / Sub-techniques | Tactics | Mitigations |
|-----------------------------|---------|-------------|
| T1222, T1222.002            | TA0003  | M1027       |



## 7.2.4 Ensure shadow group is empty (Automated)

### Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

### Description:

The shadow group allows system programs which require access the ability to read the `/etc/shadow` file. No users should be assigned to the shadow group.

### Rationale:

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

### Audit:

Run the following commands and verify no results are returned:

```
# awk -F: '($1=="shadow") {print $NF}' /etc/group
# awk -F: '($4 == "'$(getent group shadow | awk -F: '{print $3}' | xargs)'"')
{print "  - user: \"" $1 "\" primary group is the shadow group"}' /etc/passwd
```

### Remediation:

Run the following command to remove all users from the shadow group

```
# sed -ri 's/^(shadow:[^:]*:[^:]*:)([^\:]+$)/\1/' /etc/group
```







Change the primary group of any users with shadow as their primary group.

```
# usermod -g <primary group> <user>
```

### References:

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

## CIS Controls:

| Controls Version | Control                                                                                                                                                                                                                                                                                                                                                                                                               | IG 1                                                                                | IG 2                                                                                | IG 3                                                                                |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
| v8               | <b>3.3 <u>Configure Data Access Control Lists</u></b><br>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><br>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            | TA0005  | M1022       |

## 7.2.5 Ensure no duplicate UIDs exist (Automated)

### Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

### Description:

Although the **useradd** program will not let you create a duplicate User ID (UID), it is possible for an administrator to manually edit the **/etc/passwd** file and change the UID field.

### Rationale:

Users must be assigned unique UIDs for accountability and to ensure appropriate access protections.

### Audit:

Run the following script and verify no results are returned:

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

{
  while read -r l_count l_uid; do
    if [ "$l_count" -gt 1 ]; then
      echo -e "Duplicate UID: \"$l_uid\" Users: \"$(awk -F: '($3 == n) { print $1 }' n=$l_uid /etc/passwd | xargs)\""
    fi
    done < <(cut -f3 -d":" /etc/passwd | sort -n | uniq -c)
  }
}
```

### Remediation:

Based on the results of the audit script, establish unique UIDs and review all files owned by the shared UIDs to determine which UID they are supposed to belong to.

### 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 |
|-----------------------------|---------|-------------|
| T1078, T1078.001, T1078.003 | TA0005  | M1027       |

## 7.2.6 Ensure no duplicate GIDs exist (Automated)

### Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

### Description:

Although the **groupadd** program will not let you create a duplicate Group ID (GID), it is possible for an administrator to manually edit the **/etc/group** file and change the GID field.

### Rationale:

User groups must be assigned unique GIDs for accountability and to ensure appropriate access protections.

### Audit:

Run the following script and verify no results are returned:

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

{
  while read -r l_count l_gid; do
    if [ "$l_count" -gt 1 ]; then
      echo -e "Duplicate GID: \"$l_gid\" Groups: \"$(awk -F: '($3 == $l_gid) {
print $1 }' n=$l_gid /etc/group | xargs)\"
    fi
    done << (cut -f3 -d: /etc/group | sort -n | uniq -c)
  }
```

### Remediation:

Based on the results of the audit script, establish unique GIDs and review all files owned by the shared GID to determine which group they are supposed to belong to.

### References:

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

### Additional Information:

You can also use the **grpck** command to check for other inconsistencies in the **/etc/group** file.

**MITRE ATT&CK Mappings:**

| <b>Techniques / Sub-techniques</b> | <b>Tactics</b> | <b>Mitigations</b> |
|------------------------------------|----------------|--------------------|
| T1078, T1078.001,<br>T1078.003     | TA0005         | M1027              |

## 7.2.7 Ensure no duplicate user names exist (Automated)

### Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

### Description:

Although the **useradd** program will not let you create a duplicate user name, it is possible for an administrator to manually edit the **/etc/passwd** file and change the user name.

### Rationale:

If a user is assigned a duplicate user name, it will create and have access to files with the first UID for that username in **/etc/passwd**. For example, if "test4" has a UID of 1000 and a subsequent "test4" entry has a UID of 2000, logging in as "test4" will use UID 1000. Effectively, the UID is shared, which is a security problem.

### Audit:

Run the following script and verify no results are returned:

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

{
  while read -r l_count l_user; do
    if [ "$l_count" -gt 1 ]; then
      echo -e "Duplicate User: \"$l_user\" Users: \"$(awk -F: '($1 == n) {
print $1 }' n=$l_user /etc/passwd | xargs)\""
    fi
  done <<(cut -f1 -d":" /etc/group | sort -n | uniq -c)
}
```

### Remediation:

Based on the results of the audit script, establish unique user names for the users. File ownerships will automatically reflect the change as long as the users have unique UIDs.

### 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 |
|-----------------------------|---------|-------------|
| T1078, T1078.001, T1078.003 | TA0004  | M1027       |

## 7.2.8 Ensure no duplicate group names exist (Automated)

### Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

### Description:

Although the **groupadd** program will not let you create a duplicate group name, it is possible for an administrator to manually edit the **/etc/group** file and change the group name.

### Rationale:

If a group is assigned a duplicate group name, it will create and have access to files with the first GID for that group in **/etc/group** . Effectively, the GID is shared, which is a security problem.

### Audit:

Run the following script and verify no results are returned:

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

{
  while read -r l_count l_group; do
    if [ "$l_count" -gt 1 ]; then
      echo -e "Duplicate Group: \"$l_group\" Groups: \"$(awk -F: '($1 == n) { print $1 }' n=$l_group /etc/group | xargs)\""
      fi
    done <<(cut -f1 -d":" /etc/group | sort -n | uniq -c)
  }
```

### Remediation:

Based on the results of the audit script, establish unique names for the user groups. File group ownerships will automatically reflect the change as long as the groups have unique GIDs.

### References:

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



**MITRE ATT&CK Mappings:**

| <b>Techniques / Sub-techniques</b> | <b>Tactics</b> | <b>Mitigations</b> |
|------------------------------------|----------------|--------------------|
| T1078, T1078.001,<br>T1078.003     | TA0004         | M1027              |

## 7.2.9 Ensure local interactive user home directories are configured (Automated)

### Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

### Description:

The user home directory is space defined for the particular user to set local environment variables and to store personal files. While the system administrator can establish secure permissions for users' home directories, the users can easily override these. Users can be defined in `/etc/passwd` without a home directory or with a home directory that does not actually exist.

### Rationale:

Since the user is accountable for files stored in the user home directory, the user must be the owner of the directory. Group or world-writable user home directories may enable malicious users to steal or modify other users' data or to gain another user's system privileges. If the user's home directory does not exist or is unassigned, the user will be placed in "/" and will not be able to write any files or have local environment variables set.

### Audit:

Run the following script to Ensure:

- local interactive user home directories exist
- Ensure local interactive users own their home directories
- Ensure local interactive user home directories are mode 750 or more restrictive

```

#!/usr/bin/env bash

{
    l_output="" l_output2="" l_heout2="" l_hoout2="" l_haout2=""
    l_valid_shells="^($( awk -F\ / '$NF != "nologin" {print}' /etc/shells | sed
-rn '/^\//{s,/,\//,g;p}' | paste -s -d '|' - ))$"
    unset a_uarr && a_uarr=() # Clear and initialize array
    while read -r l_epu l_eph; do # Populate array with users and user home
location
        a_uarr+=("$l_epu $l_eph")
        done <<< "$ (awk -v pat="$l_valid_shells" -F: '$(NF) ~ pat { print $1 " "
$(NF-1) }' /etc/passwd)"
        l_asize="${#a_uarr[@]}" # Here if we want to look at number of users
before proceeding
        [ "$l_asize" -gt "10000" ] && echo -e "\n ** INFO **\n - \"$l_asize\"
Local interactive users found on the system\n - This may be a long running
check\n"
        while read -r l_user l_home; do
            if [ -d "$l_home" ]; then
                l_mask='0027'
                l_max="$( printf '%o' $(( 0777 & ~$l_mask )) )"
                while read -r l_own l_mode; do
                    [ "$l_user" != "$l_own" ] && l_hoout2="$l_hoout2\n - User:
\"$l_user\" Home \"$l_home\" is owned by: \"$l_own\""
                    if [ $(( $l_mode & $l_mask )) -gt 0 ]; then
                        l_haout2="$l_haout2\n - User: \"$l_user\" Home \"$l_home\" is
mode: \"$l_mode\" should be mode: \"$l_max\" or more restrictive"
                    fi
                    done <<< "$(stat -Lc '%U %#a' "$l_home")"
                else
                    l_heout2="$l_heout2\n - User: \"$l_user\" Home \"$l_home\" Doesn't
exist"
                fi
                done <<< "$(printf '%s\n' "${a_uarr[@]}")"
                [ -z "$l_heout2" ] && l_output="$l_output\n - home directories exist" ||
l_output2="$l_output2$l_heout2"
                [ -z "$l_hoout2" ] && l_output="$l_output\n - own their home directory"
|| l_output2="$l_output2$l_hoout2"
                [ -z "$l_haout2" ] && l_output="$l_output\n - home directories are mode:
\"$l_max\" or more restrictive" || l_output2="$l_output2$l_haout2"
                [ -n "$l_output" ] && l_output=" - All local interactive users:$l_output"
                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"
                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"
                fi
            fi
        }
}

```

**Remediation:**

If a local interactive users' home directory is undefined and/or doesn't exist, follow local site policy and perform one of the following:

- Lock the user account
- Remove the user from the system
- create a directory for the user. If undefined, edit `/etc/passwd` and add the absolute path to the directory to the last field of the user.

Run the following script to:

- Remove excessive permissions from local interactive users home directories
- Update the home directory's owner







```

#!/usr/bin/env bash

{
    l_output2=""
    l_valid_shells="^( $( awk -F\ / ' $NF != "nologin" {print}' /etc/shells | sed
-rn '/^\\/{s/,/,\\\\/,g;p}' | paste -s -d '|' - ) ) $"
    unset a_uarr && a_uarr=() # Clear and initialize array
    while read -r l_epu l_eph; do # Populate array with users and user home
location
        a_uarr+=("$l_epu $l_eph")
        done <<< "$(awk -v pat="$l_valid_shells" -F: '$(NF) ~ pat { print $1 " "
$(NF-1) }' /etc/passwd)"
        l_asize="${#a_uarr[@]}" # Here if we want to look at number of users
before proceeding
        [ "$l_asize" -gt "10000" ] && echo -e "\n ** INFO **\n - \"$l_asize\"
Local interactive users found on the system\n - This may be a long running
process\n"
        while read -r l_user l_home; do
            if [ -d "$l_home" ]; then
                l_mask='0027'
                l_max="$( printf '%o' $( ( 0777 & ~$l_mask ) )"
                while read -r l_own l_mode; do
                    if [ "$l_user" != "$l_own" ]; then
                        l_output2="$l_output2\n - User: \"$l_user\" Home \"$l_home\"
is owned by: \"$l_own\" \n - changing ownership to: \"$l_user\" \n"
                        chown "$l_user" "$l_home"
                    fi
                    if [ $( ( $l_mode & $l_mask ) ) -gt 0 ]; then
                        l_output2="$l_output2\n - User: \"$l_user\" Home \"$l_home\"
is mode: \"$l_mode\" should be mode: \"$l_max\" or more restrictive\n -
removing excess permissions\n"
                        chmod g-w,o-rwx "$l_home"
                    fi
                done <<< "$(stat -Lc '%U %#a' "$l_home")"
            else
                l_output2="$l_output2\n - User: \"$l_user\" Home \"$l_home\"
Doesn't exist\n - Please create a home in accordance with local site
policy"
            fi
            done <<< "$(printf '%s\n' "${a_uarr[@]}")"
            if [ -z "$l_output2" ]; then # If l_output2 is empty, we pass
                echo -e " - No modification needed to local interactive users home
directories"
            else
                echo -e "\n$l_output2"
            fi
        }
}

```

## CIS Controls:

| Controls Version | Control                                                                                                                                                                                                                                                                                                                                                                                                               | IG 1                                                                                | IG 2                                                                                | IG 3                                                                                |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
| v8               | <b>3.3 <u>Configure Data Access Control Lists</u></b><br>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><br>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            | TA0005  | M1022       |

## 7.2.10 Ensure local interactive user dot files access is configured (Automated)

### Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

### Description:

While the system administrator can establish secure permissions for users' "dot" files, the users can easily override these.

- **.forward** file specifies an email address to forward the user's mail to.
- **.rhost** file provides the "remote authentication" database for the rcp, rlogin, and rsh commands and the rcmd() function. These files bypass the standard password-based user authentication mechanism. They specify remote hosts and users that are considered trusted (i.e. are allowed to access the local system without supplying a password)
- **.netrc** file contains data for logging into a remote host or passing authentication to an API.
- **.bash\_history** file keeps track of the user's commands.

### Rationale:

User configuration files with excessive or incorrect access may enable malicious users to steal or modify other users' data or to gain another user's system privileges.

### Audit:

Run the following script to verify local interactive user dot files:

- Don't include **.forward**, **.rhost**, or **.netrc** files
- Are mode 0644 or more restrictive
- Are owned by the local interactive user
- Are group owned by the user's primary group
- **.bash\_history** is mode 0600 or more restrictive

**Note:** If a **.netrc** file is required, and follows local site policy, it should be mode **0600** or more restrictive.

```

#!/usr/bin/env bash

{
    a_output2=(); a_output3=()
    l_maxsize="1000" # Maximum number of local interactive users before
warning (Default 1,000)
    l_valid_shells="^($( awk -F\ / ' $NF != "nologin" {print}' /etc/shells | sed
-rn '/^\/\/{s,/,\ \ \ \/,g;p}' | paste -s -d '|' - ))$"
    a_user_and_home=() # Create array with local users and their home
directories
    while read -r l_local_user l_local_user_home; do # Populate array with
users and user home location
        [[ -n "$l_local_user" && -n "$l_local_user_home" ]] &&
a_user_and_home+=("$l_local_user:$l_local_user_home")
        done <<< "$ (awk -v pat="$l_valid_shells" -F: '$(NF) ~ pat { print $1 " "
$(NF-1) }' /etc/passwd)"
        l_asize="${#a_user_and_home[@]}" # Here if we want to look at number of
users before proceeding
        [ "${#a_user_and_home[@]}" -gt "$l_maxsize" ] && printf '%s\n' "" " **
INFO **" \
        " - \"$l_asize\" Local interactive users found on the system" \
        " - This may be a long running check" ""
        file_access_chk()
        {
            a_access_out=()
            l_max="$( printf '%o' $(( 0777 & ~$l_mask )) )"
            if [ $(( $l_mode & $l_mask )) -gt 0 ]; then
                a_access_out+=(" - File: \"$l_hdfilename\" is mode: \"$l_mode\" and
should be mode: \"$l_max\" or more restrictive")
            fi
            if [[ ! "$l_owner" =~ ($l_user) ]]; then
                a_access_out+=(" - File: \"$l_hdfilename\" owned by: \"$l_owner\" and
should be owned by \"${l_user} // or \"")
            fi
            if [[ ! "$l_gowner" =~ ($l_group) ]]; then
                a_access_out+=(" - File: \"$l_hdfilename\" group owned by:
\"$l_gowner\" and should be group owned by \"${l_group} // or \"")
            fi
        }
        while IFS=: read -r l_user l_home; do
            a_dot_file=(); a_netrc=(); a_netrc_warn=(); a_bhout=(); a_hdirout=()
            if [ -d "$l_home" ]; then
                l_group="$(id -gn "$l_user" | xargs)"; l_group="${l_group} // ||"
                while IFS= read -r -d $'\0' l_hdfilename; do
                    while read -r l_mode l_owner l_gowner; do
                        case "$(basename "$l_hdfilename")" in
                            .forward | .rhost )
                                a_dot_file+=(" - File: \"$l_hdfilename\" exists") ;;
                            .netrc )
                                l_mask='0177'; file_access_chk
                                if [ "${#a_access_out[@]}" -gt 0 ]; then
                                    a_netrc+=("${a_access_out[@]}")
                                else
                                    a_netrc_warn+=(" - File: \"$l_hdfilename\" exists")
                                fi ;;
                            .bash_history )
                                l_mask='0177'; file_access_chk

```



```

[ "${#a_access_out[@]}" -gt 0 ] &&
a_bhout+="${a_access_out[@]}" ;;
* )
l_mask='0133'; file_access_chk
[ "${#a_access_out[@]}" -gt 0 ] &&
a_hdirout+="${a_access_out[@]}" ;;
esac
done < <(stat -Lc '%#a %U %G' "$l_hdfilename")
done < <(find "$l_home" -xdev -type f -name '.*' -print0)
fi
if [[ "${#a_dot_file[@]}" -gt 0 || "${#a_netrc[@]}" -gt 0 ||
"${#a_bhout[@]}" -gt 0 || "${#a_hdirout[@]}" -gt 0 ]]; then
a_output2+="- User: \"$l_user\" Home Directory: \"$l_home\"\"
"${a_dot_file[@]}" "${a_netrc[@]}" "${a_bhout[@]}" "${a_hdirout[@]}"
fi
[ "${#a_netrc_warn[@]}" -gt 0 ] && a_output3+="- User: \"$l_user\"
Home Directory: \"$l_home\"\" "${a_netrc_warn[@]}"
done <<< "$(printf '%s\n' "${a_user_and_home[@]}")"
if [ "${#a_output2[@]}" -le 0 ]; then # If l_output2 is empty, we pass
[ "${#a_output3[@]}" -gt 0 ] && printf '%s\n' " ** WARNING **"
"${a_output3[@]}"
printf '%s\n' "- Audit Result:" " ** PASS **"
else
printf '%s\n' "- Audit Result:" " ** FAIL **" " - * Reasons for audit
failure * : " "${a_output2[@]}" ""
[ "${#a_output3[@]}" -gt 0 ] && printf '%s\n' " ** WARNING **"
"${a_output3[@]}"
fi
}

```

## Remediation:

Making global modifications to users' files without alerting the user community can result in unexpected outages and unhappy users. Therefore, it is recommended that a monitoring policy be established to report user dot file permissions and determine the action to be taken in accordance with site policy.

The following script will:

- remove excessive permissions on **dot** files within interactive users' home directories
- change ownership of **dot** files within interactive users' home directories to the user
- change group ownership of **dot** files within interactive users' home directories to the user's primary group
- list **.forward** and **.rhost** files to be investigated and manually deleted

```

#!/usr/bin/env bash

{
    a_output2=(); a_output3=()
    l_maxsize="1000" # Maximum number of local interactive users before
warning (Default 1,000)
    l_valid_shells="^($( awk -F\ / ' $NF != "nologin" {print}' /etc/shells | sed
-rn '/^\/\/{s,/,\|\\\|/,g;p}' | paste -s -d '|' - ))$"
    a_user_and_home=() # Create array with local users and their home
directories
    while read -r l_local_user l_local_user_home; do # Populate array with
users and user home location
        [[ -n "$l_local_user" && -n "$l_local_user_home" ]] &&
a_user_and_home+=("$l_local_user:$l_local_user_home")
        done <<< "$ (awk -v pat="$l_valid_shells" -F: '$(NF) ~ pat { print $1 " "
$(NF-1) }' /etc/passwd)"
        l_asize="${#a_user_and_home[@]}" # Here if we want to look at number of
users before proceeding
        [ "${#a_user_and_home[@]}" -gt "$l_maxsize" ] && printf '%s\n' "" " **
INFO **" \
        " - \"$l_asize\" Local interactive users found on the system" \
        " - This may be a long running check" ""
        file_access_fix()
        {
            a_access_out=()
            l_max="$( printf '%o' $(( 0777 & ~$l_mask)) )"
            if [ $(( $l_mode & $l_mask )) -gt 0 ]; then
                printf '%s\n' "" " - File: \"$l_hdfilename\" is mode: \"$l_mode\" and
should be mode: \"$l_max\" or more restrictive" \
                "      Updating file: \"$l_hdfilename\" to be mode: \"$l_max\" or more
restrictive"
                chmod "$l_change" "$l_hdfilename"
            fi
            if [[ ! "$l_owner" =~ ($l_user) ]]; then
                printf '%s\n' "" " - File: \"$l_hdfilename\" owned by: \"$l_owner\" and
should be owned by \"$${l_user//|/ or }\"" \
                "      Updating file: \"$l_hdfilename\" to be owned by \"$${l_user//|/ or
}\""
                chown "$l_user" "$l_hdfilename"
            fi
            if [[ ! "$l_gowner" =~ ($l_group) ]]; then
                printf '%s\n' "" " - File: \"$l_hdfilename\" group owned by:
\"$l_gowner\" and should be group owned by \"$${l_group//|/ or }\"" \
                "      Updating file: \"$l_hdfilename\" to be group owned by
\"$${l_group//|/ or }\""
                chgrp "$l_group" "$l_hdfilename"
            fi
        }
        while IFS=: read -r l_user l_home; do
            a_dot_file=(); a_netrc=(); a_netrc_warn=(); a_bhout=(); a_hdirout=()
            if [ -d "$l_home" ]; then
                l_group="$(id -gn "$l_user" | xargs)"; l_group="$${l_group//|/}"
                while IFS= read -r -d $'\0' l_hdfilename; do
                    while read -r l_mode l_owner l_gowner; do
                        case "$(basename "$l_hdfilename")" in
                            .forward | .rhost )
                                a_dot_file+=(" - File: \"$l_hdfilename\" exists" "







```

```
Please review and manually delete this file") ;;
.netrc )
    l_mask='0177'; l_change="u-x,go-rwx"; file_access_fix
    a_netrc_warn+=("    - File: \"\$l_hdfilename\" exists") ;;
.bash_history )
    l_mask='0177'; l_change="u-x,go-rwx"; file_access_fix ;;
* )
    l_mask='0133'; l_change="u-x,go-wx"; file_access_fix ;;
esac
done < <(stat -Lc '%#a %U %G' "\$l_hdfilename")
done < <(find "\$l_home" -xdev -type f -name '*' -print0)
fi
[ "\${#a_dot_file[@]}" -gt 0 ] && a_output2+=(" - User: \"\$l_user\" Home
Directory: \"\$l_home\" \"\${a_dot_file[@]}")
[ "\${#a_netrc_warn[@]}" -gt 0 ] && a_output3+=(" - User: \"\$l_user\"
Home Directory: \"\$l_home\" \"\${a_netrc_warn[@]}")
done <<< "\$(printf '%s\n' "\${a_user_and_home[@]}")"
[ "\${#a_output3[@]}" -gt 0 ] && printf '%s\n' "" " ** WARNING **"
"\${a_output3[@]}" ""
[ "\${#a_output2[@]}" -gt 0 ] && printf '%s\n' "" "\${a_output2[@]}"
}
```

## 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               | <b>3.3 <u>Configure Data Access Control Lists</u></b><br>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><br>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, T1222.001,<br>T1222.002, T1552,<br>T1552.003, T1552.004 | TA0005  | M1022       |