

Recommendations

1 Initial Setup

Items in this section are advised for all systems but may be difficult or require extensive preparation after the initial setup of the system.

1.1 Filesystem

The file system is generally a built-in layer used to handle the data management of the storage.

1.1.1 Configure Filesystem Kernel Modules

Several uncommon filesystem types are supported under Linux. Removing support for unneeded filesystem types reduces the local attack surface of the system. If a filesystem type is not needed it should be disabled. Native Linux file systems are designed to ensure that built-in security controls function as expected. Non-native filesystems can lead to unexpected consequences to both the security and functionality of the system and should be used with caution. Many filesystems are created for niche use cases and are not maintained and supported as the operating systems are updated and patched. Users of non-native filesystems should ensure that there is attention and ongoing support for them, especially in light of frequent operating system changes.

Standard network connectivity and Internet access to cloud storage may make the use of non-standard filesystem formats to directly attach heterogeneous devices much less attractive.

Note: This should not be considered a comprehensive list of filesystems. You may wish to consider additions to those listed here for your environment. For the current available file system modules on the system see `/usr/lib/modules/$(uname -r)/kernel/fs`

Start up scripts

Kernel modules loaded directly via `insmod` will ignore what is configured in the relevant `/etc/modprobe.d/*.conf` files. If modules are still being loaded after a reboot whilst having the correctly configured `blacklist` and `install` command, check for `insmod` entries in start up scripts such as `.bashrc`.

You may also want to check `/lib/modprobe.d/`. Please note that this directory should not be used for user defined module loading. Ensure that all such entries resides in `/etc/modprobe.d/*.conf` files.

Return values

Using `/bin/false` as the command in disabling a particular module serves two purposes; to convey the meaning of the entry to the user and cause a non-zero return value. The latter can be tested for in scripts. Please note that `insmod` will ignore what is configured in the relevant `/etc/modprobe.d/*.conf` files. The preferred way to load modules is with `modprobe`.

1.1.1.1 Ensure cramfs kernel module is not available (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The **cramfs** filesystem type is a compressed read-only Linux filesystem embedded in small footprint systems. A **cramfs** image can be used without having to first decompress the image.

Rationale:

Removing support for unneeded filesystem types reduces the local attack surface of the system. If this filesystem type is not needed, disable it.

Audit:

Run the following script to verify:

- **IF** - the **cramfs** kernel module is available in ANY installed kernel, verify:

- An entry including **/bin/true** or **/bin/false** exists in a file within the **/etc/modprobe.d/** directory
- The module is deny listed in a file within the **/etc/modprobe.d/** directory
- The module is not loaded in the running kernel

- **IF** - the **cramfs** kernel module is not available on the system, or pre-compiled into the kernel, no additional configuration is necessary

```

#!/usr/bin/env bash

{
    a_output=() a_output2=() a_output3=() l_dl="" l_mod_name="cramfs"
    l_mod_type="fs"
    l_mod_path="$(readlink -f /lib/modules/**/kernel/$l_mod_type | sort -u)"
    f_module_chk()
    {
        l_dl="y" a_showconfig=()
        while IFS= read -r l_showconfig; do
            a_showconfig+=("$l_showconfig")
            done < <(modprobe --showconfig | grep -P --
'\b(install|blacklist)\h+'${l_mod_chk_name//-/}_'\b')
            if ! lsmmod | grep "$l_mod_chk_name" &> /dev/null; then
                a_output+=(" - kernel module: \"$l_mod_name\" is not loaded")
            else
                a_output2+=(" - kernel module: \"$l_mod_name\" is loaded")
            fi
            if grep -Pq -- '\bbinstall\h+'${l_mod_chk_name//-/}_
/_}'\h+(\usr)?\bin\/(true|false)\b' <<< "${a_showconfig[*]}"; then
                a_output+=(" - kernel module: \"$l_mod_name\" is not loadable")
            else
                a_output2+=(" - kernel module: \"$l_mod_name\" is loadable")
            fi
            if grep -Pq -- '\bbblacklist\h+'${l_mod_chk_name//-/}_'\b' <<<
"${a_showconfig[*]}"; then
                a_output+=(" - kernel module: \"$l_mod_name\" is deny listed")
            else
                a_output2+=(" - kernel module: \"$l_mod_name\" is not deny listed")
            fi
        }
        for l_mod_base_directory in $l_mod_path; do
            if [ -d "$l_mod_base_directory/${l_mod_name//-/}/" ] && [ -n "$(ls -A
"$l_mod_base_directory/${l_mod_name//-/}/")" ]; then
                a_output3+=(" - \"$l_mod_base_directory\"")
                l_mod_chk_name="$l_mod_name"
                [[ "$l_mod_name" =~ overlay ]] && l_mod_chk_name="$l_mod_name:-2"
                [ "$l_dl" != "y" ] && f_module_chk
            else
                a_output+=(" - kernel module: \"$l_mod_name\" doesn't exist in
\"$l_mod_base_directory\"")
            fi
        done
        [ "${#a_output3[@]}" -gt 0 ] && printf '%s\n' "" " -- INFO --" " - module:
\"$l_mod_name\" exists in:" "${a_output3[@]}"
        if [ "${#a_output2[@]}" -le 0 ]; then
            printf '%s\n' "" "- Audit Result:" " ** PASS **" "${a_output[@]}"
        else
            printf '%s\n' "" "- Audit Result:" " ** FAIL **" " - Reason(s) for
audit failure:" "${a_output2[@]}"
            [ "${#a_output[@]}" -gt 0 ] && printf '%s\n' "- Correctly set:"
"${a_output[@]}"
        fi
    }
}

```

Remediation:

Run the following script to unload and disable the `cramfs` module:

- **IF** - the `cramfs` kernel module is available in ANY installed kernel:

- Create a file ending in `.conf` with `install cramfs /bin/false` in the `/etc/modprobe.d/` directory
- Create a file ending in `.conf` with `blacklist cramfs` in the `/etc/modprobe.d/` directory
- Run `modprobe -r cramfs 2>/dev/null; rmmod cramfs 2>/dev/null` to remove `cramfs` from the kernel

- **IF** - the `cramfs` kernel module is not available on the system, or pre-compiled into the kernel, no remediation is necessary





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

{
  a_output2=() a_output3=() l_dl="" l_mod_name="cramfs" l_mod_type="fs"
  l_mod_path="$(readlink -f /lib/modules/**/kernel/$l_mod_type | sort -u)"
  f_module_fix()
  {
    l_dl="y" a_showconfig=()
    while IFS= read -r l_showconfig; do
      a_showconfig+=("$l_showconfig")
      done < <(modprobe --showconfig | grep -P --
'\b(install|blacklist)\h+' "${l_mod_chk_name//-/}_"' \b')
      if lsmod | grep "$l_mod_chk_name" &> /dev/null; then
        a_output2+=(" - unloading kernel module: \"$l_mod_name\"")
        modprobe -r "$l_mod_chk_name" 2>/dev/null; rmmod "$l_mod_name"
2>/dev/null
      fi
      if ! grep -Pq -- '\binstall\h+' "${l_mod_chk_name//-/}_"' \b' <<<
"${a_showconfig[*]}"; then
        a_output2+=(" - setting kernel module: \"$l_mod_name\" to
\"$(readlink -f /bin/false)\"")
        printf '%s\n' "install $l_mod_chk_name $(readlink -f /bin/false)" >>
/etc/modprobe.d/"$l_mod_name".conf
      fi
      if ! grep -Pq -- '\bblacklist\h+' "${l_mod_chk_name//-/}_"' \b' <<<
"${a_showconfig[*]}"; then
        a_output2+=(" - denylisting kernel module: \"$l_mod_name\"")
        printf '%s\n' "blacklist $l_mod_chk_name" >>
/etc/modprobe.d/"$l_mod_name".conf
      fi
    }
    for l_mod_base_directory in $l_mod_path; do # Check if the module exists
on the system
      if [ -d "$l_mod_base_directory/${l_mod_name//-/}/" ] && [ -n "$(ls -A
"$l_mod_base_directory/${l_mod_name//-/}/")" ]; then
        a_output3+=(" - \"$l_mod_base_directory\"")
        l_mod_chk_name="$l_mod_name"
        [[ "$l_mod_name" =~ overlay ]] && l_mod_chk_name="${l_mod_name::-2}"
        [ "$l_dl" != "y" ] && f_module_fix
      else
        printf '%s\n' " - kernel module: \"$l_mod_name\" doesn't exist in
\"$l_mod_base_directory\""
      fi
    done
    [ "${#a_output3[@]}" -gt 0 ] && printf '%s\n' "" " -- INFO --" " - module:
\"$l_mod_name\" exists in:" "${a_output3[@]}"
    [ "${#a_output2[@]}" -gt 0 ] && printf '%s\n' "" "${a_output2[@]}" ||
printf '%s\n' "" " - No changes needed"
    printf '%s\n' "" " - remediation of kernel module: \"$l_mod_name\"
complete" ""
  }
}
```

References:

1. NIST SP 800-53 Rev. 5: CM-7
2. STIG Finding ID: V-230498

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<u>4.8 Uninstall or Disable Unnecessary Services on Enterprise Assets and Software</u> Uninstall or disable unnecessary services on enterprise assets and software, such as an unused file sharing service, web application module, or service function.			
v7	<u>9.2 Ensure Only Approved Ports, Protocols and Services Are Running</u> Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1005, T1005.000	TA0005	M1050

1.1.1.2 Ensure freevxfs kernel module is not available (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The **freevxfs** filesystem type is a free version of the Veritas type filesystem. This is the primary filesystem type for HP-UX operating systems.

Rationale:

Removing support for unneeded filesystem types reduces the local attack surface of the system. If this filesystem type is not needed, disable it.

Audit:

Run the following script to verify:

- **IF** - the **freevxfs** kernel module is available in ANY installed kernel, verify:

- An entry including **/bin/true** or **/bin/false** exists in a file within the **/etc/modprobe.d/** directory
- The module is deny listed in a file within the **/etc/modprobe.d/** directory
- The module is not loaded in the running kernel

- **IF** - the **freevxfs** kernel module is not available on the system, or pre-compiled into the kernel, no additional configuration is necessary

```

#!/usr/bin/env bash

{
    a_output=() a_output2=() a_output3=() l_dl="" l_mod_name="freevxfs"
    l_mod_type="fs"
    l_mod_path="$(readlink -f /lib/modules/**/kernel/$l_mod_type | sort -u)"
    f_module_chk()
    {
        l_dl="y" a_showconfig=()
        while IFS= read -r l_showconfig; do
            a_showconfig+=("$l_showconfig")
            done < <(modprobe --showconfig | grep -P --
'\b(install|blacklist)\h+'${l_mod_chk_name//-/}_'\b')
            if ! lsmod | grep "$l_mod_chk_name" &> /dev/null; then
                a_output+=(" - kernel module: \"$l_mod_name\" is not loaded")
            else
                a_output2+=(" - kernel module: \"$l_mod_name\" is loaded")
            fi
            if grep -Pq -- '\binstall\h+'${l_mod_chk_name//-/}_
/_}'\h+(\usr)?\bin\/(true|false)\b' <<< "${a_showconfig[*]}"; then
                a_output+=(" - kernel module: \"$l_mod_name\" is not loadable")
            else
                a_output2+=(" - kernel module: \"$l_mod_name\" is loadable")
            fi
            if grep -Pq -- '\bblacklist\h+'${l_mod_chk_name//-/}_'\b' <<<
"${a_showconfig[*]}"; then
                a_output+=(" - kernel module: \"$l_mod_name\" is deny listed")
            else
                a_output2+=(" - kernel module: \"$l_mod_name\" is not deny listed")
            fi
        }
        for l_mod_base_directory in $l_mod_path; do
            if [ -d "$l_mod_base_directory/${l_mod_name//-/}/" ] && [ -n "$(ls -A
"$l_mod_base_directory/${l_mod_name//-/}/")" ]; then
                a_output3+=(" - \"$l_mod_base_directory\"")
                l_mod_chk_name="$l_mod_name"
                [[ "$l_mod_name" =~ overlay ]] && l_mod_chk_name="$l_mod_name:-2"
                [ "$l_dl" != "y" ] && f_module_chk
            else
                a_output+=(" - kernel module: \"$l_mod_name\" doesn't exist in
\"$l_mod_base_directory\"")
            fi
        done
        [ "${#a_output3[@]}" -gt 0 ] && printf '%s\n' "" " -- INFO --" " - module:
\"$l_mod_name\" exists in:" "${a_output3[@]}"
        if [ "${#a_output2[@]}" -le 0 ]; then
            printf '%s\n' "" "- Audit Result:" " ** PASS **" "${a_output[@]}"
        else
            printf '%s\n' "" "- Audit Result:" " ** FAIL **" " - Reason(s) for
audit failure:" "${a_output2[@]}"
            [ "${#a_output[@]}" -gt 0 ] && printf '%s\n' "- Correctly set:"
"${a_output[@]}"
        fi
    }
}

```

Remediation:

Run the following script to unload and disable the `freevxfs` module:

- **IF** - the `freevxfs` kernel module is available in ANY installed kernel:

- Create a file ending in `.conf` with `install freevxfs /bin/false` in the `/etc/modprobe.d/` directory
- Create a file ending in `.conf` with `blacklist freevxfs` in the `/etc/modprobe.d/` directory
- Run `modprobe -r freevxfs 2>/dev/null; rmmod freevxfs 2>/dev/null` to remove `freevxfs` from the kernel

- **IF** - the `freevxfs` kernel module is not available on the system, or pre-compiled into the kernel, no remediation is necessary





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

{
  a_output2=() a_output3=() l_dl="" l_mod_name="freevxfs" l_mod_type="fs"
  l_mod_path="$(readlink -f /lib/modules/**/kernel/$l_mod_type | sort -u)"
  f_module_fix()
  {
    l_dl="y" a_showconfig=()
    while IFS= read -r l_showconfig; do
      a_showconfig+=("$l_showconfig")
      done < <(modprobe --showconfig | grep -P --
'\b(install|blacklist)\h+'${l_mod_chk_name//-/}_'\b')
      if lsmod | grep "$l_mod_chk_name" &> /dev/null; then
        a_output2+=(" - unloading kernel module: \"$l_mod_name\"")
        modprobe -r "$l_mod_chk_name" 2>/dev/null; rmmod "$l_mod_name"
2>/dev/null
      fi
      if ! grep -Pq -- '\binstall\h+'${l_mod_chk_name//-/}_'\b' <<<
"${a_showconfig[*]}"; then
        a_output2+=(" - setting kernel module: \"$l_mod_name\" to
\"$(readlink -f /bin/false)\"")
        printf '%s\n' "install $l_mod_chk_name $(readlink -f /bin/false)" >>
/etc/modprobe.d/"$l_mod_name".conf
      fi
      if ! grep -Pq -- '\bblacklist\h+'${l_mod_chk_name//-/}_'\b' <<<
"${a_showconfig[*]}"; then
        a_output2+=(" - denylisting kernel module: \"$l_mod_name\"")
        printf '%s\n' "blacklist $l_mod_chk_name" >>
/etc/modprobe.d/"$l_mod_name".conf
      fi
    }
    for l_mod_base_directory in $l_mod_path; do # Check if the module exists
on the system
      if [ -d "$l_mod_base_directory/${l_mod_name//-/}/" ] && [ -n "$(ls -A
"$l_mod_base_directory/${l_mod_name//-/}/")" ]; then
        a_output3+=(" - \"$l_mod_base_directory\"")
        l_mod_chk_name="$l_mod_name"
        [[ "$l_mod_name" =~ overlay ]] && l_mod_chk_name="${l_mod_name::-2}"
        [ "$l_dl" != "y" ] && f_module_fix
      else
        printf '%s\n' " - kernel module: \"$l_mod_name\" doesn't exist in
\"$l_mod_base_directory\""
      fi
    done
    [ "${#a_output3[@]}" -gt 0 ] && printf '%s\n' "" " -- INFO --" " - module:
\"$l_mod_name\" exists in:" "${a_output3[@]}"
    [ "${#a_output2[@]}" -gt 0 ] && printf '%s\n' "" "${a_output2[@]}" ||
printf '%s\n' "" " - No changes needed"
    printf '%s\n' "" " - remediation of kernel module: \"$l_mod_name\"
complete" ""
  }
}
```

References:

1. NIST SP 800-53 Rev. 5: CM-7

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<u>4.8 Uninstall or Disable Unnecessary Services on Enterprise Assets and Software</u> Uninstall or disable unnecessary services on enterprise assets and software, such as an unused file sharing service, web application module, or service function.			
v7	<u>9.2 Ensure Only Approved Ports, Protocols and Services Are Running</u> Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1005, T1005.000	TA0005	M1050

1.1.1.3 Ensure hfs kernel module is not available (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The **hfs** filesystem type is a hierarchical filesystem that allows you to mount Mac OS filesystems.

Rationale:

Removing support for unneeded filesystem types reduces the local attack surface of the system. If this filesystem type is not needed, disable it.

Audit:

Run the following script to verify:

- **IF** - the **hfs** kernel module is available in ANY installed kernel, verify:

- An entry including **/bin/true** or **/bin/false** exists in a file within the **/etc/modprobe.d/** directory
- The module is deny listed in a file within the **/etc/modprobe.d/** directory
- The module is not loaded in the running kernel

- **IF** - the **hfs** kernel module is not available on the system, or pre-compiled into the kernel, no additional configuration is necessary

```

#!/usr/bin/env bash

{
    a_output=() a_output2=() a_output3=() l_dl="" l_mod_name="hfs"
    l_mod_type="fs"
    l_mod_path="$(readlink -f /lib/modules/**/kernel/$l_mod_type | sort -u)"
    f_module_chk()
    {
        l_dl="y" a_showconfig=()
        while IFS= read -r l_showconfig; do
            a_showconfig+=("$l_showconfig")
            done < <(modprobe --showconfig | grep -P --
'\b(install|blacklist)\h+'${l_mod_chk_name//-/}_'\b')
            if ! lsmod | grep "$l_mod_chk_name" &> /dev/null; then
                a_output+=(" - kernel module: \"$l_mod_name\" is not loaded")
            else
                a_output2+=(" - kernel module: \"$l_mod_name\" is loaded")
            fi
            if grep -Pq -- '\bbinstall\h+'${l_mod_chk_name//-/}_
/_}'\h+(\usr)?\bin\/(true|false)\b' <<< "${a_showconfig[*]}"; then
                a_output+=(" - kernel module: \"$l_mod_name\" is not loadable")
            else
                a_output2+=(" - kernel module: \"$l_mod_name\" is loadable")
            fi
            if grep -Pq -- '\bbblacklist\h+'${l_mod_chk_name//-/}_'\b' <<<
"${a_showconfig[*]}"; then
                a_output+=(" - kernel module: \"$l_mod_name\" is deny listed")
            else
                a_output2+=(" - kernel module: \"$l_mod_name\" is not deny listed")
            fi
        }
        for l_mod_base_directory in $l_mod_path; do
            if [ -d "$l_mod_base_directory/${l_mod_name//-/}/" ] && [ -n "$(ls -A
"$l_mod_base_directory/${l_mod_name//-/}/")" ]; then
                a_output3+=(" - \"$l_mod_base_directory\"")
                l_mod_chk_name="$l_mod_name"
                [[ "$l_mod_name" =~ overlay ]] && l_mod_chk_name="$l_mod_name:-2"
                [ "$l_dl" != "y" ] && f_module_chk
            else
                a_output+=(" - kernel module: \"$l_mod_name\" doesn't exist in
\"$l_mod_base_directory\"")
            fi
        done
        [ "${#a_output3[@]}" -gt 0 ] && printf '%s\n' "" " -- INFO --" " - module:
\"$l_mod_name\" exists in:" "${a_output3[@]}"
        if [ "${#a_output2[@]}" -le 0 ]; then
            printf '%s\n' "" "- Audit Result:" " ** PASS **" "${a_output[@]}"
        else
            printf '%s\n' "" "- Audit Result:" " ** FAIL **" " - Reason(s) for
audit failure:" "${a_output2[@]}"
            [ "${#a_output[@]}" -gt 0 ] && printf '%s\n' "- Correctly set:"
"${a_output[@]}"
        fi
    }
}

```

Remediation:

Run the following script to unload and disable the `hfs` module:

- **IF** - the `hfs` kernel module is available in ANY installed kernel:

- Create a file ending in `.conf` with `install hfs /bin/false` in the `/etc/modprobe.d/` directory
- Create a file ending in `.conf` with `blacklist hfs` in the `/etc/modprobe.d/` directory
- Run `modprobe -r hfs 2>/dev/null; rmmod hfs 2>/dev/null` to remove `hfs` from the kernel

- **IF** - the `hfs` kernel module is not available on the system, or pre-compiled into the kernel, no remediation is necessary






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

{
  a_output2=() a_output3=() l_dl="" l_mod_name="hfs" l_mod_type="fs"
  l_mod_path="$(readlink -f /lib/modules/**/kernel/$l_mod_type | sort -u)"
  f_module_fix()
  {
    l_dl="y" a_showconfig=()
    while IFS= read -r l_showconfig; do
      a_showconfig+=("$l_showconfig")
      done < <(modprobe --showconfig | grep -P --
'\b(install|blacklist)\h+'${l_mod_chk_name//-/}_'\b')
      if lsmod | grep "$l_mod_chk_name" &> /dev/null; then
        a_output2+=(" - unloading kernel module: \"$l_mod_name\"")
        modprobe -r "$l_mod_chk_name" 2>/dev/null; rmmod "$l_mod_name"
2>/dev/null
      fi
      if ! grep -Pq -- '\binstall\h+'${l_mod_chk_name//-/}_
/_'\b'\h+(\usr)?\bin\/(true|false)\b' <<< "${a_showconfig[*]}"; then
        a_output2+=(" - setting kernel module: \"$l_mod_name\" to
\"$(readlink -f /bin/false)\"")
        printf '%s\n' "install $l_mod_chk_name $(readlink -f /bin/false)" >>
/etc/modprobe.d/"$l_mod_name".conf
      fi
      if ! grep -Pq -- '\bblacklist\h+'${l_mod_chk_name//-/}_'\b' <<<
"${a_showconfig[*]}"; then
        a_output2+=(" - denylisting kernel module: \"$l_mod_name\"")
        printf '%s\n' "blacklist $l_mod_chk_name" >>
/etc/modprobe.d/"$l_mod_name".conf
      fi
    }
    for l_mod_base_directory in $l_mod_path; do # Check if the module exists
on the system
      if [ -d "$l_mod_base_directory/${l_mod_name//-/}/" ] && [ -n "$(ls -A
"$l_mod_base_directory/${l_mod_name//-/}/")" ]; then
        a_output3+=(" - \"$l_mod_base_directory\"")
        l_mod_chk_name="$l_mod_name"
        [[ "$l_mod_name" =~ overlay ]] && l_mod_chk_name="$l_mod_name::-2"
        [ "$l_dl" != "y" ] && f_module_fix
      else
        printf '%s\n' " - kernel module: \"$l_mod_name\" doesn't exist in
\"$l_mod_base_directory\""
      fi
    done
    [ "${#a_output3[@]}" -gt 0 ] && printf '%s\n' "" " -- INFO --" " - module:
\"$l_mod_name\" exists in:" "${a_output3[@]}"
    [ "${#a_output2[@]}" -gt 0 ] && printf '%s\n' "" "${a_output2[@]}" ||
printf '%s\n' "" " - No changes needed"
    printf '%s\n' "" " - remediation of kernel module: \"$l_mod_name\"
complete" ""
  }
}
```

References:

1. NIST SP 800-53 Rev. 5: CM-7

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<u>4.8 Uninstall or Disable Unnecessary Services on Enterprise Assets and Software</u> Uninstall or disable unnecessary services on enterprise assets and software, such as an unused file sharing service, web application module, or service function.			
v7	<u>9.2 Ensure Only Approved Ports, Protocols and Services Are Running</u> Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1005, T1005.000	TA0005	M1050

1.1.1.4 Ensure hfsplus kernel module is not available (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The **hfsplus** filesystem type is a hierarchical filesystem designed to replace **hfs** that allows you to mount Mac OS filesystems.

Rationale:

Removing support for unneeded filesystem types reduces the local attack surface of the system. If this filesystem type is not needed, disable it.

Audit:

Run the following script to verify:

- **IF** - the **hfsplus** kernel module is available in ANY installed kernel, verify:

- An entry including **/bin/true** or **/bin/false** exists in a file within the **/etc/modprobe.d/** directory
- The module is deny listed in a file within the **/etc/modprobe.d/** directory
- The module is not loaded in the running kernel

- **IF** - the **hfsplus** kernel module is not available on the system, or pre-compiled into the kernel, no additional configuration is necessary

```

#!/usr/bin/env bash

{
    a_output=() a_output2=() a_output3=() l_dl="" l_mod_name="hfsplus"
    l_mod_type="fs"
    l_mod_path="$(readlink -f /lib/modules/**/kernel/$l_mod_type | sort -u)"
    f_module_chk()
    {
        l_dl="y" a_showconfig=()
        while IFS= read -r l_showconfig; do
            a_showconfig+=("$l_showconfig")
            done < <(modprobe --showconfig | grep -P --
'\b(install|blacklist)\h+'${l_mod_chk_name//-/}_'\b')
            if ! lsmod | grep "$l_mod_chk_name" &> /dev/null; then
                a_output+=(" - kernel module: \"$l_mod_name\" is not loaded")
            else
                a_output2+=(" - kernel module: \"$l_mod_name\" is loaded")
            fi
            if grep -Pq -- '\bbinstall\h+'${l_mod_chk_name//-/}_
/_}'\h+(\usr)?\bin\/(true|false)\b' <<< "${a_showconfig[*]}"; then
                a_output+=(" - kernel module: \"$l_mod_name\" is not loadable")
            else
                a_output2+=(" - kernel module: \"$l_mod_name\" is loadable")
            fi
            if grep -Pq -- '\bbblacklist\h+'${l_mod_chk_name//-/}_'\b' <<<
"${a_showconfig[*]}"; then
                a_output+=(" - kernel module: \"$l_mod_name\" is deny listed")
            else
                a_output2+=(" - kernel module: \"$l_mod_name\" is not deny listed")
            fi
        }
        for l_mod_base_directory in $l_mod_path; do
            if [ -d "$l_mod_base_directory/${l_mod_name//-/}/" ] && [ -n "$(ls -A
"$l_mod_base_directory/${l_mod_name//-/}/")" ]; then
                a_output3+=(" - \"$l_mod_base_directory\"")
                l_mod_chk_name="$l_mod_name"
                [[ "$l_mod_name" =~ overlay ]] && l_mod_chk_name="$l_mod_name:-2"
                [ "$l_dl" != "y" ] && f_module_chk
            else
                a_output+=(" - kernel module: \"$l_mod_name\" doesn't exist in
\"$l_mod_base_directory\"")
            fi
        done
        [ "${#a_output3[@]}" -gt 0 ] && printf '%s\n' "" " -- INFO --" " - module:
\"$l_mod_name\" exists in:" "${a_output3[@]}"
        if [ "${#a_output2[@]}" -le 0 ]; then
            printf '%s\n' "" "- Audit Result:" " ** PASS **" "${a_output[@]}"
        else
            printf '%s\n' "" "- Audit Result:" " ** FAIL **" " - Reason(s) for
audit failure:" "${a_output2[@]}"
            [ "${#a_output[@]}" -gt 0 ] && printf '%s\n' "- Correctly set:"
"${a_output[@]}"
        fi
    }
}

```

Remediation:

Run the following script to unload and disable the `hfsplus` module:

- **IF** - the `hfsplus` kernel module is available in ANY installed kernel:

- Create a file ending in `.conf` with `install hfsplus /bin/false` in the `/etc/modprobe.d/` directory
- Create a file ending in `.conf` with `blacklist hfsplus` in the `/etc/modprobe.d/` directory
- Run `modprobe -r hfsplus 2>/dev/null; rmmod hfsplus 2>/dev/null` to remove `hfsplus` from the kernel

- **IF** - the `hfsplus` kernel module is not available on the system, or pre-compiled into the kernel, no remediation is necessary





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

{
    a_output2=() a_output3=() l_dl="" l_mod_name="hfsplus" l_mod_type="fs"
    l_mod_path="$(readlink -f /lib/modules/**/kernel/$l_mod_type | sort -u)"
    f_module_fix()
    {
        l_dl="y" a_showconfig=()
        while IFS= read -r l_showconfig; do
            a_showconfig+=("$l_showconfig")
            done < <(modprobe --showconfig | grep -P --
'\b(install|blacklist)\h+'${l_mod_chk_name//-/}_'\b')
            if lsmod | grep "$l_mod_chk_name" &> /dev/null; then
                a_output2+=(" - unloading kernel module: \"$l_mod_name\"")
                modprobe -r "$l_mod_chk_name" 2>/dev/null; rmmod "$l_mod_name"
2>/dev/null
            fi
            if ! grep -Pq -- '\binstall\h+'${l_mod_chk_name//-/}_
/_'\b'\h+(\usr)?\bin\/(true|false)\b' <<< "${a_showconfig[*]}"; then
                a_output2+=(" - setting kernel module: \"$l_mod_name\" to
\"$(readlink -f /bin/false)\"")
                printf '%s\n' "install $l_mod_chk_name $(readlink -f /bin/false)" >>
/etc/modprobe.d/"$l_mod_name".conf
            fi
            if ! grep -Pq -- '\bblacklist\h+'${l_mod_chk_name//-/}_'\b' <<<
"${a_showconfig[*]}"; then
                a_output2+=(" - denylisting kernel module: \"$l_mod_name\"")
                printf '%s\n' "blacklist $l_mod_chk_name" >>
/etc/modprobe.d/"$l_mod_name".conf
            fi
        }
        for l_mod_base_directory in $l_mod_path; do # Check if the module exists
on the system
            if [ -d "$l_mod_base_directory/${l_mod_name//-/}/" ] && [ -n "$(ls -A
"$l_mod_base_directory/${l_mod_name//-/}/")" ]; then
                a_output3+=(" - \"$l_mod_base_directory\"")
                l_mod_chk_name="$l_mod_name"
                [[ "$l_mod_name" =~ overlay ]] && l_mod_chk_name="${l_mod_name::-2}"
                [ "$l_dl" != "y" ] && f_module_fix
            else
                printf '%s\n' " - kernel module: \"$l_mod_name\" doesn't exist in
\"$l_mod_base_directory\""
            fi
        done
        [ "${#a_output3[@]}" -gt 0 ] && printf '%s\n' "" " -- INFO --" " - module:
\"$l_mod_name\" exists in:" "${a_output3[@]}"
        [ "${#a_output2[@]}" -gt 0 ] && printf '%s\n' "" "${a_output2[@]}" ||
printf '%s\n' "" " - No changes needed"
        printf '%s\n' "" " - remediation of kernel module: \"$l_mod_name\"
complete" ""
    }
}
```

References:

1. NIST SP 800-53 Rev. 5: CM-7

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<u>4.8 Uninstall or Disable Unnecessary Services on Enterprise Assets and Software</u> Uninstall or disable unnecessary services on enterprise assets and software, such as an unused file sharing service, web application module, or service function.			
v7	<u>9.2 Ensure Only Approved Ports, Protocols and Services Are Running</u> Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1005, T1005.000	TA0005	M1050

1.1.1.5 Ensure jffs2 kernel module is not available (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The **jffs2** (journaling flash filesystem 2) filesystem type is a log-structured filesystem used in flash memory devices.

Rationale:

Removing support for unneeded filesystem types reduces the local attack surface of the system. If this filesystem type is not needed, disable it.

Audit:

Run the following script to verify:

- **IF** - the **jffs2** kernel module is available in ANY installed kernel, verify:

- An entry including **/bin/true** or **/bin/false** exists in a file within the **/etc/modprobe.d/** directory
- The module is deny listed in a file within the **/etc/modprobe.d/** directory
- The module is not loaded in the running kernel

- **IF** - the **jffs2** kernel module is not available on the system, or pre-compiled into the kernel, no additional configuration is necessary


```

#!/usr/bin/env bash

{
    a_output=() a_output2=() a_output3=() l_dl="" l_mod_name="jffs2"
    l_mod_type="fs"
    l_mod_path="$(readlink -f /lib/modules/**/kernel/$l_mod_type | sort -u)"
    f_module_chk()
    {
        l_dl="y" a_showconfig=()
        while IFS= read -r l_showconfig; do
            a_showconfig+=("$l_showconfig")
            done < <(modprobe --showconfig | grep -P --
'\b(install|blacklist)\h+'${l_mod_chk_name//-/}_'\b')
            if ! lsmmod | grep "$l_mod_chk_name" &> /dev/null; then
                a_output+=(" - kernel module: \"$l_mod_name\" is not loaded")
            else
                a_output2+=(" - kernel module: \"$l_mod_name\" is loaded")
            fi
            if grep -Pq -- '\bbinstall\h+'${l_mod_chk_name//-/}_
/_}'\h+(\usr)?\bin\(/(true|false)\b' <<< "${a_showconfig[*]}"; then
                a_output+=(" - kernel module: \"$l_mod_name\" is not loadable")
            else
                a_output2+=(" - kernel module: \"$l_mod_name\" is loadable")
            fi
            if grep -Pq -- '\bbblacklist\h+'${l_mod_chk_name//-/}_'\b' <<<
"${a_showconfig[*]}"; then
                a_output+=(" - kernel module: \"$l_mod_name\" is deny listed")
            else
                a_output2+=(" - kernel module: \"$l_mod_name\" is not deny listed")
            fi
        }
        for l_mod_base_directory in $l_mod_path; do
            if [ -d "$l_mod_base_directory/${l_mod_name//-/}/" ] && [ -n "$(ls -A
"$l_mod_base_directory/${l_mod_name//-/}/")" ]; then
                a_output3+=(" - \"$l_mod_base_directory\"")
                l_mod_chk_name="$l_mod_name"
                [[ "$l_mod_name" =~ overlay ]] && l_mod_chk_name="$l_mod_name:-2)"
                [ "$l_dl" != "y" ] && f_module_chk
            else
                a_output+=(" - kernel module: \"$l_mod_name\" doesn't exist in
\"$l_mod_base_directory\"")
            fi
        done
        [ "${#a_output3[@]}" -gt 0 ] && printf '%s\n' "" " -- INFO --" " - module:
\"$l_mod_name\" exists in:" "${a_output3[@]}"
        if [ "${#a_output2[@]}" -le 0 ]; then
            printf '%s\n' "" "- Audit Result:" " ** PASS **" "${a_output[@]}"
        else
            printf '%s\n' "" "- Audit Result:" " ** FAIL **" " - Reason(s) for
audit failure:" "${a_output2[@]}"
            [ "${#a_output[@]}" -gt 0 ] && printf '%s\n' "- Correctly set:"
"${a_output[@]}"
        fi
    }
}

```

Remediation:

Run the following script to unload and disable the `jffs2` module:

- **IF** - the `jffs2` kernel module is available in ANY installed kernel:

- Create a file ending in `.conf` with `install jffs2 /bin/false` in the `/etc/modprobe.d/` directory
- Create a file ending in `.conf` with `blacklist jffs2` in the `/etc/modprobe.d/` directory
- Run `modprobe -r jffs2 2>/dev/null; rmmod jffs2 2>/dev/null` to remove `jffs2` from the kernel

- **IF** - the `jffs2` kernel module is not available on the system, or pre-compiled into the kernel, no remediation is necessary





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

{
  a_output2=() a_output3=() l_dl="" l_mod_name="jffs2" l_mod_type="fs"
  l_mod_path="$(readlink -f /lib/modules/**/kernel/$l_mod_type | sort -u)"
  f_module_fix()
  {
    l_dl="y" a_showconfig=()
    while IFS= read -r l_showconfig; do
      a_showconfig+=("$l_showconfig")
      done < <(modprobe --showconfig | grep -P --
'\b(install|blacklist)\h+'"$l_mod_chk_name//-/_"'"'\b')
      if lsmod | grep "$l_mod_chk_name" &> /dev/null; then
        a_output2+=(" - unloading kernel module: \"$l_mod_name\"")
        modprobe -r "$l_mod_chk_name" 2>/dev/null; rmmod "$l_mod_name"
2>/dev/null
      fi
      if ! grep -Pq -- '\binstall\h+'"$l_mod_chk_name//-/_"'"'\b' <<<
/"$a_showconfig[*]"; then
        a_output2+=(" - setting kernel module: \"$l_mod_name\" to
\"$(readlink -f /bin/false)\"")
        printf '%s\n' "install $l_mod_chk_name $(readlink -f /bin/false)" >>
/etc/modprobe.d/"$l_mod_name".conf
      fi
      if ! grep -Pq -- '\bblacklist\h+'"$l_mod_chk_name//-/_"'"'\b' <<<
/"$a_showconfig[*]"; then
        a_output2+=(" - denylisting kernel module: \"$l_mod_name\"")
        printf '%s\n' "blacklist $l_mod_chk_name" >>
/etc/modprobe.d/"$l_mod_name".conf
      fi
    }
    for l_mod_base_directory in $l_mod_path; do # Check if the module exists
on the system
      if [ -d "$l_mod_base_directory/$l_mod_name/-/\/" ] && [ -n "$(ls -A
"$l_mod_base_directory/$l_mod_name/-/\/" )" ]; then
        a_output3+=(" - \"$l_mod_base_directory\"")
        l_mod_chk_name="$l_mod_name"
        [[ "$l_mod_name" =~ overlay ]] && l_mod_chk_name="$l_mod_name::-2"
        [ "$l_dl" != "y" ] && f_module_fix
      else
        printf '%s\n' " - kernel module: \"$l_mod_name\" doesn't exist in
\"$l_mod_base_directory\""
      fi
    done
    [ "${#a_output3[@]}" -gt 0 ] && printf '%s\n' "" " -- INFO --" " - module:
\"$l_mod_name\" exists in:" "${a_output3[@]}"
    [ "${#a_output2[@]}" -gt 0 ] && printf '%s\n' "" "${a_output2[@]}" ||
printf '%s\n' "" " - No changes needed"
    printf '%s\n' "" " - remediation of kernel module: \"$l_mod_name\"
complete" ""
  }
}
```

References:

1. NIST SP 800-53 Rev. 5: CM-7

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<u>4.8 Uninstall or Disable Unnecessary Services on Enterprise Assets and Software</u> Uninstall or disable unnecessary services on enterprise assets and software, such as an unused file sharing service, web application module, or service function.			
v7	<u>9.2 Ensure Only Approved Ports, Protocols and Services Are Running</u> Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1005, T1005.000	TA0005	M1050

1.1.1.6 Ensure overlayfs kernel module is not available (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

overlayfs is a Linux filesystem that layers multiple filesystems to create a single unified view which allows a user to "merge" several mount points into a unified filesystem.

Rationale:

The **overlayfs** has known CVE's: CVE-2023-32629, CVE-2023-2640, CVE-2023-0386. Disabling the **overlayfs** reduces the local attack surface by removing support for unnecessary filesystem types and mitigates potential risks associated with unauthorized execution of setuid files, enhancing the overall system security.

Impact:

WARNING: If Container applications such as Docker, Kubernetes, Podman, Linux Containers (LXC), etc. are in use proceed with caution and consider the impact on containerized workloads, as disabling the **overlayfs may severely disrupt containerization.**

Audit:

Run the following script to verify:

- **IF** - the **overlayfs** kernel module is available in ANY installed kernel, verify:

- An entry including **/bin/true** or **/bin/false** exists in a file within the **/etc/modprobe.d/** directory
- The module is deny listed in a file within the **/etc/modprobe.d/** directory
- The module is not loaded in the running kernel

- **IF** - the **overlayfs** kernel module is not available on the system, or pre-compiled into the kernel, no additional configuration is necessary

```

#!/usr/bin/env bash

{
    a_output=() a_output2=() a_output3=() l_dl="" l_mod_name="overlayfs"
    l_mod_type="fs"
    l_mod_path="$(readlink -f /lib/modules/**/kernel/$l_mod_type | sort -u)"
    f_module_chk()
    {
        l_dl="y" a_showconfig=()
        while IFS= read -r l_showconfig; do
            a_showconfig+=("$l_showconfig")
            done < <(modprobe --showconfig | grep -P --
'\b(install|blacklist)\h+'${l_mod_chk_name//-/}_'\b')
            if ! lsmod | grep "$l_mod_chk_name" &> /dev/null; then
                a_output+=(" - kernel module: \"$l_mod_name\" is not loaded")
            else
                a_output2+=(" - kernel module: \"$l_mod_name\" is loaded")
            fi
            if grep -Pq -- '\bbinstall\h+'${l_mod_chk_name//-/}_
/_}'\h+(\usr)?\bin\/(true|false)\b' <<< "${a_showconfig[*]}"; then
                a_output+=(" - kernel module: \"$l_mod_name\" is not loadable")
            else
                a_output2+=(" - kernel module: \"$l_mod_name\" is loadable")
            fi
            if grep -Pq -- '\bbblacklist\h+'${l_mod_chk_name//-/}_'\b' <<<
"${a_showconfig[*]}"; then
                a_output+=(" - kernel module: \"$l_mod_name\" is deny listed")
            else
                a_output2+=(" - kernel module: \"$l_mod_name\" is not deny listed")
            fi
        }
        for l_mod_base_directory in $l_mod_path; do
            if [ -d "$l_mod_base_directory/${l_mod_name//-/}/" ] && [ -n "$(ls -A
"$l_mod_base_directory/${l_mod_name//-/}/")" ]; then
                a_output3+=(" - \"$l_mod_base_directory\"")
                l_mod_chk_name="$l_mod_name"
                [[ "$l_mod_name" =~ overlay ]] && l_mod_chk_name="$l_mod_name:-2"
                [ "$l_dl" != "y" ] && f_module_chk
            else
                a_output+=(" - kernel module: \"$l_mod_name\" doesn't exist in
\"$l_mod_base_directory\"")
            fi
        done
        [ "${#a_output3[@]}" -gt 0 ] && printf '%s\n' "" " -- INFO --" " - module:
\"$l_mod_name\" exists in:" "${a_output3[@]}"
        if [ "${#a_output2[@]}" -le 0 ]; then
            printf '%s\n' "" "- Audit Result:" " ** PASS **" "${a_output[@]}"
        else
            printf '%s\n' "" "- Audit Result:" " ** FAIL **" " - Reason(s) for
audit failure:" "${a_output2[@]}"
            [ "${#a_output[@]}" -gt 0 ] && printf '%s\n' "- Correctly set:"
"${a_output[@]}"
        fi
    }
}

```

Remediation:

Run the following script to unload and disable the `overlayfs` module:

- **IF** - the `overlayfs` kernel module is available in ANY installed kernel:

- Create a file ending in `.conf` with `install overlayfs /bin/false` in the `/etc/modprobe.d/` directory
- Create a file ending in `.conf` with `blacklist overlayfs` in the `/etc/modprobe.d/` directory
- Run `modprobe -r overlayfs 2>/dev/null; rmmod overlayfs 2>/dev/null` to remove `overlayfs` from the kernel

- **IF** - the `overlayfs` kernel module is not available on the system, or pre-compiled into the kernel, no remediation is necessary

```

#!/usr/bin/env bash

{
    a_output2=() a_output3=() l_dl="" l_mod_name="overlayfs" l_mod_type="fs"
    l_mod_path="$(readlink -f /lib/modules/**/kernel/$l_mod_type | sort -u)"
    f_module_fix()
    {
        l_dl="y" a_showconfig=()
        while IFS= read -r l_showconfig; do
            a_showconfig+=("$l_showconfig")
            done < <(modprobe --showconfig | grep -P --
'\b(install|blacklist)\h+'${l_mod_chk_name//-/}_'\b')
            if lsmod | grep "$l_mod_chk_name" &> /dev/null; then
                a_output2+=(" - unloading kernel module: \"$l_mod_name\"")
                modprobe -r "$l_mod_chk_name" 2>/dev/null; rmmod "$l_mod_name"
2>/dev/null
            fi
            if ! grep -Pq -- '\binstall\h+'${l_mod_chk_name//-/}_
/_'\h+(\usr)?\bin\/(true|false)\b' <<< "${a_showconfig[*]}"; then
                a_output2+=(" - setting kernel module: \"$l_mod_name\" to
\"$(readlink -f /bin/false)\"")
                printf '%s\n' "install $l_mod_chk_name $(readlink -f /bin/false)" >>
/etc/modprobe.d/"$l_mod_name".conf
            fi
            if ! grep -Pq -- '\bblacklist\h+'${l_mod_chk_name//-/}_'\b' <<<
"${a_showconfig[*]}"; then
                a_output2+=(" - denylisting kernel module: \"$l_mod_name\"")
                printf '%s\n' "blacklist $l_mod_chk_name" >>
/etc/modprobe.d/"$l_mod_name".conf
            fi
        }
        for l_mod_base_directory in $l_mod_path; do # Check if the module exists
on the system
            if [ -d "$l_mod_base_directory/${l_mod_name//-/}/" ] && [ -n "$(ls -A
"$l_mod_base_directory/${l_mod_name//-/}/")" ]; then
                a_output3+=(" - \"$l_mod_base_directory\"")
                l_mod_chk_name="$l_mod_name"
                [[ "$l_mod_name" =~ overlay ]] && l_mod_chk_name="${l_mod_name::-2}"
                [ "$l_dl" != "y" ] && f_module_fix
            else
                printf '%s\n' " - kernel module: \"$l_mod_name\" doesn't exist in
\"$l_mod_base_directory\""
            fi
        done
        [ "${#a_output3[@]}" -gt 0 ] && printf '%s\n' "" " -- INFO --" " - module:
\"$l_mod_name\" exists in:" "${a_output3[@]}"
        [ "${#a_output2[@]}" -gt 0 ] && printf '%s\n' "" "${a_output2[@]}" ||
printf '%s\n' "" " - No changes needed"
        printf '%s\n' "" " - remediation of kernel module: \"$l_mod_name\"
complete" ""
    }
}

```


References:

1. NIST SP 800-53 Rev. 5: CM-7
2. <https://docs.kernel.org/filesystems/overlayfs.html>
3. https://wiki.archlinux.org/title/Overlay_filesystem
4. <https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=overlayfs>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	4.8 <u>Uninstall or Disable Unnecessary Services on Enterprise Assets and Software</u> Uninstall or disable unnecessary services on enterprise assets and software, such as an unused file sharing service, web application module, or service function.		●	●
v7	9.2 <u>Ensure Only Approved Ports, Protocols and Services Are Running</u> Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system.		●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1005, T1005.000	TA0005	M1050

1.1.1.7 Ensure squashfs kernel module is not available (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

The **squashfs** filesystem type is a compressed read-only Linux filesystem embedded in small footprint systems. A **squashfs** image can be used without having to first decompress the image.

Rationale:

Removing support for unneeded filesystem types reduces the local attack surface of the system. If this filesystem type is not needed, disable it.

Impact:

As Snap packages utilize **squashfs** as a compressed filesystem, disabling **squashfs** will cause Snap packages to fail.

Snap application packages of software are self-contained and work across a range of Linux distributions. This is unlike traditional Linux package management approaches, like APT or RPM, which require specifically adapted packages per Linux distribution on an application update and delay therefore application deployment from developers to their software's end-user. Snaps themselves have no dependency on any external store ("App store"), can be obtained from any source and can be therefore used for upstream software deployment.

Audit:

Run the following script to verify:

- **IF** - the **squashfs** kernel module is available in ANY installed kernel, verify:

- An entry including **/bin/true** or **/bin/false** exists in a file within the **/etc/modprobe.d/** directory
- The module is deny listed in a file within the **/etc/modprobe.d/** directory
- The module is not loaded in the running kernel

- **IF** - the **squashfs** kernel module is not available on the system, or pre-compiled into the kernel, no additional configuration is necessary

```

#!/usr/bin/env bash

{
    a_output=() a_output2=() a_output3=() l_dl="" l_mod_name="squashfs"
    l_mod_type="fs"
    l_mod_path="$(readlink -f /lib/modules/**/kernel/$l_mod_type | sort -u)"
    f_module_chk()
    {
        l_dl="y" a_showconfig=()
        while IFS= read -r l_showconfig; do
            a_showconfig+=("$l_showconfig")
            done < <(modprobe --showconfig | grep -P --
'\b(install|blacklist)\h+'${l_mod_chk_name//-/}_'\b')
            if ! lsmod | grep "$l_mod_chk_name" &> /dev/null; then
                a_output+=(" - kernel module: \"$l_mod_name\" is not loaded")
            else
                a_output2+=(" - kernel module: \"$l_mod_name\" is loaded")
            fi
            if grep -Pq -- '\bbinstall\h+'${l_mod_chk_name//-/}_
/_}'\h+(\usr)?\bin\/(true|false)\b' <<< "${a_showconfig[*]}"; then
                a_output+=(" - kernel module: \"$l_mod_name\" is not loadable")
            else
                a_output2+=(" - kernel module: \"$l_mod_name\" is loadable")
            fi
            if grep -Pq -- '\bbblacklist\h+'${l_mod_chk_name//-/}_'\b' <<<
"${a_showconfig[*]}"; then
                a_output+=(" - kernel module: \"$l_mod_name\" is deny listed")
            else
                a_output2+=(" - kernel module: \"$l_mod_name\" is not deny listed")
            fi
        }
        for l_mod_base_directory in $l_mod_path; do
            if [ -d "$l_mod_base_directory/${l_mod_name//-/}/" ] && [ -n "$(ls -A
"$l_mod_base_directory/${l_mod_name//-/}/")" ]; then
                a_output3+=(" - \"$l_mod_base_directory\"")
                l_mod_chk_name="$l_mod_name"
                [[ "$l_mod_name" =~ overlay ]] && l_mod_chk_name="$l_mod_name:-2"
                [ "$l_dl" != "y" ] && f_module_chk
            else
                a_output+=(" - kernel module: \"$l_mod_name\" doesn't exist in
\"$l_mod_base_directory\"")
            fi
        done
        [ "${#a_output3[@]}" -gt 0 ] && printf '%s\n' "" " -- INFO --" " - module:
\"$l_mod_name\" exists in:" "${a_output3[@]}"
        if [ "${#a_output2[@]}" -le 0 ]; then
            printf '%s\n' "" "- Audit Result:" " ** PASS **" "${a_output[@]}"
        else
            printf '%s\n' "" "- Audit Result:" " ** FAIL **" " - Reason(s) for
audit failure:" "${a_output2[@]}"
            [ "${#a_output[@]}" -gt 0 ] && printf '%s\n' "- Correctly set:"
"${a_output[@]}"
        fi
    }
}

```

Note: On operating systems where `squashfs` is pre-build into the kernel:

- This is considered an acceptable "passing" state
- The kernel **should not** be re-compiled to remove `squashfs`
- This audit will return a passing state with "module: "squashfs" doesn't exist in ..."

Remediation:

Run the following script to unload and disable the `udf` module:

- **IF** - the `squashfs` kernel module is available in ANY installed kernel:

- Create a file ending in `.conf` with `install squashfs /bin/false` in the `/etc/modprobe.d/` directory
- Create a file ending in `.conf` with `blacklist squashfs` in the `/etc/modprobe.d/` directory
- Run `modprobe -r squashfs 2>/dev/null; rmmod squashfs 2>/dev/null` to remove `squashfs` from the kernel

- **IF** - the `squashfs` kernel module is not available on the system, or pre-compiled into the kernel, no remediation is necessary





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

{
  a_output2=() a_output3=() l_dl="" l_mod_name="squashfs" l_mod_type="fs"
  l_mod_path="$(readlink -f /lib/modules/**/kernel/$l_mod_type | sort -u)"
  f_module_fix()
  {
    l_dl="y" a_showconfig=()
    while IFS= read -r l_showconfig; do
      a_showconfig+=("$l_showconfig")
      done < <(modprobe --showconfig | grep -P --
'\b(install|blacklist)\h+'${l_mod_chk_name//-/}_'\b')
      if lsmod | grep "$l_mod_chk_name" &> /dev/null; then
        a_output2+=(" - unloading kernel module: \"$l_mod_name\"")
        modprobe -r "$l_mod_chk_name" 2>/dev/null; rmmod "$l_mod_name"
2>/dev/null
      fi
      if ! grep -Pq -- '\binstall\h+'${l_mod_chk_name//-/}_
/_'\b'\h+(\usr)?\bin\/(true|false)\b' <<< "${a_showconfig[*]}"; then
        a_output2+=(" - setting kernel module: \"$l_mod_name\" to
\"$(readlink -f /bin/false)\"")
        printf '%s\n' "install $l_mod_chk_name $(readlink -f /bin/false)" >>
/etc/modprobe.d/"$l_mod_name".conf
      fi
      if ! grep -Pq -- '\bblacklist\h+'${l_mod_chk_name//-/}_'\b' <<<
"${a_showconfig[*]}"; then
        a_output2+=(" - denylisting kernel module: \"$l_mod_name\"")
        printf '%s\n' "blacklist $l_mod_chk_name" >>
/etc/modprobe.d/"$l_mod_name".conf
      fi
    }
    for l_mod_base_directory in $l_mod_path; do # Check if the module exists
on the system
      if [ -d "$l_mod_base_directory/${l_mod_name//-/}/" ] && [ -n "$(ls -A
"$l_mod_base_directory/${l_mod_name//-/}/")" ]; then
        a_output3+=(" - \"$l_mod_base_directory\"")
        l_mod_chk_name="$l_mod_name"
        [[ "$l_mod_name" =~ overlay ]] && l_mod_chk_name="${l_mod_name::-2}"
        [ "$l_dl" != "y" ] && f_module_fix
      else
        printf '%s\n' " - kernel module: \"$l_mod_name\" doesn't exist in
\"$l_mod_base_directory\""
      fi
    done
    [ "${#a_output3[@]}" -gt 0 ] && printf '%s\n' "" " -- INFO --" " - module:
\"$l_mod_name\" exists in:" "${a_output3[@]}"
    [ "${#a_output2[@]}" -gt 0 ] && printf '%s\n' "" "${a_output2[@]}" ||
printf '%s\n' "" " - No changes needed"
    printf '%s\n' "" " - remediation of kernel module: \"$l_mod_name\"
complete" ""
  }
}
```

References:

1. NIST SP 800-53 Rev. 5: CM-7

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<u>4.8 Uninstall or Disable Unnecessary Services on Enterprise Assets and Software</u> Uninstall or disable unnecessary services on enterprise assets and software, such as an unused file sharing service, web application module, or service function.			
v7	<u>9.2 Ensure Only Approved Ports, Protocols and Services Are Running</u> Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1005, T1005.000	TA0005	M1050

1.1.1.8 Ensure udf kernel module is not available (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

The **udf** filesystem type is the universal disk format used to implement ISO/IEC 13346 and ECMA-167 specifications. This is an open vendor filesystem type for data storage on a broad range of media. This filesystem type is necessary to support writing DVDs and newer optical disc formats.

Rationale:

Removing support for unneeded filesystem types reduces the local attack surface of the system. If this filesystem type is not needed, disable it.

Impact:

Microsoft Azure requires the usage of **udf**.

udf **should not** be disabled on systems run on Microsoft Azure.

Audit:

Run the following script to verify:

- **IF** - the **udf** kernel module is available in ANY installed kernel, verify:

- An entry including **/bin/true** or **/bin/false** exists in a file within the **/etc/modprobe.d/** directory
- The module is deny listed in a file within the **/etc/modprobe.d/** directory
- The module is not loaded in the running kernel

- **IF** - the **udf** kernel module is not available on the system, or pre-compiled into the kernel, no additional configuration is necessary

```

#!/usr/bin/env bash

{
    a_output=() a_output2=() a_output3=() l_dl="" l_mod_name="udf"
    l_mod_type="fs"
    l_mod_path="$(readlink -f /lib/modules/**/kernel/$l_mod_type | sort -u)"
    f_module_chk()
    {
        l_dl="y" a_showconfig=()
        while IFS= read -r l_showconfig; do
            a_showconfig+=("$l_showconfig")
            done < <(modprobe --showconfig | grep -P --
'\b(install|blacklist)\h+'${l_mod_chk_name//-/}_'\b')
            if ! lsmod | grep "$l_mod_chk_name" &> /dev/null; then
                a_output+=(" - kernel module: \"$l_mod_name\" is not loaded")
            else
                a_output2+=(" - kernel module: \"$l_mod_name\" is loaded")
            fi
            if grep -Pq -- '\bbinstall\h+'${l_mod_chk_name//-/}_'\b' <<< "${a_showconfig[*]}"; then
                a_output+=(" - kernel module: \"$l_mod_name\" is not loadable")
            else
                a_output2+=(" - kernel module: \"$l_mod_name\" is loadable")
            fi
            if grep -Pq -- '\bbblacklist\h+'${l_mod_chk_name//-/}_'\b' <<<
"${a_showconfig[*]}"; then
                a_output+=(" - kernel module: \"$l_mod_name\" is deny listed")
            else
                a_output2+=(" - kernel module: \"$l_mod_name\" is not deny listed")
            fi
        }
        for l_mod_base_directory in $l_mod_path; do
            if [ -d "$l_mod_base_directory/${l_mod_name//-/}/" ] && [ -n "$(ls -A
"$l_mod_base_directory/${l_mod_name//-/}/")" ]; then
                a_output3+=(" - \"$l_mod_base_directory\"")
                l_mod_chk_name="$l_mod_name"
                [[ "$l_mod_name" =~ overlay ]] && l_mod_chk_name="$l_mod_name:-2"
                [ "$l_dl" != "y" ] && f_module_chk
            else
                a_output+=(" - kernel module: \"$l_mod_name\" doesn't exist in
\"$l_mod_base_directory\"")
            fi
        done
        [ "${#a_output3[@]}" -gt 0 ] && printf '%s\n' "" " -- INFO --" " - module:
\"$l_mod_name\" exists in:" "${a_output3[@]}"
        if [ "${#a_output2[@]}" -le 0 ]; then
            printf '%s\n' "" "- Audit Result:" " ** PASS **" "${a_output[@]}"
        else
            printf '%s\n' "" "- Audit Result:" " ** FAIL **" " - Reason(s) for
audit failure:" "${a_output2[@]}"
            [ "${#a_output[@]}" -gt 0 ] && printf '%s\n' "- Correctly set:"
"${a_output[@]}"
        fi
    }
}

```


Remediation:

Run the following script to unload and disable the `udf` module:

- **IF** - the `udf` kernel module is available in ANY installed kernel:

- Create a file ending in `.conf` with `install udf /bin/false` in the `/etc/modprobe.d/` directory
- Create a file ending in `.conf` with `blacklist udf` in the `/etc/modprobe.d/` directory
- Run `modprobe -r udf 2>/dev/null; rmmod udf 2>/dev/null` to remove `udf` from the kernel

- **IF** - the `udf` kernel module is not available on the system, or pre-compiled into the kernel, no remediation is necessary





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

{
  a_output2=() a_output3=() l_dl="" l_mod_name="udf" l_mod_type="fs"
  l_mod_path="$(readlink -f /lib/modules/**/kernel/$l_mod_type | sort -u)"
  f_module_fix()
  {
    l_dl="y" a_showconfig=()
    while IFS= read -r l_showconfig; do
      a_showconfig+=("$l_showconfig")
      done < <(modprobe --showconfig | grep -P --
'\b(install|blacklist)\h+'${l_mod_chk_name//-/}_'\b')
      if lsmod | grep "$l_mod_chk_name" &> /dev/null; then
        a_output2+=(" - unloading kernel module: \"$l_mod_name\"")
        modprobe -r "$l_mod_chk_name" 2>/dev/null; rmmod "$l_mod_name"
2>/dev/null
      fi
      if ! grep -Pq -- '\binstall\h+'${l_mod_chk_name//-/}_
/_'\b'\h+(\usr)?\bin\/(true|false)\b' <<< "${a_showconfig[*]}"; then
        a_output2+=(" - setting kernel module: \"$l_mod_name\" to
\"$(readlink -f /bin/false)\"")
        printf '%s\n' "install $l_mod_chk_name $(readlink -f /bin/false)" >>
/etc/modprobe.d/"$l_mod_name".conf
      fi
      if ! grep -Pq -- '\bblacklist\h+'${l_mod_chk_name//-/}_'\b' <<<
"${a_showconfig[*]}"; then
        a_output2+=(" - denylisting kernel module: \"$l_mod_name\"")
        printf '%s\n' "blacklist $l_mod_chk_name" >>
/etc/modprobe.d/"$l_mod_name".conf
      fi
    }
    for l_mod_base_directory in $l_mod_path; do # Check if the module exists
on the system
      if [ -d "$l_mod_base_directory/${l_mod_name//-/}/" ] && [ -n "$(ls -A
"$l_mod_base_directory/${l_mod_name//-/}/")" ]; then
        a_output3+=(" - \"$l_mod_base_directory\"")
        l_mod_chk_name="$l_mod_name"
        [[ "$l_mod_name" =~ overlay ]] && l_mod_chk_name="$l_mod_name::-2"
        [ "$l_dl" != "y" ] && f_module_fix
      else
        printf '%s\n' " - kernel module: \"$l_mod_name\" doesn't exist in
\"$l_mod_base_directory\""
      fi
    done
    [ "${#a_output3[@]}" -gt 0 ] && printf '%s\n' "" " -- INFO --" " - module:
\"$l_mod_name\" exists in:" "${a_output3[@]}"
    [ "${#a_output2[@]}" -gt 0 ] && printf '%s\n' "" "${a_output2[@]}" ||
printf '%s\n' "" " - No changes needed"
    printf '%s\n' "" " - remediation of kernel module: \"$l_mod_name\"
complete" ""
  }
}
```

References:

1. NIST SP 800-53 Rev. 5: CM-7

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<u>4.8 Uninstall or Disable Unnecessary Services on Enterprise Assets and Software</u> Uninstall or disable unnecessary services on enterprise assets and software, such as an unused file sharing service, web application module, or service function.			
v7	<u>9.2 Ensure Only Approved Ports, Protocols and Services Are Running</u> Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1005, T1005.000	TA0005	M1050

1.1.1.9 Ensure usb-storage kernel module is not available (Automated)

Profile Applicability:

- Level 1 - Server
- Level 2 - Workstation

Description:

USB storage provides a means to transfer and store files ensuring persistence and availability of the files independent of network connection status. Its popularity and utility has led to USB-based malware being a simple and common means for network infiltration and a first step to establishing a persistent threat within a networked environment.

Rationale:

Restricting USB access on the system will decrease the physical attack surface for a device and diminish the possible vectors to introduce malware.

Impact:

Disabling the **usb-storage** module will disable any usage of USB storage devices.

If requirements and local site policy allow the use of such devices, other solutions should be configured accordingly instead. One example of a commonly used solution is **USBGuard**.

Audit:

Run the following script to verify:

- **IF** - the **usb-storage** kernel module is available in ANY installed kernel, verify:

- An entry including **/bin/true** or **/bin/false** exists in a file within the **/etc/modprobe.d/** directory
- The module is deny listed in a file within the **/etc/modprobe.d/** directory
- The module is not loaded in the running kernel

- **IF** - the **usb-storage** kernel module is not available on the system, or pre-compiled into the kernel, no additional configuration is necessary

```

#!/usr/bin/env bash

{
    a_output=() a_output2=() a_output3=() l_dl="" l_mod_name="usb-storage"
    l_mod_type="drivers"
    l_mod_path="$(readlink -f /lib/modules/**/kernel/$l_mod_type | sort -u)"
    f_module_chk()
    {
        l_dl="y" a_showconfig=()
        while IFS= read -r l_showconfig; do
            a_showconfig+=("$l_showconfig")
            done < <(modprobe --showconfig | grep -P --
'\b(install|blacklist)\h+'${l_mod_chk_name//-/}_'\b')
            if ! lsmod | grep "$l_mod_chk_name" &> /dev/null; then
                a_output+=(" - kernel module: \"$l_mod_name\" is not loaded")
            else
                a_output2+=(" - kernel module: \"$l_mod_name\" is loaded")
            fi
            if grep -Pq -- '\binstall\h+'${l_mod_chk_name//-/}_
/_}'\h+(\usr)?\bin\/(true|false)\b' <<< "${a_showconfig[*]}"; then
                a_output+=(" - kernel module: \"$l_mod_name\" is not loadable")
            else
                a_output2+=(" - kernel module: \"$l_mod_name\" is loadable")
            fi
            if grep -Pq -- '\bblacklist\h+'${l_mod_chk_name//-/}_'\b' <<<
"${a_showconfig[*]}"; then
                a_output+=(" - kernel module: \"$l_mod_name\" is deny listed")
            else
                a_output2+=(" - kernel module: \"$l_mod_name\" is not deny listed")
            fi
        }
        for l_mod_base_directory in $l_mod_path; do
            if [ -d "$l_mod_base_directory/${l_mod_name//-/}/" ] && [ -n "$(ls -A
"$l_mod_base_directory/${l_mod_name//-/}/")" ]; then
                a_output3+=(" - \"$l_mod_base_directory\"")
                l_mod_chk_name="$l_mod_name"
                [[ "$l_mod_name" =~ overlay ]] && l_mod_chk_name="$l_mod_name:-2)"
                [ "$l_dl" != "y" ] && f_module_chk
            else
                a_output+=(" - kernel module: \"$l_mod_name\" doesn't exist in
\"$l_mod_base_directory\"")
            fi
        done
        [ "${#a_output3[@]}" -gt 0 ] && printf '%s\n' "" " -- INFO --" " - module:
\"$l_mod_name\" exists in:" "${a_output3[@]}"
        if [ "${#a_output2[@]}" -le 0 ]; then
            printf '%s\n' "" "- Audit Result:" " ** PASS **" "${a_output[@]}"
        else
            printf '%s\n' "" "- Audit Result:" " ** FAIL **" " - Reason(s) for
audit failure:" "${a_output2[@]}"
            [ "${#a_output[@]}" -gt 0 ] && printf '%s\n' "- Correctly set:"
"${a_output[@]}"
        fi
    }
}

```

Remediation:

Run the following script to unload and disable the `usb-storage` module:

- **IF** - the `usb-storage` kernel module is available in ANY installed kernel:

- Create a file ending in `.conf` with `install usb-storage /bin/false` in the `/etc/modprobe.d/` directory
- Create a file ending in `.conf` with `blacklist usb-storage` in the `/etc/modprobe.d/` directory
- Run `modprobe -r usb-storage 2>/dev/null; rmmod usb-storage 2>/dev/null` to remove `usb-storage` from the kernel

- **IF** - the `usb-storage` kernel module is not available on the system, or pre-compiled into the kernel, no remediation is necessary

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

{
    a_output2=() a_output3=() l_dl="" l_mod_name="usb-storage"
l_mod_type="drivers"
    l_mod_path="$(readlink -f /lib/modules/**/kernel/$l_mod_type | sort -u)"
    f_module_fix()
    {
        l_dl="y" a_showconfig=()
        while IFS= read -r l_showconfig; do
            a_showconfig+=("$l_showconfig")
            done < <(modprobe --showconfig | grep -P --
'\b(install|blacklist)\h+' "${l_mod_chk_name//-/}_"' \b')
            if lsmod | grep "$l_mod_chk_name" &> /dev/null; then
                a_output2+=(" - unloading kernel module: \"$l_mod_name\"")
                modprobe -r "$l_mod_chk_name" 2>/dev/null; rmmod "$l_mod_name"
2>/dev/null
            fi
            if ! grep -Pq -- '\binstall\h+' "${l_mod_chk_name//-/}_"' \b' <<<
"${a_showconfig[*]}"; then
                a_output2+=(" - setting kernel module: \"$l_mod_name\" to
\"$(readlink -f /bin/false)\"")
                printf '%s\n' "install $l_mod_chk_name $(readlink -f /bin/false)" >>
/etc/modprobe.d/"$l_mod_name".conf
            fi
            if ! grep -Pq -- '\bblacklist\h+' "${l_mod_chk_name//-/}_"' \b' <<<
"${a_showconfig[*]}"; then
                a_output2+=(" - denylisting kernel module: \"$l_mod_name\"")
                printf '%s\n' "blacklist $l_mod_chk_name" >>
/etc/modprobe.d/"$l_mod_name".conf
            fi
        }
        for l_mod_base_directory in $l_mod_path; do # Check if the module exists
on the system
            if [ -d "$l_mod_base_directory/${l_mod_name//-/}/" ] && [ -n "$(ls -A
"$l_mod_base_directory/${l_mod_name//-/}/")" ]; then
                a_output3+=(" - \"$l_mod_base_directory\"")
                l_mod_chk_name="$l_mod_name"
                [[ "$l_mod_name" =~ overlay ]] && l_mod_chk_name="${l_mod_name::-2}"
                [ "$l_dl" != "y" ] && f_module_fix
            else
                printf '%s\n' " - kernel module: \"$l_mod_name\" doesn't exist in
\"$l_mod_base_directory\""
            fi
        done
        [ "${#a_output3[@]}" -gt 0 ] && printf '%s\n' "" " -- INFO --" " - module:
\"$l_mod_name\" exists in:" "${a_output3[@]}"
        [ "${#a_output2[@]}" -gt 0 ] && printf '%s\n' "" "${a_output2[@]}" ||
printf '%s\n' "" " - No changes needed"
        printf '%s\n' "" " - remediation of kernel module: \"$l_mod_name\"
complete" ""
    }
}
```

References:






1. NIST SP 800-53 Rev. 5: SI-3

Additional Information:

An alternative solution to disabling the usb-storage module may be found in USBGuard.

Use of USBGuard and construction of USB device policies should be done in alignment with site policy.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	10.3 <u>Disable Autorun and Autoplay for Removable Media</u> Disable autorun and autoplay auto-execute functionality for removable media.			
v7	13.7 <u>Manage USB Devices</u> If USB storage devices are required, enterprise software should be used that can configure systems to allow the use of specific devices. An inventory of such devices should be maintained.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1052, T1052.001, T1091, T1091.000, T1200, T1200.000	TA0001, TA0010	M1034

1.1.1.10 Ensure unused filesystems kernel modules are not available (Manual)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

Filesystem kernel modules are pieces of code that can be dynamically loaded into the Linux kernel to extend its filesystem capabilities, or so-called base kernel, of an operating system. Filesystem kernel modules are typically used to add support for new hardware (as device drivers), or for adding system calls.

Rationale:

While loadable filesystem kernel modules are a convenient method of modifying the running kernel, this can be abused by attackers on a compromised system to prevent detection of their processes or files, allowing them to maintain control over the system. Many rootkits make use of loadable filesystem kernel modules in this way.

Removing support for unneeded filesystem types reduces the local attack surface of the system. If this filesystem type is not needed, disable it. The following filesystem kernel modules have known CVE's and should be made unavailable if no dependencies exist:

- **afs** - CVE-2022-37402
- **ceph** - CVE-2022-0670
- **cifs** - CVE-2022-29869
- **exfat** CVE-2022-29973
- **ext** CVE-2022-1184
- **fat** CVE-2022-22043
- **fscache** CVE-2022-3630
- **fuse** CVE-2023-0386
- **gfs2** CVE-2023-3212
- **nfs_common** CVE-2023-6660
- **nfsd** CVE-2022-43945
- **smbfs_common** CVE-2022-2585

Impact:

This list may be quite extensive and covering all edges cases is difficult. Therefore, it's crucial to carefully consider the implications and dependencies before making any changes to the filesystem kernel module configurations.

Audit:

Run the following script to:

- Look at the filesystem kernel modules available to the currently running kernel.
- Exclude mounted filesystem kernel modules that don't currently have a CVE
- List filesystem kernel modules that are not fully disabled, or are loaded into the kernel

Review the generated output

```

#!/usr/bin/env bash

{
  a_output=(); a_output2=(); a_modprobe_config=(); a_excluded=(); a_available_modules=()
  a_ignore=("xfs" "vfat" "ext2" "ext3" "ext4")
  a_cve_exists=("afs" "ceph" "cifs" "exfat" "ext" "fat" "fscache" "fuse" "gfs2" "nfs_common"
  "nfsd" "smbfs_common")
  f_module_chk()
  {
    l_out2=""; grep -Pq -- "\b$l_mod_name\b" <<< "${a_cve_exists[*]}" && l_out2=" <- CVE
exists!"
    if ! grep -Pq -- '\bblacklist\b' "$l_mod_name" <<< "${a_modprobe_config[*]"; then
      a_output2+=(" - Kernel module: \"$l_mod_name\" is not fully disabled $l_out2")
    elif ! grep -Pq -- '\binstall\b' "$l_mod_name" <<< "${a_modprobe_config[*]"; then
      a_output2+=(" - Kernel module: \"$l_mod_name\" is not fully disabled $l_out2")
    fi
    if lsmod | grep "$l_mod_name" &> /dev/null; then # Check if the module is currently loaded
      l_output2+=(" - Kernel module: \"$l_mod_name\" is loaded" "")
    fi
  }
  while IFS= read -r -d $'\0' l_module_dir; do
    a_available_modules+=("${basename "$l_module_dir"}")
    done < <(find "$(readlink -f /lib/modules/"$(uname -r)"/kernel/fs)" -mindepth 1 -maxdepth 1 -
type d ! -empty -print0)
    while IFS= read -r l_exclude; do
      if grep -Pq -- "\b$l_exclude\b" <<< "${a_cve_exists[*]"; then
        a_output2+=(" - ** WARNING: kernel module: \"$l_exclude\" has a CVE and is currently
mounted! **")
      elif
        grep -Pq -- "\b$l_exclude\b" <<< "${a_available_modules[*]"; then
          a_output+=(" - Kernel module: \"$l_exclude\" is currently mounted - do NOT unload or
disable")
        fi
        ! grep -Pq -- "\b$l_exclude\b" <<< "${a_ignore[*]"; && a_ignore+=("$l_exclude")
      done < <(findmnt -knD | awk '{print $2}' | sort -u)
      while IFS= read -r l_config; do
        a_modprobe_config+=("$l_config")
        done < <(modprobe --showconfig | grep -P '^h*(blacklist|install)')
        for l_mod_name in "${a_available_modules[@]"; do # Iterate over all filesystem modules
          [[ "$l_mod_name" =~ overlay ]] && l_mod_name="$l_mod_name:-2"
          if grep -Pq -- "\b$l_mod_name\b" <<< "${a_ignore[*]"; then
            a_excluded+=(" - Kernel module: \"$l_mod_name\"")
          else
            f_module_chk
          fi
        done
        [ "${#a_excluded[@]}" -gt 0 ] && printf '%s\n' "" -- INFO -- \
"The following intentionally skipped" \
"${a_excluded[@]}"
        if [ "${#a_output2[@]}" -le 0 ]; then
          printf '%s\n' "" -- No unused filesystem kernel modules are enabled "${a_output[@]}" ""
        else
          printf '%s\n' "" -- Audit Result: --" " ** REVIEW the following **" "${a_output2[@]}"
          [ "${#a_output[@]}" -gt 0 ] && printf '%s\n' "" -- Correctly set: --" "${a_output[@]}" ""
        fi
      }
}

```

WARNING: disabling or denylisting filesystem modules that are in use on the system may be FATAL. It is extremely important to thoroughly review this list.

Remediation:

- **IF** - the module is available in the running kernel:

- Unload the filesystem kernel module from the kernel
- Create a file ending in **.conf** with install filesystem kernel modules **/bin/false** in the **/etc/modprobe.d/** directory
- Create a file ending in **.conf** with deny list filesystem kernel modules in the **/etc/modprobe.d/** directory

WARNING: unloading, disabling or denylisting filesystem modules that are in use on the system maybe FATAL. It is extremely important to thoroughly review the filesystems returned by the audit before following the remediation procedure.

*Example of unloading the **gfs2** kernel module:*

```
# modprobe -r gfs2 2>/dev/null
# rmmod gfs2 2>/dev/null
```

*Example of fully disabling the **gfs2** kernel module:*

```
# printf '%s\n' "blacklist gfs2" "install gfs2 /bin/false" >>
/etc/modprobe.d/gfs2.conf
```

Note:

- Disabling a kernel module by modifying the command above for each unused filesystem kernel module
- The example **gfs2** must be updated with the appropriate module name for the command or example script below to run correctly.

Below is an example Script that can be modified to use on various filesystem kernel modules manual remediation process:

Example Script

```

#!/usr/bin/env bash





{
    a_output2=(); a_output3=(); l_dl="" # Initialize arrays and clear
variables
    l_mod_name="gfs2" # set module name
    l_mod_type="fs" # set module type
    l_mod_path="$(readlink -f /lib/modules/**/kernel/$l_mod_type | sort -u)"
    f_module_fix()
    {
        l_dl="y" # Set to ignore duplicate checks
        a_showconfig=() # Create array with modprobe output
        while IFS= read -r l_showconfig; do
            a_showconfig+=("$l_showconfig")
            done < <(modprobe --showconfig | grep -P --
'\b(install|blacklist)\h+'"$l_mod_name"/-/_}"'\b')
            if lsmod | grep "$l_mod_name" &> /dev/null; then # Check if the module
is currently loaded
                a_output2+=(" - unloading kernel module: \"$l_mod_name\"")
                modprobe -r "$l_mod_name" 2>/dev/null; rmmod "$l_mod_name"
2>/dev/null
            fi
            if ! grep -Pq -- '\binstall\h+'"$l_mod_name"/-
/_}"'\h+(\usr)?\bin\/(true|false)\b' <<< "${a_showconfig[*]}"; then
                a_output2+=(" - setting kernel module: \"$l_mod_name\" to
\"$(readlink -f /bin/false)\"")
                printf '%s\n' "install $l_mod_name $(readlink -f /bin/false)" >>
/etc/modprobe.d/"$l_mod_name".conf
            fi
            if ! grep -Pq -- '\bblacklist\h+'"$l_mod_name"/-/_}"'\b' <<<
"${a_showconfig[*]}"; then
                a_output2+=(" - denylisting kernel module: \"$l_mod_name\"")
                printf '%s\n' "blacklist $l_mod_name" >>
/etc/modprobe.d/"$l_mod_name".conf
            fi
        }
        for l_mod_base_directory in $l_mod_path; do # Check if the module exists
on the system
            if [ -d "$l_mod_base_directory/${l_mod_name}/-/\}" ] && [ -n "$(ls -A
"$l_mod_base_directory/${l_mod_name}/-/\}")" ]; then
                a_output3+=(" - \"$l_mod_base_directory\"")
                [[ "$l_mod_name" =~ overlay ]] && l_mod_name="$l_mod_name::-2"
                [ "$l_dl" != "y" ] && f_module_fix
            else
                echo -e " - kernel module: \"$l_mod_name\" doesn't exist in
\"$l_mod_base_directory\""
            fi
        done
        [ "${#a_output3[@]}" -gt 0 ] && printf '%s\n' "" " -- INFO --" " - module:
\"$l_mod_name\" exists in: " "${a_output3[@]}"
        [ "${#a_output2[@]}" -gt 0 ] && printf '%s\n' "" "${a_output2[@]}" ||
printf '%s\n' "" " - No changes needed"
        printf '%s\n' "" " - remediation of kernel module: \"$l_mod_name\"
complete" ""
    }
}

```

References:

1. <https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=filesystem>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<u>4.8 Uninstall or Disable Unnecessary Services on Enterprise Assets and Software</u> Uninstall or disable unnecessary services on enterprise assets and software, such as an unused file sharing service, web application module, or service function.			
v7	<u>9.2 Ensure Only Approved Ports, Protocols and Services Are Running</u> Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system.			

1.1.2 Configure Filesystem Partitions

Directories that are used for system-wide functions can be further protected by placing them on separate partitions. This provides protection for resource exhaustion and enables the use of mounting options that are applicable to the directory's intended use. Users' data can be stored on separate partitions and have stricter mount options. A user partition is a filesystem that has been established for use by the users and does not contain software for system operations.

The recommendations in this section are easier to perform during initial system installation. If the system is already installed, it is recommended that a full backup be performed before repartitioning the system.

Note:

-IF- you are repartitioning a system that has already been installed (This may require the system to be in single-user mode):

- Mount the new partition to a temporary mountpoint e.g. `mount /dev/sda2 /mnt`
- Copy data from the original partition to the new partition. e.g. `cp -a /var/tmp/* /mnt`
- Verify that all data is present on the new partition. e.g. `ls -la /mnt`
- Unmount the new partition. e.g. `umount /mnt`
- Remove the data from the original directory that was in the old partition. e.g. `rm -Rf /var/tmp/*` Otherwise it will still consume space in the old partition that will be masked when the new filesystem is mounted.
- Mount the new partition to the desired mountpoint. e.g. `mount /dev/sda2 /var/tmp`
- Update `/etc/fstab` with the new mountpoint. e.g. `/dev/sda2 /var/tmp xfs defaults,rw,nosuid,nodev,noexec,relatime 0 0`

1.1.2.1 Configure /tmp

The `/tmp` directory is a world-writable directory used to store data used by the system and user applications for a short period of time. This data should have no expectation of surviving a reboot, as this directory is intended to be emptied after each reboot.

1.1.2.1.1 Ensure /tmp is a separate partition (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The `/tmp` directory is a world-writable directory used for temporary storage by all users and some applications.

- **IF** - an entry for `/tmp` exists in `/etc/fstab` it will take precedence over entries in systemd default unit file.

Note: In an environment where the main system is diskless and connected to iSCSI, entries in `/etc/fstab` may not take precedence.

`/tmp` can be configured to use `tmpfs`.

`tmpfs` puts everything into the kernel internal caches and grows and shrinks to accommodate the files it contains and is able to swap unneeded pages out to swap space. It has maximum size limits which can be adjusted on the fly via `mount -o remount`.

Since `tmpfs` lives completely in the page cache and on swap, all `tmpfs` pages will be shown as "Shmem" in `/proc/meminfo` and "Shared" in `free`. Notice that these counters also include shared memory. The most reliable way to get the count is using `df` and `du`.

`tmpfs` has three mount options for sizing:

- **size**: The limit of allocated bytes for this `tmpfs` instance. The default is half of your physical RAM without swap. If you oversize your `tmpfs` instances the machine will deadlock since the OOM handler will not be able to free that memory.
- **nr_blocks**: The same as size, but in blocks of `PAGE_SIZE`.
- **nr_inodes**: The maximum number of inodes for this instance. The default is half of the number of your physical RAM pages, or (on a machine with highmem) the number of lowmem RAM pages, whichever is the lower.

These parameters accept a suffix k, m or g and can be changed on remount. The size parameter also accepts a suffix % to limit this `tmpfs` instance to that percentage of your physical RAM. The default, when neither **size** nor **nr_blocks** is specified, is **size=50%**.

Rationale:

Making `/tmp` its own file system allows an administrator to set additional mount options such as the `noexec` option on the mount, making `/tmp` useless for an attacker to install executable code. It would also prevent an attacker from establishing a hard link to a system `setuid` program and wait for it to be updated. Once the program was updated, the hard link would be broken, and the attacker would have his own copy of the program. If the program happened to have a security vulnerability, the attacker could continue to exploit the known flaw.

This can be accomplished by either mounting `tmpfs` to `/tmp`, or creating a separate partition for `/tmp`.

Impact:

By design files saved to `/tmp` should have no expectation of surviving a reboot of the system. `tmpfs` is ram based and all files stored to `tmpfs` will be lost when the system is rebooted.

If files need to be persistent through a reboot, they should be saved to `/var/tmp` not `/tmp`.

Since the `/tmp` directory is intended to be world-writable, there is a risk of resource exhaustion if it is not bound to `tmpfs` or a separate partition.

Running out of `/tmp` space is a problem regardless of what kind of filesystem lies under it, but in a configuration where `/tmp` is not a separate file system it will essentially have the whole disk available, as the default installation only creates a single `/` partition. On the other hand, a RAM-based `/tmp` (as with `tmpfs`) will almost certainly be much smaller, which can lead to applications filling up the filesystem much more easily. Another alternative is to create a dedicated partition for `/tmp` from a separate volume or disk. One of the downsides of a disk-based dedicated partition is that it will be slower than `tmpfs` which is RAM-based.

Audit:

Run the following command and verify the output shows that **/tmp** is mounted. Particular requirements pertaining to mount options are covered in ensuing sections.

```
# findmnt -kn /tmp
```

Example output:

```
/tmp tmpfs tmpfs rw,nosuid,nodev,noexec
```

Ensure that systemd will mount the **/tmp** partition at boot time.

```
# systemctl is-enabled tmp.mount
```

Example output:

```
generated
```

Verify output is not **masked** or **disabled**.

Note: By default, systemd will output **generated** if there is an entry in **/etc/fstab** for **/tmp**. This just means systemd will use the entry in **/etc/fstab** instead of its default unit file configuration for **/tmp**.

Remediation:

First ensure that systemd is correctly configured to ensure that **/tmp** will be mounted at boot time.

```
# systemctl unmask tmp.mount
```

For specific configuration requirements of the **/tmp** mount for your environment, modify **/etc/fstab**.

Example of using **tmpfs** with specific mount options:

```
tmpfs /tmp tmpfs defaults,rw,nosuid,nodev,noexec,relatime,size=2G 0  
0
```

Note: the **size=2G** is an example of setting a specific size for **tmpfs**.





Example of using a volume or disk with specific mount options. The source location of the volume or disk will vary depending on your environment:

```
<device> /tmp <fstype> defaults,nodev,nosuid,noexec 0 0
```

References:

1. <https://www.freedesktop.org/wiki/Software/systemd/APIFileSystems/>
2. <https://www.freedesktop.org/software/systemd/man/systemd-fstab-generator.html>
3. <https://www.kernel.org/doc/Documentation/filesystems/tmpfs.txt>
4. NIST SP 800-53 Rev. 5: CM-7

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<u>4.8 Uninstall or Disable Unnecessary Services on Enterprise Assets and Software</u> Uninstall or disable unnecessary services on enterprise assets and software, such as an unused file sharing service, web application module, or service function.			
v7	<u>9.2 Ensure Only Approved Ports, Protocols and Services Are Running</u> Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1499, T1499.001	TA0005	M1022

1.1.2.1.2 Ensure nodev option set on /tmp partition (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The **nodev** mount option specifies that the filesystem cannot contain special devices.

Rationale:

Since the **/tmp** filesystem is not intended to support devices, set this option to ensure that users cannot create a block or character special devices in **/tmp**.

Audit:

- **IF** - a separate partition exists for **/tmp**, verify that the **nodev** option is set.

Run the following command to verify that the **nodev** mount option is set.

Example:

```
# findmnt -kn /tmp | grep -v nodev
```

Nothing should be returned

Remediation:

- **IF** - a separate partition exists for **/tmp**.

Edit the **/etc/fstab** file and add **nodev** to the fourth field (mounting options) for the **/tmp** partition.

Example:

```
<device> /tmp <fstype> defaults,rw,nosuid,nodev,noexec,relatime 0 0
```





Run the following command to remount **/tmp** with the configured options:

```
# mount -o remount /tmp
```

References:

1. See the **fstab(5)** manual page for more information.
2. NIST SP 800-53 Rev. 5: CM-7

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<u>4.8 Uninstall or Disable Unnecessary Services on Enterprise Assets and Software</u> Uninstall or disable unnecessary services on enterprise assets and software, such as an unused file sharing service, web application module, or service function.			
v7	<u>9.2 Ensure Only Approved Ports, Protocols and Services Are Running</u> Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1200, T1200.000	TA0005	M1022

1.1.2.1.3 Ensure nosuid option set on /tmp partition (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The **nosuid** mount option specifies that the filesystem cannot contain **setuid** files.

Rationale:

Since the **/tmp** filesystem is only intended for temporary file storage, set this option to ensure that users cannot create **setuid** files in **/tmp**.

Audit:

- **IF** - a separate partition exists for **/tmp**, verify that the **nosuid** option is set.

Run the following command to verify that the **nosuid** mount option is set.

Example:

```
# findmnt -kn /tmp | grep -v nosuid
```

Nothing should be returned

Remediation:

- **IF** - a separate partition exists for **/tmp**.

Edit the **/etc/fstab** file and add **nosuid** to the fourth field (mounting options) for the **/tmp** partition.

Example:

```
<device> /tmp <fstype> defaults,rw,nosuid,nodev,noexec,relatime 0 0
```







Run the following command to remount **/tmp** with the configured options:

```
# mount -o remount /tmp
```

References:

1. See the **fstab(5)** manual page for more information.
2. NIST SP 800-53 Rev. 5: AC-3, MP-2

CIS Controls:

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1548, T1548.001	TA0005	M1022

1.1.2.1.4 Ensure noexec option set on /tmp partition (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The **noexec** mount option specifies that the filesystem cannot contain executable binaries.

Rationale:

Since the **/tmp** filesystem is only intended for temporary file storage, set this option to ensure that users cannot run executable binaries from **/tmp**.

Impact:

Setting the **noexec** option on **/tmp** may prevent installation and/or updating of some 3rd party software.

Audit:

- **IF** - a separate partition exists for **/tmp**, verify that the **noexec** option is set.

Run the following command to verify that the **noexec** mount option is set.

Example:

```
# findmnt -kn /tmp | grep -v noexec
```

Nothing should be returned

Remediation:

- **IF** - a separate partition exists for **/tmp**.

Edit the **/etc/fstab** file and add **noexec** to the fourth field (mounting options) for the **/tmp** partition.

Example:

```
<device> /tmp <fstype> defaults,rw,nosuid,nodev,noexec,relatime 0 0
```







Run the following command to remount **/tmp** with the configured options:

```
# mount -o remount /tmp
```

References:

1. See the `fstab(5)` manual page for more information.
2. NIST SP 800-53 Rev. 5: AC-3, MP-2

CIS Controls:

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1204, T1204.002	TA0005	M1022

1.1.2.2 Configure /dev/shm

The `/dev/shm` directory is a world-writable directory that can function as shared memory that facilitates inter process communication (IPC)

1.1.2.2.1 Ensure /dev/shm is a separate partition (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The `/dev/shm` directory is a world-writable directory that can function as shared memory that facilitates inter process communication (IPC).

Rationale:

Making `/dev/shm` its own file system allows an administrator to set additional mount options such as the `noexec` option on the mount, making `/dev/shm` useless for an attacker to install executable code. It would also prevent an attacker from establishing a hard link to a system `setuid` program and wait for it to be updated. Once the program was updated, the hard link would be broken and the attacker would have his own copy of the program. If the program happened to have a security vulnerability, the attacker could continue to exploit the known flaw.

This can be accomplished by mounting `tmpfs` to `/dev/shm`.

Impact:

Since the `/dev/shm` directory is intended to be world-writable, there is a risk of resource exhaustion if it is not bound to a separate partition.

`/dev/shm` utilizing `tmpfs` can be resized using the `size={size}` parameter in the relevant entry in `/etc/fstab`.

Audit:

-IF- `/dev/shm` is to be used on the system, run the following command and verify the output shows that `/dev/shm` is mounted. Particular requirements pertaining to mount options are covered in ensuing sections.

```
# findmnt -kn /dev/shm
```

Example output:

```
/dev/shm tmpfs tmpfs rw,nosuid,nodev,noexec,relatime,seclabel
```

Remediation:

For specific configuration requirements of the `/dev/shm` mount for your environment, modify `/etc/fstab`.





Example:

```
tmpfs    /dev/shm          tmpfs
defaults,rw,nosuid,nodev,noexec,relatime,size=2G  0 0
```

References:

1. <https://www.freedesktop.org/wiki/Software/systemd/APIFileSystems/>
2. <https://www.freedesktop.org/software/systemd/man/systemd-fstab-generator.html>
3. NIST SP 800-53 Rev. 5: CM-7

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	4.8 <u>Uninstall or Disable Unnecessary Services on Enterprise Assets and Software</u> Uninstall or disable unnecessary services on enterprise assets and software, such as an unused file sharing service, web application module, or service function.			
v7	9.2 <u>Ensure Only Approved Ports, Protocols and Services Are Running</u> Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1499, T1499.001	TA0005	M1022

1.1.2.2.2 Ensure nodev option set on /dev/shm partition (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The **nodev** mount option specifies that the filesystem cannot contain special devices.

Rationale:

Since the **/dev/shm** filesystem is not intended to support devices, set this option to ensure that users cannot attempt to create special devices in **/dev/shm** partitions.

Audit:

- **IF** - a separate partition exists for **/dev/shm**, verify that the **nodev** option is set.

```
# findmnt -kn /dev/shm | grep -v 'nodev'
```

Nothing should be returned

Remediation:

- **IF** - a separate partition exists for **/dev/shm**.

Edit the **/etc/fstab** file and add **nodev** to the fourth field (mounting options) for the **/dev/shm** partition. See the **fstab(5)** manual page for more information.

Example:

```
tmpfs /dev/shm tmpfs defaults,rw,nosuid,nodev,noexec,relatime 0 0
```

Run the following command to remount **/dev/shm** with the configured options:

```
# mount -o remount /dev/shm
```

Note: It is recommended to use **tmpfs** as the device/filesystem type as **/dev/shm** is used as shared memory space by applications.







References:

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

Additional Information:

Some distributions mount `/dev/shm` through other means and require `/dev/shm` to be added to `/etc/fstab` even though it is already being mounted on boot. Others may configure `/dev/shm` in other locations and may override `/etc/fstab` configuration. Consult the documentation appropriate for your distribution.

CIS Controls:

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1200, T1200.000	TA0005	M1022

1.1.2.2.3 Ensure nosuid option set on /dev/shm partition (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The **nosuid** mount option specifies that the filesystem cannot contain **setuid** files.

Rationale:

Setting this option on a file system prevents users from introducing privileged programs onto the system and allowing non-root users to execute them.

Audit:

- **IF** - a separate partition exists for **/dev/shm**, verify that the **nosuid** option is set.

```
# findmnt -kn /dev/shm | grep -v 'nosuid'
```

Nothing should be returned

Remediation:

- **IF** - a separate partition exists for **/dev/shm**.

Edit the **/etc/fstab** file and add **nosuid** to the fourth field (mounting options) for the **/dev/shm** partition. See the **fstab(5)** manual page for more information.

Example:

```
tmpfs /dev/shm tmpfs defaults,rw,nosuid,nodev,noexec,relatime 0 0
```

Run the following command to remount **/dev/shm** with the configured options:

```
# mount -o remount /dev/shm
```

Note: It is recommended to use **tmpfs** as the device/filesystem type as **/dev/shm** is used as shared memory space by applications.







References:

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

Additional Information:

Some distributions mount **/dev/shm** through other means and require **/dev/shm** to be added to **/etc/fstab** even though it is already being mounted on boot. Others may configure **/dev/shm** in other locations and may override **/etc/fstab** configuration. Consult the documentation appropriate for your distribution.

CIS Controls:

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1548, T1548.001	TA0005	M1038

1.1.2.2.4 Ensure noexec option set on /dev/shm partition (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The **noexec** mount option specifies that the filesystem cannot contain executable binaries.

Rationale:

Setting this option on a file system prevents users from executing programs from shared memory. This deters users from introducing potentially malicious software on the system.

Audit:

- **IF** - a separate partition exists for **/dev/shm**, verify that the **noexec** option is set.

```
# findmnt -kn /dev/shm | grep -v 'noexec'
```

Nothing should be returned

Remediation:

- **IF** - a separate partition exists for **/dev/shm**.

Edit the **/etc/fstab** file and add **noexec** to the fourth field (mounting options) for the **/dev/shm** partition.

Example:

```
tmpfs /dev/shm tmpfs defaults,rw,nosuid,nodev,noexec,relatime 0 0
```

Run the following command to remount **/dev/shm** with the configured options:

```
# mount -o remount /dev/shm
```

Note: It is recommended to use **tmpfs** as the device/filesystem type as **/dev/shm** is used as shared memory space by applications.

References:

1. See the **fstab(5)** manual page for more information.
2. NIST SP 800-53 Rev. 5: AC-3, MP-2

CIS Controls:

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1204, T1204.002	TA0005	M1022

1.1.2.3 Configure /home

Please note that home directories can be mounted anywhere and are not necessarily restricted to **/home**, nor restricted to a single location, nor is the name restricted in any way.

Finding user home directories can be done by looking in **/etc/passwd**, looking over the mounted file systems with **mount** or querying the relevant database with **getent**.

```
for user in $(awk -F ':' '{print $1}' /etc/passwd); do echo "${user} - $(sudo  
getent passwd ${user} | awk -F ':' '{print $NF}')
```

1.1.2.3.1 Ensure separate partition exists for /home (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

The `/home` directory is used to support disk storage needs of local users.

Rationale:

The default installation only creates a single `/` partition. Since the `/home` directory contains user generated data, there is a risk of resource exhaustion. It will essentially have the whole disk available to fill up and impact the system as a whole. In addition, other operations on the system could fill up the disk unrelated to `/home` and impact all local users.

Configuring `/home` as its own file system allows an administrator to set additional mount options such as `noexec/nosuid/nODEV`. These options limit an attacker's ability to create exploits on the system. In the case of `/home` options such as `usrquota/grpquota` may be considered to limit the impact that users can have on each other with regards to disk resource exhaustion. Other options allow for specific behavior. See `man mount` for exact details regarding filesystem-independent and filesystem-specific options.

As `/home` contains user data, care should be taken to ensure the security and integrity of the data and mount point.

Impact:

Resizing filesystems is a common activity in cloud-hosted servers. Separate filesystem partitions may prevent successful resizing or may require the installation of additional tools solely for the purpose of resizing operations. The use of these additional tools may introduce their own security considerations.

Audit:

Run the following command and verify output shows **/home** is mounted:

```
# findmnt -kn /home  
  
/home    /dev/sdb  ext4    rw,nosuid,nodev,noexec,relatime,seclabel
```

Remediation:

For new installations, during installation create a custom partition setup and specify a separate partition for **/home**.

For systems that were previously installed, create a new partition and configure **/etc/fstab** as appropriate.







References:

1. AJ Lewis, "LVM HOWTO", <http://tldp.org/HOWTO/LVM-HOWTO/>
2. NIST SP 800-53 Rev. 5: CM-7

Additional Information:

When modifying **/home** it is advisable to bring the system to emergency mode (so auditd is not running), rename the existing directory, mount the new file system, and migrate the data over before returning to multi-user mode.

CIS Controls:

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1499, T1499.001	TA0005	M1038

1.1.2.3.2 Ensure nodev option set on /home partition (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The **nodev** mount option specifies that the filesystem cannot contain special devices.

Rationale:

Since the **/home** filesystem is not intended to support devices, set this option to ensure that users cannot create a block or character special devices in **/home**.

Audit:

- **IF** - a separate partition exists for **/home**, verify that the **nodev** option is set.

Run the following command to verify that the **nodev** mount option is set.

Example:

```
# findmnt -kn /home | grep -v nodev  
  
Nothing should be returned
```

Remediation:

- **IF** - a separate partition exists for **/home**.

Edit the **/etc/fstab** file and add **nodev** to the fourth field (mounting options) for the **/home** partition.

Example:

```
<device> /home    <fstype>          defaults,rw,nosuid,nodev,noexec,relatime 0 0
```







Run the following command to remount **/home** with the configured options:

```
# mount -o remount /home
```

References:

1. See the **fstab(5)** manual page for more information.
2. NIST SP 800-53 Rev. 5: AC-3, MP-2

CIS Controls:

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1200, T1200.000	TA0005	M1038

1.1.2.3.3 Ensure nosuid option set on /home partition (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The **nosuid** mount option specifies that the filesystem cannot contain **setuid** files.

Rationale:

Since the **/home** filesystem is only intended for user file storage, set this option to ensure that users cannot create **setuid** files in **/home**.

Audit:

- **IF** - a separate partition exists for **/home**, verify that the **nosuid** option is set. Run the following command to verify that the **nosuid** mount option is set.

Example:

```
# findmnt -kn /home | grep -v nosuid  
  
Nothing should be returned
```

Remediation:

- **IF** - a separate partition exists for **/home**.

Edit the **/etc/fstab** file and add **nosuid** to the fourth field (mounting options) for the **/home** partition.

Example:

```
<device> /home    <fstype>          defaults,rw,nosuid,nodev,noexec,relatime 0 0
```

Run the following command to remount **/home** with the configured options:

```
# mount -o remount /home
```

References:

1. See the **fstab(5)** manual page for more information.
2. NIST SP 800-53 Rev. 5: AC-3, MP-2

CIS Controls:

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1548, T1548.001	TA0005	M1022

1.1.2.4 Configure /var

The `/var` directory is used by daemons and other system services to temporarily store dynamic data. Some directories created by these processes may be world-writable.

1.1.2.4.1 Ensure separate partition exists for /var (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

The `/var` directory is used by daemons and other system services to temporarily store dynamic data. Some directories created by these processes may be world-writable.

Rationale:

The reasoning for mounting `/var` on a separate partition is as follows.

The default installation only creates a single `/` partition. Since the `/var` directory may contain world writable files and directories, there is a risk of resource exhaustion. It will essentially have the whole disk available to fill up and impact the system. In addition, other operations on the system could fill up the disk unrelated to `/var` and cause unintended behavior across the system as the disk is full. See `man auditd.conf` for details.

Configuring `/var` as its own file system allows an administrator to set additional mount options such as `noexec/nosuid/nODEV`. These options limit an attacker's ability to create exploits on the system. Other options allow for specific behavior. See `man mount` for exact details regarding filesystem-independent and filesystem-specific options.

An example of exploiting `/var` may be an attacker establishing a hard-link to a system `setuid` program and waiting for it to be updated. Once the program is updated, the hard-link can be broken and the attacker would have their own copy of the program. If the program happened to have a security vulnerability, the attacker could continue to exploit the known flaw.

Impact:

Resizing filesystems is a common activity in cloud-hosted servers. Separate filesystem partitions may prevent successful resizing or may require the installation of additional tools solely for the purpose of resizing operations. The use of these additional tools may introduce their own security considerations.

Audit:

Run the following command and verify output shows **/var** is mounted.

Example:

```
# findmnt -kn /var  
  
/var /dev/sdb ext4 rw,nosuid,nodev,noexec,relatime,seclabel
```

Remediation:

For new installations, during installation create a custom partition setup and specify a separate partition for **/var**.

For systems that were previously installed, create a new partition and configure **/etc/fstab** as appropriate.







References:

1. AJ Lewis, "LVM HOWTO", <http://tldp.org/HOWTO/LVM-HOWTO/>
2. NIST SP 800-53 Rev. 5: CM-7

Additional Information:

When modifying **/var** it is advisable to bring the system to emergency mode (so auditd is not running), rename the existing directory, mount the new file system, and migrate the data over before returning to multi-user mode.

CIS Controls:

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1499, T1499.001	TA0006	M1022

1.1.2.4.2 Ensure nodev option set on /var partition (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The **nodev** mount option specifies that the filesystem cannot contain special devices.

Rationale:

Since the **/var** filesystem is not intended to support devices, set this option to ensure that users cannot create a block or character special devices in **/var**.

Audit:

- **IF** - a separate partition exists for **/var**, verify that the **nodev** option is set.

Run the following command to verify that the **nodev** mount option is set.

Example:

```
# findmnt -kn /var | grep -v nodev
```

Nothing should be returned

Remediation:

- **IF** - a separate partition exists for **/var**.

Edit the **/etc/fstab** file and add **nodev** to the fourth field (mounting options) for the **/var** partition.

Example:

```
<device> /var <fstype> defaults,rw,nosuid,nodev,noexec,relatime 0 0
```







Run the following command to remount **/var** with the configured options:

```
# mount -o remount /var
```

References:

1. See the **fstab(5)** manual page for more information.
2. NIST SP 800-53 Rev. 5: AC-3, MP-2

CIS Controls:

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1200, T1200.000	TA0005	M1022

1.1.2.4.3 Ensure nosuid option set on /var partition (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The **nosuid** mount option specifies that the filesystem cannot contain **setuid** files.

Rationale:

Since the **/var** filesystem is only intended for variable files such as logs, set this option to ensure that users cannot create **setuid** files in **/var**.

Audit:

- **IF** - a separate partition exists for **/var**, verify that the **nosuid** option is set.

Run the following command to verify that the **nosuid** mount option is set.

Example:

```
# findmnt -kn /var | grep -v nosuid  
  
Nothing should be returned
```

Remediation:

- **IF** - a separate partition exists for **/var**.

Edit the **/etc/fstab** file and add **nosuid** to the fourth field (mounting options) for the **/var** partition.

Example:

```
<device> /var    <fstype>        defaults,rw,nosuid,nodev,noexec,relatime 0 0
```

Run the following command to remount **/var** with the configured options:

```
# mount -o remount /var
```

References:

1. See the **fstab(5)** manual page for more information.
2. NIST SP 800-53 Rev. 5: AC-3, MP-2

CIS Controls:

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1548, T1548.001	TA0005	M1022

1.1.2.5 Configure /var/tmp

The `/var/tmp` directory is a world-writable directory used for temporary storage by all users and some applications. Temporary files residing in `/var/tmp` are to be preserved between reboots.

1.1.2.5.1 Ensure separate partition exists for /var/tmp (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

The `/var/tmp` directory is a world-writable directory used for temporary storage by all users and some applications. Temporary files residing in `/var/tmp` are to be preserved between reboots.

Rationale:

The default installation only creates a single `/` partition. Since the `/var/tmp` directory is world-writable, there is a risk of resource exhaustion. In addition, other operations on the system could fill up the disk unrelated to `/var/tmp` and cause potential disruption to daemons as the disk is full.

Configuring `/var/tmp` as its own file system allows an administrator to set additional mount options such as `noexec/nosuid/nodev`. These options limit an attacker's ability to create exploits on the system.

Impact:

Resizing filesystems is a common activity in cloud-hosted servers. Separate filesystem partitions may prevent successful resizing or may require the installation of additional tools solely for the purpose of resizing operations. The use of these additional tools may introduce their own security considerations.

Audit:

Run the following command and verify output shows `/var/tmp` is mounted.

Example:

```
# findmnt -kn /var/tmp  
  
/var/tmp    /dev/sdb ext4    rw,nosuid,nodev,noexec,relatime,seclabel
```

Remediation:

For new installations, during installation create a custom partition setup and specify a separate partition for `/var/tmp`.

For systems that were previously installed, create a new partition and configure `/etc/fstab` as appropriate.







References:

1. AJ Lewis, "LVM HOWTO", <http://tldp.org/HOWTO/LVM-HOWTO/>
2. NIST SP 800-53 Rev. 5: CM-7

Additional Information:

When modifying `/var/tmp` it is advisable to bring the system to emergency mode (so auditd is not running), rename the existing directory, mount the new file system, and migrate the data over before returning to multi-user mode.

CIS Controls:

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1499, T1499.001	TA0005	M1022

1.1.2.5.2 Ensure nodev option set on /var/tmp partition (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The **nodev** mount option specifies that the filesystem cannot contain special devices.

Rationale:

Since the **/var/tmp** filesystem is not intended to support devices, set this option to ensure that users cannot create a block or character special devices in **/var/tmp**.

Audit:

- **IF** - a separate partition exists for **/var/tmp**, verify that the **nodev** option is set. Run the following command to verify that the **nodev** mount option is set.

Example:

```
# findmnt -kn /var/tmp | grep -v nodev  
  
Nothing should be returned
```

Remediation:

- **IF** - a separate partition exists for **/var/tmp**.

Edit the **/etc/fstab** file and add **nodev** to the fourth field (mounting options) for the **/var/tmp** partition.

Example:

```
<device> /var/tmp      <fstype>      defaults,rw,nosuid,nodev,noexec,relatime 0  
0
```







Run the following command to remount **/var/tmp** with the configured options:

```
# mount -o remount /var/tmp
```

References:

1. See the **fstab(5)** manual page for more information.
2. NIST SP 800-53 Rev. 5: AC-3, MP-2

CIS Controls:

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1499, T1499.001	TA0005	M1022

1.1.2.5.3 Ensure nosuid option set on /var/tmp partition (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The **nosuid** mount option specifies that the filesystem cannot contain **setuid** files.

Rationale:

Since the **/var/tmp** filesystem is only intended for temporary file storage, set this option to ensure that users cannot create **setuid** files in **/var/tmp**.

Audit:

- **IF** - a separate partition exists for **/var/tmp**, verify that the **nosuid** option is set. Run the following command to verify that the **nosuid** mount option is set.

Example:

```
# findmnt -kn /var/tmp | grep -v nosuid  
  
Nothing should be returned
```

Remediation:

- **IF** - a separate partition exists for **/var/tmp**.

Edit the **/etc/fstab** file and add **nosuid** to the fourth field (mounting options) for the **/var/tmp** partition.

Example:

```
<device> /var/tmp      <fstype>      defaults,rw,nosuid,nodev,noexec,relatime 0  
0
```







Run the following command to remount **/var/tmp** with the configured options:

```
# mount -o remount /var/tmp
```

References:

1. See the **fstab(5)** manual page for more information.
2. NIST SP 800-53 Rev. 5: AC-3, MP-2

CIS Controls:

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1548, T1548.001	TA0005	M1022

1.1.2.5.4 Ensure noexec option set on /var/tmp partition (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The **noexec** mount option specifies that the filesystem cannot contain executable binaries.

Rationale:

Since the **/var/tmp** filesystem is only intended for temporary file storage, set this option to ensure that users cannot run executable binaries from **/var/tmp**.

Audit:

- **IF** - a separate partition exists for **/var/tmp**, verify that the **noexec** option is set. Run the following command to verify that the **noexec** mount option is set.

Example:

```
# findmnt -kn /var/tmp | grep -v noexec
```

Nothing should be returned

Remediation:

- **IF** - a separate partition exists for **/var/tmp**.

Edit the **/etc/fstab** file and add **noexec** to the fourth field (mounting options) for the **/var/tmp** partition.

Example:

```
<device> /var/tmp    <fstype>    defaults,rw,nosuid,nodev,noexec,relatime 0  
0
```







Run the following command to remount **/var/tmp** with the configured options:

```
# mount -o remount /var/tmp
```

References:

1. See the **fstab(5)** manual page for more information.
2. NIST SP 800-53 Rev. 5: AC-3, MP-2

CIS Controls:

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1204, T1204.002	TA0005	M1022

1.1.2.6 Configure `/var/log`

The `/var/log` directory is used by system services to store log data.

1.1.2.6.1 Ensure separate partition exists for /var/log (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

The `/var/log` directory is used by system services to store log data.

Rationale:

The default installation only creates a single `/` partition. Since the `/var/log` directory contains log files which can grow quite large, there is a risk of resource exhaustion. It will essentially have the whole disk available to fill up and impact the system as a whole.

Configuring `/var/log` as its own file system allows an administrator to set additional mount options such as `noexec/nosuid/nodev`. These options limit an attacker's ability to create exploits on the system. Other options allow for specific behavior. See `man mount` for exact details regarding filesystem-independent and filesystem-specific options.

As `/var/log` contains log files, care should be taken to ensure the security and integrity of the data and mount point.

Impact:

Resizing filesystems is a common activity in cloud-hosted servers. Separate filesystem partitions may prevent successful resizing, or may require the installation of additional tools solely for the purpose of resizing operations. The use of these additional tools may introduce their own security considerations.

Audit:

Run the following command and verify output shows `/var/log` is mounted:

```
# findmnt -kn /var/log
/var/log /dev/sdb ext4    rw,nosuid,nodev,noexec,relatime,seclabel
```

Remediation:

For new installations, during installation create a custom partition setup and specify a separate partition for `/var/log`.

For systems that were previously installed, create a new partition and configure `/etc/fstab` as appropriate.






References:

1. AJ Lewis, "LVM HOWTO", <http://tldp.org/HOWTO/LVM-HOWTO/>
2. NIST SP 800-53 Rev. 5: CM-7

Additional Information:

When modifying `/var/log` it is advisable to bring the system to emergency mode (so auditd is not running), rename the existing directory, mount the new file system, and migrate the data over before returning to multiuser mode.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.3 Ensure Adequate Audit Log Storage Ensure that logging destinations maintain adequate storage to comply with the enterprise's audit log management process.			
v7	6.4 Ensure adequate storage for logs Ensure that all systems that store logs have adequate storage space for the logs generated.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1499, T1499.001	TA0005	M1022

1.1.2.6.2 Ensure nodev option set on /var/log partition (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The **nodev** mount option specifies that the filesystem cannot contain special devices.

Rationale:

Since the **/var/log** filesystem is not intended to support devices, set this option to ensure that users cannot create a block or character special devices in **/var/log**.

Audit:

- **IF** - a separate partition exists for **/var/log**, verify that the **nodev** option is set. Run the following command to verify that the **nodev** mount option is set.

Example:

```
# findmnt -kn /var/log | grep -v nodev  
  
Nothing should be returned
```

Remediation:

- **IF** - a separate partition exists for **/var/log**.

Edit the **/etc/fstab** file and add **nodev** to the fourth field (mounting options) for the **/var/log** partition.

Example:

```
<device> /var/log    <fstype>           defaults,rw,nosuid,nodev,noexec,relatime 0  
0
```







Run the following command to remount **/var/log** with the configured options:

```
# mount -o remount /var/log
```

References:

1. See the **fstab(5)** manual page for more information.
2. NIST SP 800-53 Rev. 5: AC-3, MP-2

CIS Controls:

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1200, T1200.000	TA0005	M1038

1.1.2.6.3 Ensure nosuid option set on /var/log partition (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The **nosuid** mount option specifies that the filesystem cannot contain **setuid** files.

Rationale:

Since the **/var/log** filesystem is only intended for log files, set this option to ensure that users cannot create **setuid** files in **/var/log**.

Audit:

- **IF** - a separate partition exists for **/var/log**, verify that the **nosuid** option is set. Run the following command to verify that the **nosuid** mount option is set.

Example:

```
# findmnt -kn /var/log | grep -v nosuid  
  
Nothing should be returned
```

Remediation:

- **IF** - a separate partition exists for **/var/log**.

Edit the **/etc/fstab** file and add **nosuid** to the fourth field (mounting options) for the **/var/log** partition.

Example:

```
<device> /var/log    <fstype>           defaults,rw,nosuid,nodev,noexec,relatime 0  
0
```

Run the following command to remount **/var/log** with the configured options:

```
# mount -o remount /var/log
```

References:

1. See the **fstab(5)** manual page for more information.
2. NIST SP 800-53 Rev. 5: AC-3, MP-2

CIS Controls:

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1548, T1548.001	TA0005	M1022

1.1.2.6.4 Ensure noexec option set on /var/log partition (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The **noexec** mount option specifies that the filesystem cannot contain executable binaries.

Rationale:

Since the **/var/log** filesystem is only intended for log files, set this option to ensure that users cannot run executable binaries from **/var/log**.

Audit:

- **IF** - a separate partition exists for **/var/log**, verify that the **noexec** option is set. Run the following command to verify that the **noexec** mount option is set.

Example:

```
# findmnt -kn /var/log | grep -v noexec
```

Nothing should be returned

Remediation:

- **IF** - a separate partition exists for **/var/log**.

Edit the **/etc/fstab** file and add **noexec** to the fourth field (mounting options) for the **/var/log** partition.

Example:

```
<device> /var/log    <fstype>    defaults,rw,nosuid,nodev,noexec,relatime 0  
0
```







Run the following command to remount **/var/log** with the configured options:

```
# mount -o remount /var/log
```

References:

1. See the **fstab(5)** manual page for more information.
2. NIST SP 800-53 Rev. 5: AC-3, MP-2

CIS Controls:

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1204, T1204.002	TA0005	M1022

1.1.2.7 Configure `/var/log/audit`

The auditing daemon, `auditd`, stores log data in the `/var/log/audit` directory.

1.1.2.7.1 Ensure separate partition exists for /var/log/audit (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

The auditing daemon, `auditd`, stores log data in the `/var/log/audit` directory.

Rationale:

The default installation only creates a single `/` partition. Since the `/var/log/audit` directory contains the `audit.log` file which can grow quite large, there is a risk of resource exhaustion. It will essentially have the whole disk available to fill up and impact the system as a whole. In addition, other operations on the system could fill up the disk unrelated to `/var/log/audit` and cause `auditd` to trigger its `space_left_action` as the disk is full. See `man auditd.conf` for details.

Configuring `/var/log/audit` as its own file system allows an administrator to set additional mount options such as `noexec/nosuid/noddev`. These options limit an attacker's ability to create exploits on the system. Other options allow for specific behavior. See `man mount` for exact details regarding filesystem-independent and filesystem-specific options.

As `/var/log/audit` contains audit logs, care should be taken to ensure the security and integrity of the data and mount point.

Impact:

Resizing filesystems is a common activity in cloud-hosted servers. Separate filesystem partitions may prevent successful resizing or may require the installation of additional tools solely for the purpose of resizing operations. The use of these additional tools may introduce their own security considerations.

Audit:

Run the following command and verify output shows `/var/log/audit` is mounted:

```
# findmnt -kn /var/log/audit  
  
/var/log/audit /dev/sdb ext4    rw,nosuid,nodev,noexec,relatime,seclabel
```

Remediation:

For new installations, during installation create a custom partition setup and specify a separate partition for `/var/log/audit`.

For systems that were previously installed, create a new partition and configure `/etc/fstab` as appropriate.






References:

1. AJ Lewis, "LVM HOWTO", <http://tldp.org/HOWTO/LVM-HOWTO/>
2. NIST SP 800-53 Rev. 5: CM-7

Additional Information:

When modifying `/var/log/audit` it is advisable to bring the system to emergency mode (so auditd is not running), rename the existing directory, mount the new file system, and migrate the data over before returning to multi-user mode.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.3 <u>Ensure Adequate Audit Log Storage</u> Ensure that logging destinations maintain adequate storage to comply with the enterprise's audit log management process.			
v7	6.4 <u>Ensure adequate storage for logs</u> Ensure that all systems that store logs have adequate storage space for the logs generated.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1499, T1499.001	TA0005	M1022

1.1.2.7.2 Ensure nodev option set on /var/log/audit partition (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The **nodev** mount option specifies that the filesystem cannot contain special devices.

Rationale:

Since the **/var/log/audit** filesystem is not intended to support devices, set this option to ensure that users cannot create a block or character special devices in **/var/log/audit**.

Audit:

- **IF** - a separate partition exists for **/var/log/audit**, verify that the **nodev** option is set. Run the following command to verify that the **nodev** mount option is set.

Example:

```
# findmnt -kn /var/log/audit | grep -v nodev
```

Nothing should be returned

Remediation:

- **IF** - a separate partition exists for **/var/log/audit**.

Edit the **/etc/fstab** file and add **nodev** to the fourth field (mounting options) for the **/var/log/audit** partition.

Example:

```
<device> /var/log/audit <fstype>  
defaults,rw,nosuid,nodev,noexec,relatime 0 0
```







Run the following command to remount **/var/log/audit** with the configured options:

```
# mount -o remount /var/log/audit
```

References:

1. See the **fstab(5)** manual page for more information.
2. NIST SP 800-53 Rev. 5: AC-3, MP-2

CIS Controls:

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1200, T1200.000	TA0005	M1022

1.1.2.7.3 Ensure nosuid option set on /var/log/audit partition (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The **nosuid** mount option specifies that the filesystem cannot contain **setuid** files.

Rationale:

Since the **/var/log/audit** filesystem is only intended for variable files such as logs, set this option to ensure that users cannot create **setuid** files in **/var/log/audit**.

Audit:

- **IF** - a separate partition exists for **/var/log/audit**, verify that the **nosuid** option is set.

Run the following command to verify that the **nosuid** mount option is set.

Example:

```
# findmnt -kn /var/log/audit | grep -v nosuid
```

Nothing should be returned

Remediation:

- **IF** - a separate partition exists for **/var/log/audit**.

Edit the **/etc/fstab** file and add **nosuid** to the fourth field (mounting options) for the **/var/log/audit** partition.

Example:

```
<device> /var/log/audit <fstype>  
defaults,rw,nosuid,nodev,noexec,relatime 0 0
```







Run the following command to remount **/var/log/audit** with the configured options:

```
# mount -o remount /var/log/audit
```

References:

1. See the **fstab(5)** manual page for more information.
2. NIST SP 800-53 Rev. 5: AC-3, MP-2

CIS Controls:

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1548, T1548.001	TA0005	M1022

1.1.2.7.4 Ensure noexec option set on /var/log/audit partition (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The **noexec** mount option specifies that the filesystem cannot contain executable binaries.

Rationale:

Since the **/var/log/audit** filesystem is only intended for audit logs, set this option to ensure that users cannot run executable binaries from **/var/log/audit**.

Audit:

- **IF** - a separate partition exists for **/var/log/audit**, verify that the **noexec** option is set.

Run the following command to verify that the **noexec** mount option is set.

Example:

```
# findmnt -kn /var/log/audit | grep -v noexec  
Nothing should be returned
```

Remediation:

- **IF** - a separate partition exists for **/var/log/audit**.

Edit the **/etc/fstab** file and add **noexec** to the fourth field (mounting options) for the **/var/log/audit** partition.

Example:

```
<device> /var/log/audit    <fstype>  
defaults,rw,nosuid,nodev,noexec,relatime 0 0
```







Run the following command to remount **/var/log/audit** with the configured options:

```
# mount -o remount /var/log/audit
```

References:

1. See the **fstab(5)** manual page for more information.
2. NIST SP 800-53 Rev. 5: AC-3, MP-2

CIS Controls:

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1204, T1204.002	TA0005	M1022

1.2 Package Management

Patch management procedures may vary widely between enterprises. Large enterprises may choose to install a local updates server that can be used in place of their distributions servers, whereas a single deployment of a system may prefer to get updates directly. Updates can be performed automatically or manually, depending on the site's policy for patch management. Organizations may prefer to test patches against their environment on a non-production system before rolling out to production.

Outdated software is vulnerable to cyber criminals and hackers. Software updates help reduce the risk to your organization. The release of software update notes often reveals the patched exploitable entry points to the public. Public knowledge of these exploits can make your organization more vulnerable to malicious actors attempting to gain entry to your system's data.

Software updates often offer new and improved features and speed enhancements.

For the purpose of this benchmark, the requirement is to ensure that a patch management process is defined and maintained, the specifics of which are left to the organization.

1.2.1 Configure Package Repositories

Patch management procedures may vary widely between enterprises. Large enterprises may choose to install a local updates server that can be used in place of their distributions servers, whereas a single deployment of a system may prefer to get updates directly. Updates can be performed automatically or manually, depending on the site's policy for patch management. Organizations may prefer to test patches against their environment on a non-production system before rolling out to production.

Outdated software is vulnerable to cyber criminals and hackers. Software updates help reduce the risk to your organization. The release of software update notes often reveals the patched exploitable entry points to the public. Public knowledge of these exploits can leave your organization more vulnerable to malicious actors attempting to gain access to your system's data.

Note: Creation of an appropriate patch management policy is left to the organization.

1.2.1.1 Ensure GPG keys are configured (Manual)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

Most package managers implement GPG key signing to verify package integrity during installation.

Rationale:

It is important to ensure that updates are obtained from a valid source to protect against spoofing that could lead to the inadvertent installation of malware on the system.

Audit:

Verify GPG keys are configured correctly for your package manager:

```
# apt-key list
```

Note:

- **apt-key list** is deprecated. Manage keyring files in **trusted.gpg.d** instead (see apt-key(8)).
- With the deprecation of **apt-key** it is recommended to use the **Signed-By** option in **sources.list** to require a repository to pass apt-secure(8) verification with a certain set of keys rather than all trusted keys apt has configured.

- OR -

1. Run the following script and verify GPG keys are configured correctly for your package manager:

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

{
  for file in /etc/apt/trusted.gpg.d/*.{gpg,asc}
  /etc/apt/sources.list.d/*.{gpg,asc} ; do
    if [ -f "$file" ]; then
      echo -e "File: $file"
      gpg --list-packets "$file" 2>/dev/null | awk '/keyid/ &&
!seen[$NF]++ {print "keyid:", $NF}'
      gpg --list-packets "$file" 2>/dev/null | awk '/Signed-By:/ {print
"signed-by:", $NF}'
      echo -e
    fi
  done
}
```

2. REVIEW and VERIFY to ensure that GPG keys are configured correctly for your package manager IAW site policy.













Remediation:

Update your package manager GPG keys in accordance with site policy.

References:

1. NIST SP 800-53 Rev. 5: SI-2
2. <https://manpages.debian.org/stretch/apt/sources.list.5.en.html>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<u>7.3 Perform Automated Operating System Patch Management</u> Perform operating system updates on enterprise assets through automated patch management on a monthly, or more frequent, basis.			
v8	<u>7.4 Perform Automated Application Patch Management</u> Perform application updates on enterprise assets through automated patch management on a monthly, or more frequent, basis.			
v7	<u>3.4 Deploy Automated Operating System Patch Management Tools</u> Deploy automated software update tools in order to ensure that the operating systems are running the most recent security updates provided by the software vendor.			
v7	<u>3.5 Deploy Automated Software Patch Management Tools</u> Deploy automated software update tools in order to ensure that third-party software on all systems is running the most recent security updates provided by the software vendor.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1195, T1195.001, T1195.002	TA0001	M1051

1.2.1.2 Ensure package manager repositories are configured (Manual)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

Systems need to have package manager repositories configured to ensure they receive the latest patches and updates.

Rationale:

If a system's package repositories are misconfigured important patches may not be identified or a rogue repository could introduce compromised software.

Audit:

Run the following command and verify package repositories are configured correctly:

```
# apt-cache policy
```













Remediation:

Configure your package manager repositories according to site policy.

References:

1. NIST SP 800-53 Rev. 5: SI-2

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<u>7.3 Perform Automated Operating System Patch Management</u> Perform operating system updates on enterprise assets through automated patch management on a monthly, or more frequent, basis.			
v8	<u>7.4 Perform Automated Application Patch Management</u> Perform application updates on enterprise assets through automated patch management on a monthly, or more frequent, basis.			
v7	<u>3.4 Deploy Automated Operating System Patch Management Tools</u> Deploy automated software update tools in order to ensure that the operating systems are running the most recent security updates provided by the software vendor.			
v7	<u>3.5 Deploy Automated Software Patch Management Tools</u> Deploy automated software update tools in order to ensure that third-party software on all systems is running the most recent security updates provided by the software vendor.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1195, T1195.001, T1195.002	TA0001	M1051

1.2.2 Configure Package Updates

1.2.2.1 Ensure updates, patches, and additional security software are installed (Manual)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

Periodically patches are released for included software either due to security flaws or to include additional functionality.

Rationale:

Newer patches may contain security enhancements that would not be available through the latest full update. As a result, it is recommended that the latest software patches be used to take advantage of the latest functionality. As with any software installation, organizations need to determine if a given update meets their requirements and verify the compatibility and supportability of any additional software against the update revision that is selected.

Audit:

Verify there are no updates or patches to install:

```
# apt update  
# apt -s upgrade
```

Remediation:

Run the following command to update all packages following local site policy guidance on applying updates and patches:

```
# apt update  
  
# apt upgrade  
- OR -  
# apt dist-upgrade
```

References:













1. NIST SP 800-53 Rev. 5: SI-2

Additional Information:

Site policy may mandate a testing period before installation onto production systems for available updates.

- `upgrade` - is used to install the newest versions of all packages currently installed on the system from the sources enumerated in `/etc/apt/sources.list` - OR - `/etc/apt/sources.list.d/ubuntu.sources`. Packages currently installed with new versions available are retrieved and upgraded; under no circumstances are currently installed packages removed, or packages not already installed retrieved and installed. New versions of currently installed packages that cannot be upgraded without changing the install status of another package will be left at their current version. An update must be performed first so that apt knows that new versions of packages are available.
- `dist-upgrade` - in addition to performing the function of `upgrade`, also intelligently handles changing dependencies with new versions of packages; apt has a "smart" conflict resolution system, and it will attempt to upgrade the most important packages at the expense of less important ones if necessary. So, `dist-upgrade` command may remove some packages. The `/etc/apt/sources.list` - OR - `/etc/apt/sources.list.d/ubuntu.sources` file contains a list of locations from which to retrieve desired package files. See also `apt_preferences(5)` for a mechanism for overriding the general settings for individual packages.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<u>7.3 Perform Automated Operating System Patch Management</u> Perform operating system updates on enterprise assets through automated patch management on a monthly, or more frequent, basis.			
v8	<u>7.4 Perform Automated Application Patch Management</u> Perform application updates on enterprise assets through automated patch management on a monthly, or more frequent, basis.			
v7	<u>3.4 Deploy Automated Operating System Patch Management Tools</u> Deploy automated software update tools in order to ensure that the operating systems are running the most recent security updates provided by the software vendor.			
v7	<u>3.5 Deploy Automated Software Patch Management Tools</u> Deploy automated software update tools in order to ensure that third-party software on all systems is running the most recent security updates provided by the software vendor.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1195, T1195.001	TA0005	M1051

1.3 Mandatory Access Control

Mandatory Access Control (MAC) provides an additional layer of access restrictions to processes on top of the base Discretionary Access Controls. By restricting how processes can access files and resources on a system the potential impact from vulnerabilities in the processes can be reduced.

Impact: Mandatory Access Control limits the capabilities of applications and daemons on a system, while this can prevent unauthorized access the configuration of MAC can be complex and difficult to implement correctly preventing legitimate access from occurring.

1.3.1 Configure AppArmor

AppArmor provides a Mandatory Access Control (MAC) system that greatly augments the default Discretionary Access Control (DAC) model. Under AppArmor MAC rules are applied by file paths instead of by security contexts as in other MAC systems. As such it does not require support in the filesystem and can be applied to network mounted filesystems for example. AppArmor security policies define what system resources applications can access and what privileges they can do so with. This automatically limits the damage that the software can do to files accessible by the calling user. The user does not need to take any action to gain this benefit. For an action to occur, both the traditional DAC permissions must be satisfied as well as the AppArmor MAC rules. The action will not be allowed if either one of these models does not permit the action. In this way, AppArmor rules can only make a system's permissions more restrictive and secure.

References:

1. AppArmor Documentation: <http://wiki.apparmor.net/index.php/Documentation>
2. Ubuntu AppArmor Documentation: <https://help.ubuntu.com/community/AppArmor>
3. SUSE AppArmor Documentation: <https://www.suse.com/documentation/apparmor/>

1.3.1.1 Ensure AppArmor is installed (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

AppArmor provides Mandatory Access Controls.

Rationale:

Without a Mandatory Access Control system installed only the default Discretionary Access Control system will be available.

Audit:

Run the following command to verify that **apparmor** is installed:

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

Run the following command to verify that **apparmor-utils** is installed:

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

Remediation:







Install AppArmor.

```
# apt install apparmor apparmor-utils
```

References:

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

CIS Controls:

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1068, T1068.000, T1565, T1565.001, T1565.003	TA0003	M1026

1.3.1.2 Ensure AppArmor is enabled in the bootloader configuration (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

Configure AppArmor to be enabled at boot time and verify that it has not been overwritten by the bootloader boot parameters.

Note: This recommendation is designed around the grub bootloader, if LILO or another bootloader is in use in your environment enact equivalent settings.

Rationale:

AppArmor must be enabled at boot time in your bootloader configuration to ensure that the controls it provides are not overridden.

Audit:

Run the following command to verify that all **linux** lines have the **apparmor=1** parameter set:

```
# grep "^s*linux" /boot/grub/grub.cfg | grep -v "apparmor=1"
```

Nothing should be returned.

Run the following command to verify that all **linux** lines have the **security=apparmor** parameter set:

```
# grep "^s*linux" /boot/grub/grub.cfg | grep -v "security=apparmor"
```

Nothing should be returned.

Remediation:

Edit **/etc/default/grub** and add the **apparmor=1** and **security=apparmor** parameters to the **GRUB_CMDLINE_LINUX=** line

```
GRUB_CMDLINE_LINUX="apparmor=1 security=apparmor"
```







Run the following command to update the **grub2** configuration:

```
# update-grub
```

References:

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

CIS Controls:

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1068, T1068.000, T1565, T1565.001, T1565.003	TA0003	M1026

1.3.1.3 Ensure all AppArmor Profiles are in enforce or complain mode (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

AppArmor profiles define what resources applications are able to access.

Rationale:

Security configuration requirements vary from site to site. Some sites may mandate a policy that is stricter than the default policy, which is perfectly acceptable. This item is intended to ensure that any policies that exist on the system are activated.

Audit:

Run the following command and verify that profiles are loaded, and are in either enforce or complain mode:

```
# apparmor_status | grep profiles
```

Review output and ensure that profiles are loaded, and in either enforce or complain mode:

```
37 profiles are loaded.  
35 profiles are in enforce mode.  
2 profiles are in complain mode.  
4 processes have profiles defined.
```

Run the following command and verify no processes are unconfined

```
# apparmor_status | grep processes
```

Review the output and ensure no processes are unconfined:

```
4 processes have profiles defined.  
4 processes are in enforce mode.  
0 processes are in complain mode.  
0 processes are unconfined but have a profile defined.
```


Remediation:

Run the following command to set all profiles to enforce mode:

```
# aa-enforce /etc/apparmor.d/*
```

- OR -

Run the following command to set all profiles to complain mode:







```
# aa-complain /etc/apparmor.d/*
```

Note: Any unconfined processes may need to have a profile created or activated for them and then be restarted.

References:

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

CIS Controls:

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1497	TA0005	

1.3.1.4 Ensure all AppArmor Profiles are enforcing (Automated)

Profile Applicability:

- Level 2 - Server
- Level 2 - Workstation

Description:

AppArmor profiles define what resources applications are able to access.

Rationale:

Security configuration requirements vary from site to site. Some sites may mandate a policy that is stricter than the default policy, which is perfectly acceptable. This item is intended to ensure that any policies that exist on the system are activated.

Audit:

Run the following commands and verify that profiles are loaded and are not in complain mode:

```
# apparmor_status | grep profiles
```

Review output and ensure that profiles are loaded, and in enforce mode:

```
34 profiles are loaded.
34 profiles are in enforce mode.
0 profiles are in complain mode.
2 processes have profiles defined.
```

Run the following command and verify that no processes are unconfined:

```
apparmor_status | grep processes
```

Review the output and ensure no processes are unconfined:

```
2 processes have profiles defined.
2 processes are in enforce mode.
0 processes are in complain mode.
0 processes are unconfined but have a profile defined.
```

Remediation:

Run the following command to set all profiles to enforce mode:

```
# aa-enforce /etc/apparmor.d/*
```

Note: Any unconfined processes may need to have a profile created or activated for them and then be restarted

References:

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

CIS Controls:

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1068, T1068.000, T1565, T1565.001, T1565.003	TA0005	M1048

1.4 Configure Bootloader

The recommendations in this section focus on securing the bootloader and settings involved in the boot process directly.

1.4.1 Ensure bootloader password is set (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

Setting the boot loader password will require that anyone rebooting the system must enter a password before being able to set command line boot parameters

Rationale:

Requiring a boot password upon execution of the boot loader will prevent an unauthorized user from entering boot parameters or changing the boot partition. This prevents users from weakening security (e.g. turning off AppArmor at boot time).

Impact:

If password protection is enabled, only the designated superuser can edit a GRUB 2 menu item by pressing "e" or access the GRUB 2 command line by pressing "c"

If GRUB 2 is set up to boot automatically to a password-protected menu entry the user has no option to back out of the password prompt to select another menu entry. Holding the SHIFT key will not display the menu in this case. The user must enter the correct username and password. If unable to do so, the configuration files will have to be edited via a LiveCD or other means to fix the problem

You can add **--unrestricted** to the menu entries to allow the system to boot without entering a password. A password will still be required to edit menu items.

More Information: <https://help.ubuntu.com/community/Grub2/Passwords>

Audit:

Run the following commands and verify output matches:

```
# grep "^set superusers" /boot/grub/grub.cfg
set superusers="<username>"
# awk -F. '/^\s*password/ {print $1"."$2"."$3}' /boot/grub/grub.cfg
password_pbkdf2 <username> grub.pbkdf2.sha512
```

Remediation:

Create an encrypted password with **grub-mkpasswd-pbkdf2**:

```
# grub-mkpasswd-pbkdf2 --iteration-count=600000 --salt=64  
  
Enter password: <password>  
Reenter password: <password>  
PBKDF2 hash of your password is <encrypted-password>
```

Add the following into a custom **/etc/grub.d** configuration file:

```
cat <<EOF  
exec tail -n +2 $0  
set superusers="<username>"  
password_pbkdf2 <username> <encrypted-password>  
EOF
```

The superuser/user information and password should not be contained in the **/etc/grub.d/00_header** file as this file could be overwritten in a package update. If there is a requirement to be able to boot/reboot without entering the password, edit **/etc/grub.d/10_linux** and add **--unrestricted** to the line **CLASS=**
Example:

```
CLASS="--class gnu-linux --class gnu --class os --unrestricted"
```

Run the following command to update the **grub2** configuration:

```
# update-grub
```

Default Value:

This recommendation is designed around the grub bootloader, if LILO or another bootloader is in use in your environment enact equivalent settings.

Replace **/boot/grub/grub.cfg** with the appropriate grub configuration file for your environment.







References:

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

Additional Information:

Changes to **/etc/grub.d/10_linux** may be overwritten during updates to the **grub-common** package. You should review any changes to this file before rebooting otherwise the system may unexpectedly prompt for a password on the next boot.

CIS Controls:

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1542, T1542.000	TA0003	M1046

1.4.2 Ensure access to bootloader config is configured (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The grub configuration file contains information on boot settings and passwords for unlocking boot options.

Rationale:

Setting the permissions to read and write for root only prevents non-root users from seeing the boot parameters or changing them. Non-root users who read the boot parameters may be able to identify weaknesses in security upon boot and be able to exploit them.

Audit:

Run the following command and verify **Uid** and **Gid** are both **0/root** and **Access** is **0600** or more restrictive.

```
# stat -Lc 'Access: (%#a/%A)  Uid: ( %u/ %U) Gid: ( %g/ %G) '
/boot/grub/grub.cfg
Access: (0600/-rw-----)  Uid: ( 0/ root) Gid: ( 0/ root)
```

Remediation:

Run the following commands to set permissions on your grub configuration:

```
# chown root:root /boot/grub/grub.cfg
# chmod u-x,go-rwx /boot/grub/grub.cfg
```

Default Value:

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

References:







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

Additional Information:

This recommendation is designed around the grub bootloader, if LILO or another bootloader is in use in your environment enact equivalent settings.

Replace `/boot/grub/grub.cfg` with the appropriate grub configuration file for your environment

CIS Controls:

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1542, T1542.000	TA0005, TA0007	M1022

1.5 Configure Additional Process Hardening

1.5.1 Ensure address space layout randomization is enabled (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

Address space layout randomization (ASLR) is an exploit mitigation technique which randomly arranges the address space of key data areas of a process.

Rationale:

Randomly placing virtual memory regions will make it difficult to write memory page exploits as the memory placement will be consistently shifting.

Audit:

Run the following script to verify the following kernel parameter is set in the running configuration and correctly loaded from a kernel parameter configuration file:

- `kernel.randomize_va_space` is set to 2

Note: kernel parameters are loaded by file and parameter order precedence. The following script observes this precedence as part of the auditing procedure. The parameters being checked may be set correctly in a file. If that file is superseded, the parameter is overridden by an incorrect setting later in that file, or in a canonically later file, that "correct" setting will be ignored both by the script and by the system during a normal kernel parameter load sequence.

```

#!/usr/bin/env bash

{
    a_output=(); a_output2=(); a_parlist=(kernel.randomize_va_space=2)
    l_ufwscf="$([ -f /etc/default/ufw ] && awk -F= '/^\s*IPT_SYSCTL=/ {print $2}' /etc/default/ufw)"
    f_kernel_parameter_chk()
    {
        l_running_parameter_value="$(sysctl "$l_parameter_name" | awk -F= '{print $2}' | xargs)" # Check running configuration
        if grep -Pq -- '\b'"$l_parameter_value"' \b' <<<
"$l_running_parameter_value"; then
            a_output+=(" - \"$l_parameter_name\" is correctly set to
\"$l_running_parameter_value\""
                "    in the running configuration")
        else
            a_output2+=(" - \"$l_parameter_name\" is incorrectly set to
\"$l_running_parameter_value\"" \
                "    in the running configuration" \
                "    and should have a value of: \"$l_value_out\"")
        fi
        unset A_out; declare -A A_out # Check durable setting (files)
        while read -r l_out; do
            if [ -n "$l_out" ]; then
                if [[ $l_out =~ ^\s*# ]]; then
                    l_file="$l_out// # /}"
                else
                    l_kpar="$(awk -F= '{print $1}' <<< "$l_out" | xargs)"
                    [ "$l_kpar" = "$l_parameter_name" ] &&
A_out+=([ "$l_kpar" ]="$l_file")
                fi
            fi
            done < <("$l_systemdsysctl" --cat-config | grep -Po
'^\h*([\^#\n\r]+|\#\h*\[/[\^#\n\r\h]+\.\conf\b)')
            if [ -n "$l_ufwscf" ]; then # Account for systems with UFW (Not covered
by systemd-sysctl --cat-config)
                l_kpar="$(grep -Po "^\h*$l_parameter_name\b" "$l_ufwscf" | xargs)"
                l_kpar="$l_kpar//\./."
                [ "$l_kpar" = "$l_parameter_name" ] &&
A_out+=([ "$l_kpar" ]="$l_ufwscf")
            fi
            if (( ${#A_out[@]} > 0 )); then # Assess output from files and generate
output
                while IFS="" read -r l_fkpname l_file_parameter_value; do
                    l_fkpname="$l_fkpname// /}";
l_file_parameter_value="$l_file_parameter_value// /}"
                    if grep -Pq -- '\b'"$l_parameter_value"' \b' <<<
"$l_file_parameter_value"; then
                        a_output+=(" - \"$l_parameter_name\" is correctly set to
\"$l_file_parameter_value\"" \
                            "    in \"$(printf '%s' "${A_out[@]}")\"")
                    else
                        a_output2+=(" - \"$l_parameter_name\" is incorrectly set to
\"$l_file_parameter_value\"" \
                            "    in \"$(printf '%s' "${A_out[@]}")\"" \
                            "    and should have a value of: \"$l_value_out\"")
                    fi
                done
            fi
        }
    }
}

```

```

done < <(grep -Po -- "\^\\h*$l_parameter_name\\h*\\h*\\H+"
"${A_out[@]}")
else
a_output2+=(" - \"$l_parameter_name\" is not set in an included
file" \
    ** Note: \"$l_parameter_name\" May be set in a file that's
ignored by load procedure **)
fi
}
l_systemdsysctl="$(readlink -f /lib/systemd/systemd-sysctl)"
while IFS="=" read -r l_parameter_name l_parameter_value; do # Assess and
check parameters
l_parameter_name="${l_parameter_name// /}";
l_parameter_value="${l_parameter_value// /}"
l_value_out="${l_parameter_value//-/ through }";
l_value_out="${l_value_out//|/ or }"
l_value_out="$(tr -d '()' <<< "$l_value_out")"
f_kernel_parameter_chk
done < <(printf '%s\n' "${a_parlist[@]}")
if [ "${#a_output2[@]}" -le 0 ]; then
printf '%s\n' "" "- Audit Result:" " ** PASS **" "${a_output[@]}" ""
else
printf '%s\n' "" "- Audit Result:" " ** FAIL **" " - Reason(s) for
audit failure:" "${a_output2[@]}"
[ "${#a_output[@]}" -gt 0 ] && printf '%s\n' "" "- Correctly set:"
"${a_output[@]}" ""
fi
}

```

Remediation:

Set the following parameter in `/etc/sysctl.conf` or a file in `/etc/sysctl.d/` ending in `.conf`:

- `kernel.randomize_va_space = 2`

Example:

```
# printf "%s\n" "kernel.randomize_va_space = 2" >> /etc/sysctl.d/60-
kernel_sysctl.conf
```

Run the following command to set the active kernel parameter:

```
# sysctl -w kernel.randomize_va_space=2
```

Note: If these settings appear in a canonically later file, or later in the same file, these settings will be overwritten





Default Value:

`kernel.randomize_va_space = 2`

References:

1. <http://manpages.ubuntu.com/manpages/focal/man5/sysctl.d.5.html>
2. CCI-000366: The organization implements the security configuration settings
3. NIST SP 800-53 Rev. 5: CM-6

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	10.5 <u>Enable Anti-Exploitation Features</u> Enable anti-exploitation features on enterprise assets and software, where possible, such as Microsoft® Data Execution Prevention (DEP), Windows® Defender Exploit Guard (WDEG), or Apple® System Integrity Protection (SIP) and Gatekeeper™.			
v7	8.3 <u>Enable Operating System Anti-Exploitation Features/ Deploy Anti-Exploit Technologies</u> Enable anti-exploitation features such as Data Execution Prevention (DEP) or Address Space Layout Randomization (ASLR) that are available in an operating system or deploy appropriate toolkits that can be configured to apply protection to a broader set of applications and executables.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1068, T1068.000	TA0002	M1050

1.5.2 Ensure ptrace_scope is restricted (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The `ptrace()` system call provides a means by which one process (the "tracer") may observe and control the execution of another process (the "tracee"), and examine and change the tracee's memory and registers.

The sysctl settings (writable only with CAP_SYS_PTRACE) are:

- **0** - classic ptrace permissions: a process can `PTRACE_ATTACH` to any other process running under the same uid, as long as it is dumpable (i.e. did not transition uids, start privileged, or have called `prctl(PR_SET_DUMPABLE...)` already). Similarly, `PTRACE_TRACEME` is unchanged.
- **1** - restricted ptrace: a process must have a predefined relationship with the inferior it wants to call `PTRACE_ATTACH` on. By default, this relationship is that of only its descendants when the above classic criteria is also met. To change the relationship, an inferior can call `prctl(PR_SET_PTRACER, debugger, ...)` to declare an allowed debugger PID to call `PTRACE_ATTACH` on the inferior. Using `PTRACE_TRACEME` is unchanged.
- **2** - admin-only attach: only processes with `CAP_SYS_PTRACE` may use ptrace with `PTRACE_ATTACH`, or through children calling `PTRACE_TRACEME`.
- **3** - no attach: no processes may use ptrace with `PTRACE_ATTACH` nor via `PTRACE_TRACEME`. Once set, this sysctl value cannot be changed.

Rationale:

If one application is compromised, it would be possible for an attacker to attach to other running processes (e.g. Bash, Firefox, SSH sessions, GPG agent, etc) to extract additional credentials and continue to expand the scope of their attack.

Enabling restricted mode will limit the ability of a compromised process to `PTRACE_ATTACH` on other processes running under the same user. With restricted mode, ptrace will continue to work with root user.

Audit:

Run the following script to verify the following kernel parameter is set in the running configuration and correctly loaded from a kernel parameter configuration file:

- `kernel.yama.ptrace_scope` is set to a value of: 1, 2, or 3

Note: kernel parameters are loaded by file and parameter order precedence. The following script observes this precedence as part of the auditing procedure. The parameters being checked may be set correctly in a file. If that file is superseded, the parameter is overridden by an incorrect setting later in that file, or in a canonically later file, that "correct" setting will be ignored both by the script and by the system during a normal kernel parameter load sequence.


```

#!/usr/bin/env bash

{
    a_output=(); a_output2=(); a_parlist=("kernel.yama.pttrace_scope=(1|2|3)")
    l_ufwscf="$([ -f /etc/default/ufw ] && awk -F= '/^\s*IPT_SYSCTL=/ {print $2}' /etc/default/ufw)"
    f_kernel_parameter_chk()
    {
        l_running_parameter_value="$(sysctl "$l_parameter_name" | awk -F= '{print $2}' | xargs)" # Check running configuration
        if grep -Pq -- '\b'"$l_parameter_value"' \b' <<<
"$l_running_parameter_value"; then
            a_output+=(" - \"$l_parameter_name\" is correctly set to
\"$l_running_parameter_value\"")
            "    in the running configuration")
        else
            a_output2+=(" - \"$l_parameter_name\" is incorrectly set to
\"$l_running_parameter_value\"") \
            "    in the running configuration" \
            "    and should have a value of: \"$l_value_out\"")
        fi
        unset A_out; declare -A A_out # Check durable setting (files)
        while read -r l_out; do
            if [ -n "$l_out" ]; then
                if [[ $l_out =~ ^\s*# ]]; then
                    l_file="$l_out// # /}"
                else
                    l_kpar="$(awk -F= '{print $1}' <<< "$l_out" | xargs)"
                    [ "$l_kpar" = "$l_parameter_name" ] &&
A_out+=([ "$l_kpar" ]="$l_file")
                fi
            fi
            done < <("$l_systemdsysctl" --cat-config | grep -Po
'^\h*([\^#\n\r]+|\#\h*\[/[\^#\n\r\h]+\.\conf\b)')
            if [ -n "$l_ufwscf" ]; then # Account for systems with UFW (Not covered
by systemd-sysctl --cat-config)
                l_kpar="$(grep -Po "^\h*$l_parameter_name\b" "$l_ufwscf" | xargs)"
                l_kpar="$l_kpar//\./."
                [ "$l_kpar" = "$l_parameter_name" ] &&
A_out+=([ "$l_kpar" ]="$l_ufwscf")
            fi
            if (( ${#A_out[@]} > 0 )); then # Assess output from files and generate
output
                while IFS="" read -r l_fkpname l_file_parameter_value; do
                    l_fkpname="$l_fkpname// /}";
                    l_file_parameter_value="$l_file_parameter_value// /}"
                    if grep -Pq -- '\b'"$l_parameter_value"' \b' <<<
"$l_file_parameter_value"; then
                        a_output+=(" - \"$l_parameter_name\" is correctly set to
\"$l_file_parameter_value\"") \
                        "    in \"$(printf '%s' "${A_out[@]}")\"")
                    else
                        a_output2+=(" - \"$l_parameter_name\" is incorrectly set to
\"$l_file_parameter_value\"") \
                        "    in \"$(printf '%s' "${A_out[@]}")\"") \
                        "    and should have a value of: \"$l_value_out\"")
                    fi
                done
            fi
        }
    }
}

```

```

done < <(grep -Po -- "\^h*$l_parameter_name\h*=\h*\H+"
"${A_out[@]}")
else
a_output2+=(" - \"$l_parameter_name\" is not set in an included
file" \
    ** Note: \"$l_parameter_name\" May be set in a file that's
ignored by load procedure **)
fi
}
l_systemdsysctl="$(readlink -f /lib/systemd/systemd-sysctl)"
while IFS="=" read -r l_parameter_name l_parameter_value; do # Assess and
check parameters
l_parameter_name="${l_parameter_name// /}";
l_parameter_value="${l_parameter_value// /}"
l_value_out="${l_parameter_value//-/ through }";
l_value_out="${l_value_out//|/ or }"
l_value_out="$(tr -d '()' <<< "$l_value_out")"
f_kernel_parameter_chk
done < <(printf '%s\n' "${a_parlist[@]}")
if [ "${#a_output2[@]}" -le 0 ]; then
printf '%s\n' "" "- Audit Result:" " ** PASS **" "${a_output[@]}" ""
else
printf '%s\n' "" "- Audit Result:" " ** FAIL **" " - Reason(s) for
audit failure:" "${a_output2[@]}"
[ "${#a_output[@]}" -gt 0 ] && printf '%s\n' "" "- Correctly set:"
"${a_output[@]}" ""
fi
}

```

Remediation:

Set the **kernel.yama.ptrace_scope** parameter in **/etc/sysctl.conf** or a file in **/etc/sysctl.d/** ending in **.conf** to a value of **1, 2, or 3**:

```

kernel.yama.ptrace_scope = 1
- OR -
kernel.yama.ptrace_scope = 2
- OR -
kernel.yama.ptrace_scope = 3

```

Example:

```

# printf "%s\n" "kernel.yama.ptrace_scope = 1" >> /etc/sysctl.d/60-
kernel_sysctl.conf

```

Run the following command to set the active kernel parameter:

```

# sysctl -w kernel.yama.ptrace_scope=1

```

Note:

- If a value of **2** or **3** is preferred, or required by local site policy, replace the **1** with the desired value of **2** or **3** in the example above
- If this setting appears in a canonically later file, or later in the same file, the setting will be overwritten

Default Value:

kernel.yama.ptrace_scope = 0





References:

1. <https://www.kernel.org/doc/Documentation/security/Yama.txt>
2. <https://github.com/raj3shp/termspy>
3. NIST SP 800-53 Rev. 5: CM-6

Additional Information:

Ptrace is very rarely used by regular applications and is mostly used by debuggers such as **gdb** and **strace**.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<u>4.8 Uninstall or Disable Unnecessary Services on Enterprise Assets and Software</u> Uninstall or disable unnecessary services on enterprise assets and software, such as an unused file sharing service, web application module, or service function.			
v7	<u>9.2 Ensure Only Approved Ports, Protocols and Services Are Running</u> Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1055, T1055.008	TA0005	M1040

1.5.3 Ensure core dumps are restricted (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

A core dump is the memory of an executable program. It is generally used to determine why a program aborted. It can also be used to glean confidential information from a core file. The system provides the ability to set a soft limit for core dumps, but this can be overridden by the user.

Rationale:

Setting a hard limit on core dumps prevents users from overriding the soft variable. If core dumps are required, consider setting limits for user groups (see `limits.conf(5)`). In addition, setting the `fs.suid_dumpable` variable to 0 will prevent setuid programs from dumping core.

Audit:

Run the following command and verify output matches:

```
# grep -Ps -- '^h*\*\h+hard\h+core\h+0\b' /etc/security/limits.conf
/etc/security/limits.d/*
* hard core 0
```

Run the following script to verify `fs.suid_dumpable = 0`:

Run the following script to verify the following kernel parameter is set in the running configuration and correctly loaded from a kernel parameter configuration file:

- `fs.suid_dumpable` is set to 0

Note: kernel parameters are loaded by file and parameter order precedence. The following script observes this precedence as part of the auditing procedure. The parameters being checked may be set correctly in a file. If that file is superseded, the parameter is overridden by an incorrect setting later in that file, or in a canonically later file, that "correct" setting will be ignored both by the script and by the system during a normal kernel parameter load sequence.

```

#!/usr/bin/env bash

{
    a_output=(); a_output2=(); a_parlist=("fs.suid_dumpable=0")
    l_ufwscf="$([ -f /etc/default/ufw ] && awk -F= '/^\s*IPT_SYSCTL=/ {print $2}' /etc/default/ufw)"
    f_kernel_parameter_chk()
    {
        l_running_parameter_value="$(sysctl "$l_parameter_name" | awk -F= '{print $2}' | xargs)" # Check running configuration
        if grep -Pq -- '\b'"$l_parameter_value"' \b' <<<
"$l_running_parameter_value"; then
            a_output+=(" - \"$l_parameter_name\" is correctly set to
\"$l_running_parameter_value\"")
            "      in the running configuration")
        else
            a_output2+=(" - \"$l_parameter_name\" is incorrectly set to
\"$l_running_parameter_value\"") \
            "      in the running configuration" \
            "      and should have a value of: \"$l_value_out\"")
        fi
        unset A_out; declare -A A_out # Check durable setting (files)
        while read -r l_out; do
            if [ -n "$l_out" ]; then
                if [[ $l_out =~ ^\s*# ]]; then
                    l_file="$l_out// # /}"
                else
                    l_kpar="$(awk -F= '{print $1}' <<< "$l_out" | xargs)"
                    [ "$l_kpar" = "$l_parameter_name" ] &&
A_out+=([ "$l_kpar" ]="$l_file")
                fi
            fi
            done < <("$l_systemdsysctl" --cat-config | grep -Po
'^\h*([\^#\n\r]+|\#\h*\[/[\^#\n\r\h]+\.\conf\b)')
            if [ -n "$l_ufwscf" ]; then # Account for systems with UFW (Not covered
by systemd-sysctl --cat-config)
                l_kpar="$(grep -Po "^\h*$l_parameter_name\b" "$l_ufwscf" | xargs)"
                l_kpar="$l_kpar//\./."
                [ "$l_kpar" = "$l_parameter_name" ] &&
A_out+=([ "$l_kpar" ]="$l_ufwscf")
            fi
            if (( ${#A_out[@]} > 0 )); then # Assess output from files and generate
output
                while IFS="" read -r l_fkpname l_file_parameter_value; do
                    l_fkpname="$l_fkpname// /}";
                    l_file_parameter_value="$l_file_parameter_value// /}"
                    if grep -Pq -- '\b'"$l_parameter_value"' \b' <<<
"$l_file_parameter_value"; then
                        a_output+=(" - \"$l_parameter_name\" is correctly set to
\"$l_file_parameter_value\"") \
                        "      in \"$(printf '%s' "${A_out[@]}")\"")
                    else
                        a_output2+=(" - \"$l_parameter_name\" is incorrectly set to
\"$l_file_parameter_value\"") \
                        "      in \"$(printf '%s' "${A_out[@]}")\"") \
                        "      and should have a value of: \"$l_value_out\"")
                    fi
                done
            fi
        }
    }
}

```

```

done < <(grep -Po -- "\h*$l_parameter_name\h*=\h*\H+"
"${A_out[@]}")
else
a_output2+=(" - \"$l_parameter_name\" is not set in an included
file" \
    ** Note: \"$l_parameter_name\" May be set in a file that's
ignored by load procedure **)
fi
}
l_systemdsysctl="$(readlink -f /lib/systemd/systemd-sysctl)"
while IFS="=" read -r l_parameter_name l_parameter_value; do # Assess and
check parameters
l_parameter_name="${l_parameter_name// /}";
l_parameter_value="${l_parameter_value// /}"
l_value_out="${l_parameter_value//-/ through }";
l_value_out="${l_value_out//|/ or }"
l_value_out="$(tr -d '(){}' <<< "$l_value_out")"
f_kernel_parameter_chk
done < <(printf '%s\n' "${a_parlist[@]}")
if [ "${#a_output2[@]}" -le 0 ]; then
printf '%s\n' "" "- Audit Result:" " ** PASS **" "${a_output[@]}" ""
else
printf '%s\n' "" "- Audit Result:" " ** FAIL **" " - Reason(s) for
audit failure:" "${a_output2[@]}"
[ "${#a_output[@]}" -gt 0 ] && printf '%s\n' "" "- Correctly set:"
"${a_output[@]}" ""
fi
}

```

Run the following command to check if systemd-coredump is installed:

```
# systemctl list-unit-files | grep coredump
```

if anything is returned systemd-coredump is installed

Remediation:

Add the following line to `/etc/security/limits.conf` or a `/etc/security/limits.d/*` file:

```
* hard core 0
```

Set the following parameter in `/etc/sysctl.conf` or a file in `/etc/sysctl.d/` ending in `.conf`:

- `fs.suid_dumpable = 0`

Example:

```
# printf "\n%s" "fs.suid_dumpable = 0" >> /etc/sysctl.d/60-fs_sysctl.conf
```

Run the following command to set the active kernel parameter:

```
# sysctl -w fs.suid_dumpable=0
```

Note: If these settings appear in a canonically later file, or later in the same file, these settings will be overwritten

-IF- `systemd-coredump` is installed:

edit `/etc/systemd/coredump.conf` and add/modify the following lines:

```
Storage=none
ProcessSizeMax=0
```

Run the command:

```
systemctl daemon-reload
```

References:

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1005, T1005.000	TA0007	M1057

1.5.4 Ensure prelink is not installed (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

prelink is a program that modifies ELF shared libraries and ELF dynamically linked binaries in such a way that the time needed for the dynamic linker to perform relocations at startup significantly decreases.

Rationale:

The prelinking feature can interfere with the operation of AIDE, because it changes binaries. Prelinking can also increase the vulnerability of the system if a malicious user is able to compromise a common library such as libc.

Audit:

Verify **prelink** is not installed:

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

Nothing should be returned.

Remediation:

Run the following command to restore binaries to normal:

```
# prelink -ua
```

Uninstall **prelink** using the appropriate package manager or manual installation:

```
# apt purge prelink
```

References:

1. NIST SP 800-53 Rev. 5: CM-6, CM-1, CM-3

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.14 <u>Log Sensitive Data Access</u> Log sensitive data access, including modification and disposal.			●
v7	14.9 <u>Enforce Detail Logging for Access or Changes to Sensitive Data</u> Enforce detailed audit logging for access to sensitive data or changes to sensitive data (utilizing tools such as File Integrity Monitoring or Security Information and Event Monitoring).			●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1055, T1055.009, T1065, T1065.001	TA0002	M1050

1.5.5 Ensure Automatic Error Reporting is not enabled (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The Apport Error Reporting Service automatically generates crash reports for debugging

Rationale:

Apport collects potentially sensitive data, such as core dumps, stack traces, and log files. They can contain passwords, credit card numbers, serial numbers, and other private material.

Audit:

Run the following command to verify that the Apport Error Reporting Service is not enabled:

```
# dpkg-query -s apport &> /dev/null && grep -Psi --  
'^\h*enabled\h*=\h*[\^0]\b' /etc/default/apport
```

Nothing should be returned

Run the following command to verify that the apport service is not active:

```
# systemctl is-active apport.service | grep '^active'
```

Nothing should be returned

Remediation:

Edit `/etc/default/apport` and add or edit the enabled parameter to equal `0`:

```
enabled=0
```

Run the following commands to stop and mask the apport service

```
# systemctl stop apport.service  
# systemctl mask apport.service
```

- OR -





Run the following command to remove the apport package:

```
# apt purge apport
```

Default Value:

```
enabled=1
```

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<u>4.8 Uninstall or Disable Unnecessary Services on Enterprise Assets and Software</u> Uninstall or disable unnecessary services on enterprise assets and software, such as an unused file sharing service, web application module, or service function.			
v7	<u>9.2 Ensure Only Approved Ports, Protocols and Services Are Running</u> Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system.			

1.6 Configure Command Line Warning Banners

Presenting a warning message prior to the normal user login may assist in the prosecution of trespassers on the computer system. Changing some of these login banners also has the side effect of hiding OS version information and other detailed system information from attackers attempting to target specific exploits at a system.

Guidelines published by the US Department of Defense require that warning messages include at least the name of the organization that owns the system, the fact that the system is subject to monitoring and that such monitoring is in compliance with local statutes, and that use of the system implies consent to such monitoring. It is important that the organization's legal counsel review the content of all messages before any system modifications are made, as these warning messages are inherently site-specific. More information (including citations of relevant case law) can be found at <http://www.justice.gov/criminal/cybercrime/>

The `/etc/motd`, `/etc/issue`, and `/etc/issue.net` files govern warning banners for standard command line logins for both local and remote users.

Note: The text provided in the remediation actions for these items is intended as an example only. Please edit to include the specific text for your organization as approved by your legal department.

1.6.1 Ensure message of the day is configured properly (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The contents of the `/etc/motd` file are displayed to users after login and function as a message of the day for authenticated users.

Unix-based systems have typically displayed information about the OS release and patch level upon logging in to the system. This information can be useful to developers who are developing software for a particular OS platform. If `mingetty(8)` supports the following options, they display operating system information: `\m` - machine architecture `\r` - operating system release `\s` - operating system name `\v` - operating system version

Rationale:

Warning messages inform users who are attempting to login to the system of their legal status regarding the system and must include the name of the organization that owns the system and any monitoring policies that are in place. Displaying OS and patch level information in login banners also has the side effect of providing detailed system information to attackers attempting to target specific exploits of a system. Authorized users can easily get this information by running the "`uname -a`" command once they have logged in.

Audit:

Run the following command and verify that the contents match site policy:

```
# cat /etc/motd
```

Run the following command and verify no results are returned:

```
# grep -E -i "(\\v|\\r|\\m|\\s|$(grep '^ID=' /etc/os-release | cut -d= -f2 | sed -e 's/"/\\g'))" /etc/motd
```

Remediation:

Edit the `/etc/motd` file with the appropriate contents according to your site policy, remove any instances of `\m` , `\r` , `\s` , `\v` or references to the `OS platform`

- OR -

- IF - the `motd` is not used, this file can be removed.

Run the following command to remove the `motd` file:

```
# rm /etc/motd
```

References:

1. NIST SP 800-53 Rev. 5: CM-6, CM-1, CM-3

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1082, T1082.000, T1592, T1592.004	TA0007	

1.6.2 Ensure local login warning banner is configured properly (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The contents of the `/etc/issue` file are displayed to users prior to login for local terminals.

Unix-based systems have typically displayed information about the OS release and patch level upon logging in to the system. This information can be useful to developers who are developing software for a particular OS platform. If `mingetty(8)` supports the following options, they display operating system information: `\m` - machine architecture `\r` - operating system release `\s` - operating system name `\v` - operating system version - or the operating system's name

Rationale:

Warning messages inform users who are attempting to login to the system of their legal status regarding the system and must include the name of the organization that owns the system and any monitoring policies that are in place. Displaying OS and patch level information in login banners also has the side effect of providing detailed system information to attackers attempting to target specific exploits of a system. Authorized users can easily get this information by running the "`uname -a`" command once they have logged in.

Audit:

Run the following command and verify that the contents match site policy:

```
# cat /etc/issue
```

Run the following command and verify no results are returned:

```
# grep -E -i "(\\v|\\r|\\m|\\s|$(grep '^ID=' /etc/os-release | cut -d= -f2 | sed -e 's/"/\\g'))" /etc/issue
```

Remediation:

Edit the `/etc/issue` file with the appropriate contents according to your site policy, remove any instances of `\m`, `\r`, `\s`, `\v` or references to the **OS platform**

Example:

```
# echo "Authorized users only. All activity may be monitored and reported." > /etc/issue
```

References:

1. NIST SP 800-53 Rev. 5: CM-6, CM-1, CM-3

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1082, T1082.000, T1592, T1592.004	TA0007	

1.6.3 Ensure remote login warning banner is configured properly (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The contents of the `/etc/issue.net` file are displayed to users prior to login for remote connections from configured services.

Unix-based systems have typically displayed information about the OS release and patch level upon logging in to the system. This information can be useful to developers who are developing software for a particular OS platform. If `mingetty(8)` supports the following options, they display operating system information: `\m` - machine architecture `\r` - operating system release `\s` - operating system name `\v` - operating system version

Rationale:

Warning messages inform users who are attempting to login to the system of their legal status regarding the system and must include the name of the organization that owns the system and any monitoring policies that are in place. Displaying OS and patch level information in login banners also has the side effect of providing detailed system information to attackers attempting to target specific exploits of a system. Authorized users can easily get this information by running the "`uname -a`" command once they have logged in.

Audit:

Run the following command and verify that the contents match site policy:

```
# cat /etc/issue.net
```

Run the following command and verify no results are returned:

```
# grep -E -i "(\\v|\\r|\\m|\\s|$(grep '^ID=' /etc/os-release | cut -d= -f2 | sed -e 's/"/\\g'))" /etc/issue.net
```

Remediation:

Edit the `/etc/issue.net` file with the appropriate contents according to your site policy, remove any instances of `\m`, `\r`, `\s`, `\v` or references to the **OS platform**

Example:

```
# echo "Authorized users only. All activity may be monitored and reported." > /etc/issue.net
```

References:

1. NIST SP 800-53 Rev. 5: CM-6, CM-1, CM-3

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1018, T1018.000, T1082, T1082.000, T1592, T1592.004	TA0007	

1.6.4 Ensure access to /etc/motd is configured (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The contents of the `/etc/motd` file are displayed to users after login and function as a message of the day for authenticated users.

Rationale:

- **IF** - the `/etc/motd` file does not have the correct access configured, it could be modified by unauthorized users with incorrect or misleading information.

Audit:

Run the following command and verify that if `/etc/motd` exists, **Access** is **644** or more restrictive, **Uid** and **Gid** are both `0/root`:

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

Access: (0644/-rw-r--r--)  Uid: ( 0/ root) Gid: ( 0/ root)
-- OR --
Nothing is returned
```

Remediation:

Run the following commands to set mode, owner, and group on `/etc/motd`:

```
# chown root:root $(readlink -e /etc/motd)
# chmod u-x,go-wx $(readlink -e /etc/motd)
```

- OR -







Run the following command to remove the `/etc/motd` file:

```
# rm /etc/motd
```

References:

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

CIS Controls:

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

MITRE ATT&CK Mappings:

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

1.6.5 Ensure access to /etc/issue is configured (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The contents of the `/etc/issue` file are displayed to users prior to login for local terminals.

Rationale:

- **IF** - the `/etc/issue` file does not have the correct access configured, it could be modified by unauthorized users with incorrect or misleading information.

Audit:

Run the following command and verify **Access** is **644** or more restrictive and **Uid** and **Gid** are both **0/root**:

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

Remediation:

Run the following commands to set mode, owner, and group on `/etc/issue`:

```
# chown root:root $(readlink -e /etc/issue)
# chmod u-x,go-wx $(readlink -e /etc/issue)
```







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

MITRE ATT&CK Mappings:

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

1.6.6 Ensure access to /etc/issue.net is configured (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The contents of the `/etc/issue.net` file are displayed to users prior to login for remote connections from configured services.

Rationale:

- **IF** - the `/etc/issue.net` file does not have the correct access configured, it could be modified by unauthorized users with incorrect or misleading information.

Audit:

Run the following command and verify **Access** is **644** or more restrictive and **Uid** and **Gid** are both **0/root**:

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

Remediation:

Run the following commands to set mode, owner, and group on `/etc/issue.net`:

```
# chown root:root $(readlink -e /etc/issue.net)
# chmod u-x,go-wx $(readlink -e /etc/issue.net)
```







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

MITRE ATT&CK Mappings:

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

1.7 Configure GNOME Display Manager

The GNOME Display Manager (GDM) is a program that manages graphical display servers and handles graphical user logins.

This subsection requires user profiles to already exist on the system. A profile is a list of configuration databases.

Sample profile:

```
user-db:user
system-db:local
system-db:site
```

Configuring a single user and multiple system databases allows for layering of preferences. Settings from the user database file take precedence over the settings in the local database file, and the local database file in turn takes precedence over the site database file.

Note:

- **- IF -** GDM is not installed on the system, this section can be skipped
- The Remediation Procedure commands in this section **MUST** be done from a command window on a graphical desktop or an error will be returned.

1.7.1 Ensure GDM is removed (Automated)

Profile Applicability:

- Level 2 - Server

Description:

The GNOME Display Manager (GDM) is a program that manages graphical display servers and handles graphical user logins.

Rationale:

If a Graphical User Interface (GUI) is not required, it should be removed to reduce the attack surface of the system.

Impact:

Removing the GNOME Display manager will remove the Graphical User Interface (GUI) from the system.

Audit:

Run the following command and verify **gdm3** is not installed:

```
# dpkg-query -W -f='${binary:Package}\t${Status}\t${db:Status-Status}\n' gdm3
gdm3          unknown ok not-installed      not-installed
```

Remediation:





Run the following commands to uninstall **gdm3** and remove unused dependencies:

```
# apt purge gdm3
# apt autoremove gdm3
```

References:

1. NIST SP 800-53 Rev. 5: CM-11

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<u>4.8 Uninstall or Disable Unnecessary Services on Enterprise Assets and Software</u> Uninstall or disable unnecessary services on enterprise assets and software, such as an unused file sharing service, web application module, or service function.			
v7	<u>9.2 Ensure Only Approved Ports, Protocols and Services Are Running</u> Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1543, T1543.002	TA0002	M1033

1.7.2 Ensure GDM login banner is configured (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

GDM is the GNOME Display Manager which handles graphical login for GNOME based systems.

Rationale:

Warning messages inform users who are attempting to login to the system of their legal status regarding the system and must include the name of the organization that owns the system and any monitoring policies that are in place.

Audit:

Run the following commands to verify that the text banner on the login screen is enabled and set:

```
# gsettings get org.gnome.login-screen banner-message-enable  
true  
# gsettings get org.gnome.login-screen banner-message-text  
'Authorized uses only. All activity may be monitored and reported'
```

Remediation:

- **IF** - A user profile is already created run the following commands to set and enable the text banner message on the login screen:

```
# gsettings set org.gnome.login-screen banner-message-text 'Authorized uses only. All activity may be monitored and reported'
# gsettings set org.gnome.login-screen banner-message-enable true
```

Note:

- **banner-message-text** may be set in accordance with local site policy
- **gsettings** commands in this section **MUST** be done from a command window on a graphical desktop or an error will be returned.
- The system must be restarted after all **gsettings** configurations have been set in order for CIS-CAT Assessor to appropriately assess.

- **OR/IF** - A user profile does not exist:

1. Create or edit the gdm profile in the **/etc/dconf/profile/gdm** with the following lines:

```
user-db:user
system-db:gdm
file-db:/usr/share/gdm/greeter-dconf-defaults
```

Note: gdm is the name of a dconf database.

2. Create a gdm keyfile for machine-wide settings in **/etc/dconf/db/gdm.d/01-banner-message:**

```
[org/gnome/login-screen]
banner-message-enable=true
banner-message-text='Type the banner message here.'
```

3. Update the system databases

```
# dconf update
```

Note:

- Users must log out and back in again before the system-wide settings take effect.
- There is no character limit for the banner message. gnome-shell autodetects longer stretches of text and enters two column mode.
- The banner message cannot be read from an external file.

Default Value:

disabled

References:

1. <https://help.gnome.org/admin/system-admin-guide/stable/login-banner.html.en>
2. NIST SP 800-53 Rev. 5: CM-1,CM-2, CM-6, CM-7, IA-5

Additional Information:

Additional options and sections may appear in the `/etc/dconf/db/gdm.d/01-banner-message` file.

If a different GUI login service is in use, consult your documentation and apply an equivalent banner.

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1078, T1078.001, T1078.002, T1078.003, T1087, T1087.001, T1087.002	TA0007	M1028

1.7.3 Ensure GDM disable-user-list option is enabled (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

GDM is the GNOME Display Manager which handles graphical login for GNOME based systems.

The **disable-user-list** option controls if a list of users is displayed on the login screen

Rationale:

Displaying the user list eliminates half of the Userid/Password equation that an unauthorized person would need to log on.

Audit:

Run the following command and to verify that the **disable-user-list** option is enabled:

```
# gsettings get org.gnome.login-screen disable-user-list
true
```

Remediation:

- **IF** - A user profile exists run the following command to enable the **disable-user-list**:

```
# gsettings set org.gnome.login-screen disable-user-list true
```

Note:

- **gsettings** commands in this section **MUST** be done from a command window on a graphical desktop or an error will be returned.
- The system must be restarted after all **gsettings** configurations have been set in order for CIS-CAT Assessor to appropriately assess.

- **OR/IF** - A user profile does not exist:

1. Create or edit the gdm profile in **/etc/dconf/profile/gdm** with the following lines:

```
user-db:user
system-db:gdm
file-db:/usr/share/gdm/greeter-dconf-defaults
```

Note: gdm is the name of a dconf database.

2. Create a gdm keyfile for machine-wide settings in **/etc/dconf/db/gdm.d/00-login-screen**:

```
[org/gnome/login-screen]
# Do not show the user list
disable-user-list=true
```

3. Update the system databases:

```
# dconf update
```

Note: When the user profile is created or changed, the user will need to log out and log in again before the changes will be applied.

Default Value:

false

References:

1. <https://help.gnome.org/admin/system-admin-guide/stable/login-userlist-disable.html.en>
2. NIST SP 800-53 Rev. 5: CM-1, CM-2, CM-6, CM-7, IA-5

Additional Information:

If a different GUI login service is in use and required on the system, consult your documentation to disable displaying the user list

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1078, T1078.001, T1078.002, T1078.003, T1087, T1087.001, T1087.002	TA0007	M1028

1.7.4 Ensure GDM screen locks when the user is idle (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

GNOME Desktop Manager can make the screen lock automatically whenever the user is idle for some amount of time.

Rationale:

Setting a lock-out value reduces the window of opportunity for unauthorized user access to another user's session that has been left unattended.

Audit:

Run the following commands to verify that the screen locks when the user is idle:

```
# gsettings get org.gnome.desktop.screensaver lock-delay
uint32 5
# gsettings get org.gnome.desktop.session idle-delay
uint32 900
```

Notes:

- **lock-delay=uint32 {n}** - should be 5 seconds or less and follow local site policy
- **idle-delay=uint32 {n}** - Should be 900 seconds (15 minutes) or less, not 0 (disabled) and follow local site policy

Remediation:

- **IF** - A user profile is already created run the following commands to enable screen locks when the user is idle:

```
# gsettings set org.gnome.desktop.screensaver lock-delay 5
# gsettings set org.gnome.desktop.session idle-delay 900
```

Note:

- **gsettings** commands in this section **MUST** be done from a command window on a graphical desktop or an error will be returned.
- The system must be restarted after all **gsettings** configurations have been set in order for CIS-CAT Assessor to appropriately assess.

- **OR/IF**- A user profile does not exist:

1. Create or edit the user profile in the **/etc/dconf/profile/** and verify it includes the following:

```
user-db:user
system-db:{NAME_OF_DCONF_DATABASE}
```

Note: **local** is the name of a dconf database used in the examples.

2. Create the directory **/etc/dconf/db/local.d/** if it doesn't already exist:
3. Create the key file **/etc/dconf/db/local.d/00-screensaver** to provide information for the **local** database:

Example key file:

```
# Specify the dconf path
[org/gnome/desktop/session]

# Number of seconds of inactivity before the screen goes blank
# Set to 0 seconds if you want to deactivate the screensaver.
idle-delay=uint32 180

# Specify the dconf path
[org/gnome/desktop/screensaver]

# Number of seconds after the screen is blank before locking the screen
lock-delay=uint32 0
```

Note: You must include the uint32 along with the integer key values as shown.

4. Run the following command to update the system databases:







```
# dconf update
```

5. Users must log out and back in again before the system-wide settings take effect.

References:

1. <https://help.gnome.org/admin/system-admin-guide/stable/desktop-lockscreens.html.en>

CIS Controls:

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1461	TA0027	M1012

1.7.5 Ensure GDM screen locks cannot be overridden (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

GNOME Desktop Manager can lock down specific settings by using the lockdown mode in dconf to prevent users from changing specific settings.

To lock down a dconf key or subpath, create a locks subdirectory in the keyfile directory. The files inside this directory contain a list of keys or subpaths to lock. Just as with the keyfiles, you may add any number of files to this directory.

Rationale:

Setting a lock-out value reduces the window of opportunity for unauthorized user access to another user's session that has been left unattended.

Without locking down the system settings, user settings take precedence over the system settings.

Audit:

Run the following script to verify that the screen lock cannot be overridden:

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

{
  a_output=() a_output2=()
  f_check_setting()
  {
    grep -Psrilq -- "^h*$2\b" /etc/dconf/db/local.d/locks/* && \
    echo "- \"$3\" is locked" || echo "- \"$3\" is not locked or not set"
  }
  declare -A settings=(
    ["idle-delay"]="/org/gnome/desktop/session/idle-delay"
    ["lock-delay"]="/org/gnome/desktop/screensaver/lock-delay"
  )
  for setting in "${!settings[@]}"; do
    result=$(f_check_setting "$setting" "${settings[$setting]}" "$setting")
    if [[ $result == *"is not locked"* || $result == *"not set to false"*
  ]]; then
      a_output2+=("$result")
    else
      a_output+=("$result")
    fi
  done
  printf '%s\n' "" "- Audit Result:"
  if [ "${#a_output2[@]}" -gt 0 ]; then
    printf '%s\n' " ** FAIL **" " - Reason(s) for audit failure:"
    "${a_output2[@]}"
    [ "${#a_output[@]}" -gt 0 ] && printf '%s\n' "" "- Correctly set:"
    "${a_output[@]}"
  else
    printf '%s\n' " ** PASS **" "${a_output[@]}"
  fi
}
```

Remediation:

1. To prevent the user from overriding these settings, create the file **/etc/dconf/db/local.d/locks/00-screensaver** with the following content:

```
# Lock desktop screensaver settings
/org/gnome/desktop/session/idle-delay
/org/gnome/desktop/screensaver/lock-delay
```

2. Update the system databases:

```
# dconf update
```







Note:

- A user profile must exist in order to apply locks.
- Users must log out and back in again before the system-wide settings take effect.

References:

1. <https://help.gnome.org/admin/system-admin-guide/stable/desktop-locksreen.html.en>
2. <https://help.gnome.org/admin/system-admin-guide/stable/dconf-lockdown.html.en>
3. NIST SP 800-53 Rev. 5: CM-11

CIS Controls:

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

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1456	TA0027	M1001

1.7.6 Ensure GDM automatic mounting of removable media is disabled (Automated)

Profile Applicability:

- Level 1 - Server
- Level 2 - Workstation

Description:

By default GNOME automatically mounts removable media when inserted as a convenience to the user.

Rationale:

With automounting enabled anyone with physical access could attach a USB drive or disc and have its contents available in system even if they lacked permissions to mount it themselves.

Impact:

The use of portable hard drives is very common for workstation users. If your organization allows the use of portable storage or media on workstations and physical access controls to workstations is considered adequate there is little value add in turning off automounting.

Audit:

Run the following commands to verify automatic mounting is disabled:

```
# gsettings get org.gnome.desktop.media-handling automount
false
# gsettings get org.gnome.desktop.media-handling automount-open
false
```


Remediation:

- **IF** - A user profile exists run the following commands to ensure automatic mounting is disabled:

```
# gsettings set org.gnome.desktop.media-handling automount false
# gsettings set org.gnome.desktop.media-handling automount-open false
```

Note:

- **gsettings** commands in this section **MUST** be done from a command window on a graphical desktop or an error will be returned.
- The system must be restarted after all **gsettings** configurations have been set in order for CIS-CAT Assessor to appropriately assess.

- **OR/IF** - A user profile does not exist:

1. Create a file `/etc/dconf/db/local.d/00-media-automount` with following content:

```
[org/gnome/desktop/media-handling]
automount=false
automount-open=false
```

2. After creating the file, apply the changes using below command :







```
# dconf update
```

Note: Users must log out and back in again before the system-wide settings take effect.

References:

1. <https://access.redhat.com/solutions/20107>
2. NIST SP 800-53 Rev. 5: CM-1, CM-2, CM-6, CM-7, IA-5

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<u>10.3 Disable Autorun and Autoplay for Removable Media</u> Disable autorun and autoplay auto-execute functionality for removable media.			
v7	<u>8.5 Configure Devices Not To Auto-run Content</u> Configure devices to not auto-run content from removable media.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1091, T1091.000	TA0008	M1042

1.7.7 Ensure GDM disabling automatic mounting of removable media is not overridden (Automated)

Profile Applicability:

- Level 1 - Server
- Level 2 - Workstation

Description:

By default GNOME automatically mounts removable media when inserted as a convenience to the user.

By using the lockdown mode in dconf, you can prevent users from changing specific settings. To lock down a dconf key or subpath, create a locks subdirectory in the keyfile directory. The files inside this directory contain a list of keys or subpaths to lock. Just as with the keyfiles, you may add any number of files to this directory.

Rationale:

With automounting enabled anyone with physical access could attach a USB drive or disc and have its contents available in system even if they lacked permissions to mount it themselves.

Impact:

The use of portable hard drives is very common for workstation users

Audit:

Run the following script to verify automatic mounting of removable media is not overridden and correctly configured in a configuration file:

- **automount=false**
- **automount-open=false**

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

{
  a_output=() a_output2=()
  check_setting()
  {
    grep -Psrilq "^h*$1\h*=\h*false\b" /etc/dconf/db/local.d/locks/* 2>
/dev/null && \
    echo "- \"$3\" is locked and set to false" || echo "- \"$3\" is not
locked or not set to false"
  }
  declare -A settings=(
    ["automount"]="org/gnome/desktop/media-handling"
    ["automount-open"]="org/gnome/desktop/media-handling"
  )
  for setting in "${!settings[@]}"; do
    result=$(check_setting "$setting" "${settings[$setting]}" "$setting")
    if [[ $result == *"is not locked"* || $result == *"not set to false"*
]]; then
      a_output2+=("$result")
    else
      a_output+=("$result")
    fi
  done
  printf '%s\n' "" "- Audit Result:"
  if [ "${#a_output2[@]}" -gt 0 ]; then
    printf '%s\n' "  ** FAIL **" " - Reason(s) for audit failure:"
    "${a_output2[@]}"
    [ "${#a_output[@]}" -gt 0 ] && printf '%s\n' "" "- Correctly set:"
    "${a_output[@]}"
  else
    printf '%s\n' "  ** PASS **" "${a_output[@]}"
  fi
}
```

Remediation:

1. To prevent the user from overriding these settings, create the file `/etc/dconf/db/local.d/locks/00-media-automount` with the following content:

```
[org/gnome/desktop/media-handling]
automount=false
automount-open=false
```

2. Update the systems databases:

```
# dconf update
```

Note:

- A user profile must exist in order to apply locks.
- Users must log out and back in again before the system-wide settings take effect.

References:

1. <https://help.gnome.org/admin/system-admin-guide/stable/dconf-lockdown.html.en>
2. NIST SP 800-53 Rev. 5: CM-1, CM-2, CM-6, CM-7, IA-5
3. <https://manpages.ubuntu.com/manpages/trusty/man1/gsettings.1.html>
4. <https://access.redhat.com/solutions/20107>

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1091, T1091.000	TA0001, TA0008	M1042

1.7.8 Ensure GDM autorun-never is enabled (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The **autorun-never** setting allows the GNOME Desktop Display Manager to disable autorun through GDM.

Rationale:

Malware on removable media may taking advantage of Autorun features when the media is inserted into a system and execute.

Audit:

Run the following command to verify that **autorun-never** is set to **true** for GDM:

```
# gsettings get org.gnome.desktop.media-handling autorun-never
true
```

Remediation:

- **IF** - A user profile exists run the following command to set **autorun-never** to **true** for GDM users:

```
# gsettings set org.gnome.desktop.media-handling autorun-never true
```

Note:

- **gsettings** commands in this section **MUST** be done from a command window on a graphical desktop or an error will be returned.
- The system must be restarted after all **gsettings** configurations have been set in order for CIS-CAT Assessor to appropriately assess.

- **OR/IF** - A user profile does not exist:

1. create the file **/etc/dconf/db/local.d/locks/00-media-autorun** with the following content:

```
[org/gnome/desktop/media-handling]
autorun-never=true
```

2. Update the systems databases:

```
# dconf update
```

Note: Users must log out and back in again before the system-wide settings take effect.







Default Value:

false

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	<u>10.3 Disable Autorun and Autoplay for Removable Media</u> Disable autorun and autoplay auto-execute functionality for removable media.			
v7	<u>8.5 Configure Devices Not To Auto-run Content</u> Configure devices to not auto-run content from removable media.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1091, T1091.000	TA0001, TA0008	M1042

1.7.9 Ensure GDM autorun-never is not overridden (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The autorun-never setting allows the GNOME Desktop Display Manager to disable autorun through GDM.

By using the lockdown mode in dconf, you can prevent users from changing specific settings.

To lock down a dconf key or subpath, create a locks subdirectory in the keyfile directory. The files inside this directory contain a list of keys or subpaths to lock. Just as with the keyfiles, you may add any number of files to this directory.

Rationale:

Malware on removable media may taking advantage of Autorun features when the media is inserted into a system and execute.

Audit:

Run the following script to verify that **autorun-never=true** cannot be overridden:

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

{
    # Function to check and report if a specific setting is locked and set to
    true
    check_setting() {
        grep -Psrlq "^\h*$1\h*=\h*true\b" /etc/dconf/db/local.d/locks/* 2>
/dev/null && echo "- \"$3\" is locked and set to false" || echo "- \"$3\" is
not locked or not set to false"
    }
    # Array of settings to check
    declare -A settings=(["autorun-never"]="org/gnome/desktop/media-
handling")
    # Check GNOME Desktop Manager configurations
    l_output=() l_output2=()
    for setting in "${!settings[@]}"; do
        result=$(check_setting "$setting")
        l_output+=("$result")
        if [[ $result == *"is not locked"* || $result == *"not set to true"*
]]; then
            l_output2+=("$result")
        fi
    done
    # Report results
    if [ ${#l_output2[@]} -ne 0 ]; then
        printf '%s\n' "- Audit Result:" " ** FAIL **"
        printf '%s\n' "- Reason(s) for audit failure:"
        for msg in "${l_output2[@]}"; do
            printf '%s\n' "$msg"
        done
    else
        printf '%s\n' "- Audit Result:" " ** PASS **"
    fi
}
```

Remediation:

1. To prevent the user from overriding these settings, create the file `/etc/dconf/db/local.d/locks/00-media-autorun` with the following content:

```
[org/gnome/desktop/media-handling]
autorun-never=true
```

2. Update the systems databases:

```
# dconf update
```







Note:

- A user profile must exist in order to apply locks.
- Users must log out and back in again before the system-wide settings take effect.

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	10.3 <u>Disable Autorun and Autoplay for Removable Media</u> Disable autorun and autoplay auto-execute functionality for removable media.			
v7	8.5 <u>Configure Devices Not To Auto-run Content</u> Configure devices to not auto-run content from removable media.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1091, T1091.000	TA0001, TA0008	M1028

1.7.10 Ensure XDMCP is not enabled (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

X Display Manager Control Protocol (XDMCP) is designed to provide authenticated access to display management services for remote displays

Rationale:

XDMCP is inherently insecure.

- XDMCP is not a ciphered protocol. This may allow an attacker to capture keystrokes entered by a user
- XDMCP is vulnerable to man-in-the-middle attacks. This may allow an attacker to steal the credentials of legitimate users by impersonating the XDMCP server.

Audit:

Run the following script and verify the output:

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

{
  while IFS= read -r l_file; do
    awk '/\[xdmcp\]/{ f = 1;next } /\[/{ f = 0 } f {if
(/^\s*Enable\s*=\s*true/) print "The file: \"'$l_file'\" includes: \"' $0
\" in the \"[xdmcp]\" block\"}' \"$l_file"
    done <<(grep -Psil -- '^\h*\[xdmcp\]'
/etc/{gdm3,gdm}/{custom,daemon}.conf)
  }
}
```

Nothing should be returned

Remediation:

Edit all files returned by the audit and remove or comment out the `Enable=true` line in the `[xdmcp]` block:

Example file:

```
# GDM configuration storage
#
# See /usr/share/gdm/gdm.schemas for a list of available options.

[daemon]
# Uncomment the line below to force the login screen to use Xorg
#WaylandEnable=false

# Enabling automatic login
# AutomaticLoginEnable = true
# AutomaticLogin = user1

# Enabling timed login
# TimedLoginEnable = true
# TimedLogin = user1
# TimedLoginDelay = 10

[security]

[xdmcp]
# Enable=true <- **This line should be removed or commented out**

[chooser]

[debug]
# Uncomment the line below to turn on debugging
# More verbose logs
# Additionally lets the X server dump core if it crashes
#Enable=true
```





Default Value:

false (This is denoted by no `Enabled=` entry in the `[xdmcp]` block)

References:

1. NIST SP 800-53 Rev. 5: SI-4

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<u>4.8 Uninstall or Disable Unnecessary Services on Enterprise Assets and Software</u> Uninstall or disable unnecessary services on enterprise assets and software, such as an unused file sharing service, web application module, or service function.			
v7	<u>9.2 Ensure Only Approved Ports, Protocols and Services Are Running</u> Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system.			

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1040, T1040.000, T1056, T1056.001, T1557, T1557.000	TA0002	M1050