Security Configuration Assessment Report for

CIS-CAT Host IP Address:

CIS Ubuntu Linux 18.04 LTS Benchmark v2.0.1

No profile was selected.

Report generated by the Center for Internet Security's Configuration Assessment Tool (CIS-CAT) .
For further information, please visit The Center for Internet Security or send an e-mail to feedback@cisecurity.org.

Summary

	Pass	Fail	Error	Unkn.			
1 Initial Setup	**0**	**0**	**0**	**0**	**0.0**	**0.0**	**0%**
1.1 Filesystem Configuration	0	0	0	0	0.0	0.0	0%
1.1.1 Disable unused filesystems	0	0	0	0	0.0	0.0	0%
1.2 Configure Software Updates	0	0	0	0	0.0	0.0	0%
1.3 Configure sudo	0	0	0	0	0.0	0.0	0%
1.4 Filesystem Integrity Checking	0	0	0	0	0.0	0.0	0%
1.5 Secure Boot Settings	0	0	0	0	0.0	0.0	0%
1.6 Additional Process Hardening	0	0	0	0	0.0	0.0	0%
1.7 Mandatory Access Control	0	0	0	0	0.0	0.0	0%
1.7.1 Configure AppArmor	0	0	0	0	0.0	0.0	0%
1.8 Warning Banners	0	0	0	0	0.0	0.0	0%
1.8.1 Command Line Warning Banners	0	0	0	0	0.0	0.0	0%
2 Services	**0**	**0**	**0**	**0**	**0.0**	**0.0**	**0%**
2.1 inetd Services	0	0	0	0	0.0	0.0	0%
2.2 Special Purpose Services	0	0	0	0	0.0	0.0	0%
2.2.1 Time Synchronization	0	0	0	0	0.0	0.0	0%
2.3 Service Clients	0	0	0	0	0.0	0.0	0%
3 Network Configuration	**0**	**0**	**0**	**0**	**0.0**	**0.0**	**0%**
3.1 Network Parameters (Host Only)	0	0	0	0	0.0	0.0	0%
3.2 Network Parameters (Host and Router)	0	0	0	0	0.0	0.0	0%
3.3 TCP Wrappers	0	0	0	0	0.0	0.0	0%
3.4 Uncommon Network Protocols	0	0	0	0	0.0	0.0	0%
3.5 Firewall Configuration	0	0	0	0	0.0	0.0	0%
3.5.1 Ensure Firewall software is installed	0	0	0	0	0.0	0.0	0%
3.5.2 Configure UncomplicatedFirewall	0	0	0	0	0.0	0.0	0%
3.5.3 Configure nftables	0	0	0	0	0.0	0.0	0%
3.5.4 Configure iptables	0	0	0	0	0.0	0.0	0%
3.5.4.1 Configure IPv4 iptables	0	0	0	0	0.0	0.0	0%
3.5.4.2 Configure IPv6 ip6tables	0	0	0	0	0.0	0.0	0%
4 Logging and Auditing	**0**	**0**	**0**	**0**	**0.0**	**0.0**	**0%**
4.1 Configure System Accounting (auditd)	0	0	0	0	0.0	0.0	0%
4.1.1 Ensure auditing is enabled	0	0	0	0	0.0	0.0	0%
4.1.2 Configure Data Retention	0	0	0	0	0.0	0.0	0%
4.2 Configure Logging	0	0	0	0	0.0	0.0	0%
4.2.1 Configure rsyslog	0	0	0	0	0.0	0.0	0%
4.2.2 Configure journald	0	0	0	0	0.0	0.0	0%
5 Access, Authentication and Authorization	**0**	**0**	**0**	**0**	**0.0**	**0.0**	**0%**
5.1 Configure cron	0	0	0	0	0.0	0.0	0%
5.2 SSH Server Configuration	0	0	0	0	0.0	0.0	0%
5.3 Configure PAM	0	0	0	0	0.0	0.0	0%
5.4 User Accounts and Environment	0	0	0	0	0.0	0.0	0%
5.4.1 Set Shadow Password Suite Parameters	0	0	0	0	0.0	0.0	0%
6 System Maintenance	**0**	**0**	**0**	**0**	**0.0**	**0.0**	**0%**
6.1 System File Permissions	0	0	0	0	0.0	0.0	0%
6.2 User and Group Settings	0	0	0	0	0.0	0.0	0%
Total	**0**	**0**	**0**	**0**	**0.0**	**0.0**	**0%**

Note: Actual scores are subject to rounding errors. The sum of these values may not result in the exact overall score.

Profiles

This benchmark contains 4 profiles.No profile was selected for this assessment.

Level 1 - Server	Items in this profile intend to: • be practical and prudent; • provide a clear security benefit; and • not inhibit the utility of the technology beyond acceptable means. This profile is intended for servers. Show Profile XML
Level 2 - Server	This profile extends the "Level 1 - Server" profile. Items in this profile exhibit one or more of the following characteristics: • are intended for environments or use cases where security is paramount. • acts as defense in depth measure. • may negatively inhibit the utility or performance of the technology. This profile is intended for servers. Show Profile XML
Level 1 - Workstation	Items in this profile intend to: • be practical and prudent; • provide a clear security benefit; and • not inhibit the utility of the technology beyond acceptable means. This profile is intended for workstations. Show Profile XML
Level 2 - Workstation	This profile extends the "Level 1 - Workstation" profile. Items in this profile exhibit one or more of the following characteristics: • are intended for environments or use cases where security is paramount. • acts as defense in depth measure. • may negatively inhibit the utility or performance of the technology. This profile is intended for workstations. Show Profile XML

Assessment Results

Display Failures Only

Assessment Details

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 Configuration

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, make sure the data has been copied over to the new partition, unmount it and then remove the data from the directory that was in the old partition. Otherwise it will still consume space in the old partition that will be masked when the new filesystem is mounted. For example, if a system is in single-user mode with no filesystems mounted and the administrator adds a lot of data to the `/tmp` directory, this data will still consume space in `/` once the `/tmp` filesystem is mounted unless it is removed first.

1.1.1 Disable unused filesystems

A number of 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.

1.2 Configure Software Updates

Ubuntu Linux use apt to install and update software packages. 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. Many large enterprises prefer to test patches on a non-production system before rolling out to production.

For the purpose of this benchmark, the requirement is to ensure that a patch management system is configured and maintained. The specifics on patch update procedures are left to the organization.

1.3 Configure sudo

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

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

1.4 Filesystem Integrity Checking

AIDE is a file integrity checking tool, similar in nature to Tripwire. While it cannot prevent intrusions, it can detect unauthorized changes to configuration files by alerting when the files are changed. When setting up AIDE, decide internally what the site policy will be concerning integrity checking. Review the AIDE quick start guide and AIDE documentation before proceeding.

1.5 Secure Boot Settings

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

1.6 Additional Process Hardening

1.7 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.

Note: Apparmor is the default MAC provided with Ubuntu systems.

Note: Additional Mandatory Access Control systems to include SELinux exist. If a different Mandatory Access Control systems is used, please follow it's vendors guidance for proper implementation in place of the guidance provided in this section

1.7.1 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/

1.8 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/

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.

1.8.1 Command Line Warning Banners

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

2 Services

While applying system updates and patches helps correct known vulnerabilities, one of the best ways to protect the system against as yet unreported vulnerabilities is to disable all services that are not required for normal system operation. This prevents the exploitation of vulnerabilities discovered at a later date. If a service is not enabled, it cannot be exploited. The actions in this section of the document provide guidance on some services which can be safely disabled and under which circumstances, greatly reducing the number of possible threats to the resulting system. Additionally some services which should remain enabled but with secure configuration are covered as well as insecure service clients.

2.1 inetd Services

inetd is a super-server daemon that provides internet services and passes connections to configured services. While not commonly used inetd and any unneeded inetd based services should be disabled if possible.

2.2 Special Purpose Services

This section describes services that are installed on systems that specifically need to run these services. If any of these services are not required, it is recommended that they be disabled or deleted from the system to reduce the potential attack surface.

2.2.1 Time Synchronization

It is recommended that physical systems and virtual guests lacking direct access to the physical host's clock be configured to synchronize their time using a service such as systemd-timesyncd, chrony, or ntp.

If access to a physical host's clock is available and configured according to site policy, this section can be

skipped.

2.3 Service Clients

A number of insecure services exist. While disabling the servers prevents a local attack against these services, it is advised to remove their clients unless they are required.

Note : This should not be considered a comprehensive list of insecure service clients. You may wish to consider additions to those listed here for your environment.

3 Network Configuration

This section provides guidance on for securing the network configuration of the system through kernel parameters, access list control, and firewall settings.

3.1 Network Parameters (Host Only)

The following network parameters are intended for use if the system is to act as a host only. A system is considered host only if the system has a single interface, or has multiple interfaces but will not be configured as a router.

3.2 Network Parameters (Host and Router)

The following network parameters are intended for use on both host only and router systems. A system acts as a router if it has at least two interfaces and is configured to perform routing functions.

3.3 TCP Wrappers

3.4 Uncommon Network Protocols

The Linux kernel modules support several network protocols that are not commonly used. If these protocols are not needed, it is recommended that they be disabled in the kernel.

Note: This should not be considered a comprehensive list of uncommon network protocols, you may wish to consider additions to those listed here for your environment.

3.5 Firewall Configuration

A Host based firewall Provides defense against external and internal threats by refusing unauthorized connections, to stop intrusion and provide a strong method of access control policy.

this section is intended only to ensure the resulting firewall rules are in place, not how they are configured

3.5.1 Ensure Firewall software is installed

In order to configure Firewall protection for you system, a Firewall software package needs to be installed

3.5.2 Configure UncomplicatedFirewall

UncomplicatedFirewall (ufw) is a frontend for iptables. ufw provides a framework for managing netfilter, as well as a command-line interface for manipulating the firewall.

3.5.3 Configure nftables

nftables is a subsystem of the Linux kernel providing filtering and classification of network packets/datagrams/frames and is the successor to iptables. The biggest change with the successor nftables is its simplicity. With iptables, we have to configure every single rule and use the syntax which can be compared with normal commands. With nftables, the simpler syntax, much like BPF (Berkely Packet Filter) means shorter lines and less repetition. Support for nftables should also be compiled into the kernel, together with the related nftables modules. Please ensure that your kernel supports nf_tables before choosing this option.

Note: This section broadly assumes starting with an empty nftables firewall ruleset (established by flushing the rules with nft flush ruleset). Remediation steps included onlyaffect the live system, you will also need to configure your default firewall configuration to apply on boot. Configuration of a live systems firewall directly over a remote connection will often result in being locked out. It is advised to have a known good firewall configuration set to run on boot and to configure an entire firewall structure in a script that is then run and tested before saving to boot.

The following will implement the firewall rules of this section and open ICMP, IGMP, and port 22(ssh) from anywhere. Opening the ports for ICMP, IGMP, and port 22(ssh) needs to be updated in accordence with local site policy. Allow port 22(ssh) needs to be updated to only allow systems requiring ssh connectivity to connect, as per site policy.

Save the script bellow as `/etc/nftables/nftables.rules`

```
#!/sbin/nft -f

# This nftables.rules config should be saved as /etc/nftables/nftables.rules

# flush nftables rulesset
flush ruleset

# Load nftables ruleset

# nftables config with inet table named filter

table inet filter {
# Base chain for input hook named input (Filters inbound network packets)
chain input {
type filter hook input priority 0; policy drop;

# Ensure loopback traffic is configured
iif "lo" accept
ip saddr 127.0.0.0/8 counter packets 0 bytes 0 drop
ip6 saddr ::1 counter packets 0 bytes 0 drop

# Ensure established connections are configured
ip protocol tcp ct state established accept
ip protocol udp ct state established accept
ip protocol icmp ct state established accept

```

```
# Accept port 22(SSH) traffic from anywhere
tcp dport ssh accept

# Accept ICMP and IGMP from anywhere
icmpv6 type { destination-unreachable, packet-too-big, time-exceeded, parameter-problem, mld-listener-query, mld-listener-report, mld-listener-done, nd-router-solicit, nd-router-advert, nd-neighbor-solicit, nd-neighbor-advert, ind-neighbor-solicit, ind-neighbor-advert, mld2-listener-report } accept
icmp type { destination-unreachable, router-advertisement, router-solicitation, time-exceeded, parameter-problem } accept
ip protocol igmp accept
}

# Base chain for hook forward named forward (Filters forwarded network packets)
chain forward {
type filter hook forward priority 0; policy drop;
}

# Base chain for hook output named output (Filters outbount network packets)
chain output {
type filter hook output priority 0; policy drop;
# Ensure outbound and established connections are configured
ip protocol tcp ct state established,related,new accept
ip protocol udp ct state established,related,new accept
ip protocol icmp ct state established,related,new accept
}
}
```

Run the following command to load the file into nftables

```
# nft -f /etc/nftables/nftables.rules
```

All changes in the nftables subsections are temporary.

To make these changes permanent:

Run the following command to create the nftables.rules file

```
nft list ruleset > /etc/nftables/nftables.rules
```

Add the following line to `/etc/sysconfig/nftables.conf`

```
include "/etc/nftables/nftables.rules"
```

3.5.4 Configure iptables

IPtables is an application that allows a system administrator to configure the IPv4 and IPv6 tables, chains and rules provided by the Linux kernel firewall. While several methods of configuration exist this section is intended only to ensure the resulting IPtables rules are in place, not how they are configured. If IPv6 is in use in your environment, similar settings should be applied to the IP6tables as well.

3.5.4.1 Configure IPv4 iptables

Iptables is used to set up, maintain, and inspect the tables of IP packet filter rules in the Linux kernel.

Several different tables may be defined. Each table contains a number of built-in chains and may also contain user-defined chains.

Each chain is a list of rules which can match a set of packets. Each rule specifies what to do with a packet that matches. This is called a 'target', which may be a jump to a user-defined chain in the same table.

Note: This section broadly assumes starting with an empty IPtables firewall ruleset (established by flushing the rules with iptables -F). Remediation steps included only affect the live system, you will also need to configure your default firewall configuration to apply on boot. Configuration of a live systems firewall directly over a remote connection will often result in being locked out. It is advised to have a known good firewall configuration set to run on boot and to configure an entire firewall structure in a script that is then run and tested before saving to boot. The following script will implement the firewall rules of this section and open port 22(ssh) from anywhere:

```
#!/bin/bash

# Flush IPtables rules
iptables -F

# Ensure default deny firewall policy
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

# Ensure loopback traffic is configured
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
iptables -A INPUT -s 127.0.0.0/8 -j DROP

# Ensure outbound and established connections are configured
iptables -A OUTPUT -p tcp -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -p udp -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -p icmp -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A INPUT -p tcp -m state --state ESTABLISHED -j ACCEPT
iptables -A INPUT -p udp -m state --state ESTABLISHED -j ACCEPT
iptables -A INPUT -p icmp -m state --state ESTABLISHED -j ACCEPT

# Open inbound ssh(tcp port 22) connections
iptables -A INPUT -p tcp --dport 22 -m state --state NEW -j ACCEPT
```

3.5.4.2 Configure IPv6 ip6tables

Ip6tables is used to set up, maintain, and inspect the tables of IPv6 packet filter rules in the Linux kernel. Several different tables may be defined. Each table contains a number of built-in chains and may also contain user-defined chains. Each chain is a list of rules which can match a set of packets. Each rule specifies what to do with a packet that matches. This is called a `target', which may be a jump to a user-defined chain in the same table.

If IPv6 in enabled on the system, the ip6tables should be configured.

Note: This section broadly assumes starting with an empty ip6tables firewall ruleset (established by flushing the rules with ip6tables -F). Remediation steps included only affect the live system, you will also need to configure your default firewall configuration to apply on boot. Configuration of a live systems

firewall directly over a remote connection will often result in being locked out. It is advised to have a known good firewall configuration set to run on boot and to configure an entire firewall structure in a script that is then run and tested before saving to boot. The following script will implement the firewall rules of this section and open port 22(ssh) from anywhere:

```
#!/bin/bash

# Flush ip6tables rules
ip6tables -F

# Ensure default deny firewall policy
ip6tables -P INPUT DROP
ip6tables -P OUTPUT DROP
ip6tables -P FORWARD DROP

# Ensure loopback traffic is configured
ip6tables -A INPUT -i lo -j ACCEPT
ip6tables -A OUTPUT -o lo -j ACCEPT
ip6tables -A INPUT -s ::1 -j DROP

# Ensure outbound and established connections are configured
ip6tables -A OUTPUT -p tcp -m state --state NEW,ESTABLISHED -j ACCEPT
ip6tables -A OUTPUT -p udp -m state --state NEW,ESTABLISHED -j ACCEPT
ip6tables -A OUTPUT -p icmp -m state --state NEW,ESTABLISHED -j ACCEPT
ip6tables -A INPUT -p tcp -m state --state ESTABLISHED -j ACCEPT
ip6tables -A INPUT -p udp -m state --state ESTABLISHED -j ACCEPT
ip6tables -A INPUT -p icmp -m state --state ESTABLISHED -j ACCEPT

# Open inbound ssh(tcp port 22) connections
ip6tables -A INPUT -p tcp --dport 22 -m state --state NEW -j ACCEPT
```

4 Logging and Auditing

The items in this section describe how to configure logging, log monitoring, and auditing, using tools included in most distributions.

It is recommended that `rsyslog` be used for logging (with `logwatch` providing summarization) and `auditd` be used for auditing (with `aureport` providing summarization) to automatically monitor logs for intrusion attempts and other suspicious system behavior.

In addition to the local log files created by the steps in this section, it is also recommended that sites collect copies of their system logs on a secure, centralized log server via an encrypted connection. Not only does centralized logging help sites correlate events that may be occurring on multiple systems, but having a second copy of the system log information may be critical after a system compromise where the attacker has modified the local log files on the affected system(s). If a log correlation system is deployed, configure it to process the logs described in this section.

Because it is often necessary to correlate log information from many different systems (particularly after a security incident) it is recommended that the time be synchronized among systems and devices connected to the local network.

It is important that all logs described in this section be monitored on a regular basis and correlated to determine trends. A seemingly innocuous entry in one log could be more significant when compared to

an entry in another log.

Note on log file permissions: There really isn't a "one size fits all" solution to the permissions on log files. Many sites utilize group permissions so that administrators who are in a defined security group, such as "wheel" do not have to elevate privileges to root in order to read log files. Also, if a third party log aggregation tool is used, it may need to have group permissions to read the log files, which is preferable to having it run setuid to root. Therefore, there are two remediation and audit steps for log file permissions. One is for systems that do not have a secured group method implemented that only permits root to read the log files (`root:root 600`). The other is for sites that do have such a setup and are designated as `root:securegrp 640` where `securegrp` is the defined security group (in some cases `wheel`).

4.1 Configure System Accounting (auditd)

System auditing, through `auditd` , allows system administrators to monitor their systems such that they can detect unauthorized access or modification of data. By default, auditd will audit SELinux AVC denials, system logins, account modifications, and authentication events. Events will be logged to `/var/log/audit/audit.log` . The recording of these events will use a modest amount of disk space on a system. If significantly more events are captured, additional on system or off system storage may need to be allocated.

The recommendations in this section implement an audit policy that produces large quantities of logged data. In some environments it can be challenging to store or process these logs and as such they are marked as Level 2 for both Servers and Workstations. **Note:** For 64 bit systems that have arch as a rule parameter, you will need two rules: one for 64 bit and one for 32 bit systems. For 32 bit systems, only one rule is needed.

Note: Several recommendations in this section filter based off of `auid>=1000` for unprivileged non-system users. Some distributions split at `UID 500` instead, consult your documentation and/or the `UID_MIN` setting in `/etc/login.defs` to determine which is appropriate for you.

Note: Once all audit rules have been added to a file or files in the `/etc/audit/rules.d/` directory, the auditd service must be re-started, or the system rebooted, for the new rules to be included.

4.1.1 Ensure auditing is enabled

The capturing of system events provides system administrators with information to allow them to determine if unauthorized access to their system is occurring.

4.1.2 Configure Data Retention

When auditing, it is important to carefully configure the storage requirements for audit logs. By default, auditd will max out the log files at 5MB and retain only 4 copies of them. Older versions will be deleted. It is possible on a system that the 20 MBs of audit logs may fill up the system causing loss of audit data. While the recommendations here provide guidance, check your site policy for audit storage requirements.

4.2 Configure Logging

Logging services should be configured to prevent information leaks and to aggregate logs on a remote server so that they can be reviewed in the event of a system compromise and ease log analysis.

4.2.1 Configure rsyslog

The `rsyslog` software is recommended as a replacement for the `syslogd` daemon and provides improvements over `syslogd` , such as connection-oriented (i.e. TCP) transmission of logs, the option to log to database formats, and the encryption of log data en route to a central logging server. **Note:** This section only applies if `rsyslog` is installed on the system.

4.2.2 Configure journald

systemd-journald is a system service that collects and stores logging data. It creates and maintains structured, indexed journals based on logging information that is received from a variety of sources: Kernel log messages, via kmsg

Any changes made to the systemd-journald configuration will require a re-start of systemd-journald

5 Access, Authentication and Authorization

5.1 Configure cron

5.2 SSH Server Configuration

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

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

Command to remove the SSH daemon:

```
# apt purge openssh-server
```

Note: Once all configuration changes have been made to `/etc/ssh/sshd_config` , the sshd configuration must be reloaded:

Command to re-load the SSH daemon configuration:

```
# systemctl reload sshd
```

5.3 Configure PAM

PAM (Pluggable Authentication Modules) is a service that implements modular authentication modules on UNIX systems. PAM is implemented as a set of shared objects that are loaded and executed when a program needs to authenticate a user. Files for PAM are typically located in the `/etc/pam.d` directory. PAM must be carefully configured to secure system authentication. While this section covers some of PAM, please consult other PAM resources to fully understand the configuration capabilities.

5.4 User Accounts and Environment

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

environment.

5.4.1 Set Shadow Password Suite Parameters

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

6 System Maintenance

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

6.1 System File Permissions

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

6.2 User and Group Settings

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

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