System Configuration

Security Design

System Examination

System Configuration

Firewalls and Filters

Hardening Software

Backups and Change Management

Access Control and Authentication

Virtual Private Networking

Logging and Monitoring

Security Policy and Management Support

SELinux

Linux Systems Security

System Hardening

- Hardening is the process of removing, disabling, or limiting access in order to reduce attack choices, a.k.a. limiting the attack surface
- Software packages which cannot be properly limited may have to be removed and alternatives found
- Reducing connection options is a common tactic - limit to specific protocols, source IPs, force proxy connections, etc.
- Automating responses to detection of unexpected or unwanted system behaviour can help but can be abused in a confused deputy scenario unless done carefully
- Standards compliance and best practice guidelines can provide a minimum set of hardening requirements

Supplier Recommendations

- Recommendations from OS suppliers typically include:
  - Automated security updates
  - Implementing enforced strong password policies, MFA, encryption, etc.
  - Stricter system defaults
  - Security tools recommendations
  - Firewalls and filters available
  - Data security options

General Principles

- Utilize available restriction directives for all services
- Eliminate unneeded system and service accounts, be aware of any multipurpose uses of your linux account list (e.g. for default service user lists)
- Protect your logs, know where they are and what should be in them, use analysis tools to ease this task
- Know what you have and have what you know - set a baseline for what your systems should and shouldn't be running

Automatic Software Updating

- unattended-upgrades package is installed and enabled by default since 14.04.3 server
- Enable/disable automatic upgrades using dpkg-reconfigure, configure progress reports in /etc/apt/apt.conf.d/*upgrades files per the Ubuntu guidance, https://ubuntu.com/server/docs/package-management
- Consider what kinds of updates you want happening, may need canonical account to enable full updates as of 2023
- Make sure you can send and receive email before enabling progress report emails
- Firmware and kernel updates may require reboot unless signed up with Canonical for live update service

Security Stance Evaluation Tools

- The available security tools list is constantly growing and changing
- Many are included with the system and enabled for use at install (e.g. sudo, ssh, system monitoring tools)
- Review your installed security packages and install any security stance evaluation tools your distribution did not include that were covered in the system examination lesson
- May have to add universe and multiverse repositories to your /etc/apt/sources.list for some tools, see https://help.ubuntu.com/lts/serverguide/configuration.html for guidance regarding these repositories

Services

- Immediately after installation, your system will have a number of services already running in default configuration and very likely they are not protected by firewall rules or hardened by configuration settings
- The service command can be used to reveal what init script services are running
- The initctl command can be used to reveal the upstart services that are running
- The systemctl command can be used to reveal the systemd services that are running, systems running systemd usually also have the service command set to just run the systemctl command for you - service configuration/definition files are kept in /etc/systemd/system
- The man pages, supplier website, community website, user guides (/usr/share/doc), and google et al can be used to determine what a service is for, how it is configured, and help you figure out if it is a necessary service

Service Investigation

- Manually identifying potential vulnerabilities is time-consuming, requires imagination, and is error-prone
- An sample investigation might look something like this:
  - acpid is a service commonly run on Linux systems, it provides a mechanism to process system events using a configuration file
  - man acpid (or http://linux.die.net/man/8/acpid) shows that among other things, acpid listens on a socket file and will provide event data to any requesting process
  - It accepts multiple connections to that file
  - It can be told to use any file name as a pid file, overwriting whatever file might have been there
  - The man page says it has no known bugs
  - The default acpid configuration is in /etc/acpi
  - Those configuration files and directories are world-readable

Service Vulnerabilities

- google and its friends are your friends, but some friends may not always be helpful, complete, or accurate
- Searching *acpid security* can help you identify concerns not mentioned in acpid documentation, or ways acpid has been used that were never intended or foreseen
- http://www.cvedetails.com/vulnerability-list/vendor_id-9660/product_id-17268/Tim-Hockin-Acpid.html details a few vulnerability concerns with acpid from back in 2009
- The concerns about permissions can be addressed by tightening permissions on the acpi configuration and runtime files
- systemd has the ability to limit all sorts of things for services it runs, but most of them weren't designed to live within those kinds of restrictions so it can break things as much as help - see https://www.ctrl.blog/entry/systemd-service-hardening.html for more information on using systemd-analyze security
- Not all services run under systemd, some are standalone, so investigation requires time and effort

Package/Service Removal or Disabling

- apt-cache show pkgname provides details about a package
- apt-cache rdepends pkgname shows which packages depend on acpid, not the installed list
- apt --simulate purge pkgname can show if removing a package will cause other packages to be removed
- Remove services you do not require, software packages can be removed, if they were installed automatically as dependencies and are no longer required, using apt autoremove
- Disabling using systemctl disable servicename can be helpful if you don't want a service running but the package it belongs to provides dependencies to other required software - consider tightening permissions on unused but vulnerable software which you cannot purge due to dependencies

Connection Options

- Many services have configurable connection choices, both for methods and capacity
- Firewall tools and other configuration files may have separate options for ipv6 compared to ipv4
- ipv6 is commonly overlooked because it is not explicitly required by most programs
- If you don't need ipv6 support, disable it in /etc/sysctl.conf by following the recommendations at http://askubuntu.com/questions/309461/how-to-disable-ipv6-permanently
- Note the discussion in the comments about the wisdom of disabling ipv6 and the two competing views of the concept
- Be prepared to re-enable it if you get connected to an ipv6 network in the future

Linux System Accounts

- The default installation includes many user accounts even if the software that might use such an account isn’t installed
- Remove the ones you don’t need, you can find which accounts do not own at least one file with a trivial script, a similarly trivial script can identify groups with no files or users

```
for user in `cut -d : -f 1 /etc/passwd`; do
   [ $(sudo find / -ignore_readdir_race -user "$user" -print -quit |wc -l) -gt 0 ] ||
      echo "'$user' owns no files"
done
```

- You should also check to see if the username is documented before removing it, or exists because the group it belongs to is used to run system tools, checking otherwise unused accounts for mentions of them in configuration files or as process owners may be helpful
- These types of reference accounts usually do not have shells associated with them, and often do not have home directories, or their home directories are simply references to where their programs or spools are kept
- Package installations or upgrades may add users without telling you, or without you noticing

Non-firewall Connection Restrictions

- TCP wrappers are a tool that can add a layer of protection to external connections
- It is an older mechanism and not often used anymore, but still can be useful
- /etc/hosts.allow and /etc/hosts.deny list service names and associate them with IP addresses or ranges
- This can provide configuration flexibility unavailable from the configuration files included with those services
- TCP wrappers were originally used to wrap daemons started from the inetd and xinetd service programs, but many network service programs have support for TCP wrappers built into them (e.g. sshd)

Resource Limits

- /etc/security/limits.conf specifies limits on system resources available to users and groups
- It changes fairly regularly and is distro-specific
- Documentation is usually in the file and there is a man page for the file
- Review the settings in your file and tweak if appropriate
- Be sure you have adequate free space in your filesystems and swap space for your expected load or it becomes trivial to cause a DoS on your machine

Automated Responses

- Log files can be used to see what happened after the fact
- An automated response might be more useful if you can intercept the unusual behaviour or problem
- An example would be forcing reboot if you run out of memory in a DoS attack to avoid corrupting your system by trying to continue running without free memory (e.g. set vm.panic_on_oom and kernel.panic in /etc/sysctl.conf)
- Follow vendor recommendations for settings such as these, found in hardening guides

Hardening Guides

- Review the sections *initial setup*, *server setup*, and *disable services* as shown at http://bookofzeus.com/harden-ubuntu/ to see some recommendations for Ubuntu Server configuration
- Note that this is only an example, and this particular guide is not provided or approved by Canonical, the company that supports Ubuntu commercially and is becoming a bit long in the tooth
- Ubuntu's server guide has some recommendations from the vendor, and is tailored to each version of the OS

Filesystem Encryption

- Encrypting filesystems prevents access to data by non-authorized users
- This provides security for physical devices in the case of theft or drive duplication
- Can be used to secure external backup devices particularly if they are stored offsite
- Incurs a performance penalty which can be significant depending on your hardware
- Consider encrypting only filesystems with sensitive user data, lots of OS runtime data is kept in tmpfs and other pseudo-filesystems
- see https://help.ubuntu.com/lts/serverguide/ecryptfs.html for guidance regarding encrypting filesystems in Ubuntu

Kernel Tuning For Security

- The Linux kernel is highly customizable at runtime
- The primary mechanisms for this are the sysctl interface for manipulating kernel settings, and /proc filesystem which provides a filename-based way to retrieve and set kernel data
- What to tune and how to tune it are specific to your distro, system role, and services/applications - most distros provide defaults suited to the intended distro use
- See https://linux-audit.com/linux-hardening-with-sysctl/ for some ideas on where to start when doing kernel tuning

sysctl Basics

- /etc/sysctl.conf contains settings applied at boot
- sysctl command can retrieve or set current kernel settings
- sysctl -a --pattern setting-name-pattern is used to retrieve currently applied settings
- sysctl -w setting=value is used to dynamically apply a new setting value, sysctl -p loads all settings in a file
- To make changes permanent, edit sysctl.conf
- # or ; are used to indicate a comment in sysctl.conf

Configuration Checking Tools

- Lynis is an example of a typical tool which can review a system's configuration looking for common misconfigurations or exposures
- Available as a package, but it is important to ensure you have the current version
- The problems it identifies are not necessarily problems for your server and must be evaluated with respect to the administration practices for your server
- These caveats are applicable no matter which configuration checking tool you use