Hardening Software

Linux Systems Security

Hardening Software

- Hardening is a process
- Involves reducing or eliminating ways in which software may be used for unintended purposes
- Login and file access control can be used to limit access for local users
- Service access control can be used to limit remote as well as local users
- Most hardening is based on the capabilities for configuration present in the software providing access control

What can users do?

- Users need to be able to invoke functionality in software and services
- Involves permission to run commands
- Involves connecting to services
- Involves making services perform tasks on our behalf

Command Permissions

- Just because a command has some logic to prevent misuse doesn't mean it is safe
- Dynamically linked executables are dependent on the security of the libraries they use
- setuid or setgid programs present extra risks related to the potential for software design flaws, check to see which are not statically linked
- Buffer overflow issues are not a problem as long as your system has NX/PAE support in the CPU (Any relatively modern 64-bit CPU not intended for educational or embedded use) - 32-bit OS versions are the issue here

https://en.wikipedia.org/wiki/Executable_space_protection

Service Invocation

- What can a user make a service do?
- What restrictions can be placed on that?
- What sanitization of input can be done?
- How can we limit a compromised service?
- Limits may be configurable in the software that provides the service
- System capabilities may be useful to provide controls or boundaries on software that the software itself doesn't provide

Restricting Executables

- For each setuid or setgid program in the system, if it is executable by random users, decide if it needs to be set that way
- If you want to change the default ownership or permissions on a file belonging to a package, use the command dpkg-statoverride
- e.g. su and sudo programs are executable by any user, if you restrict those to sudo group users in your sudo configuration anyway, then change the executables to group execute for sudo group only

  dpkg-statoverride --update --add root sudo 4550 /usr/bin/sudo

- Using dpkg-statoverride updates the package database so that future updates and reinstalls do not change the mode back to the default

chroot Jails

- Normally processes can attempt access to any file in the system by specifying a pathname to reach it - the entire file namespace is available to all processes and only file permissions safeguard sensitive files
- You can use chroot to create a limited view of the filesystem for services that have no requirement for a complete view of the filesystem - even root cannot escape the limited view
- e.g. You can limit vsftpd users to only their home directories using the chroot_local_user option in the vsftpd.conf file, this is only useful if the users do not have shell access
- chroot is sometimes used for services to limit their reach in the case of a compromised service

AppArmor

- apparmor provides mandatory access control for executables, uses profiles in /etc/apparmor.d to specify what an executable is allowed to do
- Not all profiles are enabled by default, complain and enforce profiles are available
- Local customizations should be done in /etc/apparmor.d/local to avoid overwrites by package updates
- Additional profiles can be installed by installing apparmor-profiles package, beware careful with these as some are experimental
- Complain profiles are used to learn the access patterns of programs without preventing them from working, run aa-logprof from apparmor-utils to see what they complain about
- https://wiki.debian.org/AppArmor/HowToUse

Temporary Privileges

- Ubuntu default install adds the created user to the sudo group and allows anyone that belongs to the sudo group to run any command as root, probably not what you want for a production system
- Configured in /etc/sudoers which is the default policy plugin, logs sudo usage by default in /var/log/auth.log
- You can create a file per user or command or whatever in /etc/sudoers.d and it will be included in your sudoers configuration
- sudoedit program can be used if the only purpose of sudo is to edit a file, editor is specified in an environment variable (SUDO_EDITOR, VISUAL, EDITOR) or the sudoers file
- sudo -l tells you what commands you can use sudo for
- sudo -u username uses sudo to run commands as a specified username, root is the default

Sudo Configuration Basics

- /etc/sudoers file is the default policy specified in /etc/sudo.conf, if the sudo.conf file does not exist, sudoers is assumed
- It can define aliases or lists of users, groups, hosts, and commands
- General format of user specification entries is:

user-spec host = [options] [(user : group)] command list

- see sudoers(5) for full detail, refer to Examples section for sample lines to achieve various goals
- Edit the sudoers file using the command *visudo*
- The EDITOR variable can be used to specify an editor to use, such as nano or vi (i.e. *export EDITOR=vi*)

Hardening Services
Selected Examples

https://www.nuharborsecurity.com/ubuntu-server-hardening-guide-2/

- Services are provided by software
- That software is configurable
- Some configuration choices are specific to functionality, some are specific to security
- Some functionality choices have significant impacts on security
- There are lots of online guides for both OS and software hardening

DNS with BIND

- Stay up to date!
- Set your authoritative servers non-recursive and only accessible by the intended clients (firewall and acls with allow-query/blackhole)
- Use caching-only servers throughout your intranet to improve responsiveness to clients and reduce the opportunity for DoS-enabled attacks
- Use separate servers for internal vs. external use and strictly firewall the internal use servers to keep them responding quickly to valid queries (slowed servers are a vector for cache poisoning)

DNS with BIND

- Use the allow-query/blackhole statements in your named.conf.options file to refuse service to private and non-routed addresses which you are not using - refusals get logged
- Notify can be used by a LAN-resident attacker to cause a secondary server to try to DoS its primary. allow-notify on secondary servers can disable it completely while the attacker is expelled from the network - OOB updates of the secondary may be necessary in the interim

DNS with BIND

- Dynamically updated zones require special attention and transaction signatures with update-policy are recommended for them - do not use allow-update!
- If you use DHCP to dynamically update DNS, refer to the DHCP Server guide at https://isc.org for instructions on adding TSIG to your updates
- DHCP client services can be used to invoke load on a DNS server via dynamic updates, default config does not allow A record overwrites by other than the host that created them

DNS with BIND

- The rndc control channel is limited to localhost by default, if you open that up for remote control be sure to appropriately firewall port 953 and set up allow and keys statements to limit access
- Disable rndc using an empty controls {} statement if you do not plan to use rndc
- Review the other allow style options in the ARM for your version of BIND and stay up to date
- Consider DNS over HTTPS to improve protection from reconnaissance activities

Apache

- Review conf-available/security.conf
- Review recommendations in Ubuntu Hardening Guide
    - Prevent other sites from framing our pages (clickjacking), enable headers module
    - Disable TRACE
    - FileETag can be used to obtain file attributes, webdav uses this, consider disabling

Apache

- Review recommendations in Ubuntu Hardening Guide
  - Use a LimitExcept GET POST stanza and specify deny from all to turn off unnecessary request methods for Directory and Location stanzas, can break some protocols that ride on top of http like webdav
  - Use Header to edit Set-Cookie and Set-Cookie2 headers to have ;HttpOnly;Secure at the end
  - Use Header set X-XSS-Protection “1; mode-block” to reduce cross-site scripting attacks
  - Eliminate world read/access for most apache2 directories

Apache

- Review recommendations in Ubuntu Hardening Guide
  - Consider setting up logs on a per-virtual-host basis by prefixing log filenames with host names
  - Consider lowering Timeout in apache2.conf to prevent DoS by open connection count
  - Set ServerTokens Prod, ServerSignature off, Header unset Server, Header unset X-Powered-By
  - Consider using https only if load is not a problem

Apache

- Review recommendations in Ubuntu Hardening Guide
  - Add rewrite rules to forbid http/1.0 in site file, enable rewrite module
  - Consider disabling Indexes, Includes, and ExecCGI, and FollowSymLinks in Directory stanza's Options line
  - Consider setting a transfer size limit in Directory/Location stanza using LimitRequestBody
  - Check loaded modules to be sure they are required, disable otherwise

Modsecurity

- Web Application Firewall
- Enhanced traffic logging - request/response body capture
- Real-time monitoring and blocking based on rulesets permits application protection while waiting for application updates/improvements
- Can be embedded into web app server, or run on a proxy host
- https://github.com/SpiderLabs/ModSecurity/wiki/Reference-Manual
- https://www.digitalocean.com/community/tutorials/how-to-set-up-mod_security-with-apache-on-debian-ubuntu

mod_evasive

- Additional configuration for modsecurity to add evasive behaviour when it detects one of several types of attack patterns
- https://github.com/jzdziarski/mod_evasive
- apt-get install libapache2-mod-evasive, then edit mod_evasive.conf in mods-enabled directory, then create log file and set ownership on it, restart apache2

MySQL

- Oracle's best practices for MySQL security
- Run mysql_secure_installation to check for insecure install-time setup
- Review user access to databases to ensure it is the minimum necessary for the user's needs
- Consider changing the root username for mysql to something more difficult to guess, may break some configuration programs on older distros
- Disable local-infile in /etc/mysql/my.cnf
- Review bind-address in my.cnf to ensure it is restricted appropriately
- Set /etc/mysql to be owned by mysql:mysql and set mode 750 on it

Postfix

- From http://www.postfix.org/POSTSCREEN_README.html :

postscreen(8) is part of a multi-layer defense.

- As the first layer, postscreen(8) blocks connections from zombies and other spambots that are responsible for about 90% of all spam. It is implemented as a single process to make this defense as inexpensive as possible.
- The second layer implements more complex SMTP-level access checks with Postfix SMTP servers, policy daemons, and Milter applications.
- The third layer performs light-weight content inspection with the Postfix built-in header_checks and body_checks. This can block unacceptable attachments such as executable programs, and worms or viruses with easy-to-recognize signatures.
- The fourth layer provides heavy-weight content inspection with external content filters. Typical examples are Amavisd-new, SpamAssassin, and Milter applications.

Postfix

- See /usr/share/postfix/main.cf.dist for a commented list of options for your mail server which may or may not have been customized at install
- Look for the section on TRUST AND RELAY CONTROL
- Depending on how your distro configures your networks, the automatic local network detection code may not work properly so you may want or need to specify your trusted IPs for relaying
- See http://www.postfix.org/SMTPD_ACCESS_README.html for current recommendations on securing your mail with respect to relaying and access restrictions
- In particular, you should follow the vendor recommendations for setting up SASL authentication and use fail2ban email-related jails

Postfix

- Postfix has a number of parameters which can be used to provide limits on connections and messages
- The parameters can be placed in your main.cf if you want settings other than the defaults
- Documentation for them can be found in http://www.postfix.org/TUNING_README.html
- The default settings are suitable for typical servers and handle moderate workloads on average hardware
- Before changing the limits on resource usage, review the STRESS_README on the same site
- Use SMTPS (port 465) instead of SMTP (port 25) wherever possible, likely only your intranet servers

Dovecot

- Security was a primary motive for writing dovecot
- Less than 20 security concerns have been raised in almost 10 years of dovecot deployment, most are not actual issues, just ideas on avoiding potential issues
- Do not use unencrypted POP or IMAP - Dovecot SSL Configuration Guide

CUPS

- CUPS has some settings we might want to lock down, it is designed primarily to support local users
- Discover printers on the network - specify servers
- Share printers on the network - restrict browsing
- Limit the networks listened to for print requests
- Carefully construct your Location stanzas to only allow the connections you want, use SSLListen or SSLPort to listen for non-localhost connections to the admin web interface or disable the web interface
- Use VPNs for remote clients, allowing random connections is ill-advised because DoS of CUPS is too easy

https://www.cups.org/documentation.php/doc-1.4/security.html

FTP

- Use chroot jail for non-shell local users
- Use anonymous to share public documents, disable local users if you don’t need them
- Use sftp for authenticated remote access, not ftp

Vulnerabilities

- Check for heartbleed vulnerability
- Check for bash shellshock vulnerability
- Set MaxStartups to a more appropriate value in /etc/ssh/sshd_config to prevent parallelized attacks, consider reducing LoginGraceTime to reduce DoS on sshd
- Perform audits for personal software installations