



Auditing Your Unix and Linux Operating Systems

Mike Schiller, CISA

*Manager, IT Asset Management, Customer Service
and Critical Processes, Texas Instruments*

*Co-Author, 'IT Auditing: Using Controls to Protect
Information Assets'*



Agenda

- The Basics
- Account Management and Password Controls
- File Security and Controls
- Network Security and Controls
- Audit Logs
- Security Monitoring and General Controls
- Tools





The Basics



The Basics

Why Audit Your Operating System?

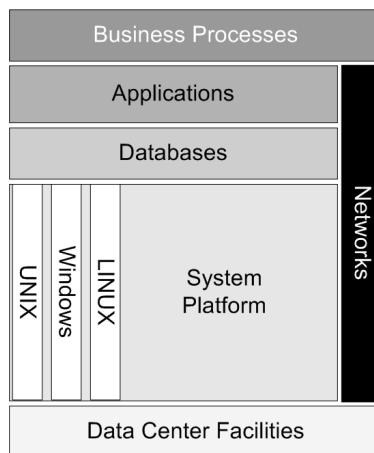


Table copyright 2007 The McGraw-Hill Companies





The Basics

Unix Variants (examples)

- Sun Solaris
- HP-UX
- SCO Unix
- AIX
- IRIX

Linux Variants (examples)

- Red Hat
- Debian
- Suse
- Gentoo

Note: commands in this presentation are Solaris and Red Hat but should work with most variants



The Basics

Key Concepts

- Everything in Unix is a file
- The root of the file system is /
 - Everything else branches off from the root
- 'root' account = admin / super-user
- If you can alter a file that someone is executing, you can easily capture his/her account





The Basics

File System Permissions

- Every file and directory has permissions specified for:
 - Owner
 - Group
 - World (Other)
- Three types of access are available:
 - Read (assigned a value of 4)
 - Write (assigned a value of 2)
 - Execute (assigned a value of 1)



The Basics

File System Permissions

- Example: If permissions on a file or directory are displayed as `rw-r-xr--`
 - This means that the file's owner has read, write, and execute permissions on the file, the file's group has read and execute permissions, and everyone else has read permissions.
 - This can also be described as 754 ('rw'=7, 'r-x'=5, 'r--'=4).
- Example: If permissions on a file or directory are displayed as `rw-r-----`
 - This means that the file's owner has read and write permissions on the file, the file's group has read permissions, and everyone else has no permissions.
 - This can also be described as 640 ('rw'=6, 'r--'=4, '---'=0).





The Basics

Interaction between file and directory permissions

| | | Directory Permissions | | | |
|------------------|----|-----------------------|---------------|----------------------|-------------------------------------|
| | | - | r | x | wx |
| File Permissions | - | No access | No access | No access | Delete file |
| | r | No access | No access | Read data | Delete file or read data |
| | w | No access | No access | Add to or clear data | Delete file or add to or clear data |
| | rw | No access | No access | Update data | Delete file or update data |
| | x | Can't execute | Can't execute | Execute | Delete file or execute |



The Basics

/etc/passwd file

- Contains information on the system's users
- **Format:** account:password:UID:GID:GECOS:directory:shell
 - Account - Name representing the user to the system (the name used when logging in).
 - Password - Encrypted password (but note that it may be kept in /etc/shadow instead).
 - UID - Numeric user ID
 - GID - Numeric group ID for the user's primary group
 - GECOS - Optional field used to store arbitrary additional information about the account. It often contains the real name and/or employee ID of the user.
 - Directory - Location of the user's home directory
 - Shell - User's default shell. The shell is the command line environment that interprets commands and passes them to the kernel.
- Be aware of centralized account management mechanisms (e.g. NIS, NIS+, LDAP)





The Basics

/etc/shadow file

- Contains password information
- Format:

`account:password:lastchange:min:max:warn:inactive:expired:reserved`

- Account - Name representing the user to the system
- Password - Encrypted password.
- Lastchange - Number of days since the password was changed
- Min - Minimum number of days allowed between password changes.
- Max - Maximum number of days allowed between password changes.
- Warn - Number of days before "Max" at which point the user will be warned that they need to change their password.
- Inactive - Number of days of inactivity after which the user's account will be disabled
- Expired - Number of days that the account has been disabled.
- Reserved - An extra field that is not used.



The Basics

/etc/group file

- Contains information on user groups
- Format: `name:password:GID:users`

- Name - Name of the group
- Password - Group password (if used)
- GID - Numeric group ID
- Users - List of users who are a member of the group. Note that members of the group who are assigned to it through their GID in /etc/passwd won't necessarily be in this list.





The Basics

Methods for Executing Commands with Elevated Privilege

- **sudo**
 - Allows a specific user to execute a specific command with the privilege of another user
 - Frequently used for selective root access
 - Configured in /etc/sudoers
- **SUID**
 - A SUID (Set-UID) file allows users to execute that file under the privileges of the ID that owns the file
 - Frequently used to allow all users to execute a specific command with root access
 - Permissions on a SUID file are displayed as (example):
 - `rwsr-sr-x`



The Basics

Important Linux and UNIX Navigation Commands for Auditors

- 'cd' - Change Directory
- 'ls' - List Directory Contents
 - 'ls -l' uses long listing format for the files within the directory, displaying file permissions
 - 'ls -ld' provides the long listing format for the directory itself
 - 'ls -al' provides the long listing format for the files within the directory, including the (.) dotfiles
- 'more', 'cat', 'less' - List File Contents
- 'ypcat' - List NIS File Contents ('niscat' for NIS+)
- 'sudo' – allows privileged execution of specific commands
- 'find'





Unix/Linux Test Steps

Categories:

- Account Management and Password
- File Security
- Network Security
- Audit Logs
- Security Monitoring and General Controls



Account Management and Passwords





Account Management and Password Test Steps

Note: For all steps in this section, remember to consider centralized account management tools (e.g. NIS, NIS+, LDAP)

1. Review account management processes (adds, deletes)
2. Ensure UID's are unique
 - If two users have the same UID, they can fully access each other's files and can 'kill' each other's processes.
 - `cat /etc/passwd | awk -F: '{print $3}' | uniq -d`
3. Ensure passwords are shadowed
 - `/etc/passwd` is world readable by design; `/etc/shadow` is not
 - Prevents the use of password-cracking tools against encrypted passwords



Account Management and Password Test Steps

4. Review file permissions of password and shadow password files
 - Access to alter these files provides ability to perform account management and escalate privileges
 - The `/etc/passwd` file should only be writable by 'root' and the `/etc/shadow` file should only be readable by 'root'.

- `ls -l /etc/passwd`

Expected output:

```
-r--r--r-- 1 root sys 728 Jan 26 16:23 /etc/passwd
```

- `ls -l /etc/shadow`

Expected output:

```
-r----- 1 root sys 374 Jan 26 16:23 /etc/shadow
```





Account Management and Password Test Steps

5. Evaluate the strength of system passwords
 - Review password composition controls (e.g. min password length, max password age, min password age)
 - `more /etc/default/passwd` (Solaris)
 - `more /etc/login.defs` (Red Hat)
 - Look for the presence of tools to enhance password composition requirements (e.g. npasswd)
 - Execute password-cracking tools to identify weak passwords
 - Review process for setting and communicating initial passwords



Account Management and Password Test Steps

7. Evaluate the usage of groups and determine their restrictiveness
 - Review the contents of the password and group file(s)
6. Review for usage of shared accounts
 - Review the contents of the password file(s)
 - The owner of each account should be obvious
 - GECOS field in `/etc/passwd` file if often used for this
 - Question any accounts that seem to be shared
 - If a shared account is necessary (e.g. application account)
 - Users should log in as themselves first
 - Use 'su' or 'sudo' to access the shared account
 - Shared accounts can be locked to force this behavior
 - Review `/etc/sudoers` file





Account Management and Password Test Steps

8. Evaluate access to super-user (root-level) access
 - Review password file(s) and ID all accounts with UID of 0
 - Question the need for any besides 'root' to have UID 0
 - Evaluate control of passwords for UID 0 accounts
 - Prevent direct 'root' logins to ensure accountability
 - Sysadmins should log in as themselves first
 - Use 'su' or 'sudo' to access 'root'
 - Use files such as /etc/default/login, /etc/securetty, /etc/sshd_config, and /etc/ftpusers to force this behavior
 - Review /etc/sudoers file



Account Management and Password Test Steps

9. Review the security of directories in the default user path and in root's path
 - If not secure, filename spoofing is possible
 - The default setting for users' paths may be found in /etc/default/login, /etc/profile, or one of the files in /etc/skel.
 - View the contents of these files with 'more filename'
 - Then review permissions on directories in path with 'ls -ld directoryname'
 - To view your own path: 'echo \$PATH'
 - Have sysadmin use this command to show root's path
 - Evaluate use of 'current directory' (depicted by '.') in paths





Account Management and Password Test Steps

10. Review the security of user home directories and config files

- Can allow privileged access to accounts
- Location of home directories can be viewed in the password file
- 'ls -ld' command will show permissions on a directory
- 'ls -al' command will show the permission on the files (including the config files) within a directory
- Typically want to limit write access to user's home directory and the configuration files to only the user (owner)



File Security





File Security Test Steps

1. Review file permissions for critical files and related directories
 - Typical targets for review:
 - /bin, /usr/bin, /sbin, /usr/sbin, /usr/local/bin (programs that interpret commands)
 - /etc (files that contain such information as passwords, group memberships, and trusted hosts and files that control the execution of various daemons)
 - /usr or /var (contain various accounting logs)
 - The kernel (core of the O/S)
 - Key application data and intellectual property specific to the server being reviewed
 - 'ls -ld' command will show permissions on a directory
 - 'ls -l' command will show the permission on a file



File Security Test Steps

Common UNIX and Linux directories

- /bin - location of most of the system binaries (programs)
- /sbin - contains binaries that are reserved for use by privileged accounts
- /etc - contains system configuration files
- /home - typical location for user home directories
- /var - contains information that programs need to keep track of as they run (such as the process ID on the system); usually contains log files as well
- /lib - system and application libraries; these aren't executed directly, but are used by applications as they run
- /opt - many add-on packages will be installed here
- /usr - place for user-added packages; often /usr will duplicate many of the top-level directories within itself, so you'll have /usr/etc, /usr/bin, etc.
- /root - the home directory for the root account is often here
- /tmp - temporary directory that any user can typically access
- /dev - you will find device files in this directory representing the hardware in your system

Text copyright 2007 The McGraw-Hill Companies





File Security Test Steps

2. Review the system for open directories (drwxrwxrwx)
 - Anyone can delete files within the directory and replace them with their own files of the same name
 - By placing the sticky bit on the directory (drwxrwxrwt), only the owner of a file can delete it
 - `find / -type d -perm -777`
 - Finds directories with world write permissions
 - Use judgment – focus on directories where key data is stored



File Security Test Steps

3. Evaluate the security of SUID files on the system
 - SUID (Set-UID) files allow users to execute them under the privileges of another UID
 - If an SUID file is writable by someone other than the owner, it may be possible for the owning account to be compromised
 - `find / -perm -u+s`
 - Provides a list of all SUID files (must be run by superuser)
 - Review file permissions using `'ls -l'` command





File Security Test Steps

4. Review the default umask value
 - Determines what permissions new files and directories will have by default
 - Type 'umask' to see your own umask
 - The umask subtracts privileges when files and directories are created
 - Normal default 777
 - Minus the umask 027
 - Default permissions on this server 750
 - This provides full access to the owner, read and execute access to the group, and no access to the world.
 - Recommended default umask values:
 - 027 (group write and all world access removed)
 - 037 (group write/execute and all world access removed)
 - Note that users can change their umask value
 - Want a secure default, requiring conscious decision to reduce security



File Security Test Steps

5. Review the security of files referenced within crontab entries
 - A cron executes a program at a preset time
 - The crontab contains all of the crons scheduled on the system
 - All crons are run as if the owner of the crontab is running them
 - If a file being executed within a crontab is not secure, it may allow for the execution of arbitrary commands
 - Use 'ls -l' on /usr/spool/cron/crontabs or /var/spool/cron/crontabs to see contents of crontab directory
 - Use 'more' to see contents of each file within directory
 - Use 'ls -l' and 'ls -ld' to see the permissions of the files being executed within crontab entries and their directories





Network Security



Network Security Test Steps

1. Evaluate necessity and security of enabled services
 - Every network service is a potential vector for attack
 - Unnecessary and unsecured network services allow someone who has no business being on the system to either gain access to the system or disrupt the system
 - Use 'netstat -an' to determine active services
 - If a service isn't needed, disable it
 - Having a secure OS baseline to start with reduces unnecessary services
 - If a service is needed, ensure security patches are being monitored and applied and that the service is configured securely
 - Execute a network vulnerability-scanning tool in order to check for current vulnerabilities in the environment





Network Security Test Steps

2. Evaluate the usage of trusted access
 - Trusted access provides the ability for users to access the system remotely without the usage of a password
 - /etc/hosts.equiv file creates trust relationships with specific machines
 - .rhosts files creates trust relationships with specific users on specific machines (located within individual home directories)
 - Security of the trusting system is dependent on the security of the trusted system
 - Delete where possible; minimize and secure otherwise
 - Ensure monitoring and approval mechanisms exist
 - Avoid '+' - defines all systems on the network as trusted



Network Security Test Steps

3. Review for the usage of secure protocols
 - Certain protocols (e.g. telnet, ftp, rsh, rlogin, and rcp) transmit all information in clear text, including userID and password
 - These can be disabled and replaced with secure (encrypted) alternatives
 - Telnet, rsh, and rlogin can be replaced by ssh
 - Ftp can be replaced by sftp or scp
 - Rcp can be replaced by scp





Network Security Test Steps

4. Evaluate the usage of .netrc files
 - Used to automate logons, primarily with FTP
 - May contain passwords
 - `find / -name '.netrc' -print -exec more {} \;`
 - Will display the contents of all .netrc files on the system
 - Must be run as superuser for thorough list
 - Ensure file permissions are locked down



Network Security Test Steps

5. Ensure a legal warning banner is displayed when connecting
 - Text is frequently located in /etc/issue and /etc/sshd_config
6. Review the usage of modems on the server
 - Bypass corporate perimeter security and allow direct access to the machine from outside the network
 - Preferable to have access to a machine channeled through standard corporate external access mechanisms such as VPN or RAS





Audit Logs



Audit Log Test Steps

1. Evaluate the contents, security, monitoring, and retention of system audit logs
 - **sudo log**
 - Typically /usr/adm/sulog, /var/adm/sulog, or /var/log/auth.log
 - **sudo log**
 - Typically written to the syslog but this can be changed in /etc/sudoers
 - **syslog**
 - /etc/syslog.conf file determines where each message type is routed
 - **Invalid logon attempts**
 - /var/adm/loginlog on Solaris, /etc/btmp on HP-UX
 - **wtmp**
 - Typically /usr/adm/wtmp, /var/adm/wtmp or /etc/wtmp
 - **utmp**
 - Typically located at /etc/utmp on UNIX and /var/run/utmp on Linux





Security Monitoring and General Controls



Security Monitoring and General Controls

1. Review procedures for monitoring the state of security on the system
 - Level of monitoring should be contingent on the criticality of the system and the inherent risk of the environment
 - Four primary types of security monitoring
 - Network vulnerability scanning
 - Host-based vulnerability scanning
 - Intrusion detection
 - Intrusion prevention
 - Assess frequency of monitoring
 - Look for evidence that the security monitoring tools are actually used and acted upon





Security Monitoring and General Controls

2. Review security of standard build for new systems
 - Audit a system freshly created from the baseline
 - Determines whether new systems are secure by default
 - If you start with a standard secure OS build, you only need to add the network services required by the application



Security Monitoring and General Controls

3. Ensure appropriate physical controls and operations are in place to provide for system protection and availability
 - Physical security
 - Environmental controls
 - Capacity planning
 - Change management
 - System monitoring
 - Backup processes
 - Disaster recovery planning
 - Secure coding practices and reviews for custom applications





Tools



Tools

The open source community has provided many tools that can assist the auditor:

- Nessus (network vulnerability scanner)
 - <http://www.nessus.org/>
 - <http://www.openvas.org/>
- NMAP (check for open ports)
 - <http://www.insecure.org/nmap/>
- Chkrootkit (identify both known rootkits running on a system and “suspicious” files or processes)
 - <http://www.chkrootkit.org/>
 - <http://www.netadmintools.com/art279.html>
- John the Ripper and Crack (check password strength)
 - <http://www.crypticide.com/users/alecm/security/c50-faq.html>
 - <http://www.openwall.com/john/>
- Tiger / TARA (host-based vulnerability scanner)
 - <http://savannah.nongnu.org/projects/tiger/>
 - <http://www-arc.com/tara/>
- Tripwire (intrusion detection / integrity checker)
 - <http://sourceforge.net/projects/tripwire/>

Note: the author assumes no responsibility for the impact of using these tools in your environment.





Tools

Print resources:

- “Practical UNIX & Internet Security” by Simson Garfinkel, Gene Spafford, and Alan Schwartz, published by O'Reilly Media, Inc.
- “Essential System Administration” by Eelen Frisch, published by O'Reilly Media, Inc.
- “IT Auditing: Using Controls to Protect Information Assets” by Chris Davis, Mike Schiller, and Kevin Wheeler ☺



Tools

Online resources:

- <http://isaca.org/>
 - standards and security guidance
- <http://www.sans.org/rr/>
 - certifications and other documents from SANS
- http://www.nsa.gov/ia/guidance/security_configuration_guides/index.shtml
 - security configuration guides from the National Security Agency
- <http://csrc.nist.gov/publications/PubsSPs.html>
 - security guidelines from the National Institute of Standards and Technologies
- <http://www.insecure.org/tools.html>
 - top 100 security tools as generated from a survey of NMAP users
- <http://seclists.org/>
 - security-oriented mailing lists
- <http://www.securityfocus.com/>
 - mailing lists, news, vulnerabilities
- <http://cve.mitre.org/>
 - along with the vulnerability database section of securityfocus, a good site to begin research on potential vulnerabilities





Thank you!

