Linux Access Control

Linux System Administration
COMP2018 Fall 2019

Linux Process Concepts

- A process is composed of the code for a program, the data for that program, and control information about that program as it runs, with that control information stored by process id number (pid) in the kernel's process table

- Along with a lot of other information, process control information includes the id of the user who started the program, the group id for the process, and the environment of the process

- Processes are started by the kernel, aided by the kernel, ended by the kernel, and restricted by the kernel based on the process owner and group ids

- Access to file and kernel resources is controlled by process owner and group ids in conjunction with file ownerships and permissions

- Access to services is controlled by each service individually

Monitoring Processes

- The ps command displays information about processes
- The pstree command displays the ancestry of processes
- The uptime command displays a summary of system usage
- The top command displays a continuously updating display of the top consumers of system resources along with a summary of important system statistics
- The htop command simplifies the statistics display but is otherwise the same as the top command

Administrative Access

- Most programs (which become processes when they are run) in a system can be run by any user, e.g. cat, more, vi, mkdir, cd, rm, etc.

- Superuser privilege is required to make changes to the system, e.g. modify configuration files, enable or disable services or devices, etc.

- For increased security and logging of access, we login as a normal user and use su or sudo to perform superuser tasks

- Extended sessions requiring root are often performed by starting a root shell using su or sudo

- The installation process for most distributions includes setting up this basic access for a default account

User Account Creation, Modification, And Removal

- adduser (configured in /etc/adduser.conf), deluser are used to create and delete user accounts
- There are user account name restrictions (length and content usually limited to 32 characters, alphanumeric characters, - and _)
- useradd (configured in /etc/default/useradd, /etc/login.defs), usermod, userdel are the underlying commands used by the user-friendly commands above
- passwd, chage are used to set, remove, and maintain password-related information for user accounts, including locking and unlocking accounts
- User id and group id numbers have defaults so that they can be automatically assigned to new user accounts
- When processes create files, their user and primary group id numbers are used to set the user and group ownership of the created files
- The ownership ids for the files stay with the files, not with the user, so changes made to the user accounts do not affect existing file ownerships

User Account Management

- pwck is used to check for inconsistencies and incomplete account setups as found the the /etc/passwd, /etc/shadow files where user accounts are stored

- Home directories for users are created when the accounts are created and populated with the contents of /etc/skel

- When removing user accounts finding files owned by a user can be done with the find command

- User accounts are managed by root, using the su command (requires belonging the wheel group by default) or sudo command (requires belonging to the sudo group by default) which is configured in the /etc/sudoers file

User Account Monitoring

- Reviewing user accounts is a system maintenance task
- lslogins, getent passwd, getent group are commands to view what accounts are configured on the system
- id, who, w, last, lastcomm are commands to view current user information, or historical access information without directly combing through the system logs

Groups

- groupadd, groupmod, groupdel are the low level commands to create, modify and remove group definitions from the system
- addgroup, delgroup are friendlier commands
- grpck is used to perform consistency checking on the /etc/group and /etc/gshadow files
- Finding files owned by a group can be done using the find command

Special Groups

- sudo and wheel groups for sudo and su commands
- lpadmin group for administering printers
- lxd, libvirtd groups for lxd container access
- plugdev group for removable device mount/umount
- https://wiki.debian.org/SystemGroups is an example of vendor group definitions for the Debian distro

Regular File Permissions

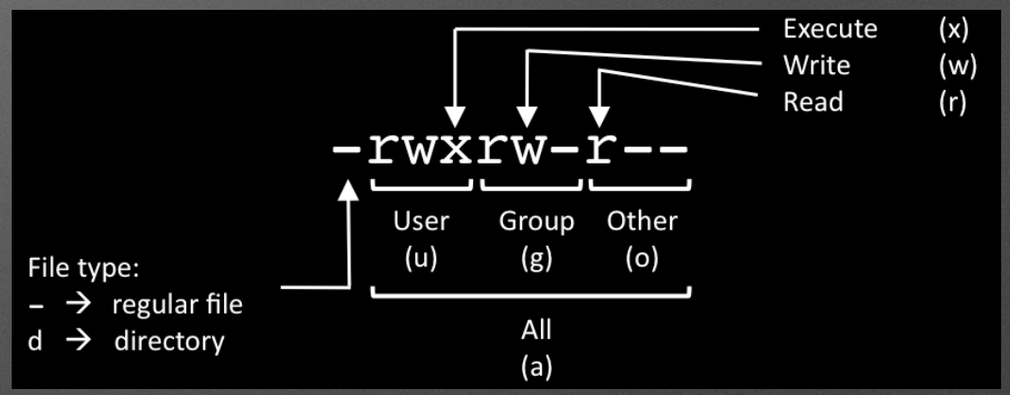
- read - can get the content of the file
- write - can change the content of the file
- execute - can try to run the content of the file as a command
- set-user-id (setid, setuid) - when run as a command, the process will run as the owner of the file, no matter what user account started the process
- set-group-id (setgid) - when run as a command, the process will run as the group that owns the file, no matter what user account started the process
- umask - can be used to view or set the permissions mask used for file creation

Directory Permissions

- read - can view the list of files in the directory
- write - can change the list of files in the directory (create and remove files)
- access - can access the files in the directory, required to create or remove files from the directory, as well as to try to use files in the directory
- set-group-id (setgid) - controls whether files created in the directory belong to the group that owns the directory, or the group id of the process creating the files
- sticky - prevents removal of files you do not own from the directory
- umask - can be used to view or set the permissions mask used for directory creation

Working With Permissions

- ls -l shows permissions
- chmod changes permissions



  - symbolic syntax for modifying permissions
  - numeric syntax for explicitly setting permissions

Permission	Numerical Value	Relative Value	On Files	On Directories
SUID	4	u+s	User executes file with permissions of file owner.	No meaning.
SGID	2	g+s	User executes file with permissions of group owner.	File created in directory gets the same group owner.
Sticky bit	1	+t	No meaning.	Users are prevented from deleting files from other users.

Octal	Decimal	Permission	Representation
000	0 (0+0+0)	No Permission	---
001	1 (0+0+1)	Execute	--x
010	2 (0+2+0)	Write	-w-
011	3 (0+2+1)	Write + Execute	-wx
100	4 (4+0+0)	Read	r--
101	5 (4+0+1)	Read + Execute	r-x
110	6 (4+2+0)	Read + Write	rw-
111	7 (4+2+1)	Read + Write + Execute	rwx

Extended Attributes

- ls -l shows a dot at the end of normal permissions if there are extended attributes
- lsattr to show attributes
- chattr to set or change attributes
- chattr man page shows definitions of attributes

File Ownership

- root is required to change a file's owner
- chown can change owner, group, or both
- recursive option available
- chgrp command is for group change only
- root is not required to change a file's group, but there are still rules
- Orphaned files can be found using the find command

Service Access Control

- Services such as ssh have their own configuration files
- Those configuration files can specify access restrictions unique to each service
- Firewalls can be used to further restrict access to network services

LDAP

- Directory service for system information management
- Users can be added to LDAP and then LDAP can be used to authenticate user access
- LDAP provides more sophisticated mechanisms for access control at the cost of complexity in management of users
- The name service switch (/etc/nsswitch.conf) controls whether LDAP is used and how
- https://wiki.debian.org/LDAP/NSS is an example of how a distro might provide LDAP integration

SELinux

- SELinux is security-enhanced linux
- It was developed by and with the NSA
- Provides the ability to label every object in the system and apply a least-privilege security model when any access is attempted to labeled objects
- Most users and administrators do not directly work with SELinux, but simply deploy vendor-supplied SELinux controls