Root Access

Linux System Administration
Fall 2022

Being root

- There are no restrictions on what root can do
- When you are root, a typo can brick the system
- If root login is compromised, access is complete

Playing it safe

- Limit access to root login
  - Disable account or limit account to local logins
- Grant root access as needed with sudo
  - Fine grained control on commands
- Grant access via trusted commands
  - Special permission bits
  - Special file capabilities

Permissions Reserved for root

- Mounting a filesystem
- Changing a file or folder user & group ownership
- Bypassing the kernel to obtain or send network traffic on the physical wire itself
- Listening on TCP / UDP ports lower than 1024
- Controlling systemd
  - Reboot, start/stop system processes, etc.

Limiting root logins

- Set root login shell to /sbin/nologin
- PAM
  - /etc/pam.d/system-auth
  - /etc/pam.d/sshd
  - /etc/security/access.conf
- /etc/ssh/sshd_config
  - PermitRootLogin
    - no
    - yes
    - prohibit-password
    - forced-commands-only

root login policy

- No root access is the most restrictive
  - Could be a problem if the machine is hung
- No ssh logins is good for DMZ or untrusted net
  - Console access can be a useful backup
- Limited ssh logins
  - Good for remote administration

sudo



- su "do" (super user do)
- Preferred way of elevating permissions
  - Allow by user and by system command access
  - Logging of commands
- Periodically require password
- Partly developed at CU Boulder

sudo configuration

- Main configuration
  - `/etc/sudoers`
- Local configuration
  - `/etc/sudoers.d/*`
  - File name does not matter
    - Choose by function
    - Facilitates distributed management
- Edit configuration files using `visudo`

sudo environment

- `env_reset`
  - Limits variables from user shell transferred to root
- `secure_path`
  - Modifies PATH used to find commands
- Integrates with PAM

Permitted Commands

- Allow root and wheel (or sudo) to do anything
  - `root ALL=(ALL) ALL`
  - `%wheel ALL=(ALL) ALL`
- Generic format
  - `user host = command`
  - `user host = (user) command`
    - % refers to a group
    - , for a list
    - ! to exclude
    - ALL means no restriction

Aliases

- Host_Alias
  ```
  Host_Alias FILESERVERS = www,nfs
  ```
- User_Alias
  ```
  User_Alias MANAGERS = mscott,jhalpert
  ```
- Cmnd_Alias
  ```
  Cmnd_Alias START /usr/bin/systemctl start,/usr/bin/systemctl restart
  ```
- Aliases are context sensitive

Wildcards

- Match command string
  - Could be the command, parameter or files
  - Symbol matching, e.g. `[0-9]`
  - Wildcards
    - `*` zero or more characters
    - `?` zero or one characters
- Command allowed if it matches any line in sudoers
- Matches are BNF not regexes
  - Exclude matches with !

Wildcard examples

- Allow mscott access to `/var/log/messages` and backups
  - `mscott ALL=/bin/cat /var/log/messages`
  - `mscott ALL=/bin/cat /var/log/messages.[1-9]`
- Never end a command in *
  - Allows users to sneak additional stuff at the end of the command
  - `mscott ALL=/bin/cat /var/log/messages*`
    allows
    `sudo cat /var/log/messages.1` and
    `sudo cat /var/log/messages /etc/shadow`

No password

- Execute a command with elevated privileges without requiring the usual sudo password

```
mscott ALL=NOPASSWD: /usr/bin/systemctl restart apache2
```

- ***Use with caution!!!***
  - Password encourages the user to think before acting
  - Limit the command to the absolute minimum

sudo logging

- Commands run using sudo are logged
  - debian `/var/log/auth.log`
  - redhat `/var/log/audit/audit.log`
- Remote logging possible



sudo tips

- Test your configurations thoroughly
  - Could be a significant security hole
- Use aliases to simplify user experience
- Maintain customization in `/etc/sudoers.d/`
  - Works best with distributed management

Executables with privilege escalation

- setuid
  - run using UID of file owner of executable
- setgid
  - runs using GID of file group of executable
- setcap
  - Allow executable access to specific facilities otherwise requiring root

Extra Permission Bits

NUMBER	PERMISSION	RWX
7	setuid, setguid, stickybit	111
6	setuid and setgid	110
5	setuid, and stickybit	101
4	setuid	100
3	setgid, and stickybit	011
2	setgid	010
1	stickybit	001
0	none	000

Changing your password

- /etc/shadow
  - Ownership `root:shadow`
  - Permissions `-rw-r-----`
- /usr/bin/passwd
  - Ownership `root:root`
  - Permissions `-rwsr-xr-x`
  - Program is trusted to change the right user