Section Overview

What You Will Learn

- File and directory permissions.
- Sharing data securely.
- Special permissions.
- File attributes.
- Access Control Lists (ACLs).
- Rootkits.

File System Security

Special Modes

Setuid

- When a process is started, it runs using the starting user's UID and GID.
- setuid = **S**et **U**ser **ID** upon execution.
- `-rwsr-xr-x 1 root root /usr/bin/passwd`
- `ping`
- `chsh`
- setuid files are an attack surface.
- Not honored on shell scripts.

Octal Permissions

setuid	setgid	sticky	
0	0	0	Value for off
1	1	1	Binary value for on
4	2	1	Base 10 value for on

Adding the Setuid Attribute

```
chmod u+s /path/to/file

chmod 4755 /path/to/file
```

Removing the Setuid Attribute

```
chmod u-s /path/to/file
```

```
chmod 0755 /path/to/file
```

Finding Setuid Files

```
find / -perm /4000

# Older style:
find / -perm +4000
```

Finding Setuid Files

```
find / -perm /4000 -ls


# Older style:
find / -perm +4000 -ls
```

Only the Owner Should Edit Setuid Files

	Symbolic	Octal
Good:	`-rwsr-xr-x`	`4755`
Bad:	`-rwsrwxr-x`	`4775`
Really bad:	`-rwsrwxrwx`	`4777`

Setgid

- setgid = **S**et **G**roup **ID** upon execution.
- `-rwxr-sr-x 1 root tty /usr/bin/wall`
- `crw--w---- 1 bob  tty /dev/pts/0`

Finding Setgid Files

```
find / -perm /2000 -ls

# Older style:
find / -perm +2000 -ls
```

Adding the Setgid Attribute

```
chmod g+s /path/to/file
```

```
chmod 2755 /path/to/file
```

Adding the Setuid & Setgid Attributes

```
chmod ug+s /path/to/file

chmod 6755 /path/to/file
```

Removing the Setgid Attribute

```
chmod g-s /path/to/file
```

```
chmod 0755 /path/to/file
```

Setgid on Directories

- setgid on a directory causes new files to inherit the group of the directory.
- setgid causes directories to inherit the setgid bit.
- Is not retroactive.
- Great for working with groups.

Use an Integrity Checker

- Other options to `find`.
- Tripwire
- AIDE (Advanced Intrusion Detection Environment)
- OSSEC
- Samhain
- Package managers

The Sticky Bit

- Use on a directory to only allow the owner of the file/directory to delete it.
- Used on /tmp:

```
drwxrwxrwt 10 root root 4096 Feb 1 09:47 /tmp
```

Adding the Sticky Bit

```
chmod o+s /path/to/directory
```

```
chmod 1777 /path/to/directory
```

Removing the Sticky Bit

```
chmod o-t /path/to/directory
```

```
chmod 0777 /path/to/directory
```

Reading `ls` Output

- A capitalized special permission means the underlying normal permission is not set.
- A lowercase special permission means the underlying normal permission set.

Reading `ls` Output

```
$ ls -l test
-rw-r--r-- 1 root root 0 Feb 14 11:21 test
$ chmod u+s test
$ ls -l test
-rwSr--r-- 1 root root 0 Feb 14 11:21 test
$ chmod u+x test
$ ls -l test
-rwsr--r-- 1 root root 0 Feb 14 11:21 test
```

Reading `ls` Output

```
$ ls -l test
-rw-r--r-- 1 root root 0 Feb 14 11:21 test
$ chmod u+s test
$ ls -l test
-rwSr--r-- 1 root root 0 Feb 14 11:21 test
$ chmod u+x test
$ ls -l test
-rwsr--r-- 1 root root 0 Feb 14 11:21 test
```

Reading `ls` Output

```
$ ls -l test
-rw-r--r-- 1 root root 0 Feb 14 11:21 test
$ chmod u+s test
$ ls -l test
-rwSr--r-- 1 root root 0 Feb 14 11:21 test
$ chmod u+x test
$ ls -l test
-rwsr--r-- 1 root root 0 Feb 14 11:21 test
```

Reading `ls` Output

```
-rwxrwSr-- 1 root root 0 Feb 14 11:21 test

drwxr-xr-T 2 root root 0 Feb 14 11:30 testd
```

File Attributes

File Attributes (xattr)

- Supported by many file systems.
- ext2, ext3, ext4
- XFS
- Btrfs, ReiserFS, JFS
- OCFS2, OrangeFS, Lustre
- SqaushFS, F2FS

Attribute: `i` immutable

- The file cannot be:
  - modified
  - deleted
  - renamed
  - hard linked to
- Unset the attribute in order to delete it.

Attribute: `a` append

- Append only.
- Existing contents cannot be modified.
- Cannot be deleted while attribute is set.
- Use this attribute on log files.
- Safeguard the audit trail.

Other Attributes

- Not every attribute is supported.
- `man ext4`, `man xfs`, `man brtfs`, etc.
- Example: `s` secure deletion

Viewing Attributes

- Use the `lsattr` command.

```
# lsattr /etc/motd
---------------- /etc/motd
# lsattr /var/log/messages
-----a---------- /var/log/messages
```

Viewing Attributes

- Use the `lsattr` command.

```
# lsattr /etc/motd
---------------- /etc/motd
# lsattr /var/log/messages
-----a---------- /var/log/messages
```

Viewing Attributes

- Use the `lsattr` command.

```
# lsattr /etc/motd
---------------- /etc/motd
# lsattr /var/log/messages
-----a---------- /var/log/messages
```

Modifying Attributes

- Use the `chattr` command.
- `+` adds attributes.
- `-` removes attributes.
- `=` sets the exact attributes.

Examples

```
# lsattr /var/log/messages
---------------- /var/log/messages
# chattr +a /var/log/messages
# lsattr /var/log/messages
-----a---------- /var/log/messages
```

Examples

```
# lsattr /var/log/messages
-----a---------- /var/log/messages
# chattr -a /var/log/messages
# lsattr /var/log/messages
---------------- /var/log/messages
```

Examples

```
# lsattr /etc/hosts
---------------- /etc/hosts
# chattr =is /etc/hosts
# lsattr /etc/hosts
s---i----------- /etc/hosts
```

Examples

```
# lsattr /etc/hosts
s---i----------- /etc/hosts
# chattr = /etc/hosts
# lsattr /etc/hosts
---------------- /etc/hosts
```

File Attributes Demo

Access Control Lists

ACLs

- ACL = Access Control List
- Provides additional control
- Example: Give one user access to a file.
- Traditional solution is to create another group.
    - Increases management overhead of groups.

```
groupa: tom, jane
groupb: tom, jane, bob
```

ACLs

- Ensure file system mounted with ACL support

```
mount -o acl /path/to/dev /path/to/mount
tune2fs -o acl /path/to/dev
```

- Check:

```
tune2fs -l /path/to/dev | grep options
```

Types of ACLs

- Access
  - Control access to a specific file or directory.
- Default
  - Used on directories only.
  - Files without access rules use the default ACL rules.
  - Not retroactive.
  - Optional.

ACLs Can Be Configured:

- Per user
- Per group
- For users not in the file’s group
- Via the effective rights mask

Creating ACLs

- Use the `setfacl` command.
- May need to install the ACL tools.
- Modify or add ACLs:

```
setfacl -m ACL FILE_OR_DIRECTORY
```

User ACLs / Rules

`u:uid:perms` Set the access ACL for a user.

```
setfacl -m u:jason:rwx start.sh
```

```
setfacl -m u:sam:xr start.sh
```

Group ACLs / Rules

`g:gid:perms` Sets the access ACL for a group.

```
setfacl -m g:sales:rw sales.txt
```

Mask ACLs / Rules

`m:perms` Sets the effective rights mask.

```
setfacl -m m:rx sales.txt
```

Other ACLs / Rules

`o:perms` Sets the access ACL for others.

```
setfacl -m o:r sales.txt
```

Creating Multiple ACLs at Once

```
setfacl -m u:bob:r,g:sales:rw sales.txt
```

Default ACLs

`d:[ugo]:perms` Sets the default ACL.

```
setfacl -m d:g:sales:rw sales
```

Setting ACLs Recursively (-R)

```
setfacl -R -m g:sales:rw sales
```

Removing ACLs

Format:

```
setfacl -x ACL FILE_OR_DIRECTORY
```

Removing ACLs

Format:

```
setfacl -x ACL FILE_OR_DIRECTORY
```

Examples:

```
setfacl -x u:jason sales.txt
```

Removing ACLs

Format:

```
setfacl -x ACL FILE_OR_DIRECTORY
```

Examples:

```
setfacl -x u:jason sales.txt
setfacl -x g:sales sales.txt
```

Removing ACLs

Format:

```
setfacl -b FILE_OR_DIRECTORY
```

Example:

```
setfacl -b sales.txt
```

Viewing ACLs

```
$ getfacl sales.txt
# file: sales.txt
# owner: root
# group: root
user::rw-
group::r--
other::r--
```

Viewing ACLs

```
$ getfacl sales.txt
# file: sales.txt
# owner: root
# group: root
user::rw-
group::r--
other::r--
```

Viewing ACLs

```
$ getfacl sales.txt
# file: sales.txt
# owner: root
# group: root
user::rw-
group::r--
other::r--
```

```
$ setfacl -m u:jason:rw sales.txt
$ getfacl sales.txt
# file: sales.txt
# owner: root
# group: root
user::rw-
user:jason:rw-
group::r--
mask::rw-
other::r--
```

```
$ setfacl -m u:jason:rw sales.txt
$ getfacl sales.txt
# file: sales.txt
# owner: root
# group: root
user::rw-
user:jason:rw-
group::r--
mask::rw-
other::r--
```

```
# file: sales
# owner: root
# group: root
user::rwx
group::r-x
other::r-x
default:user::rwx
default:group::r-x
default:group:sales:rw-
default:mask::rwx
default:other::r-x
```

```
# file: sales
# owner: root
# group: root
user::rwx
group::r-x
other::r-x
default:user::rwx
default:group::r-x
default:group:sales:rw-
default:mask::rwx
default:other::r-x
```

Detecting Files with ACLs

```
$ ls -l
-rw-r--r--  1 root root 0 Feb 16 21:00 loans.txt
-rw-rw-r--+ 1 root root 0 Feb 16 21:09 sales.txt
-rwxr-xr-x  1 root root 0 Feb 17 11:00 start.sh
```

Detecting Files with ACLs

```
$ ls -l
-rw-r--r--  1 root root 0 Feb 16 21:00 loans.txt
-rw-rw-r--+ 1 root root 0 Feb 16 21:09 sales.txt
-rwxr-xr-x  1 root root 0 Feb 17 11:00 start.sh
```

Access Control Lists - Demo

Rootkits

Rootkits

- Software used to gain root access and remain undetected.
- They attempt to hide from system administrators and antivirus software.

Rootkits

- User space rootkits replace common commands such as `ls`, `ps`, `find`, `netstat`, etc.
- Kernel space rootkits add or replace parts of the core operating system.
  - Loadable Kernel Modules (LKMs)
  - /dev/kmem
  - /dev/mem

Rootkit Detection

- Use a file integrity checker for user space rootkits. (AIDE, tripwire, OSSEC, etc.)
- Identify inconsistent behavior of a system.
  - High CPU utilization without corresponding processes.
  - High network load or unusual connections.

Rootkit Detection

- Kernel mode rootkits have to be running in order to hide themselves.
- Halt the system and examine the storage.
    - Use a known good operating system to do the investigation.
    - Use bootable media, for example.

chkrootkit

- Shell script that searches for rootkits.
  - Detects modification of system binaries.
  - Checks for promiscuous mode.
  - Checks for missing lastlog and utmp entries.
  - Looks for LKM trojans.
- Run interactively or schedule execution.
- http://www.chkrootkit.org

Rootkit Hunter / RKHunter

- Shell script that searches for rootkits.
- http://rkhunter.sourceforge.net

```
# rkhunter --update
# rkhunter --propupd
# rkhunter -c
# cat /var/log/rkhunter.log
# rkhunter -c --rwo
# rkhunter --cronjob
```

Rootkit Hunter Configuration

```
/etc/rkhunter.conf:


MAIL-ON-WARNING=admins@example.com
```

Rootkit Hunter Configuration

/etc/rkhunter.conf:

```
MAIL-ON-WARNING=admins@example.com

ALLOWHIDDENDIR="/dev/.udev"
ALLOWHIDDENFILE="/dev/.blkid.tab"
ALLOWHIDDENFILE="/dev/.blkid.tab.old"
```

OSSEC - http://ossec.github.io/

- Host Intrusion Detection System (HIDS)
- More than just rookit detection: log analysis, file integrity checking, alerting.
- Syscheck module - user mode rootkit detection.
- Rootcheck module - both user mode and kernel mode rootkit detection.

OSSEC Rootcheck

- Searches for file names known to be associated with user mode rootkits.
- Signature based rootkit detection.
- Queries the OS for information and looks for inconsistent results.
  - Compares netstat with bind() results.
  - Many other checks.

Rootkit Removal

- Keep a copy of the data if possible.
- Learn how to keep it from happening again.
- Reinstall core OS components and start investigating.
    - Not recommended. Easy to make a mistake.
- Safest is to reinstall the OS from trusted media.

Rootkit Prevention

- Use good security practices:

Rootkit Prevention

- Use good security practices:
  - Physical
  - Account
  - Network

Rootkit Prevention

- Use good security practices:
  - Physical
  - Account
  - Network
- Use file integrity monitoring:
  - AIDE
  - Tripware
  - OSSEC

Rootkit Prevention

- Use good security practices:
  - Physical
  - Account
  - Network
- Use file integrity monitoring:
  - AIDE
  - Tripware
  - OSSEC
- Keep your systems patched.

RKHunter Demo