CSCI 476: Computer Security

Lecture 5: Set-UID and Environment Variables

Reese Pearsall
Fall 2022

Announcements

Lab 1 Due **FRIDAY** 9/16 @ 11:59 PM
• Shouldn't be too bad

Note taker still needed

How would you protect your computer and its resources?

Access Control

who can do **what** to **whom?**

users/groups
what is their identity?

Objects
Usually things on a filesystem

permissions (read/write/execute)
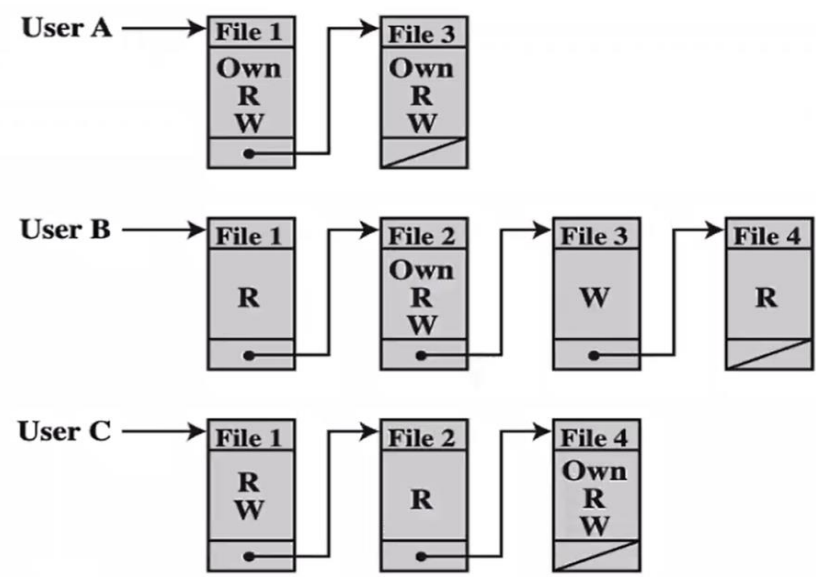Ok, I know the who– what are *you* permitted to do?

Access Control

who can do **what** to **whom?**

OBJECTS

SUBJECTS		File 1	File 2	File 3	File 4
	User A	Own Read Write		Own Read Write	
	User B	Read	Own Read Write	Write	Read
	User C	Read Write	Read		Own Read Write

Access Control Matrix

What are some issues with this?

MONTANA STATE UNIVERSITY
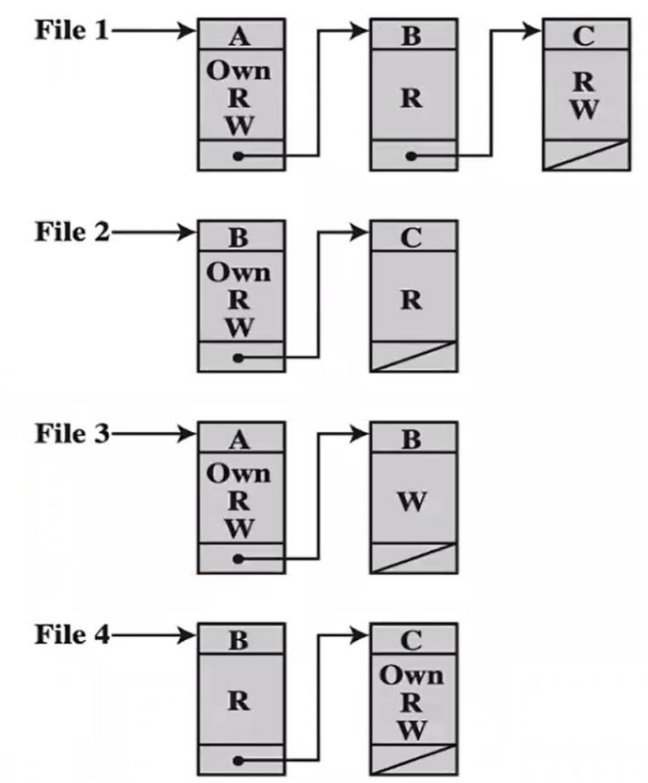
Access Control

Access Control list (ACL)

Wont take up as much memory!
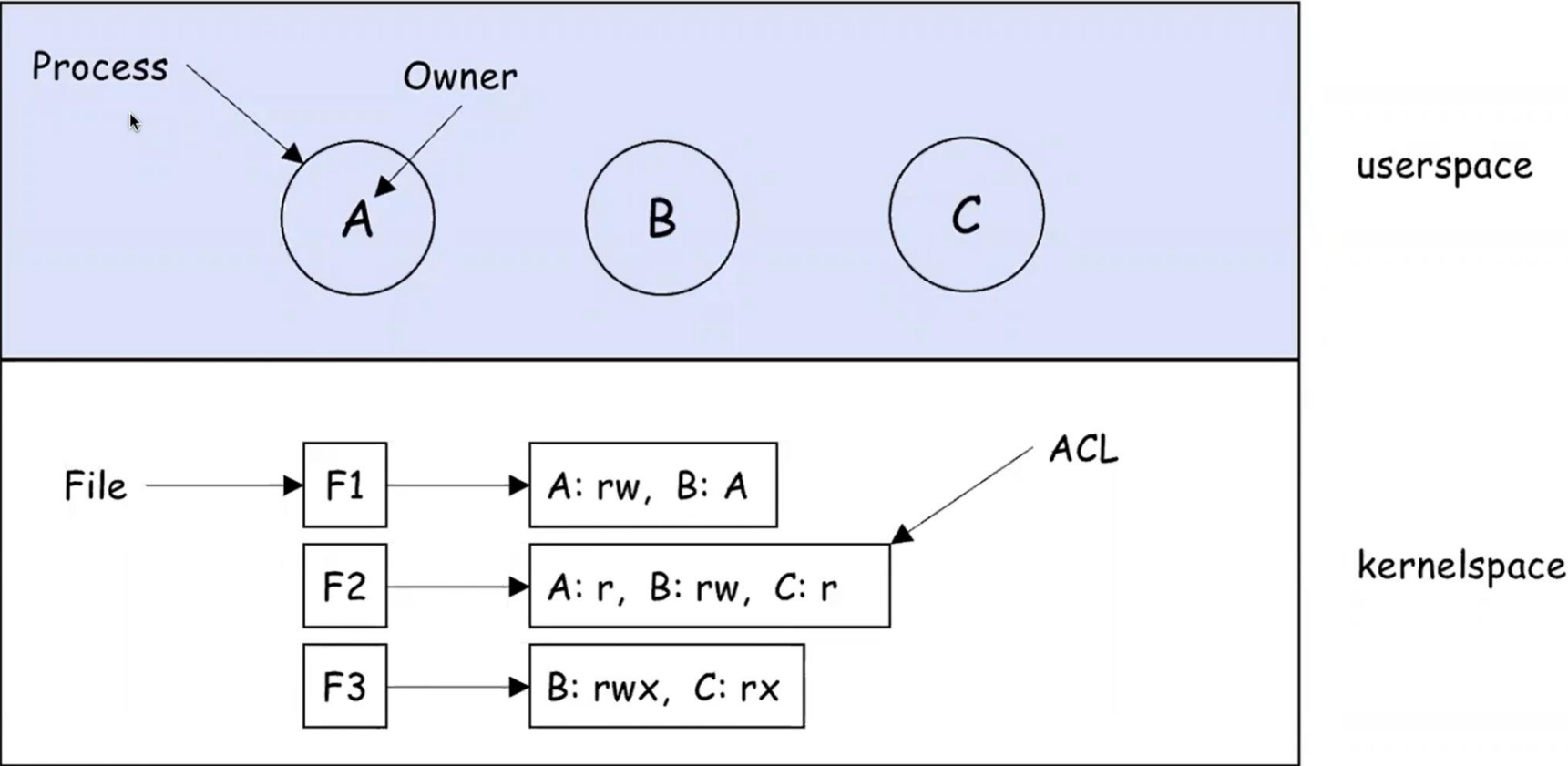
Access Control

Unix File Modes and Permissions

Every Unix file has a set of permissions
that determine whether someone can
read, write, or run the file

```
ls -l ~
ls -l /dev
```

```
[09/13/22]seed@VM:~$ ls -l ~
total 44
drwxr-xr-x 2 seed seed 4096 Nov 24  2020 Desktop
drwxr-xr-x 2 seed seed 4096 Nov 24  2020 Documents
drwxr-xr-x 2 seed seed 4096 Nov 24  2020 Downloads
drwxrwxr-x 2 seed seed 4096 Sep  1 14:37 lab0
drwxr-xr-x 2 seed seed 4096 Nov 24  2020 Music
drwxrwxr-x 2 seed seed 4096 Sep  6 15:23 os-review
drwxr-xr-x 2 seed seed 4096 Nov 24  2020 Pictures
drwxr-xr-x 2 seed seed 4096 Nov 24  2020 Public
drwxrwxr-x 2 seed seed 4096 Aug 25 13:41 shared
drwxr-xr-x 2 seed seed 4096 Nov 24  2020 Templates
drwxr-xr-x 2 seed seed 4096 Nov 24  2020 Videos
```

MONTANA
STATE UNIVERSITY

Unix File Modes and Permissions

Every Unix file has a set of permissions
that determine whether someone can
read, write, or run the file

```
ls -l ~
ls -l /dev
```

Permissions for the file

```
[09/13/22]seed@VM:~$ ls -l ~
total 44
drwxr-xr-x 2 seed seed 4096 Nov 24  2020 Desktop
drwxr-xr-x 2 seed seed 4096 Nov 24  2020 Documents
drwxr-xr-x 2 seed seed 4096 Nov 24  2020 Downloads
drwxrwxr-x 2 seed seed 4096 Sep  1 14:37 lab0
drwxr-xr-x 2 seed seed 4096 Nov 24  2020 Music
drwxrwxr-x 2 seed seed 4096 Sep  6 15:23 os-review
drwxr-xr-x 2 seed seed 4096 Nov 24  2020 Pictures
drwxr-xr-x 2 seed seed 4096 Nov 24  2020 Public
drwxrwxr-x 2 seed seed 4096 Aug 25 13:41 shared
drwxr-xr-x 2 seed seed 4096 Nov 24  2020 Templates
drwxr-xr-x 2 seed seed 4096 Nov 24  2020 Videos
```

MONTANA
STATE UNIVERSITY

Unix File Modes and Permissions

Every Unix file has a set of permissions
that determine whether someone can
read, write, or run the file

```
ls -l ~
ls -l /dev
```

Owner/group information

Permissions for the file

```
[09/13/22]seed@VM:~$ ls -l ~
total 44
drwxr-xr-x 2 seed seed 4096 Nov 24  2020 Desktop
drwxr-xr-x 2 seed seed 4096 Nov 24  2020 Documents
drwxr-xr-x 2 seed seed 4096 Nov 24  2020 Downloads
drwxrwxr-x 2 seed seed 4096 Sep  1 14:37 lab0
drwxr-xr-x 2 seed seed 4096 Nov 24  2020 Music
drwxrwxr-x 2 seed seed 4096 Sep  6 15:23 os-review
drwxr-xr-x 2 seed seed 4096 Nov 24  2020 Pictures
drwxr-xr-x 2 seed seed 4096 Nov 24  2020 Public
drwxrwxr-x 2 seed seed 4096 Aug 25 13:41 shared
drwxr-xr-x 2 seed seed 4096 Nov 24  2020 Templates
drwxr-xr-x 2 seed seed 4096 Nov 24  2020 Videos
```

Unix File Modes and Permissions

Every Unix file has a set of permissions
that determine whether someone can
read, write, or run the file

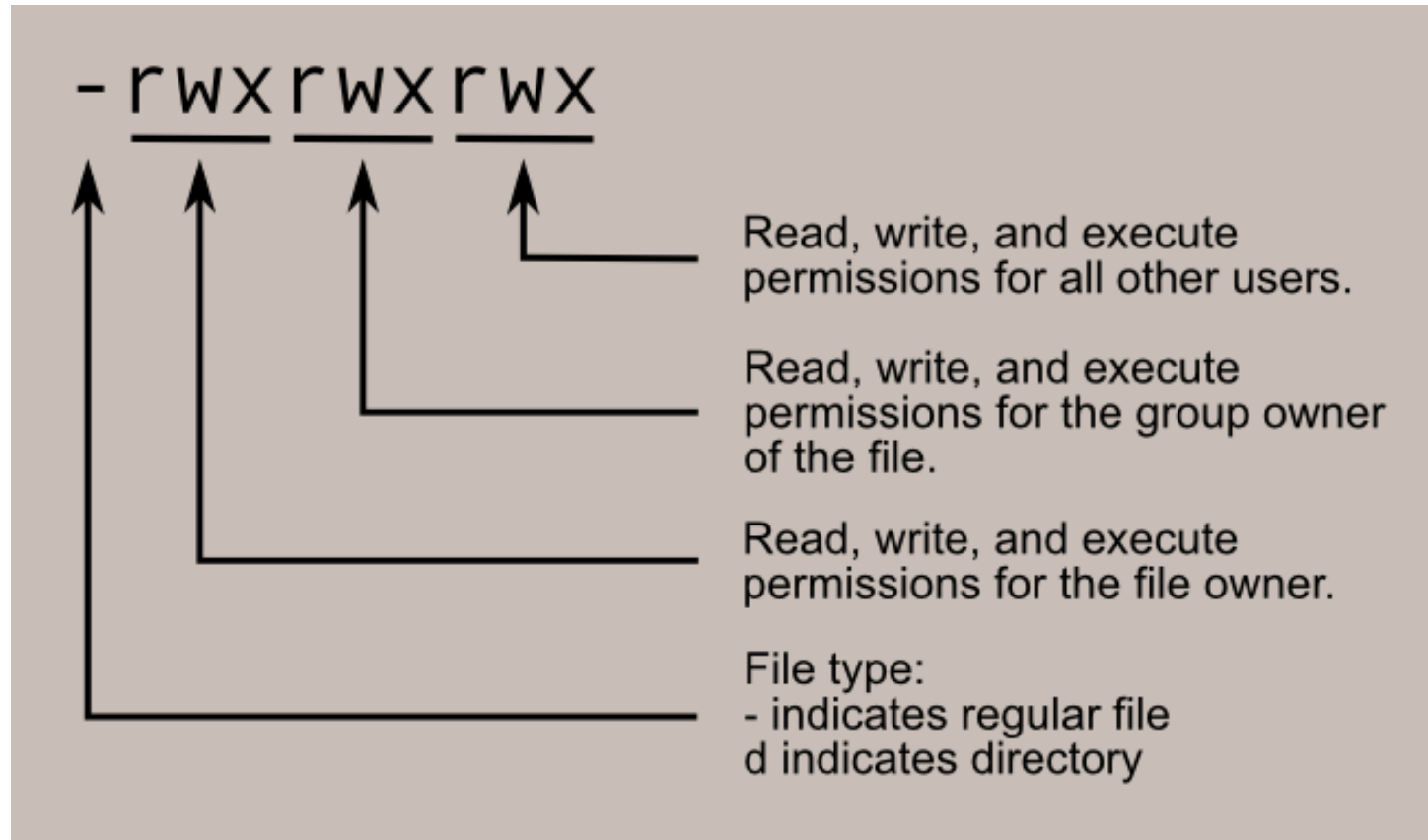```
$ ls -l file
-rw-r--r-- owner group date/time file
```

File permissions (4 parts)
- [file type][user][group][other]

# Unix File Modes and Permissions

File permissions (4 parts)
- [file type][user][group][other]



`- rwx rwx rwx`

Read, write, and execute permissions for all other users.

Read, write, and execute permissions for the group owner of the file.

Read, write, and execute permissions for the file owner.

File type:
- indicates regular file
d indicates directory

# Unix File Modes and Permissions

Suppose you have the following file:

```
$ ls -l F
-rw-rw-r-- B G ... F
```

If user **A** asks to perform some operation **O** on a file object **F**, the OS checks:

- Is **A** the owner of **F**?

13

Unix File Modes and Permissions

Suppose you have the following file:

If user **A** asks to perform some operation **O** on a file object **F**, the OS checks:

- Is **A** the owner of **F**?

    No, B is the owner

Unix File Modes and Permissions

Suppose you have the following file:

```
$ ls -l F
-rw-rw-r--  B  G  ...  F
```

If user **A** asks to perform some operation **O** on a file object **F**, the OS checks:

- Is **A** the owner of **F**?
- Is **A** a member of **F**'s group?

Unix File Modes and Permissions

Suppose you have the following file:

```
$ ls -l F
-rw-rw-r--  B G ... F
```

If user **A** asks to perform some operation **O** on a file object **F**, the OS checks:

- Is **A** the owner of **F**?
- Is **A** a member of **F**'s group?   Suppose G = {B,C,F}

  A is not in F's group

Unix File Modes and Permissions

Suppose you have the following file:

```
$ ls -l F
-rw-rw-r-- B G ... F
```

If user **A** asks to perform some operation **O** on a file object **F**, the OS checks:

- Is **A** the owner of **F**?
- Is **A** a member of **F**'s group?
- Otherwise, what can they do?

Unix File Modes and Permissions

Suppose you have the following file:

```
$ ls -l F
-rw-rw-r-- B G ... F
```

If user **A** asks to perform some operation **O** on a file object **F**, the OS checks:

- Is **A** the owner of **F**?
- Is **A** a member of **F**'s group?
- Otherwise, what can they do?

Everyone can **read** file F

Unix File Modes and Permissions

- [file type][user][group][other]

Suppose **user C** asks to **execute** a **file object F2**. Will they be able to do so?

```
$ ls -l F
-rwxrwxrwx   B  H   ...   F1
-rwxr-xr--   D  G   ...   F2
-rw-r-----   D  H   ...   F3
-rw-rw-rw-   B  G   ...   F4
```

Note:
- Group = G = {A, C, K, M, Q, Z}
- Group = H = {A, B, C, Q}

Unix File Modes and Permissions

- [file type][user][group][other]

Suppose **user C** asks to **execute** a **file object F2**. Will they be able to do so?

```
$ ls -l F
-rwxrwxrwx    B  H   ...    F1
-rwxr-xr--    D  G   ...    F2
-rw-r-----    D  H   ...    F3
-rw-rw-rw-    B  G   ...    F4
```

Note:

- Group = G = {A, C, K, M, Q, Z}
- Group = H = {A, B, C, Q}

Limitations of File-Based Access Control
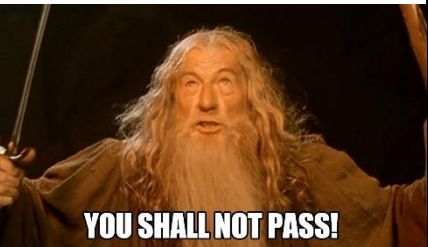
When would a non-privilege user require more power/permissions?

Limitations of File-Based Access Control

When would a non-privilege user require more power/permissions?

Changing password!


```
[seed@VM][~]$ ls -al /etc/passwd
-rw-r--r-- 1 root root 2886 Nov 24 09:12 /etc/passwd

[seed@VM][~]$ ls -al /etc/shadow
-rw-r----- 1 root shadow 1514 Nov 24 09:12 /etc/shadow
```
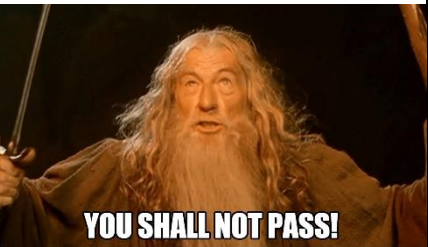
Limitations of File-Based Access Control

When would a non-privilege user require more power/permissions?

Changing password!


```
[seed@VM][~]$ ls -al /etc/passwd
-rw-r--r-- 1 root root 2886 Nov 24 09:12 /etc/passwd

[seed@VM][~]$ ls -al /etc/shadow
-rw-r----- 1 root shadow 1514 Nov 24 09:12 /etc/shadow
```
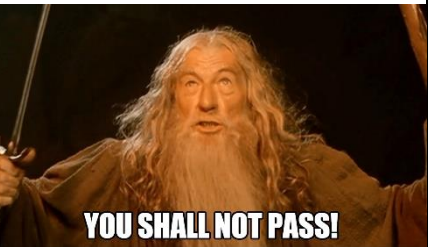
/etc/passwd and **/etc/shadow** hold encrypted passwords for the user, in order to change our password, we will need to have access to those directories

Limitations of File-Based Access Control

When would a non-privilege user require more power/permissions?

Changing password!


```
[seed@VM][~]$ ls -al /etc/passwd
-rw-r--r-- 1 root root 2886 Nov 24 09:12 /etc/passwd

[seed@VM][~]$ ls -al /etc/shadow
-rw-r----- 1 root shadow 1514 Nov 24 09:12 /etc/shadow
```
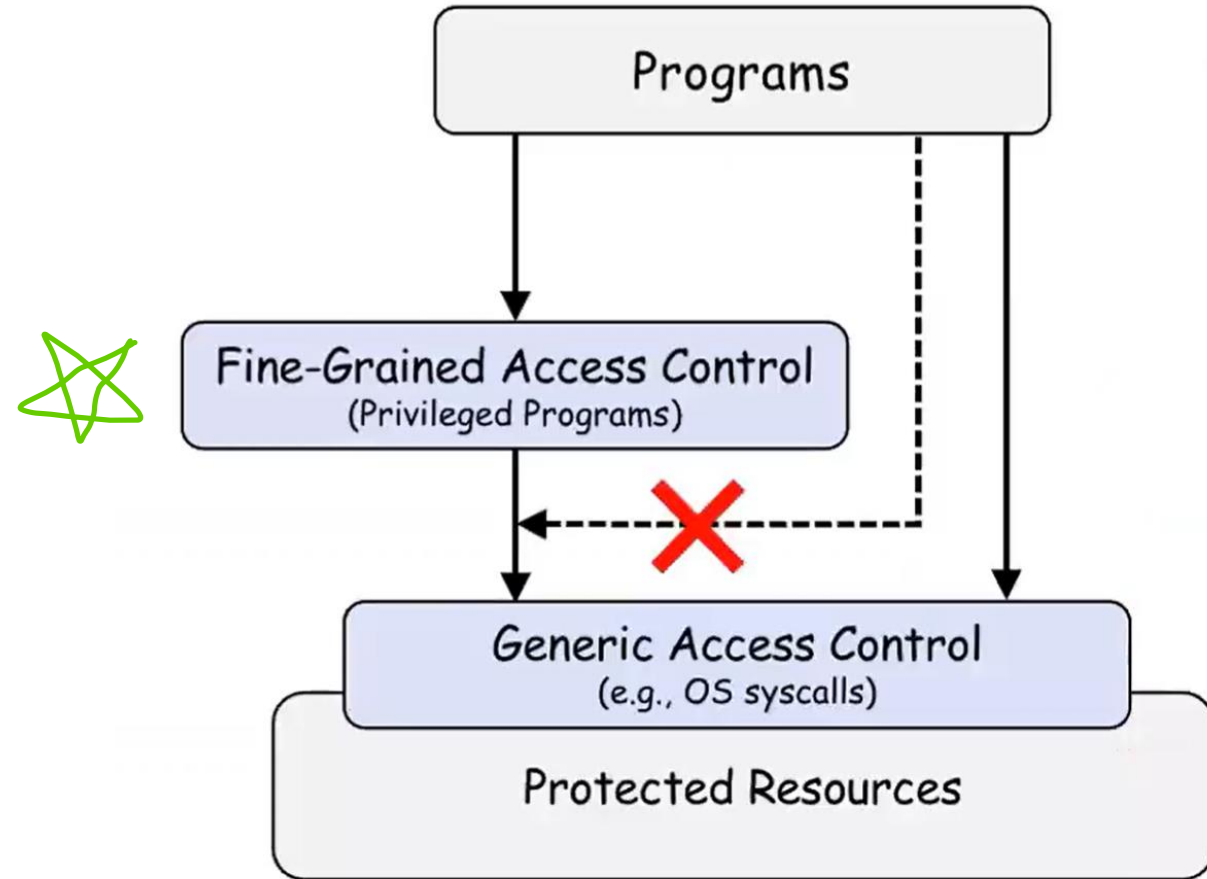
/etc/passwd and **/etc/shadow** hold encrypted passwords for the user, in order to change our password, we will need to have access to those directories

root (aka admin) is the only person that has write permissions!

Limitations of File-Based Access Control

Instead of having a user deal with sensitive actions, lets have a privileged program do it for us!

Types of Privileged Programs

- **Daemons**
  - ➢ Computer program that runs in the background
  - ➢ Needs to run as root or other privileged users

- **Set-UID Programs**
  - ➢ Widely used in UNIX systems
  - ➢ A normal program… but marked with a special bit

The superman story

Superman got tired of saving the city every day

So, he decided to create a "super suit" that would give normal people his powers

Problem: Not all super people are good………

The superman story

Superman got tired of saving the city every day

So, he decided to create a "super suit" that would give normal people his powers

Problem: Not all super people are good………

Super suit 2.0

Super suit with a dope computer
Programmed to perform a specific task
No way to deviate from the pre-programmed task

The superman story
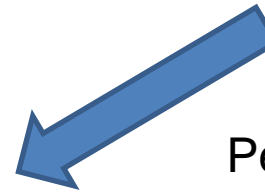

Task: Stop Bowser
1. Fly North
2. Turn left and move forward
3. Punch

Super suit 2.0

People can hop in, and do the specific task to stop bowser

The superman story

Task: Stop Bowser
1. Fly North
2. Turn left and move forward
3. Punch

The superman story

Task: Stop Bowser
1. Fly North
2. Turn left and move forward
3. Punch

The superman story

Task: Stop Bowser
1. Fly North
2. Turn left and move forward
3. Punch

The superman story

Task: Stop Bowser
1. Fly North
2. Turn left and move forward
3. Punch

The superman story

Task: Stop Bowser
1. Fly North
2. Turn left and move forward
3. Punch

This works great! People can only do the predetermined task and don't have control!

The superman story

Task: Stop Bowser
1. Fly North
2. Turn left and move forward
3. Punch

This works great! People can only do the predetermined task and don't have control!

Exploitable?

The superman story

Task: Stop Bowser
1. Fly North
2. Turn left and move forward
3. Punch

Suppose I come along, and I see the power suit

And I decide to flip the suit around

Now what happens???

The superman story

Task: Stop Bowser
1. Fly North
2. Turn left and move forward
3. Punch

Suppose I come along,
and I see the power suit

And I decide to flip the suit around

Now what happens???

The superman story

Task: Stop Bowser
1. Fly North
2. Turn left and move forward
3. Punch

Suppose I come along, and I see the power suit

And I decide to flip the suit around

Now what happens???

The superman story

Task: Stop Bowser
1. Fly North
2. Turn left and move forward
3. Punch

Suppose I come along,
and I see the power suit

And I decide to flip the suit around

Now what happens???

The superman story

Task: Stop Bowser
1. Fly North
2. Turn left and move forward
3. Punch

Suppose I come along,
and I see the power suit

And I decide to flip the suit around

Now what happens???

The superman story

Task: Stop Bowser
1. Fly North
2. Turn left and move forward
3. Punch

Suppose I come along, and I see the power suit

And I decide to flip the suit around

I still followed the steps, but now we have a totally different outcome

My plan was to rob the bank, and I had friends waiting this whole time!

Set-UID In a Nutshell

Set-UID allows a user to run a program with the program owner's privilege
- User runs a program w/ temporarily elevated privileges

Created to deal with inflexibilities of UNIX access control

Example: The `passwd` program

```
[seed@VM][~]$ ls -al /usr/bin/passwd
-rwsr-xr-x 1 root root 68208 May 28  2020 /usr/bin/passwd
```

Set-UID In a Nutshell

Set-UID allows a user to run a program with the program owner's privilege
- User runs a program w/ temporarily elevated privileges

Every process has two User IDs
- Real UID (RUID)– Identifies the **owner** of the process
- Effective UID (EUID)– Identifies **current privilege** of the process

When a normal program is executed
- RUID == EUID

When a Set-UID program is executed
- RUID != EUID
- EUID == ID of the program's owner

If a program owner == root,
The program runs with root privileges

Set-UID In a Nutshell

Set-UID allows a user to run a program with the program owner's privilege
- User runs a program w/ temporarily elevated privileges

Every process has two User IDs
- Real UID (RUID)– Identifies the **owner** of the process
- Effective UID (EUID)– Identifies **current privilege** of the process

When a normal program is executed
- RUID == EUID

When a Set-UID program is executed
- RUID != EUID
- EUID == ID of the program's owner

**If a program owner == root,
The program runs with root privileges**

Set-UID Program Demo

```
[seed@VM][~]$ cp /bin/cat ./mycat
[seed@VM][~]$ sudo chown root mycat
[seed@VM][~]$ ls -al mycat
-rwxr-xr-x 1 root seed 43416 Jan 25 21:15 mycat
```

Change the owner of a file to root

Set-UID Program Demo

```
[seed@VM][~]$ cp /bin/cat ./mycat
[seed@VM][~]$ sudo chown root mycat
[seed@VM][~]$ ls -al mycat
-rwxr-xr-x 1 root seed 43416 Jan 25 21:15 mycat
```

Change the owner of a file to root

```
[seed@VM][~]$ mycat /etc/shadow
mycat: /etc/shadow: Permission denied
```

Running to program (normally)

Set-UID Program Demo

```
[seed@VM][~]$ cp /bin/cat ./mycat
[seed@VM][~]$ sudo chown root mycat
[seed@VM][~]$ ls -al mycat
-rwxr-xr-x 1 root seed 43416 Jan 25 21:15 mycat
```

Change the owner of a file to root

```
[seed@VM][~]$ mycat /etc/shadow
mycat: /etc/shadow: Permission denied
```

Running to program (normally)

```
[seed@VM][~]$ sudo chmod 4755 mycat
[seed@VM][~]$ ls -al mycat
-rwsr-xr-x 1 root seed 43416 Jan 25 21:15 mycat
[seed@VM][~]$ mycat /etc/shadow
root:!:18590:0:99999:7:::
daemon:*:18474:0:99999:7:::
```

Enable the Set-UID bit

We have successfully made a Set-UID program!