**Lecture 24**

**Advance-Permission-SUID-SGID-STICKYBIT-SUDOERS**

**Advance Permissions**

* Last time it was file level permissions.
  + 
* Now,
* 
* These r advance level permissions

Definitions

SUID, GUID, and sticky bit are special permissions in Unix-based operating systems that can be applied to files and directories. Here are brief definitions for each:

**SUID (Set User ID):** When the SUID bit is set on an executable file, the program will run with the privileges of the file's owner, rather than the user who is running the program. This is often used to allow users to perform certain tasks that they would not normally have permission to do.

**GUID (Set Group ID):** When the GUID bit is set on an executable file, the program will run with the privileges of the file's group, rather than the user who is running the program. This is similar to SUID, but applies to group permissions instead of user permissions.

**Sticky Bit:** When the sticky bit is set on a directory, only the owner of a file can delete or rename it within that directory, even if other users have write permissions for the directory. This is often used to protect files that are shared among multiple users, such as in a public directory.

**SUID**

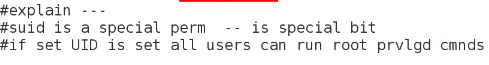
SUID, which stands for "Set User ID", is a Linux/Unix file system attribute that allows users to execute a file with the permissions of the file's owner, instead of with their own permissions. The SUID bit can be set on an executable file, allowing any user to execute the file with the privileges of the file owner.

The purpose of SUID is to enable users to execute specific programs that require special privileges, such as changing system configuration or accessing sensitive data. Without SUID, these tasks can only be performed by privileged users, such as root, which can be a security risk as it provides too much power to a single user.



The SUID bit is commonly used in system-level programs, such as su, passwd, and ping, which require elevated privileges to perform their tasks. By setting the SUID bit on these programs, normal users can execute them without requiring administrator privileges.

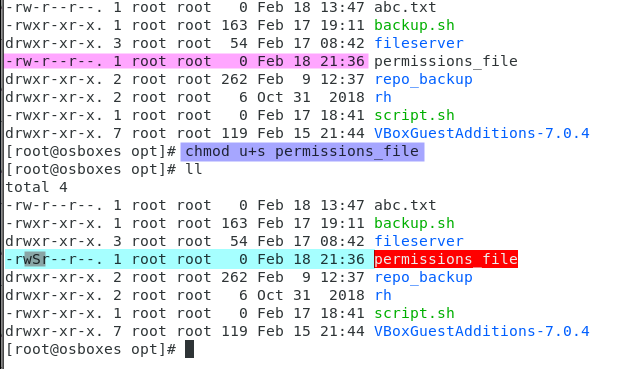
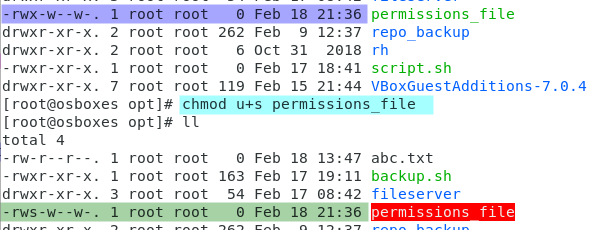
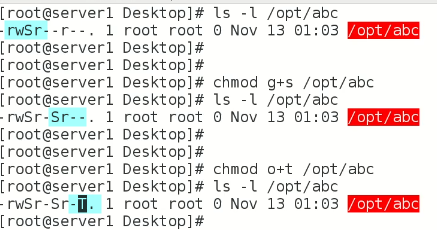
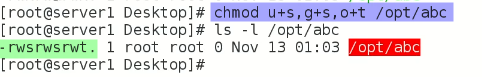
It's important to note that SUID can also be a security risk if it's not used correctly. Care should be taken when setting the SUID bit, as it can potentially give a regular user too much power, leading to potential security vulnerabilities.

* 
* As we know permissions of a file have 3 fields

|  |  |  |
| --- | --- | --- |
| rwx **s** | rwx **s** | rwx **t** |

* S in user/owner field means “SUID” is set
  + Small “s” means SUID + x permission is set
  + Capital “S” means only SUID is set
* S in group field means “SGID” is set
  + Small “s” means SUID + x permission is set
  + Capital “S” means only SUID is set
* T in others field means “stickybit” is set.
  + Small “t” means stickybit + x permission is set
  + Capital “T” means only stickybit is set
* Its bit tricky we will never typw capital “S”” or “T” after chmod 🡪 we will always type small “s” or “t”
* Because if x permission is set the s will automatically turn into small “s” other wise capital “S”, similarly with “t” and “T”

**How to set SUID**

* $ chmod u+s <file>
* 
* 
* Now after changing permission and adding “x” permission the SUID is in small “s”.
* Similarly for “SGID
* $ chmod g+s <file>
* And for stickybit
* $ chmod o+t <file>
* 
* After giving 777 permission,
* 
* SUID, GUID and Stickybit are set with small letters shown because “x” permission was set this time.
* How to set special permissions in octal way,

Special permissions (SUID, GUID, and sticky bit) can be set in octal notation using the following values:

**SUID: 4**

**GUID: 2**

**Sticky bit: 1**

To set special permissions in octal notation, you add the values of the permissions you want to set. For example, to set both SUID and GUID on a file or directory, you would add 4 and 2 to get a value of 6. To set all three special permissions, you would add 4 + 2 + 1 to get a value of 7.

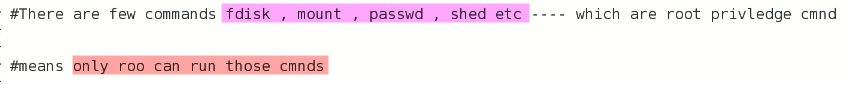
Here are some examples of how to set special permissions using octal notation:

**To set SUID on a file or directory: chmod 4755 filename**

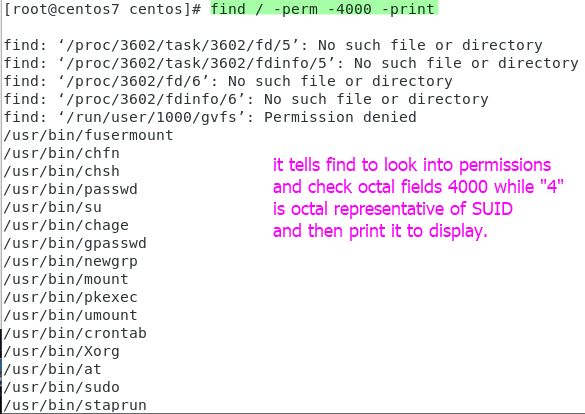
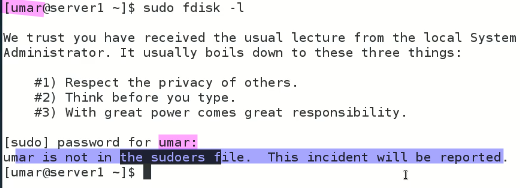
**To set GUID on a file or directory: chmod 2755 filename**

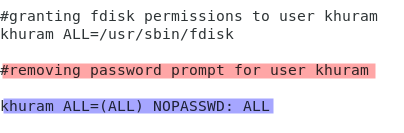
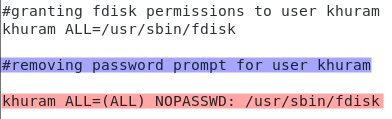
**To set the sticky bit on a directory: chmod 1755 directoryname**

To set all three special permissions on a file or directory: chmod 6777 filename

* 
* How normal user can run this command 🡪 for example fdisk command
* 🡪 set UID for that command.
* Note:- SUID effects commands, programs, and script
* As we can see SUID is not a solution because it gives access to all users and hance it is vulnerable.
* Here comes “SUDOER” in action,

**SUDOER -🡪 very very important topic**

* What is the difference b/w SUID & SUODERS?
* Why a user can changed its password even “passwd” is a root privileged command? 🡪 (interview question)
* Because the **SUID** is set on passwd command
  + Note:- Whenever the password is set through “passwd” command, it updates /etc/shadow file too.
* 
* To check how many SUIDs are set on how many files,
  + Find /-perm -4000 -print
  + 
  + For stickybit “-1000” and “-7000” for all special permissions
  +  this command will change the permission with find command.
* How to configure Sudoer? For a specific user so that it may use special permissions instead of all other normal users.
* $ /etc/suoders
* 
* Avoid directly editing suoder file
* $ visudo --? Is the command or tool
* At the end of file add this row 🡪 it is called adding a user into SUDOERS
* 
* <user> ALL=<absolute path of the command **e.g /usr/sbin/fdisk**> + <flag>
* If we remove SUID 🡪 $ chmod u-s /usr/sbin/fdisk
* $ fdisk -l will not run even the user is in SUDOERS.
* Now that specific user must tell that it is in SUDOERS and must read SUDOERS
* How to validate?
* **$ sudo fdisk -l 🡪** use sudo before command 🡪 it will prompt for password 🡪 after validating the password the command will run.
* 
* If a user which is not in SUODERS try to open a root privileged command
* 
* It will not open it and gives warning.
* TIP:-
* In Ubuntu “root” is disabled and “Sudo” is configured that’s why we always type Sudo before every critical command.
* A picture containing text

  Description automatically generated
* **Task**
* *Is it possible to disable password prompt for a normal user who have SUOD access? If yes what is the procedure?*
* *Open $ visudo and add following code below user privileges.*
* 
* Instead of ALL in the end absolute path of that command may can be given **/usr/sbin/fdisk**
* 
* Explanation

**username:** Replace this with the actual username of the user that you want to modify.

**ALL:** Specifies that the user can run the specified command(s) on any host.

**=(ALL):** Specifies that the user can run the specified command(s) as any user or group.

**NOPASSWD:**  Specifies that the user will not be prompted to enter their password when using sudo.

**ALL:** Specifies that the user can run any command with sudo privileges.