

Case Study: UNIX (*)

- Domain switching
 - If user mode process in domain = (UID,GID) makes a system call it switches to kernel mode and also switches domain (usually to UID = 0 = root)
 - Necessary since system calls may need to modify or access resources which would normally not be accessible directly by user processes
 - BUT system first checks domain = (UID,GID) to see if the system call can be allowed in the first place (e.g. *chown()* system call is only allowed if UID=0)
 - SETUID and SETGID concept
 - Normal operation:
 - If process in domain (UID,GID) does *exec(file, ...)* the ensuing process is also in domain (UID,GID)
 - SETUID operation:
 - Executable file object, *file*, has a special execute bit set which defines an *Effective UID*, EUID = fUID associated with file
 - If process in domain (UID,GID) does *exec(file, ...)* the ensuing process is then in the domain (EUID,GID). Process has switched protection domain from (UID, GID) to (EUID,GID) and since EUID = fUID the process obtains fUID access rights
 - SETGID operation: Analog to SETUID operation, but uses EGID rather EUID
 - Why? Allows users to gain controlled access to protected objects which would otherwise be denied

3/02/00

COS214, Roberto Togneri, E&E Eng, Univ. of Western Australia

7.17

Use of SETUID for the /bin/passwd program:

/bin/passwd program is used to change user passwords and requires read/write access to /etc/passwd, but:

/etc/passwd cannot have read/write access allowed for user mode processes
/bin/passwd process will inherit the access rights of invoking user

∴ The passwd process, like other user processes, cannot read/write the /etc/passwd file

Solution? Use SETUID as follows:

```
-rws--x--x 1 root system ... .. /bin/passwd
```

```
-rw-r--r-- 1 root system ... .. /etc/passwd
```

When user process (UID = 100, GID = 20) does *exec("/bin/passwd", ...)* the ensuing passwd process runs in the domain (EUID = root = 0, GID = 20) and hence has rw- access to the file /etc/passwd!

Problem? If /bin/passwd is badly designed user can exploit this to modify or corrupt the /etc/passwd file