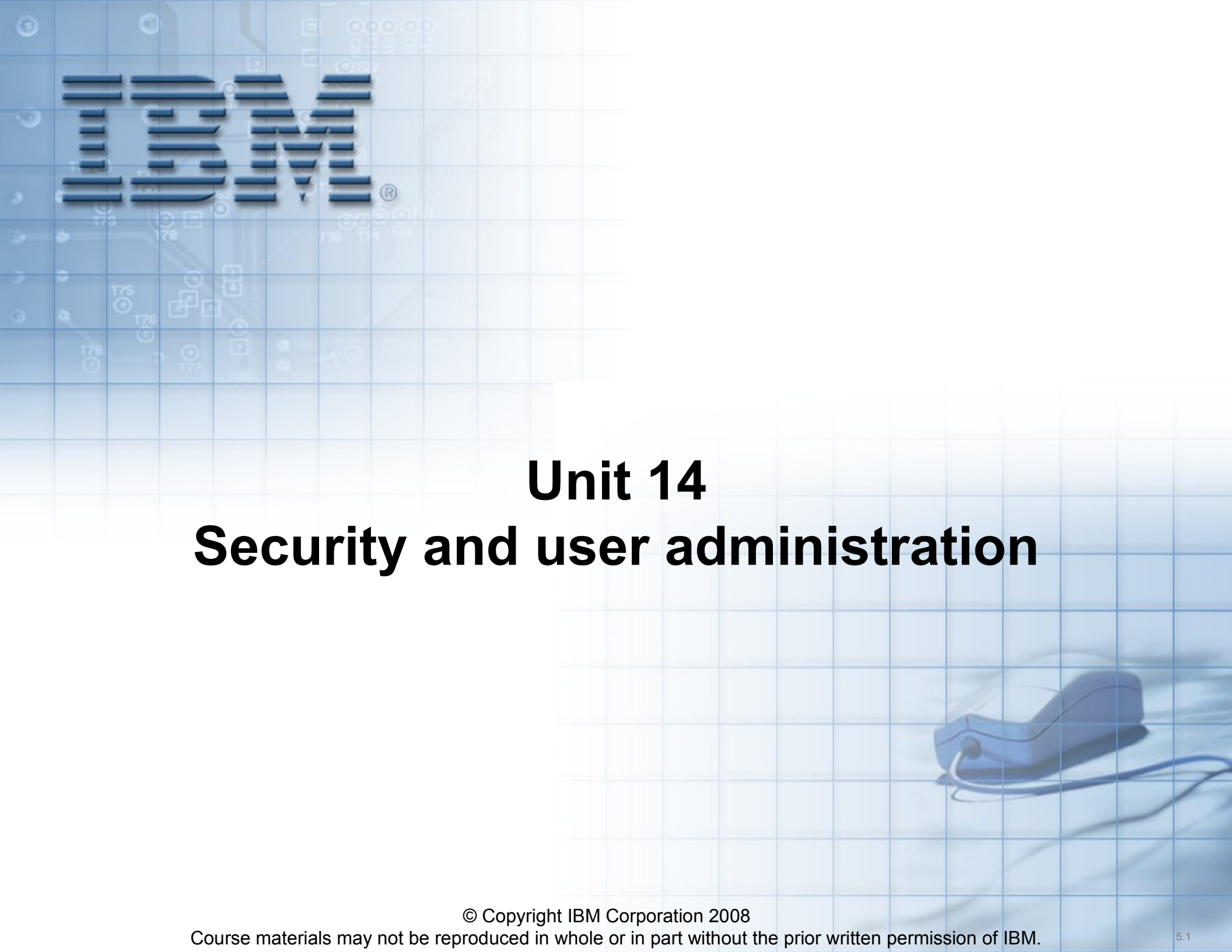


Unit 14
Security and user administration

Unit objectives

After completing this unit, you should be able to:

- Define the concepts of users and groups, and explain how and when these should be allocated on the system
- Describe ways of controlling root access on the system
- Explain the uses of SUID, SGID, and SVTX permission bits
- Administer user accounts and groups
- Identify the data files associated with users and security

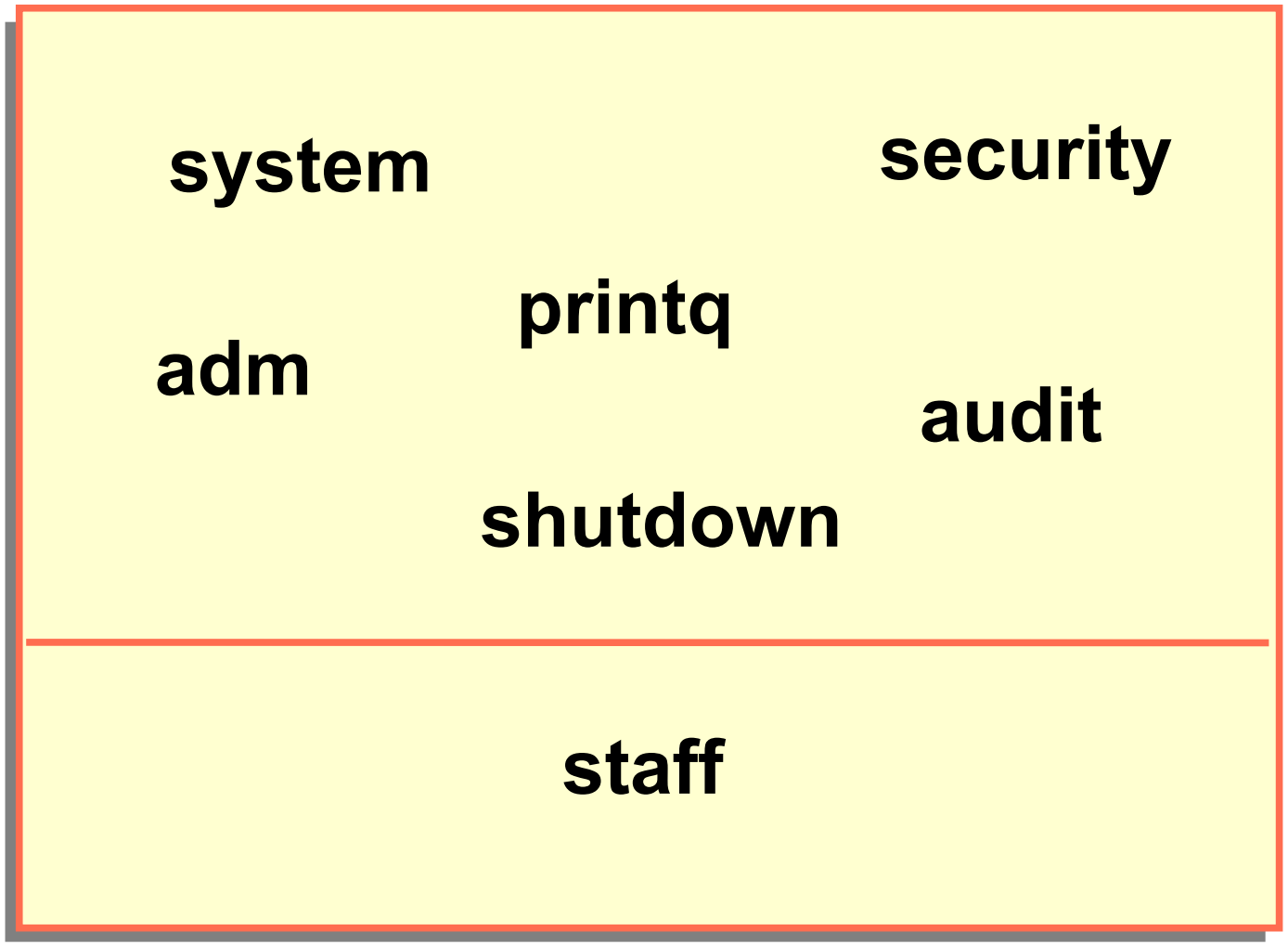
User accounts

- Each user has a unique name, numeric ID, and password
- File ownership is determined by a numeric user ID
- The owner is usually the user who created the file, but ownership can be transferred by **root**
- Default users:
  - **root** Superuser
  - **adm**, **sys**, **bin**, ... IDs that own system files but cannot be used for login

Groups

- A group is a set of users, all of whom need access to a given set of files.
- Every user is a member of at least one group and can be a member of several groups.
- The user has access to a file if any group in the user's groupset provides access. To list the groupset, use the **`groups`** command.
- The user's real group ID is used for file ownership on creation. To change the real group ID, use the **`newgrp`** command.
- Default groups:
  - System administrators: **system**
  - Ordinary users: **staff**

Group hierarchy

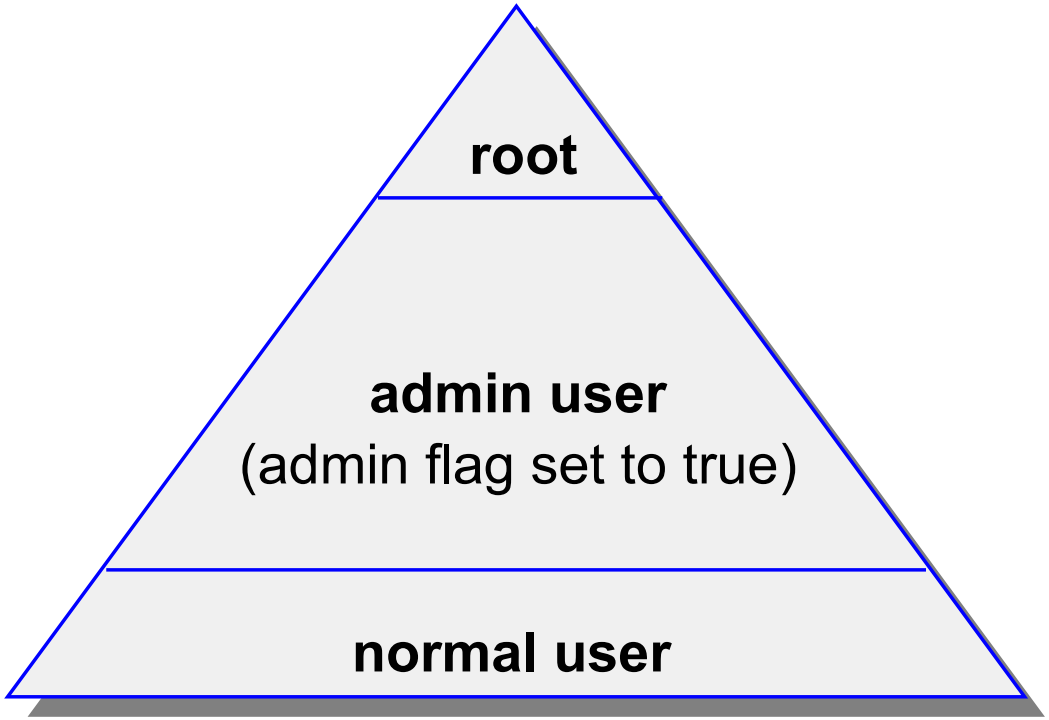


Rights to administrative functions

Ordinary user

User hierarchy

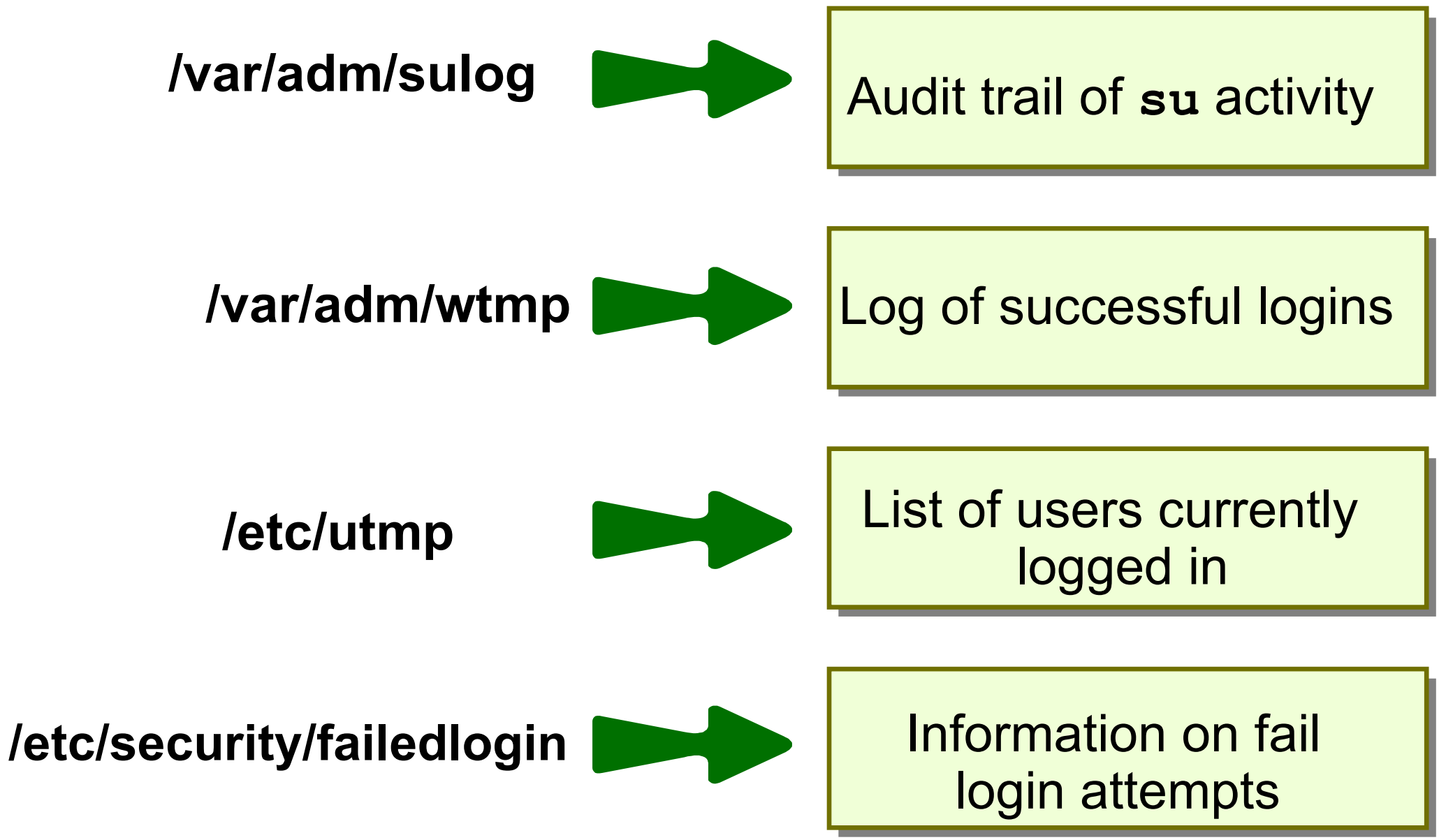
- To protect important users and groups from members of the **security** group, AIX has **admin users** and **admin groups**
- Only **root** can add, remove, or change an **admin user** or **admin group**
- Any user on the system can be defined as an **admin user** regardless of the group they are in



Controlling access to the root account

- Restrict access to privileged logins
- **root**'s passwords should be changed on an unannounced schedule by the system administrator
- Assign different **root** passwords to different machines
- System administrators should always login as themselves first and then **`su`** to **root** instead of logging in as **root**. This helps provide an audit trail for **root** usage
- Do not include unsecured directories in **root**'s **`PATH`**

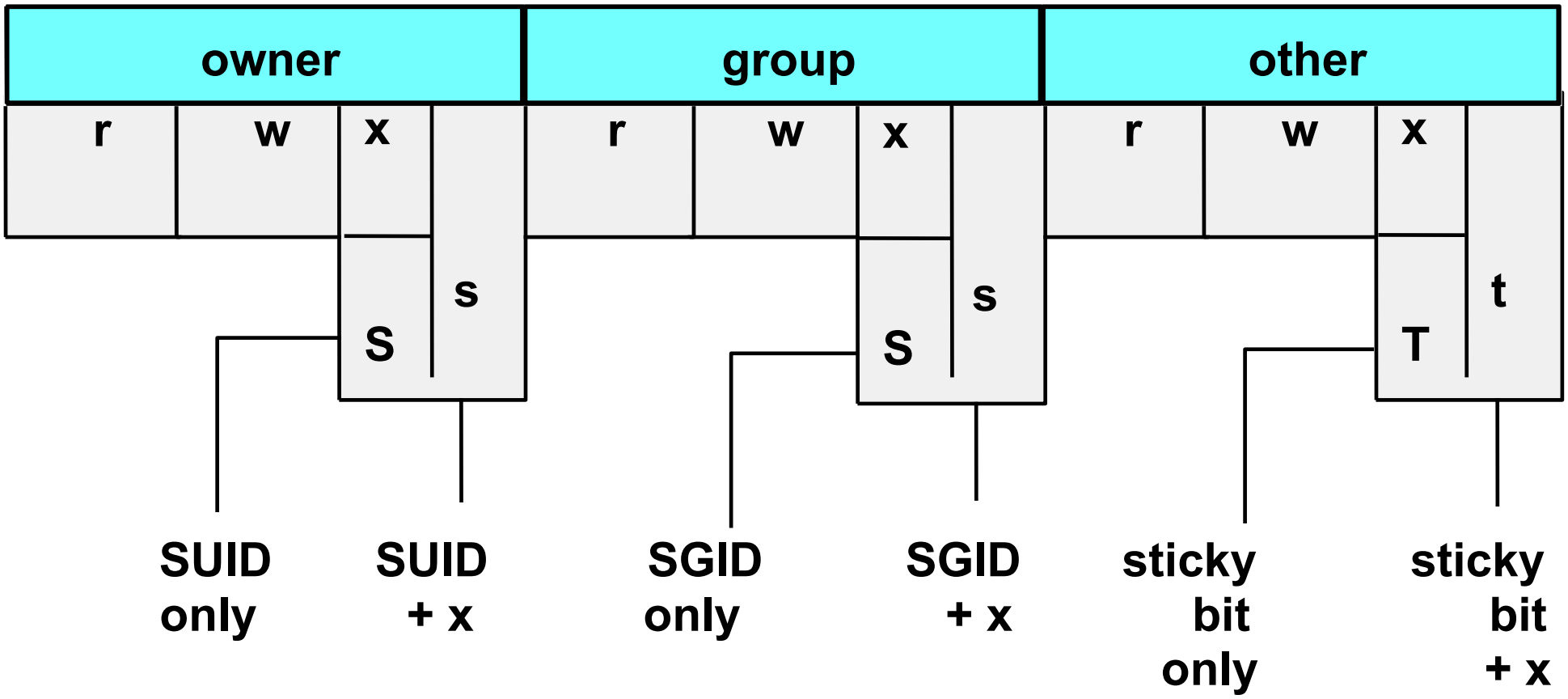
Security logs



File/Directory permissions

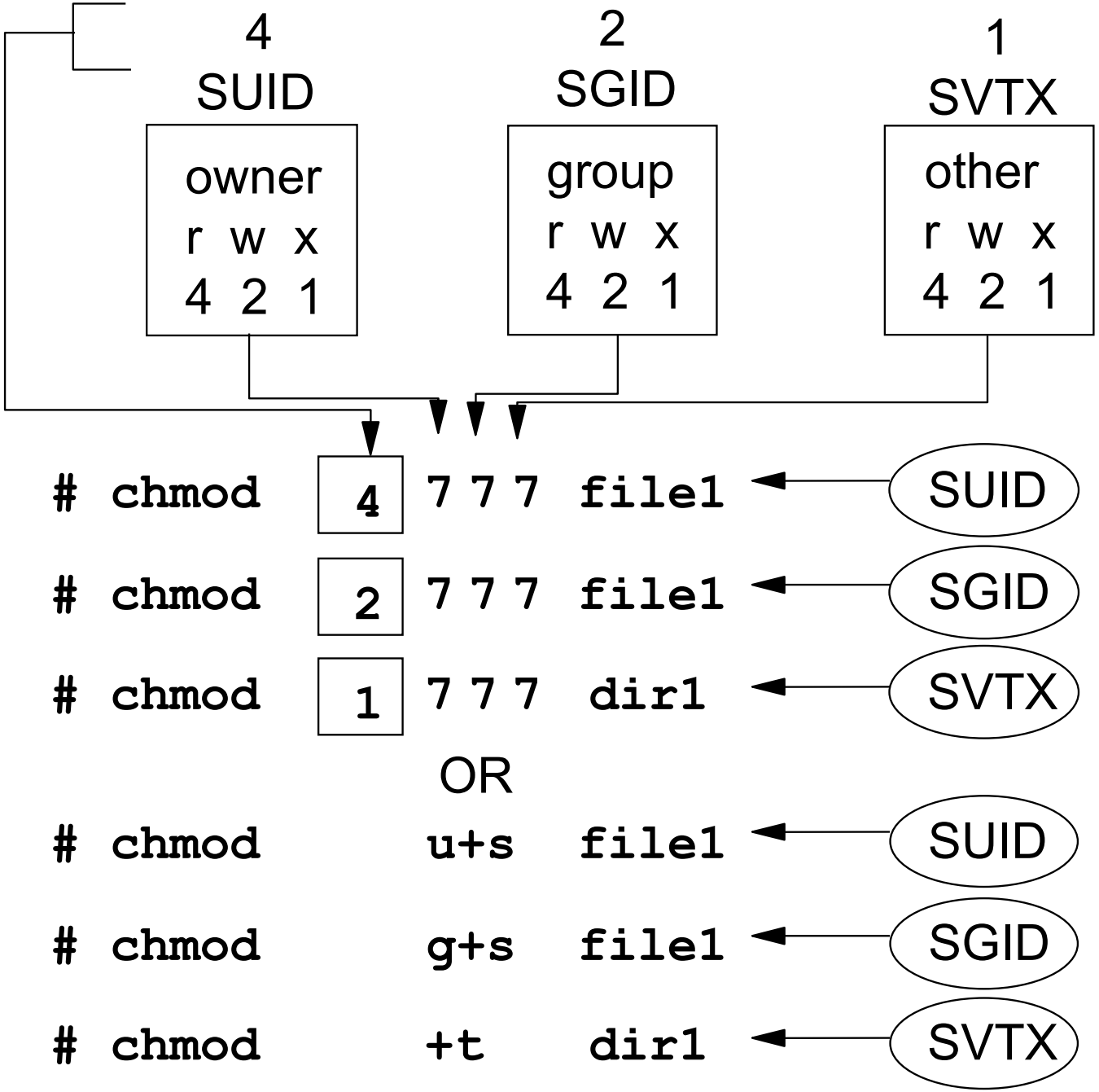
File	Perm. Bit	Directory
Read content of file	r	List content of directory
Modify content of file	w	Create and remove files in directory
Use file name to execute as a command	x	Give access to directory
Run program with effective UID of owner	SUID	--------
Run program with effective GID of group	SGID	Files created in directory inherit the same group as the directory
--------	SVTX	Must be owner of files to delete files from directory

Reading permissions




```
# ls -ld  /usr/bin/passwd  /usr/bin/crontab  /tmp

-r-sr-xr-x    root security   ...      /usr/bin/passwd
-r-sr-sr-x    root cron       ...      /usr/bin/crontab
drwxrwxrwt    bin  bin        ...      /tmp
```

Changing permissions



umask

- The **umask** governs permissions on new files and directories
- System default **umask** is 022
- A **umask** of 027 is recommended
- If the **umask** value is set to 022, then any ordinary files or directories created inherit the following permissions:
  - Ordinary file: **`rw-r--r--`**
  - Directory: **`rwxr-xr-x`**
- **/etc/security/user** specifies default and individual user **umask** values

Changing ownership

The **chown** command:

```
# chown fred file1
```

The **chgrp** command:

```
# chgrp staff file1
```

Changing both user and group ownership:

```
# chown fred:staff file1
# chown fred.staff file1
```

Role based access control (RBAC)

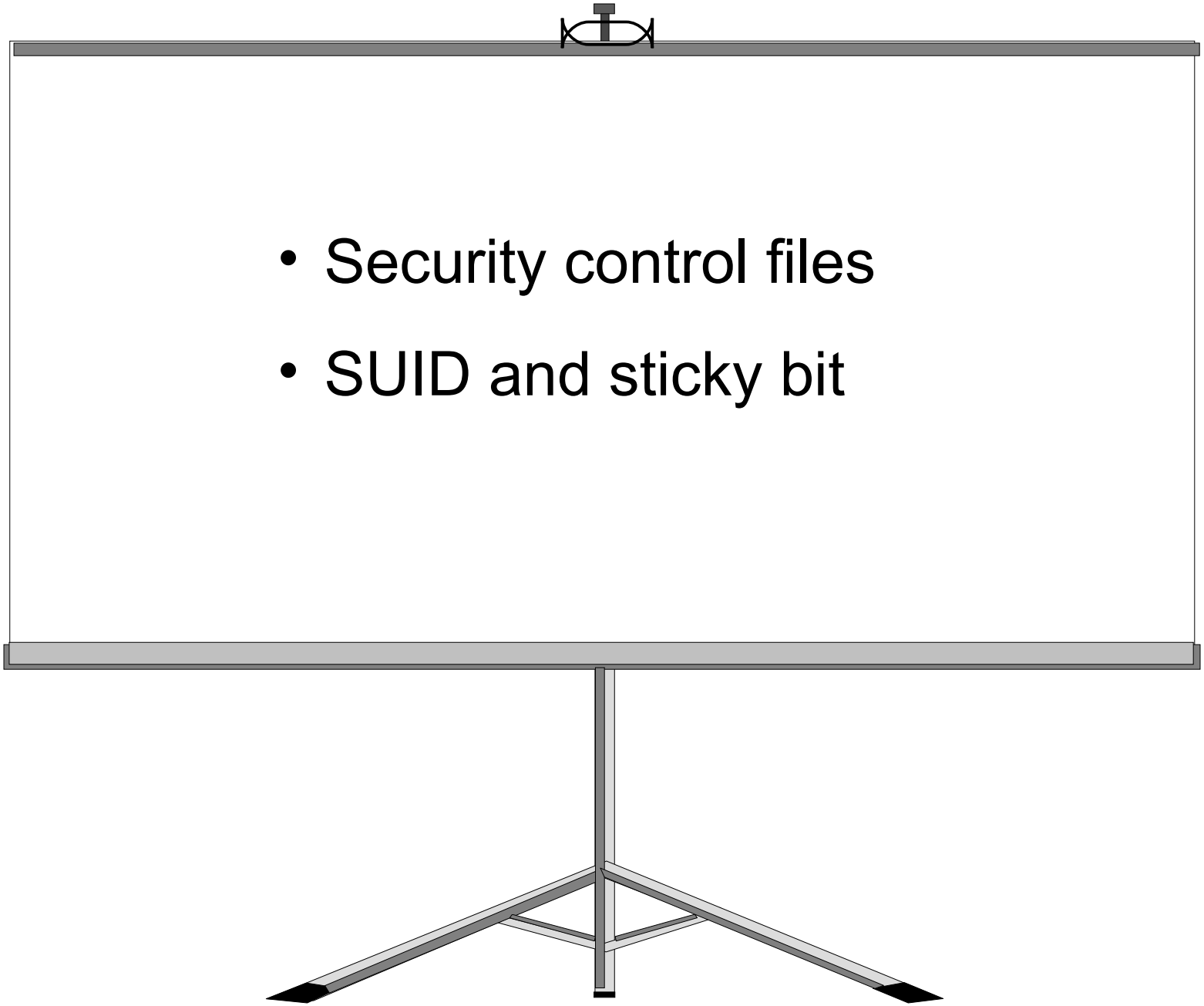
- Fine grained delegation of authority
  - Roles assigned as an attribute of the user or group
- Legacy RBAC (AIX V4.2+):
  - User space implementation
  - Role assignment alone was insufficient
- Enhanced RBAC (AIX 6.1):
  - Covers user and kernel space
  - Effective role assignment without additional configuration
  - AIX 6.1 SP1 provides 10 predefined roles
- User can activate/inactivate roles as needed
  - Create subshell with role in effect:

```
$ swrole  SysBoot
```

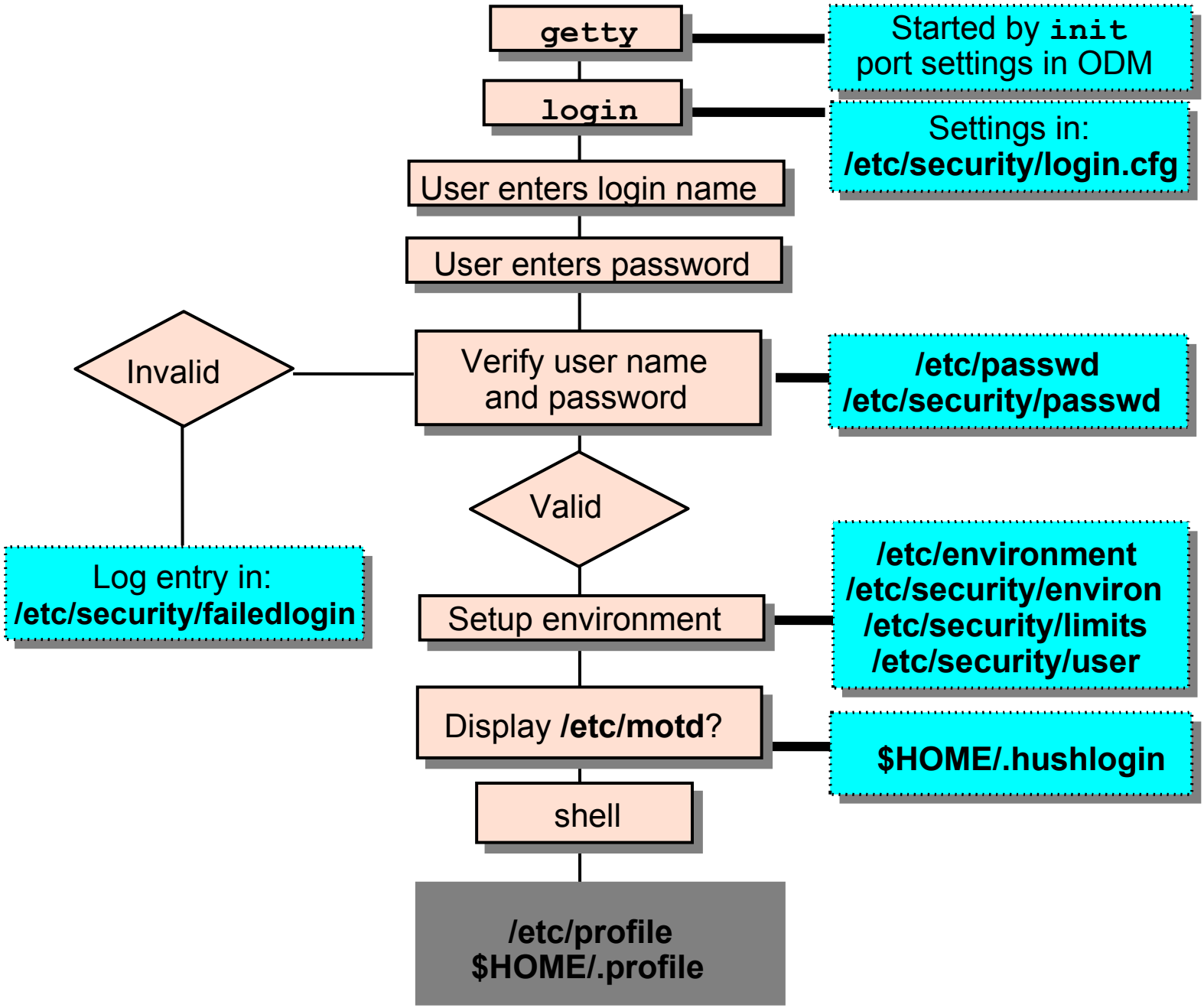
Predefined enhanced RBAC roles

- isso - Information System Security Officer
- sa - System Administrator
- so – System Operator
- AccountAdmin - User and Group Account Administration
- BackupRestore -Backup and Restore Administration
- DomainAdmin - Remote Domain Administration
- FSAdmin - File System Administration
- SecPolicy - Security Policy Administration
- SysBoot - System Boot Administration
- SysConfig - System Configuration

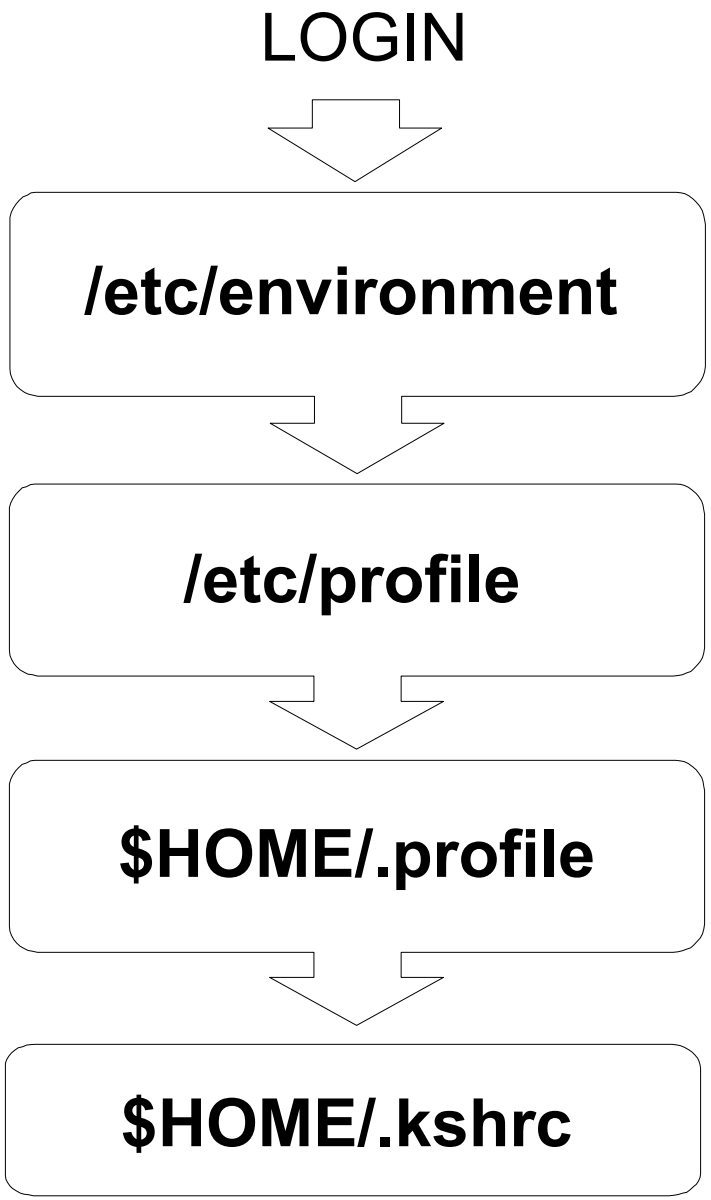
Exercise 15: Security files



Login sequence



User initialization process



Establishes base environment
sets **PATH**, **TZ**, **LANG**, and **NLSPATH**

Shell script run at all logins
sets **TERM**, **MAILMSG**, and **MAIL**

User's personal file to customize their environment
PATH, **ENV**, **PS1**

User's personal file to customize the Korn shell environment
set –o vi, **alias**

Security and users

```
# smit security
```

```
                        Security & Users

Move cursor to desired item and press Enter.
Users
Groups
Passwords
Login Controls
Roles
PKI
LDAP
Role Based Access Control (RBAC)
Trusted Execution

F1=Help    F2=Refresh   F3=Cancel   F8=Image
F9=Shell   F10=Exit     Enter=Do
```

SMIT users

```
# smit users
```

```
                          Users

Move cursor to desired item and press Enter.

Add a User
Change a User's Password
Change / Show Characteristics of a User
Lock / Unlock a User's Account
Reset User's Failed Login Count
Remove a User
List All Users


F1=Help      F2=Refresh    F3=Cancel    F8=Image
F9=Shell     F10=Exit      Enter=Do
```

List all users

The **lsuser** command:

lsuser [-c | -f] [-a attribute …] {ALL | username …}

Example:

```
# lsuser -a id home ALL
root id=0 home=/
daemon id=1 home=/etc
bin id=2 home=/bin
...
john id=200 home=/home/john
...
```

Add a user to the system

```
# smit mkuser
```

```
                              Add a User

...
                                                [Entry Fields]

* User NAME                                    [ ]
  User ID                                      [ ]                        #
  ADMINISTRATIVE USER?                          false                     +
  Primary GROUP                                [ ]                        +
  Group SET                                    [ ]                        +
  ADMINISTRATIVE GROUPS                        [ ]                        +
  ROLES                                        [ ]                        +
  Another user can SU TO USER?                  true                      +
  SU GROUPS                                    [ALL]                      +
  HOME directory                               [ ]
  Initial PROGRAM                              [ ]
  User INFORMATION                             [ ]
  EXPIRATION date (MMDDhhmmyy)                 [0]
  Is this user ACCOUNT LOCKED?                  false                     +
[MORE ...37]
...
```

Change / Show Characteristics of a User

```
# smit chuser
```

```
                Change / Show Characteristics of a User

...
                                                [Entry Fields]

* User NAME                                      george
  User ID                                       [206]                    #
  ADMINISTRATIVE USER?                  false                        +
  Primary GROUP                                 [staff ]                 +
  Group SET                                     [staff,security]         +
  ADMINISTRATIVE GROUPS                         [ ]                      +
  ROLES                                 [ ]                          +
  Another user can SU TO USER?                   true                    +
  SU GROUPS                                     [ALL]                    +
  HOME directory                                [/home/george ]
  Initial PROGRAM                               [/usr/bin/ksh ]
  User INFORMATION                              [ ]
  EXPIRATION date (MMDDhhmmyy)                  [0]
  Is this user ACCOUNT LOCKED?                   false                   +
[MORE ...37]
...
```

Remove a user from the system

- The **rmuser** command or SMIT can be used to delete a user from the system.

```
# rmuser -p team01
```

- When you remove a user, that user's home directory is not deleted. Therefore, you must remember to manually *clean up* the directories of users you remove. (Remember to backup important files first!)

```
# rm -r /home/team01
```

Passwords

- A new user ID cannot be used until a password is assigned
- There are two commands available for making password changes:

```
# passwd [username]
```

```
# pwdadm username
```

- SMIT invokes the **passwd** command
- An ordinary user can use the **passwd** command to change own password
- Only **root** or member of **security** group can change password of another user

Regaining root's password

- Boot from CD-ROM, NIM, or a bootable tape
- Select option 3: **Start Maintenance Mode for System Recovery** from the **Installation and Maintenance** menu
- Follow the options to activate the **root** volume group and obtain a shell
- Once a shell is available, execute the **`passwd`** command to change **root**'s password
- Enter the following command:
  `# sync ; sync`
- Reboot the system

SMIT groups

```
# smit groups
```

```
                          Groups

Move cursor to desired item and press Enter.

List All Groups
Add a Group
Change / Show Characteristics of a Group
Remove a Group




F1=Help      F2=Refresh     F3=Cancel     F8=Image
F9=Shell     F10=Exit       Enter=Do
```

List all groups

The **lsgroup** command:

lsgroup [-c | -f] [-a attribute …] {ALL | groupname …}

Example:

```
# lsgroup ALL
system id=0 admin=true users=root,test2 registry=compat
staff id=1 admin=false users=ipsec,team01,team02,team03,
team04,team05,test1,daemon registry=compat
bin id=2 admin=true users=root,bin registry=compat
sys id=3 admin=true users=root,bin,sys registry=compat
adm id=4 admin=true users=bin,adm registry=compat
uucp id=5 admin=true users=uucp,nuucp registry=compat
...
ipsec id=200 admin=false users= registry=compat
```

Add a Group

```
# smit mkgroup
```

```
                          Add a Group

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                          [Entry Fields]

* Group NAME                              [support]
  ADMINISTRATIVE group?                   false           +
  Group ID                                [300]           #
  USER list                               [fred,barney]   +
  ADMINISTRATOR list                      [fred]          +
  Projects                                [ ]             +
  Initial Keystore Mode                   [ ]             +
  Keystore Encryption Algorithm           [ ]             +
  Keystore Access                         [ ]             +

F1=Help        F2=Refresh       F3=Cancel       F4=List
F5=Reset       F6=Command       F7=Edit         F8=Image
F9=Shell       F10=Exit         Enter=Do
```

Change / remove groups

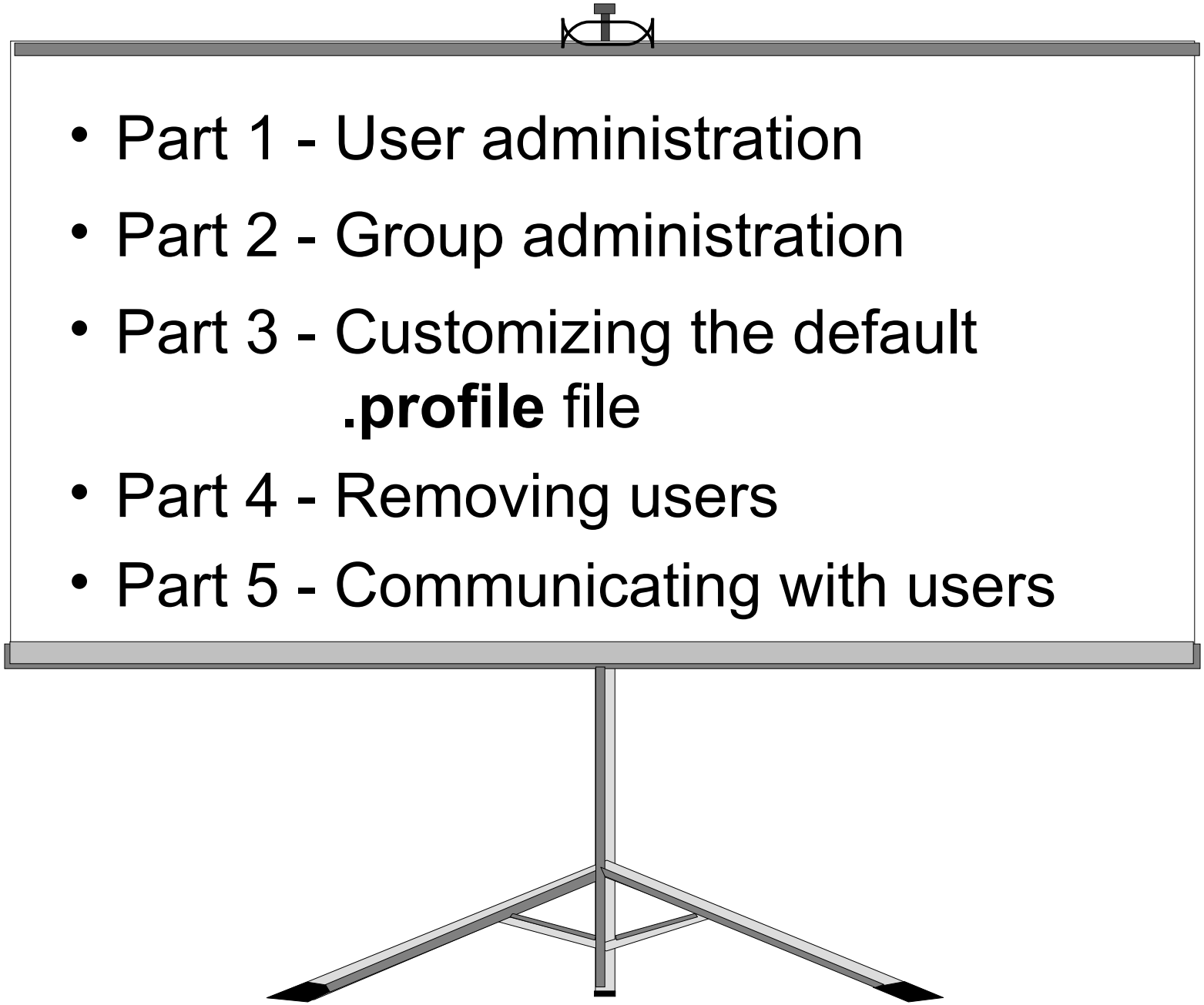
```
# smit chgroup
```

```
                        Change Group Attributes

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                          [Entry Fields]
  Group NAME                              [Support]
  Group ID                                [300]                          #
  ADMINISTRATIVE group?                   False                          +
  USER list                               [fred, barney, wilma]          +
  ADMINISTRATOR list                      [fred]                         +
  Projects                                [ ]                            +
  Initial Keystore Mode                   [ ]                            +
  Keystore Encryption Algorithm           [ ]                            +
  Keystore Access                         [ ]                            +

F1=Help           F2=Refresh          F3=Cancel       F4=List
F5=Reset          F6=Command          F7=Edit         F8=Image
F9=Shell          F10=Exit            Enter=Do
```

Message of the day

- The file /**etc/motd** contains text that is displayed every time a user logs in
- This file should only contain information necessary for the users to see
- If the **$HOME/.hushlogin** file exists in a user's home directory, then the contents of the **/etc/motd** file are not displayed to that user

Exercise 16: User administration (parts 1-5)



- Part 1 - User administration
- Part 2 - Group administration
- Part 3 - Customizing the default **.profile** file
- Part 4 - Removing users
- Part 5 - Communicating with users

Security files

- Files used to contain user attributes and control access:
  - **/etc/passwd** Valid users (not passwords)
  - **/etc/group** Valid groups
  - **/etc/security** Directory not accessible to normal users
  - **/etc/security/passwd** User passwords
  - **/etc/security/user** User attributes, password restrictions
  - **/etc/security/group** Group attributes
  - **/etc/security/limits** User limits
  - **/etc/security/environ** User environment settings
  - **/etc/security/login.cfg** Login settings

/etc/passwd file

```
# cat /etc/passwd
```

root:!:0:0::/:/usr/bin/ksh
daemon:!:1:1::/etc:
bin:!:2:2::/bin:
sys:!:3:3::/usr/sys:
adm:!:4:4::/var/adm:
uucp:!:5:5::/usr/lib/uucp:
guest:!:100:100::/home/guest:
nobody:!:4294967294:4294967294::/:
lpd:!:9:4294967294::/:
lp:*:11:11::/var/spool/lp:/bin/false
invscout:*:6:12::/var/adm/invscout:/usr/bin/ksh
snapp:*:200:13:snapp login user:/usr/sbin/snapp:/usr/sbin/snappd
nuucp:*:7:5:uucp login user:/var/spool/uucppublic:/usr/sbin/uucp/uucico
ipsec:*:201:1::/etc/ipsec:/usr/bin/ksh
esaadmin:*:811:0::/home/esaadmin:/usr/bin/ksh
john:!:200:0:x7560 5th floor:/home/john:/usr/bin/ksh
bill:*:201:1::/home/bill:/usr/bin/ksh

/etc/security/passwd file

```
# cat /etc/security/passwd
```

```
root:
        password = 92t.mzJBjlfbY
        lastupdate = 885485990
        flags =

daemon:
        password = *

bin:
        password = *
...

john:
        password = q/gD6q.ss21x.
        lastupdate = 884801337
        flags = ADMCHG,ADMIN,NOCHECK
```

/etc/security/user file (1 of 2)

```
# cat /etc/security/user
```

```
default:
        admin = false
        login = true
        su = true
        daemon = true
        rlogin = true
        sugroups = ALL
        admgroups =
        ttys = ALL
        auth1 = SYSTEM
        auth2 = NONE
        tpath = nosak
        umask = 022
        expires = 0
...
```

/etc/security/user file (2 of 2)

```
default
...
        SYSTEM = "compat"
        logintimes =
        pwdwarntime = 0
        account_locked = false
        loginretries = 0
        histexpire = 0
        histsize = 0
        minage = 0
        maxage = 0
        maxexpired = -1
        minalpha = 0
        minother = 0
        minlen = 0
        mindiff = 0
        maxrepeats = 8
        dictionlist  =
        pwdchecks =
```

Group files

```
# more /etc/group
```

```
system:!:0:root,john
staff:!:john
bin:!:2:root,bin
sys:!:3:root,bin,sys
...
usr:!:100:guest
accounts:!:200:john
...
```

```
# more /etc/security/group
```

```
system:
            admin=true
staff:
            admin=false
accounts:
            admin=false
            adms=john
            projects=system
```

/etc/security/login.cfg file

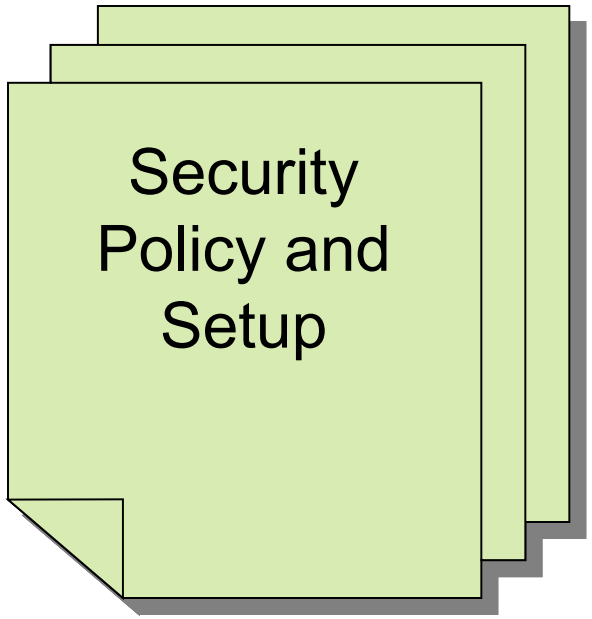
```
default:
  herald =“Authorized use only.\n\rlogin:"
  logintimes =
  logindisable = 0
  logininterval = 0
  loginreenable = 0
  logindelay = 0
  pwdprompt = "Password: "
  usernameecho = false
```

Validating the user environment

- **`pwdck`** verifies the validity of local authentication information:
  - **`pwdck {-n|-p|-t|-y} {ALL | username}`**
  - Verifies that **/etc/passwd** and **/etc/security/passwd** are consistent with each other and with **/etc/security/login.cfg** and **/etc/security/user**
- **`usrck`** verifies the validity of a user definition:
  - **`usrck {-l|-b|-n|-p|-t|-y} {ALL | username}`**
  - Checks each user name in **/etc/passwd**, **/etc/security/user**, **/etc/security/limits** and **/etc/security/passwd**
  - Checks are made to ensure that each has an entry in **/etc/group** and **/etc/security/group**
- **`grpck`** verifies the validity of a group:
  - **`grpck {-n|-p|-t|-y} {ALL | groupname }`**
  - Verifies that the files **/etc/passwd**, **/etc/security/user**, **/etc/group** and **/etc/security/group** are consistent

Documenting security policy and setup

- Identify the different types of users and what data they will need to access
- Organize groups around the type of work that is to be done
- Organize ownership of data to fit with the group structure
- Set SVTX on shared directories
- Remember that UNIX/AIX has no concept of application ownership



Checkpoint (1 of 2)

- What are the benefits of using the **`su`** command to switch user to **root** over logging in as **root**?

_____

_____

5. Why is a umask of 027 recommended?

_____

_____

- As a member of the **security** group, which password command would you use?

_____

- Which password change command does SMIT use?

_____

13. True or False? When you delete a user from the system, all the user's files and directories are also deleted.

Checkpoint solutions (1 of 2)

- What are the benefits of using the **su** command to switch user to **root** over logging in as **root**?

  A log (which can be monitored) of all users executing the **su** command is kept in the **sulog**.

- Why is a **umask** of 027 recommended?

  This value removes all permission bits for the "others" category, which enhances security.

- As a member of the **security** group, which password command would you use?
  pwdadm (This command does not prompt for the **root** password or the old password of the user whose password is being changed.)

- Which password change command does SMIT use?
  passwd

- True or False? When you delete a user from the system, all the user's files and directories are also deleted.

Checkpoint (2 of 2)

1. If an ordinary user forgets their password, can the system administrator find out by querying the system as to what the user's password was set to? _____ Why? _____
   _____
2. Password restrictions are set in which of the following files?
   - **/etc/passwd**
   - **/etc/security/passwd**
   - **/etc/security/restrictions**
   - **/etc/security/user**
3. Which of the following statements are true?
   - A user can only belong to one group
   - A member of the **security** group can administer user accounts
   - An admin user is a user whose account cannot be administered by any member of the **security** group (except **root**)
   - The **`chmod g+s`** command sets the SUID permission of a file
   - The **root** user, commonly known as the superuser has UID=0 and GID=0

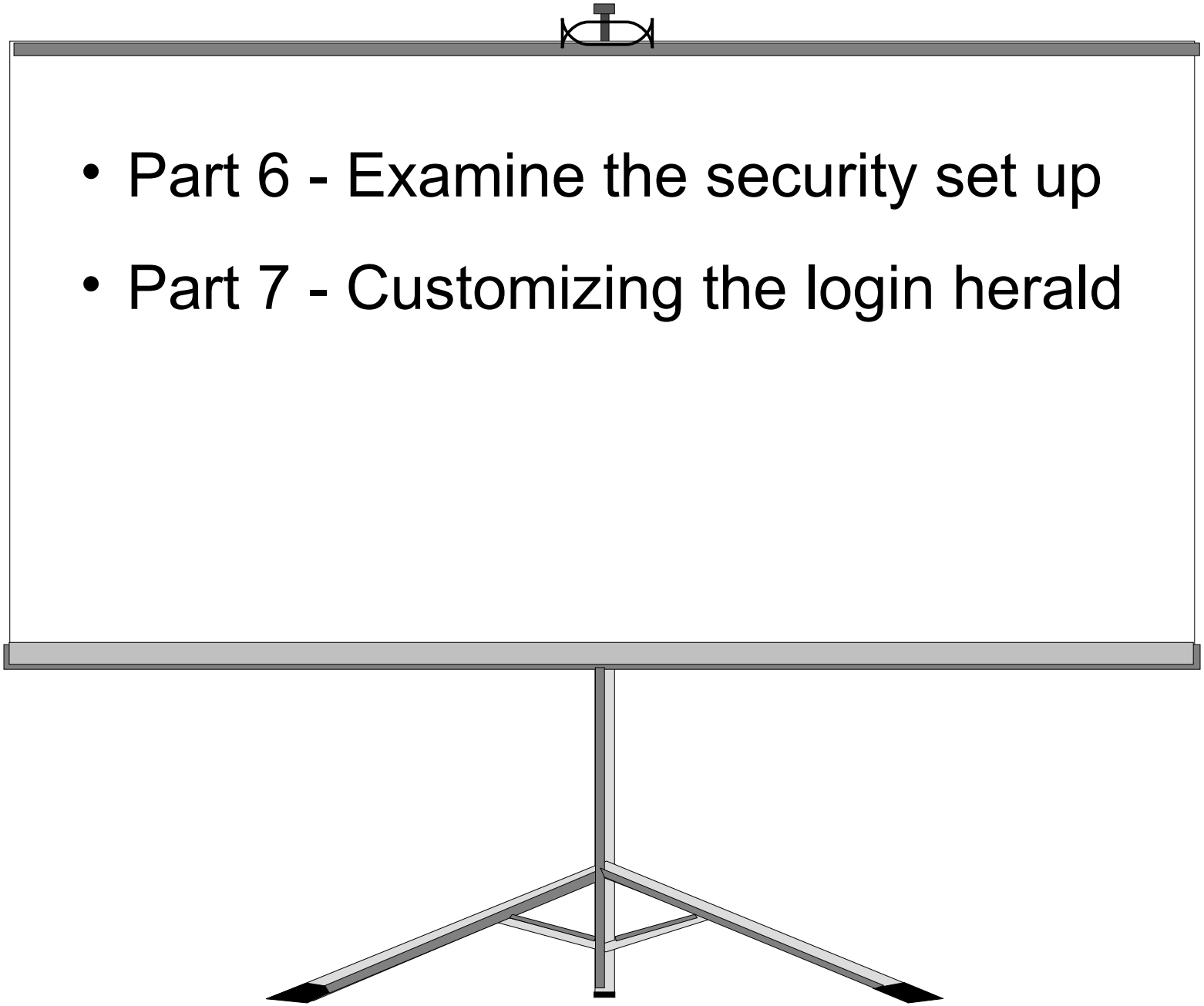
Checkpoint solutions (2 of 2)

- If an ordinary user forgets their password, can the system administrator find out by querying the system as to what the user's password was set to? No, because the passwords are held in encrypted format, so even the system administrator cannot tell what the password was set to.

2. Password restrictions are set in which of the following files?
   - **/etc/passwd**
   - **/etc/security/passwd**
   - **/etc/security/restrictions**
   - **/etc/security/user**

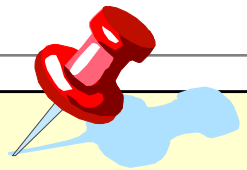
3. Which of the following statements are true?
   - A user can only belong to one group
   - A member of the **security** group can administer user accounts
   - An admin user is a user whose account cannot be administered by any member of the **security** group (except **root**)
   - The `chmod g+s` command sets the SUID permission of a file
   - The **root** user, commonly known as the superuser has UID=0 and GID=0

Exercise 16: User administration (parts 6-7)



- Part 6 - Examine the security set up
- Part 7 - Customizing the login herald

Unit summary



- User and groups can be added and deleted from the system by using SMIT or by using high level commands.
- Passwords must be set for all users using either **`pwdadm`** or **`passwd`**.
- Administrative users and groups can only be administered by root.
- Every user must be in at least one group.
- Certain groups give users additional privileges.
- Security files are located in ASCII text files in the **/etc** and **/etc/security** directories.