

LUENTO 21

Security

Ch 16 [Stal 05]

1

Perinteinen, symmetrinen salaus

Sama avain molemmilla!

Secret key shared by sender and recipient

Secret key shared by sender and recipient

(Fig 16.14 [Stal 05])

2

Julkisen avaimen salakirjoitusmenetelmä

Salaus

(a) Encryption (Fig 16.15 (a) [Stal 05])

3

Tunnistus julkisen avaimen menetelmällä

(b) Authentication (Fig 16.15 (b) [Stal 05])

4

Figure 16.1 Scope of System Security [MAEK87]

Requirements

- confidentiality, secrecy
 - Information accessible for reading only by authorized parties
- integrity
 - Information (assets) can be modified only by authorized parties
- availability
- authenticity
 - System must be able to verify the identity of the user

6

Threats

DoS – denial of service
(Fig 16.2 [Stal 05])

(b) Interruption

(c) Interception

(d) Modification

(e) Fabrication

See also Tbl 16.1 [Stal 05]

7

Passive attacks (sniffing, listening)

Fig 16.3 [Stal 05]

- Breaks confidentiality, but not integrity
- release of contents, traffic analysis

(a) Release of message contents

8

Active attacks

- Breaks integrity
- masquerade
- Replay
- modification of msg contents
- DoS = denial of service
 - overload, sabotage

(b) Modification of message contents

Fig 16.4 (a) [Stal 05]

Fig 16.4 (b) [Stal 05]

Fig 16.4 (c) [Stal 05]

Fig 16.4 (d) [Stal 05]

9

Malicious software - classification

Malicious software

- Needs host program
 - Trapdoors (takaovi)
 - Logic bombs (aikapommi)
 - Trojan horse
- Independent
 - Virus
 - Worm
 - Zombie (valloittaja)

Replicate (Virus, Worm, Zombie)

(Fig 16.8 [Stal 05])

10

```

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)
    
```

Fig. 9-10. (a) Normal code. (b) Code with a trap door inserted. [Tane 01]

(a) (b)

Fig. 9-9. (a) Correct login screen. (b) Phony login screen.

11

Buffer overflow [Tane 01]

Virtual address space

0xFFFF... Stack

Stack pointer

Main's local variables

Return addr

A's local variables

Buffer B

Program

(a) (b) (c)

Fig. 9-11. (a) Situation when the main program is running. (b) After the procedure A has been called. (c) Buffer overflow shown in gray.

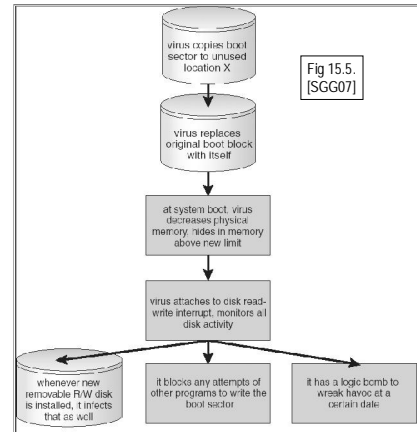
12

Viruses

- Virus dropper inserts virus onto the system
- Many categories of viruses,
 - Parasitic (in a File)
 - Memory resident
 - Boot
 - Macro
 - Polymorphic
 - Stealth

13

A Boot-sector Computer Virus



14

```

(a) MOV A,R1    MOV A,R1    MOV A,R1    MOV A,R1    MOV A,R1
    ADD B,R1    NOP          ADD #0,R1   OR R1,R1    TST R1
    ADD C,R1    ADD B,R1    ADD B,R1    ADD B,R1    ADD C,R1
    SUB #4,R1   NOP          OR R1,R1    MOV R1,R5   MOV B,R5
    MOV R1,X    ADD C,R1    ADD C,R1    ADD C,R1    ADD B,R1
                                     SHL #0,R1    SHL R1,0    CMP R2,R5
    SUB #4,R1   ADD C,R1    SUB #4,R1   SUB #4,R1   SUB #4,R1
    NOP          JMP +1     SUB #4,R1   SUB #4,R1   JMP +1
    MOV R1,X    MOV R1,X    MOV R1,X    MOV R1,X    MOV R1,X
    MOV R1,X    MOV R1,X    MOV R5,Y    MOV R5,Y    MOV R5,Y
    
```

Fig. 9-17. Examples of a polymorphic virus. [Tane 01]

15

Automatic virus detection/ analysis: Digital Immune System

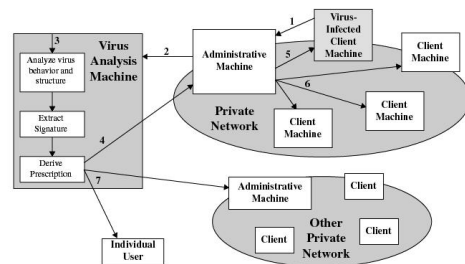


Figure 16.9 Digital Immune System

16

Intruders

- Growing problem
 - Using unauthorise account **masquerader**
 - Misusing own account **misfeasor**
 - Hiding the usage **clandestine user**
 - get root access, hide tracks
- How
 - Guess / try / crack passwds
 - Trojan horse
 - Sniffing
 - Clear text username /passwd in a message
 - Phishing
 - Man in the browser
 - Malprogram that activates only when browser access bank pages, then collects data and send it somewhere

17

Access rights

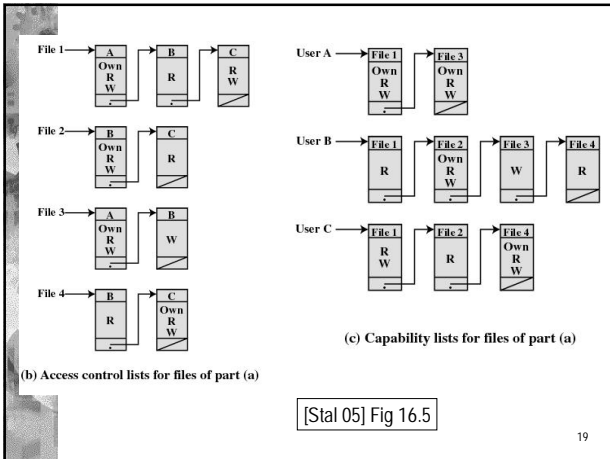
- Identified user (rows in access matrix)
- Domains: groups of resources (and users) that have certain access restrictions (rows and columns)
- Resources, files: (columns in the access matrix)
- Access rights checked by OS (at every access, or at open)

| | File 1 | File 2 | File 3 | File 4 | Account 1 | Account 2 |
|--------|---------------|---------------|---------------|---------------|-------------------|-------------------|
| User A | Own R W | | Own R W | | Inquiry Credit | |
| User B | R | Own R W | W | R | Inquiry Debit | Inquiry Credit |
| User C | R W | R | | Own R W | | Inquiry Debit |

(a) Access matrix

[Stal 05]

18



Access control policies

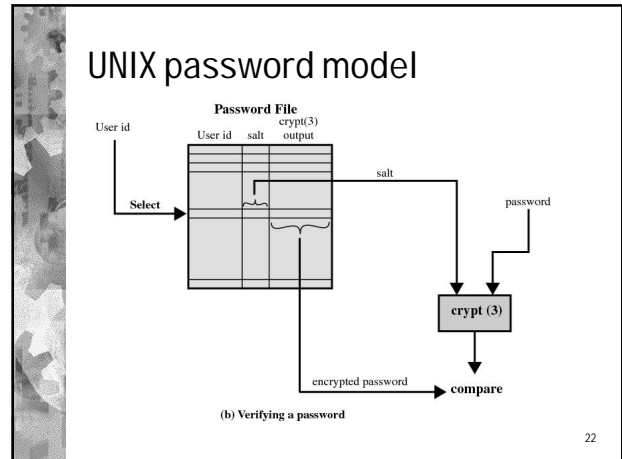
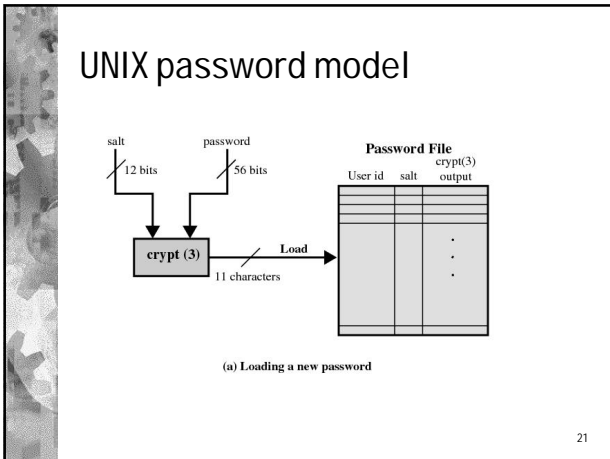
- DAC – discretionary access control
 - Owner decides
 - User can dynamically change access rights
 - User in with normal user data
- MAC – mandatory access control
 - Central policy defines the rights
 - Users cannot change access rights
 - Used with classified information

Choose, Option, Select

Remove read?

Obligatory

20



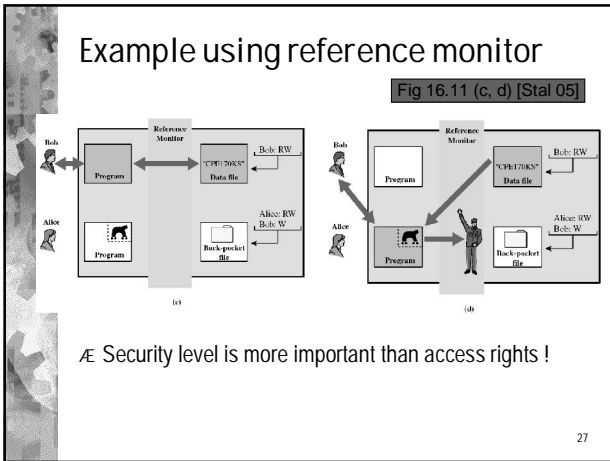
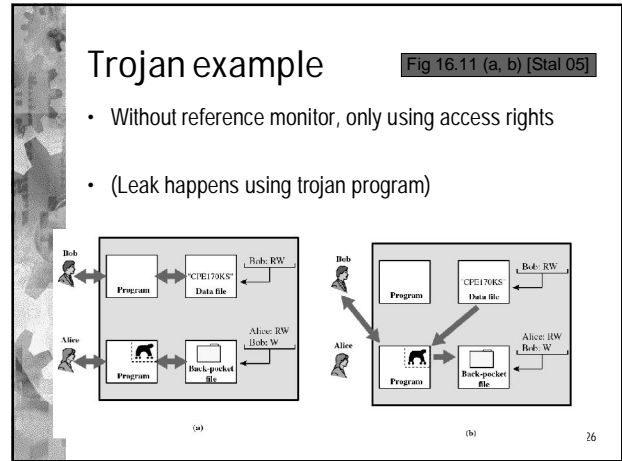
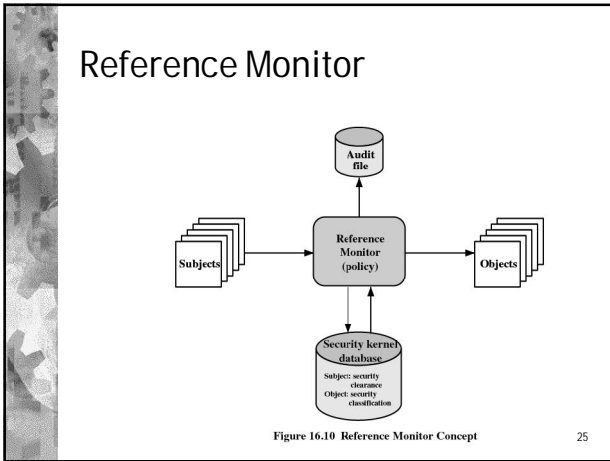
Luotettu järjestelmä (trusted system)

23

Multilevel Security

- Information classification
 - unclassified, confidential, secret, top secret
- Users have trust level
- No-read-up eli simple security property
 - Everyone can see (read) only own level or below classified data
- No-write-down eli *-property eli star-property
 - Everyone can produce (write) data only to own and upper levels
 - No leaks down (unless explicit permission)
- Needs Mandatory Access Control
- Example: reference monitor
 - Mediation, isolation, verifiability

24




UNIX tietoturva


Ch 10.7 [Tane 01]

UNIX access rights

- File attributes (i-node)
 - uid, gid
 - Access rights for u,g,o (modes rwx)
- User (process) uid and gid from
 - `/etc/passwd` uid and primary effective gid
 - `/etc/group` other gids
- uid and gid passed to child processes and new files
 - Files: Umask in process control block



UNIX access rights



- File attributes (i-node)
 - uid, gid
 - Access rights for u,g,o (modes rwx)
- User (process) uid and gid from
 - `/etc/passwd` uid and primary effective gid
 - `/etc/group` other gids
- uid and gid passed to child processes and files
 - Files: Umask in process control block
- Temporary change of rights, example:
 - only root has w-right to `/etc/passwd` file
 - `passwd` program file has set SETUID bit
 - effective userid is the owner of the file `passwd`
 - User process gets root access rights for the execution duration of `passwd`-program (because root is the owner), thus, it can change its own password
 - Similar SETGID bit for groups
 - effective groupid

normal
- rw- --- --- 1 root

- srw- s--- t--- 1 root

advanced special permissions

"sticky bit" keep file on swap device

Windows 2000 Tietoturva



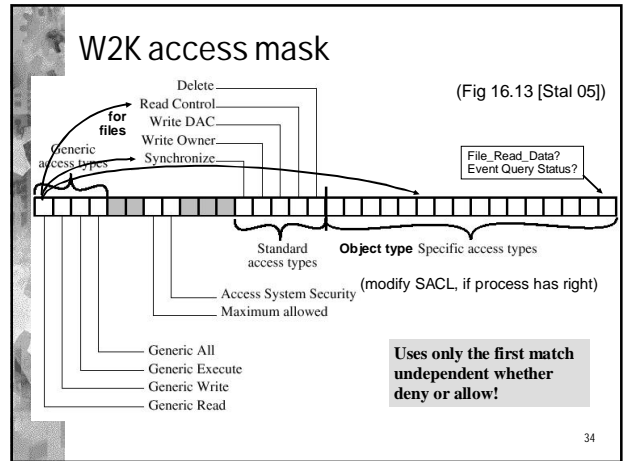
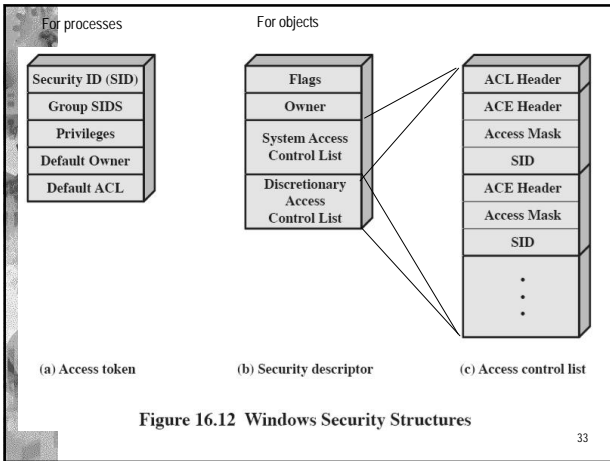
31

Computer Security Classifications

<http://www.dynamoo.com/orange/summary.htm>

- U.S. Department of Defense outlines four divisions of computer security: A, B, C, and D.
 - D – Minimal security.
 - C – Provides discretionary protection through auditing. Divided into C1 and C2. C1 identifies cooperating users with the same level of protection. C2 allows user-level access control.
 - B – All the properties of C, however each object may have unique sensitivity labels. Divided into B1, B2, and B3.
 - A – Uses formal design and verification techniques to ensure security.
- Windows on level C2: personal login, some restricted access to files

32



W2K DACL Example (NTFS)

Permissions for Administrators:

| Permission | Allow | Deny |
|---------------------|-------------------------------------|--------------------------|
| Full Control | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| Modify | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| Read & Execute | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| Read | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| Write | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| Special Permissions | <input checked="" type="checkbox"/> | <input type="checkbox"/> |

Advanced Security Settings for demo.txt:

| Type | Name | Permission | Inherited From |
|-------|---|----------------|----------------|
| Allow | Administrators (BUILTIN\Administrators) | Full Control | ACE Inherited |
| Allow | Administrators (NTLVL\Administrators) | Full Control | C:\ |
| Allow | Trusted Users (NTLVL\TrustedUsers) | Full Control | C:\ |
| Allow | SYSTEM | Full Control | C:\ |
| Allow | Users (NTLVL\Users) | Read & Execute | C:\ |

35