

# IN1805 I – Operating System Concepten

## Hoofdstuk 14: Protection

# Protectie en Security

- *Protectie*: mechanismen ten behoeve van het beschermen van resources tegen niet-geauthoriseerde toegang (intern)
- *Security*: houdt ook rekening met de externe omgeving.
  - b.v. computer achter slot en grendel

# Doel van protectie

- Een *access-policy* beschrijft voor ieder object welke toegang door wie is toegestaan (*geauthoriseerd*)
- *Protectie mechanismen* zorgen ervoor dat de toegang overeenkomstig de *access-policy* is
- *protectie* omvat:
  - middelen om de *policy* te specificeren
  - mechanismen om de *policy* af te dwingen
- *note*:
  - *policy* zegt **WAT**
  - *mechanisme* zegt **HOE**

# protectie domeinen (1)

- Systeem te beschouwen als een collectie van processen en objecten (hardware en software)
- object :
  - heeft unieke naam (id)
  - toegang via bepaalde operaties
- eis: toegangsmogelijkheden van een proces mogen niet groter zijn dan hetgeen geauthoriseerd is.
- wens: toegangsmogelijkheden van een proces op ieder moment te beperken tot wat op dat moment nodig is.  
( *least privilege principle, a.k.a. need-to-know principle* )

## protectiedomeinen (2)

- Een proces bevindt zich steeds in één *protectiedomein*.
- Een protectiedomein is een verzameling van *access rechten*.
- Een access recht : <objectnaam, rechtenverzameling >
- protectiedomeinen niet noodzakelijk disjunct
- relatie tussen proces en domeinen kan zijn:
  - statisch,  
voldoet meestal niet aan need-to-know principe
  - dynamisch:
    - proces kan van domain switchen
    - aanpassen van het domein mogelijk

## Protectiedomeinen (3)

Protectiedomeinen op verschillende niveaus mogelijk

- gebruiker:  
domein switching bij login
- proces  
domein switch door zenden van een bericht aan een ander proces
- procedure  
domein switch bij aanroep van procedure

# Protectiedomeinen (4)

## voorbeelden

- dual mode systeem
  - user mode: alleen nonprivileged instructies uit te voeren
  - monitor mode: priv. en nonpriv. instructies uit te voeren
- UNIX
  - domein hoort bij user
  - domeinswitching door tijdelijk van userid te veranderen, d.m.v. *SETUID-bit*. **Gevaarlijk !**
- alternatieve oplossingen (andere systemen)
  - privileged programma's in aparte (beschermd) directory, of
  - geen SETUID-bit, maar speciale verzoeken laten uitvoeren door daemon.
- algemeen: schrijf privileged programma's heel voorzichtig!

# Access Matrix Model

- Matrix  $A$  beschrijft toestand van het protectiesysteem
- kolom per object ( $O$ )
- rij per domein ( $D$ )
  - evt per subject (user of proces)
- $A[i,j]$  definieert rechtenverzameling van  $D_i$  t.o.v.  $O_j$
- policy bepaalt welke waarden  $A[i,j]$  mag hebben
- mechanisme moet ervoor zorgen dat toegang overeenkomstig de rechten is.

## Access Matrix model (2)

- een domein is zelf ook een object
  - het **switch** recht maakt domein switching mogelijk
- Wijzigingen in de access matrix mogelijk door:
  - **owner** : geeft zeggenschap over een kolom (d.w.z. een object)
  - **control**: geeft zeggenschap over een rij (d.w.z. een domein)
  - **copy**: geeft mogelijkheid een entry te kopiëren (alleen in een kolom), verschillende vormen:
    - *copy* : recht wordt gedupliceerd
    - *transfer* : recht verdwijnt uit de oorspronkelijke entry
    - *gelimiteerde copy* : copy mag niet opnieuw gecopieerd worden.

N.B. matrix biedt geen middel om de propagatie van informatie tegen te gaan.

# Access Matrix Model implementatie (1)

- Eis: als operatie  $M$  op  $O_j$  in  $D_i$  wordt uitgevoerd, verifieer of  $M$  voorkomt in rechtenverzameling  $A[i,j]$

Implementatie mogelijkheden :

- Eén grote matrix
  - te groot voor main memory
  - voor het grootste deel leeg
  - maakt geen gebruik van groepering

# Access Matrix Model implementatie (2)

## Implementatiemogelijkheden(vervolg)

- access list : per object een lijst van doubletten (te bewaren bij het object):
  - <domein, rechtenverzameling >
  - komt overeen met de niet-lege elementen van een kolom
  - eenvoudig uit te breiden met default rechten. (lijst alleen te doorzoeken wanneer een niet-default operatie werd gevraagd)

# Access Matrix Model implementatie (3)

## Implementatiemogelijkheden (vervolg)

- capability list : per domein een lijst van capabilities.  
capability = objectid + rechtenverzameling
  - uitvoeren van operatie M op object  $O_j$  toegestaan op 'vertoon' van juiste capability.
  - capability niet rechtstreeks toegankelijk voor de gebruiker, bescherming door:
    - capabilities in beschermd deel van het geheugen
    - one-way functions
    - speciale architectuur: *tagged objects*

# Access Matrix Model implementatie (3A)

- one-way function  $f$ 
  - $y=f(x)$  is makkelijk uit te rekenen, maar bij gegeven  $y$  is het niet doenlijk een  $x$  te vinden zodat  $y=f(x)$
- te gebruiken om capabilities te beschermen,
  - Bepaal bij creatie van een object  $O_i$  een random getal  $N_i$
  - Bewaar  $N_i$  bij  $O_i$
  - Bepaal rechten  $R_i$ , Bereken  $Y_i = f(N_i \text{ XOR } R_i)$
  - Capability  $C = O_i, R_i, Y_i$
  - Acties van access control mechanisme bij access poging dmv  $C$ :
    - Zoek  $N_i$ , neem  $R_i$  uit  $C$ , bereken  $Y_i' = f(N_i \text{ XOR } R_i)$
    - verifieer dat  $Y_i' = Y_i$
    - Zo ja, access OK, zo nee access niet toegestaan

# Access Matrix Model implementatie (4)

## Implementatiemogelijkheden (vervolg)

- lock en key mechanisme
- compromis tussen access list en capability list
- bij een resource hoort een lock list (L)
- bij een domein hoort een key list (K)
- $K_i$  geeft toegang tot een resource via  $L_i$
- meerdere resources kunnen hetzelfde lock gebruiken
- meerdere processen kunnen dezelfde keys hebben
- toegangsrechten in te trekken door lock te veranderen
- gebruikers hebben geen rechtstreekse toegang tot hun keys (vergelijkbaar met capabilities)

# Access Matrix Model

## implementatie (5) : overzicht

- access list
  - nauw gekoppeld aan gebruikers.
  - rechten makkelijk in te trekken
  - overzicht van mogelijkheden van een proces moeilijk te krijgen
- capability list
  - capabilities minder sterk gekoppeld aan gebruikers
  - rechten moeilijk in te trekken
  - eenvoudig overzicht per proces te krijgen
- lock-key mechanisme
  - minder sterk gekoppeld aan gebruikers
  - rechten eenvoudig in te trekken
  - eenvoudig een overzicht van de keys per proces te krijgen

# IN1805 I – Operating System Concepten

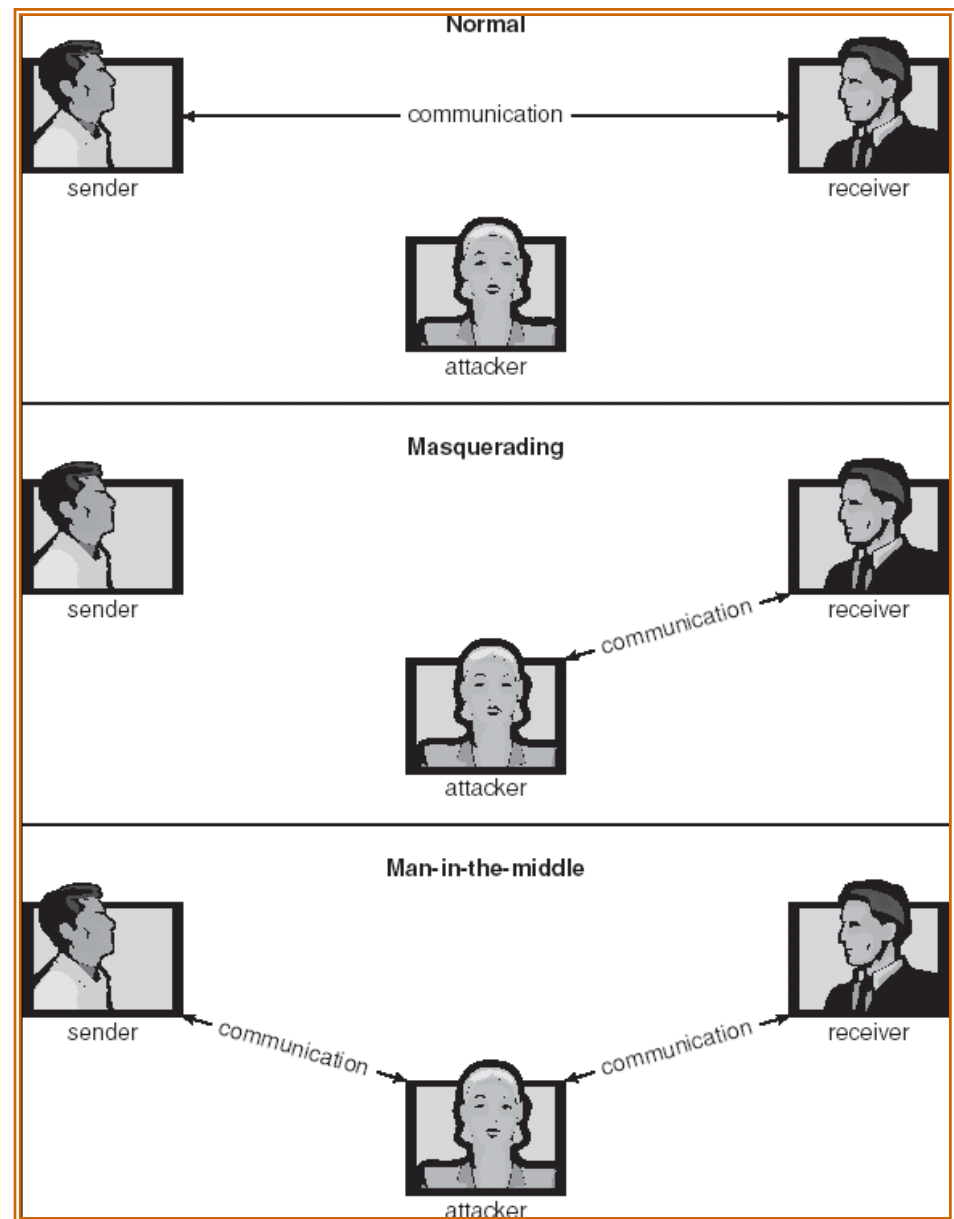
## Hoofdstuk 15: Security

# Veiligheid

- Een systeem is veilig (secure) als de resources worden gebruikt in overeenstemming met de policy ('de regels')
- Inbreuken op de veiligheid kunnen zijn:
  - schending van **Confidentiality**: ongeoorloofd lezen van informatie
  - schending van **Integrity**: ongeoorloofd wijzigen van informatie
  - schending van **Availability**: ongeoorloofd vernietigen of onbereikbaar maken van gegevens
- Inbreuk kan met of zonder opzet worden veroorzaakt

# Generieke aanvallen

- Alice, Bob, and Eve



# Veiligheidsmaatregelen

Veiligheidsmaatregelen op verschillende niveaus:

- fysieke maatregelen.
  - b.v. computerruimte op slot
- procedureel (menselijk)
  - b.v. duidelijke regels m.b.t. bevoegdheden
- systeemtechnisch
  - hard- en softwaremechanismen

*Maatregelen moeten in evenwicht zijn*

# Authenticatie (1)

- Authenticatie mechanismen moeten de identiteit van een gebruiker of proces met zekerheid kunnen vaststellen.
- Authenticatie meestal op grond van :
  - iets wat men **heeft** (b.v een bankpasje), en
  - iets wat men **weet** (b.v. een pincode)
  - soms: wordt gebruik gemaakt van een 'persoonlijk' attribuut, b.v. vingerafdruk, stem, iris

## Authenticatie (2)

- In OS vaak authenticatie d.m.v. password  
dit is zwak, vanwege:
  - te raden
  - brute force methoden
  - afluisteren
- beter: gebruik maken van challenge/response methode
- eenmalige passwords
- passwords op te slaan in het systeem d.m.v. one-way function

# Bedreigingen (1)

- Trojaanse paarden
  - een programma dat behalve zijn echte functie ook geheime onderdelen bevat die misbruik maken van het domein waarin het draait. b.v. programma geschreven door A, misbruikt het domein van B
- Trapdoor
  - Een bijzonder Trojaans paard, wordt geactiveerd door een bepaalde wijze van aanroepen
- Worm
  - Een programma dat zichzelf vele malen kan kopiëren, en daardoor de normale werking van het systeem onmogelijk kan maken. (v.b. de Internet worm (1988))

## Bedreigingen (2)

- Virus :
  - Een fragment code in een programma, dat wanneer het wordt uitgevoerd, zichzelf naar een ander programma copieert.
  - zal vaak lange tijd niet hinderlijk zijn
  - slaat op een gegeven moment toe,
    - b.v. door wissen van de hard-disk, en
    - een boodschap op het scherm
  - bescherming:
    - alleen vertrouwde floppies, bulletin boards, ftp-sites gebruiken,
    - virusscanners
    - geen verdachte bijlagen openen onder windows

## Bedreigingen (3)

- Buffer overflow :
  - zorg voor te veel input (input string, commandline argument) zodat het return adres op de stack overschreven wordt
  - zorg dat de procedure retourneert naar eigen code (onderdeel vd input)
  - start een nieuw programma met de rechten vd huidige user (server)
  - have fun!

# Management technieken t.b.v. veiligheid

- threat monitoring:
  - verdachte patronen signaleren  
b.v. te vaak een fout password intoetsen
- audit log :
  - registreert alle accessen. Na inbreuk te zien wat er gebeurd is
- watch dog:
  - periodiek, maar **niet op vaste tijden** systeemcomponenten inspecteren.  
b.v. veranderingen in systeemprogramma's

# Veiligheids-klassificatie (1)

DoD (Department of Defense) : **TSEC** (Trusted Computer System Evaluation Criteria) = *Orange Book*

onderscheidt 4 divisies A t/m D (A hoogst, D laagst), sommige verder verdeeld in klassen.

D: geen bescherming (b.v. MS-DOS)

C: *discretionary* protection (d.w.z. bescherming op basis van behoefte)

C1: gebruiker kan zijn informatie beschermen tegen anderen. (voorbeeld: veel UNIX versies)

C2: C1 + rechten te specificeren per individu. Gebruikers moeten zich identificeren bij login

+ gebruikers individueel aanspreekbaar (accountable) op hun gedrag.

## Veiligheids-klassificatie (2)

B: *mandatory* protectie (bescherming verplicht door systeem)

B1: C2 + *sensitivity label* per object

(b.v. confidential, secret, topsecret)

clearance level per individu

B2: B1 + sensitivity label voor fysieke objecten

+ auditing van *covert channels*

B3: B2 + monitoren van verdachte events

A: A1: als B3, maar ontwikkeld met formele ontwerp en specificatie methoden.