Giornata di studi sul Trusted Computing

Da Multics a TCPA

breve storia di insicurezza quotidiana

Andrea Pasquinucci
www.ucci.it

All'inizio furono i militari....

(A.D. 1960...)

MULTICS

- MIT, AT&T, IBM, GE 1965 -> 1969 (poi ...)
- *Support at least 1000 users*
- *Reliable*
- *Sufficient control of access to allow selective sharing of information (=> Bell-LaPadula)*
- Approccio Top-Down

Da Multics a UNIX

- Funzionava con 1 utente, quasi con 2, non con 3 !!!
- UNIX (Thompson & Ritchie 1969)
  - Approccio bottom-up
  - Modulare
  - Multi-utente, Multi-tasking, Multi-...
  - eccetera...

UNIX: Modello della Sicurezza

- Super-User *root* ("Deus *in* machina")
- Utenti a cui è vietato fare alcune cose

- Modello **DAC** (*Discretionary Access Control*)

UNIX: Modello della Sicurezza

- Kernel-space: parte più interna OS che gestisce HW, periferiche ecc. (ring 0)
- User-space: programmi utente (ring 1 o 3)

- Kernel-space separato da user-space via HW

E' necessaria una protezione HW

UNIX: Modello della Sicurezza

- SCOPO: **Integrità della esecuzione del codice** (accesso alle periferiche, HW, multi-... ecc.)

- ASSUME: *utenti e programmi NON sono maligni*

Da UNIX a DOS

- UNIX troppo complicato per piattaforme a basso costo (richiede competenza, Sys-manager, ed HW ai tempi costoso)
- DOS (1981)
  - 1 solo utente
  - Sicurezza garantita principalmente dall'assenza di *bugs* nel SW :--((

Approccio Teorico

- Militare: garantire la sicurezza assoluta (?)
- Bell La Padula (1975)
  - Write Up / Read Down => Confidenzialità
- Biba (1977)
  - Write Down / Read Up => Integrità
- Modelli più complessi: sicurezza NON dimostrabile !

TRUST

- Livello di **Fiducia**: Quanto mi fido che un sistema svolga un compito come deve
- Orange Book (TCSEC 1985): fiducia bassa, media, alta ... ?
- ITSEC (1991) => CTCPEC (1993) => Common Criteria v2 (1998)
  - *Target of Evaluation* (TOE)
  - *Protection Profile* (PP) :--((

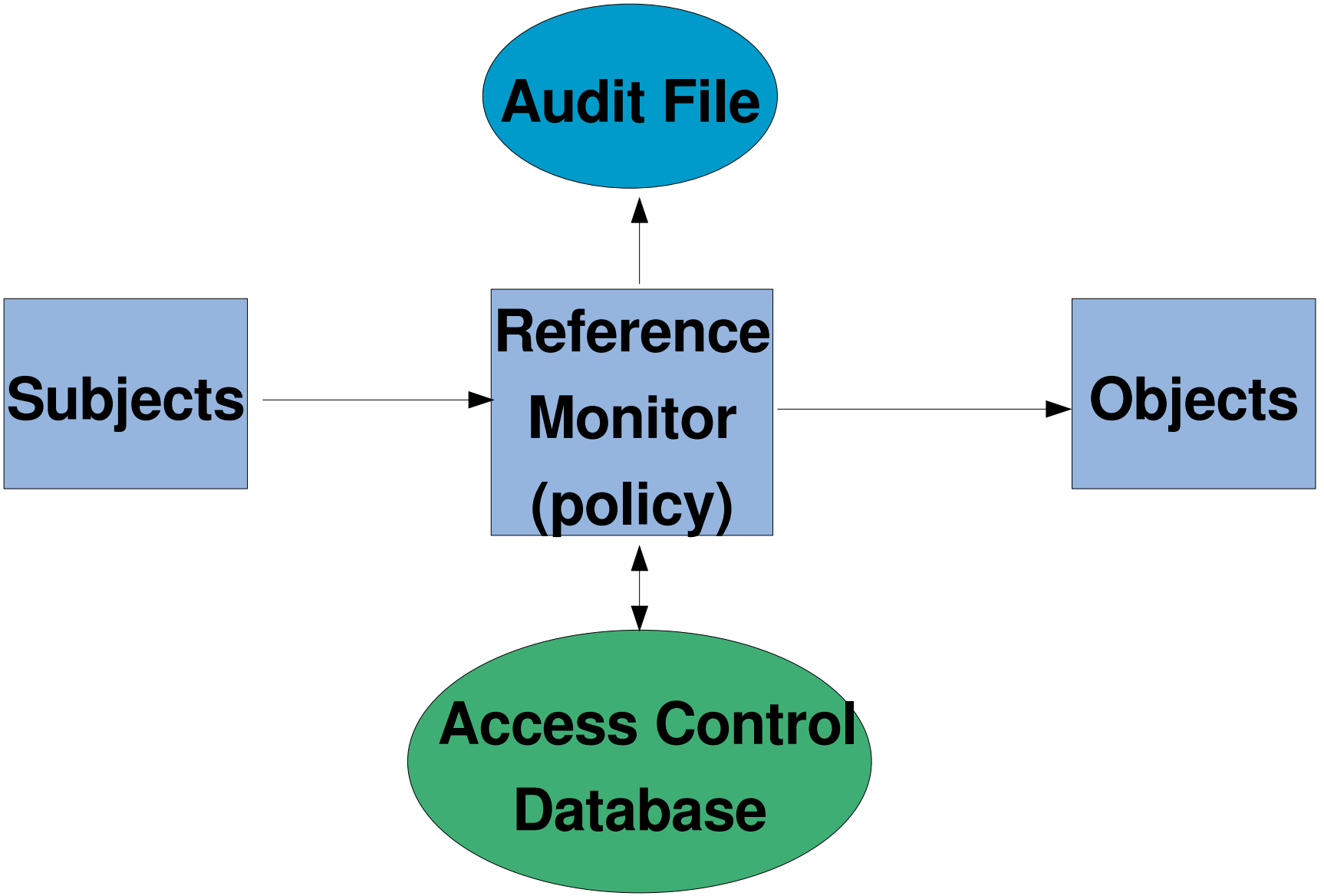
CHE SI FA ?

Sin dagli anni 1980 si capì che esiste UNA SOLA SOLUZIONE PRATICA:

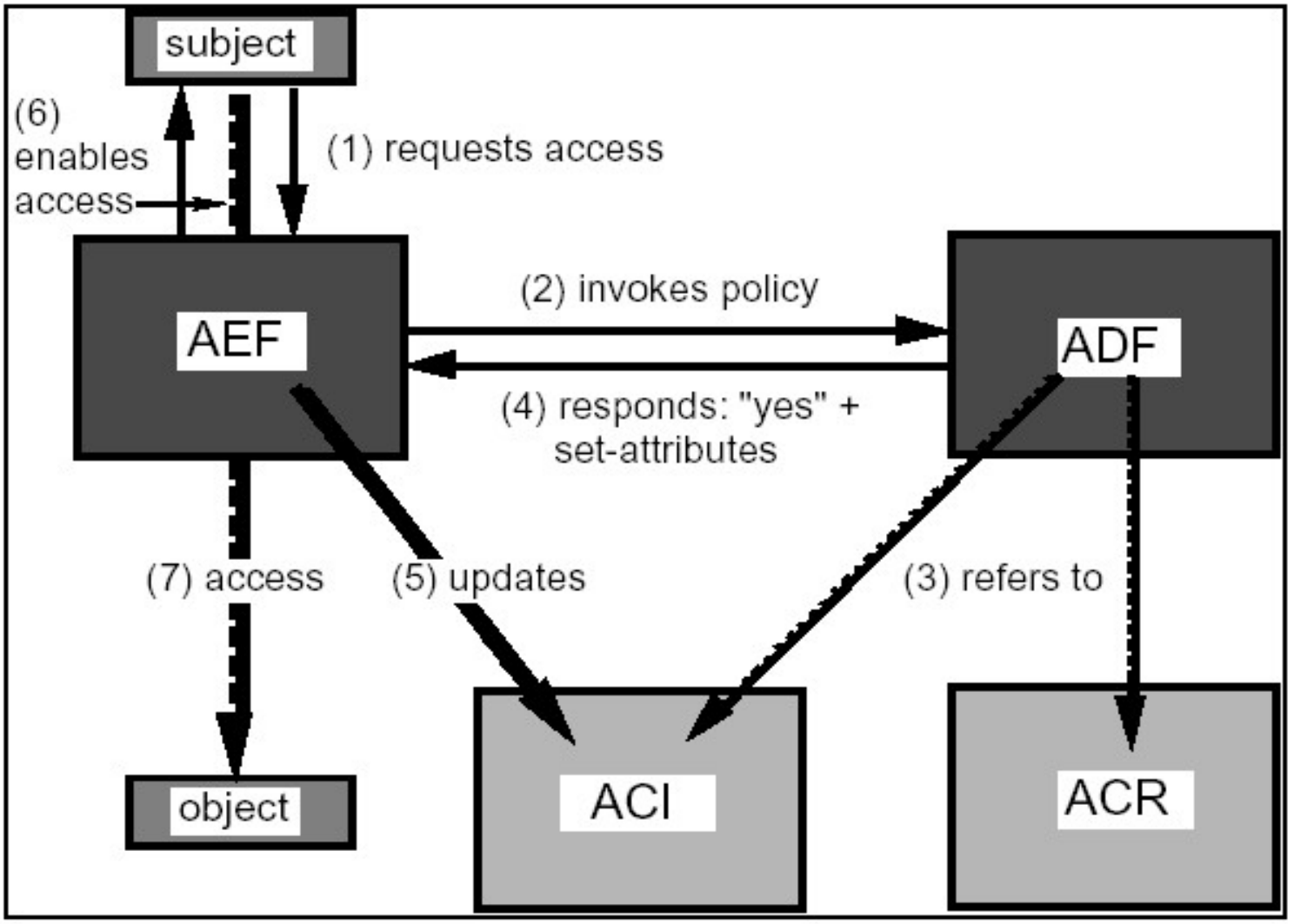
- **Security Kernel** in protected mode (kernel space)
- **Trusted Computing Base** / **Reference Monitor** (TC/RM)
  - Controlla ogni richiesta dall'user-space secondo regole fissate da un **Security Officer**

TC/RM

- E' necessario che il TC/RM sia **ISOLATO** in HW, non modificabile ecc.
- Il Security Officer decide le regole che implementano la *Politica di Sicurezza* e le carica nel *Access Control Database*
- **Mandatory Access Control** (MAC)


Audit File
Subjects
Reference Monitor (policy)
Objects
Access Control Database

RSBAC


subject
(6)
enables
access
(1) requests access
AEF
(2) invokes policy
ADF
(4) responds: "yes" +
set-attributes
(7) access
(5) updates
(3) refers to
object
ACI
ACR

Problemi più difficili

- Covert Channels
- Accesso a periferiche
- ...

E negli anni 1990 ...

- Ci si dimentica di tutto ciò perché il problema / interesse principale è sviluppare APPLICA-ZIONI più veloci, più potenti, più user-friendly ...
- Anche se questo può voler dire eseguire l'applicazione (codice utente) in kernel-space

  :--((
- (Sicurezza garantita dall'assenza di bugs! giusto?)

... e negli anni 2000

- Arrivano le applicazioni multimediali e con la spinta degli interessi economici in gioco,
- i problemi di sicurezza di massa (virus et al.),
- il valore economico delle transazioni via Web dall'home-banking al private-banking agli ordini just-in-time =>

si riscopre il TC/RM che con le tecnologie attuali è finalmente realizzabile ==> TCPA

Futuro ?

- Il futuro della sicurezza HW/OS è *oggi* il TC/RM (la cui incarnazione attuale è il TCPA)
- Dobbiamo 'solo' svilupparlo correttamente ed utilizzarlo per gli scopi opportuni

Copyright