confinamento e virtualizzazione

oltre i permessi dei file...

- nei sistemi operativi standard il supporto per il confinamento è abbastanza flessibile per quanto riguarda i files
- scarso per quanto riguarda molte altre operazioni
  - es. socket, visualizzazione processi
- sotto unix è difficile negare permessi a utenti specifici
  - non ci sono vere acl
- alle volte sono utili forme più stringenti (anche se meno standard) di confinamento

chroot jail

- chroot è un comando unix
- esegue un processo con una nuova radice
  - es. la radice / del processo eseguito coincide con /var/www del processo che esegue
  - non c'è modo di leggere files posti all'esterno della nuova radice
- nella nuova radice deve essere ricreata la struttura di un filesystem Unix anche minimale
  - comandi di base se necessari
  - librerie dinamiche
- chroot deve essere eseguito da root

chroot jail: limiti

- chroot restringe l'accesso solo al filesystem
- dall'interno si può ancora...
  - ispezionare i processi
  - usare la rete
    - magari per trasferire degli eseguibili all'interno del jail
  - lanciar processi (avendo gli eseguibili a disposizione)
    - i processi figli avranno la stessa root del padre
  - chiamare qualsiasi system call (avendo l'eseguibile giusto)

os-level virtualization

- il kernel puo' supportare il confinamento
  - confina tutte (o molte) risorse, non solo il filesystem
  - es. solo i processi del jail sono visibili, rete limitabile, potenzialmente anche l'uso della cpu è limitabile
- il jail esegue lo stesso kernel dell'host
  - efficiente
- implementazioni
  - Linux: vserver, virtuozzo
  - FreeBSD: jails
  - Solaris: container

full virtualization

- il software emula l'hw
  - il processore può non essere emulato (almeno per le istruzioni non privilegiate)
  - problemi di efficienza per l'I/O
  - due kernel: guest e host
  - i due kernel possono essere completamente diversi
- software
  - processore nativo: VMware, VirtualBox, VirtualPC (Microsoft), Xen (con hw assisted virt.), VirtualIron, KVM
  - processore emulato (inefficente): QEMU (GNU GPL)
- hw assisted virtualization
  - Intel Virtualization Technology (Intel VT)
  - AMD virtualization (AMD-V)

para-virtualization

- richiede il porting dei sistemi operativi guest
  - interfaccia mostrata al guest del tutto diverse da quelle hw
- Xen (senza hw assisted virt.)
  - hypervisor (caricato al boot al posto del sistema host)
  - molto efficiente
  - permette migrazione tra host fisici on-the-fly
  - open e supportato commercialmente
- User Mode Linux
  - solo linux in linux
  - molto leggero
  - supportato come architettura del kernel linux
  - usato per www.netkit.org

isolamento dei servizi e consolidamento

- servizi con politiche di gestione diverse dovrebbero essere isolati tra loro
- il miglior isolamento è dato da server distinti
  - proliferazione dei server (spesso sottoutilizzati)
  - alti costi → server di basso livello → bassa affidabilità
- consolidamento
  - poco hw di buona qualità e ridondato
  - uso di tecnologia di virtualizzazione
  - tanti server virtuali

SELinux

National Security Agency

- The National Security Agency/Central Security Service is America's cryptologic organization. It coordinates, directs, and performs highly specialized activities to protect U.S. government information systems.

SE Linux

- “prototipo” sviluppato da NSA per provare/mostrare l'efficacia del concetto di “mandatory access control”.
- parte del kernel 2.6
  - attivabile come opzione di compilazione SECURITY_SELINUX
- attivo di default su distribuzioni Fedora

SELinux e MAC

- discretionary access control (dac)
  - l'utente può passare i suoi diritti ad altri soggetti, anche indirettamente e/o involontariamente, es. trojan
- mandatory access control (mac)
  - “confinare i programmi utente e i server di sistema all'interno di un insieme di privilegi minimo indispensabile per svolgere il loro compito”
  - può non esistere un utente che può far tutto!
    - dipende dalla configurazione

MAC vs DAC in Linux

- la sicurezza di un sistema Linux classico (DAC) dipende da...
  - permessi assegnati ai file
    - l'accortezza degli utenti!
  - correttezza del kernel
  - correttezza di tutte le “applicazioni” privilegiate e loro configurazione
  - assenza di trojan horses
- un problema in ciascuna di queste aree può compromettere parte o l'intero sistema
- la sicurezza di un sistema SE Linux dipende **SOLO** dalla correttezza del kernel e dalla configurazione delle security policy

SE Linux ha bisogno di...

- utility proprie di SE Linux
  - configurazione delle policy, ecc
- alcuni pacchetti vanno sostituiti con delle versioni modificate per l'uso delle caratteristiche di SE Linux
  - es. login, su, ecc.
  - in debian pacchetto selinux-basics
- l'installazione normalmente parte da un sistema Linux convenzionale
- in fedora selinux e' default
    - la security policy è però molto lasca

compatibilità

- tutte applicazioni sono pienamente compatibili
  - SE Linux non cambia nulla di ciò che un processo vede
  - i mandatory access control vegnono eseguiti dopo i controlli standard di Linux se questi hanno avuto successo
- alcuni moduli kernel sono stati modificati per interagire con SE Linux

vulnerabilità

- SE linux non contiene (in linea di principio) meno vulnerabilità di linux!
- SE linux è solo molto più flessibile e accurato dal punto di vista delle policy
  - permette di confinare i processi all'interno di “domini” più stretti
  - una vulnerabilità in un processo è quindi molto meno pericolosa

concetti

- Linux
  - utenti e gruppi (UID, GID)
  - permessi sui file
  - root (UID=0, nessun controllo)
- SE Linux
  - identity (identifica una “persona fisica”)
  - domain (ciascun processo gira in un dominio che determina le oprazioni ammesse)
  - role (determina quali domini possono essere usati, RBAC)

concetti

- ciascuna identity è associata a uno o più ruoli
  - esempio: utente e amministratore
    - le policy dicono quali sono i ruoli di una identità
  - da una certa identity si può passare da un ruolo ad un altro inserendo la password che autentica per quell'indentity (serve a verificare chi e' davanti al terminale)
- dato il ruolo si sa quale sono i domini in cui si possono “muovere” i nostri processi
  - esempio un utente può normalmente lanciare un comando per cambiare la password
  - il processo che cambia la password non può fare altro che modificare il file delle password e non, ad esempio cancellare altri file

modalità

- enforcing mode
  - si effettuano tutti i controlli e si forzano le politiche
  - questa è la modalità in cui si ha mandatory access control
- permissive mode
  - si effettuano i controlli ma non si forzano le politiche, quando il reference monitor nega l'accesso vengono solo generati dei log
  - questa è la modalità in cui si possono effettuare attività amministrative
- switch tra le due modalità a runtime
  - SECURITY_SELINUX_DEVELOP=y
  - echo 1 >/selinux/enforce
  - echo 0 >/selinux/enforce # se permesso dalla policy

sicurezza al boot

- scelta se selinux è attivato con “selinux=0/1”
  - SECURITY_SELINUX_BOOTPARAM=y
  - SECURITY_SELINUX_BOOTPARAM_VALUE=0 o 1
- per sistemi veramente sicuri dovrebbe essere posto BOOTPARAM=n e altro
  - cioè non è possible disattivare selinux dai parametri di boot
  - ma anche: no single user mode, no init=/bin/bash
- in alternativa GRUB passwords
- BIOS
    - password del bios
    - disattivazione boot da dispositivi removibili
- sicurezza fisica
  - si può sempre rubare l’hard disk
  - lucchetto al case, lucchetto allo chassis e porta chiusa a chiave
  - o filesystem criptati

SELinux è un trusted system?

- SELinux non è un trusted system
  - ci vuole una certificazione
  - non basta una configurazione del kernel per ottenere una certificazione
- una specifica configurazione contenente SELinux può essere un trusted system
- un sistema certificato basato su SELinux
  - Red Hat Enterprise Linux (RHEL) Version 4 Update 1 AS/WS
  - Common Criteria EAL4+, gennaio 2006