# Securitatea sistemului

SO: Curs 13

# Cuprins

- Sisteme sigure
- Separarea privilegiilor
- Securitatea fișierelor
- Cel mai mic privilegiu
- Identitatea utilizatorilor

# Suport de curs

- Operating Systems Concepts
  - Capitolul 14 – Protection
  - Capitolul 15 – Security
    - Secțiunile 15.1, 15.2, 15.5
- Modern Operating Systems
  - Capitolul 9 – Security
    - Secțiunile 9.4, 9.6

# SISTEME SIGURE

# Sistem sigur

- Face ce este proiectat să facă
- Face doar ce este proiectat să facă
- Folosește corespunzător resursele date
- Asigură confidențialitate informațiilor
- Un sistem nesigur
  - Are comportament nedeterminist
  - Are comportament exploatabil (controlabil de atacator)
  - Folosește necorespunzător/abuziv resurele sistemului
  - Permite accesul la date confidențiale

# Bug și vulnerabilitate

- Bug: problemă de implementare/proiectare
  - Generează comportament nevalid, în general eroare
- Vulnerabilitate (bug de securitate)
  - Problema poate fi exploatată
  - Un atacator poate folosi/controla bug-ul/vulnerabilitatea
  - Un model de atack (attack vector) rezultă în atacator obținând acces sau controlând accesul la resurse

# Elemente de proiectare a unui sistem sigur

- Separarea privilegiilor
  - Procesele au anumite privilegii
  - O acțiune diferită -> un proces nou cu alte privilegii
  - Un proces compromis nu compromite altele
- Cel mai mic privilegiu
  - Nu lași unui proces mai multe privilegii (permisiuni) decât are nevoie
  - Dacă un proces este compromis pagubele sunt conținute
- Limitarea resurselor
  - Nu se oferă mai multe resurse decât este necesar

# Elemente de proiectare a unui sistem sigur (2)

- Identitatea utilizatorilor
  - Autentificare, credențiale robuste
- Monitorizare
  - Procese
  - Folosirea resurselor
  - Acționat în momentul în care sunt probleme de securitate
- Confidențialitate: criptare, integritate
- Asigurarea calității
  - Secure coding, secure programming, defensive programming, secure by design
  - Security auditing, fuzz testing
  - Software updates

# SEPARAREA PRIVILEGIILOR

# Separarea privilegiilor

- componente separate au roluri separate
- cuplat cu principiul celui mai mic privilegiu
- exemplu: Postfix; master (root) + smtpd, pickup, cleanup, qmgr (postfix)
- folosirea utilizatorului nobody
- privilegii administrative (complete): kernel-mode sau root
  - de folosit rar
  - de delegat roluri (sudo)

# Separația kernel mode – user mode

- Instrucțiunile privilegiate sunt executate în spațiul kernel
  - accesul la I/O
  - alocarea de resurse
  - handler-ele de întrerupere
  - gestiunea sistemului
- Suportul procesorului
  - niveluri de privilegiu (rings)
  - x86: nivelul 0 (kernel), nivelul 3 (user)

# Utilizatorul administativ (root)

- Are acces complet la sistem
- Are privilegiile nucleului în sistemul de operare
- Procesele administrative rulează cu permisiuni de root
- Accesul trebuie limitat
- Dispozitivele mobile în general nu oferă acces de tipul “root” în mod direct

# Utilizatori și permisiuni

- Roluri dedicate sunt oferite altor utilizatori
- Un utilizator are un subset de privilegii și acces limitat la resursele sistemului
- Accesul la sistemul de fișiere se realizează prin permisiuni la sistemul de fișiere
- Dacă un proces al unui utilizator este compromis pagubele sunt limitate la permisiunile acelui utilizator

# Stabilirea permisiunilor/privilegiilor

- DAC: Discretionary access control
  - Există noțiunea de owner
  - Owner-ul poate stabili permisiuni/privilegii
  - Permisiunile pe sistem de fișiere
- MAC: Mandatory access control
  - Kernel-ul/Sistemul/Administratorul decide întregul set de privilegii
  - Nu există owner
  - Modificările sunt efectuate doar de sistem/root
  - SELinux
  - Suport și pe sistemele mobile (Android, iOS)

# SECURITATEA FIȘIERELOR

# Permisiuni pe sistemul de fișiere

- o formă de separare de privilegii
- controlul accesului
  - anumiți utilizatori au anumite drepturi pe resursele din sistemul de fișiere
- matrice de control al accesului
- liste de acces

# Drepturi pe fișiere

- Asocierea drepturilor de acces pentru utilizatori la fișiere
- Citire, scriere, ștergere, execuție
- Creare fișier, listare, ștergere fișier, parcurgere

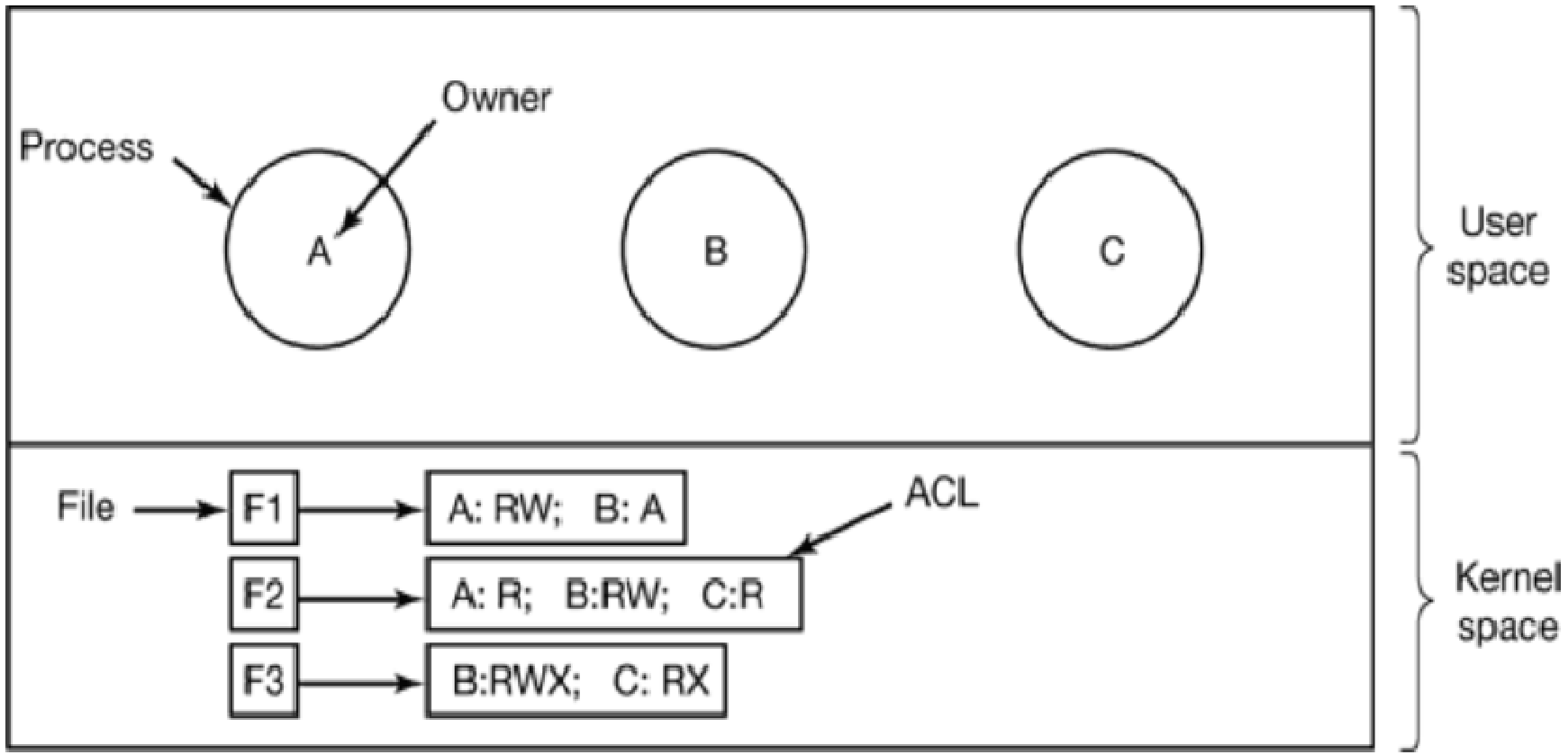
# Matrice de acces

| Domain | File1 | File2 | File3 | File4 | File5 | File6 | Printer1 | Plotter2 |
|---|---|---|---|---|---|---|---|---|
| 1 | Read | Read<br>Write | | | | | | |
| 2 | | | Read | Read<br>Write<br>Execute | Read<br>Write | | Write | |
| 3 | | | | | | Read<br>Write<br>Execute | Write | Write |

# Drepturi pe fișiere în Unix

- Matrice de acces
- Domeniile sunt
  - utilizator (user) - deținătorul fișierului
  - grup (group) – grupul deținător al fișierului
  - alții (others)
- Drepturi
  - read (r) – citire, listare
  - write (w) – scriere, creare fișier
  - execute (x) – execuție, parcurgere

# Liste de acces



# Liste de acces (2)

- POSIX ACL
  - implementate pe sisteme de fișiere Linux cu extended attributes
  - getfacl, setfacl
- Drepturi pe fișiere în Windows
  - ACL pe NTFS
  - read, write, list, read and execute, modify, full control

# Capabilități

- O cheie asociată unor acțiuni privilegiate sau unor drepturi de acces
- Un mod de delegare de permisiuni din partea utilizatorului root către un utilizator neprivilegiat
- Pot fi interschimbate între entități
  - nu este un lucru obișnuit în sistemele de operare actuale
- Capabilități POSIX (IEEE 1003.1e)
  - CAP_NET_BIND_SERVICE
  - CAP_SYS_CHROOT
  - CAP_NET_RAW
- man 7 capabilities
  - nu se poate accesa un director/fișier din afara ierarhiei impuse de noul director rădăcină

# Escaladarea privilegiilor

- permiterea obținerii de permisiuni suplimentare pentru acțiuni privilegiate
  - sudo
  - setuid
- de obicei sunt atacate conturile privilegiate
- sunt exploatate programele care rulează ca root - atenție specială executabilelor cu bitul setuid activat
  - bug în aplicații - obținerea unor drepturi necuvenite

# Bitul setuid

- set user-ID on execution bit
- Real user ID
- Effective user ID
- Bitul setuid (chmod 4777)
  - permite configurarea euid ca utilizatorul ce deține executabilul
- setuid
  - total privilege revocation (real user ID, effective user ID)
- seteuid
  - temporary privilege revocation (effective user ID)

# main() în ping.c

```
int
main(int argc, char **argv)
{
        struct hostent *hp;
        int ch, hold, packlen;
        int socket_errno;
        u_char *packet;
        char *target, hnamebuf[MAXHOSTNAMELEN];
        char rspace[3 + 4 * NROUTES + 1]; /* record route space */

        icmp_sock = socket(AF_INET, SOCK_RAW, IPPROTO_ICMP);
        socket_errno = errno;

        uid = getuid();
        if (setuid(uid)) {
                perror("ping: setuid");
                exit(-1);
        }
[...]
```

# CEL MAI MIC PRIVILEGIU

# Cel mai mic privilegiu

- Un proces nu are acces la mai multe privilegii sau resurse decât are nevoie
- Rulează ca utilizator cu permisiuni reduse
- Rulează într-un mediu izolat (jail, sandbox), cu acces doar la anumite resurse
- Rulează cu limitări: poate folosi resurse doar până într-o limită

# Permisiuni reduse

- Nu rulează ca root
- Utilizatorul care rulează nu are permisiuni de citire sau scriere pe fișiere de care nu are nevoie
- Procesul nu are capabilități de care nu are nevoie
- Procesului nu i se permite escaladarea de privilegii
- Dacă are nevoie de permisiuni de citire, nu se dau permisiuni de scriere
- Pentru permisiuni distincte se creează o altă entitate (separare de privilegii)

# Mediu izolat

- Acces strict la resursele necesare
- Poate fi vorba de sistemul de fișiere sau de operații posibile: pe socketi, apeluri de sistem, IPC-uri
- Jailing sau sandboxing
- În general avut în vedere pe dispozitivele mobile; aplicațiile rulează în câte un sandbox
- În cazul unui atac pagubele sunt limitate la nivelul sandbox-ului
- chroot, containere, virtualizare

# chroot

- Modifică directorul rădăcină asociat procesului.
  - nu se poate accesa un director/fișier din afara ierarhiei impuse de noul director rădăcină
  - chroot jail
- Comanda chroot
- Apelul chroot

```
chroot(“/var/spool/postfix”);
```

# Containere

- Namespace dedicat de procese și alte resurse
- chroot++ (nu doar la nivel de sistem de fișiere)
- Procesele pot să se vadă doar între ele într-un container
- Se partiționează accesul la anumite resurse
- OpenVZ, LXC, docker

# Virtualizare

- Mai multe avantaje pentru folosirea virtualizării
  - Un avantaj este izolarea
- Un întreg sistem de operare este izolat
- Orice probleme (bug-uri, vulnerabilități, performanță) sunt limitate la nivelul mașinii virtuale
- VMware, VirtualBox, KVM, Xen, Hyper-V

# Limitări

- Resursele să fie accesate până într-un prag
- Throttling
  - nu se poate trece de o viteză (un prag instantaneu de folosire): maxim 20% din procesor, maxim 512Kbps, maxim 10 procese pornite simultan
- Capping
  - nu se poate trece peste o limită totală de folosire a unei resurse: maxim 3GB de spațiu pe disc ocupat, maxim 2 ore timp pe procesor
- Previne atacuri de tipul denial of service

# getrlimit/setrlimit

- Apelurile folosite pentru limitarea resurselor în Unix
- În general o limită soft și o limită hard
  - Limita soft este cea folosită
  - Limita soft poate fi schimbată până în limita hard
  - Limita hard este impusă de sistem
- Exemple: dimensiune maximă spațiu de adresă, număr maxim de procese, număr maxim de fișiere deschise, dimensiunea maximă a unui fișier
- Folosit de /etc/security/limits.conf și comanda ulimit
- Apelul getrusage pentru contabilizarea folosirii resursei

# Cote

- limitări la nivelul sistemului de fișiere
  - număr maxim de fișiere care să fie create de un utilizator
  - dimensiunea maximă a spațiului ocupat de fișierele unui anumit utilizator
- în Linux 4 valori de configurat la nivel de utilizator/grup
  - limitarea numărului de fișiere/inode-uri (soft/hard)
  - limitarea spațiului ocupat la nivel de blocuri (soft/hard)
  - soft: limită soft, se trimite un warning
  - hard: limită hard, se interzice trecerea peste
- necesită suportul sistemului de fișiere

# IDENTITATEA UTILIZATORILOR

# Autentificarea utilizatorilor

- Accesul utilizatorilor în sistem
- Parolă
- Cheie publică
- Voice recognition, identificatori biometrici

# /etc/passwd + /etc/shadow

- /etc/passwd
  - user:password_hash:uid:gid:...
  - problemă
    - accesul utilizatorilor (nevoie de informații diferite de password_hash)
- /etc/shadow
  - user:password_hash:...
  - security enforcing
    - număr de zile între schimbat parola
    - număr de zile după care contul este dezactivat
    - ....

# Password hash în Unix

- man 3 crypt
- Implicit DES
- `$id$salt$encrypted`
- ID: 1 (MD5), 2a (Blowfish), 5 (SHA-256), 6 (SHA-512)
- salt este folosit pentru a adăuga un nivel suplimentar de criptare a parolei
  - un salt pe 12 biți înseamnă 4096 de posibilități de criptare

# Problema parolelor

- Pot fi ghicite, uitate, slabe
- 6.46 million LinkedIn passwords leaked online (June 2012)
  - http://www.zdnet.com/article/6-46-million-linkedin-passwords-leaked-online/
- $12,000 computer (Project Erebus v2.5), 8 AMD Radeon HD7970 GPU cards (August 2012)
  - are nevoie de 12 ore pentru a folosi brute force pe întreg spațiul de parole de 8 caractere printabile
  - http://arstechnica.com/security/2012/08/passwords-under-assault/

# Forme de enforcing pentru parole

- Anumite tipuri de caractere
- Expirarea parolei după un interval
- Passphrase în loc de parole
- Folosirea altor forme de autentificare (chei publice)
- One Time Password (OTP)

# Autentificarea prin chei publice

- Cheie publică + cheie privată
- Cheia publică este pe sistem (server)
- Cheia privată este folosită pentru autentificare
- Legătură matematică
  - one way function
  - de fapt este two-way, insa nu este computațional fezabil să se calculeze inversul funcției (încă..)
- RSA, DSA

# OTP

- One Time Password
- Time-synchronized OTP
  - RSA SecurID
- Algoritm matematic
  - s – initial seed
  - f – one-way function
    - cryptographic hash function
      - it is easy to compute the hash value for any given message,
      - it is infeasible to find a message that has a given hash,
      - it is infeasible to modify a message without changing its hash,
      - it is infeasible to find two different messages with the same hash.
  - f(f(f(f(...f(s)...))), ..., f(s)

# Cuvinte cheie

- sistem sigur
- bug
- vulnerabilitate
- atac
- separarea privilegiilor
- escaladarea privilegiilor
- kernel mode
- root
- setuid
- utilizatori
- DAC
- MAC
- liste de acces
- permisiuni
- cel mai mic privilegiu
- sandboxing
- chroot
- containere
- virtualizare
- limitări
- setrlimit/getrlimit
- ulimit
- Cote
- /etc/passwd, /etc/shadow
- password hash
- OTP