

# Securitatea sistemelor de calcul

Introducere

Marius Minea

28 septembrie 2010

# Ce conține acest curs ?

Securitatea *sistemelor*

sistem de operare + aplicații

securitatea rețelelor

# Ce conține acest curs ?

Securitatea *sistemelor*

sistem de operare + aplicații  
securitatea rețelelor

*Programare* cu accent pe securitate

vulnerabilități posibile și metode de prevenire

# Ce conține acest curs ?

## Securitatea *sistemelor*

sistem de operare + aplicații  
securitatea rețelelor

## *Programare* cu accent pe securitate

vulnerabilități posibile și metode de prevenire

## *Criptografie*

fundamente pentru întreg domeniul de securitate

# Ce conține acest curs ?

## Securitatea *sistemelor*

sistem de operare + aplicații  
securitatea rețelelor

## *Programare* cu accent pe securitate

vulnerabilități posibile și metode de prevenire

## *Criptografie*

fundamente pentru întreg domeniul de securitate

## *Protocoale* de securitate și modelarea lor

exemple din diverse domenii

principii și tehnici de modelare și analiză

## Ce este securitatea ?

“Security is [...] preventing adverse consequences from the intentional and unwarranted actions of others” [Bruce Schneier, *Beyond Fear*]

“Computer Security deals with the *prevention* and *detection* of *unauthorized* actions by users of a computer system” [D. Gollmann]

## Ce este securitatea ?

“Security is [...] preventing adverse consequences from the intentional and unwarranted actions of others” [Bruce Schneier, *Beyond Fear*]

“Computer Security deals with the *prevention* and *detection* of *unauthorized* actions by users of a computer system” [D. Gollmann]

Un sistem de securitate *previne* atacuri  
posibil și: detecție, recuperare/reparare

## Ce este securitatea ?

“Security is [...] preventing adverse consequences from the intentional and unwarranted actions of others” [Bruce Schneier, *Beyond Fear*]

“Computer Security deals with the *prevention* and *detection* of *unauthorized* actions by users of a computer system” [D. Gollmann]

Un sistem de securitate *previne* atacuri  
posibil și: detecție, recuperare/reparare

Securitatea tratează acțiuni *intenționate*  
acțiuni întâmplătoare: *siguranța* (safety  $\neq$  security !)



## Ce este securitatea ?

“Security is [...] preventing adverse consequences from the intentional and unwarranted actions of others” [Bruce Schneier, *Beyond Fear*]

“Computer Security deals with the *prevention* and *detection* of *unauthorized* actions by users of a computer system” [D. Gollmann]

Un sistem de securitate *previne* atacuri  
posibil și: detecție, recuperare/reparare

Securitatea tratează acțiuni *intenționate*  
acțiuni întâmplătoare: *siguranța* (safety  $\neq$  security !)

Interesează acțiuni *neautorizate* (dpdv al victimei); nu neapărat ilegale

## Ce este securitatea ?

“Security is [...] preventing adverse consequences from the intentional and unwarranted actions of others” [Bruce Schneier, *Beyond Fear*]

“Computer Security deals with the *prevention* and *detection* of *unauthorized* actions by users of a computer system” [D. Gollmann]

Un sistem de securitate *previne* atacuri  
posibil și: detecție, recuperare/reparare

Securitatea tratează acțiuni *intenționate*  
acțiuni întâmplătoare: *siguranța* (safety  $\neq$  security !)

Interesează acțiuni *neautorizate* (dpdv al victimei); nu neapărat ilegale

Implică existența unui *atacator*, care vizează *resurse* (assets)

# Cum păstrăm securitatea?

Cunoscând:

detalii tehnice (sisteme de operare, rețele, programare, criptografie)

# Cum păstrăm securitatea?

Cunoscând:

detalii tehnice (sisteme de operare, rețele, programare, criptografie)

Gândind:

*security mindset* [v. Schneier]

ca un atacator (aspecte tehnice și sociale!)

# Cum păstrăm securitatea?

Cunoscând:

detalii tehnice (sisteme de operare, rețele, programare, criptografie)

Gândind:

*security mindset* [v. Schneier]

ca un atacator (aspecte tehnice și sociale!)

Înțelegând:

noțiuni fundamentale: ce trebuie protejat? cum? care sunt atacurile?  
principii (de proiectare / construcție): generale, nu neapărat tehnice

## Cum evaluăm securitatea?

[ B. Schneier, *Beyond Fear* ]

1. Ce resurse se doresc a fi protejate ?
2. Care sunt riscurile la care sunt supuse ?
3. Cat de mult reduce soluția de securitate aceste riscuri ?
4. Ce alte riscuri apar *datorită* soluției alese ?
5. Ce costuri și compromisuri implică soluția aleasă ?

# Obiective de securitate

## Confidențialitate

- protejarea (ascunderea) informației sau resurselor
- în mod tipic prin criptografie
  - sau alte mecanisme (nefăcute publice)
- poate fi confidențială chiar  *existența*, nu doar  *conținutul*
- inclusiv ascunderea resurselor

## Integritate

Availability (*disponibilitate*)

# Obiective de securitate

*confidențialitate*

*integritate*

= încrederea în date sau resurse

– exprimată prin prevenirea modificărilor neautorizate

– distingem:

– integritatea *conținutului* datelor

– integritatea *originii* (autentificare)

Mecanisme legate de integritate:

– mecanisme de *prevenție*

a modificării neautorizate a datelor (ex. din exterior)

a modificării datelor în moduri neautorizate (ex. din interior)

– mecanisme de *detecție*

[M. Bishop: Computer Security: Art and Science, Pearson, 2003]

*disponibilitate*



# Obiective de securitate

*confidențialitate*

*integritate*

*disponibilitate*

= posibilitatea de a folosi o informație sau resursă în modul dorit

– un sistem care nu e disponibil poate fi mai rău ca unul inexistent

– disponibilitatea e analizată de obicei în condițiile unor presupuneri (de ex. statistice) asupra mediului extern

    dacă presupunerile nu sunt satisfăcute, sistemul poate fi compromis

– *denial of service attacks* – pot fi dificil de detectat, dacă se încadrează (parțial) în tiparul statistic permis

## Obiective de securitate: alte clasificări

**P**rivacy, **A**vailability-Authentication, **I**ntegrity, **N**on-repudiation

Hexada Parkeriană (Donn Parker, 2002)

confidențialitate,

*posesiune/control* (importantă și fără a viola confidențialitatea)

integritate

*autenticitate* (a originii sau autorului)

disponibilitate

*utilitate* (ex. date convertite în format inutil  $\neq$  disponibilitate)

# Alte obiective de securitate

[Handbook of Applied Cryptography]

semnătură

autorizare

controlul accesului

marcare temporală / *timestamping*

dovadă / *witnessing* (de altcineva decât originatorul)

confirmare

anonimitate

revocare

trasabilitate / *accountability*

## Amenințări (threats)

- confidențialitatea, integritatea, disponibilitatea sunt *servicii* oferite
- discutăm *amenințări* (potențiale) și *atacuri* (reale) asupra serviciilor

Clasificarea amenințărilor [R. Shirey, cf. M. Bishop]

- dezvăluire (disclosure)
- decepție (forțarea acceptării de date false)
- disruption – întreruperea / împiedicarea funcționării corecte
- uzurpare = controlul neautorizat al unei părți a sistemului

# Mecanisme de amenintare

Microsoft STRIDE threat model

*Spoofing* identity - impersonare

*Tampering* with data - falsificare / atac la integritate

*Repudiation* - negarea efectuării unei acțiuni

*Information disclosure* - atac la confidențialitate

*Denial of service* - atac la disponibilitate

*Elevation of privilege* - creșterea neautorizată a drepturilor

# Mecanisme de amenințare

intercepție (snooping)

caz particular: (passive) wiretapping

modificare / alterare (a datelor) ⇒ decepție

dar și intrerupere / uzurpare (dobândirea controlului)

active wiretapping, man-in-the-middle attack

(modificarea activă a conținutului)

impersonare (masquerading, spoofing)

repudierea originii (e.g. în tranzacții comerciale)

*denial of receipt* – o formă de decepție

întârzierea – poate fi întrerupere de serviciu, sau chiar uzurpare

*denial of service*

# Principii de proiectare a sistemelor, vizând securitate

[Saltzer & Schroeder: The Protection of Information in Computer Systems, 1975]

- a) *Economy of mechanism*: design cât mai simplu, verificare facilă  
⇒ parte integrantă la proiectare, nu adăugate ulterior

# Principii de proiectare a sistemelor, vizând securitate

[Saltzer & Schroeder: The Protection of Information in Computer Systems, 1975]

- a) *Economy of mechanism*: design cât mai simplu, verificare facilă  
⇒ parte integrantă la proiectare, nu adăugate ulterior
  
- b) *Fail-safe defaults*: implicit, accesul e interzis + reguli pt. permisiuni  
NU invers: implicit permis, reguli de interzicere



# Principii de proiectare a sistemelor, vizând securitate

[Saltzer & Schroeder: The Protection of Information in Computer Systems, 1975]

- a) *Economy of mechanism*: design cât mai simplu, verificare facilă  
⇒ parte integrantă la proiectare, nu adăugate ulterior
- b) *Fail-safe defaults*: implicit, accesul e interzis + reguli pt. permisiuni  
NU invers: implicit permis, reguli de interzicere
- c) *Complete mediation*: orice acces trebuie verificat  
(inclusiv cazuri de excepție, mentenanță, etc.)  
NU se bazează pe decizii luate/memorate anterior

# Principii de proiectare a sistemelor, vizând securitate

[Saltzer & Schroeder: The Protection of Information in Computer Systems, 1975]

- a) *Economy of mechanism*: design cât mai simplu, verificare facilă  
⇒ parte integrantă la proiectare, nu adăugate ulterior
- b) *Fail-safe defaults*: implicit, accesul e interzis + reguli pt. permisiuni  
NU invers: implicit permis, reguli de interzicere
- c) *Complete mediation*: orice acces trebuie verificat  
(inclusiv cazuri de excepție, mentenanță, etc.)  
NU se bazează pe decizii luate/memorate anterior
- d) *Open design*: securitatea nu trebuie să se bazeze pe păstrarea secretă a mecanismelor (not: security through obscurity)  
⇒ mecanismele pot fi analizate public, pentru creșterea încrederii

## Saltzer and Schroeder (cont.)

e) *Separation of privilege*: separarea crește robustețea

## Saltzer and Schroeder (cont.)

- e) *Separation of privilege*: separarea crește robustețea
- f) *Least privilege*: fiecare program și utilizator ar trebui să opereze cu setul de privilegii minim necesar pentru sarcina dată

## Saltzer and Schroeder (cont.)

- e) *Separation of privilege*: separarea crește robustețea
- f) *Least privilege*: fiecare program și utilizator ar trebui să opereze cu setul de privilegii minim necesar pentru sarcina dată
- g) *Least common mechanism*: minimizați resursele comune, interferența între utilizatori, mecanismele pe care se bazează toți

## Saltzer and Schroeder (cont.)

- e) *Separation of privilege*: separarea crește robustețea
- f) *Least privilege*: fiecare program și utilizator ar trebui să opereze cu setul de privilegii minim necesar pentru sarcina dată
- g) *Least common mechanism*: minimizați resursele comune, interferența între utilizatori, mecanismele pe care se bazează toți
- h) *Psychological acceptability*:  
să nu interfereze nepotrivit cu activitatea obișnuită  
dacă mecanismele nu sunt simple, vor fi utilizate greșit sau ocolite

## Saltzer and Schroeder (cont.)

- e) *Separation of privilege*: separarea crește robustețea
- f) *Least privilege*: fiecare program și utilizator ar trebui să opereze cu setul de privilegii minim necesar pentru sarcina dată
- g) *Least common mechanism*: minimizați resursele comune, interferența între utilizatori, mecanismele pe care se bazează toți
- h) *Psychological acceptability*:  
să nu interfereze nepotrivit cu activitatea obișnuită  
dacă mecanismele nu sunt simple, vor fi utilizate greșit sau ocolite

Suplimentar:

*Work factor*: comparați efortul necesar cu resursele atacatorului

*Compromise recording*: în caz de eșec, o alarmă e totuși utilă

## Principii de securitate (cont.)

principiul verigii cele mai slabe (weakest link)

este cea care determina securitatea întregului sistem

principiul protecției adecvate

nu securitate maximă, ci utilitate la cost/risc acceptabil

principiul eficienței (v. și acceptabilității):

potrivite, ușor de folosit, pentru a fi folosite (și în plus, corect)

principiul apărării în adâncime (defense in depth)

mai multe nivele de protecție

[Ninghui Li, CS 426: Computer Security, curs, Purdue University]



## O taxonomie a incidentelor

[după J.D. Howard, P. Meunier, in Handbook of Computer Security]

Principalul obiect de studiu: *eveniment* (incident legat de securitate)

- compus dintr-o *acțiune* executată asupra unei *ținte*
- acțiunea poate fi executată cu o *unealtă*
- exploatănd un anumit tip de *vulnerabilitate*
- cu un anumit *rezultat* (în mod normal neautorizat)

## Acțiuni în cadrul unui atac

- “probe”: a accesa o țintă pentru a-i determina anumite caracteristici
- “scan”: accesul sistematic (“probe”) la mai multe ținte
- “flood”: accesul repetat la o țintă pentru a o supraîncărca
- autentificare: prezentarea unei identități pentru verificare și acces ulterior
- circumvenție (bypass): ocolirea unui proces (de control/autorizare) prin folosirea unei metode alternative de a accesa o țintă
- spoof/masquerade: a-și asuma identitatea
- citire
- copiere
- sustragere (luare în posesie și eliminarea originalului)
- modificare
- ștergere

# Taxonomie a incidentelor (continuare)

*Ținta* unei acțiuni:

- entități logice (cont, proces, date)
- entități fizice (componentă, calculator, rețea, ansamblu de rețele)

*Atac*

- serie de acțiuni (ale unui atacator) împotriva unei ținte
- efectuate *cu intenție*
- pentru a obține un *rezultat neautorizat*

*Vulnerabilitate* – la nivel de

- proiectare
- implementare
- configurare

## Taxonomie a incidentelor (continuare)

*Unelte* într-un atac:

- atac fizic (furt, distrugere, scoatere din uz)
- schimb de informații (la nivel uman) – *social engineering*
- comandă utilizator (folosind interfață disponibilă, ex. telnet pe port)
- script sau program (tot interfața la nivel de proces, dar automatizat: shell script, troian, program de spart parole)
- agent autonom: acționează independent de utilizator (virus, worm)
- pachet de unelte (toolkit): set de scripturi/programe/agenți, e.g. rootkit
- unealtă distribuită (sisteme multiple, cu atac temporizat coordonat)
- “data tap” – acces direct la date (radiatie electromagnetică, etc.)

## Rezultatul unui atac

- acces neautorizat (suplimentar) la un sistem sau o rețea
- dezvăluire de informație (atac la confidențialitate)
- corupere de informație (atac la integritate)
- denial of service (atac la disponibilitate)
- furt de resurse (folosire neautorizată) – caz particular de uzurpare

## Securitatea: probleme generale [Schneier]

- moduri de eroare: pasiv vs. activ (nu face vs. face ce nu trebuie)
- pericolul erorilor în cazuri rare
- *security imbalances* – efectul tehnologiilor pe scară largă
- sisteme fragile (brittle) vs. reziliente la erori
- metode de protecție *adaptive* la situații neprevăzute
- monoculturi (sistemelor omogene) – vulnerabile la același atac
  - e.g. majoritatea covârșitoare a sistemelor rulează Windows...
- securitatea e o problemă umană / socială

# Security and Trust

În securitate, facem *afirmații* despre diverse entități.

Aceste afirmații nu sunt *absolute*, se bazează pe anumite *presupuneri*.

⇒ securitatea e o problemă de încredere: în cine și ce putem avea încredere?

Ken Thompson: Reflections on Trusting Trust (Turing Award Lecture '83)  
inserarea unui *troian* în programul de login și compilatorul de C  
pentru a accepta și o parolă specială, cunoscută de el  
prin folosirea de cod care se *autoreproduce*

“You can't trust code that you did not create yourself”

“No amount of source-level verification or scrutiny will prevent you from using untrusted code”

## Exemplu: protecția fișierelor în UNIX (recap.)

- fiecare fișier e identificat prin utilizatorul și grupul “proprietar”
- biți separați pentru citire (r), scriere (w), execuție/căutare (x)
- pentru fiecare: proprietar/user (u), grup (g), restul/others (o)

Semnificația pentru *directoare*: mai complicată decât pentru fișiere:

- **r** e necesar pentru `read()`, `readdir()`, `opendir()`  $\Rightarrow$  pt. `ls`
- **x** (numit și “search”) e necesar pentru `chdir()` și `stat()` pe un fișier



## Permisiuni pentru fișiere în UNIX

Ce permisiuni necesită citirea unui fișier ?

## Permisiuni pentru fişiere în UNIX

Ce permisiuni necesită citirea unui fişier ?

**x** pe toată calea și **r** pe fişier

## Permisiuni pentru fişiere în UNIX

Ce permisiuni necesită citirea unui fişier ?

**x** pe toată calea și **r** pe fişier

Ce permisiuni necesită `ls -l` ?

## Permisiuni pentru fişiere în UNIX

Ce permisiuni necesită citirea unui fişier ?

**x** pe toată calea și **r** pe fişier

Ce permisiuni necesită `ls -l` ?

necesită informații din *inode*, deci **x** pe director

## Permisiuni pentru fişiere în UNIX

Ce permisiuni necesită citirea unui fişier ?

**x** pe toată calea și **r** pe fişier

Ce permisiuni necesită `ls -l` ?

necesită informații din *inode*, deci **x** pe director

Ce permisiuni necesită ștergerea unui fişier ?

## Permisiuni pentru fişiere în UNIX

Ce permisiuni necesită citirea unui fişier ?

**x** pe toată calea și **r** pe fişier

Ce permisiuni necesită `ls -l` ?

necesită informații din *inode*, deci **x** pe director

Ce permisiuni necesită ștergerea unui fişier ?

**w** în director, și **x**

Nu necesită **w** pentru fişier!

## Permisiuni pentru fişiere în UNIX

Ce permisiuni necesită citirea unui fişier ?

**x** pe toată calea și **r** pe fişier

Ce permisiuni necesită `ls -l` ?

necesită informații din *inode*, deci **x** pe director

Ce permisiuni necesită ștergerea unui fişier ?

**w** în director, și **x**

Nu necesită **w** pentru fişier!

Ce se poate face având **x** pe director dar nu **r** ?

## Permisiuni pentru fişiere în UNIX

Ce permisiuni necesită citirea unui fişier ?

**x** pe toată calea și **r** pe fişier

Ce permisiuni necesită `ls -l` ?

necesită informații din *inode*, deci **x** pe director

Ce permisiuni necesită ștergerea unui fişier ?

**w** în director, și **x**

Nu necesită **w** pentru fişier!

Ce se poate face având **x** pe director dar nu **r** ?

Se poate ajunge la un fişier cunoscut, dar nu se poate căuta un fişier



## Permisiuni pentru fişiere în UNIX

Ce permisiuni necesită citirea unui fişier ?

**x** pe toată calea și **r** pe fişier

Ce permisiuni necesită `ls -l` ?

necesită informații din *inode*, deci **x** pe director

Ce permisiuni necesită ștergerea unui fişier ?

**w** în director, și **x**

Nu necesită **w** pentru fişier!

Ce se poate face având **x** pe director dar nu **r** ?

Se poate ajunge la un fişier cunoscut, dar nu se poate căuta un fişier

– biți speciali:

– sticky bit: pt. director: fişierul poate fi șters doar de proprietar

– set user ID: execută având ca ID *efectiv* utilizator proprietarul fişierului

– set group ID: execută având ca ID *efectiv* de grup grupul fişierului