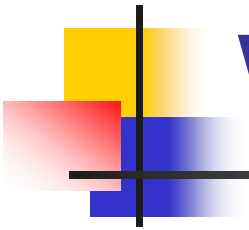
Practical computer security

Željko Vrba
University of Oslo

(includes content by: Vera Goebel, Matija Pužar and Andrew Tannenbaum)



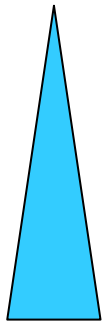
What is computer security?

- Very broad term
- File system security
- Network security
- Data integrity and confidentialty
- Privilege separation (local system)
- Data loss
- Physical security
- Social engineering

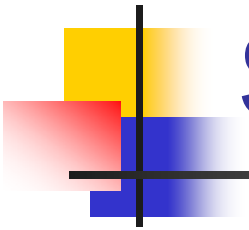
Threat model

- 1st step in designing a system: define TM
  - WHAT is being protected from WHOM?
  - internal/external attack
  - active/passive intruders
- ideally, the system should be **designed** upfront for security; not added later

Casual prying by non-technical users
Script-kiddies
Snooping by insiders
Determined attempt to make money
Commercial, military or government espionage



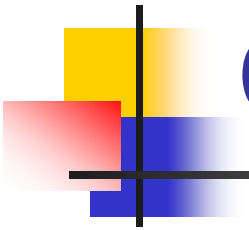
level of effort (and cost!) needed to protect the system



System design

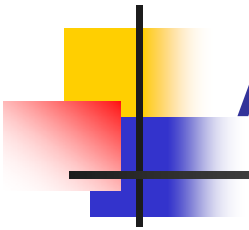
1. System design should be public
   - security by obscurity does not work
2. Reasonable defaults (no/minimum access)
3. Check for current authority, not only at the beginning
4. Give each process least privilege possible
5. Protection mechanism should be
   - simple
   - uniform
   - in lowest layers of system – security is not an add-on feature
6. Scheme should be psychologically acceptable (consider fingerprints, iris scans, etc.)

And… keep it simple



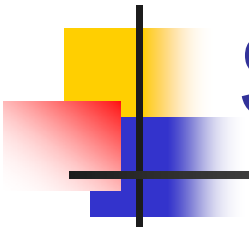
OS security model

- none (single-user OS, e.g. DOS)
- UNIX (owner,group,others) rights
  - each user is a member of at least one group
- Windows NT
  - users and groups; more complex than UNIX, esp. in a domain
- prerequisite: user authentication



Authentication

- Gaining access to the system
- Something the user knows
  - username + password, pass-phrase
- Something the user has
  - physical object (magnetic card, smartcard)
- Something the user is
  - physical characteristics (biometrics)
- Multi-factor authentication



Single-sign on

- Many services requiring multiple passwords
- Users write down their passwords
- Solution: SSO
- Kerberos (http://web.mit.edu/kerberos/www/)
  - resilient against network attacks
- Variant used on NT domains

UNIX permissions

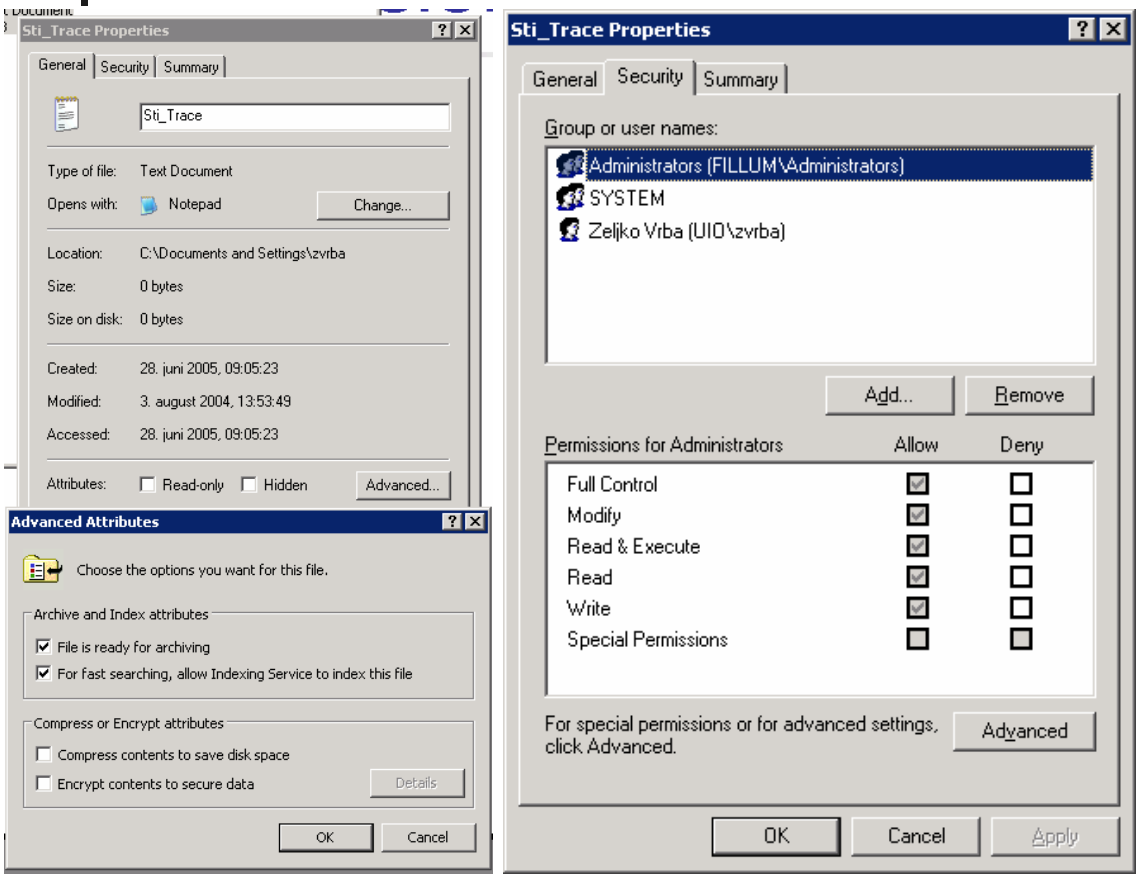
```
-rw-r--r--    1 zvrba    ifi-a00     96031 2005-06-12 21:20 workplan.pdf
drwx--x--x    2 zvrba    ifi-a00      1024 2005-11-05 16:34 www_docs/
-rw-------    1 zvrba    ifi-a00      2402 2005-10-12 08:05 xorg.conf
```

owner, group, others; **r**ead, **w**rite, e**x**ecute; sometimes surprising semantics (e.g. delete files you don't own)

SUID, SGID and sticky bits

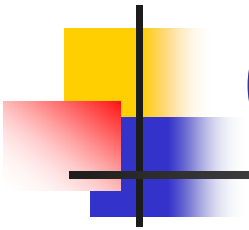
recently: ACLs on files (getfacl, setfacl commands)

Windows permissions



both users and groups in the ACL!

security descriptors for each **object**



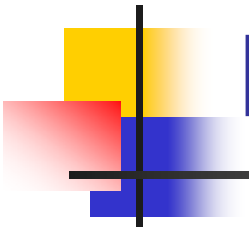
(Accidental) data loss

Common Causes

1. Acts of God: fires, floods, wars
2. Hardware or software errors: CPU malfunction, bad disk or memory, program bugs
3. Human errors: data entry, wrong tape mounted

Often cause more damage than intruders!

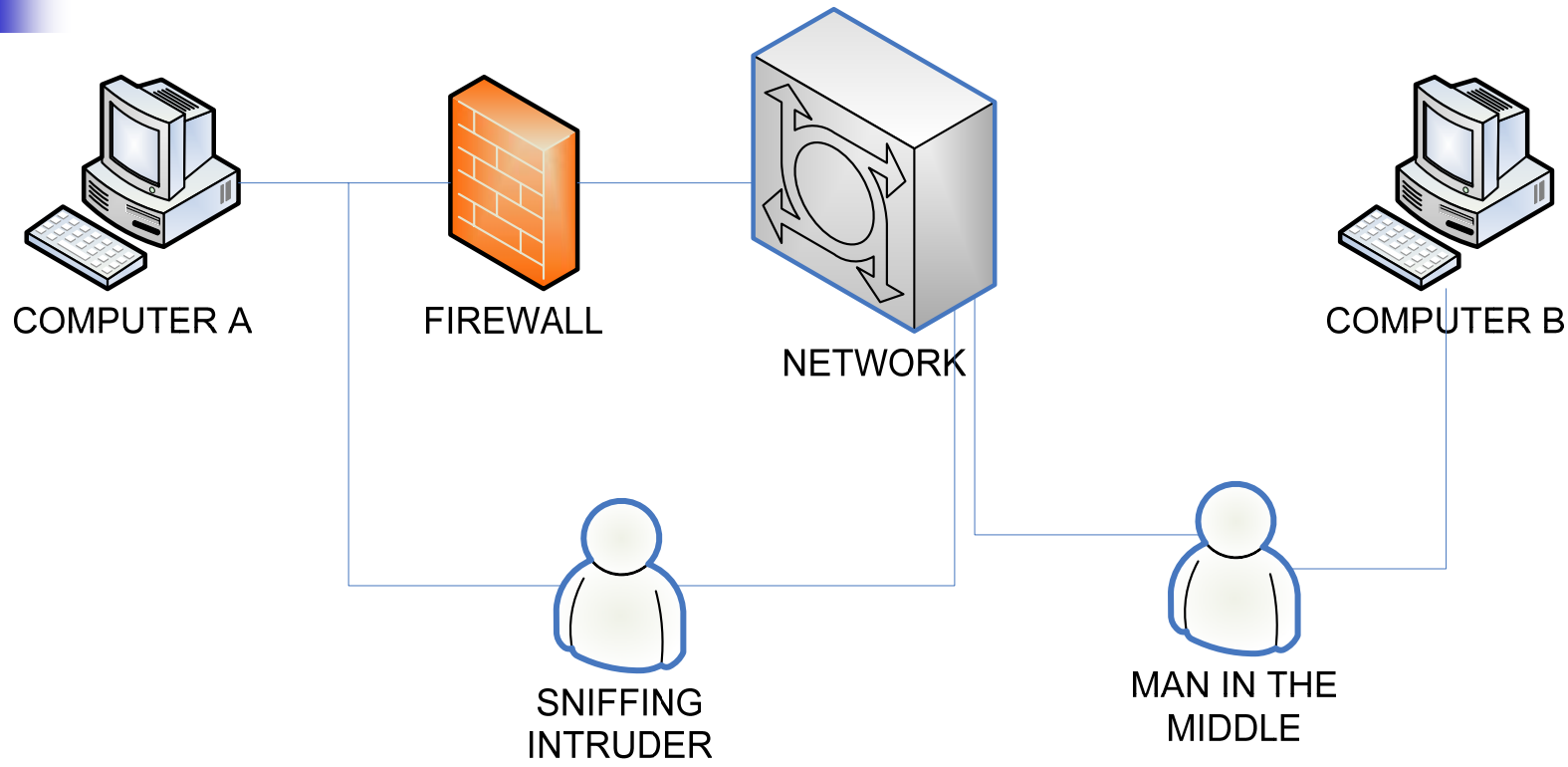
BACKUP! (data and location)



File system encryption

- encryption: data forever lost if key (or passphrase) is lost; **backup!**
- individual file vs. partition encryption
- transparent vs. non-transparent (separate program)
- windows upgrades and reinstallations can destroy the original EFS key!

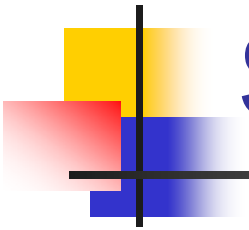
Network attacks



data sniffing and/or modification; replay and man-in-the-middle attacks

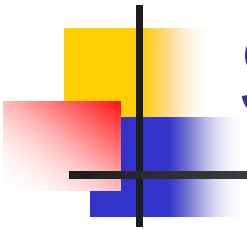
use ssh, gpg/pgp and https: provide both data integrity and confidentiality

a question of **trust!**



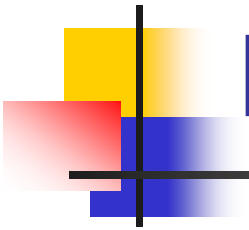
Spoofing and phishing

- trick users into typing personal data into non-authentic (but plausible!) forms
- e.g. frequent PayPal spam
- recently: Nordea bank suffered such attack
- always double check mails, mail headers and URLS
- never reveal sensitive data unless the query is expected (and even then be cautious!)



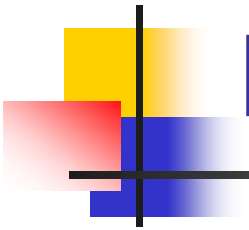
System integrity

- Integrity of all system components (configuration files, binaries, etc.)
  - tools like tripwire
- Discretionary access controls: UNIX
- Mandatory access constrols: the administrator is in charge



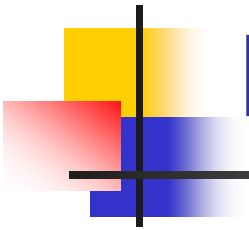
Reasonable defaults

- e.g. NetBSD vs. Linux sendmail installation
- make use of mount options
  - /usr read-only
  - /var read-write, noexec, nodev
  - nosuid where applicable



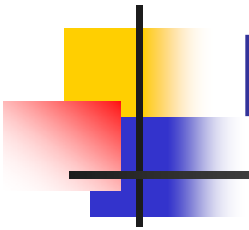
Mandatory access controls

- Linux
  - grsecurity (http://www.grsecurity.org)
  - SELinux (http://www.nsa.gov/selinux)
  - can cause compatibility problems with “badly behaved” software
- FreeBSD
  - MAC (http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/mac.html)
  - BIBA and MLS policies
- Proactive: confine damage if break-in occurs



MAC example: Bell-LaPadula

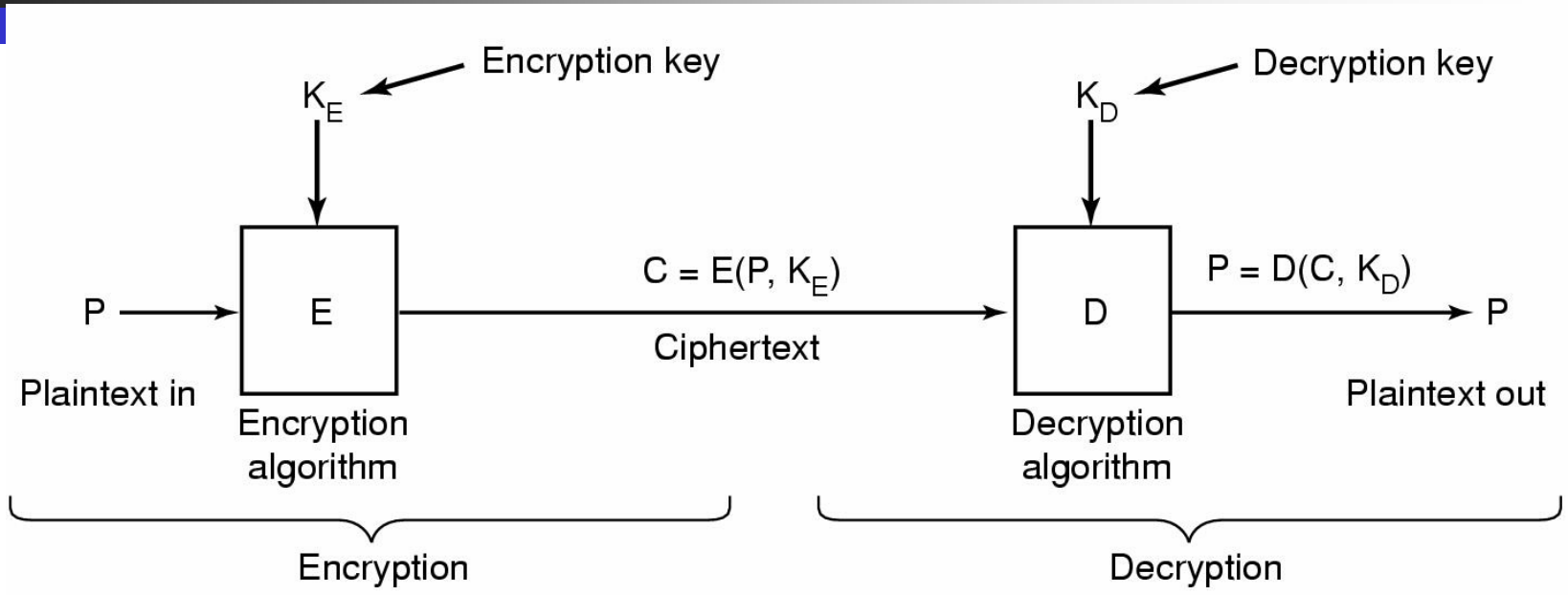
- Solve the confinement problem (information leakage)
- each object has a classification (label)
- each subject has a clearance and a security level
- two fundamental axioms:
  - the simple security property
  - the *-property



Physical security

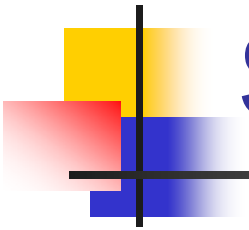
- Often neglected, but:
  - physical access is unrestricted access
  - do you trust your system administrator?
- Precautions:
  - data encryption (only cold storage safe!)
  - remove all recording devices
  - faraday cage

Cryptography



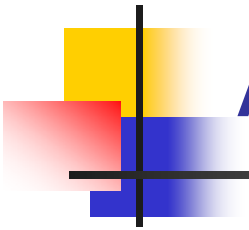
an important **foundation**, but far from enough; **secure protocols** are also needed. **NEVER design your own, use proven and tested solutions, e.g. OpenSSL (http://www.openssl.org)**

random numbers often forgotten, but crucial part of secure system (fatal Netscape flaw)



Symmetric cryptography

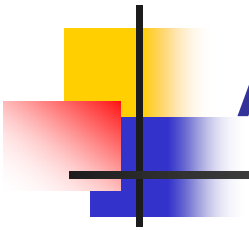
- One key both for encryption and decryption
- Good algorithm is **publicly reviewed**
  - DES (being phased out, replaced by AES/Rijndael)
  - IDEA, Blowfish, Twofish, Serpent, etc.
- Problem: secure key exchange



Asymmetric cryptography (1)

- All users pick (public, private) *key pair*
  - public key cryptography
- Digital signatures and encryption
  - public key: encryption/verification
  - private key: decryption/signing
- algorithms:
  - DSA, RSA (1024 bits min. secure today)
  - ECC has smaller key size

Asymmetric cryptography (2)



- Slow; mixed scheme is used:
  - *random symmetric* key for bulk encryption
  - encrypted by public key of recipient
- Trust problems
  - Is the key owner really he who claims to be?
  - hierarchical: X509 certificate authorities (e.g. Verisign) and why should we trust them?
  - DAG: PGP
  - other: ssh (personal verification)
- Key distribution: public servers (e.g. LDAP) and how trustworthy they are?

Digital signatures

- rely on one-way hash functions as cryptographic primitive
  - SHA-0, and MD5 (**both broken!**)
  - SHA-1 (considerabely weakened), SHA256, etc.
  - RIPEMD 160
- ensure **data integrity** and **authenticity**