Introduction to Computer Security
Identification and Authentication

Pavel Laskov
Wilhelm Schickard Institute for Computer Science
Resource access: a big picture

1. Identification
Which object $O$ requests access to resource $R$?

2. Authentication
Is $O$ indeed what it claims to be?

3. Authorization
Is $O$ authorized to access $R$?

4. Accountability
Log the transaction access $(O, R)$
Simple identification/authentication

- Ask for an identity (e.g. user name)
- Check if an identity is known
- Ask for a shared secret (e.g. a password)
- Check if the secret matches the identity
- Password is stored in an encrypted form using a one-way hash function:
  
  root:!:14118:0:
  laskov:$1$/et/grJh$xssVNwpdA35TwsSt7Yjvb/:14118:0:
Identity management

- A digital identity is a set of properties assigned to a given object (e.g. access rights)

Identity management:
- Creation and deletion of identities
- Management of properties assigned to identities
- Secure storage of identities
- Secure handling of queries regarding identities and their properties.
A directory is a specialized database optimized for searching and browsing.

LDAP entries are collections of attributes identified by a unique distinguished name (dn).

Entries are characterized by types that determine their format and syntax (e.g. ou = “Organisational Unit”).

Entries are stored in a hierarchy. A relative distinguished name defines a search path to an entry.
LDAP directory example

Directory Root (Top)

- o=IBM
  - ou=Marketing
    - cn=mbarlen
      - objectClass=Person
      - objectClass=ePerson
      - mail=marion@ibm.com
      - sn=Barlen
      - givenName=Marion
      - telephoneNumber=112
    - cn=Klaus
      - objectClass=Person
      - objectClass=ePerson
      - mail=Ktebbe@ibm.com
      - sn=Tebbe
  - ou=Support
    - cn=tbarlen
      - objectClass=Person
      - objectClass=ePerson
      - mail=thomas@acme.com
      - sn=Barlen
- c=us
  - o=ACMESupply
    - cn=tbarlen
      - objectClass=Person
      - objectClass=ePerson
      - mail=thomas@acme.com
      - sn=Barlen
      - deviceID=PrinterSales
      - objectClass=cimPrinter
      - objectClass=ePrinter
      - location=Printer room 3rd floor
      - owner=John Doe
      - QueueName=Lsprt01
      - maxCopies=10
  - o=iSeriesShop
    - cn=tbarlen
      - objectClass=Person
      - objectClass=ePerson
      - mail=thomas@acme.com
      - sn=Barlen
      - deviceID=PrinterSales
      - objectClass=cimPrinter
      - objectClass=ePrinter
      - location=Printer room 3rd floor
      - owner=John Doe
      - QueueName=Lsprt01
      - maxCopies=10
LDAP applications

- User account management (e.g. Apple Open Directory, POSIX Accounts, Microsoft Active Directory Service)
- Address books (Lotus Notes, Outlook, Thunderbird, Evolution)
- Authentication (e.g. PAM: Pluggable Authentication Module)
- User data in email servers and spam filters
## Authentication modes

<table>
<thead>
<tr>
<th></th>
<th>User authentication</th>
<th>Entity authentication</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Object</strong></td>
<td>human being</td>
<td>non-human being</td>
</tr>
<tr>
<td><strong>Applications</strong></td>
<td>- login</td>
<td>- remote access</td>
</tr>
<tr>
<td></td>
<td>- remote access</td>
<td>- communication security</td>
</tr>
<tr>
<td><strong>Speed</strong></td>
<td>low</td>
<td>high</td>
</tr>
<tr>
<td><strong>Secret only</strong></td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Detailed</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
User authentication

- **Something you know:** passwords, passphrases, shared secrets (e.g. mother’s maiden name), puzzles
- **Something you have:** smart cards, security tokens
- **Something you are:** biometrics, signature dynamics, keyboard dynamics, voice print
What is a good password?

- Long passwords are harder to break but tedious to type.
- Random passwords are hard to break but next to impossible to memorize.
- Writing down a password on a sticky note doesn’t help.
- Password expiration reduces the damage but increases the risk of forgetting.
- Automatic password generation
  - easily memorizable words
  - passphrases: sentences hashed into passwords
- Passwords are only usable for services that are often used.
Password generation examples

What length should your password be?

20

Include symbols in password? ✓

Generate password

Your new strong password is:

Q8k1m:PE#/B44-O/1B*4

Remember it as:

QUICK 8 kmart 1 microsoft :
PARIS ELVIS # / BRITNEY 4 4
- OPRAH / 1 BRITNEY * 4
Password generation examples

Strong Password Generator

What length should your password be?
20

Include symbols in password? ✓

Generate password

Your new strong password is:
Q8k1m:PE#/B44-O/1B*4

Remember it as:
QUICK 8 kmart 1 microsoft :
PARIS ELVIS # / BRITNEY 4 4
- OPRAH / 1 BRITNEY * 4

This online tool helps you to choose a randomly generated password that is easy to read and remember.

- ergoshreathauClinicity
- O8xourTarpIonsion
- Geowanwoupheneeness
- fíxWrieclorfrinkity
- comDoMiewrieness
- Umplingtrengkayment
- Interfinkltrydom
- banriaaruPleymer
- Llcpfrvoyslerwerwr
- geoshorhourcleyquity

Generating options:

Syllable counts: 3 ✓ Using Prefix: ✓ Using uppercase: ✓

Generate
Password maintenance

- Generation and distribution
  - manual: physical presence
  - automatic: off-band distribution (e.g. per post)
- Password synchronization
  - distribution of the same password to multiple systems
- Forgotten passwords:
  - self-service password reset
  - assisted password reset
Password threats

- Brute force search
- Guessing
- Dictionary attacks
- Keylogging
- “Shoulder surfing”
- Identity spoofing / phishing
- Social engineering
Smart cards

- Physical tamper-proof credentials
- Memory cards: simple storage of information (e.g. medical insurance)
- Microprocessor cards:
  - advanced functionality (e.g. transactions)
  - cryptographic operations (e.g. key validation and key-pair generation)
- Java cards: a mini-JVM on a chip (e.g. for code update).
- Main problem: theft and loss risk
Biometric identification

- Unforgeable features of a given person:
  - fingerprints
  - hand geometry
  - hand topography
  - iris scan
  - retina scan

- ...or behavioral characteristics:
  - signature dynamics
  - keyboard dynamics
  - voice print
Disadvantages of biometric authentication

- Non-zero probability of erroneous classification
- High cost and time consumption
- Impossibility to revoke
- Privacy issues and social acceptance
Two-factor authentication

- A combination of any two authentication modes
- Example: SecurID
  - PIN assigned to user
  - token automatically generated in hardware every 30 seconds
- Clock synchronization between a token generator and an authentication server required
Motivation:
- Reduce the need to memorize passwords
- Reduce the time spent on typing passwords
- Reducing password maintenance effort

Solutions:
- Local password containers (e.g. KDE Wallet)
- Physical authentication devices
- Client-server architectures (Kerberos, Active Directory)
Kerberos: a brief history

- Development at MIT in the project Athena in the 1980s, versions 1 – 3 for internal use, version 4 released in 1988.
- Version 5 released in 1993 fixed some security flaws of the previous version; version 4 withdrawn in 2006.
- Banned for export by US government until 2000 (due to the use of DES); re-implemented in Sweden at KTH.
- Currently, both implementations – MIT (Athena) and KTH (Heimdal) – support AES as the main encryption instrument.
- Supported by all major OS (Windows 2000 up, Mac OS X, Solaris, Red Hat Enterprize)
Kerberos design criteria

- **Security against eavesdropping:** no password transmission in clear text.
- **Reliability:** every use of a service requires prior authentication.
- **Transparency:** user is not aware of any authentication beyond an initial login.
- **Scalability:** support for a large number of servers and clients.
Kerberos authentication protocol

Client

Application Server

Key Distribution Center (KDC)

Authentication Server (AS)

Ticket Granting Server (TGS)

Database

AS_REQ

AS_REP

TGS_REQ

TGS_REP

AP_REQ

AP_REP
Kerberos details: user login

- User enters his user name $U$ and password $P$ at a client workstation.
- The hash value of his password $K_u = h(P)$ becomes a secret key of the client/user.
Kerberos details: user authentication

- The client sends his user ID $U$ and the requested service $S$ to the authentication server AS:

$$C \rightarrow AS : (U, S)$$

- If the user ID is found in a database, AS generates a TGS session key $k_{TGS}$ and a ticket-generating ticket $TGT$:

$$TGT : (U, S, k_{TGS})$$

- AS sends $TGT$ encrypted to C:

$$C \leftarrow AS : \{S, k_{TGS}\}_{K_u} \{TGT\}_{K_{TGS}}$$
The client generates an authenticator:

$$A : \{U\}_{k_{TGS}}$$

The client sends a ticket granting request to a ticket granting server:

$$C \rightarrow TGS : (S, A)\{TGT\}_{K_{TGS}}$$

TGS verifies that $U$ in $A$ matches $U$ in $TGT$, generates a service session key $k_S$ and puts it into a service ticket:

$$T_S : (U, S, k_S)$$

TGS sends $T_S$ encrypted with the service key $K_S$ to C:

$$C \leftarrow TGS : \{S, k_S\}_{k_{TGS}}\{T_S\}_{K_S}$$
Kerberos details: service authentication

- The client generates an authenticator:
  \[ A : \{U\}_{K_S} \]

- The client sends a request to a service:
  \[ C \rightarrow S : (A)\{T_S\}_{K_S} \]

- The service checks if \( U \) in \( A \) matches \( U \) in \( T_S \) and if so accepts the request.
Kerberos resumé

- Provides secure authentication in an insecure network.
- A de-facto standard (at least open-source) in distributed authentication services.
- Relatively complex in installation.
- Single point of failure.
Identity management and directory tools provide mechanisms for maintaining and searching for object identities.

Passwords and shared secrets are the most common albeit not sufficiently secure tool for user authentication.

Alternative user authentication mechanisms are smart cards, biometric methods and two-factor authentication.

Single sign-on methods such as Kerberos provide means for automating authentication in large distributed environments.