Security and Authentication
Authentication and Security

• A major problem with computer communication – Trust

• Who is sending you those bits

• What they allow to do in your system
Authentication

• In distributed systems, services are rendered in response to incoming messages.

• It is important that the server know for sure who the client is!

• The simple solution is to send the user name and password with every request
Kerberos History

- Developed at MIT in early 1980’s
- Computing shift from mainframes to workstations
- Pools of distributed workstations connected to servers
- Concept of “Network Credentials”
- Two commercial and non-compatible versions V4 and V5
- Principles and systems are relevant until today
Kerberos

• Authentication service, based on a secure authentication server and on encryption

• The server knows all passwords, but they are never transmitted across the network

• Passwords are used to generate encryption keys.
Kerberos Environment

Diagram showing the Kerberos environment with users, terminals, and servers connected to the KRB (Kerberos Key Distribution Center).
Kerberos Environment

Separation between two actions:

– Authentication – logging into the “network”

– Communication – holding a session between two parties
Kerberos Architecture
Kerberos Protocol

• The client workstation where the user is trying to log in sends the user name U to the server.

• The Kerberos server does the following:
  – It looks up the user’s password p, and uses a one-way function to create an encryption key $K_p$ from it.
  – It generates a new session key $K_s$ for this login session.
  – It bundles the session key with the user name: $\{U,K_s\}$.
– It uses its own secret encryption key $K_k$ to encrypt this.

– It bundles the session key with the created unforgeable ticket, creating \{Ks, \{U,Ks\}$K_k$\}.

– Finally, the whole thing is encrypted using the user-key that was generated from the user’s password, leading to \{Ks, \{U,Ks\}$K_k$\}$K_p$.

This is sent back to the client.
Kerberos Protocol

• The client does the following steps:
  – It prompts the user for his password $p$, immediately computes $K_p$, and erases the password.
  – Using $K_p$, the client decrypts the message it got from the server, and obtains $K_s$ and $\{U, K_s\}_{K_k}$.
  – It erases the user key $K_p$. 
Now What?

• Now, the client can send authenticated requests to the Kerberos server.

• Each request is composed of two parts:
  – The request itself, $R$, encrypted using $K_s$,
  – The unforgeable ticket.

• The server decrypts the ticket using its secret key $K_k$, and finds $U$ and $K_s$. 
But...

- An eavesdropper can copy the whole request message and retransmit it.
- The Kerberos server does not provide any real services. All it does is to provide keys for other servers.
Finally

• Kerberos will send the allocated key $K_f$ to the client encrypted by $K_s$, and also send it to the file server using $K_b$

• The client will then be able to use $K_f$ to convince the file server of its identity
  - perform operations on files
Introduction to Security

Based on Slides by Shlomo Kipnis,
Introduction to Security Course
What is Security?

• Making sure that bad things do not happen

• Reducing the chances that bad things will happen

• Lowering the impact of bad things

• Providing means to recover from bad things
Security Challenges

• Securing a variety of **different systems**
• Securing **interfaces** between different systems
• Different security **goals and needs**
• Attackers seek **weakest link** in the system
• Security people must protect **all links** in the system
• Maintaining **system usability**
• Keeping **security costs** under control
Threats & Attacks

- Unauthorized access
- Denial of service
- Computer viruses
- Trojan horses
- Information loss
- Data leaks
- Data manipulation
- Data theft
- Data destruction
- Program manipulation
Eavesdropping and Packet Sniffing

• Description: Acquiring information without changing it

• Means: Packet sniffers, routers, gateways, capturing and filtering out packets

• Threats: Sniffing can be used to catch various information sent over the network
  – Login + Password
  – Credit card numbers
  – E-mails and other messages
  – Traffic analysis
**Snooping**

- **Description:** Acquiring information without modifying it
- **Means:** Browsing documents on disk or main memory
  - Using legitimate privileges (insiders)
  - Hacking into a system (outsiders)
  - Stealing laptops
  - Monitoring keyboard strokes
  - Observing timing information (covert channels)
- **Threats:**
  - Obtaining sensitive information (files with credit card numbers)
  - Discovering passwords, secret keys, etc.
Tampering

- **Description**: Modifying or destroying stored data
- **Means**: Insiders misusing privileges or outsiders breaking into system
- **Threats**:
  - Change records – school grades, prison records, taxpayers’ debts (NY $13 million property tax fraud)
  - Erase audit trails (by hacker)
  - Plant Trojan-horses for password gaining, and other uses
Spoofing

- **Description**: Impersonating other users or computers to obtain privileges

- **Means**:
  - Account stealing, password guessing, social engineering
  - IP spoofing: E-mail forging, false IP From address, hijacking
  - IP connections

- **Threats**:
  - Forged messages ("exam is cancelled")
  - Denial of Service (IP attacks, SYN attacks, Ping-of-Death)
Jamming

- **Description**: Disabling a system or service
- **Means**: Engaging host in numerous (legitimate) activities until exhausting its resources; spoofing return addresses to avoid tracing
- **Threats**:
  - Consume all resources on the attacked machines, e.g., memory (SYN attack), disk (E-mail attack)
  - Exploit bug to shut down hosts (ping-of-death)
Code Injection

• **Description**: Injecting malicious code to execute on host with high privileges and infecting other hosts

• **Means**:  
  – Virus: attached to executable, spread through infected floppy disks, E-mail attachments, macros  
  – Worm: replicate over the Internet

• **Threats**:  
  – Everything…
Methods
Exploiting Flaws

- Exploit vulnerabilities in software to penetrate systems
  - Buffer overflow (e.g., ‘finger’, Internet Worm, Web Site apps)
  - Mobile code security flaws (Java, ActiveX)

- Knowledge spreads faster than remedy
  - Hacker bulletins
  - Advisories:
    - Flaws/fixes repositories, e.g., CERT
    - Publicly available software kits to detect known vulnerabilities, e.g., SATAN, ISS
    - But they are not always followed readily, and are often used to the advantage of hackers
      - Publicly available hacker kits on the net, e.g., RootKit (Unix)
Password and Key Cracking

• Guessing: family member names, phone numbers, etc.

• Dictionary Attack: systematic search
  – Crack: dictionary attack extended with common patterns
    • crack is now employed by sys-admins and the passwd program

• Exhaustive search:
  – Crypt-analysis tools evolve continually
  – The Internet provides a massively parallel computing resource

• Crypt-analysis, bad generators, timing analysis

• Smart-card cracking via fault injection
Social Engineering

• Spoofing a “real system”:
  – Login screen
  – Phone numbers
  – ATM story

• Spoofing a “service”:
  – Stealing credit card numbers and PINs
  – Stealing passwords

• Agent-in-the-Middle Attacks
  – Special print of newspaper
  – Router, gateway, bulletin boards, etc.
Buffer Overflow

Based On Slides by Tomer Harpaz

Advanced OS seminar
Buffer Overflows

• Common

• Stack or heap

• Overwriting control-data or sensitive data
Memory Organization

0x0000 - 0xffffffff

Process Control Block (PCB)

- Stack
- Heap
- Data
- Text

PCB:
- Pointer
- Process state
- Process number
- Program counter
- Registers
- Memory limits
- List of open files

In the kernel (not accessible directly to the user process)

User context
Memory Organization (cont.)

[Diagram showing stack organization with stack frames for DrawSquare and DrawLine subroutines, including stack pointer, frame pointer, and memory addresses for locals and parameters.]
Stack Buffer Overflow

Unallocated Stack Space

Char c[0]

Char c[11]

Char c[12]

Char *bar

Saved Frame pointer

Return Address

Parent Routine's Stack

Unallocated Stack Space

h
e
l
l

Char c[12]

Char *bar

Saved Frame pointer

Return Address

Parent Routine's Stack
Stack Buffer Overflow (cont.)

![Diagram of stack buffer overflow]

- Unallocated Stack Space
- Parent Routine’s Stack
- Little Endian 0x80C03508
- Memory Addresses
- Stack Growth
- Address 0x80C03508

- A A A A
- A A A A
- A A A A
- A A A A
- A A A A
- A A A A
- \x08 \x35 \xCD \x80
Solutions

• Os level
  – Exec shield
  – Address space layout randomization
  – Etc..

• Programmer level
  – fgets (not gets)
  – strncpy (not strcpy)
  – Etc…