Operating Systems

Security and protection

Stephan Sigg

Distributed and Ubiquitous Systems
Technische Universität Braunschweig

January 31, 2011
Overview and Structure

- Introduction to operating systems
  - History
  - Architectures
- Processes
  - Processes, Threads, IPC, Scheduling
  - Synchronisation
  - Deadlocks
- Memory management
  - Paging
  - Segmentation
- Filesystems
- Security and Protection
- Distributed systems
- Cryptography
Outline

Security and protection

1. Goals of protection
   - Principles of protection
   - Domain of protection
   - Access control

2. System security
   - The security problem
   - Program threats
   - System and network threats
   - Cryptography as a security tool
   - User authentication
   - Implementing security defenses
   - Firewalls to protect systems and networks
Protection refers to a mechanism for controlling the access of programs, processes or users to the resources.

Must provide means for specifying the controls to be imposed.

We distinguish between protection and security.

Security is a measure of confidence that the integrity of a system and its data will be preserved.
Security and protection

Goals of protection

Reasons for protection

- Prevent the mischievous, intentional violation of an access restriction
- Ensure that each program component uses system resources only in ways consistent with system policies

A protection oriented system provides means to distinguish between authorised and unauthorised usage
A key principle for protection is the principle of least privilege.

Programs, users and systems are given just enough privileges to perform their tasks.

Should enable to provide privileges when needed and disable them otherwise.

Separate account for each user.
Security and protection

Domain of protection

- A computer system is a collection of processes and objects
- Hardware objects: CPU, memory, printers, disks...
- Software objects: Files, programs, semaphores, ...
- A process should be allowed to access only those objects for which it has authorisation

<table>
<thead>
<tr>
<th></th>
<th>read</th>
<th>write</th>
<th>execute</th>
<th>print</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Disk</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Book</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Paper</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>
Security and protection
Domain of protection

Domain structure

- To facilitate this scheme, processes operate within a protection domain.
- Each domain defines a set of objects and the types of operations that may be invoked on each object.
- Domains are not necessarily disjoint.
Security and protection

Domain of protection

Domain structure
A domain can be realised in a variety of ways

- Each user may be a domain
  - Set of objects that can be accessed depends on the identity of the user

- Each process may be a domain
  - The set of objects that can be accessed depends on the identity of the process
  - Domain switching occurs when one process sends a message to another process and waits for a response

- Each procedure may be a domain
  - The set of objects that can be accessed corresponds to the local variables defined within a procedure
  - Domain switching occurs when a procedure call is made
Security and protection
Access control

- The access to objects can be restricted in a similar way as the access to files.
- For each object, access control information is added.
- Example: Role-based access control (Solaris 10)
  - Processes are assigned privileges.
  - A privilege is the right to execute a system call or to use an option within that call.
Outline

Security and protection

1. Goals of protection
   - Principles of protection
   - Domain of protection
   - Access control

2. System security
   - The security problem
   - Program threats
   - System and network threats
   - Cryptography as a security tool
   - User authentication
   - Implementing security defenses
   - Firewalls to protect systems and networks
Security and protection
System security

- Protection is strictly an internal problem
- Security, however, requires also consideration of the external environment
- A protection system is ineffective, for instance, if user authentication is compromised
Security and protection

The security problem

- Security violations can be categorised as intentional or accidental
- It is easier to protect against accidental security violations
- Protection methods mostly consider accidental security violations
- A threat is a potential for a security violation
- An attack is the attempt to break security
Security and protection
The security problem

Breach of confidentiality  Unauthorised reading of data. Goal of the intruder: Capture secret data

Breach of integrity  Unauthorised modification of data. E.g. modification of source code

Breach of availability  Unauthorised destruction of data

Theft of service  Unauthorised use of resources. E.g. intruder may install a daemon that acts as a file server

Denial of service  Preventing legitimate use of the system.
Security and protection

The security problem

Methods to breach security

Masquerading:

- One participant in a communication pretends to be someone else
- Attacker breach authentication
- Can also gain access that would not normally be allowed
Security and protection

The security problem

Methods to breach security

Replay attack:

- Replay a captured exchange of data
- Intruder repeats a valid data transmission
- This may even constitute the complete attack (e.g. repeat a request to transfer money)
- Frequently combined with message modification in order to escalate privileges
Security and protection
The security problem

Methods to breach security

Man in the middle attack:

- Intruder sits in the data flow of a communication
- masquerading as the sender to the receiver
- masquerading as the receiver to the sender
- May be combined with session hijacking
  - Active communication session is intercepted
Security and protection

The security problem

- Optimum protection of the system is not possible
- The cost for the intruder can, however, be made sufficiently high
- In order to deter most intruders
- In some cases (e.g. DoS-attacks) it is preferable to prevent the attack but sufficient to detect it to take countermeasures
Security and protection
The security problem

To protect a system, we must take security measures at four levels:

- Physical
- Human
- Operating system
- Network
Security and protection

The security problem

To protect a system, we must take security measures at four levels:

**Physical** The site containing the computer system must be physically secured against armed or surreptitious entry by intruders
Security and protection
The security problem

To protect a system, we must take security measures at four levels:

**Human** Authorisation must be done carefully to assure that only appropriate users have access to the system. Users may also be tricked into providing access rights (e.g. phishing)
Security and protection
The security problem

To protect a system, we must take security measures at four levels:

Operating system  System must protect itself from accidental or purposeful security breaches
- Runaway process could constitute an accidental denial-of-service attack
- Query to a service could reveal passwords
- Stack overflow could allow the launching of unauthorised processes
- ...

Stephan Sigg  Operating Systems
Security and protection

The security problem

To protect a system, we must take security measures at four levels:

- **Network**
  - Interception of data on network lines could reveal private data;
  - Interception of data could constitute a remote denial-of-service attack
Security and protection

The security problem

To protect a system, we must take security measures at four levels:

- Physical
- Human
- Operating system
- Network

The security of a system is as weak as its weakest point. Therefore, all aspects must be addressed for security to be maintained.
Security and protection

Program threats

- Processes, along with the kernel are the only means of accomplishing work on a computer
- Therefore, a security thread is most often posed by a program
  - Trojan horse
  - Trap door
  - Logic bomb
  - Stack and buffer overflow
  - Viruses
Security and protection

The security problem

Trojan horse:

- Many systems have mechanism for allowing programs to be executed by other users.
- A text editor, for instance, may search an opened file for specific keywords and act accordingly.
- A code segment that misuses its environment is called a Trojan horse.
- Another variation is spyware.
Security and protection
The security problem

Trap door:
- The designer of a program may leave a hole in the software that only he is capable of using.
- A trap door might also be included in the compiler so that the scan of the source code alone will not reveal the security threat.
- Trap doors pose a difficult problem since the whole source code has to be scanned in order to detect them.
Security and protection
The security problem

Logic bomb:

- A program that initiates a security incident only under certain circumstances is called a Logic bomb.
- It is hard to detect since under normal operations no security hole is apparent.
Security and protection
The security problem

Stack and buffer overflow:

- Most common way for an attacker outside the system
- Attack via dial-up or network connection
- Authorised user of the system may also use this exploit for privilege escalation
- Exploits a bug in a program
  - Might be simple case of poor programming where programmer did not use code bounds for an input field
  - The attacker sends more data than the program was expecting
Security and protection
The security problem

Stack and buffer overflow:

- An attacker might write a program to do the following:
  1. Overflow an input field, command line argument or input buffer until it writes into the stack.
  2. Overwrite the current return address on the stack with the address of the exploit code to be loaded.
  3. Place exploit code in the next space of the stack.
Security and protection

The security problem

Example

- Consider an input field on a web-form
- The attacker could send extra characters to overflow the buffer and reach the stack
- When the buffer-reading subroutine returns from execution, the return address is the exploit code
- The code runs with access rights of the attacked program
Security and protection
The security problem

Example

1: #include <stdio.h>
2: #define BUFFER_SIZE 256
3: int main(int argc, char argv[]){
4:  char buffer[BUFFER_SIZE]
5:  if argc ¡ 2 then
6:     return -1;
7:  else
8:      strcpy(buffer,argv[1]);
9:  return 0;
10: end if
11: }

Stephan Sigg Operating Systems 32/61
Security and protection

The security problem

Layout of a typical stack frame

- Return address
- saved frame pointer
- automatic variables
- parameters
Security and protection
The security problem

Layout of a modified stack frame

- Return address
- buffer(BUFFER_SIZE -1)
- buffer(1)
- buffer(0)
- Address of modified code
- NO_OP
- modified shell code

Stephan Sigg Operating Systems 34/61
Security and protection

The security problem

- Recent versions of AMD and Intel x86 chips include the NX feature to prevent this type of attack.
- The use of the feature is supported in several x86 operations systems such as Linux and Windows XP SP2.
- The hardware implementation involves the use of a new bit in the page tables in the CPU.
- The bit marks the associated page as non executable so that the instructions cannot be read from it and executed.
Security and protection

The security problem

Viruses:

- A virus is a fragment of code embedded in a legitimate program
- Viruses are designed to ‘infect’ other programs
- Viruses are typically very specific to architectures, operating systems and applications
**Security and protection**

**The security problem**

Viruses can belong to several categories

- **File**  The virus infects a system by appending itself to a file. It changes the start of the program so that execution jumps to its code. Afterwards, the control is returned to the program.

- **Boot**  The virus infects the boot sector of the system, executing every time the system is booted and before the operating system is loaded. The viruses do not appear in the file system (memory virus).

- **Macro**  Macro viruses are written in a high-level language such as Visual Basic. Triggered when a program capable of executing the macro is run. For instance, contained in a spreadsheet file.
Security and protection
The security problem

Viruses can belong to several categories

**Source code** This virus looks for source code and modifies it to include the virus and to help to spread it.

**Polymorphic** Changes each time it is installed to avoid detection by antivirus software. Changes do not affect the virus’s functionality but change its signature. The signature can be, for instance the sequence of bytes that make up the virus code.

**Encrypted** The virus is encrypted to avoid detection. Contains decryption code along with the virus. First encrypts and then executes.
Security and protection

The security problem

Viruses can belong to several categories

Stealth  Attempts to avoid detection by modifying parts of the system that could be used to detect it. For instance, it could modify the `read` system call so that the infected code of a file is skipped.

Tunneling  Attempts to bypass detection by installing itself in the interrupt-handler chain or, for instance, in device drivers.

Multipartite  Virus is able to infect multiple parts of a system (boot sectors, memory, files. This makes it more difficult to remove the virus.

Armored  Coded to make it hard for antivirus researchers to unravel and understand it.
Security and protection
System and network threats

- System and network threats involve the abuse of services and network connections
- The services are available over a network, the more likely it is that a bug is available to exploit
- Increasingly, operating system are secure by default
- This means that many services (e.g. FTP, telnet, ssh) are disabled by default
- They have to be specifically enabled by the system administrator
Security and protection
System and network threats

- Worms
- Port scanning
- Denial of service
Security and protection
System and network threats

Worms

- A worm is a process that uses the spawn mechanism to ravage system performance
- The worm spawns copies of itself, using up system resources
Security and protection
System and network threats

Diagram:
- Target system
- Infected system
- Rsh attack
- Finger attack
- Sendmail attack
- Request for worm
- Worm sent

Security and protection
System and network threats

Port scanning

- Port scanning is a means for an intruder to detect a system's vulnerabilities to attack.
- If a vulnerability for a specific system running typically on a specific port is known, the attacker could try to connect to the specific port and test if the program in question is actually available on this port.
- Since port scans are detectable, they are frequently launched from Zombie systems.
Security and protection
System and network threats

Denial of service

- Denial of service attacks aim at disrupting legitimate use of a system or facility.
- This is frequently easier than breaking into a machine or facility.
- Example:
  - If many TCP connections are requested by a remote host but the opened session is never acknowledged, a server system may end up with a multiple of open sessions.
  - These might eat up all the network resources of the system and disable any further legitimate TCP connection.
Security and protection

System and network threats

Denial of service

- These attacks are typically stopped at the network level to update the operating system and reduce their vulnerability.
- Since the attacks use the same mechanisms as normal operation, it is impossible to prevent them.
Security and protection
Cryptography as a security tool

- Cryptography can be utilised as a tool for computer security
- In a networked system, an operating system can never be absolutely sure about the identity of its communication partner.
- Cryptography can help here to remove the necessity to trust the network
- Constraints the number of potential senders/receivers of a message
- Typically based on keys that are selectively distributed to computers in a network
Security and protection
Cryptography as a security tool

Encryption

- Encryption constrains the possible receivers of a message

key exchange
plaintext
message
attacker
secure channel
insecure channel
encryption channel
write
read

Encryption algorithm E
plaintext
message

Decryption algorithm D
plaintext
message
m=D(k)(m)
c=E(k)(m)

Stephan Sigg Operating Systems 48/61
Security and protection
Cryptography as a security tool

Encryption

Symmetric encryption:

- The same key is used to encrypt and to decrypt
- The secrecy of $E(k)$ must be protected as well as $D(k)$
- If the same key is utilised for an extended amount of data, it becomes vulnerable to an attack
- Examples: AES (advanced encryption standard), DES (data-encryption standard)
Encryption

Asymmetric encryption:

- Different encryption and decryption keys are used
- We distinguish between public and private keys
- Uses one-way function for which the inverse operation is much harder to execute (e.g. factorisation)
Security and protection
Cryptography as a security tool

Encryption
Asymmetric encryption:

Encryption algorithm $E$
Decryption algorithm $D$

plaintext $\rightarrow$ ciphertext $\rightarrow$ plaintext

access public key

message

plaintext
insecure channel

write
encryption key (public)

read
decryption key (private)

$E(k)(m)$
$m = D(k)(m)$
Security and protection
Cryptography as a security tool

Encryption
Asymmetric encryption:
- Exkurs: RSA Cryptosystems
Security and protection
Cryptography as a security tool

Authentication

- Encryption constrains the amount of potential senders of a message
- Constraining the amount of possible receivers of a message is called authentication
- Authentication is complementary to encryption
- Observe, that an encrypted message can also prove the identity of the sender.
- If $D(k_e, n)(E(k_e, N)(m))$ produces a valid
- Authentication is also useful for proving that a message has not been modified.
Security and protection
Cryptography as a security tool

Key distribution

- With symmetric cryptography, the transmission of the key becomes a great challenge and possible vulnerability

Out-of-band transmission Paper document, conversation, untypical transmission technique/band

- Since this might be insecure and is possible also very elaborate, public key cryptosystems are utilised

- Since public keys need not be secured, a key distribution center can be utilised.

- Vulnerable to man-in-the-middle-attacks

- With trusted authorities the authentication of individual communication partners can then be verified
Security and protection

User authentication

- The system has also to authenticate the user
- Typically based on one or more of the following
  - The possession of something (key or card)
  - Knowledge of something (identifier and password)
  - Attribute of the user (fingerprint, retina pattern, signature)
Security and protection

User authentication

Issues with passwords

- How to keep the password stored and save in the system.
- In UNIX systems, encryption is used to secure the password.
- However, with brute force attacks, these passwords may be decoded in acceptable time off-line.
- Therefore, the file containing the passwords is often visible to a superuser only.
- Programs that use the password run setuid to root in order to be allowed to read the file.
- Typically based on one or more of the following:
  - The possession of something (key or card)
  - Knowledge of something (identifier and password)
  - Attribute of the user (fingerprint, retina pattern, signature)
Security and protection
Implementing security defenses

- Several layers of security are better than one layer
- Penetration tests to test security
  - Password security
  - Unauthorised privileged programs
  - Unexpectedly long running processes
- Intrusion detection
  - Signature based detection
  - Anomaly detection
  - Virus protection
Security and protection
Firewalls to protect systems and networks

- With a firewall a computer can be connected to an untrustworthy network
- A firewall is a computer, appliance or router between the system and the network
- The firewall limits the network between these two security domains and monitors and logs all connections
- Firewall may disallow ports, IP addresses, or communication directions (in/out)
Security and protection

Summary

- Protection
- Security
  - Threats
  - Cryptography as security tool
  - Firewalls
Security and protection
Questions, discussion, remarks

Questions?
Literature

Recommended literature