# OPERATING SYSTEMS

## Security

**Suresh Jamadagni**
Department of Computer Science

# OPERATING SYSTEMS

## Slides Credits for all PPTs of this course

- The slides/diagrams in this course are an **adaptation**, **combination**, and **enhancement** of material from the following resources and persons:

1. Slides of Operating System Concepts, Abraham Silberschatz, Peter Baer Galvin, Greg Gagne - 9th edition 2013 and some slides from 10th edition 2018
2. Some conceptual text and diagram from Operating Systems - Internals and Design Principles, William Stallings, 9th edition 2018
3. Some presentation transcripts from A. Frank – P. Weisberg
4. Some conceptual text from Operating Systems: Three Easy Pieces, Remzi Arpaci-Dusseau, Andrea Arpaci Dusseau

# OPERATING SYSTEMS

## Security

- **Security**, requires not only an adequate protection system but also consideration of the ***external*** environment within which the system operates.
- Computer resources must be guarded against unauthorized access, malicious destruction or alteration, and accidental introduction of inconsistency.
- These resources include information stored in the system (both data and code), as well as the CPU, memory, disks, tapes, and networking that are the computer.
- In many applications, ensuring the security of the computer system is worth considerable effort
- We say that a system is **secure** if its resources are used and accessed as intended under all circumstances

## Security

- Security violations (or misuse) of the system can be categorized as intentional (malicious) or accidental.
- It is easier to protect against accidental misuse than against malicious misuse
- A **threat** is the potential for a security violation, such as the discovery of a vulnerability, whereas an **attack** is the attempt to break security.

## Security Threats

- **Breach of confidentiality**. This type of violation involves unauthorized reading of data (or theft of information). Typically, a breach of confidentiality is the goal of an intruder
- **Breach of integrity**. This violation involves unauthorized modification of data.
- **Breach of availability**. This violation involves unauthorized destruction of data.
- **Theft of service**. This violation involves unauthorized use of resources.
- **Denial of service**. This violation involves preventing legitimate use of the system.

# OPERATING SYSTEMS

## Security Threats

- Attackers use several standard methods in their attempts to breach security. The most common is **masquerading**, in which one participant in a communication pretends to be someone else (another host or another person).
- By masquerading, attackers breach **authentication**, the correctness of identification; they can then gain access that they would not normally be allowed or escalate their privileges—obtain privileges to which they would not normally be entitled.
- A **replay attack** consists of the malicious or fraudulent repeat of a valid data transmission. Sometimes the replay comprises the entire attack. But frequently it is done along with **message modification**, again to escalate privileges

## Security Threats

- **man-in-the-middle attack**, in which an attacker sits in the data flow of a communication, masquerading as the sender to the receiver, and vice versa.
- In a network communication, a man-in-the-middle attack may be preceded by a **session hijacking**, in which an active communication session is intercepted.
- Absolute protection of the system from malicious abuse is not possible, but the cost to the perpetrator can be made sufficiently high to deter most intruders

# OPERATING SYSTEMS

## Security measures

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

1. **Physical** - The site or sites containing the computer systems must be physically secured against armed or surreptitious entry by intruders
2. **Human** - Authorization must be done carefully to assure that only appropriate users have access to the system
3. **Operating system** - The system must protect itself from accidental or purposeful security breaches.
4. **Network** - Much computer data in modern systems travels over private leased lines, shared lines like the Internet, wireless connections, or dial-up lines. Intercepting these data could be just as harmful as breaking into a computer, and interruption of communications could result in diminishing users' trust in the system

# Program Threats

- Processes, along with the kernel, are the only means of accomplishing work on a computer.
- Therefore, writing a program that creates a breach of security, or causing a normal process to change its behavior and create a breach, is a common goal of attackers

**Trojan Horse**

- A code segment that misuses its environment is called a **Trojan horse**
- Many systems have mechanisms for allowing programs written by users to be executed by other users. If these programs are executed in a domain that provides the access rights of the executing user, the other users may misuse these rights

# Program Threats

**Trojan Horse**

- A variation of the Trojan horse is a program that emulates a login program. An unsuspecting user starts to log in at a terminal and notices that he has apparently mistyped his password. He tries again and is successful. What has happened is that his authentication key and password have been stolen by the login emulator, which was left running on the terminal by the thief.
- Another variation on the Trojan horse is **spyware**. Spyware sometimes accompanies a program that the user has chosen to install. Most frequently, it comes along with freeware or shareware programs, but sometimes it is included with commercial software.
- The goal of spyware is to or capture information from the user's system and return it to a central Site
- Spyware is a micro example of a macro problem: violation of the principle of least privilege

# Program Threats

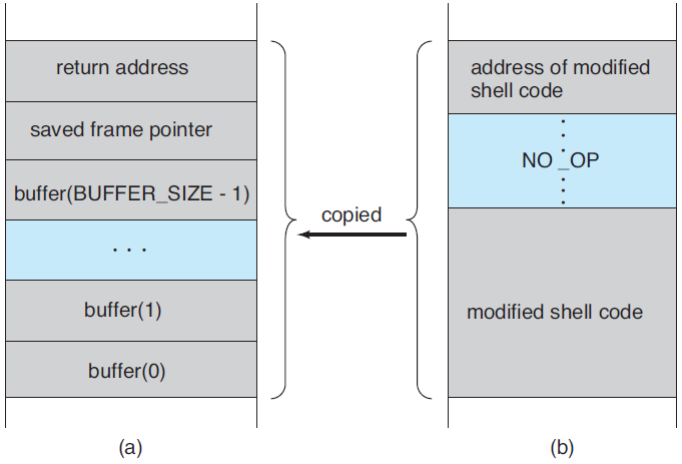
**Trap Door**

- The designer of a program or system might leave a hole in the software that only he/she is capable of using
- For instance, the code might check for a specific user ID or password, and it might circumvent normal security procedures.
- A clever trap door could be included in a compiler. The compiler could generate standard object code as well as a trap door, regardless of the source code being compiled
- Trap doors pose a difficult problem because, to detect them, we have to analyze all the source code for all components of a system

# Program Threats

## Stack and Buffer overflow

- The stack- or buffer-overflow attack is the most common way for an attacker outside the system, on a network or dial-up connection, to gain unauthorized access to the target system.
- The attack exploits a bug in a program. The bug can be a simple case of poor programming, in which the programmer neglected to code bounds checking on an input field. In this case, the attacker sends more data than the program was expecting.




Hypothetical stack frame for (a) before and (b) after.

# OPERATING SYSTEMS

## Program Threats


- **Viruses**
- A virus is a fragment of code embedded in a legitimate program.
- Viruses are self-replicating and are designed to “infect” other programs.
- They can wreak havoc in a system by modifying or destroying files and causing system crashes and program malfunctions.
- As with most penetration attacks, viruses are very specific to architectures, operating systems, and applications.
- Once a virus reaches a target machine, a program known as a **virus dropper** inserts the virus into the system.
- The virus dropper is usually a Trojan horse, executed for other reasons but installing the virus as its core activity. Once installed, the virus may do any one of a number of things.

# OPERATING SYSTEMS

## System Threats

- System and network threats involve the abuse of services and network connections.
- System and network threats create a situation in which operating-system resources and user files are misused.
- Sometimes, a system and network attack is used to launch a program attack, and vice versa.
- The more ***open*** an operating system is—the more services it has enabled and the more functions it allows—the more likely it is that a bug is available to exploit

**Worms**

- A **worm** is a process that uses the **spawn** mechanism to duplicate itself.
- The worm spawns copies of itself, using up system resources and perhaps locking out all other processes

# OPERATING SYSTEMS

## Defenses against system attacks

- The broadest tool available to system designers and users to thwart attacks is **cryptography**.
- In an isolated computer, the operating system can reliably determine the sender and recipient of all inter-process communication, since it controls all communication channels in the computer.
- In a network of computers, the situation is quite different. A networked computer receives bits “from the wire” with no immediate and reliable way of determining what machine or application sent those bits.
- **Cryptography** is used to constrain the potential senders and/or receivers of a message.
- Modern cryptography is based on secrets called **keys** that are selectively distributed to computers in a network and used to process messages.
- Cryptography enables a recipient of a message to verify that the message was created by some computer possessing a certain key. Similarly, a sender can encode its message so that only a computer with a certain key can decode the message

# OPERATING SYSTEMS

## Defenses against system attacks

- The broadest tool available to system designers and users to thwart attacks is **cryptography**.
- In an isolated computer, the operating system can reliably determine the sender and recipient of all inter-process communication, since it controls all communication channels in the computer.
- In a network of computers, the situation is quite different. A networked computer receives bits “from the wire” with no immediate and reliable way of determining what machine or application sent those bits.
- **Cryptography** is used to constrain the potential senders and/or receivers of a message.
- Modern cryptography is based on secrets called **keys** that are selectively distributed to computers in a network and used to process messages.
- Cryptography enables a recipient of a message to verify that the message was created by some computer possessing a certain key. Similarly, a sender can encode its message so that only a computer with a certain key can decode the message

## Defenses against system attacks

- Constraining the set of potential senders of a message is called **authentication**.
- Authentication is thus complementary to encryption
- Authentication is also useful for proving that a message has not been modified

**SSL 3.0**

- SSL 3.0 is a cryptographic protocol that enables two computers to communicate securely—that is, so that each can limit the sender and receiver of messages to the other.
- It is perhaps the most commonly used cryptographic protocol on the Internet today, since it is the standard protocol by which web browsers communicate securely with web servers.
- The SSL protocol is initiated by a client *c* to communicate securely with a server.
- Asymmetric cryptography is used so that a client and a server can establish a secure **session key** that can be used for symmetric encryption of the session between the two

# Defenses against system attacks


**User authentication**

- A major security problem for operating systems is **user authentication**.
- The protection system depends on the ability to identify the programs and processes currently executing, which in turn depends on the ability to identify each user of the system
- The most common approach to authenticating a user identity is the use of **passwords**.

**Securing passwords**

- The UNIX system uses secure hashing to avoid the necessity of keeping its password list secret.
- Because the list is hashed rather than encrypted, it is impossible for the system to decrypt the stored value and determine the original password
- Some systems do not allow the use of dictionary words as passwords
- Multi-factor authentication requires users to provide more than one piece of information to authenticate successfully to an account or Linux host. The additional information may be a one-time password (OTP)

# OPERATING SYSTEMS

## Defenses against system attacks

**ssh**

- SSH, or Secure Shell, constitutes a cryptographic network protocol designed to enable secure communication between two systems over networks that may not be secure.
- This protocol is widely employed for remote access to servers and the secure transmission of files between computers. In essence, SSH acts as a secure conduit, establishing a confidential channel for communication in scenarios where the network may pose security risks
- This technology is instrumental for professionals seeking a reliable and secure method of managing servers and transferring sensitive data across computers in a controlled and protected manner.
- ssh runs at TCP/IP port 22.
- https://www.geeksforgeeks.org/ssh-command-in-linux-with-examples/

# Implementing Security Defenses


**Security policy**

- The first step toward improving the security of any aspect of computing is to have a **security policy**.
- Policies vary widely but generally include a statement of what is being secured
- Without a policy in place, it is impossible for users and administrators to know what is permissible, what is required, and what is not allowed

**Vulnerability Assessment**

- The core activity of most vulnerability assessments is a **penetration test**, in which the entity is scanned for known vulnerabilities.
- Vulnerability scans typically focus on short or easy-to-guess passwords, unauthorized privileged programs such as setuid programs, unauthorized programs in system directories, Improper directory protections on user and system directories and changes to system programs detected with checksum values

# Implementing Security Defenses

## Intrusion detection

- **Intrusion detection**, as its name suggests, strives to detect attempted or successful intrusions into computer systems and to initiate appropriate responses to the intrusions
- Intrusion Detection (IDS) systems raise an alarm when an intrusion is detected, while Intrusion Prevention (IDP) systems act as routers, passing traffic unless an intrusion is detected at which point that traffic is blocked
- **Anomaly detection**, attempts through various techniques to detect anomalous behavior within computer systems. Of course, not all anomalous system activity indicates an intrusion, but the presumption is that intrusions often induce anomalous behavior
- An example of an anomaly-detection tool is the **Tripwire file system** integrity checking tool for UNIX, developed at Purdue University. Tripwire operates on the premise that many intrusions result in modification of system directories and files.

# Implementing Security Defenses


## Virus protection

- The best protection against computer viruses is prevention, or the practice of **safe computing**.
- Antivirus programs are often used to provide this protection.
- Some of these programs are effective against only particular known viruses. They work by searching all the programs on a system for the specific pattern of instructions known to make up the virus.
- When they find a known pattern, they remove the instructions, **disinfecting** the program.
- Antivirus programs may have catalogs of thousands of viruses for which they search

# Implementing Security Defenses

## Auditing, Accounting, and Logging

- Auditing, accounting, and logging can decrease system performance, but they are useful in several areas, including security.
- Logging can be general or specific. All system-call executions can be logged for analysis of program behavior (or misbehavior). More typically, suspicious events are logged.
- Authentication failures and authorization failures can tell us quite a lot about break-in attempts.
- Accounting can be used to find performance changes, which in turn can reveal security problems.

# Implementing Security Defenses

## Firewalling to Protect Systems and Networks

- A **firewall** is a computer, appliance, or router that sits between the trusted and the untrusted.
- A network firewall limits network access between the two **security domains** and monitors and logs all connections.
- It can also limit connections based on source or destination address, source or destination port, or direction of the connection.
- For instance, web servers use HTTP to communicate with web browsers. A firewall therefore may allow only HTTP to pass from all hosts outside the firewall to the web server within the firewall.
- Of course, a firewall itself must be secure and attack-proof. Otherwise, its ability to secure connections can be compromised
- **System-call firewalls** sit between applications and the kernel, monitoring system-call execution.


# THANK YOU

**Suresh Jamadagni**
Department of Computer Science Engineering

**sureshjamadagni@pes.edu**