Question & Answers

SECURITY

Sercan Külcü | Operating Systems | 10.04.2023

Contents

Why is security a critical concern in modern computing environments?

Security is crucial in modern computing due to the widespread interconnectivity of systems across diverse networks. This increased accessibility exposes computers to a variety of security risks. Common threats include malware such as viruses, worms, and Trojans, as well as ransomware, phishing attacks, and social engineering tactics. As networks expand and become more complex, safeguarding data and systems against these vulnerabilities becomes essential.

How does OS play a central role in providing security mechanisms and policies?

The operating system (OS) serves as a key component in managing both hardware and software resources, enabling programs to run efficiently. It plays a pivotal role in security by controlling access to system resources, ensuring that only authorized users or applications can interact with critical data. The OS enforces security policies, protecting the system from unauthorized access and malicious activities. It also integrates security features such as firewalls, antivirus programs, and encryption to safeguard data integrity and confidentiality. Through these mechanisms, the OS helps maintain a secure computing environment.

What is authentication?

Authentication is the process of confirming the identity of a user or device before granting access to system resources or data. It ensures that only authorized entities can interact with sensitive information or services. This process prevents unauthorized access and acts as a

fundamental layer of security in any system, safeguarding both data and resources from potential threats. Authentication typically involves methods such as passwords, biometrics, or tokens to validate identity.

What is access control?

Access control is the mechanism that regulates who or what can access system resources based on identity and permissions. It ensures that only users or devices with appropriate privileges can interact with sensitive data or services. By enforcing predefined security policies, access control limits unauthorized access and helps protect systems from misuse or breaches. This process is often implemented through methods like role-based access control (RBAC) or discretionary access control (DAC).

What is encryption?

Encryption is the technique of transforming data into an unreadable format using algorithms to safeguard it from unauthorized access. It ensures that only users with the correct decryption key can access or alter the original data. Encryption plays a vital role in securing data during storage and transmission, especially over insecure networks like the Internet, preventing interception or tampering by unauthorized parties.

What is a firewall?

A firewall is a security device that filters network traffic based on defined rules to protect a system or network from unauthorized access. It sits between trusted internal networks and untrusted external networks, like the Internet, serving as a barrier against potential threats. Firewalls can be hardware-based, software-based, or a hybrid of both. By

monitoring and controlling data flow, firewalls prevent malicious traffic, such as viruses, worms, and malware, from infiltrating the network and compromising resources.

What is a virtual private network (VPN)?

A virtual private network (VPN) is a secure connection that enables remote users to access a private network through the Internet. It protects data from eavesdropping and unauthorized access by encrypting traffic between the user and the network. VPNs utilize various protocols, such as OpenVPN, IPsec, and SSL/TLS, to establish encrypted communication channels, ensuring privacy and security during data transmission over potentially unsafe networks.

What is a rootkit?

A rootkit is a form of malware that conceals its presence on a system while providing an attacker with elevated access. It can be installed by exploiting software vulnerabilities or through social engineering tactics. Once active, a rootkit enables an attacker to remotely control the system without detection, often facilitating data theft, launching further attacks, or maintaining long-term access. Rootkits are particularly dangerous due to their ability to remain hidden and operate unnoticed.

What is a buffer overflow attack?

A buffer overflow attack occurs when a program writes more data to a buffer than it can accommodate, causing the excess to spill into adjacent memory. This vulnerability can be exploited by an attacker to execute arbitrary code or gain unauthorized access to system resources. To prevent buffer overflow attacks, proper input validation, bounds

checking, and security features like data execution prevention (DEP) are essential. These measures ensure that the program only processes data within the allocated buffer size.

What is a denial-of-service (DoS) attack?

A denial-of-service (DoS) attack aims to disrupt the normal operation of a system or network by overwhelming it with excessive traffic or requests. This prevents legitimate users from accessing the service, causing a loss of availability. DoS attacks can be carried out through methods like traffic flooding, exploiting system vulnerabilities, or using a botnet to coordinate the assault. Defenses against DoS attacks include techniques like rate limiting, traffic filtering, and deploying specialized hardware or services to absorb and mitigate the impact.

What are emerging trends and technologies in computer security?

Artificial intelligence (AI) and machine learning (ML) for detecting and responding to threats

Blockchain technology for secure data storage and transmission

Quantum computing for advancing cryptography

Hardware-based security, such as Trusted Platform Modules (TPMs) and Secure Enclaves

Cloud-based security services and solutions

How do operating systems protect against attacks that exploit hardware vulnerabilities, such as Meltdown and Spectre?

Operating systems protect against hardware vulnerabilities like Meltdown and Spectre by using software-based mitigations. Techniques such as kernel page-table isolation, Retpoline, and indirect branch prediction barriers prevent malicious code from accessing sensitive kernel memory. These measures also address speculative execution flaws in modern processors, blocking potential exploits that rely on speculative execution to leak data.

What is a sandbox?

A sandbox is a security technique that isolates untrusted or unknown software in a controlled environment to prevent it from accessing critical system resources. It limits the software's permissions, restricting access to only the necessary resources for execution. Sandboxing can be implemented at the application level, where each program runs in its own sandbox, or at the system level, where multiple applications are confined within a shared sandbox. This approach helps protect against potential malicious behavior or data breaches.

What is a security policy?

A security policy is a defined set of rules and procedures that govern the use of systems and networks within an organization. It establishes guidelines for acceptable use, access control, data protection, and other security practices to align with the organization's goals. Security policies also ensure compliance with legal, regulatory, and industry standards,

helping protect organizational assets and maintain a secure environment.

What is a security audit?

A security audit is a comprehensive assessment of a system or network's security. It involves reviewing security policies, procedures, and controls, testing configurations, and identifying potential vulnerabilities and attack vectors. Security audits can be conducted internally by an organization's security team or externally by third-party experts. The findings help prioritize security enhancements, guide investments, and ensure continued adherence to security standards and regulations.

What are the mechanisms and policies provided by operating systems for system security?

Operating systems implement several mechanisms and policies to ensure security, including authentication, access control, encryption, firewalls, and security audits. Authentication verifies user identity, while access control enforces restrictions based on user privileges. Encryption protects data from unauthorized access or alteration, and firewalls block unwanted network traffic. Security audits assess system vulnerabilities and compliance with organizational standards, helping to identify risks and ensure proper usage. These measures work together to maintain a secure environment for system resources and data.

How does an operating system protect system resources, data, and applications from unauthorized access and modification?

An operating system safeguards system resources, data, and applications from unauthorized access and modification using various security mechanisms:

Authentication: Verifies the identity of users and processes before granting access to system resources.

Access Control: Enforces policies to define who can access which resources and the actions they can perform.

Encryption: Protects data by encoding it, ensuring only authorized users can decrypt and access it.

Firewalls: Control network traffic, blocking unauthorized access to system resources.

Sandboxing: Isolates untrusted applications in a controlled environment to prevent access to sensitive resources.

Auditing: Tracks and logs system activity to detect suspicious behavior and ensure compliance with security policies.

Privilege Escalation Prevention: Limits user rights and prevents processes from gaining unauthorized access to higher-level privileges.

Data Integrity Checks: Uses techniques like checksums or hash functions to ensure data is not tampered with.

Patch Management: Regular updates and patches are applied to fix vulnerabilities, minimizing the risk of exploitation.

Secure Boot: Ensures that only trusted software can run during the system's startup process, preventing malicious code from being loaded early.

Least Privilege Principle: Restricts user and application permissions to the minimum necessary for functionality, reducing the potential attack surface.

Virtualization: Isolates applications or entire environments in virtual machines or containers, preventing malicious activity in one environment from affecting others.

Intrusion Detection and Prevention Systems (IDPS): Monitors system activity for abnormal behavior and attempts to block or alert on potential attacks.

Security-Enhanced (SE) Modules: Adds extra security layers, such as SELinux or AppArmor, enforcing mandatory access control policies beyond standard discretionary control.

Multi-factor Authentication (MFA): Requires more than one form of verification (e.g., password and fingerprint) to access sensitive system resources, adding an additional layer of security.

Memory Protection: Prevents processes from accessing memory that does not belong to them, reducing the risk of exploits like buffer overflow attacks.

Data Execution Prevention (DEP): Restricts the execution of code in certain areas of memory, such as the stack, to prevent attacks from executing injected malicious code.

Address Space Layout Randomization (ASLR): Randomizes the memory locations used by system and application processes, making it harder for attackers to predict memory addresses for exploit attempts.

System Call Filtering: Limits the system calls a process can make, minimizing the potential for exploitation by narrowing the available attack vectors.

Role-Based Access Control (RBAC): Assigns roles to users and processes, ensuring that only authorized actions can be performed based on predefined permissions tied to those roles.

File Integrity Monitoring: Continuously checks critical system files for changes or tampering, alerting administrators if modifications occur outside of normal operations.

Secure Shell (SSH): Provides encrypted remote access to systems, preventing unauthorized access and ensuring secure communication over potentially untrusted networks.

Containerization: Isolates applications in containers, providing an additional layer of security by keeping them separate from the host system and other applications.

Trusted Execution Environments (TEEs): Offers a secure area within the processor for running sensitive operations and protecting data even from higher-level system access.

How does an operating system implement process isolation?

Process isolation is achieved by ensuring that each running process is allocated its own separate memory space and is unable to directly access the memory or resources of other processes. This is accomplished using hardware features such as memory management units (MMUs) and virtual memory. The OS uses paging or segmentation to enforce this separation. Process isolation is critical for security because it prevents one process from interfering with or exploiting another, ensuring that sensitive data remains protected and limiting the scope of potential attacks.

What is the significance of encrypted file systems in securing data?

Encrypted file systems are crucial for protecting data at rest by ensuring that stored files are rendered unreadable without the proper decryption keys. These systems use encryption algorithms (e.g., AES) to encrypt data before it is written to disk, and the data is decrypted only when accessed by an authorized user or process. By using full-disk encryption or file-level encryption, sensitive information remains protected even if the physical storage is compromised (e.g., stolen hard drives). This prevents unauthorized access to the data, especially in the event of theft, ensuring that even if an attacker gains physical access to the storage device, they cannot read the encrypted contents without the decryption key.