Exame de Segurança de Sistemas e Dados (SSD) - Answers

Época Normal – 2023/2024

Question 1 - Answer

In the context of Multilevel Security (MLS), "high water line" and "low water line" refer to security level management policies for processes handling data at different classification levels. The high water line policy states that when a process reads data at a higher security level, the process's security level is permanently raised to that higher level and cannot be lowered again during its execution, preventing any subsequent writes to lower security levels that could cause information leakage. Conversely, the low water line policy maintains that when a process writes to a lower security level, its clearance level is permanently lowered to that level, preventing it from reading higher classified information afterward. These policies are designed to prevent covert channels and ensure that information cannot flow from high security levels to low security levels through process manipulation.

Question 2 - Answer

Chord is a distributed hash table (DHT) protocol that organizes nodes in a circular ring structure where each node is assigned a unique identifier using consistent hashing, typically with SHA-1, creating a 160-bit identifier space. Unlike Kademlia, which uses XOR distance metrics and a binary tree structure with k-buckets for routing, Chord uses a simple ring topology where each node maintains a finger table with logarithmic entries pointing to nodes at exponentially increasing distances around the ring. Chord's internal processes include node joining (where new nodes find their position and update finger tables), stabilization (periodic maintenance of successor pointers), and lookup operations that route queries through the finger table in O(log N) hops. While Kademlia focuses on network proximity and parallel lookups with its XOR metric providing symmetric routing, Chord emphasizes simplicity and provable correctness with its ring-based approach, though this can make it less resilient to churn and network partitions compared to Kademlia's more robust routing structure.

Question 3 - Answer

SS7 (Signaling System 7) architecture faces several critical vulnerabilities that significantly impact Two-Factor Authentication (2FA) security. Four major attacks include: location tracking attacks where adversaries can pinpoint a subscriber's exact location by exploiting the MAP (Mobile Application Part) protocol; SMS interception attacks that allow attackers to redirect SMS messages containing 2FA codes to their own devices; denial of service attacks that can overwhelm network elements and disrupt service availability; and subscriber data harvesting where attackers can access sensitive subscriber information including IMSI numbers and authentication vectors. The global impact on 2FA usage is severe because these attacks fundamentally undermine the "something you have" factor in authentication - if an attacker can intercept SMS-based 2FA codes through SS7 vulnerabilities, the second authentication factor becomes compromised, making 2FA systems that rely on SMS essentially ineffective against sophisticated adversaries with SS7 access, leading to the industry's shift toward app-based authenticators and hardware tokens that don't rely on the cellular network infrastructure.

Question 4 - Answer

FIDO (Fast Identity Online) provides significant advantages in federated authentication systems by eliminating shared secrets and reducing the attack surface through public key cryptography and local biometric verification. In a typical federated scenario, the architecture involves four main entities: the user with a FIDO-enabled device (containing a secure authenticator), the Relying Party (RP) or service provider, the FIDO server that handles the cryptographic protocols, and the Identity Provider (IdP) in the federation. The process works by having the user's device generate a unique key pair for each service during registration, with the private key stored securely on the device (often in a Trusted Platform Module) and the public key registered with the service through the FIDO server. During authentication, the user provides a local biometric or PIN to unlock the authenticator, which then signs a challenge from the service using the private key, providing strong authentication without transmitting any shared secrets over the network. This approach offers advantages including resistance to phishing attacks (due to origin binding), elimination of password database breaches at the RP, reduced user friction through biometric authentication, and enhanced privacy since each service receives a unique key pair that cannot be correlated across services.

Question 5 - Answer

The "Confused Deputy" problem, described in Norm Hardy's classic paper, occurs when a more privileged program (the deputy) is tricked by a less privileged user (the principal) into performing actions that the user couldn't directly perform themselves, essentially abusing the deputy's elevated privileges to bypass access controls. The classic example involves a compiler that has write access to both user directories and system billing files - a malicious user could specify a system file as the output target, causing the compiler to overwrite critical system files with compiled code, since the compiler runs with elevated privileges necessary for its normal operation. The fundamental issue is that the deputy cannot distinguish between actions it should perform on behalf of the user versus actions it should perform using its own authority. The primary solution is the implementation of capability-based security systems where instead of running with ambient authority, programs receive explicit capabilities (unforgeable tokens) that grant specific permissions for specific resources, ensuring that the deputy can only act on resources that the principal has explicitly authorized, thus preventing the confused deputy from being tricked into misusing its privileges for unauthorized access.

Question 6 - Answer

K-anonymity is a privacy model designed to protect individual privacy in datasets by ensuring that each record is indistinguishable from at least k-1 other records with respect to quasi-identifiers, making it impossible to identify any individual with probability greater than 1/k. Quasi-identifiers are attributes that, while not directly identifying individuals like Social Security numbers, can be combined with external data sources to potentially re-identify individuals - examples include age, gender, ZIP code, and occupation which when combined might uniquely identify someone in a population. However, k-anonymity faces three major attacks: homogeneity attacks occur when all records in a k-anonymous group share the same sensitive attribute value, allowing an attacker to infer sensitive information even without exact identification; background knowledge attacks happen when attackers possess external information that can be used to narrow down possibilities within an anonymized group, potentially leading to identification; and composition attacks involve correlating multiple k-anonymous releases of the same dataset over time, where changes between releases can reveal information about individuals, particularly when records are added, removed, or modified between publications, ultimately compromising the anonymity guarantees that k-anonymity was designed to provide.

Question 7 - Answer

Stuxnet was a sophisticated cyberweapon specifically designed to sabotage Iran's nuclear enrichment program by targeting industrial control systems, particularly the Siemens SCADA systems controlling uranium enrichment centrifuges at the Natanz facility. The malware employed multiple attack vectors including infection through USB drives containing weaponized .lnk files that exploited zero-day vulnerabilities, network propagation using SMB exploits and print spooler vulnerabilities, and privilege escalation through stolen digital certificates from legitimate companies like Realtek and JMicron. Once inside the target network, Stuxnet would specifically search for Siemens Step7 software and Profibus/Profinet industrial networks, then inject malicious code into Programmable Logic Controllers (PLCs) that controlled the centrifuge operations. The physical mechanism involved manipulating the frequency converters that control centrifuge rotor speeds - Stuxnet would periodically alter the rotor frequencies to speeds outside normal operational parameters while simultaneously feeding false normal readings back to operators' monitoring systems, causing physical damage to the delicate centrifuge equipment through mechanical stress and vibration, ultimately setting back Iran's nuclear program by destroying hundreds of centrifuges while remaining undetected for an extended period.

Question 8 - Answer

A rootkit is malicious software designed to maintain persistent, unauthorized access to a computer system while hiding its presence from users and security software by operating at a low system level and manipulating system functions to conceal its activities. Rootkits are classified into three main categories: User-mode rootkits operate at the application level and modify system APIs and user-space functions, such as the Sony BMG rootkit that modified Windows API calls to hide CD copy protection software; Kernel-mode rootkits operate with the highest system privileges by loading malicious drivers or modifying kernel structures, like the Rustock botnet rootkit that installed itself as a kernel driver to hide network communications and maintain persistence; and Hardware/Firmware rootkits operate below the operating system level by infecting system firmware, BIOS, or hardware components, such as the theoretical proof-of-concept rootkits that modify hard drive firmware or the more recent UEFI rootkits that infect the Unified Extensible Firmware Interface. Each category represents an escalation in sophistication and difficulty of detection, with hardware rootkits being the most dangerous as they can survive operating system reinstallation and are extremely difficult to detect using traditional antivirus software that operates at higher system levels.

Question 9 - Answer

A program cannot be completely obfuscated due to fundamental theoretical limitations, primarily because any obfuscation scheme that preserves program functionality must maintain some observable behavior that can be analyzed by an adversary with sufficient computational resources and time. The impossibility of perfect obfuscation was formally proven through results showing that certain functions, called "unobfuscatable functions," cannot be obfuscated without revealing information about their internal secrets - for example, any program that contains embedded cryptographic keys or passwords must eventually use those secrets in a way that makes them observable during execution or analysis. Additionally, code obfuscation faces the same fundamental challenges as cryptography: if the obfuscated program can be executed by legitimate users, then a determined attacker with access to the same execution environment can potentially reverse-engineer the obfuscation through dynamic analysis, timing attacks, power analysis, or other side-channel methods. While practical obfuscation techniques can significantly increase the time and resources required for reverse engineering, making attacks economically infeasible for many scenarios, they cannot provide absolute security guarantees, and advances in automated analysis tools, machine learning-based deobfuscation, and increased computational power continue to erode the effectiveness of obfuscation techniques over time.

Question 10 - Answer

A Digital Rights Management (DRM) system is a technology framework designed to control access to and usage of digital content by enforcing licensing agreements and preventing unauthorized copying, sharing, or modification of copyrighted material. DRM systems rely heavily on cryptography for content protection, typically employing symmetric encryption algorithms like AES to encrypt the actual content and asymmetric cryptography for secure key distribution and license management, while "scrambling" refers to the process of encrypting or encoding content to make it unreadable without proper decryption keys, often involving additional techniques like watermarking and access control mechanisms. A concrete example of DRM implementation is Netflix's content protection system, which uses multiple layers including Widevine DRM that encrypts video streams with rotating keys, employs hardware-based security modules in supported devices to prevent key extraction, implements adaptive streaming with different encryption keys for different quality levels, and utilizes browser-based encrypted media extensions (EME) to securely deliver decryption keys only to authorized applications running in trusted execution environments. The system continuously authenticates the client device and software stack, revokes access for compromised devices, and employs additional anti-piracy measures like forensic watermarking to trace leaked content back to specific accounts, demonstrating how modern DRM systems integrate cryptographic protection with comprehensive access control and monitoring capabilities.