University of Malaya (Faculty of Computer Science & Information Technology)

WQD7010 Network & Security (Mid-Term Test)

Date: 15 May 2022

Time: 10.00 AM – 1.00 PM

Please answer all **Two Questions** provided

This is an OPEN BOOK test

1. Introduction to Network Security
2. CloudSecure Inc. is developing a cloud-based, multi-tenant software-as-a-service (SaaS) platform for secure document storage and sharing. The platform will be used by businesses of various sizes to store and collaborate on sensitive documents. As a security consultant, you have been tasked with designing a comprehensive security framework for the platform. Outline a high-level security architecture that incorporates at least four of the security services defined in the OSI security architecture. Explain how each service will be implemented and how they will work together to ensure the confidentiality, integrity, and availability of the stored documents. (**2 Marks**)
3. In the context of the CloudSecure Inc. platform, describe a scenario where the security goals of two different tenants using the platform might conflict. For example, one tenant may prioritize confidentiality while another may prioritize availability. Discuss how the security architecture you designed in part (a) can be adapted to address these conflicting security goals while still maintaining an acceptable level of security for all tenants. Identify any potential trade-offs or compromises that may need to be made. **(2 Marks)**

**a) High-level security architecture for CloudSecure Inc.'s SaaS platform:**

1. **Access Control:**

* **Implement role-based access control (RBAC) to ensure users can only access documents they are authorized to view or modify.**
* **Use multi-factor authentication (MFA) to verify user identities before granting access to the platform.**

1. **Data Confidentiality:**

* **Encrypt all documents at rest using a strong encryption algorithm (e.g., AES-256).**
* **Use client-side encryption, where each tenant has their own unique encryption key, to ensure that even CloudSecure Inc. cannot access the documents.**
* **Implement secure key management to protect encryption keys from unauthorized access.**

1. **Data Integrity:**

* **Use digital signatures to verify the authenticity and integrity of stored documents.**
* **Implement version control and auditing to track changes made to documents and ensure non-repudiation.**

1. **Availability:**

* **Use redundant storage and backup mechanisms to protect against data loss due to hardware failures or disasters.**
* **Implement load balancing and auto-scaling to ensure the platform remains accessible during periods of high demand.**
* **Employ DDoS protection measures to mitigate the risk of availability attacks.**

**These security services will work together to create a layered defense, with access control ensuring only authorized users can access the platform, data confidentiality protecting sensitive documents from unauthorized viewing, data integrity verifying the authenticity and reliability of stored documents, and availability measures ensuring the platform remains accessible and functional for all tenants.**

**b) Conflicting security goals scenario: Suppose Tenant A is a healthcare provider that stores highly sensitive patient records on the CloudSecure Inc. platform. They prioritize confidentiality above all else to ensure compliance with strict privacy regulations. Tenant B, on the other hand, is a fast-paced marketing agency that frequently collaborates on time-sensitive documents. They prioritize availability and ease of access to ensure their team can work efficiently.**

**To address these conflicting security goals, the security architecture could be adapted as follows:**

* **Implement a tiered encryption model, where Tenant A's documents are encrypted with a higher-strength algorithm (e.g., AES-256) and require MFA for access, while Tenant B's documents use a slightly lower-strength algorithm (e.g., AES-128) and have a more streamlined access process.**
* **Allow tenants to customize their data retention and backup policies. Tenant A may require more frequent backups and longer retention periods to ensure data is not lost, while Tenant B may prefer shorter retention periods to free up storage space and reduce costs.**
* **Implement a flexible RBAC model that allows each tenant to define their own roles and access policies based on their specific security and operational requirements.**

**The trade-off in this scenario is that Tenant B may have a slightly increased risk of unauthorized access or data breaches compared to Tenant A, due to the more relaxed encryption and access controls. However, this risk can be mitigated through regular security audits, employee training, and the implementation of other security best practices.**

**Overall, the key is to design a security architecture that is flexible enough to accommodate the diverse needs of different tenants while still providing a strong baseline of security for all users of the platform.**

1. Block Cipher
2. In CBC mode, each plaintext block is XORed with the previous ciphertext block before being encrypted, ensuring that identical plaintext blocks will encrypt to different ciphertext blocks if their preceding blocks differ. The first plaintext block is XORed with an initialization vector (IV) since there is no preceding ciphertext block.

For CBC encryption, the plaintext must consist of a sequence of complete blocks. If the final plaintext block is not complete, it is typically padded using a method like appending a "1" bit followed by the minimum number of "0" bits necessary to fill out the block. However, some security experts recommend that senders pad every message, even when the final block is already complete.

Explain the potential security benefits of always including a padding block in CBC mode, even when the padding is not strictly necessary to complete the final block. Why might this practice be considered a good habit in secure communication protocols?? (**2 Marks)**

**For this padding method, the padding bits can be removed unambiguously, provided the receiver can determine that the message is indeed padded. One way to ensure that the receiver does not mistakenly remove bits from an unpadded message is to require the sender to pad every message, including messages in which the final block is already complete. For such messages, an entire block of padding is appended.**

1. If a bit error occurs in the transmission of a ciphertext character in 8-bit CTR mode, how far does the error propagate?? (**2 Marks)**

**In CTR (Counter) mode, the keystream is generated independently of the plaintext and ciphertext. The keystream is created by encrypting successive counter values using the block cipher. The resulting keystream is then XORed with the plaintext to produce the ciphertext.**

**If a bit error occurs in the transmission of a ciphertext character in 8-bit CTR mode, the error will only affect the corresponding plaintext character after decryption. The error does not propagate to any other characters in the plaintext message.**

**This is because in CTR mode, each ciphertext bit is the result of XORing the corresponding plaintext bit with a bit from the keystream. During decryption, the ciphertext is XORed with the same keystream to recover the original plaintext. If a bit error occurs in the ciphertext during transmission, it will only affect the plaintext bit at the same position after decryption.**

**Therefore, in 8-bit CTR mode, a single bit error in the ciphertext will only corrupt the corresponding byte (8 bits) of the plaintext and will not propagate to any other characters.**

1. RC4 utilizes a secret internal configuration. This configuration can be visualized as a special ordering, or scrambling, of all the potential values for a collection named "S" and two counters, "i" and "j." (**Net Sec Ess 6ed, Problem 2.10 page 75**)
2. Part a: Basic Storage Requirements

Imagine storing this configuration in a basic manner, assigning a unique identifier to each possible combination. How many bits would we need in this scenario? (**2 Marks)**

**Simply store i, j, and S, which requires 8 + 8 + (256 x 8) = 2064 bits**

1. Part b: Information Content

Let's shift our focus to the actual information content held within the configuration. To determine this, we need to find the total number of distinct configuration arrangements (orderings) that exist. Once we have that number, we can calculate the number of bits required to represent it all. Here, we use the base-2 logarithm to express the information content in bits.

How many bits would be required to represent all the possible variations of the internal configuration? (**2 Marks**)

**The number of states is [256! x 2562] ≈ 21700. Therefore, 1700 bits are required.**

1. Hash Function and Digital Signature
2. Imagine a scenario where H(z) is a hash function designed to be resistant to collisions, converting a message of any bit length into an n-bit hash output. Can we then conclude that for every pair of unique messages, such as p and q where p ≠ q, the outputs H(p) and H(q) will also be unique? Please provide an explanation for this. **(2 marks)**

**The statement is false. Such a function cannot be one-to-one because the number of inputs to the function is of arbitrary, but the number of unique outputs is 2*n*. Thus, there are multiple inputs that map into the same output.**

1. Consider the description of an early concept for digitally signing messages using shared-secret encryption. The concept involves a sender pre-creating pairs of 56-bit keys (s1, S1, s2, S2, ..., sn, Sn) for an n-bit message. These keys are privately stored.

Publicly, the sender also sets up two groups of 64-bit validation elements (p1, P1, p2, P2, ..., pn, Pn and q1, Q1, q2, Q2, ..., qn, Qn), with each pair of elements linked to the keys through a defined encryption process where qi = E(si, pi) and Qi = E(Si, Pi).

The actual signing of a message depends on the bit values, with specific keys attached to the message according to whether a bit is 0 or 1. For instance, if the initial bits of a message are 011, the keys added to the message would be S1, S2, and s3.

Reflect on these questions based on the given setup:

1. What method should the recipient use to verify the authenticity of the message? (**2 marks**)
2. Assess the security strength of this signature technique. Assume that at that time, 56-bit keys were considered acceptable. (**2 Marks**)
3. Identify the maximum number of times this set of keys may be reused for different messages without compromising security. (**1 Mark)**
4. Are there any practical challenges this method could encounter? Look at the concept, not the 56-bit keys. (**1 Mark)**

.

**a. The recipient validates the message by ensuring that the first 56-bit key in the signature will encipher validation parameter p1 into E(s1, p1) if the first bit of the message is 0, or that it will encipher P1 into E(S1, P1) if the first bit of the message is 1; the second 56-bit key in the signature will encipher validation parameter p2 into E(s2, p2) if the second bit of the message is 0, or it will encipher P2 into E(S2, P2) if the second bit of the message is 1; and so on.**

**b. Only the sender, who knows the private values of si and Si and who originally creates pi and Pi from pi and Pi can disclose a key to the receiver. An opponent would have to discover the value of the secret keys from the plaintext-ciphertext pairs of the public key, which was computationally infeasible at the time that 56-bit keys were considered secure.**

**c. This is a one-time system, because half of the keys are revealed the first time.**

**d. A separate key must be included in the signature for each bit of the message resulting in a huge digital signature.**