Topic 1: Cryptography

A company encrypts sensitive transaction data using a block cipher with a block size of 8 bits. The data is encrypted using CFB (Cipher Feedback Mode) with an initial IV of 10100101. The resulting ciphertext of a 32-bit transaction message is:

11011010 01101001 10101101 11001100

An attacker intercepts the ciphertext and knows that the plaintext could be one of the following:

P1: 11001100 10100101 11001100 10100101

P2: 11001100 10100101 11110000 01010101

P3: 11111111 10100101 11001100 10100101

What could the attacker infer about the plaintext?

(a) Must be P1

(b) Must be P2

(c) Must be P3

(d) Could be P1 or P3, but not P2

Solution

In **CFB (Cipher Feedback Mode)**, each block of plaintext is XORed with the previous ciphertext block (or the IV for the first block) to generate the ciphertext.

Let's break it down step by step:

- **IV = 10100101** is XORed with the first plaintext block.
- This results in the first ciphertext block: **11011010**.

Now, the attacker knows the ciphertext and has potential plaintext candidates to work with:

1. **For P1**:
    - First block: **11001100**.
    - XOR the IV (10100101) with the first block of P1:
        - 10100101 = 01101001 00111001 ⊕ 10100101 = 01101001 (which, when processed, could lead to the first ciphertext block).
    - This is consistent with the second ciphertext block **01101001**, so P1 is possible.
2. **For P2**:
    - First block: **11001100** (same as P1).
    - XOR the IV (10100101) with the first block of P2:
        - 10100101 = 01101001 00111001 ⊕ 10100101 = 01101001 (same result).
    - But the third block of the ciphertext does not match what would be expected for P2, as P2's second block is significantly different, causing the next blocks to not match the ciphertext.
3. **For P3**:
    - First block: **11111111**.
    - XOR the IV (10100101) with the first block of P3:
        - 10100101 = 01011010 11111111 ⊕ 10100101 = 01011010 (which could produce the correct ciphertext).
    - This would lead to similar results in subsequent blocks, so P3 is also possible.

Since **P2** does not match the pattern and blocks, but **P1** and **P3** could both be possibilities based on how CFB works, the attacker can infer:

"The plaintext could be P1 or P3, but not P2."

Thus, the correct answer is **(d) Could be P1 or P3, but not P2**.

Topic 2: Networking Fundamentals

A company is setting up networks for three separate office branches. Each branch requires its own subnet. Branch A has 50 employees, Branch B has 75 employees, and Branch C has 30 employees. Each employee needs an IP address for their devices, and the company wants to allocate an additional 15% of addresses for future growth. What is the most efficient subnet mask you can assign to each branch to meet current and future needs?

(a) 255.255.255.224
(b) 255.255.255.240
(c) 255.255.255.192
(d) 255.255.255.128

Solution

Answer:
The most efficient subnet mask is **(c) 255.255.255.192**.

Explanation:

To determine the most efficient subnet mask, we need to calculate the number of IP addresses required for each branch, including the additional 15% for growth.

Branch A:

- **Employees:** 50
- **Future growth (15%):** 50×1.15=57.550 \times 1.15 = 57.550×1.15=57.5, rounded up to 58.
- **Required IP addresses:** 58.

Branch B:

- **Employees:** 75
- **Future growth (15%):** 75×1.15=86.2575 \times 1.15 = 86.2575×1.15=86.25, rounded up to 87.
- **Required IP addresses:** 87.

Branch C:

- **Employees:** 30
- **Future growth (15%):** 30×1.15=34.530 \times 1.15 = 34.530×1.15=34.5, rounded up to 35.
- **Required IP addresses:** 35.

Now, let's evaluate the subnet masks and the number of available addresses they provide:

- **(a) 255.255.255.224 (CIDR /27):**
    - Provides 32 IP addresses per subnet (25−2=302^5 - 2 = 3025−2=30 usable addresses).
    - This is not enough for any branch.
- **(b) 255.255.255.240 (CIDR /28):**
    - Provides 16 IP addresses per subnet (24−2=142^4 - 2 = 1424−2=14 usable addresses).

    - This is insufficient for all branches.

- **(c) 255.255.255.192 (CIDR /26):**
    - Provides 64 IP addresses per subnet (26−2=622^6 - 2 = 6226−2=62 usable addresses).
    - Sufficient for Branch A (58) and Branch C (35), but not for Branch B (87).
- **(d) 255.255.255.128 (CIDR /25):**
    - Provides 128 IP addresses per subnet (27−2=1262^7 - 2 = 12627−2=126 usable addresses).
    - More than enough for Branch A, B, and C, but not as efficient as option (c) for smaller branches.

Thus, **255.255.255.192** (option (c)) is the most efficient mask for Branch A and C, and for Branch B, a larger subnet mask like **255.255.255.128** would be needed. If the goal is to use one subnet mask for all branches, **255.255.255.128** would work, but it's less efficient than using **255.255.255.192** for smaller branches.

Topic: Red Teaming and Introduction to CTF

A company has a web application that allows users to submit forms containing special characters. However, the input fields were not properly sanitized. During a red team assessment, the team injected malicious code into the input field, causing the database to return sensitive information, such as customer records and internal data, when they submitted the form. What type of exploitation did the red team use?

(a) Cross-Site Scripting (XSS)
(b) SQL Injection
(c) Buffer Overflow
(d) Privilege Escalation

Solution

Answer:
(b) SQL Injection

Explanation:

In this scenario, the red team exploited the lack of input validation to inject malicious SQL code into the input field of a web application. SQL Injection occurs when an attacker inserts or manipulates SQL queries into an input field to access, modify, or delete database information that they wouldn't normally have access to.

The fact that the red team was able to retrieve sensitive customer records and internal data directly from the database confirms that **SQL Injection** was the technique used.

- **Cross-Site Scripting (XSS)**: Involves injecting malicious scripts into web pages to target other users, not the database.
- **Buffer Overflow**: Involves overflowing the memory buffer with data, potentially leading to system crashes or arbitrary code execution.
- **Privilege Escalation**: Involves gaining higher privileges or permissions than initially granted, usually after already compromising a system.

Topic: Digital Forensics

During a digital forensics investigation, the team carefully collects and secures data from various devices, including hard drives, mobile phones, and network logs, to ensure that no data is altered or tampered with. The team is following strict protocols to maintain the integrity of the evidence, which will be used in court if necessary. What stage of the digital forensics process is the team currently in?

(a) Analysis
(b) Reporting
(c) Collection
(d) Preservation

Solution

Answer:
(d) Preservation

Explanation:

The scenario describes the process of securing and maintaining the integrity of data so that it remains unchanged and can be used as evidence later. This is the **preservation** stage of digital forensics, where forensic investigators ensure that data is protected from modification or tampering.

- **Collection** refers to physically acquiring the data from devices, often happening concurrently with preservation.
- **Analysis** involves examining the collected data to find relevant evidence.
- **Reporting** is the final stage where findings are documented in a report.