Okay, I can create a mock scenario for risk management based on the vulnerability scan results.

Risk Management Strategies: Mock Scenario

Project Goal: To present risk management understanding through practical application, including risk identification from vulnerability scan results, treatment recommendations, and risk monitoring procedures.

I. Identification of Risks from Vulnerability Scan Results

Let's assume a vulnerability scan was conducted on a web server, and the following critical vulnerabilities were identified:

Vulnerability 1: SQL Injection Vulnerability in Login Form

- **Description:** The web server's login form is vulnerable to SQL injection attacks. This allows an attacker to inject malicious SQL code into the form's input fields, potentially gaining unauthorized access to the database.
- **CVSS Score:** 9.8 (Critical)
- **Explanation:**
    - This is a critical risk because successful exploitation could lead to the complete compromise of the database, which likely contains sensitive user credentials and other confidential data.
    - The impact is high, affecting confidentiality, integrity, and availability.
    - The likelihood is also high, as SQL injection is a well-known and easily exploitable vulnerability.
- **Treatment Recommendation:**
    - **Remediation:** The vulnerability must be fixed immediately by implementing parameterized queries or using an Object-Relational Mapping (ORM) library to sanitize user inputs. Input validation on the server-side is also crucial.
- **Basic Mitigation Steps:**
    - **Isolate the web server:** To prevent further damage, the web server should be isolated from other systems until the vulnerability is patched.
    - **Apply a web application firewall (WAF):** A WAF can provide temporary protection by filtering out malicious SQL injection attempts.
    - **Notify relevant stakeholders:** Inform the security team, system administrators, and data owners about the vulnerability and the steps being taken to address it.

Vulnerability 2: Unpatched Remote Code Execution Vulnerability in Web Server Software

- **Description:** The web server software (e.g., Apache, Nginx) is running an outdated version with a known remote code execution (RCE) vulnerability. This allows an attacker to execute arbitrary code on the server.
- **CVSS Score:** 10.0 (Critical)
- **Explanation:**
    - This is a critical risk because successful exploitation would give the attacker complete control over the web server.

    - The impact is catastrophic, as the attacker could steal data, install malware, or use the server as a launchpad for other attacks.
    - The likelihood is high, especially if the vulnerability is publicly known and exploit code is available.
- **Treatment Recommendation:**
    - **Remediation:** The web server software must be patched or upgraded to the latest secure version immediately.
- **Basic Mitigation Steps:**
    - **Take the server offline:** If patching cannot be done immediately, the server should be taken offline to prevent exploitation.
    - **Network segmentation:** Ensure the web server is on a segmented network to limit the impact of a potential breach on other systems.
    - **Intrusion Detection System (IDS):** Implement or ensure an IDS is in place to detect any exploitation attempts.

II. Risk Monitoring Procedure

Risk Monitoring Procedure: Critical Vulnerability Tracking

- **Purpose:** To ensure that identified critical vulnerabilities are tracked, remediated, and verified in a timely manner.
- **Scope:** This procedure applies to all critical vulnerabilities identified through vulnerability scans or other security assessments.
- **Roles and Responsibilities:**
    - **Security Team:** Responsible for conducting vulnerability scans, identifying critical vulnerabilities, assigning remediation tasks, and tracking progress.
    - **System Administrators:** Responsible for implementing remediation steps as assigned by the security team.
    - **Risk Management Team:** Responsible for overseeing the risk management process and ensuring that risks are mitigated according to organizational policies.
- **Procedure Steps:**
    - **Vulnerability Identification:**
        - Vulnerability scans are conducted regularly (e.g., weekly or monthly) using automated scanning tools.
        - The security team reviews the scan results and identifies critical vulnerabilities based on CVSS scores and potential impact.
    - **Risk Assessment:**
        - For each critical vulnerability, a risk assessment is performed to determine the likelihood and impact of exploitation.
        - The risk level is calculated (e.g., using a risk matrix) and documented.
    - **Remediation Planning:**
        - The security team assigns remediation tasks to the appropriate system administrators.
        - A remediation timeline is established based on the severity of the vulnerability and the complexity of the fix.
    - **Remediation Implementation:**

        - System administrators implement the remediation steps according to the assigned tasks and timeline.
        - Any issues or delays are reported to the security team.
    - **Verification:**
        - The security team verifies that the remediation steps have been implemented correctly and that the vulnerability has been resolved.
        - A follow-up scan may be performed to confirm the fix.
    - **Documentation:**
        - All steps of the process, including vulnerability details, risk assessments, remediation actions, and verification results, are documented in a central tracking system.
    - **Escalation:**
        - If remediation is not completed within the agreed-upon timeline, the risk is escalated to the Risk Management Team for further action.
    - **Reporting:**
        - Regular reports on the status of critical vulnerabilities are provided to management and relevant stakeholders.
- **Monitoring Metrics:**
    - Number of critical vulnerabilities identified.
    - Time to remediate critical vulnerabilities.
    - Number of vulnerabilities exceeding the remediation timeline.
    - Percentage of vulnerabilities successfully remediated.
- **Justification:** This procedure ensures that critical vulnerabilities, which pose the greatest risk to the organization, are addressed promptly and effectively. Regular monitoring and reporting provide visibility into the remediation process and help to ensure accountability.

This mock scenario provides a comprehensive example of risk management strategies, including the identification of critical risks from vulnerability scan results and a detailed risk monitoring procedure.