National University of Computer and Emerging Sciences

Islamabad Campus

# Information Security (CS3002)

# Final Examination

**Total Time (Hrs):** 3

**Total Marks:** 200

**Total Questions:** 7

**Course Instructor(s):**

Dr. Noshina Tariq

**Section(s): BS(DS) M, N, U**

**Date:** Dec 30, 2024

| _____ | _____ | _____ |
|---|---|---|
| **Roll No** | **Course Section** | **Student Signature** |

**Do not write below this line.**

**Attempt all the questions.**

**[CLOs 1, 3, and 4: Explain key concepts of information security, analyze real world scenarios, and identify appropriate techniques to tackle and solve problems of real life in the discipline of IS]**

**Q1:** An organization has reported unusual activity on its network. The cybersecurity team has identified three distinct incidents: **[4 x 10=40 marks]**

1. **Incident A:** An advanced business optimization software, marketed as an "AI-powered efficiency enhancer," was deployed organization-wide after being recommended by several industry experts. The application appeared functional and legitimate, offering useful features that improved task automation. However, network monitoring tools began detecting unusually high volumes of encrypted outbound traffic originating from the software, sent to servers with no identifiable ownership or valid registration records. Despite this, endpoint protection solutions failed to flag the software as malicious. Moreover, the application intermittently requested administrative privileges, often during non-operational hours, without any discernible reason. Its legitimate interface and functionality made it difficult to assess whether the behavior was part of its intended design or a covert activity embedded within the software.
2. **Incident B:** In a regional office, systems connected to the internal network began exhibiting abnormal behavior during non-working hours. Devices attempted unauthorized access to other systems across the subnet, triggering a cascade of unusual activity. Network logs revealed the exploitation of a vulnerability in the file-sharing protocol, allowing malicious activity to propagate rapidly without requiring user input. Within a few hours, over 85% of connected systems were impacted, showing symptoms such as overwritten configurations, corrupted system files, and significantly degraded performance. Attempts to isolate the compromised systems were unsuccessful, as the malicious activity re-executed itself upon each system reboot. The code displayed adaptive behavior, altering its patterns dynamically to evade endpoint security measures, leaving administrators struggling to regain control of the network.
3. **Incident C:** Employees using an internal file management tool for processing high-priority business data reported erratic crashes and corrupted files across multiple systems. This issue was initially dismissed as a technical malfunction, but closer inspection revealed that the corruption selectively targeted critical files such as spreadsheets and presentations. IT staff traced the issue to a software update that had been automatically distributed through the organization's internal update mechanism. The update included a component embedding

encrypted instructions that triggered file corruption during specific workflows. Notably, this corruption did not appear to be random; it seemed to disrupt operations deliberately but without demanding any ransom or issuing threats, leaving the organization puzzled about the intent and origin of the malicious behavior.

4. **Incident D:** The IoT-based smart facility systems in the organization experienced widespread malfunction, disrupting critical operations. Temperature control systems began fluctuating uncontrollably, lighting systems turned on and off without user input, and door access systems became unresponsive. Investigations revealed that many IoT devices were still using factory-default credentials, which had been exploited to gain unauthorized access. Furthermore, the communication protocols between devices and their management platform lacked proper encryption, making them vulnerable to interception and manipulation. Logs were unavailable, as the devices lacked any monitoring or logging mechanisms, further complicating the investigation. Additionally, some devices ceased functioning entirely after the attack, suggesting that their firmware might have been overwritten or rendered inoperative, leaving administrators unable to restore normal operations promptly.

Using your knowledge of malware, answer the following:

1. Based on the observed behavior, identify the most probable threat category associated with each incident. Your analysis should detail how the characteristics of the incident align with specific categories of malicious activity.
2. For each incident, propose a mitigation strategy addressing the specific nature of the threat. Ensure your recommendations consider the technical and operational aspects of the organization.
3. What organizational measures should be implemented to prevent such incidents in the future? Suggest at least three actions, considering the vulnerabilities exposed.
4. Incident D demonstrates the risks associated with IoT environments. Outline the key components of a security framework that would mitigate such risks. Your framework should consider device communication, user authentication, and operational updates, but you are not limited to these elements.

Solution: **1. Identifying Threat Categories**

For each incident, the threat categories and alignment with characteristics are as follows:

**Incident A: Trojan Horse**

- **Characteristics:** The software appears legitimate and functional but exhibits covert behavior (e.g., sending encrypted data to unknown servers and requesting administrative privileges at odd times). These are hallmarks of Trojan horse malware, designed to perform malicious actions while appearing benign.

**Incident B: Worm**

- **Characteristics:** Rapid, self-propagating activity exploiting a file-sharing protocol vulnerability and dynamically adapting to evade detection. These traits are consistent with a network worm.

**Incident C: Logic Bomb**

- **Characteristics:** Malicious code embedded in an update mechanism, triggered under specific conditions to corrupt critical files deliberately. This behavior aligns with a logic bomb.

**Incident D: IoT-Specific Malware**

- **Characteristics:** Exploits default credentials, lacks encryption, and manipulates IoT devices, indicating specialized malware targeting IoT vulnerabilities.

---

**2. Mitigation Strategies**

**Incident A: Trojan Horse**

- **Technical Mitigation:**
  - Conduct a forensic analysis to verify the software's behavior.
  - Isolate affected systems and monitor outbound traffic to unknown servers.
  - Use application whitelisting to restrict software permissions.
- **Operational Measures:**
  - Restrict administrative privilege requests through strict access control policies.
  - Require thorough vendor validation and software testing before deployment.

**Incident B: Worm**

- **Technical Mitigation:**
  - Patch the exploited file-sharing protocol vulnerability across all systems.
  - Deploy network segmentation to isolate infected systems and limit propagation.
  - Use behavior-based intrusion detection systems (IDS).
- **Operational Measures:**
  - Conduct regular vulnerability assessments to identify exploitable protocols.
  - Train IT staff to respond to rapid propagation scenarios.

**Incident C: Logic Bomb**

- **Technical Mitigation:**
  - Roll back the update and deploy a clean version of the software.
  - Implement version control and digital signatures for all internal updates.
  - Use endpoint protection tools with heuristic analysis to detect embedded malicious code.
- **Operational Measures:**
  - Strengthen software development lifecycle (SDLC) security, including code reviews.
  - Audit internal update mechanisms regularly.

**Incident D: IoT-Specific Malware**

- **Technical Mitigation:**
  - Replace factory-default credentials with strong, unique passwords.
  - Enforce encrypted communication protocols between IoT devices and management platforms.
  - Implement firmware integrity checks and backups.

- **Operational Measures:**
  - Train facility managers on IoT-specific security risks.
  - Schedule routine security reviews for IoT deployments.

---

**3. Organizational Measures**

1. **Comprehensive Cybersecurity Policy:**
   - Establish policies for secure software deployment, patch management, and access control.
2. **Employee Training:**
   - Conduct regular cybersecurity awareness programs to educate employees about threats like social engineering and phishing.
3. **Continuous Monitoring and Incident Response:**
   - Deploy advanced threat detection tools for real-time monitoring.
   - Create a dedicated incident response team and establish clear response protocols.

---

**4. Security Framework for IoT Environments**

**Key Components:**

1. **Device Communication:**
   - Use secure communication protocols (e.g., TLS/SSL) for device-to-device and device-to-server interactions.
   - Implement mutual authentication to validate devices before communication.
2. **User Authentication:**
   - Enforce multi-factor authentication for access to IoT device management systems.
   - Maintain a centralized identity and access management system.
3. **Operational Updates:**
   - Secure update mechanisms using cryptographic signatures to ensure authenticity.
   - Schedule periodic updates to address known vulnerabilities.
4. **Monitoring and Logging:**
   - Deploy logging mechanisms to track all device activity and identify anomalies.
   - Store logs in a secure, centralized repository for auditing.
5. **Network Segmentation:**
   - Isolate IoT devices from critical business systems using VLANs or dedicated subnets.
6. **Firmware Integrity:**
   - Use hardware-based security modules (e.g., TPM) to verify firmware integrity during boot-up.
7. **Resilience Mechanisms:**
   - Implement fail-safe mechanisms to restore device functionality in case of firmware compromise.

**[CLOs 1, 3, and 4: Explain key concepts of information security, analyze real world scenarios, and identify appropriate techniques to tackle and solve problems of real life in the discipline of IS]**

**Q2:** At XYZ University, the Data Science Department's 7th-semester students participated in an information security competition. The assigned task by the organizing committee is to demonstrate an SQL injection attack on a local database and showcase its mitigation measures. **[30 marks]**

The database name is ***CSDB***, and it contains a table named ***users*** with the structure shown below:

| id | FullName | MobileNumber | EmailId | Password | RegDate | UpdateDate |
|---|---|---|---|---|---|---|
| 1 | Umer Farooq | 0313953193 | farooqumer82@yahoo.com | 827ccb0e8a706c4c34a16891f84e7b | 2019-07-23 14:14:14 | 2019-07-24 17:57:37 |

Using the above scenario, answer the following:

1. Write three different SQL injection attacks to exploit the users table, that is retrieving data, Display Database Schema, and Blind SQL Injection. Clearly show how the injection works. **[8 marks]**
2. Discuss why removing special characters would not help mitigating an SQL Injection attack with examples. **[5 marks]**
3. Propose and explain a mitigation technique to prevent this SQL injection attack for the attacks you made in **(1).** **[8 marks]**
4. Is part **(3)** a read or a write operation? Justify your answer with example. **[2 marks]**
5. When mitigating the SQL injection attack in part **(3)**, does the operation involve reading or writing? Justify your answer. **[2 marks]**
6. Discuss where the mitigation technique from part **(3)** should be applied: on the database, the application layer, or both. Justify your choice. **[5 marks]**

**Solution:**

## Systematic Solution to SQL Injection Problem

### 1. SQL Injection Queries

#### Query for Retrieving Data

SELECT * FROM users WHERE EmailId = '' OR '1'='1' -- AND Password = '';

Explanation: The injected condition '1'='1' always evaluates to true, retrieving all data from the users table. The -- comments out the remaining portion of the query.

#### Query for Displaying Database Schema

SELECT * FROM users WHERE EmailId = ''; SHOW TABLES; -- AND Password = '';

Explanation: The ; ends the first query. The SHOW TABLES command lists all tables in the database. The -- comments out the rest of the original query.

### Query for Blind SQL Injection

SELECT * FROM users WHERE EmailId = '' AND (SELECT 1 WHERE (SELECT COUNT(*) FROM users) > 0) -- AND Password = '';

Explanation: This query checks if the users table has any records by counting them with COUNT(*). The condition (SELECT COUNT(*) FROM users) > 0 evaluates as true or false, allowing attackers to infer data indirectly.

## 2. Why Removing Special Characters Doesn't Work

- Encoded Payloads: Attackers can use encoded inputs (e.g., %27 for ') to bypass character filtering.
Example: SELECT * FROM users WHERE EmailId = CHAR(39)+CHAR(79)+CHAR(82)+CHAR(39)+CHAR(49)+CHAR(61)+CHAR(49) --;

- Bypassing Filters: Filtering ' or ; doesn't protect against malicious patterns like:
SELECT * FROM users WHERE EmailId = '' UNION SELECT * FROM other_table --;

## 3. Mitigation Techniques

### Prepared Statements and Parameterized Queries

Use parameterized queries where user input is treated as data, not executable code.
Example query:
SELECT * FROM users WHERE EmailId = ? AND Password = ?;

### Input Validation

Strictly validate inputs to allow only acceptable patterns (e.g., alphanumeric for emails). Reject inputs containing suspicious characters like ', --, or ;.

### Stored Procedures

Use stored procedures to execute queries securely.
Example:

```
CREATE PROCEDURE GetUser(IN email VARCHAR(255), IN password VARCHAR(255))
BEGIN
  SELECT * FROM users WHERE EmailId = email AND Password = password;
END;
```

### Web Application Firewall (WAF)

Deploy a WAF to identify and block SQL injection patterns in incoming requests.

## 4. Is Part (3) a Read or Write Operation?

It is a Read Operation.
Justification: The prepared statement reads data from the database securely without modifying any records. For example, SELECT queries fetch data based on sanitized inputs.

## 5. Mitigation Operation Involves Reading or Writing?

It is a Reading Operation.
Justification: Input sanitization and query execution involve reading user-provided inputs and securely passing them to the database.

### 6. Where to Apply the Mitigation Techniques

Application Layer
Justification: Input is first processed in the application layer, making it the primary point to prevent SQL injections. Apply techniques like prepared statements and input validation at this level.

Database Layer
Justification: Implement stored procedures, database permissions, and activity monitoring to safeguard against unhandled threats.

Best Practice: Both Layers
Mitigating SQL injections at both the application and database layers ensures:
1. Application Layer: Blocks malicious inputs before they reach the database.
2. Database Layer: Provides additional protection using database-native features.

**[CLOs 1, 3, and 4: Explain key concepts of information security, analyze real world scenarios, and identify appropriate techniques to tackle and solve problems of real life in the discipline of IS]**

**Q3:** An educational portal allows teachers to post announcements. These announcements are displayed to students on their dashboard. During a recent assessment, a teacher unknowingly entered the following text in the announcement field:

**Welcome <students>! Click here for more details: <a href="http://example.com">Details</a>**

Students started reporting that a pop-up was appearing on their dashboard after viewing this announcement. A deeper investigation revealed that the field used to store announcements was vulnerable to XSS. **[30 marks]**

1. Explain how XSS vulnerabilities can occur in applications like this portal. What specific flaw allowed the pop-up to appear? **[8 Marks]**
2. Discuss the potential risks to the students and the portal if such XSS vulnerabilities are exploited. **[6 Marks]**
3. Suggest two techniques to mitigate XSS in this portal and explain how each prevents such attacks. **[6 Marks]**

# Solution:

## Cross-Site Scripting (XSS) Vulnerability Analysis

### 1. How XSS Vulnerabilities Occur in Applications

Cross-Site Scripting (XSS) vulnerabilities occur when an application fails to properly validate or sanitize user inputs. In this scenario, the teacher unknowingly entered an HTML tag in the announcement field. The application directly stored and rendered the input without escaping potentially harmful content.

The specific flaw is that the input field accepts raw HTML, allowing malicious scripts to execute when the data is displayed. The pop-up appeared because the embedded script executed in the browser's context when students accessed the announcement page.

## 2. Potential Risks to Students and the Portal

If XSS vulnerabilities are exploited, the following risks may arise:

1. **Data Theft**: Malicious scripts can steal session cookies, login credentials, or other sensitive data from students.

2. **Phishing Attacks**: Attackers can redirect students to fake websites to gather personal or login information.

3. **Defacement**: Attackers may inject content to deface the portal or spread misinformation.

4. **Malware Distribution**: XSS can be used to distribute malicious software to users accessing the portal.

5. **Loss of Trust**: Students and teachers may lose trust in the portal's security, impacting its reputation and usage.

## 3. Techniques to Mitigate XSS

### Technique 1: Input Sanitization and Validation

Sanitize all user inputs by removing or encoding potentially harmful content. For example, replace `<` and `>` with their HTML entities `<` and `>` to ensure scripts are displayed as text instead of being executed.

This ensures that malicious scripts are neutralized before they are stored or displayed.

### Technique 2: Use Content Security Policy (CSP)

Implement a CSP to restrict the execution of scripts from untrusted sources. CSP can block inline scripts and only allow scripts from trusted domains, reducing the risk of XSS attacks.

This prevents malicious scripts from being executed even if they are injected into the application.

**[CLOs 1, 3, and 4: Explain key concepts of information security, analyze real world scenarios, and identify appropriate techniques to tackle and solve problems of real life in the discipline of IS]**

**Q4:** An online banking system provides the following feature: **[30 marks]**

- Logged-in users can transfer money by submitting a POST request to http://bank.com/transfer with the following body:

```
{
  "amount": 1000,
  "to_account": "12345678"
}
```

- The session is authenticated using a cookie sent by the user's browser.

An attacker creates a malicious webpage with an invisible form that automatically submits this POST request when a logged-in user visits the page.

1. Explain why the attack succeeds even though the banking system authenticates the session with a cookie. **[10 Marks]**
2. Describe the consequences of a successful CSRF attack on this system for the user and the bank. **[10 Marks]**
3. Propose two mechanisms to defend against such CSRF attacks and explain how each approach mitigates the vulnerability. **[10 Marks]**

**Solution:**

# Cross-Site Request Forgery (CSRF) Vulnerability Analysis

## 1. Why the Attack Succeeds

The attack succeeds because the banking system relies solely on cookies for session authentication. When a user is logged in, the browser automatically includes the user's session cookie with all requests to the bank's domain. The malicious webpage triggers a POST request to the bank's endpoint with the hidden form, and the browser attaches the session cookie, making the request appear legitimate to the server. The server cannot distinguish between a request initiated by the user and one crafted by an attacker.

## 2. Consequences of a Successful CSRF Attack

The consequences of a successful CSRF attack include:
1. **Unauthorized Transactions**: The attacker can transfer funds from the user's account without their consent.
2. **Financial Loss**: The user may suffer financial loss due to the unauthorized transaction.
3. **Reputation Damage**: The bank's reputation may be harmed due to the perceived lack of security.
4. **Legal Implications**: The bank could face legal actions or regulatory penalties for failing to protect user data and transactions.

## 3. Mechanisms to Defend Against CSRF Attacks

### Technique 1: CSRF Tokens

Include a CSRF token with each request. The token is generated on the server and embedded in forms as a hidden input field. The server validates the token with each incoming request. Since the attacker cannot access the token, they cannot craft a valid request.
How it mitigates the vulnerability: The server will reject any request lacking a valid token, ensuring the request originates from the legitimate user.

### Technique 2: SameSite Cookies

Configure session cookies with the SameSite attribute. This restricts cookies from being sent with cross-origin requests.
How it mitigates the vulnerability: The browser will not include the session cookie when the request is initiated from a different domain, blocking the attacker's crafted request.

**[CLOs 1, 3, and 4: Explain key concepts of information security, analyze real world scenarios, and identify appropriate techniques to tackle and solve problems of real life in the discipline of IS]**

**Q5:** A social media platform uses cookies for session management, personalization, and analytics tracking. Recently, the platform faced issues where: **[20 marks]**

1. User session cookies were stolen, leading to unauthorized account access.
2. Third-party advertisers tracked users' browsing behavior across multiple websites.
3. Some cookies were sent over unencrypted HTTP requests, exposing them to attackers intercepting network traffic.

**Task**

1. Analyze each issue in the scenario and explain how it poses a security or privacy risk. Use the following breakdown: **[8 Marks]**
   - Session cookie theft
   - Third-party cookies for tracking
   - Cookies sent over HTTP
2. Propose two mitigation strategies for each issue. Explain how each strategy addresses the respective problem. For example: **[8 Marks]**
   - For session cookie theft: Discuss HttpOnly and Secure attributes.
   - For third-party tracking: Suggest cookie blocking mechanisms.
   - For cookies sent over HTTP: Recommend HTTPS enforcement and cookie attributes.
3. Explain the role of **SameSite** cookie attributes (Lax and Strict) in preventing Cross-Site Request Forgery (CSRF) and how they can enhance the security of the platform. **[4 Marks]**

**Solution:**

# Cookie Security and Privacy Risk Analysis

## 1. Analysis of Security and Privacy Risks

### Session Cookie Theft

Risk: If session cookies are stolen, attackers can impersonate users and gain unauthorized access to their accounts. This can lead to data breaches, unauthorized transactions, or abuse of platform features.

### Third-Party Cookies for Tracking

Risk: Third-party advertisers use cookies to track users' browsing behavior across multiple websites, violating user privacy. This can lead to unwanted targeted advertisements and potential misuse of personal data.

### Cookies Sent Over HTTP

Risk: Cookies sent over unencrypted HTTP connections are exposed to attackers intercepting network traffic, making them vulnerable to theft via man-in-the-middle (MITM) attacks.

## 2. Mitigation Strategies

### Session Cookie Theft

1. **HttpOnly Attribute**: Marks cookies as inaccessible to client-side scripts, preventing theft via XSS attacks.
2. **Secure Attribute**: Ensures cookies are sent only over encrypted HTTPS connections, reducing exposure to MITM attacks.

### Third-Party Cookies for Tracking

1. **Cookie Blocking Mechanisms**: Implement browser-level or platform-specific mechanisms to block third-party cookies.
2. **User Consent and Control**: Require explicit user consent for third-party cookie usage and provide options to opt-out of tracking.

### Cookies Sent Over HTTP

1. **HTTPS Enforcement**: Ensure all platform communications are conducted over HTTPS to encrypt cookies in transit.
2. **Secure Attribute**: Configure cookies with the Secure attribute to guarantee transmission only over encrypted channels.

## 3. Role of SameSite Cookie Attributes in Preventing CSRF

The SameSite attribute restricts cookies from being sent with cross-origin requests, mitigating CSRF attacks by ensuring cookies are only sent with requests initiated from the same origin.

### SameSite=Lax

Allows cookies to be sent with top-level navigation requests (e.g., clicking a link), but not with embedded content (e.g., images). This provides a balance between security and usability.

### SameSite=Strict

Restricts cookies to first-party requests only, enhancing security by preventing cookies from being sent with any cross-origin request. This is useful for highly sensitive applications where strict control over cookie usage is required.

**[CLOs 1, 3, and 4: Explain key concepts of information security, analyze real world scenarios, and identify appropriate techniques to tackle and solve problems of real life in the discipline of IS]**

**Q6:** A small business recently experienced a series of cybersecurity incidents, including phishing attacks, ransomware, and a Distributed Denial-of-Service (DDoS) attack. The company has decided to implement a risk management strategy using the **NIST Cybersecurity Framework** to protect its systems and data. **[20 marks]**

1. Explain the five core functions of the **NIST Cybersecurity Framework** (*Identify, Protect, Detect, Respond, Recover*), and provide one example of an activity or measure for each function as applied to the small business. **[8 Marks]**
2. For each of the following incidents, describe how the business can mitigate the risk using the framework: **[8 Marks]**
    - Phishing attacks targeting employees' credentials.
    - Ransomware encrypting critical business data.
    - DDoS attacks disrupting the business's online services.
3. Small businesses often have limited resources for cybersecurity. Suggest two practical and cost-effective measures that align with the **Protect** and **Detect** functions of the NIST Framework. **[4 Marks]**

**Solution:**

## Cybersecurity Risk Management Using the NIST Framework

### 1. Core Functions of the NIST Cybersecurity Framework

The NIST Cybersecurity Framework outlines five core functions to manage and reduce cybersecurity risks effectively. Each function addresses specific aspects of risk management and can be applied to the small business as follows:

#### Identify

Definition: Develop an understanding of the business's cybersecurity risks and assets.
Example Activity: Conduct an inventory of IT assets and data to identify critical systems and potential vulnerabilities.

#### Protect
Definition: Implement safeguards to ensure critical infrastructure and data are protected.
Example Activity: Enforce access control policies and use multi-factor authentication for employee accounts.

#### Detect
Definition: Establish processes to identify cybersecurity incidents.
Example Activity: Set up real-time monitoring and alerts for suspicious activity on the network.

#### Respond
Definition: Develop plans to respond to cybersecurity incidents effectively.
Example Activity: Create an incident response plan detailing steps to handle phishing or ransomware attacks.

#### Recover
Definition: Restore systems and operations affected by cybersecurity incidents.
Example Activity: Maintain secure and regularly updated backups of critical business data.

### 2. Mitigating Risks for Specific Incidents

#### Phishing Attacks
Mitigation: Use the Protect function to implement email filtering and employee training on identifying phishing emails. The Detect function can be used to monitor for unusual login activity indicating compromised credentials.

#### Ransomware
Mitigation: Use the Protect function to enforce regular software updates and enable endpoint protection. The Respond function involves disconnecting infected systems from the network and using clean backups to restore data.

#### DDoS Attacks
Mitigation: Use the Protect function to configure a web application firewall (WAF) and limit the impact of large-scale requests. The Detect function involves using monitoring tools to identify abnormal traffic patterns in real time.

### 3. Cost-Effective Cybersecurity Measures

#### Protect Function
1. Use strong passwords and implement multi-factor authentication to reduce unauthorized access risks.

#### Detect Function
1. Deploy free or low-cost intrusion detection systems (e.g., Snort) to monitor network traffic and identify potential threats.

**[CLOs 1, 3, and 4: Explain key concepts of information security, analyze real world scenarios, and identify appropriate techniques to tackle and solve problems of real life in the discipline of IS]**

**Q7:** A mid-sized financial organization recently transitioned to a cloud-based system to store sensitive customer data and support real-time transactions. While the new infrastructure promised improved scalability and performance, several issues emerged shortly after deployment: **[30 Marks]**

1. **Event A:** A security audit conducted by an external consultancy flagged misconfigured access controls in the cloud-based storage system. The permissions were overly permissive, potentially allowing unauthorized external users to access sensitive data, including transaction records and customer details. The audit also noted a lack of clear ownership and responsibility for managing access controls. These misconfigurations went unnoticed during the organization's cloud migration and were only discovered during the external audit.
2. **Event B:** An internal review highlighted a significant risk of insider threats. Employees were granted broad access privileges far beyond their job requirements, violating the principle of least privilege. Additionally, certain sensitive systems were accessible without adequate tracking or restrictions. The review emphasized that an insider, either accidentally or maliciously, could compromise sensitive data or disrupt critical operations. The organization's management concluded that the impact of such a scenario could be severe, yet external risks were not considered during this analysis.
3. **Event C:** In response to earlier findings, the organization implemented security measures to safeguard its cloud systems. These included encryption for all data stored in the cloud and multi-factor authentication (MFA) for user access. However, they overlooked monitoring mechanisms such as logging and alerting. A gap analysis later revealed that orphaned accounts of former employees were still active, creating potential entry points for attackers. Additionally, the organization had no processes to monitor for suspicious behavior in real-time, leaving it vulnerable to slow-moving attacks that could exploit these gaps.
4. **Event D:** Months after the security controls were implemented, the IT team noticed unusual patterns of access to the cloud storage. Several access attempts originated from foreign IP addresses during non-business hours. These patterns were detected during a manual review of access logs, which revealed that the organization lacked any automated monitoring or alerting tools. Furthermore, periodic risk assessments were not performed, and vulnerabilities in the cloud management platform remained unaddressed. The lack of continuous oversight allowed potential threats to remain undetected for extended periods.

## Answer the following:

1. Identify and explain the risk management process (or processes) applicable to each of the events described above. Justify your answers by connecting the details of the events to the stages of the risk management lifecycle. **[9 Marks]**
2. Evaluate the decisions and actions taken in the events described. Were they sufficient to address the risks effectively? Highlight specific gaps and weaknesses in the organization's approach and suggest improvements. **[9 Marks]**
3. Based on the events, what general weaknesses can you identify in the organization's risk management approach? Propose three strategic actions the organization should implement to improve its overall risk management capabilities. **[6 Marks]**

4. For Event D, propose a comprehensive security strategy that would mitigate the vulnerabilities identified and ensure proactive threat detection in the future. Your strategy should include both technical and procedural recommendations. **[6 Marks]**

## Cloud-Based Risk Management and Security Strategy

### 1. Applicable Risk Management Processes

#### Event A: Misconfigured Access Controls

Applicable Process: **Risk Identification and Risk Assessment**
Justification: The overly permissive access controls and lack of clear ownership were identified during the external audit. This aligns with the risk identification stage, where vulnerabilities are discovered, and the assessment stage, where their potential impact is evaluated.

#### Event B: Insider Threats

Applicable Process: **Risk Assessment and Risk Mitigation**
Justification: The internal review highlighted insider threats and the violation of the principle of least privilege, requiring mitigation strategies such as restricting access privileges and implementing stronger controls.

#### Event C: Orphaned Accounts and Monitoring Gaps

Applicable Process: **Risk Monitoring and Risk Mitigation**
Justification: The gap analysis revealed orphaned accounts and a lack of monitoring mechanisms. This falls under risk monitoring to identify ongoing vulnerabilities and risk mitigation to address them effectively.

#### Event D: Unusual Access Patterns

Applicable Process: **Risk Detection and Risk Response**
Justification: The unusual access patterns identified during manual log reviews indicate a lack of automated detection and response mechanisms. These processes focus on identifying and responding to potential threats in real-time.

### 2. Evaluation of Decisions and Actions Taken

The actions taken were partially effective but left significant gaps in addressing the identified risks. For example:

1. **Event A**: No proactive access control review was conducted during migration. Improvement: Implement strict access control policies and conduct regular audits.

2. **Event B**: Broad privileges were granted to employees. Improvement: Enforce the principle of least privilege and establish tracking for access.

3. **Event C**: Security measures like encryption and MFA were implemented, but monitoring mechanisms were overlooked. Improvement: Introduce logging and alerting systems to detect suspicious activities.

4. **Event D**: Lack of automated threat detection allowed patterns to go unnoticed. Improvement: Deploy automated monitoring and periodic risk assessments.

## 3. General Weaknesses and Strategic Actions

General Weaknesses Identified:

1. Lack of proactive monitoring and alerting mechanisms.
2. Absence of periodic risk assessments.
3. Inadequate enforcement of access controls and the principle of least privilege.

Strategic Actions to Improve Risk Management:

1. **Implement Continuous Monitoring**: Use automated tools to monitor and alert for suspicious activities in real-time.

2. **Conduct Regular Risk Assessments**: Periodically evaluate risks to identify and address vulnerabilities.

3. **Enhance Access Controls**: Enforce the principle of least privilege and regularly review access permissions.

## 4. Comprehensive Security Strategy for Event D

### Technical Recommendations

1. **Deploy Automated Monitoring Tools**: Use SIEM (Security Information and Event Management) solutions to analyze access logs and detect anomalies.

2. **Enable Geo-Blocking**: Restrict access from foreign IP addresses unless explicitly required.

3. **Conduct Vulnerability Assessments**: Regularly scan the cloud management platform for vulnerabilities and apply necessary patches.

### Procedural Recommendations

1. **Implement Risk Assessment Policies**: Perform periodic risk assessments to identify and mitigate emerging threats.

2. **Establish Incident Response Procedures**: Develop a response plan for unusual access patterns, including isolating affected systems.

3. **Conduct Employee Training**: Train employees on recognizing and reporting suspicious activities, emphasizing security best practices.