THREAT INTELLIGENCE REPORT.

TASK1

1. TYPES OF ATTACKS

1. SQL Injection (SQLi)  
   SQL Injection is a technique where malicious SQL commands are injected into input fields to interact with the database.  Attackers can access or modify the database in unauthorized ways, leading to potential data theft or alterations.
2. Cross-Site Scripting (XSS)  
   XSS involves injecting harmful scripts into web pages that are then executed in the browsers of other users. This can allow attackers to steal information or perform actions on behalf of users without their consent.
3. Cross-Site Request Forgery (CSRF)  
    CSRF tricks users into unknowingly sending requests to a web application where they are authenticated. Unauthorized actions can be executed on behalf of the user, such as changing settings or making transactions.
4. Remote Code Execution (RCE)  
   RCE attacks enable attackers to execute arbitrary code on a server or user device by exploiting application vulnerabilities. This grants attackers control over the server or device, potentially leading to a wider network breach.
5. Directory Traversal  
   Directory Traversal involves accessing files and directories outside the web server’s designated area by manipulating file paths. Attackers can retrieve sensitive or restricted files that should not be accessible.

2. EXPLOITATION OF VULNERABILITY.  
  
a. How Exploitation Provides Access  
  
When a vulnerability in a web application is exploited, it can grant attackers unauthorized access through various means which may include;

* Privilege Escalation: Attackers might gain higher access levels, allowing them to alter system settings or access confidential data.
* Data Extraction: Exploited vulnerabilities may allow attackers to access and extract sensitive information from databases or file systems.
* Network Access: Once inside the network through a compromised system, attackers can explore and exploit other systems.
* Session Hijacking: Vulnerabilities in session management can enable attackers to take over user sessions, leading to further unauthorized access.

 3. PREVENTIVE MEASURE

a. Regular Software Updates and Patching

* Apply patches and updates regularly to all software, including web applications, servers, and related frameworks. It mitigates the risk of exploitation by fixing known vulnerabilities before attackers can take advantage of them.

b. Implement Web Application Firewalls (WAF)

* Install and configure a WAF to monitor and filter web traffic between users and the application. It helps prevent malicious traffic from reaching the application and blocks potential attacks.

c. Conduct Frequent Vulnerability Scanning and Penetration Testing

* Perform regular scans and tests to identify and rectify security vulnerabilities. It allows early detection and correction of weaknesses before they can be exploited by attackers.

d. Adopt Secure Coding Practices

* Train developers in secure coding practices and review code to ensure security is integrated into the development process. It prevents the introduction of vulnerabilities during development, enhancing overall security.

e. Utilize Multi-Factor Authentication (MFA)

* Implement MFA for access to critical systems and applications. It adds an additional layer of security, making unauthorized access more difficult even if vulnerabilities are exploited.

f. Monitor and Log System and Network Activities

* Continuously track and log network traffic and system activities. This provides early warnings of suspicious activities and aids in incident response and forensic investigations.

g. Provide Employee Security Training

* Offer regular security training and awareness programs for employees. It enhances employees’ ability to recognize and avoid security threats, reducing the risk of social engineering attacks.

By adopting these preventive measures, the organization can strengthen its defenses against similar security breaches and improve overall cybersecurity resilience.

TASK 2.

INCIDENT RESPONSE PLAN.  
This incident response plan outlines a systematic approach for handling and resolving the security breach resulting from an unpatched vulnerability in a web application. It details the steps for containment, eradication, and recovery to address the incident comprehensively.

 1.PREPARATION  
  
- Incident Response Team: Form a dedicated team including IT staff, cybersecurity professionals, legal advisors, communications experts, and executive management.  
- Communication Protocols: Set up clear protocols for both internal and external communications to manage information flow effectively.  
- Resources and Tools: Ensure that necessary tools and resources for monitoring, analyzing, and addressing the breach are readily available.  
  
2. IDENTIFICATION  
  
- Detection: Utilize security alerts and monitoring systems to verify the occurrence of the breach.  
- Assessment: Analyze the breach to determine its scope, including which systems and data have been affected.  
  
 3. CONTAINMENT  
  
a. Immediate Containment  
-Disconnect Affected Systems: Isolate compromised systems from the network to prevent further damage or data loss.  
- Implement Temporary Solutions: Apply immediate fixes or workarounds to limit the breach’s impact while the root cause is being investigated.  
- Increase Surveillance: Enhance monitoring of affected systems and network areas to detect additional malicious activities.  
  
b. Long-Term Containment  
- Apply Security Patches: Install necessary patches or updates to address the vulnerability in the web application.  
- Adjust Access Controls: Reassess and modify access permissions to restrict unauthorized entry.  
- Improve Network Segmentation: Strengthen network segmentation to contain and prevent future incidents.  
  
 4. ERADICATION  
  
- Investigate Root Cause: Conduct a thorough analysis to understand how the vulnerability was exploited and ensure all remnants of the attack are identified.  
- Eliminate Threats: Remove any malicious software, backdoors, or unauthorized changes made by attackers.  
- Verify Systems: Ensure that affected systems are cleaned and returned to a secure state before they are reconnected to the network.  
- Enhance Security: Update security measures and practices to address the vulnerabilities that were exploited.  
  
  
 5. RECOVERY  
  
a. System Restoration  
- Reintegrate Systems: Gradually bring affected systems back into operation, prioritizing critical systems first.  
- Ongoing Monitoring: Continue monitoring for any signs of reoccurrence or additional issues.  
  
b. Communication  
- Internal Updates: Provide regular updates to internal stakeholders about the recovery progress and any procedural changes.  
- External Notifications: Inform affected clients, partners, and regulatory bodies as needed, disclosing the breach details and actions taken.  
  
c. Post-Incident Evaluation  
- Review Meeting: Conduct a debrief with the incident response team to evaluate the incident management and identify areas for improvement.  
- Update Procedures: Revise incident response and security policies based on lessons learned from the breach.  
- Conduct Training: Provide additional training and awareness programs for staff based on the incident’s findings.

 6. Documentation and Reporting  
  
- Incident Record: Maintain comprehensive records of the incident, including timelines, actions taken, and decision-making processes.  
- Final Report: Prepare a detailed report summarizing the breach, response efforts, and outcomes. Share this report with relevant stakeholders and regulatory agencies as necessary.  
  
  
By implementing this incident response plan, the organization can effectively manage the breach, minimize its impact, and ensure a smooth recovery, while also enhancing overall security resilience to prevent future incidents.