**Executive Summary**

Part 1 Overview

The project involves implementing a Security Operations Center (SOC) in a college network to enhance the security posture and protect sensitive information. SOC, which stands for Security Operations Center, is a centralized unit responsible for monitoring, analyzing, and responding to security events and threats. By employing various tools and technologies, the SOC plays a vital role in ensuring the security of an organization's information systems and assets.

The SOC Cycle is a continuous process that consists of several stages. First is the Monitoring stage, where data is collected and analyzed from various sources to identify potential security incidents. Once anomalies or threats are identified, they undergo further investigation in the Detection stage to determine their severity and impact. The gathered information is then analyzed in the Analysis stage to understand the nature of the incident. Appropriate actions are taken in the Response stage to contain and mitigate the incident. A detailed report is generated in the Reporting stage to document the incident, response, and lessons learned. Based on the insights gained, necessary improvements are made in the security posture of the organization in the Improvement stage.

To deploy a SOC in the college network, it is crucial to assess the current security posture and identify any vulnerabilities or gaps. Suitable security tools, including a Security Information and Event Management (SIEM) system and a Malware Information Sharing Platform (MISP), should be implemented and integrated with existing security systems and network infrastructure. The SOC team should be trained on the tools and processes involved, and proper procedures and policies should be established for incident response and reporting. Regular monitoring and analysis of security events should be conducted to proactively detect and respond to potential threats.

SIEM, which stands for Security Information and Event Management, is a software solution that collects and aggregates security event logs from various sources within an organization's network. It provides real-time monitoring, correlation, and alerting capabilities, enabling organizations to identify potential security threats and take proactive measures to prevent or mitigate them. The SIEM Cycle consists of four main stages. In the Collection stage, security event logs and data from various sources are collected and consolidated in a central repository. The collected data is then correlated and analyzed in the Correlation stage to identify patterns, anomalies, and potential security incidents. Once potential threats are identified, alerts and notifications are generated in the Detection stage for further investigation and response. The appropriate action is taken in the Response stage to mitigate the identified security incidents.

MISP, which stands for Malware Information Sharing Platform, is an open-source software solution that facilitates the sharing and analysis of cybersecurity information. It provides a collaborative platform for organizations and security professionals to share threat intelligence, malware samples, and indicators of compromise. MISP helps enhance knowledge about cybersecurity threats and enables collective efforts to combat them.

The college network is a complex infrastructure that connects various devices, systems, and services used by students, faculty, and staff. It stores and processes sensitive information like student records, research data, and intellectual property. Therefore, ensuring the security and integrity of the college network is crucial to protect the confidentiality, availability, and reliability of the information and services provided.

Threat intelligence refers to information and insights about potential cybersecurity threats and risks. It includes data about emerging vulnerabilities, attack techniques, malicious actors, and indicators of compromise. Threat intelligence provides organizations with the knowledge necessary to assess their exposure to various threats and take appropriate measures to mitigate them. It helps identify patterns and trends, enhances incident response capabilities, and enables informed decision-making regarding cybersecurity investments.

Incident response is a structured process of managing and mitigating cybersecurity incidents. It involves detecting, analyzing, containing, eradicating, and recovering from security incidents. A well-defined incident response plan outlines roles, responsibilities, and procedures to be followed during an incident. It includes methods for incident reporting, communication, coordination, and documentation. Effective incident response ensures timely and appropriate actions are taken to minimize the impact of security incidents and restore normal operations.

**2 List of Vulnerability**

|  |  |  |
| --- | --- | --- |
| S.No. | Vulnerability Name | CWE - No |
| 1 | Broken Access Control | CWE-284: Improper Access Control |
| 2 | Cryptographic Failures | CWE-310: Cryptographic Issues |
| 3 | Injection | CWE-564: SQL Injection: Hibernate |
| 4 | Insecure Design | CWE-657: Violation of Secure Design Principles |
| 5 | Security Misconfiguration | CWE 16: Configuration |
| 6 | Vulnerable and Outdated Components | CWE-1395: Dependency on Vulnerable Third-Party Component |
| 7 | Identification and Authentication Failures | CWE-287: Improper Authentication |
| 8 | Software and Data Integrity Failures | CWE-353: Missing Support for Integrity Check |
| 9 | Security Logging and Monitoring Failures | CWE-778: Insufficient Logging |
| 10 | Server-Side Request Forgery (SSRF) | CWE-918: Server-Side Request Forgery (SSRF) |

**REPORT**

**1 Vulnerability Name:-** Improper Access Control

**CWE:-** CWE-284

**OWASP Category:-** A01:2021 –Broken Access Control

**Description:-** The product does not restrict or incorrectly restricts access to a resource from an unauthorized actor.

**Business Impact**:- This weakness allows an attacker to bypass intended security restrictions and perform a variety of actions depending on the source of error and functionality of the application. An attacker might be able to perform certain actions by gaining elevated privileges, reading otherwise restricted information, executing commands, bypassing implemented security mechanisms, etc.

**2** **Vulnerability Name:-** Cryptographic Issues

**CWE : -** CWE-310

**OWASP Category:-** A02:2021 –Cryptographic Failures

**Description:-** Weaknesses in this category are related to the design and implementation of data confidentiality and integrity. Frequently these deal with the use of encoding techniques, encryption libraries, and hashing algorithms. The weaknesses in this category could lead to a degradation of the quality data if they are not addressed.

**Business Impact**:- Cryptographic Failures are a major security problem. They can lead to data breaches, identity theft, and other serious problems. The Open Web Application Security Project (OWASP) has identified ten major failures. These failures can be divided into three categories: Cryptographic design flaws, cryptographic implementation errors and cryptographic key management.

**3** **Vulnerability Name:-** SQL Injection: Hibernate

**CWE : -** CWE-564

**OWASP Category:-** A03:2021 –Injection

**Description:-** Using Hibernate to execute a dynamic SQL statement built with user-controlled input can allow an attacker to modify the statement's meaning or to execute arbitrary SQL commands.

**Business Impact**:- SQL Injection is one of the top 10 web application vulnerabilities. In simple words, SQL Injection means injecting/inserting SQL code in a query via user-inputted data. It can occur in any applications using relational databases like Oracle, MySQL, PostgreSQL and SQL Server. To perform SQL Injection, a malicious user first tries to find a place in the application where he can embed SQL code along with data. It can be the login page of any web application or any other place. So when data embedded with SQL code is received by the application, SQL code will be executed along with the application query.

**4** **Vulnerability Name:-** Violation of Secure Design Principles

**CWE : -** CWE-657

**OWASP Category:-** A04:2021 –Insecure Design

**Description:-** The product violates well-established principles for secure design.

**Business Impact**:- When creating a new application, design is one of the critical aspects of its structure. As applications have become more complex, so have the design structures needed to create a specific product. The primary source of these security concerns is malicious agents, such as hackers, who have caused enormous damage to different applications. Although designing a foolproof security system for our application may be challenging, with sound design principles, we can build a secure application that prevents malicious agents from misusing our application. This doesn't prevent bugs and breaches from these agents. However, it creates a safer application that can tackle most security concerns

**5** **Vulnerability Name:-** Configuration

**CWE : -** CWE 16

**OWASP Category:-** A05:2021 –Security Misconfiguration

**Description:-** Weaknesses in this category are typically introduced during the configuration of the software.

**Business Impact**:- Gartner defines Business Impact Analysis as “the process of determining the criticality of business activities and associated resource requirements to ensure operational resilience and continuity of operations during and after a business disruption. The BIA quantifies the impacts of disruptions on service delivery, risks to service delivery, and recovery time objectives (RTOs) and recovery point objectives (RPOs). These recovery requirements are then used to develop strategies, solutions and plans.”

**6** **Vulnerability Name:-** Dependency on Vulnerable Third-Party Component

**CWE : -** CWE-1395

**OWASP Category:-** A06:2021 –Vulnerable and Outdated Components

**Description:-** The product has a dependency on a third-party component that contains one or more known vulnerabilities.

**Business Impact**:- Rapid growth in third-party dependencies (including open-source libraries, packages and container images, etc.) has significantly changed the modern software development process. Most applications nowadays are built on a combination of in-house and external code. Public open-source repositories offer a place for developers to use, host and share software libraries, packages, container images, and other works with many useful and well-developed features. Although the use of third-party dependencies does improve the development efficiency and quality, the security risks of malicious code and vulnerability has created a backdoor for potential supply chain attacks and other malicious acts. A report released by the Synopsys Cybersecurity Research Center in April 2021 has revealed that over 80% of codebases contained vulnerable open-source component in 2020 [1] and the trend of threat actors to spread malwares through open-source repositories. This in turn shows that with the common use of third-party dependencies, the software development and maintenance process has now become a target of threat actors.

**7** **Vulnerability Name:-** Improper Authentication

**CWE : -** CWE-287

**OWASP Category:-** A07:2021 –Identification and Authentication Failures

**Description:-** When an actor claims to have a given identity, the product does not prove or insufficiently proves that the claim is correct.

**Business Impact**:- The attacker might be able to gain unauthorized access to the application and otherwise restricted areas and perform certain actions, e.g. disclose sensitive information, alter application, or even execute arbitrary code. An attacker can use a variety of vectors to exploit this weakness, including brute-force, session fixation, and Man-in-the-Middle (MitM) attacks.

**8** **Vulnerability Name:-** Missing Support for Integrity Check

**CWE : -** CWE-353

**OWASP Category:-** A08:2021 –Software and Data Integrity Failures

**Description:-** The product uses a transmission protocol that does not include a mechanism for verifying the integrity of the data during transmission, such as a checksum**.**

**Business Impact**:- If integrity check values or "checksums" are omitted from a protocol, there is no way of determining if data has been corrupted in transmission. The lack of checksum functionality in a protocol removes the first application-level check of data that can be used. The end-to-end philosophy of checks states that integrity checks should be performed at the lowest level that they can be completely implemented. Excluding further sanity checks and input validation performed by applications, the protocol's checksum is the most important level of checksum, since it can be performed more completely than at any previous level and takes into account entire messages, as opposed to single packets.

**9** **Vulnerability Name:-** Insufficient Logging

**CWE : -** CWE-778

**OWASP Category:-** A09:2021 –Security Logging and Monitoring Failures

**Description:-** When a security-critical event occurs, the product either does not record the event or omits important details about the event when logging it**.**

**Business Impact**:- Insufficient logging is the most common reason why companies fail to deal with a security breach effectively. Organizations must be equipped by logging the entire activity or it could be difficult for the organization to find the criminal. Not being able to detect at an early stage may further lead to the occurrence of continuous breaches and significant losses. To stay well informed and compliant, taking appropriate measures and having in place loggings and monitoring is essential.

**10** **Vulnerability Name:-** Server-Side Request Forgery (SSRF)

**CWE : -** CWE-918

**OWASP Category:-** A010:2021 –Server-Side Request Forgery (SSRF)

**Description:-** The web server receives a URL or similar request from an upstream component and retrieves the contents of this URL, but it does not sufficiently ensure that the request is being sent to the expected destination.

**Business Impact**:- A successful SSRF attack can often result in unauthorized actions or access to data within the organization, either in the vulnerable application itself or on other back-end systems that the application can communicate with. In some situations, the SSRF vulnerability might allow an attacker to perform arbitrary command execution. An SSRF exploit that causes connections to external third-party systems might result in malicious onward attacks that appear to originate from the organization hosting the vulnerable application.