**AI System That Verifies User Identities Based On Their Online Behavior Patterns, Adding An Extra Layer Of Security**

**Group 7.2**

**Overview :-**

This project proposes the development of an AI system that can verify user identities based on their online behavior patterns. This system would add an extra layer of security to online accounts and services by making it more difficult for fraudsters to gain unauthorized access.

The system would work by collecting and analyzing data about a user's online behavior, such as their browsing history, typing patterns, and purchase history. This data would be used to create a unique behavioral profile for each user. When a user attempts to log in to an account or access a service, the system would compare their current behavior to their behavioral profile to determine if they are a legitimate user.

This system has the potential to significantly improve the security of online accounts and services. It could help to reduce fraud, such as identity theft and account takeovers. It could also make it more convenient for users to log in to accounts and access services without having to enter passwords or other authentication credentials.

However, there are also some potential drawbacks to this system. Some people may be concerned about the privacy implications of the system, as it would be collecting and analyzing a large amount of data about their online behavior. Additionally, the system could be biased against certain groups of people if it is not designed and trained carefully.

Overall, the development of an AI system that can verify user identities based on their online behavior patterns has the potential to significantly improve the security of online accounts and services. However, it is important to carefully consider the potential privacy and bias implications of this technology before it is widely deployed.

**List of teammates –**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.no** | **name** | **college** | **contact** |
| 7.2.1  21BCY10236 | Aaditya Singh Thakur | Vellore Institute of Technology, Bhopal | 9399382977  aadityasingh2021@vitbhopal.ac.in |
| 7.2.2  21BCT0316 | Ayush Aryan | Vellore Institute of Technology, Vellore | 8077003930  ayush.aryan2021@vitstudent.ac.in |
| 7.2.3  21BPS1477 | Jeeshu Das | Vellore Institute of Technology,  Chennai | 8240094560  jeeshu.das2021@vitstudent.ac.in |
| 7.2.4  21BRS1521 | Sangrish Saigiridhar | Vellore Institute of Technology,  Chennai | 7977806850  sangrish.saigiridhar2021@vitstudent.ac.in |

**STAGE - 1**

**List of Vulnerability Table –**

|  |  |  |
| --- | --- | --- |
| **S.no** | **Vulnerability**  **Name** | **CWE - No** |
| 1 | Broken Access Control (BAC) | CWE-284, CWE-287, CWE-288 |
| 2 | Cryptographic Failures (CR) | CWE-338, CWE-326, CWE-327 |
| 3 | Injection (INF) | CWE-78, CWE-89, CWE-86 |
| 4 | Insecure Design (IDS) | CWE-404, CWE-434, CWE-446 |
| 5 | Security Misconfiguration (SCM) | CWE-1194, CWE-1187, CWE-732 |
| 6 | Vulnerable and Outdated Components (VULN) | CWE-252, CWE-276, CWE-9 |
| 7 | Identification and Authentication Failures (FIA) | CWE-287, CWE-286, CWE-288 |
| 8 | Software and Data Integrity Failures (SIF) | CWE-359, CWE-362, CWE-356 |
| 9 | Security Logging and Monitoring Failures (SLM) | CWE-134, CWE-536, CWE-1181 |
| 10 | Server-Side Request Forgery (SSRF) | CWE-918, CWE-306, CWE-310 |

**Vulnerability report of testfire.net**

**Vulnerability Name: Injection**

CWE: CWE-89

OWASP Category: A1:2021 – Injection

Description: Injection occurs when an attacker is able to inject malicious data into an application's input. This data can then be executed by the application, potentially allowing the attacker to take control of the application or steal data.

Business Impact: Injection vulnerabilities can allow attackers to:

• Steal sensitive data, such as user passwords, credit card numbers, and social security numbers.

• Take control of the application or server.

• Disrupt or disable functionality.

Vulnerability Path: http://testfire.net/login.jsp

Vulnerability Parameter: Any user-provided input that is not properly validated is a potential injection vulnerability. For example, the following search query could be used to inject malicious SQL code:

SELECT \* FROM users WHERE username = 'admin' AND password = 'password' OR '1'='1'

Steps to Reproduce:

1. Identify an input field in the application that is not properly validated.

2. Inject malicious data into the input field.

3. Observe the results.

Recommendation:

• Validate all user input.

• Use parameterized queries.

• Implement a web application firewall (WAF).

• Keep your application software up to date.

**Vulnerability Name: Broken Authentication and Session Management**

CWE: CWE-287

OWASP Category: A2:2021 – Broken Authentication and Session Management

Description: Broken authentication and session management vulnerabilities allow attackers to gain unauthorized access to user accounts and sensitive data. These vulnerabilities can be exploited in a variety of ways, such as brute-force attacks, password spraying attacks, and session fixation attacks.

Business Impact: Broken authentication and session management vulnerabilities can allow attackers to:

• Steal sensitive data, such as user passwords, credit card numbers, and social security numbers.

• Take control of user accounts and perform fraudulent transactions.

• Disrupt or disable functionality.

Vulnerability Path: http://testfire.net/login.jsp

Vulnerability Parameter: Any user-provided input that is not properly validated is a potential broken authentication and session management vulnerability. For example, an attacker could use a brute-force attack to crack a user's password if the application does not have a strong password policy in place.

Steps to Reproduce:

1. Identify an authentication or session management mechanism that is not properly implemented.

2. Exploit the vulnerability to gain unauthorized access to a user account or sensitive data.

Recommendation:

• Implement strong authentication and session management controls. This includes using strong passwords, multi-factor authentication, and invalidating sessions after a period of inactivity.

• Keep your application software up to date.

**Vulnerability Name: Sensitive Data Exposure**

CWE: CWE-200

OWASP Category: A3:2021 – Sensitive Data Exposure

Description: Sensitive data exposure vulnerabilities allow attackers to gain unauthorized access to sensitive data, such as user passwords, credit card numbers, and social security numbers. This data can be exposed in a variety of ways, such as through plaintext transmission, insecure storage, and accidental disclosure.

Business Impact: Sensitive data exposure vulnerabilities can allow attackers to:

• Steal sensitive data, such as user passwords, credit card numbers, and social security numbers.

• Commit identity theft and fraud.

• Cause financial losses for businesses and individuals.

Vulnerability Path: http://testfire.net/feedback.jsp

Vulnerability Parameter: Any user-provided input that is not properly validated is a potential sensitive data exposure vulnerability. For example, an attacker could exploit a vulnerability in the application's search form to access sensitive data that is stored in the database.

Steps to Reproduce:

1. Identify a path that allows an attacker to access sensitive data.

2. Exploit the vulnerability to access the sensitive data.

Recommendation:

• Encrypt sensitive data at rest and in transit.

• Implement access controls to restrict who can access sensitive data.

• Keep your application software up to date.

**Vulnerability Name: Insecure Direct Object Reference (IDOR)**

CWE: CWE-328

OWASP Category: A1:2021 – Injection

Description: Insecure Direct Object Reference vulnerabilities occur when an application provides direct access to objects based on user-supplied input. This can allow attackers to bypass authorization and access resources in the system directly, for example database records or files.

Business Impact: Insecure Direct Object Reference vulnerabilities can allow attackers to:

• Steal sensitive data, such as user passwords, credit card numbers, and social security numbers.

• Commit identity theft and fraud.

• Cause financial losses for businesses and individuals.

• Disrupt or deny service to the system.

Vulnerability Path: http://testfire.net/subscribe.jsp

Vulnerability Parameter: Any user-provided input that is used to access objects is a potential Insecure Direct Object Reference vulnerability. For example, an attacker could exploit a vulnerability in the application's login form to access a file belonging to another user.

Steps to Reproduce:

1. Identify a path that allows an attacker to control the input to a function that accesses objects based on user input.

2. Inject malicious input into the input.

3. Trigger the execution of the function.

Recommendation:

• Implement access controls to restrict who can access objects.

• Validate all user-supplied input before using it to access objects.

• Use indirect object references instead of direct object references.

• Keep software up to date with the latest security patches.

**Vulnerability Name: Cross-Site Scripting (XSS)**

CWE: CWE-79

OWASP Category: A7:2021 – Cross-Site Scripting

Description: Cross-site scripting (XSS) is a type of web security vulnerability that occurs when an attacker injects malicious code into a web page. This code can then be executed when the web page is rendered in a user's browser, allowing the attacker to perform actions such as stealing cookies, hijacking sessions, or redirecting users to malicious websites.

Business Impact: XSS vulnerabilities can allow attackers to:

• Inject malicious code into web pages that are viewed by other users.

• Steal cookies or other sensitive information from other users.

• Hijack the web sessions of other users.

• Redirect users to malicious websites.

Vulnerability Path: http://testfire.net/feedback.jsp

Vulnerability Parameter: Any user-provided input that is displayed on a web page without being properly sanitized is a potential XSS vulnerability. For example, an attacker could exploit a vulnerability in the application's search form to inject malicious code into a search query. When the application displays the results of the search query, the malicious code will be executed in the user's browser.

Steps to Reproduce:

1. Identify a path that allows an attacker to inject malicious code into the output of the web application.

2. Inject malicious code into the input.

3. Trigger the execution of the code.

Recommendation:

• Sanitize all user-provided input before displaying it on a web page.

• Use a content security policy (CSP) to restrict the types of scripts that can be executed on a web page.

• Keep software up to date with the latest security patches.

**Vulnerability Name: Security Misconfiguration**

CWE: CWE-200

OWASP Category: A6:2021 – Security Misconfiguration

Description: A security misconfiguration is a weakness in an information system caused by incorrect or incomplete configuration settings. These weaknesses can be exploited by attackers to gain unauthorized access to a system or data, execute arbitrary commands, or disrupt or deny service.

Business Impact: Security misconfiguration vulnerabilities can allow attackers to:

• Gain unauthorized access to systems and data.

• Execute arbitrary commands on systems.

• Disrupt or deny service to systems.

Vulnerability Path: Any misconfiguration of a system or application is a potential security misconfiguration vulnerability. For example, an attacker could exploit a misconfigured database server to gain access to sensitive data.

Vulnerability Parameter: Any configuration setting that is not properly set is a potential security misconfiguration vulnerability. For example, an attacker could exploit a misconfigured web server to execute arbitrary commands on the system.

Steps to Reproduce:

1. Identify a misconfiguration of a system or application.

2. Exploit the misconfiguration to gain unauthorized access or execute arbitrary commands.

Recommendation:

• Follow best practices for configuring systems and applications.

• Keep software up to date with the latest security patches.

• Use a configuration management tool to track and manage configuration settings.

**Vulnerability Name: Insufficient Logging and Monitoring**

CWE: CWE-536

OWASP Category: A10:2021 – Insufficient Logging and Monitoring

Description: Insufficient logging and monitoring can allow attackers to gain unauthorized access to systems and data, execute arbitrary commands, disrupt or deny service, and remain undetected in systems.

Business Impact: Insufficient logging and monitoring can allow attackers to:

• Gain unauthorized access to systems and data.

• Execute arbitrary commands on systems.

• Disrupt or deny service to systems.

• Remain undetected in systems.

Vulnerability Path: Any system or application that does not properly log and monitor activity is a potential target for this vulnerability.

Vulnerability Parameter: Any system or application setting that is not properly configured to log and monitor activity is a potential target for this vulnerability.

Steps to Reproduce:

1. Identify a system or application that does not properly log and monitor activity.

2. Exploit the lack of logging and monitoring to gain unauthorized access, execute arbitrary commands, or disrupt or deny service.

Recommendation:

• Implement a comprehensive logging and monitoring solution to track all system and application activity.

• Review logs regularly to identify and investigate suspicious activity.

• Implement alerting mechanisms to notify administrators of suspicious activity.

**Vulnerability Name: Cross-Site Request Forgery (CSRF)**

CWE: CWE-352

OWASP Category: A5:2021 – Cross-Site Request Forgery (CSRF)

Description: CSRF vulnerabilities allow attackers to trick authenticated users into performing actions that they did not intend to perform. This can be done by forcing the user to click on a malicious link or submit a malicious form.

Business Impact: CSRF vulnerabilities can allow attackers to:

• Steal user credentials and gain unauthorized access to accounts.

• Perform actions on the user's behalf, such as transferring money, posting messages, or deleting files.

• Disrupt or deny service to systems and applications.

Vulnerability Path: Any system or application that does not properly implement CSRF protection is a potential target for this vulnerability.

Vulnerability Parameter: Any user-provided input that is used to perform actions on the user's behalf is a potential target for this vulnerability.

Steps to Reproduce:

1. Identify a system or application that does not properly implement CSRF protection.

2. Craft a malicious link or form that will force the user to perform the desired action.

3. Trick the user into clicking on the malicious link or submitting the malicious form.

Recommendation:

• Implement CSRF tokens to protect against CSRF attacks.

• Use HTTPS to encrypt all communication between the user's browser and the web application.

• Educate users about CSRF attacks and how to avoid them.

**Vulnerability Name: Insecure Deserialization**

CWE: CWE-502

OWASP Category: A8:2021 – Insecure Deserialization

Description: Insecure deserialization vulnerabilities occur when an application deserializes data without properly validating it. This can allow attackers to inject malicious code into the application, which can then be executed.

Business Impact: Insecure deserialization vulnerabilities can allow attackers to:

• Gain unauthorized access to systems and data.

• Execute arbitrary commands on systems.

• Disrupt or deny service to systems.

Vulnerability Path: Any system or application that deserializes data without properly validating it is a potential target for this vulnerability.

Vulnerability Parameter: Any serialized data that is deserialized by the application is a potential target for this vulnerability.

Steps to Reproduce:

1. Identify a system or application that deserializes data without properly validating it.

2. Craft a malicious serialized object that contains malicious code.

3. Inject the malicious serialized object into the application.

4. Trigger the deserialization of the malicious serialized object.

Recommendation:

• Use a secure deserialization library to deserialize data.

• Validate all deserialized data before using it.

• Keep software up to date with the latest security patches.

**Vulnerability Name: Using Known Vulnerable Components**

CWE: CWE-252

OWASP Category: A9:2021 – Using Known Vulnerable Components

Description: Using known vulnerable components can allow attackers to exploit known vulnerabilities in those components.

Business Impact: Using known vulnerable components can allow attackers to:

• Gain unauthorized access to systems and data.

• Execute arbitrary commands on systems.

• Disrupt or deny service to systems.

Vulnerability Path: Any system or application that uses known vulnerable components is a potential target for this vulnerability.

Vulnerability Parameter: Any component that is used by the system or application is a potential target for this vulnerability.

Steps to Reproduce:

1. Identify a system or application that uses a known vulnerable component.

2. Exploit the known vulnerability in the component to gain unauthorized access, execute arbitrary commands, or disrupt or deny service.

Recommendation:

• Keep software up to date with the latest security patches.

• Use a vulnerability scanner to identify and remediate known vulnerabilities.

• Use a software composition analysis (SCA) tool to identify known vulnerable components in your software.

**Main Website vulnerability REPORT**

**Website:** [**https://www.warframe.com/**](https://www.warframe.com/)

**Vulnerability Name: Host Fully Qualified Domain Name (FQDN) Resolution**

CWE: CWE-2009

OWASP Category: A1:2021 – Injection

Description: Host FQDN Resolution is a vulnerability that occurs when an attacker is able to resolve the FQDN of a remote host. This can be done by exploiting vulnerabilities in DNS servers, or by simply performing a reverse lookup on the IP address of the remote host.

Business Impact: Once an attacker knows the FQDN of a remote host, they can use this information to launch a variety of attacks, including:

• Phishing attacks: The attacker can send phishing emails to users that appear to be coming from a legitimate domain.

• Man-in-the-middle attacks: The attacker can intercept traffic between the user and the remote host, and impersonate either the user or the host.

• DNS cache poisoning attacks: The attacker can poison the DNS cache of the user's machine, causing it to resolve the FQDN of the remote host to a malicious IP address.

Vulnerability Path: Any application that accepts user input and then uses that input to resolve the FQDN of a remote host is at risk of this vulnerability. For example, a web application that allows users to search for products by name could be exploited if the application does not properly validate the user input.

Vulnerability Parameter: Any user-provided input that is used to resolve the FQDN of a remote host is a potential vulnerability parameter. For example, the following search query could be used to inject malicious DNS data:

product\_name=foo.com

Steps to Reproduce:

1. Identify an application that accepts user input and then uses that input to resolve the FQDN of a remote host.

2. Inject malicious data into the input field.

3. Observe the results.

Recommendation:

• Validate all user input before using it to resolve the FQDN of a remote host.

• Use a trusted DNS server.

• Implement a web application firewall (WAF).

• Keep application software up to date.

**Vulnerability Name: ICMP Timestamp Request Remote Date Disclosure**

CWE: CWE-2001

OWASP Category: M1:2021 – Insecure Cryptographic Storage

Description: ICMP Timestamp Request Remote Date Disclosure is a vulnerability that occurs when an attacker is able to send an ICMP timestamp request to a remote host and receive a response that includes the date and time set on the remote host. This information can be used by an attacker to defeat time-based authentication protocols.

Business Impact: Once an attacker knows the date and time set on a remote host, they can use this information to launch a variety of attacks, including:

• Brute-force attacks: The attacker can use the date and time information to guess the password for a time-based authentication protocol.

• Replay attacks: The attacker can record a legitimate authentication request and then replay it at a later time.

• Man-in-the-middle attacks: The attacker can intercept and modify the date and time information in an authentication request.

Vulnerability Path: Any system that responds to ICMP timestamp requests is at risk of this vulnerability. This includes most operating systems.

Vulnerability Parameter: The ICMP timestamp request message does not include any authentication information. This means that any attacker can send a timestamp request to any system.

Steps to Reproduce:

1. Send an ICMP timestamp request to a remote host.

2. Receive the ICMP timestamp response message.

3. Observe the date and time information in the response message.

Recommendation:

• Disable ICMP timestamp requests on all systems.

• Use a secure time synchronization protocol, such as NTP.

• Implement a firewall that blocks ICMP timestamp requests. Keep application software up to date.

**Vulnerability Name: TCP Port Scanning**

CWE: CWE-2000

OWASP Category: A1:2021 – Injection

Description: TCP Port Scanning is a vulnerability that occurs when an attacker is able to determine which TCP ports are open on a remote host. This can be done by sending SYN (synchronization) packets to the remote host and observing the response. If the remote host responds with a SYN/ACK (synchronization acknowledged) packet, then the port is open. If the remote host responds with an RST (reset) packet, then the port is closed.

Business Impact: Once an attacker knows which TCP ports are open on a remote host, they can use this information to launch a variety of attacks, including:

• Exploit known vulnerabilities in services running on open ports.

• Launch denial-of-service (DoS) attacks against open ports.

• Gain unauthorized access to the remote host.

Vulnerability Path: Any system that is accessible over TCP is at risk of this vulnerability.

Vulnerability Parameter: The TCP port number is the vulnerability parameter.

Steps to Reproduce:

1. Send a SYN packet to a remote host on a specific TCP port.

2. Observe the response from the remote host.

3. If the remote host responds with a SYN/ACK packet, then the port is open.

4. If the remote host responds with an RST packet, then the port is closed.

Recommendation:

• Use a firewall to block unauthorized access to TCP ports.

• Keep application software up to date.

• Use a network intrusion detection system (NIDS) to monitor for suspicious activity.

**Vulnerability Name: TCP/IP Timestamps Supported**

CWE: CWE-2009

OWASP Category: A1:2021 – Injection

Description: TCP/IP Timestamps Supported is a vulnerability that occurs when a remote host implements TCP timestamps. TCP timestamps are a feature that allows for more accurate round-trip time (RTT) measurements. However, a side effect of this feature is that the uptime of the remote host can sometimes be computed. This can be done by observing the difference between the timestamp values in the SYN and ACK packets.

Business Impact: Once an attacker knows the uptime of a remote host, they can use this information to launch a variety of attacks, including:

• Determine the patch level of the remote host.

• Launch timing-based attacks.

• Correlate information from multiple sources.

Vulnerability Path: Any system that implements TCP timestamps is at risk of this vulnerability.

Vulnerability Parameter: The TCP timestamps option is the vulnerability parameter.

Steps to Reproduce:

1. Send a SYN packet to a remote host with the TCP timestamps option enabled.

2. Receive the SYN/ACK packet from the remote host.

3. Observe the timestamp values in the SYN and ACK packets.

4. Calculate the difference between the timestamp values.

5. Use the difference in timestamp values to estimate the uptime of the remote host.

Recommendation:

• Disable TCP timestamps on all systems.

• Use a firewall to block access to the TCP timestamps option.

• Keep application software up to date.

**Vulnerability Name: Traceroute Information Disclosure**

CWE: CWE-2009

OWASP Category: A2:2021 – Sensitive Data Exposure

Description: Traceroute Information Disclosure is a vulnerability that occurs when an attacker is able to obtain traceroute information for a remote host. Traceroute information can be used to identify the network path between the attacker and the remote host. This information can be used by attackers to launch a variety of attacks, including:

• Mapping the network topology of an organization.

• Identifying potential targets for attack.

• Launching denial-of-service (DoS) attacks.

Business Impact: Once an attacker has obtained traceroute information for a remote host, they can use this information to launch a variety of attacks that can disrupt or disable business operations.

Vulnerability Path: Any system that responds to traceroute requests is at risk of this vulnerability.

Vulnerability Parameter: The traceroute request message does not include any authentication information. This means that any attacker can send a traceroute request to any system. Steps to Reproduce:

1. Send a traceroute request to a remote host.

2. Receive the traceroute response message.

3. Observe the network path information in the response message.

Recommendation:

• Disable traceroute on all systems.

• Use a firewall to block traceroute requests.

• Keep application software up to date.

**STAGE - 2**

Nessus is a highly regarded cybersecurity tool utilized for vulnerability scanning and assessment purposes. Its primary function is to analyze the security status of computer systems, networks, and applications. Nessus is celebrated for its efficiency in pinpointing potential vulnerabilities that could be exploited by malicious actors.

This tool boasts several notable features, including automated vulnerability scanning, support for a vast array of plugins and scripts, customization options for scan policies, and the capability to conduct compliance checks in line with specific standards. Moreover, Nessus excels in prioritizing risks based on severity and potential impact, thereby aiding organizations in addressing critical issues swiftly. After completing scans, Nessus generates comprehensive reports, offering valuable information on identified vulnerabilities, recommendations for remediation, and compliance status.

Nessus is indispensable across various cybersecurity applications. It plays a pivotal role in enhancing network security by identifying weaknesses like open ports, weak passwords, and outdated software configurations. Additionally, it is instrumental in web application security by detecting vulnerabilities that could be exploited by hackers. For organizations striving to meet industry-specific security standards and regulations, Nessus offers robust compliance auditing capabilities. It also aids in the continuous monitoring of systems, ensuring that emerging threats are promptly addressed.

There are different versions of Nessus available, catering to diverse user requirements. The open-source "Nessus Essentials" is a free option that provides fundamental scanning capabilities, while the commercial versions offer advanced features, professional support, and scalability, making them suitable for enterprise-level use.

In relation to the current project, the nessus tool was used to search and scan for any security vulnerabilities present within our testing website’s (119.18.49.84) backend servers.

Target website: sh123.webhostingservices.com

Target IP Address: 119.18.49.84

**Vulnerability scan report:**

1. Name: DNS Server Spoofed Request Amplification DDoS

Severity: HIGH

Plugin: 35450

Description: The DNS Server Spoofed Request Amplification DDoS vulnerability represents a security issue where a Domain Name System (DNS) server can be manipulated to amplify the effects of a Distributed Denial of Service (DDoS) attack. In this scenario, attackers send a large number of DNS queries with a spoofed source IP address to the DNS server. The DNS server, believing the requests are legitimate, responds to the spoofed address with much larger responses, effectively amplifying the DDoS attack. This technique can overwhelm the target network, causing service disruptions or downtime.

Port: This vulnerability typically targets port 53, which is the standard port for DNS traffic.

Solution: To mitigate the risk associated with DNS Server Spoofed Request Amplification DDoS, several steps can be taken:

1. Implement strict access control lists (ACLs) on DNS servers to limit the source IP addresses that can make queries.

2. Configure the DNS server to rate limit responses to avoid excessive amplification.

3. Employ firewalls or intrusion prevention systems (IPS) to filter and block spoofed DNS requests.

4. Keep DNS server software and operating systems up to date with the latest security patches to address vulnerabilities.

Business Impact: The business impact of this vulnerability can be severe. A successful attack can lead to service interruptions, loss of online presence, and potential reputational damage. Customers may experience disruptions in accessing web services, email, or other online resources. Additionally, the business may face increased operational costs to address the aftermath of such an attack, such as implementing stronger security measures and investigating the incident.

2. Name: SSL Medium Strength Cipher Suites Supported (SWEET32)

Severity: HIGH

Plugin: 42873

Description: The "SSL Medium Strength Cipher Suites Supported" vulnerability, also known as SWEET32, refers to a security issue where a web server or service supports weak and outdated cipher suites for SSL/TLS encryption. Cipher suites are combinations of cryptographic algorithms used for secure communication between a client and a server. SWEET32 specifically targets cipher suites that use 3DES (Triple Data Encryption Standard) encryption, which is considered outdated and vulnerable to attacks.

3DES, while once considered secure, is now susceptible to certain cryptographic attacks, making it unsuitable for secure communications. Attackers can exploit this vulnerability to intercept and decrypt SSL/TLS-encrypted traffic protected by these weak cipher suites.

Port: This vulnerability typically pertains to SSL/TLS communication, which often occurs on port 443 for secure web browsing (HTTPS) and various other ports used for secure services.

Solution: To mitigate the risk associated with SSL Medium Strength Cipher Suites Supported (SWEET32), the following steps can be taken:

1. Disable or phase out the use of 3DES-based cipher suites on web servers and services.

2. Upgrade SSL/TLS configurations to use modern, secure cipher suites.

3. Regularly update the SSL/TLS libraries and software to patch vulnerabilities.

4. Continuously monitor and assess SSL/TLS configurations for weaknesses.

Business Impact: The business impact of this vulnerability can be significant. If exploited, attackers can intercept sensitive data transmitted over SSL/TLS-protected connections, such as login credentials, financial information, and other confidential data. This could lead to data breaches, loss of customer trust, regulatory non-compliance, and potential legal repercussions. Remediation measures might require investment in SSL/TLS upgrades and configurations, which can impact operational costs and resources.

3. Name: TLS Version 1.0 Protocol Detection

Severity: MEDIUM

Plugin: 104743

Description: The "TLS Version 1.0 Protocol Detection" vulnerability pertains to the detection of TLS (Transport Layer Security) version 1.0 on a network or a system. TLS 1.0 is an older version of the TLS protocol and is known to have security weaknesses and vulnerabilities. This vulnerability indicates the presence of TLS 1.0 on a system, which is a security concern because it is no longer considered secure due to known vulnerabilities.

TLS 1.0 is susceptible to a range of attacks, including BEAST (Browser Exploit Against SSL/TLS) and POODLE (Padding Oracle On Downgraded Legacy Encryption), making it important to transition to more secure versions of TLS, such as TLS 1.2 or TLS 1.3.

Port: This vulnerability is not specific to a particular port; it relates to the detection of TLS 1.0 usage on a network or system, which can be identified in various network communications.

Solution: To mitigate the risk associated with TLS Version 1.0 Protocol Detection, the following steps can be taken:

1. Disable or phase out the use of TLS 1.0 on servers and network devices.

2. Upgrade to a more secure TLS version, such as TLS 1.2 or TLS 1.3.

3. Ensure that encryption libraries, web servers, and services are configured to use secure TLS versions.

4. Regularly monitor and assess TLS configurations for vulnerabilities.

Business Impact: The business impact of this vulnerability can vary depending on its prevalence and usage within the organization. Continuing to use TLS 1.0 can pose security risks, potentially leading to data breaches, unauthorized access, and other security incidents. The impact may include reputational damage, regulatory non-compliance, and legal consequences. Remediation efforts may involve upgrading TLS implementations, which can impact operational costs and resource allocation.

4. Name: TLS Version 1.1 Protocol Deprecated

Severity: MEDIUM

Plugin: 157288

Description: The "TLS Version 1.1 Protocol Deprecated" vulnerability indicates the presence of TLS (Transport Layer Security) version 1.1 on a network or system. TLS 1.1, while not necessarily insecure, is considered deprecated because it has known vulnerabilities and weaknesses compared to more modern versions of TLS. As a result, using TLS 1.1 is discouraged in favor of newer, more secure versions, such as TLS 1.2 or TLS 1.3.

Deprecated versions like TLS 1.1 may be susceptible to attacks and security vulnerabilities. While not as severely compromised as older versions, it is still advisable to transition to more secure protocols.

Port: This vulnerability is not specific to a particular port. It relates to the presence of TLS 1.1 in network communications, which can be identified in various network connections.

Solution: To mitigate the risk associated with TLS Version 1.1 Protocol Deprecated, consider the following steps:

1. Disable the use of TLS 1.1 on servers and network devices.

2. Upgrade to a more secure TLS version, such as TLS 1.2 or TLS 1.3.

3. Ensure that encryption libraries, web servers, and services are configured to use secure TLS versions.

4. Regularly monitor and assess TLS configurations for vulnerabilities.

Business Impact: The business impact of this vulnerability may not be as severe as using older and more vulnerable TLS versions, but it can still have implications for security. Continued use of TLS 1.1 may expose the organization to certain risks, including the potential for data interception and security breaches. Remediation efforts typically involve upgrading TLS configurations, which may impact operational costs and resource allocation. Addressing this vulnerability is part of an organization's commitment to maintaining a secure and up-to-date cybersecurity posture.

5. Name: SSL Anonymous Cipher Suites Supported

Severity: MEDIUM

Plugin: 31705

Description: The "SSL Anonymous Cipher Suites Supported" vulnerability pertains to the presence of anonymous cipher suites in the SSL/TLS (Secure Sockets Layer/Transport Layer Security) configuration of a server or service. Anonymous cipher suites allow for SSL/TLS connections without client authentication, meaning that a client can connect to the server without verifying its identity. This is a security concern because it opens the door to potential security risks, as attackers can exploit anonymous connections to intercept sensitive data.

Port: This vulnerability is not specific to a particular port but relates to the SSL/TLS configuration of the server. It can impact any port or service that uses SSL/TLS.

Solution: To mitigate the risk associated with SSL Anonymous Cipher Suites Supported, consider the following steps:

1. Disable or remove anonymous cipher suites from the SSL/TLS configuration.

2. Configure SSL/TLS to require client authentication for connections.

3. Regularly review and update SSL/TLS configurations to ensure they are in compliance with best security practices.

Business Impact: The business impact of this vulnerability can be significant. Anonymous cipher suites can allow unauthorized clients to establish connections, potentially leading to data interception and unauthorized access. This can result in data breaches, loss of customer trust, and legal consequences, particularly if sensitive or regulated data is exposed. Remediation efforts may involve adjusting SSL/TLS configurations and monitoring, which can affect operational costs and resources. Addressing this vulnerability is crucial for maintaining secure communications and protecting sensitive information.

6. Name: SSL RC4 Cipher Suites Supported (Bar Mitzvah)

Severity: MEDIUM

Plugin: 65821

Description: The "SSL RC4 Cipher Suites Supported (Bar Mitzvah)" vulnerability identifies the presence of RC4 (Rivest Cipher 4) cipher suites in the SSL/TLS (Secure Sockets Layer/Transport Layer Security) configuration of a server or service. RC4 is a stream cipher that was once widely used for encryption in SSL/TLS protocols. However, RC4 is known to have vulnerabilities that make it insecure, and its use is discouraged.

The "Bar Mitzvah" attack refers to a specific cryptographic attack that can be carried out against SSL/TLS connections using RC4. This attack allows attackers to recover small parts of the plaintext and can be exploited to gain unauthorized access to encrypted data.

Port: This vulnerability is not specific to a particular port but relates to the SSL/TLS configuration of the server. It can impact any port or service that uses SSL/TLS.

Solution: To mitigate the risk associated with SSL RC4 Cipher Suites Supported (Bar Mitzvah), consider the following steps:

1. Disable or remove RC4 cipher suites from the SSL/TLS configuration.

2. Upgrade to more secure cipher suites, such as AES-based ciphers.

3. Ensure that encryption libraries, web servers, and services are configured to use secure cipher suites.

4. Regularly review and update SSL/TLS configurations to align with best security practices.

Business Impact: The business impact of this vulnerability can be significant. The use of RC4 cipher suites exposes the organization to potential attacks that can result in data interception and unauthorized access to encrypted information. This can lead to data breaches, loss of customer trust, regulatory non-compliance, and legal consequences, particularly if sensitive or regulated data is compromised. Remediation efforts may involve adjusting SSL/TLS configurations and monitoring, which can affect operational costs and resource allocation. Addressing this vulnerability is essential for maintaining secure communications and protecting sensitive data.

7. Name: DNS Server Cache Snooping Remote Information Disclosure

Severity: MEDIUM

Plugin: 12217

Description: The "DNS Server Cache Snooping Remote Information Disclosure" vulnerability is related to DNS (Domain Name System) servers and their susceptibility to cache snooping attacks. Cache snooping is a technique where an attacker queries a DNS server to retrieve information about the cached records it contains, potentially revealing sensitive or confidential information.

Cache snooping can provide attackers with insights into the DNS cache, which may include details about visited domains, IP addresses, and network infrastructure. This information disclosure can be exploited for various malicious purposes, such as reconnaissance for targeted attacks.

Port: This vulnerability typically involves DNS traffic on port 53, which is the standard port used for DNS queries.

Solution: To mitigate the risk associated with DNS Server Cache Snooping Remote Information Disclosure, consider the following steps:

1. Implement Access Control Lists (ACLs) to restrict access to DNS servers, allowing queries only from trusted sources.

2. Disable recursive DNS queries on authoritative DNS servers.

3. Monitor and log DNS queries to detect and respond to suspicious activity.

4. Regularly review and update DNS server configurations and security settings.

Business Impact: The business impact of this vulnerability can be significant. A successful cache snooping attack can provide attackers with valuable information about an organization's network and online activities. This information can be used for launching targeted attacks, data breaches, and other malicious activities. The impact may include reputational damage, loss of sensitive data, and potential legal consequences. Remediation efforts may involve strengthening DNS server configurations, monitoring, and access controls to mitigate the risk of cache snooping attacks.

8. Name: SMTP Service Cleartext Login Permitted

Severity: LOW

Plugin: 54582

Description: The "SMTP Service Cleartext Login Permitted" vulnerability indicates that the Simple Mail Transfer Protocol (SMTP) service on a server or system permits cleartext (unencrypted) login for authentication. SMTP is a protocol used for sending and receiving email messages. Allowing cleartext login means that login credentials, such as usernames and passwords, are transmitted over the network in an unencrypted form, making them susceptible to interception by eavesdroppers.

This vulnerability poses a security risk as it can lead to unauthorized access to email accounts and potential exposure of sensitive email contents.

Port: SMTP typically operates on port 25 or 587 for plaintext SMTP, but this vulnerability could affect any port where SMTP services are offered.

Solution: To mitigate the risk associated with SMTP Service Cleartext Login Permitted, consider the following steps:

1. Configure SMTP servers to use secure authentication mechanisms, such as STARTTLS or SSL/TLS for encrypted communication.

2. Disable cleartext authentication methods and enforce the use of secure, encrypted authentication protocols.

3. Implement access controls and strong password policies to enhance email security.

Business Impact: The business impact of this vulnerability is relatively low in terms of severity. However, it still poses a security risk as it can expose email credentials and potentially lead to unauthorized access to email accounts. Unauthorized access to email accounts can result in privacy breaches, unauthorized use of company email resources, and potential exposure of sensitive information. Mitigating this vulnerability helps maintain the confidentiality and integrity of email communications and user accounts.

**STAGE – 3**

**Title: Ability of SOC / SEIM**

In this report, we will explore the ability of Security Operations Centers (SOC) and Security Information and Event Management (SIEM) systems, with a focus on their respective cycles and the role they play in managing security incidents. We will also touch upon Threat Intelligence, Incident Response, and introduce the IBM QRadar tool. Additionally, we will discuss how these concepts apply to your college network and the deployment possibilities. This report will conclude with an overview of the previous stages and provide insights into the future scope of SOC/SEIM.

**SOC:**

A Security Operations Center (SOC) is a centralized unit within an organization that is responsible for monitoring, analyzing, and responding to security incidents. The SOC is equipped with a team of security professionals and advanced tools that help in identifying and mitigating security threats in real-time.

**SOC Cycle:**

The SOC cycle involves several stages, such as continuous monitoring, threat detection, incident investigation, response, and recovery. It operates in a cyclical manner, with the goal of maintaining the security posture of an organization.

**SIEM:**

A Security Information and Event Management (SIEM) system is a comprehensive solution that collects, correlates, and analyzes security data from various sources. It provides a centralized platform for monitoring and managing security incidents.

**SIEM Cycle:**

The SIEM cycle begins with data collection, followed by data analysis, correlation of events, alert generation, and incident response. It plays a crucial role in identifying and responding to security threats effectively.

**MISP:**

Malware Information Sharing Platform & Threat Sharing (MISP) is an open-source threat intelligence platform used to collect, share, and correlate threat information. It helps organizations stay informed about the latest threats.

**Your College Network Information and SOC Deployment:**

To deploy a SOC in your college, you should assess the network's security needs, designate a team of experts, select the right tools, and establish protocols for incident response. A SOC can help protect sensitive data and infrastructure within the college network.

**Threat Intelligence:**

Threat intelligence involves collecting and analyzing data related to potential threats and vulnerabilities. This information is essential for proactive threat detection and mitigation.

**Incident Response:**

Incident response is the process of reacting to a security incident or breach. A well-defined incident response plan ensures that organizations can effectively contain and recover from security breaches.

**QRadar & Understanding About the Tool:**

IBM QRadar is a SIEM tool that provides real-time security monitoring and incident response capabilities. It allows security teams to detect and respond to security threats efficiently.

**Conclusion:**

In the previous stages, we discussed web application testing and analyzed Nessus reports. In this stage, we have explored the fundamental concepts of SOC, SIEM, and QRadar, focusing on their cycles and capabilities.

**Future Scope:**

The future scope for SOC and SIEM includes continuous improvement in threat detection and response capabilities. Organizations will need to stay updated on the evolving threat landscape and invest in advanced technologies and training for their security teams.

**Topics Explored:**

SOC, SIEM, MISP, Threat Intelligence, Incident Response, QRadar

**Tools Explored:**

SIEM and MISP

This report provides a basic understanding of SOC, SIEM, and QRadar, along with their cycles and roles in managing security incidents. It also emphasizes the importance of threat intelligence and incident response in maintaining a secure environment. The future scope suggests a continued focus on improving security measures in the ever-changing threat landscape.