Report Submission for Stage 1

Team 13: Hackstarts

Title: OWASP Top 10 Attacks

1. Overview

We intend to proactively identify the vulnerabilities in a web application with various levels of severity, which, in turn, can shield the application from external intrusions and attacks. Potential vulnerabilities in the application are identified using the Nessus tool.

The severity of the application is bifurcated into categories like mixed, medium, and info with critical, high, and medium impact, respectively. The link for the website is directly provided to the Nessus client for vulnerability assessment. Nessus explores the network where a particular web application is deployed. Nessus provides 3 services:

- Discovery
  - Host discovery
- Vulnerability scanning
  - Basic Network Scan
  - Advanced Scan
  - Advanced dynamic scan
  - Malware scan
  - Web Application Tests
  - Credential Patch Audit
  - Intel AMT Security Bypass
  - Spectre and Meltdown
  - WannaCry Ransomware etc.
- Compliance
  - Audit Cloud Infrastructure
  - Internal PCI network scan
  - MDM Config Audit
  - Offline Config Audit, etc.

Various web applications are tested, and related network issues are identified.

List of Team-mates

Sr No	Name	College	Contact
1	Dr Madhuri Bhavsar	Institute of Technology, Nirma University	Madhuri.bhavsar@nirmauni.ac.in
2	Dr Sharada Valiveti		Sharada.valiveti@nirmauni.ac.in
3	Dr Swati Jain		Swati.jain@nirmauni.ac.in
4	Dr Pooja Shah	Pandit Deendayal Energy University	Dr.pooja.research@gmail.com

Mapping OWASP Top 10 Vulnerabilities with one CWE Number

Sr. No.	Name of the vulnerability	Reference -CWE
1	Broken Access Control	CWE-287: Improper Authentication Weakness
2	Cryptographic Failures	CWE-261: Weak Encoding for Password
3	Injection	CWE-94: Improper control of generation of code
4	Insecure Design	CWE-657: Violation of secure design principles
5	Security Misconfiguration	CWE-20: Improper input validation
6	Vulnerable and Outdated Components	CWE-937: Using components with known vulnerabilities
7	Identification and Authentication Failures	CWE-255: Credentials Management Errors
8	Software and Data Integrity Failures	CWE-353: Missing support for integrity check
9	Security logging and monitoring features	CWE-778: Insufficient logging
10	Server Side Request Forgery	CWE-918: Server Side Request Forgery

REPORT

1. Vulnerability Name: - Broken Access Control

CWE: CWE-287

OWASP/SANS Category: Improper Authentication Weakness

Description:

When an actor claims to have a given identity, the product does not prove or insufficiently proves that the claim is correct. This weakness is caused during the implementation of an architectural security tactic. The code performs authentication with the user-provided username and password. If successful, it sets the logged-in and user cookies to "remember" that the user has already logged in. Finally, the code performs administrator tasks if the logged-in user has the "Administrator" username, as recorded in the user cookie.

Business Impact:

The attack can severely impact the sanctity of the data. Data integrity is compromised, and the attacker can exploit user information, including administrator privileges. Hence, the entire application becomes vulnerable. The administrator can add fake users, edit the data of genuine users, and harm the look and feel by modifying web application images, thereby carrying out various attacks on the network.

2. Vulnerability Name: Cryptographic Failures

CWE: CWE-261

OWASP/SANS Category: Weak Encoding for Password

Description:

Obscuring a password with trivial encoding does not protect it. Password management issues occur when a password is stored in plaintext in an application's properties or configuration file. A programmer can attempt to remedy the problem by obscuring the password with an encoding function, such as base 64 encoding, but this effort does not adequately protect the password.

The following code reads a password from a properties file and uses the password to connect to a database.

```
...

Properties prop = new Properties();

prop.load(new FileInputStream("config.properties"));

String password = Base64.decode(prop.getProperty("password"));

DriverManager.getConnection(url, usr, password);

...
```

This code will run successfully, but anyone with access to config.properties can read the value of password and easily determine that the value has been base 64 encoded. If a devious employee has access to this information, they can use it to break into the system.

Business Impact:

If the passwords are stored in plain text, insider attacks can also happen easily and frequently. The administrator can easily update end users' passwords and manage the critical updates in the database by logging on to the device and updating essential data. This can harm the integrity of the data by managing end users' credentials, greatly violating access control.

3. Vulnerability Name: Injection

CWE: CWE-94

OWASP/SANS Category: Improper control of generation of code

Description:

This CWE entry is at the Base level of abstraction, which is a preferred level of abstraction for mapping to the root causes of vulnerabilities. Obscuring a password with a trivial encoding does not protect the password. Password management issues occur when a password is stored in plaintext in an application's properties or configuration file. A programmer can attempt to remedy the password management problem by obscuring the password with an encoding function, such as base 64 encoding, but this effort does not adequately protect the password. This weakness refers to an incorrect design related to an architectural security tactic.

The following code reads a password from a properties file and uses the password to connect to a database.

```
...

Properties prop = new Properties();

prop.load(new FileInputStream("config.properties"));

String password = Base64.decode(prop.getProperty("password"));

DriverManager.getConnection(url, usr, password);

...
```

This code will run successfully, but anyone with access to config.properties can read the password value and easily determine that it has been base 64 encoded. If a devious employee has access to this information, they can use it to break into the system.

Passwords should be encrypted with keys that are at least 128 bits in length for adequate security.

Business Impact:

The access to config.properties can enable an attacker to read the password value. Later, the value can be decoded from the underlying encoding scheme. The related attack patterns which can be used include Rainbow Table Password Cracking. Exposure to passwords can significantly impact the goodwill of the organisation, contributing to severe business impact.

4. Vulnerability Name: Insecure Design

CWE: CWE-657:

OWASP/SANS Category: Violation of secure design principles

Description:

The product violates well-established principles for secure design. Extended Description: This can introduce resultant weaknesses or make it easier for developers to introduce related weaknesses during implementation. Because code is centered around design, it can be resource-intensive to fix design problems.

Business Impact:

The business impact can be established by understanding the three use cases.

Use Case - 1:

When executable library files are used on web servers, which is common in PHP applications, the developer might perform an access check in any user-facing executable, and omit the access check from the library file itself. By directly requesting the library file (CWE-425), an attacker can bypass this access check.

Use Case - 2:

Single sign-on technology is intended to make it easier for users to access multiple resources or domains without having to authenticate each time. While this is highly convenient for the user and attempts to address problems with psychological acceptability, it also means that a compromise of a user's credentials can provide immediate access to all other resources or domains.

Use Case – 3:

The design of TCP relies on the secrecy of Initial Sequence Numbers (ISNs), as originally covered in CVE-1999-0077. If ISNs can be guessed (due to **predictability, CWE-330) or sniffed (due to lack of encryption during** transmission, CWE-312), then an attacker can hijack or spoof connections. Many TCP implementations have had variations of this problem over the years, including CVE-2004-0641, CVE-2002-1463, CVE-2001-0751, CVE-2001-0328, CVE-2001-0288, CVE-2001-0163, CVE-2001-0162, CVE-2000-0916, and CVE-2000-0328.

5. Vulnerability Name: Security Mis-configuration

CWE: CWE-20

OWASP/SANS Category: Improper Input Validation

Description:

The product receives input or data, but it does not validate or incorrectly validates that the input has the properties that are required to process the data safely and correctly.

Input validation is a frequently-used technique for checking potentially dangerous inputs in order to ensure that the inputs are safe for processing within the code, or when communicating with other components. When software does not validate input properly, an attacker is able to craft the input in a form that is not expected by the rest of the application. This will lead to parts of the system receiving unintended input, which may result in altered control flow, arbitrary control of a resource, or arbitrary code execution.

Input validation is not the only technique for processing input, however. Other techniques attempt to transform potentially-dangerous input into something safe, such as filtering (CWE-790) - which attempts to remove dangerous inputs - or encoding/escaping (CWE-116), which attempts to ensure that the input is not misinterpreted when it is included in output to another component. Other techniques exist as well (see CWE-138 for more examples.)

Input validation can be applied to:

- raw data - strings, numbers, parameters, file contents, etc.
- metadata - information about the raw data, such as headers or size

Data can be simple or structured. Structured data can be composed of many nested layers, composed of combinations of metadata and raw data, with other simple or structured data.

Many properties of raw data or metadata may need to be validated upon entry into the code, such as:

- specified quantities such as size, length, frequency, price, rate, number of operations, time, etc.
- implied or derived quantities, such as the actual size of a file instead of a specified size
- indexes, offsets, or positions into more complex data structures
- symbolic keys or other elements into hash tables, associative arrays, etc.
- well-formedness, i.e. syntactic correctness - compliance with expected syntax
- lexical token correctness - compliance with rules for what is treated as a token
- specified or derived type - the actual type of the input (or what the input appears to be)
- consistency - between individual data elements, between raw data and metadata, between references, etc.
- conformance to domain-specific rules, e.g. business logic
- equivalence - ensuring that equivalent inputs are treated the same
- authenticity, ownership, or other attestations about the input, e.g. a cryptographic signature to prove the source of the data

Implied or derived properties of data must often be calculated or inferred by the code itself. Errors in deriving properties may be considered a contributing factor to improper input validation.

Note that "input validation" has very different meanings to different people, or within different classification schemes. Caution must be used when referencing this CWE entry or mapping to it. For example, some weaknesses might involve inadvertently giving control to an attacker over an input when they should not be able to provide an input at all, but sometimes this is referred to as input validation.

Finally, it is important to emphasize that the distinctions between input validation and output escaping are often blurred, and developers must be careful to understand the difference, including how input validation is not always sufficient to prevent vulnerabilities, especially when less stringent data types must be supported, such as free-form text

Business Impact:

Improper input validation enables an attacker to affect the behavior of an application, resulting in unintended execution flow, data manipulation, or even malicious code execution.

It can be challenging to quantify the impact of improper input validation as it is the initial attack vector for many other vulnerability classes. Improper input validation can lead to SQL injection, OS command injection, cross-site scripting (XSS), denial of service (DoS), buffer overflow, remote code execution (RCE), and many other categories of exploitation.

6. Vulnerability Name: Vulnerable and Outdated Components

CWE: CWE-937

OWASP/SANS Category: Using Components with Known Vulnerabilities

Description:

Using components with known vulnerabilities poses significant risks in software development and cybersecurity. These vulnerabilities, publicly documented in databases like CVE or NVD, can be exploited by attackers to gain unauthorized access, steal data, disrupt services, or deploy malware. This can lead to data breaches, financial losses, service downtime, and legal consequences.

Mitigation Strategies:

1. Regular Updates and Patching: Keep components up-to-date with the latest patches.
2. Vulnerability Management: Continuously identify, assess, and remediate vulnerabilities using tools like OWASP Dependency-Check and Snyk.
3. Secure Development Practices: Integrate security into the software development lifecycle with practices like DevSecOps and regular security testing.
4. Trusted Sources: Use components from reputable vendors or repositories.
5. Monitoring and Response: Monitor for exploitation signs and have an incident response plan.

Business Impact:

Using components with known vulnerabilities can have severe business impacts, affecting multiple aspects of an organization's operations and reputation. Here are the key areas where these impacts are felt:

1. Financial Losses

- **Data Breaches:** Exploitation of vulnerabilities can lead to significant financial losses due to stolen sensitive data, including customer information, intellectual property, and financial records.
- **Regulatory Fines:** Non-compliance with regulations like GDPR, HIPAA, or PCI-DSS can result in hefty fines if vulnerabilities lead to data breaches.
- **Remediation Costs:** Costs associated with patching systems, conducting forensic investigations, and implementing additional security measures can be substantial.

2. Reputation Damage

- **Customer Trust:** A data breach or security incident can erode customer trust and loyalty, leading to loss of business and a tarnished brand image.
- **Market Position:** Competitors may leverage your security failures to gain a competitive advantage.

3. Operational Disruption

- **Service Downtime:** Exploitation of vulnerabilities can cause system outages, interrupting business operations and leading to loss of productivity.
- **Incident Response:** Time and resources spent on incident response and recovery divert attention from core business activities.

4. Legal and Compliance Issues

- **Lawsuits:** Customers and partners affected by security breaches may file lawsuits, leading to legal battles and additional costs.
- **Compliance Violations:** Failing to address known vulnerabilities can result in violations of industry standards and regulations, leading to legal consequences.

5. Loss of Competitive Advantage

- **Innovation Slowdown:** Focusing on remediation and incident response can slow down innovation and product development, impacting the organization's ability to stay competitive.
- **Intellectual Property Theft:** Vulnerabilities can lead to theft of proprietary technologies and trade secrets, compromising competitive advantages.

7.Vulnerability Name - Identification and Authentication Failures

CWE: CWE-255

OWASP Category: Credentials Management Errors

Description:
WE-255 refers to "Credentials Management," which is a category of software weaknesses related to the improper handling of user credentials, such as passwords and tokens. This can include issues like:

1. Insecure Storage: Storing credentials in plaintext or using weak encryption methods, making them vulnerable to unauthorized access.
2. Poor Transmission Security: Transmitting credentials without adequate encryption, exposing them to interception.
3. Insufficient Validation: Failing to properly validate user credentials, which can lead to unauthorized access.
4. Weak Password Policies: Allowing weak or easily guessable passwords.

Proper credential management is crucial to ensure the security of user accounts and sensitive data. Best practices include using strong encryption for storage, enforcing strong password policies, and implementing secure transmission protocols like HTTPS.

Business Impact:
The business impact of CWE-255 (Credentials Management) weaknesses can be significant and may include:

1. Data Breaches: Insecure handling of credentials can lead to unauthorized access, resulting in data breaches that compromise sensitive customer and company information.
2. Financial Loss: Breaches can lead to direct financial losses through fraud, as well as indirect costs from legal fees, regulatory fines, and remediation efforts.

3. Reputational Damage: Trust is critical for businesses. A data breach due to poor credentials management can severely damage a company's reputation, leading to loss of customers and reduced market share.
4. Regulatory Consequences: Many jurisdictions have strict regulations regarding data protection (like GDPR or HIPAA). Non-compliance can result in hefty fines and legal actions.
5. Operational Disruption: Addressing a breach or security incident can divert resources and disrupt normal business operations, affecting productivity and focus.
6. Increased Security Costs: After an incident, companies may need to invest in enhanced security measures, audits, and employee training to prevent future occurrences.

8.Vulnerability Name: A08 - Software and Data Integrity Failures
CWE: CWE-353
OWASP Category: Missing support for integrity check
Description:

CWE-353 refers to "Missing Verification of Critical State." This vulnerability occurs when a system does not adequately verify the state of a critical process, operation, or resource before proceeding with an action. This can lead to unexpected behavior, security issues, or exploitation.

Key points about CWE-353 include:

1. Lack of Checks: The system might fail to confirm whether certain conditions are met (e.g., user permissions, process states) before executing a sensitive operation.
2. Potential Consequences: Missing verification can lead to unauthorized access, data corruption, or other critical failures, as operations may be executed under improper conditions.
3. Example Scenarios: Scenarios include skipping checks on user input before processing transactions, not confirming that a user is still authenticated, or failing to verify that a resource is in a valid state before performing an action.
4. Mitigation: Implementing robust validation and verification checks before critical operations is essential to prevent this weakness, ensuring that systems behave as intended and reducing the risk of exploitation.

Proper verification mechanisms can enhance system integrity and security by ensuring that all necessary conditions are met before proceeding with critical actions.

Business Impact:
The business impact of CWE-353 (Missing Verification of Critical State) can be substantial and may include:

1. Unauthorized Access: Insufficient verification can allow unauthorized users to perform sensitive actions, leading to data breaches and exploitation of resources.
2. Data Integrity Issues: Operations executed without proper state checks may corrupt data or cause inconsistencies, leading to loss of trust in the data's reliability.
3. Financial Loss: Exploitation of this weakness can result in direct financial losses through fraud or theft, as well as increased costs related to incident response and remediation.
4. Regulatory Penalties: Failure to protect critical processes can lead to non-compliance with data protection regulations, resulting in fines and legal repercussions.
5. Reputational Damage: Incidents stemming from this vulnerability can harm a company's reputation, eroding customer trust and loyalty, which can impact revenue.

6. Operational Disruption: Addressing security incidents caused by this weakness can divert resources and disrupt business operations, affecting overall productivity.

9 Vulnerability Name: Security logging and monitoring features

CWE: CWE-778

OWASP Category: Insufficient logging

Description:
CWE-778 refers to "Insufficient Encryption Strength." This weakness occurs when an application uses cryptographic algorithms, protocols, or configurations that do not provide adequate security against modern threats. Here are key points about CWE-778:

1. Weak Algorithms: Utilizing outdated or weak cryptographic algorithms (e.g., MD5, SHA-1) that can be easily broken by attackers.
2. Inadequate Key Length: Using short keys that can be compromised through brute-force attacks.
3. Poor Protocols: Implementing insecure protocols (e.g., SSL 2.0, older versions of TLS) that are vulnerable to attacks.
4. Consequences: Insufficient encryption strength can lead to data breaches, unauthorized data access, and loss of confidentiality.
5. Mitigation: Employing strong, up-to-date cryptographic algorithms and protocols, along with sufficient key lengths, is crucial for maintaining data security.

Proper cryptographic practices help protect sensitive data and maintain the integrity and confidentiality of communications.

Business Impact:
The business impact of CWE-778 (Insufficient Encryption Strength) can be significant, including:

1. Data Breaches: Weak encryption can lead to unauthorized access to sensitive data, resulting in data breaches that compromise customer and company information.
2. Financial Loss: Breaches may incur direct financial losses through fraud, as well as costs associated with legal actions, regulatory fines, and incident response.
3. Regulatory Penalties: Non-compliance with data protection regulations (e.g., GDPR, HIPAA) due to inadequate encryption can lead to substantial fines and legal repercussions.
4. Reputational Damage: A publicized breach can severely damage an organization's reputation, leading to loss of customer trust and decreased market share.
5. Operational Disruption: Addressing security incidents stemming from insufficient encryption can divert resources, disrupt normal business operations, and affect overall productivity.
6. Increased Security Costs: Following a breach, businesses may need to invest in stronger encryption solutions, audits, and employee training to prevent future vulnerabilities.

10 Vulnerability Name: Server Side Request Forgery

CWE: CWE-918:

OWASP Category: 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.

By providing URLs to unexpected hosts or ports, attackers can make it appear that the server is sending the request, possibly bypassing access controls such as firewalls that prevent the attackers from accessing the URLs directly. The server can be used as a proxy to conduct port scanning of hosts in internal networks, use other URLs such as that can access documents on the system (using file://), or use other protocols such as gopher:// or tftp://, which may provide greater control over the contents of requests.

Relationship

This is an unusual category. CWE does not cover the limitations of human processes and procedures that cannot be described in terms of a specific technical weakness as resident in the code, architecture, or configuration of the software. Since "known vulnerabilities" can arise from any kind of weakness, it is not possible to map this OWASP category to other CWE entries, since it would effectively require mapping this category to ALL weaknesses.

Business impact of Server-Side Request Forgery (SSRF), as categorized by OWASP, can be severe, affecting multiple aspects of an organization's operations and reputation. Here are the key areas of impact:

1. Financial Losses

- **Data Breaches**: Exploiting SSRF can lead to unauthorized access to sensitive internal systems, resulting in data breaches. The costs associated with data breaches include regulatory fines, compensation to affected customers, and expenses for breach notification and remediation.
- **Service Downtime**: Attacks leveraging SSRF can disrupt business operations, leading to service outages and loss of productivity. This downtime can directly translate to lost revenue and increased operational costs.

2. Reputation Damage

- **Customer Trust**: Security incidents involving SSRF can erode customer trust, as they highlight vulnerabilities in the organization's infrastructure. Customers may lose confidence in the organization's ability to protect their data, leading to customer attrition.
- **Brand Image**: Publicized SSRF attacks can damage the brand's reputation, making it difficult to attract new customers and partners. The negative publicity associated with security breaches can have long-term effects on the company's market position.

3. Operational Disruption

- **Internal Network Exposure**: SSRF can be used to map and access internal network resources, leading to potential exploitation of other vulnerabilities. This can result in widespread operational disruptions as the organization scrambles to identify and patch affected systems.

- **Incident Response**: Addressing SSRF attacks requires significant resources for incident response, including forensic investigations, patch deployment, and system recovery. This diverts resources from core business activities, impacting overall productivity.

4. Legal and Compliance Issues

- **Regulatory Fines**: Non-compliance with regulations like GDPR, HIPAA, or PCI-DSS due to SSRF-related data breaches can result in substantial fines and legal penalties. Organizations are legally obligated to protect customer data, and failure to do so can have severe financial consequences.
- **Lawsuits**: Affected customers and partners may file lawsuits against the organization, seeking damages for the breach. Legal battles can be costly and time-consuming, further straining the organization's resources.

5. Loss of Competitive Advantage

- **Intellectual Property Theft**: SSRF can be used to access sensitive intellectual property, such as proprietary technologies, trade secrets, and business strategies. Competitors gaining access to this information can erode the organization's competitive advantage.
- **Innovation Slowdown**: The need to focus on security remediation and incident response can slow down innovation and product development, impacting the organization's ability to stay ahead in the market.

Stage 2

Overview :

People use web applications for almost all daily activities, including when to walk, how much to walk, how to walk, what to eat, how to cook, etc. Also, web applications are used for professional, organisational, academic, and domestic purposes. These applications also cater to the users on an individual (personal) basis and recommend upcoming activities on an almost next-access basis. Hence, the applications and the underlying processes are aware of the personal traits of all individuals who use these applications. The person is known in the digital world by the pseudonym called the ApplicationID and the digital world knows the individual through this ApplicationID.

The user presents a lot of personal information to the application through regular usage. Sometimes, the user provides the details manually using fields of the registration forms, search options, etc. Sometimes, the server is equipped with good analytics tools (including machine learning algorithms) to carry out various analytics and provide recommendations to the end user.

Hence, the web application must be very secure. Vulnerable web applications expose individuals' personal information to the outside world. The work presented here aims to restrict the end user by carrying out vulnerability analysis of the web application using Nessus tool.

The severity of the application is bifurcated into categories like mixed, medium, and info with critical, high, and medium impact, respectively. The link for the website is directly provided to the Nessus client for vulnerability assessment. Nessus explores the network where a particular web application is deployed. Nessus provides 3 services:

- Discovery
    - Host discovery
- Vulnerability scanning
    - Basic Network Scan
    - Advanced Scan
    - Advanced dynamic scan
    - Malware scan
    - Web Application Tests
    - Credential Patch Audit
    - Intel AMT Security Bypass
    - Spectre and Meltdown
    - WannaCry Ransomware etc.
- Compliance
    - Audit Cloud Infrastructure
    - Internal PCI network scan
    - MDM Config Audit
    - Offline Config Audit, etc.

Various web applications are tested, and related network issues are identified. For the sake of submission, vulnerabilities of one website are provided:

Note: This document is meant for study purposes only. This work is submitted for the Faculty Development Programme on "Cyber Security" at Nirma University. As part of clearing the course, we were supposed to submit a report on vulnerabilities as provided by the Nessus tool. The authors of this submitted work are not responsible for attacks on the website.

Target website: Can't specify

Target IP Address: x.x.x.x

List of vulnerabilities:

Sr. No.	Vulnerability name	Severity	plugins
1	Apache 2.4.x < 2.4.33 Multiple Vulnerabilities	Critical	122060
2	Apache 2.4.x < 2.4.46 Multiple Vulnerabilities	Critical	139574
3	Apache 2.4.x < 2.4.47 Multiple Vulnerabilities	Critical	150280
4	Apache 2.4.x < 2.4.54 Authentication Bypass	Critical	193421
5	Apache 2.4.x < 2.4.56 Multiple Vulnerabilities	Critical	172186
6	Apache 2.4.x >= 2.4.7 / < 2.4.52 Forward Proxy DoS / SSRF	Critical	156225
7	Apache < 2.4.49 Multiple Vulnerabilities	Critical	153584
8	Apache 2.4.x < 2.4.41 Multiple Vulnerabilities	Critical	128033
9	Apache 2.4.x < 2.4.54 Multiple Vulnerabilities	Critical	161948
10	Apache 2.4.x < 2.4.60 Multiple Vulnerabilities	Critical	201198

REPORT

1

Vulnerability Name: Apache 2.4.x < 2.4.33 Multiple Vulnerabilities

Severity: Critical

Plugin: 122060

Port: 81 / tcp / www

Description:

The version of Apache httpd installed on the remote host is equal to or greater than 2.4.7 and before 2.4.52. It is, therefore, affected by a flaw related to acting as a forward proxy.

A crafted URI sent to httpd configured as a forward proxy (ProxyRequests on) can cause a crash (NULL pointer dereference) or, for configurations mixing forward and reverse proxy declarations, allow requests to be directed to a declared Unix Domain Socket endpoint (Server-Side Request Forgery).

Solution:

The related mitigation is to change the version of the Apache server from 2.4.x to 2.4.33 or later.

Business Impact:

An out-of-bounds write vulnerability exists in mod_authnz_ldap with AuthLDAPCharsetConfig enabled. An unauthenticated, remote attacker can exploit this, via the Accept-Language header value, to cause the application to stop responding. (CVE-2017-15710).

An arbitrary file upload vulnerability exists in the FilesMatch component where a malicious filename can be crafted to match the expression check for a newline character. An unauthenticated, remote attacker can exploit this, via newline character, to upload arbitrary files on the remote host subject to the privileges of the user. (CVE-2017-15715)

Due to the attack, the application may abruptly stop running. Prospective customers will be dissatisfied, and this issue will prevent people from visiting this website. This can lead to a Denial of Service Attack.

2

Vulnerability Name: Apache 2.4.x < 2.4.46 Multiple Vulnerabilities

Severity: Critical

Plugin: 139574

Port: 81 / tcp / www

Description:

The version of Apache httpd installed on the remote host is before 2.4.52. Therefore, it is affected by a flaw related to mod_lua when handling multipart content. A carefully crafted request body can cause a buffer overflow in the mod_lua multipart parser (r: parse body () called from Lua scripts). The Apache httpd team is not aware of an exploit for the vulnerability, though it might be possible to craft one.

The version of Apache httpd installed on the remote host is prior to 2.4.46. It is, therefore, affected by multiple vulnerabilities as referenced in the 2.4.46 advisory.

- Apache HTTP server 2.4.32 to 2.4.44 mod_proxy_uwsgi info disclosure and possible RCE (CVE-2020-11984)
- Apache HTTP Server versions 2.4.20 to 2.4.43. A specially crafted value for the 'Cache-Digest' header in a HTTP/2 request would result in a crash when the server actually tries to HTTP/2 PUSH a resource afterwards.

Solution:

- Apache HTTP Server versions 2.4.20 to 2.4.43 When trace/debug was enabled for the HTTP/2 module and on certain traffic edge patterns, logging statements were made on the wrong connection, causing concurrent use of memory pools. Configuring the LogLevel of mod_http2 above info will mitigate this vulnerability for unpatched servers. (CVE-2020-11993)
- Configuring the HTTP/2 feature via H2Push off will mitigate this vulnerability for unpatched servers. (CVE-2020-9490)
- Upgrade to Apache version 2.4.46 or later.

Business Impact:

- Authenticated attackers with contributor-level access or higher could inject malicious code. This could lead to unauthorised access, data breaches, manipulation of sensitive information, and potentially compromise the overall integrity and confidentiality of the web application. It is crucial to promptly address and patch such vulnerabilities to mitigate the risk of exploitation and protect the web application and its users from potential harm.

3

Vulnerability Name: Apache 2.4.x < 2.4.46 Multiple Vulnerabilities

Severity: Critical

Plugin: 150280

Port: 81 / tcp / www

Description:

The version of Apache httpd installed on the remote host is prior to 2.4.49. It is, therefore, affected by multiple vulnerabilities as referenced in the 2.4.49 changelog.

- ap_escape_quotes() may write beyond the end of a buffer when given malicious input. No included modules pass untrusted data to these functions, but third-party / external modules may. (CVE-2021-39275)
- Malformed requests may cause the server to dereference a NULL pointer. (CVE-2021-34798)

Solution:

Upgrade to Apache version 2.4.49 or later.

Business Impact:

Exploiting this vulnerability could allow attackers to execute arbitrary code or trigger a denial of service (DoS) condition. This could lead to unauthorized access, data breaches, service disruptions, and potential compromise of the web application's availability, confidentiality, and integrity. It is critical to promptly address and patch such vulnerabilities to safeguard the web application and prevent security incidents.

4

Vulnerability Name: Apache 2.4.x < 2.4.54 Authentication Bypass

Severity: Critical

Plugin: 193421

Port: 81 / tcp / www

Description:

The version of Apache httpd installed on the remote host is prior to 2.4.46. It is, therefore, affected by multiple vulnerabilities as referenced in the 2.4.46 advisory.

- Apache HTTP server 2.4.32 to 2.4.44 mod_proxy_uwsgi info disclosure and possible RCE (CVE-2020-11984)
- Apache HTTP Server versions 2.4.20 to 2.4.43 When trace/debug was enabled for the HTTP/2 module and on certain traffic edge patterns, logging statements were made on the wrong connection, causing concurrent use of memory pools.
- Apache HTTP Server versions 2.4.20 to 2.4.43. A specially crafted value for the 'Cache-Digest' header in a HTTP/2 request would result in a crash when the server actually tries to HTTP/2 PUSH a resource afterwards.

Solution:

Configuring the LogLevel of mod_http2 above info will mitigate this vulnerability for unpatched servers. (CVE-2020-11993).

Configuring the HTTP/2 feature via H2Push off will mitigate this vulnerability for unpatched servers. (CVE-2020-9490)

Upgrade to Apache version 2.4.46 or later.

Business Impact:

While specific details about this particular Plugin ID are not readily available, vulnerabilities in web applications can potentially lead to various security risks such as unauthorized access, data breaches, injection attacks, cross-site scripting (XSS), SQL injection, and other forms of cyber threats.

5

Vulnerability Name: Apache 2.4.x < 2.4.56 Multiple Vulnerabilities

Severity: Critical

Plugin: 172186

Port: 81 / tcp / www

Description:

The version of Apache httpd installed on the remote host is prior to 2.4.56. It is, therefore, affected by multiple vulnerabilities as referenced in the 2.4.56 advisory.

- HTTP request splitting with mod_rewrite and mod_proxy: Some mod_proxy configurations on Apache HTTP Server versions 2.4.0 through 2.4.55 allow a HTTP Request Smuggling attack. Configurations are affected when mod_proxy is enabled along with some form of RewriteRule or ProxyPassMatch in which a non-specific pattern matches some portion of the user-supplied request-target (URL) data and is then re-inserted into the proxied request-target using variable substitution. For example, something like: RewriteEngine on RewriteRule ^/here/(.*)

http://example.com:8080/elsewhere?$1 http://example.com:8080/elsewhere ; [P] ProxyPassReverse /here/ http://example.com:8080/ http://example.com:8080/ Request splitting/smuggling could result in bypass of access controls in the proxy server, proxying unintended URLs to existing origin servers, and cache poisoning. Acknowledgements: finder: Lars Krapf of Adobe (CVE-2023-25690)

- Apache HTTP Server: mod_proxy_uwsgi HTTP response splitting: HTTP Response Smuggling vulnerability in Apache HTTP Server via mod_proxy_uwsgi. This issue affects Apache HTTP Server: from 2.4.30 through 2.4.55.
- Special characters in the origin response header can truncate/split the response forwarded to the client.

Solution:

Upgrade to Apache version 2.4.56 or later.

Business Impact:

This plugin is associated with the WordPress content management system (CMS) and is linked to a potential cross-site scripting (XSS) vulnerability. If exploited, attackers could inject malicious scripts into web pages viewed by other users, leading to unauthorized actions, data theft, cookie stealing, session hijacking, and other forms of attacks that compromise the confidentiality and integrity of the web application. It is essential to address and remediate such vulnerabilities promptly to prevent exploitation and protect the web application and its users from potential harm.

6

Vulnerability Name: Apache 2.4.x >= 2.4.7 / < 2.4.52 Forward Proxy DoS / SSRF

Severity: Critical

Plugin: 156225

Port: 81 / tcp / www

Discription:

The version of Apache httpd installed on the remote host is equal to or greater than 2.4.7 and prior to 2.4.52.
It is, therefore, affected by a flaw related to acting as a forward proxy.

A crafted URI sent to httpd configured as a forward proxy (ProxyRequests on) can cause a crash (NULL pointer dereference) or, for configurations mixing forward and reverse proxy declarations, can allow for requests to be directed to a declared Unix Domain Socket endpoint (Server Side Request Forgery).

Business Impact:

The "Apache 2.4.x >= 2.4.7 / < 2.4.52 Forward Proxy DoS / SSRF" vulnerability poses significant risks to businesses by allowing attackers to exploit the affected versions of the Apache HTTP server. This vulnerability can be leveraged to perform Denial of Service (DoS) attacks, rendering critical web services unavailable and disrupting business operations. Additionally, Server-Side Request Forgery (SSRF) attacks can enable malicious actors to make unauthorized requests from the server, potentially accessing internal resources and sensitive data. This can lead to data breaches, loss of customer trust, regulatory fines, and costly remediation efforts. Ensuring that Apache servers are updated to the latest secure version is essential to mitigate these risks and protect the integrity of business operations.

7

Vulnerability Name: Apache 2.4.x >= 2.4.7 / < 2.4.52 Forward Proxy DoS / SSRF

Severity: Critical

Plugin: 153584

Port: 81 / tcp / www

Description:

The version of Apache httpd installed on the remote host is prior to 2.4.49. It is, therefore, affected by multiple vulnerabilities as referenced in the 2.4.49 changelog.

- ap_escape_quotes() may write beyond the end of a buffer when given malicious input. No included modules pass untrusted data to these functions, but third-party / external modules may. (CVE-2021-39275)

- Malformed requests may cause the server to dereference a NULL pointer. (CVE-2021-34798)

Solution:

Upgrade to Apache version 2.4.49 or later.

Business Impact:

The "Apache 2.4.x >= 2.4.7 / < 2.4.52 Forward Proxy DoS / SSRF" vulnerability poses significant risks to businesses relying on Apache web servers. This security flaw can lead to Denial of Service (DoS) and Server-Side Request Forgery (SSRF) attacks, which can disrupt business operations, leading to website downtime and service unavailability. Attackers can exploit this vulnerability to overwhelm the server with malicious requests, rendering it unresponsive. Furthermore, SSRF exploits can manipulate the server into making unauthorized requests to internal systems, potentially leading to data breaches and unauthorized access to sensitive information. The resulting operational disruptions, customer dissatisfaction, and potential regulatory penalties underscore the critical need for businesses to promptly update their Apache servers to mitigate these risks.

8

Vulnerability Name: Apache 2.4.x < 2.4.41 Multiple Vulnerabilities

Severity: Critical

Plugin: 128033

Port: 81 / tcp / www

Description: The version of Apache httpd installed on the remote host is prior to 2.4.41. It is, therefore, affected by multiple vulnerabilities as referenced in the 2.4.41 advisory, including the following:

- A limited cross-site scripting issue was reported affecting the mod_proxy error page. An attacker could cause the link on the error page to be malformed and instead point to a page of their choice. This would only be exploitable where a server was set up with proxying enabled but was misconfigured in such a way that the Proxy Error page was displayed. (CVE-2019-10092)

- HTTP/2 (2.4.20 through 2.4.39) very early pushes, for example configured with H2PushResource, could lead to an overwrite of memory in the pushing request's pool, leading to crashes. The memory copied is that of the configured push link header values, not data supplied by the client. (CVE-2019-10081)

- Some HTTP/2 implementations are vulnerable to unconstrained internal data buffering, potentially leading to a denial of service. The attacker opens the HTTP/2 window so the peer can send without constraint; however, they leave the TCP window closed so the peer cannot actually write (many of) the bytes on the wire. The attacker then sends a stream of requests for a large response object. Depending on how the servers queue the responses, this can consume excess memory, CPU, or both. (CVE-2019-9517)

Solution:

Upgrade to Apache version 2.4.41 or later.

Business Impact:

The vulnerability identified in Apache 2.4.x versions prior to 2.4.41, as highlighted by plugin 128033, poses significant risks to business operations. This vulnerability encompasses multiple issues that can be exploited by malicious actors to execute arbitrary code, cause denial of service, and potentially compromise sensitive data. The business impact of such vulnerabilities includes the potential for substantial financial losses due to system downtime, increased operational costs for incident response and remediation, legal repercussions from data breaches, and damage to the organization's reputation. Additionally, exploitation of these vulnerabilities can undermine customer trust and lead to a loss of business opportunities, making it imperative for organizations to promptly apply security patches and updates to mitigate these risks.

9

Vulnerability Name: Apache 2.4.x < 2.4.54 Multiple Vulnerabilities

Severity: Critical

Plugin: 161948

Port: 81 / tcp / www

Description:

The version of Apache httpd installed on the remote host is prior to 2.4.54. It is, therefore, affected by multiple vulnerabilities as referenced in the 2.4.54 advisory.

- Read beyond bounds via ap_rwrite(): The ap_rwrite() function in Apache HTTP Server 2.4.53 and earlier may read unintended memory if an attacker can cause the server to reflect very large input using ap_rwrite() or ap_rputs(), such as with mod_luas r:puts() function. Acknowledgements: The Apache HTTP Server project would like to thank Ronald Crane (Zippenhop LLC) for reporting this issue (CVE-2022-28614)

- Read beyond bounds in ap_strcmp_match(): Apache HTTP Server 2.4.53 and earlier may crash or disclose information due to a read beyond bounds in ap_strcmp_match() when provided with an extremely large input buffer. While no code distributed with the server can be coerced into such a call, third-party modules or lua scripts that use ap_strcmp_match() may hypothetically be affected. Acknowledgements: The Apache HTTP Server project would like to thank Ronald Crane (Zippenhop LLC) for reporting this issue (CVE-2022-28615)

Note that Nessus has not tested for these issues but has instead relied only on the application's self-reported version number.

Solution:

Upgrade to Apache version 2.4.54 or later.

Business Impact:

The vulnerability identified as "Apache 2.4.x < 2.4.41 Multiple Vulnerabilities" (plugin 161948) poses a significant risk to businesses relying on Apache HTTP Server for their web services. These vulnerabilities, if exploited, can lead to unauthorized access, data breaches, service disruptions, and potential damage to the organization's reputation. Attackers could exploit these flaws to execute arbitrary code, trigger denial-of-service conditions, or access sensitive information. Consequently, businesses may face financial losses due to downtime, remediation costs, and potential legal penalties for failing to protect user data. It underscores the critical importance of timely patch management and regular security updates to mitigate such risks.

10

Vulnerability: Apache 2.4.x < 2.4.60 Multiple Vulnerabilities

Severity: Critical

Plugin: 201198

Port: 81 / tcp / www

Description:

The version of Apache httpd installed on the remote host is prior to 2.4.60. It is, therefore, affected by multiple vulnerabilities as referenced in the 2.4.60 advisory.

- Serving WebSocket protocol upgrades over a HTTP/2 connection could result in a Null Pointer dereference, leading to a crash of the server process, degrading performance. (CVE-2024-36387)

- SSRF in Apache HTTP Server on Windows allows to potentially leak NTML hashes to a malicious server via SSRF and malicious requests or content Users are recommended to upgrade to version 2.4.60 which fixes this issue. Note: Existing configurations that access UNC paths will have to configure new directive UNCList to allow access during request processing. (CVE-2024-38472)

- Encoding problem in mod_proxy in Apache HTTP Server 2.4.59 and earlier allows request URLs with incorrect encoding to be sent to backend services, potentially bypassing authentication via crafted requests. Users are recommended to upgrade to version 2.4.60, which fixes this issue. (CVE-2024-38473)

- Substitution encoding issue in mod_rewrite in Apache HTTP Server 2.4.59 and earlier allows attacker to execute scripts in directories permitted by the configuration but not directly reachable by any URL or source disclosure of scripts meant to only to be executed as CGI. Users are recommended to upgrade to version 2.4.60, which fixes this issue. Some RewriteRules that capture and substitute unsafely will now fail unless rewrite flag UnsafeAllow3F is specified. (CVE-2024-38474)

- Improper escaping of output in mod_rewrite in Apache HTTP Server 2.4.59 and earlier allows an attacker to map URLs to filesystem locations that are permitted to be served by the server but are not intentionally/directly reachable by any URL, resulting in code execution or source code disclosure.
Substitutions in server context that use a backreferences or variables as the first segment of the substitution are affected. Some unsafe RewiteRules will be broken by this change and the rewrite flag UnsafePrefixStat can be used to opt back in once ensuring the substitution is appropriately constrained.
(CVE-2024-38475)

- Vulnerability in core of Apache HTTP Server 2.4.59 and earlier are vulnerably to information disclosure, SSRF or local script execution via backend applications whose response headers are malicious or exploitable. Users are recommended to upgrade to version 2.4.60, which fixes this issue. (CVE-2024-38476)

- null pointer dereference in mod_proxy in Apache HTTP Server 2.4.59 and earlier allows an attacker to crash the server via a malicious request. Users are recommended to upgrade to version 2.4.60, which fixes this issue. (CVE-2024-38477)

- Potential SSRF in mod_rewrite in Apache HTTP Server 2.4.59 and earlier allows an attacker to cause unsafe RewriteRules to unexpectedly setup URL's to be handled by mod_proxy. Users are recommended to upgrade to version 2.4.60, which fixes this issue. (CVE-2024-39573)

Business Impact:

The vulnerability "Apache 2.4.x < 2.4.60 Multiple plugin 201198" has significant implications for businesses relying on Apache web servers. Exploitation of this vulnerability could lead to remote code execution, allowing attackers to gain unauthorized access to sensitive data, disrupt services, or launch further attacks within the network. For businesses, this poses a critical risk to data integrity, customer trust, and operational continuity. Immediate patching and proactive security measures are essential to mitigate these risks and safeguard against potential financial and reputational damage stemming from exploitation of this vulnerability.

Stage 3

Report

Title: A five-day journey into the world of Cyber Security: Basic Concepts to Advanced Tools and Technologies

SOC

SOC refers to a Security Operations Center. A SOC improves an organization's threat detection, response, and prevention capabilities by unifying and coordinating all cybersecurity technologies and operations. The team that monitors the organization’s IT Infrastructure 24/7 is either in-house or outsourced. The goal of a SOC is to detect, analyze, and respond to security incidents in real-time. Hence, the SOC team needs to maintain vigilance over the organization’s networks, systems, and applications and ensure a proactive defence posture against cyber threats.

Following are the 3 general categories of SOC activities:

1. Preparation, planning and prevention
    a. Asset Inventory: SOC maintains an exhaustive inventory of everything that needs to be protected inside or outside the data center.
    b. Route maintenance and preparation: SOC performs preventive maintenance such as applying software patches or upgrades, allow-lists and block-lists, security policies and procedures.
    c. Incident Response Planning: SOC develops organization’s response plan, which defines activities, roles, and responsibilities in the event of a threat or an incident, and the metrics by which the success of any incident response will be measured.
    d. Regular testing: SOC performs vulnerability assessment and penetration tests.
    e. Stating current: SOC stays up-to-date concerning the latest security solutions and technologies.
2. Monitoring, detection and response
    a. Continuous, around-the-clock monitoring: SOC monitors applications, servers, systems, computing devices, cloud workloads, and the network for any signs of exploits. The core monitoring, detection, and response technology for SOC is done by Security Information and Event Management (SIEM). SIEM monitors and aggregates real-time alerts and telemetry from software and hardware on the network and analyses the network to identify potential threats.
    b. Log management: The organisation collects logs to establish a network performance baseline. If this baseline is violated by a predefined threshold, an anomaly or suspicious activity is detected.
    c. Threat detection: The SOC team sorts the signals from the noise—the indications of actual cyber threats and hacker uses from the false positives—and then triages the threats by severity. Modern SIEM solutions include artificial intelligence (AI) that automates these processes and 'learns' from the data to improve its ability to spot suspicious activity over time.
    d. Incident Response: Actions under the incident response include –
        i. Root cause investigation
        ii. Shutting down compromised systems and disconnecting them from the network
        iii. Isolating compromised areas of the network and rerouting network traffic
        iv. Pausing or stopping compromised processes

    v. Deleting damaged or infected files
    vi. Running antivirus or antimalware software
    vii. Decommissioning passwords for internal and external users

3. Recovery, refinement and compliance
    a. Recovery and remediation: SOC attempts to recover the impacted assets to the state before the incident. In the event of data breach or ransomware attack, recovery might also involve cutting over to backup systems, and resetting passwords and authentication credentials.
    b. Post-mortem and refinement: To prevent a recurrence, the SOC uses any new intelligence gained from the incident to address vulnerabilities, update processes and policies better, choose new cybersecurity tools or revise the incident response plan. At a higher level, SOC team might also try to determine whether the incident reveals a new or changing cybersecurity trend for which the team needs to prepare.
    c. Compliance management: All applications, systems and tools must comply with the data privacy regulations like GDPR (Global Data Protection Regulation), CCPA (California Consumer Privacy Act), PCI DSS (Payment Card Industry Data Security Standard) and HIPAA (Health Insurance Portability and Accountability Act). SOC ensures that the users, regulators, law enforcement and other parties are notified following regulations and that the required incident data is retained for evidence and auditing.

Benefits of SOC include:

- Asset protection
- Business continuity
- Regulatory compliance
- Cost savings
- Customer trust
- Enhanced incident response
- Improved risk management
- Proactive threat detection

SOC Team comprises of

- SOC Manager – oversees all security operations
- Security Engineers – build and manage organization’s security architecture
- Security Analysts – detect, investigate and triage threats
- Threat hunters – detect and contain advanced threats

SOC Cycle

Figure 1 shows the SOC cycle.

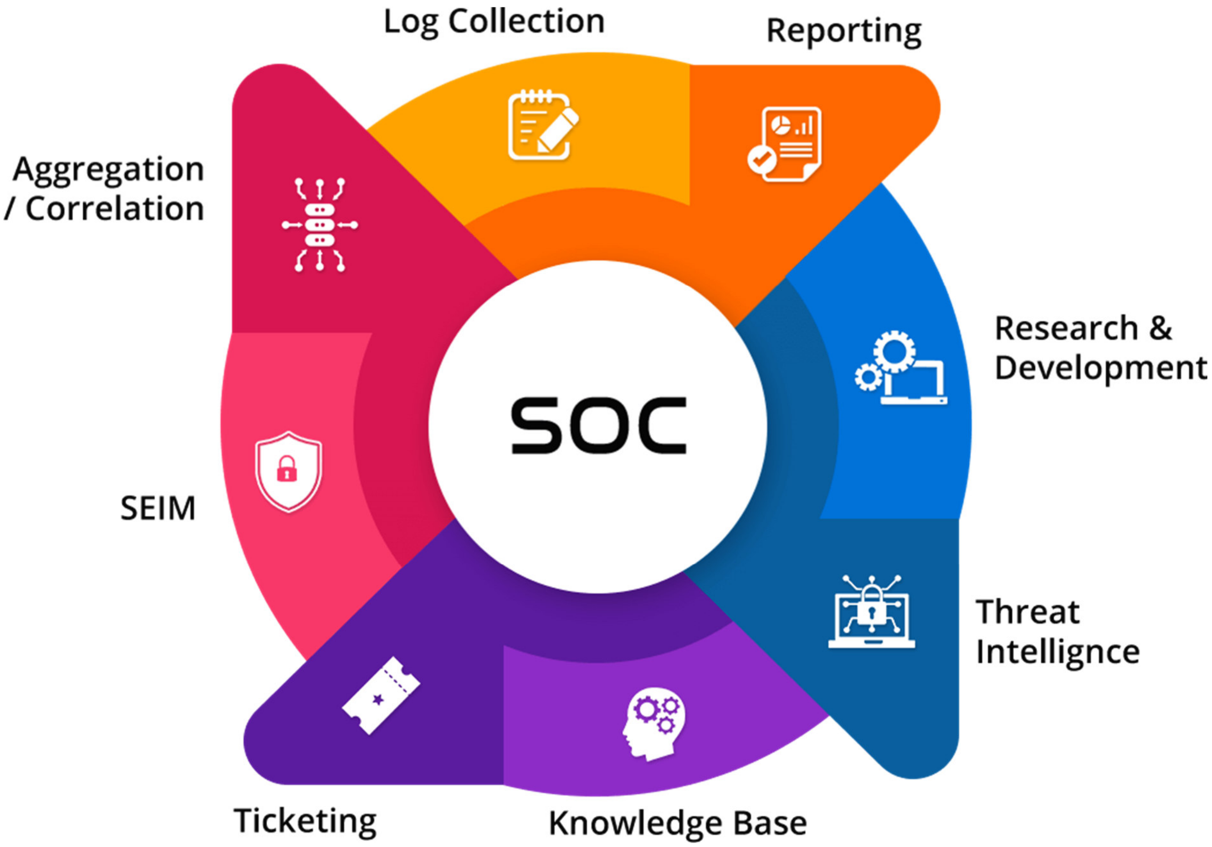

Figure 1: SOC Cycle

A well-managed Security Operations Center (SOC) is the brains of an effective cybersecurity programme. SOCs act as a center of company-wide detection and response capabilities for those entrusted with preventing cyber threats within their firm.

Functions of the SOC Cycle include –

- Active monitoring and analysis of all integrated systems
- Detection of IT vulnerabilities
- Checking Compliance
- Central management of all integrated devices
- Notifies you about attacks and threats
- Defensive measures to limit damage
- Security Assessments
- Detailed reporting

SIEM

Security information and event management, or SIEM, is a security solution that helps organizations recognize and address potential security threats and vulnerabilities before they have a chance to disrupt business operations.

At the most basic level, all SIEM solutions perform some level of data aggregation, consolidation and sorting functions to identify threats and adhere to data compliance requirements. While some solutions vary in capability, most offer the same core set of functions:

- Log Management: SIEM ingests a huge amount of event data from various sources across an organization’s IT Infrastructure, including on-premises and on-cloud environments. Event log data from endpoints, applications, data sources, cloud workloads and networks, and security hardware and software data is collected and analysed in real-time.
- Event Correlation and Analytics: Analytics can be used to identify and understand intricate data patterns, and event correlation provides insights to locate and mitigate potential threats to business security quickly.
- Incident Monitoring and Security Alerts: SIEM consolidates its analysis into a single, central dashboard where security teams monitor activity, triage alerts, identify threats and initiate response or remediation. The SIEM dashboards also include real-time visualisations that help security analysts spot spikes or trends in suspicious activity. Using customisable, predefined correlation rules, administrators can be alerted immediately and take appropriate actions to mitigate threats before they materialise into significant security issues.
- Compliance Management and Reporting: SIEM solutions are a popular choice for organizations subject to different forms of regulatory compliance. Due to automated data collection and analysis that it provides, SIEM is a valuable tool for gathering and verifying compliance data across the entire business infrastructure.

SIEM Cycle

SIEM Systems continuously monitor various data sources, including logs, events, network traffic, and system activity, to gather information about an organization's IT infrastructure's security posture. Monitoring involves collecting, analyzing, and identifying potential security incidents or anomalies.

Figure 2 shows the SIEM Cycle.

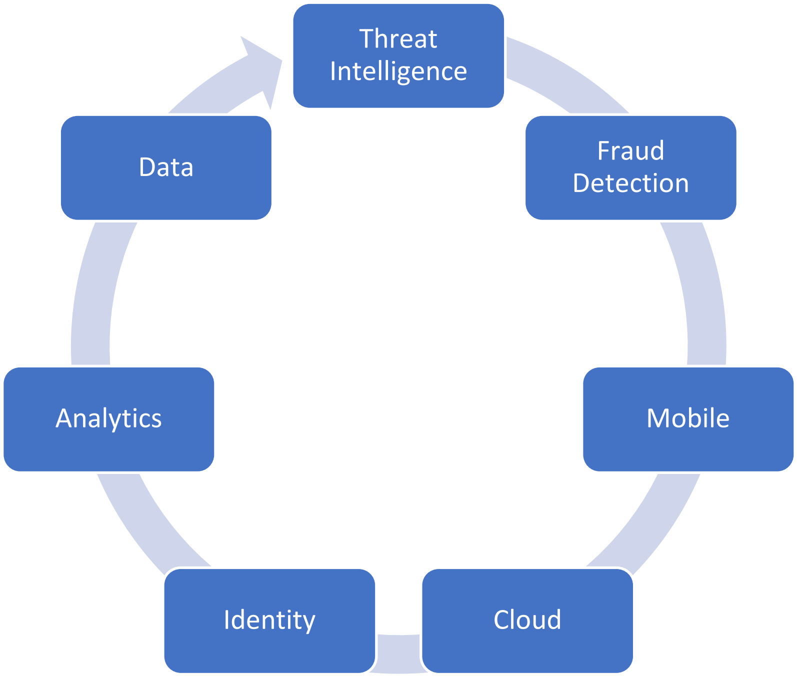

Figure 2: SIEM Cycle

Kibana tool is used in SOC to perform SIEM.

Elastic search is used to perform indexing of events –

- ElastAlert generates alerts
- Cortex is the brain of the search
- Nessus creates the report
- Graphs will be visible in the dashboard

MISP

MISP is the open source threat intelligence and sharing platform. It provides visualizations and dashboards. MISP comes with many visualization options so that analysts can easily view what they are looking for.

Figure 3 shows the interaction with MISP:

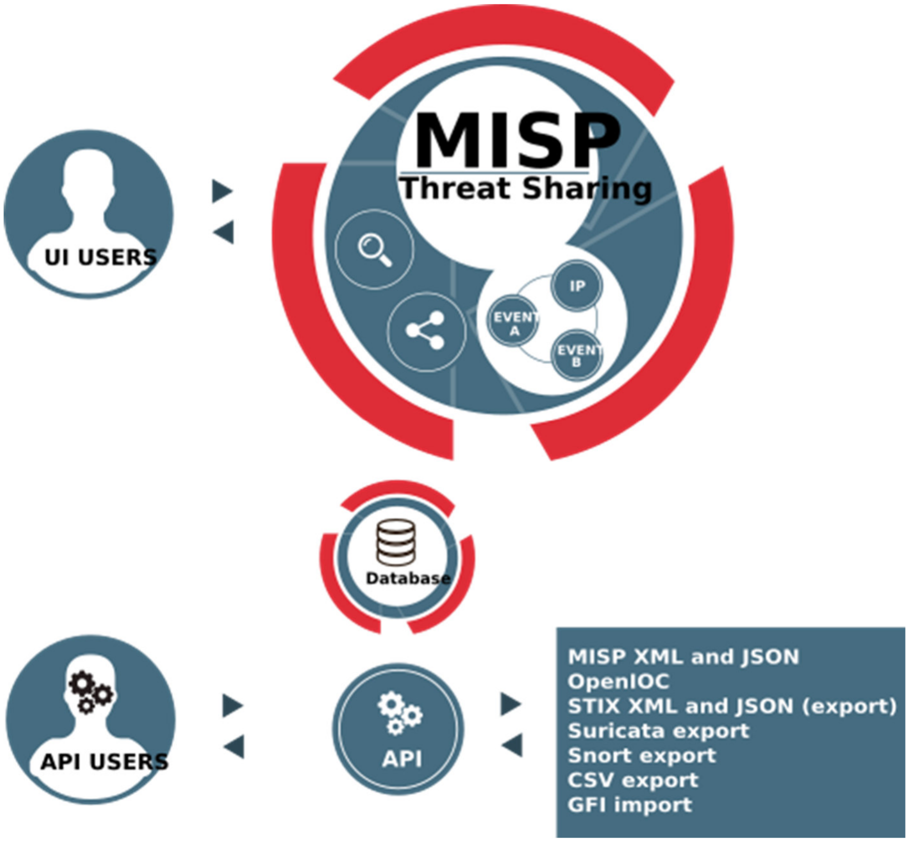

Figure 3: MISP Threat Sharing

Features of MISP:

- Efficient IoC and indicators
- Correlation
- Flexible data model to express threat intelligence, incidents or connected elements
- Built-in sharing functionality
- Intuitive user interface
- Storing data in a structured format
- Export output of the IDS, OpenIOC, plain text, CSV, MISP XML or JSON to integrate with other systems
- Import in bulk, batch, free-text, OpenIOC, GFI Sandbox, ThreatConnect CSV or MISP format
- Flexible free text import tool

- Collaborate on events and attributes allowing MISP users to propose changes or updates to attributes/indicators
- Flexible API to integrate MISP with own solutions
- Adjustable taxonomy to classify and tag events following own classification schemes or existing taxonomies
- Intelligence vocabularies called MISP galaxy and bundled with existing threat actors, malware, RAT, ransomware or MITRE ATT&CK which can be linked easily with events in MISP
- Expansion modules in python
- Sighting support to get observations from organizations concerning shared indicators and attributes
- Export data in STIX format (XML and JSON) including export/import in STIX 2.0 format
- Integrated encryption and signing of the notifications via PGP and/or S/MIME depending on user preferences
- Real-time publish-subscribe channel with MISP to automatically get all changes in ZMQ or Kafka

College Network Information

The university's network is managed by the IT administration cell. The university has many buildings (blocks). The campus includes academic buildings, administrative wings, hostels, canteens, etc. The university's network is flat. There is no hierarchy. Different network IP addresses are provided to LAN and Wi-Fi networks.

All buildings are interconnected by fiber optic cables to a central switch, which in turn is connected to the Internet through a router and firewall. All incoming and outgoing traffic is routed through the firewall. Logically the network is divided into several VLANs for different network sections. Computer laboratories are available in each block of the university, as per the requirements of the respective institutions.

These computer laboratories are connected through high-speed networks. All students, staff members and faculty members are provided with network connectivity on the University laptops (given to faculty members) or laboratory PCs (staff members and students for laboratory work) or in the personal laptops (student laptops).

The systems have Operating systems like Windows, Linux, Ubuntu, and MAC, as per the laboratory's requirements. Various categories of laboratories are provided as per the course requirements. The lab staff ensure the availability of the network and required hardware/software in the respective laboratory venues.

Deployment of SOC in Nirma University

Students are given network access on demand for the duration of their study at Nirma University. If they wish to access the university network, they are asked to provide the hardware address of their respective laptops. This hardware address is stored in the university's firewall. The hardware address is mapped to the Wi-Fi IP address as provided by the network administrator. The student also stores the IP address in his system. The following are the reasons for doing so:

1. Laptop can connect with the network.
2. The duration for which the laptop can connect to the network is also fixed.
3. The number of machines which can connect to the network is also known to the network administrator.
4. All students will have to connect to the network, provide their credentials and log on to the network
5. IT Policy has been defined for the network. All users will be permitted to access the internet through the underlying permissions or prohibitions to access the network.
6. Since users log on to the system, the firewall identifies them as students and checks for access permissions for each query they post.
7. Sr No 5 and 6 are also applicable for each access either by staff members or faculty members.
8. All systems are equipped with antivirus, windows defender, etc.

Threat Intelligence

Figure 4 shows the threat intelligence life cycle.

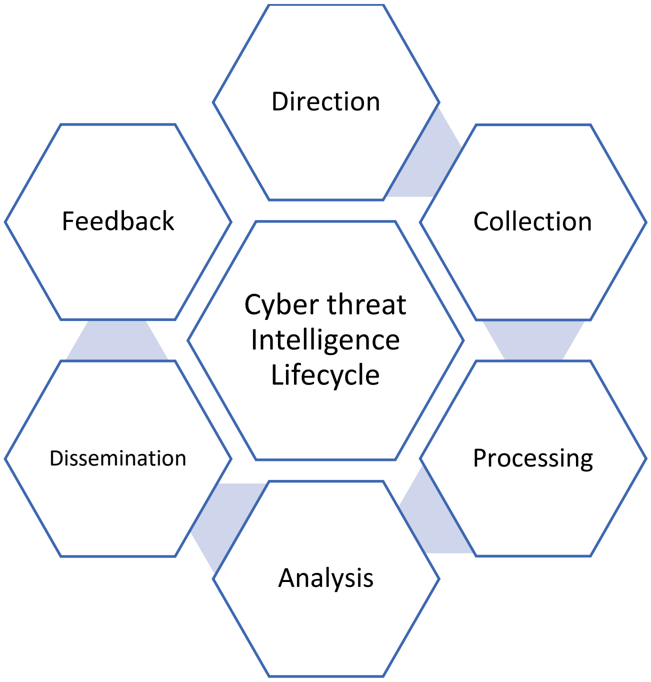

Figure 4: Threat Intelligence Lifecycle

Incident Response

Incident response follows predefined incident response procedures and playbooks. SOC team investigates and contains incidents, minimizing impact and restoring normal operations as quickly as possible.

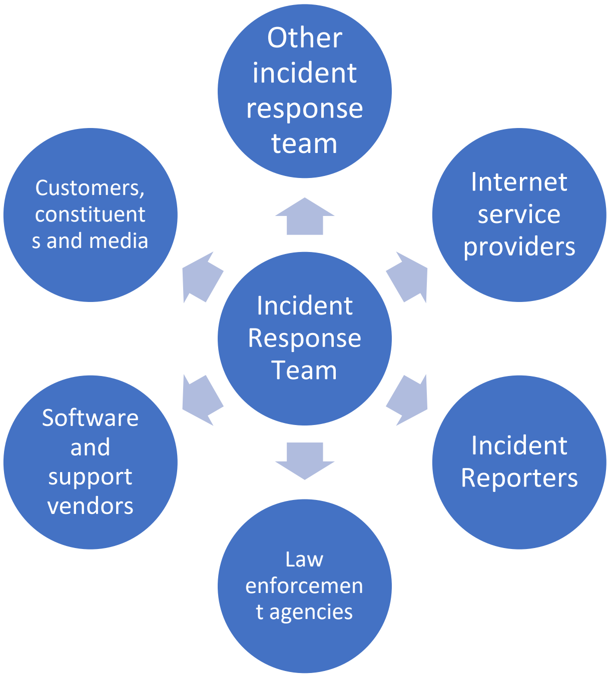
Other incident response team
Customers, constituent s and media
Internet service providers
Incident Response Team
Software and support vendors
Incident Reporters
Law enforcemen t agencies

Figure 5: Incident Response Cycle

Q-Radar and Understanding about Tool

IBM has implemented SIEM using Q-Radar. QRadar is used for incident detection and investigation. Also SIEM Administration can be performed using QRadar. QRadar can be used to carry out the following activities:

- Identify scan targets and create scan scope – QRadar seeks inputs from IT Team to identify servers, networks, owners and IPs for identifying the scan targets. Ideally, all servers in an organization must be scanned. The plans can be scheduled on the network basis. One network can be scanned per phase.
- Configure scans and run scans: The organization and IT team can decide on the type of scan required from the various scan policy types, such as discovery scan, patch scan, database scan, web scan, PCI scan, and full scan. Black box or white box testing can be carried out for unauthenticated (external point of view) or authenticated (internal point of view) scans, respectively. Properties of scans like scan policy determination, target assets, bandwidth limit, scan server, notifications, and schedule need to be finalized.
- Understand results: Main vulnerability properties include name/description, assets, CVE ID, CVSS Metric, Impact and Patches. This feature enables the IT Team to present the report to the management. Main vulnerability reports include scan overview, missing patches, monthly vulnerabilities overview, PCI Compliance Failures.
- Prioritize vulnerabilities: Identify the vulnerabilities in the underlying organization network and define the priorities of these identified vulnerabilities.
- Create a remediation plan: Assign vulnerabilities to technical user, set the due date and write a note about the reason for assignment. Configure group of assets in IBM QRadar to automatically assign their vulnerabilities to technical users. In IBM QRadar Vulnerability Manager you can configure the remediation times for different times for different types of vulnerabilities.
- Investigate vulnerabilities and related incidents: Understand the details of vulnerabilities and attack vectors like affected servers, ports / services, authentication requirement, affect of vulnerability, availability of exploits, mitigation and exploitation evidence.

Conclusions:

1. **Understanding from the Web application Testing**

   Web application testing follows the various phases of hacking as shown below:

   a. *Reconnaissance* means gathering information
   b. *Scanning* using tools like Nessus and Acunetix tools
   c. *Gaining access* to access the hardware for the intended outcome
   d. *Maintaining access* by installing backdoors to violate Confidentiality, Integrity and Availability (CIA Traid)
   e. *Clearing the tracks* implies that the hacker would delete the history or logs

   Cyberattacks can be categorised into Active and Passive attacks. Active attacks include Man-in-the-middle attacks, Distributed Denial of Service attacks (DDoS), Denial of Service attacks (DoS), Packet Sniffing, ARP Spoofing, DNS Spoofing, Session hijacking, software-based attacks, exploitation of vulnerabilities, SQL Injection, etc. Passive attacks include computer surveillance, network surveillance, wiretapping, traffic analysis, cryptanalysis, and brute-force attacks.

Hackers are of various categories, such as black, white, and grey hat hackers.

We discussed in detail malware, IP addresses, phishing, vishing, firewalls, social engineering, ransomware, virtual private networks, pen testing, and antivirus. Exploits from publicly available exploit databases like Exploit DB and Malware DB are available.

Various types of vulnerabilities can be present in the web application:

a. *Software* vulnerabilities include buffer overflow, input validation, insecure cryptographic implementation, insecure deserialisation, etc.
b. *Network* vulnerabilities include determining open ports and services, weak authentication mechanisms, misconfigured firewalls and ACLs, man-in-the-middle attacks, etc.
c. *Human-related* vulnerabilities include social engineering, insider threats, lack of security awareness, and unsecured facilities.
d. *Configuration management* vulnerabilities include weak configuration management and poor change management practices

Demonstration of OSINT framework was discussed, and asked to determine if the email address has been pwned.

Use of Artificial Intelligence for fraud detection, malware detection, network security and authentication mechanisms were discussed. It was informed that use of multimodal analytics, predictive maintenance, zero-day vulnerability detection are advantages of deep learning in cybersecurity. Challenges like hardware requirements, data collection and quality, complexity and interpretability, availability of clean data, continuously evolving cyber threats, etc. continue to exist. Discussions about supervised, unsupervised, and reinforcement learning were also essential highlights of the session.

Basically web applications can be divided into two types like static and dynamic. Static applications do not have database integration while dynamic web applications have integration of database. There are three categories of users in general, such as membership users, e-catalogue users, online forum users, etc. It was informed that the data which we access on website is only 5% of the entire data. 70% data is available in deep web and dark web comprises of 25% data approximately.

Web application architecture was also presented with the following modules:

1. Databases
2. Caching services
3. Job queue
4. Full text search service
5. Services
6. Data warehouse
7. Cloud storage
8. CDN

Various types of web application risks were discussed as under:

- Injection flaws
- Broken authentication
- Sensitive data exposure
- XML External Entity
- Broken Access control
- Security Misconfiguration
- Cross-site scripting
- Insecure deserialization
- Using components with known vulnerabilities
- Insufficient logging and monitoring
- Other threats include directory traversal, cookie snooping, buffer overflows, captcha attacks, DNS rebinding attacks, clickjacking, DMZ protocol attack, obfuscation attacks, network access attacks, platform exploits

Web Application Hacking Methodology include –

- Footprint web infrastructure
- Analyze web application
- Bypass client-side controls
- Attack authentication mechanism
- Attack authorization schemes
- Attack access controls
- Attack session management mechanism
- Perform injection attacks
- Attack application logic flaws
- Attack shared environments
- Attack DB connectivity

- Attack web application client

  OWASP Top 10 attacks (2021) are also discussed:

  A01 - Broken Access Control

  A02 - Cryptographic Failures

  A03 - Injection

  A04 - Insecure Design

  A05 - Security Misconfiguration

  A06 - Vulnerable and Outdated Components

  A07 - Identification and Authentication Failures

  A08 - Software and Data Integrity Failures

  A09 - Security logging and monitoring features

  A10 - Server Side Request Forgery

  Working of sqlmap tool to identify names of databases for a particular website, identify tables of a particular database and extracting details from a particular table was also demonstrated and experimented with.

2. Understanding from the Nessus Report

From the information available, the Nessus Report seems to be a detailed breakdown of vulnerabilities identified through a vulnerability scan conducted using Nessus, a widely used vulnerability scanning tool. The report typically categorizes vulnerabilities based on their severity levels, providing insights into potential security risks within a system or network. By leveraging this report effectively, organizations can prioritize and address vulnerabilities to enhance their overall cybersecurity posture.

The user presents a lot of personal information to the application through regular usage. Sometimes, the user provides the details manually using fields of the registration forms, search options, etc. Sometimes, the server is equipped with good analytics tools (including machine learning algorithms) to carry out various analytics and provide recommendations to the end user.

Hence, the web application must be very secure. Vulnerable web applications expose individuals' personal information to the outside world. The work presented here aims to restrict the end user by carrying out vulnerability analysis of the web application using Nessus tool.

The severity of the application is bifurcated into categories like mixed, medium, and info with critical, high, and medium impact, respectively. The link for the website is directly provided to the Nessus client for vulnerability assessment. Nessus explores the network where a particular web application is deployed. Nessus provides 3 services:

- Host discovery
- Vulnerability scanning
    - Basic Network Scan
    - Advanced Scan
    - Advanced dynamic scan
    - Malware scan
    - Web Application Tests
    - Credential Patch Audit
    - Intel AMT Security Bypass
    - Spectre and Meltdown
    - WannaCry Ransomware etc.
- Compliance
    - Audit Cloud Infrastructure
    - Internal PCI network scan
    - MDM Config Audit
    - Offline Config Audit, etc.

Various web applications are tested, and related network issues are identified. For the sake of submission, vulnerabilities of one website are provided. The same is discussed in detail in Stage 2 submission.

3. **Understanding from the SOC/SEIM/QRadar Dashboard**

Using the dashboard tab of the QRadar, one can focus on specific areas of network security. The workspace supports multiple dashboards on which views of network, activity and data can be displayed. Following customizations are possible:

1. Add and remove dashboard items.
2. Move and position items to meet your requirements.
3. Add custom dashboard items that are based on any data.

To create custom items, you can create saved searches on the Log Activity tab and choose how you want the results that are represented in your dashboard. Each dashboard chart displays real-time up-to-the-minute data. Time series graphs on the dashboard refresh every 5 minutes.

Default Dashboard:

The default dashboard is used to customize the items into functional views. These views are named as Application overview, compliance overview, network overview, system monitoring overview and threat and security monitoring.

Creating a Custom Dashboard:

- Create custom dashboards that are relevant to your responsibilities. 255 dashboards per user is the maximum; however, performance issues might occur if you create more than 10 dashboards.
- Add and remove dashboard items from default or custom dashboards.
- Move and position items to meet your requirements. When you position items, each item automatically resizes in proportion to the dashboard.
- Add custom dashboard items that are based on any data.

Investigation:

Log activities can be investigated using the Log Activity Dashboard item. The information can be rendered in various chart types, such as bar, pie, table, and time series.

Configuring dashboard chart types:

- Click the Dashboard tab.
- From the Show Dashboard list box, select the dashboard that contains the item you want to customize.
- On the header of the dashboard item you want to configure, click the Settings icon.
- Configure the chart parameters.
  - From the Value to Graph list box, select the object type that you want to graph on the chart. Options include all normalized and custom event or flow parameters that are included in your search parameters.
  - Select a chart type: Bar, pie, and table charts are only available for grouped events or flows.
  - Data accumulates so that when you run a time series saved search, a cache of event or flows data is available to display the data for the previous time period. Accumulated parameters are indicated by an asterisk (*) in the Value to Graph list box. If you select a value to graph that is not accumulated (no asterisk), time series data is not available. Select the Capture Time Series Data checkbox to enable time series capture. When you select this checkbox, the chart feature accumulates data for time series charts. By default, this option is disabled.

Offence Management:

Use the Offenses tab to access all of the data you need to understand even the most complex threats. By providing immediate context for the offense, QRadar helps you quickly identify which offenses are the most important and begin an investigation to find the source of the suspected security attack or policy breach. Steps involved in offence management include -

- *Prioritization* using offense magnitude computation
- *Chaining* to identify the root cause by connecting multiple symptoms
- *Indexing* groups events or flows from different rules indexed on same property
- *Retention* determines how long inactive and closed offenses are kept before they are removed from the QRadar console

- *Investigations* uses rules to monitor the events and flows in your network to detect security threats
- *Actions* provides the capability to act on the offenses as you investigate them

Topics Explored:

- Introduction to CEH
- Introduction to Programming Languages
- OSINT Framework
- Introduction to Linux
- Introduction to Networking
- Hacking Web Applications
- OWASP Top 10 vulnerabilities
- Vulnerability analysis
- SOC & SIEM & IBM QRadar
- Threat intelligence integration
- AI with Cybersecurity

Tools Explored:

Shodan, NSLookup, Nmap, Hydra, Testfire.net, Metasploit, Nessus, acunetix, SOC Dashboard

Future Scope:

Stage 1: Future scope of web application testing

The future scope of web application testing is vast and evolving rapidly, driven by technological advancements and increasing user expectations. As web applications become more complex and integral to daily life, the demand for robust, efficient, and automated testing solutions is rising. Emerging trends such as artificial intelligence and machine learning are set to revolutionize testing processes by enabling predictive analytics, enhancing test automation, and improving bug detection capabilities. Additionally, the growing emphasis on security and privacy will necessitate advanced testing frameworks to identify vulnerabilities and ensure compliance with stringent regulations. Continuous integration and continuous deployment (CI/CD) pipelines will further streamline testing, allowing for more frequent and reliable releases. Moreover, the proliferation of IoT devices and progressive web applications (PWAs) will expand the scope of testing to include new environments and user interactions. Overall, the future of web application testing promises to be dynamic, with a focus on innovation, efficiency, and security.

Stage 2 :- Future scope of testing process you understood

The future scope of the testing process in the context of cyber security is poised to expand significantly, driven by the increasing sophistication of cyber threats and the rapid advancement of technology. As organizations continue to digitize their operations and data, the demand for robust security testing will escalate. Future testing processes will likely incorporate advanced techniques such as artificial intelligence and machine learning to predict and identify vulnerabilities more effectively. Automation will play a critical role, enabling continuous and comprehensive security assessments, thus reducing the window of exposure to potential attacks. Moreover, the adoption of

DevSecOps practices will integrate security testing into the entire software development lifecycle, ensuring that security is a fundamental consideration from the outset. As cyber threats evolve, so too will the methodologies and tools used in testing, emphasizing the need for continuous education and adaptation within the cyber security profession. This dynamic landscape will necessitate collaboration between organizations, security professionals, and regulatory bodies to establish standards and best practices that can keep pace with emerging threats.

Stage 3 :- Future scope of SOC / SEIM

The future scope of Security Operations Centers (SOC) and Security Information and Event Management (SIEM) systems is poised for significant expansion as cyber threats become increasingly sophisticated. SOCs are expected to integrate advanced technologies such as artificial intelligence and machine learning to enhance threat detection, response times, and predictive analytics. These technologies will enable SOCs to automate routine tasks, allowing security analysts to focus on more complex threat analysis and mitigation strategies. SIEM systems, on the other hand, will continue to evolve with enhanced capabilities for real-time monitoring, advanced correlation of security events, and improved data analysis. As organizations increasingly adopt cloud computing, IoT, and remote work models, SOC and SIEM solutions will need to address the unique security challenges these environments present. Moreover, regulatory compliance and data privacy concerns will drive the demand for more robust and scalable SOC and SIEM solutions, ensuring they remain critical components of an organization's cybersecurity infrastructure.