STUDY OF CYBER SECURITY THAT BRINGS SECURITY THREATS TO LIGHT BEFORE THEY HARM BUSINESS OPERATIONS

Overview :-

A web application is application software that is accessed using a web browser. Web applications are delivered on the World Wide Web to users with an active network connection. Web application security is the idea of building websites to function as expected, even when they are under attack. The concept involves a collection of security controls engineered into a Web application to protect its assets from potentially malicious agents. Web applications, like all software, inevitably contain defects. Some of these defects constitute actual vulnerabilities that can be exploited, introducing risks to organizations.

Web application security defends against such defects. It involves leveraging secure development practices and implementing security measures throughout the software development life cycle (SDLC), ensuring that design-level flaws and implementation-level bugs are addressed. Testing the security of a Web application often involves sending different types of input to provoke errors and make the system behave in unexpected ways. These so called “negative tests” examine whether the system is doing something it isn’t designed to do.

Web applications need to freely allow traffic through a variety of ports and usually require authentication; this means they also require a complex web application vulnerability scanner. Since websites must allow traffic to come and in and out of the network, hackers often attack the most commonly used ports. This includes:

* Port 80 (HTTP): For unsecured website traffic
* Port 443 (HTTPS): For secured website traffic
* Port 21 (FTP): The file transfer protocol for transferring files to and from your servers
* Ports 25 (SMTP), for simple mail transfer protocol, and port 110 (POP3), the default unencrypted port: Email protocols often used by organizations to send and receive email.

The OWASP Top 10 is a standard awareness document for developers and web application security. It represents a broad consensus about the most critical security risks to web applications. Companies should adopt this document and start the process of ensuring that their web applications minimize these risks. Using the OWASP Top 10 is perhaps the most effective first step towards changing the software development culture within your organization into one that produces more secure code.

Tools discussed to reach the milestones

* Nessus is a platform developed by Tenable that scans for security vulnerabilities in devices, applications, operating systems, cloud services and other network resources.
* PortSwigger offers tools for web application security, testing, & scanning. Choose from a range of security tools, & identify the very latest.
* Burp Suite is an integrated platform/graphical tool for performing security testing of web applications. Its various tools work seamlessly together to support the entire testing process, from initial mapping and analysis of an application's attack surface, through to finding and exploiting security vulnerabilities.
* Security information and event management, SIEM for short, is a solution that helps organizations detect, analyze, and respond to security threats before they harm business operations.
* QRadar is a network security management platform that provides situational awareness and compliance support. QRadar uses a combination of flow-based network knowledge, security event correlation, and asset-based vulnerability assessment.

List of teammates–

| S.no | Name | Collage | contact |
| --- | --- | --- | --- |
| 1  2  3 | Dr.R.Menaka AP/IT  Dr.M.Kavitha ASP/IT  Mrs.E.Sudha AP/IT | Velalar College of Engineering and Technology,  Erode,  Tamilnadu. | 9842042627  9500719815  9488934270 |

**List of Vulnerability Table ➖**

|  |  |
| --- | --- |
| Name of The Vulnerability | References-CWE |
| Broken Access Control | CWE-35 Path Traversal: '.../...//' |
| Cryptographic Failures | CWE-319 Cleartext Transmission of Sensitive Information |
| Injection | CWE-20 Improper Input Validation |
| A04 Insecure Design | CWE-235 Improper Handling of Extra Parameters |
| A05 Security Misconfiguration | CWE-16 Configuration |
| A06 Vulnerable and Outdated Components | CWE-1104 Use of Unmaintained Third Party Components |
| A07 Identification and Authentication Failures | CWE-287 Improper Authentication |
| A08 Software and Data Integrity Failures | CWE-345 Insufficient Verification of Data Authenticity |
| A09 Security Logging and Monitoring Failures | CWE-778 Insufficient Logging |
| A10 Server Side Request Forgery (SSRF) | CWE-918 Server-Side Request Forgery (SSRF) |

**REPORT:-**

**1 Vulnerability Name**: Path Traversal: '.../...//'

**CWE** : CWE-35

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

**Description**: if pathname to access directory or files not properly neutralize '.../...//' (doubled triple dot slash) then path sequences can resolve to a location that is outside of that directory or otherwise construct a pathname resolved to a location that are not within a restricted directory,

**Business Impact**: The path traversal flaw occurs when the user parameters aren't sanitised and/or there is a lack of access control to the resources. It's then possible for an attacker to modify the parameters of the request to ask to return other resources. The impact of this flaw is generally critical. A path traversal attack (also known as directory traversal) aims to access files and directories that are stored outside the web root folder.

**Why Does The Path Traversal Vulnerability Occur?**

1. **Insecure input validation:** if user input is not properly validated, it may be possible for an attacker to inject malicious input that could be used to exploit a path traversal vulnerability.
2. **Poorly configured web servers:** if a web server is not properly configured, it may be possible for an attacker to access restricted directories.
3. **Insecure**[**file permissions**](https://www.thesecmaster.com/understand-the-role-of-file-ownership-and-permissions-in-linux/)**:** if files or directories have lax permissions, it may be possible for an attacker to gain access to them.

So, The attacker might be able:

* to read files, potentially:
  + Configuration files where there are usually secrets (credentials, keys…) which then allow to exploit new vulnerabilities,
  + Sensitive operating system files,
* to read the source code,
* to analyse the organisation of the server,
* sometimes to write on the server, which can lead to:
  + a modification of the application’s behaviour,
  + even, to take control of the server.

**How to protect yourself from path traversal?**

To avoid these flaws, several measures should be implemented:

* Do not use user input directly to call a file.
* User data shouldn’t be interpreted. It should be encoded, escaped and cleaned.
* It should be validated against a list of allowed expressions. If this isn’t possible, then the validation must confirm that there are only allowed contents (e.g. only alphanumeric characters).

**2. Vulnerability Name**: Cleartext Transmission of Sensitive Information

**CWE** : CWE-319

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

**Description**:

Cleartext is the one where the data is unencrypted and is not intended for the encryption process.

**Business Impact**:The first thing is to determine the protection needs of data in transit and at rest. Encrypts credit card numbers in a database using automatic database encryption and allowing data to automatically decrypted when retrieved, allowing a SQL injection flaw to retrieve credit card numbers in clear text. Between transmits to rest, sensitive or security-critical data in cleartext in a communication channel surely be sniffed by unauthorized actors.

For example, passwords, credit card numbers, health records, personal information, and business secrets require extra protection, mainly if that data falls under privacy laws, e.g., GDPR and PCI DSS.

**3 Vulnerability Name**: Improper Input Validation

**CWE** : CWE-20

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

**Description**: **Input validation** is a frequently-used technique for checking potentially dangerous **inputs** in order to ensure that the **inputs** are safe for processing . Improper input validation or unchecked user input is a type of vulnerability in computer software that may be used for security exploits. This vulnerability is caused when "[t]he product does not validate or incorrectly validates input that can affect the control flow or data flow of a program."

**Business Impact**: For web applications, input validation usually means verifying user inputs provided in web forms, query parameters, uploads, and so on. Missing or improper input validation is a major factor in many web security vulnerabilities, including cross-site scripting (XSS) and SQL injection. Eg. Cookies are another common source of user input. Like form data, cookies can be used to input data into your application. However, cookies are often used to store session information, and if they are not properly validated, they can be used to hijack user sessions.

**4 Vulnerability Name**: Improper Handling of Extra Parameters

**CWE** : CWE-235

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

**Description**:"The product does not handle or incorrectly handles when the number of parameters, fields, or arguments with the same name exceeds the expected amount."

**Business Impact**: Several versions of Apache Tomcat 5, 6, and 7 inefficiently handled parameters. Attackers could use this inefficency to overload the system with an extreme amount of parameters to cause a denial of service attack.

**5 Vulnerability Name**: Configuration

**CWE** : CWE-16

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

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

**Business Impact**: The application server's configuration allows detailed error messages, e.g., stack traces, to be returned to users. This potentially exposes sensitive information or underlying flaws such as component versions that are known to be vulnerable.

**6 Vulnerability Name**: Use of Unmaintained Third Party Components

**CWE** : CWE-1104

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

**Description**:Reliance on components that are no longer maintained can make it difficult or impossible to fix significant bugs, vulnerabilities, or quality issues. In effect, unmaintained code can become obsolete.

**Business Impact**: This issue makes it more difficult to maintain the product, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.

**7 Vulnerability Name**: Improper Authentication

**CWE** : CWE-287

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

**Description**:Examples of improper authentication vulnerabilities include: No authentication: When there is no authentication for a critical function, then attackers get unrestricted access easily

**Business Impact**: . Weak passwords: When users choose weak passwords, it makes it easier for attackers to guess or crack them. Authentication vulnerabilities are issues that affect authentication processes and make websites and applications susceptible to security attacks in which an attacker can masquerade as a legitimate user

**8 Vulnerability Name**: Insufficient Verification of Data Authenticity

**CWE** : CWE-345

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

**Description** his class of weaknesses is a result of trust issues between data exchange parties. If the application fails to verify data origin or its authenticity, an attacker might be able to perform spoofing attacks against vulnerable application or their clients.

**Business Impact**: This weakness occurs when the application transmits or stores authentication credentials and uses an insecure method that is susceptible to unauthorized interception and/or retrieval. When the Access Control List (ACL) connected to the NGC folder is corrupted, the error message “Your credentials could not be verified” may also appear. In this case, you can try to reset the ACL in safe mode to fix the matter.

**9 Vulnerability Name**: Insufficient Logging

**CWE** : CWE-778

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

**Description**:Exploitation of insufficient logging and monitoring is the bedrock of nearly every major incident . Attackers rely on the lack of monitoring and timely response to achieve their goals without being detected.

**Business Impact**: . One strategy for determining if you have sufficient monitoring is to examine the logs following penetration testing. The testers’ actions should be recorded sufficiently to understand what damages they may have inflicted.

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

**CWE** : CWE-918

**OWASP Category**: A10:2021:Server Side Request Forgery (SSRF)

**Description:**Server-side request forgery is a type of computer security exploit where an attacker abuses the functionality of a server causing it to access or manipulate information in the realm of that server that would otherwise not be directly accessible to the attacker.

**Business Impact**: let an attacker send crafted requests from the back-end server of a vulnerable application. Criminals usually use SSRF attacks to target internal systems that are behind firewalls and are not accessible from the external network. An attacker may also leverage SSRF to access services available through the loopback interface (127.0.0.1) of the exploited server.