Stage 1 Report – OWASP AND SANS

Prepared by	Dr.Sumaiya Thaseen, Associate Professor, School of Computer Engineering and Information Systems, Vellore Institute of Technology, Vellore, Tamil Nadu

1. Broken Access Control

<table>
<tr><td>Vulnerability Name</td><td>Broken Access Control</td></tr>
<tr><td>CWE</td><td>22 Path Traversal</td></tr>
<tr><td>OWASP Category</td><td>A01- 2021</td></tr>
<tr><td>Description</td><td>The product uses external input to construct a pathname that is intended to identify a file or directory that is located underneath a restricted parent directory, but the product does not properly neutralize special elements within the pathname that can cause the pathname to resolve to a location that is outside of the restricted directory.</td></tr>
<tr><td>Business Impact</td><td>The financial impact of broken access control can be substantial. A data breach can result in financial losses from lawsuits, regulatory fines, and reputational damage. A recent study found that the average cost of a data breach is $3.86 million, with the healthcare and financial sectors facing the highest costs.
Moreover, broken access control can lead to lost productivity and increased operational cost.
<table>
<tr><th>Scope</th><th>Impact</th></tr>
<tr><td>Integrity
Confidentiality
Availability</td><td>Technical Impact: Execute Unauthorized Code or Commands
The attacker may be able to create or overwrite critical files that are used to execute code, such as programs or libraries.</td></tr>
<tr><td>Integrity</td><td>Technical Impact: Modify Files or Directories
The attacker may be able to overwrite or create critical files, such as programs, libraries, or important data. If the targeted file is used for a security mechanism, then the attacker may be able to bypass that mechanism. For example, appending a new account at the end of a password file may allow an attacker to bypass authentication.</td></tr>
</table>
</td></tr>
</table>

	Confidentiality	**Technical Impact:** *Read Files or Directories* The attacker may be able read the contents of unexpected files and expose sensitive data. If the targeted file is used for a security mechanism, then the attacker may be able to bypass that mechanism. For example, by reading a password file, the attacker could conduct brute force password guessing attacks in order to break into an account on the system.
	Availability	**Technical Impact:** *DoS: Crash, Exit, or Restart* The attacker may be able to overwrite, delete, or corrupt unexpected critical files such as programs, libraries, or important data. This may prevent the product from working at all and in the case of a protection mechanisms such as authentication, it has the potential to lockout every user of the product.
References	https://cwe.mitre.org/data/definitions/22.html https://medium.com/geekculture/the-cost-of-broken-access-controlunderstanding-the-financial-impact-on-your-businessa5805c35a9e9	

2. Cryptographic Failures

Vulnerability Name	Cryptographic Failures
CWE	261 Weak Encoding for Password
OWASP Category	A02: 2021
Description	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.
Business Impact	Weak cryptographic implementations can lead to data breaches and unauthorized access to sensitive information, resulting in financial losses and damaged reputation. **Scope** | **Impact** Access Control | **Technical Impact:** *Gain Privileges or Assume Identity*
References	https://cwe.mitre.org/data/definitions/261.html

3. Injection

Vulnerability Name	Injection
CWE	94 -Improper Control of Generation of Code
OWASP Category	A03: 2021
Description	The product constructs all or part of a code segment using externally-influenced input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could modify the syntax or behavior of the intended code segment.

<table>
<tr><td>Business Impact</td><td>Injection attacks can cause data loss, data corruption, security breaches, and possibly the loss of control of the target host and the release of sensitive information linked to the host. A “successful” injection can also let attackers access the database without permission.
Cybercriminals could gain unwanted or administrative access to private information and resources. Potential data breaches can occur as a result of unauthorized access to resources.
<table>
<tr><th>Scope</th><th>Impact</th></tr>
<tr><td>Access Control</td><td>Technical Impact: Bypass Protection Mechanism
In some cases, injectable code controls authentication; this may lead to a remote vulnerability.</td></tr>
<tr><td>Access Control</td><td>Technical Impact: Gain Privileges or Assume Identity
Injected code can access resources that the attacker is directly prevented from accessing.</td></tr>
<tr><td>Integrity
Confidentiality
Availability</td><td>Technical Impact: Execute Unauthorized Code or Commands
Code injection attacks can lead to loss of data integrity in nearly all cases as the control-plane data injected is always incidental to data recall or writing. Additionally, code injection can often result in the execution of arbitrary code.</td></tr>
<tr><td>Non-Repudiation</td><td>Technical Impact: Hide Activities
Often the actions performed by injected control code are unlogged</td></tr>
</table>
</td></tr>
<tr><td>References</td><td>https://cwe.mitre.org/data/definitions/94.html</td></tr>
</table>

4. Insecure Design

Vulnerability Name	Insecure Design
CWE	73 External Control of File Name or Path
OWASP Category	A04: 2021

<table>
<tr><td>Description</td><td>Insecure design is a broad category representing different weaknesses, expressed as “missing or ineffective control design.” Insecure design is not the source for all other Top 10 risk categories. There is a difference between insecure design and insecure implementation. We differentiate between design flaws and implementation defects for a reason, they have different root causes and remediation. A secure design can still have implementation defects leading to vulnerabilities that may be exploited. An insecure design cannot be fixed by a perfect implementation as by definition, needed security controls were never created to defend against specific attacks. One of the factors that contribute to insecure design is the lack of business risk profiling inherent in the software or system being developed, and thus the failure to determine what level of security design is required.</td></tr>
<tr><td>Business Impact</td><td>Insecure design would allow attackers to:
Bypass the authentication mechanisms used by a web application.
Modify certain URL parameters through unauthorized channels.
Access the systems to mine them for sensitive information.
Insecure technology and vulnerabilities in critical systems may invite malicious cyber intrusions, leading to potential safety1 risks
.<table><tr><th>Scope</th><th>Impact</th></tr><tr><td>Integrity
Confidentiality</td><td>Technical Impact: Read Files or Directories; Modify Files or Directories
The application can operate on unexpected files. Confidentiality is violated when the targeted filename is not directly readable by the attacker.</td></tr><tr><td>Integrity
Confidentiality
Availability</td><td>Technical Impact: Modify Files or Directories; Execute Unauthorized Code or Commands
The application can operate on unexpected files. This may violate integrity if the filename is written to,</td></tr></table></td></tr>
<tr><td></td><td>or if the filename is for a program or other form of executable code.</td></tr>
</table>

	Availability	**Technical Impact:** *DoS: Crash, Exit, or Restart; DoS: Resource Consumption (Other)* The application can operate on unexpected files. Availability can be violated if the attacker specifies an unexpected file that the application modifies. Availability can also be affected if the attacker specifies a filename for a large file, or points to a special device or a file that does not have the format that the application expects.
References	https://cwe.mitre.org/data/definitions/73.html	

5. Security Misconfiguration

Vulnerability Name	Security Misconfiguration
CWE	16
OWASP Category	A05: 2021
Description	The application might be vulnerable if the application is: • Missing appropriate security hardening across any part of the application stack or improperly configured permissions on cloud services. • Unnecessary features are enabled or installed (e.g., unnecessary ports, services, pages, accounts, or privileges). • Default accounts and their passwords are still enabled and unchanged. • Error handling reveals stack traces or other overly informative error messages to users. • For upgraded systems, the latest security features are disabled or not configured securely. • The security settings in the application servers, application frameworks (e.g., Struts, Spring, ASP.NET), libraries, databases, etc., are not set to secure values.

	• The server does not send security headers or directives, or they are not set to secure values. • The software is out of date or vulnerable . • Without a concerted, repeatable application security configuration process, systems are at a higher risk.
Business Impact	Security misconfigurations can allow attackers to gain unauthorized access to the networks, systems and data which in turn can cause significant monetary and reputational damage to your organization.
References	https://cwe.mitre.org/data/definitions/16.html

6. Vulnerable and outdated components

Vulnerability Name	Vulnerable and outdated components
CWE	937
OWASP Category	A06: 2021
Description	You are likely vulnerable: • If you do not know the versions of all components you use (both client-side and server-side). This includes components you directly use as well as nested dependencies. • If the software is vulnerable, unsupported, or out of date. This includes the OS, web/application server, database management system (DBMS), applications, APIs and all components, runtime environments, and libraries. • If you do not scan for vulnerabilities regularly and subscribe to security bulletins related to the components you use. • If you do not fix or upgrade the underlying platform, frameworks, and dependencies in a risk-based, timely fashion. This commonly happens in environments when patching is a monthly or quarterly task under change control, leaving organizations open to days or months of unnecessary exposure to fixed vulnerabilities. • If software developers do not test the compatibility of updated, upgraded, or patched libraries.
Business Impact	Using vulnerable and outdated components can pose significant risks to web applications. Attackers can exploit these vulnerabilities

	to launch attacks such as SQL injection, cross-site scripting (XSS), remote code execution, and more. One of the most common ways that hackers target organizations is by exploiting vulnerabilities in outdated software. Outdated software risks can leave you open to a variety of hacks, including ransomware, malware, data breaches, and more.
References	https://cwe.mitre.org/data/definitions/937.html

7. Identification and Authentication Failures

Vulnerability Name	Identification and Authentication Failures
CWE	287
OWASP Category	A07: 2021
Description	Confirmation of the user's identity, authentication, and session management is critical to protect against authentication-related attacks. There may be authentication weaknesses if the application: • Permits automated attacks such as credential stuffing, where the attacker has a list of valid usernames and passwords. • Permits brute force or other automated attacks. • Permits default, weak, or well-known passwords, such as "Password1" or "admin/admin". • Uses weak or ineffective credential recovery and forgotpassword processes, such as "knowledge-based answers," which cannot be made safe. • Uses plain text, encrypted, or weakly hashed passwords data stores • Has missing or ineffective multi-factor authentication. • Exposes session identifier in the URL. • Reuse session identifier after successful login. • Does not correctly invalidate Session IDs. User sessions or authentication tokens (mainly single sign-on (SSO) tokens)

<table>
<tr><td></td><td>aren't properly invalidated during logout or a period of inactivity.</td></tr>
<tr><td>Business Impact</td><td>The failure of a system to identify and/or authenticate leaves the application susceptible to attacks and leaves user accounts/data at risk. Authentication failure poses a stringent and profound threat to an organization's security.
Hackers can use broken authentication attacks and session hijacking to gain access to the system by forging session data, such as cookies, and stealing login credentials.

<table><tr><th>Scope</th><th>Impact</th></tr><tr><td>Integrity
Confidentiality
Availability
Access Control</td><td>Technical Impact: Read Application Data; Gain Privileges or Assume Identity; Execute Unauthorized Code or Commands

This weakness can lead to the exposure of resources or functionality to unintended actors, possibly providing attackers with sensitive information or even execute arbitrary code.</td></tr></table></td></tr>
<tr><td>References</td><td>https://cwe.mitre.org/data/definitions/287.html</td></tr>
</table>

8. Software and Data Integrity Failures

Vulnerability Name	Software and Data Integrity Failures
CWE	345 Insufficient Verification of Data Authenticity
OWASP Category	A08: 2021

Description	Software and data integrity failures relate to code and infrastructure that does not protect against integrity violations. An example of this is where an application relies upon plugins, libraries, or modules from untrusted sources, repositories, and content delivery networks (CDNs). An insecure CI/CD pipeline can introduce the potential for unauthorized access, malicious code, or system compromise. Lastly, many applications now include auto-update functionality, where updates are downloaded without sufficient integrity verification and applied to the previously trusted application. Attackers could potentially upload their own updates to be distributed and run on all installations. Another example is where objects or data are encoded or serialized into a structure that an attacker can see and modify is vulnerable to insecure deserialization.
Business Impact	Software and data integrity failures refer to the state of corruption or integrity violation of data or software for which no proper checks are in place in the present infrastructure of the organization. Integrity violation of data and the software includes scenarios where any unauthorized entity modifies the data crucial to an organization's working or modifies the code of components being utilized by the software for its working. • **Software vulnerabilities** Attackers may try to find and exploit flaws or weaknesses in software to gain access to a system or commit data theft. **Example:** An attacker found a vulnerable version of server-hosting software which he then exploited to gain access to the server and plant a backdoor into the server to gain access to the server remotely.
References	https://cwe.mitre.org/data/definitions/345.html https://aspiainfotech.com/2023/01/12/securing-your-data-theimportance-of-addressing-software-and-data-integrity-failures/

9. Security Logging and Monitoring Failures

Vulnerability Name	Security Logging and Monitoring Failures

CWE	117 Improper Output Neutralization for Logs
OWASP Category	A09: 2021
Description	This category is to help detect, escalate, and respond to active breaches. Without logging and monitoring, breaches cannot be detected. Insufficient logging, detection, monitoring, and active response occurs any time: • Auditable events, such as logins, failed logins, and high-value transactions, are not logged. • Warnings and errors generate no, inadequate, or unclear log messages. • Logs of applications and APIs are not monitored for suspicious activity.
	• Logs are only stored locally. • Appropriate alerting thresholds and response escalation processes are not in place or effective. • Penetration testing and scans by dynamic application security testing (DAST) tools (such as OWASP ZAP) do not trigger alerts. • The application cannot detect, escalate, or alert for active attacks in real-time or near real-time.
Business Impact	The risks of security logging and monitoring failures can be severe, including the inability to identify and respond to security incidents and breaches. This can allow attackers to gain unauthorized access to systems and data without detection. In addition, the attackers attack systems further, maintain persistence, pivot to more systems, and tamper, extract or destroy data.
References	https://cwe.mitre.org/data/definitions/117.html

10. Server Side Request Forgery (SSRF)

Vulnerability Name	Server-Side Request Forgery (SSRF)
CWE	918 Server-Side Request Forgery (SSRF)
OWASP Category	A10: 2021

<table>
<tr><td>Description</td><td>SSRF flaws occur whenever a web application is fetching a remote resource without validating the user-supplied URL. It allows an attacker to coerce the application to send a crafted request to an unexpected destination, even when protected by a firewall, VPN, or another type of network access control list (ACL).

As modern web applications provide end-users with convenient features, fetching a URL becomes a common scenario. As a result, the incidence of SSRF is increasing. Also, the severity of SSRF is becoming higher due to cloud services and the complexity of architectures.

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.</td></tr>
<tr><td>Business Impact</td><td>A successful SSRF attack can often result in unauthorized actions or access to data within the organization. This can be in the vulnerable application, or on other back-end systems that the application can communicate with.</td></tr>
<tr><td></td><td><table>
<tr><th>Scope</th><th>Impact</th></tr>
<tr><td>Confidentiality</td><td>Technical Impact: Read Application Data</td></tr>
<tr><td>Integrity</td><td>Technical Impact: Execute Unauthorized Code or Commands</td></tr>
</table></td></tr>
<tr><td>References</td><td>https://cwe.mitre.org/data/definitions/918.html</td></tr>
</table>

SANS -TOP 20 Vulnerabilities

1. Memory Buffer Error

Vulnerability Name	Memory Buffer Error
CWE	119
SANS Category	01
Description	This flaw is usually introduced during Architecture and Design, Implementation, Operation stages of the SDLC. This buffer overflow happens when an application process tries to store more data than it can hold in the memory. Since the buffers can only store some level of data and when that level is reached and exceeded, the data flows to another memory location which can corrupt the data already contained in that buffer. This incident sometimes happens accidentally through some programming error, but the aftereffect could be disastrous, as this can erase data, steal confidential information, and even the whole application could crash because of this buffer overflow. The example below shows a buffer allocated with 8bytes storage. But it overflowed by 2bytes because of more data was sent for execution.
Business Impact	An attacker who controls the user input can read or write to arbitrary memory locations. As a result, it is possible to obtain potentially sensitive information from memory, it could also cause memory corruption and crash the application or even execute arbitrary code on the target system.
References	https://cwe.mitre.org/data/definitions/119.html https://www.immuniweb.com/vulnerability/buffererrors.html#impact

2. Cross-site Scripting

Vulnerability Name	Cross-site Scripting

CWE	79
SANS Category	SANS 02
Description	Cross-site Scripting (XSS) is an injection attack that usually happens when a malicious actor or an attacker injects malicious or harmful script into a web application which can be executed through the web browsers. Once the malicious script finds its way into the compromised system, it can be used to perform different malicious activities. Some of the malicious activities can be in the form of transferring private information like cookies that have the session information from the victim's computer to the attacker's computer.
Business Impact	Cross-site scripting (XSS) vulnerabilities continue to remain a major threat to web applications as attackers exploiting XSS attacks can gain control of the user's account and steal personal information such as passwords, bank account numbers, credit card info, personally identifiable information (PII), social security numbers, and more. The impact of cross-site scripting vulnerabilities can vary from one web application to another. It ranges from session hijacking to credential theft and other security vulnerabilities. By exploiting a cross-site scripting vulnerability, an attacker can impersonate a legitimate user and take over their account. If the victim user has administrative privileges, it might lead to severe damage such as modifications in code or databases to further weaken the security of the web application, depending on the rights of the account and the web application.
References	https://www.cypressdatadefense.com/blog/cross-site-scriptingvulnerability/

3. Unvalidated Input Error

Vulnerability Name	Unvalidated Input Error
CWE	20
SANS Category	SANS 03

<table>
<tr><td>Description</td><td>The application receives input, but fails to validate the input, whether it has all necessary details needed for it to be accepted into the system for processing.

When there is input sanitization, this can be used to check any potentially dangerous inputs in order to ensure that the inputs are</td></tr>
<tr><td></td><td>safe to be processed with the source code or when it’s an input that is needed to communicate with other components.

When such inputs are not properly sanitized or validated, then this will pave way for an attacker to send a malicious input that the main application will generously process and this will lead to changes in the control flow, arbitrary control of a resource, or arbitrary code execution.</td></tr>
<tr><td>Business Impact</td><td>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.

<table>
<tr><th>Scope</th><th>Impact</th></tr>
<tr><td>Availability</td><td>Technical Impact: DoS: Crash, Exit, or Restart; DoS: Resource Consumption (CPU); DoS: Resource Consumption (Memory)

An attacker could provide unexpected values and cause a program crash or excessive consumption of resources, such as memory and CPU.</td></tr>
<tr><td>Confidentiality</td><td>Technical Impact: Read Memory; Read Files or Directories

An attacker could read confidential data if they are able to control resource references.</td></tr>
<tr><td>Integrity
Confidentiality
Availability</td><td>Technical Impact: Modify Memory; Execute Unauthorized Code or Commands

An attacker could use malicious input to modify data or possibly alter control flow in unexpected ways, including arbitrary command execution.</td></tr>
</table>
</td></tr>
</table>

References	https://cwe.mitre.org/data/definitions/20.html

4. Sensitive Information Exposure Error

Vulnerability Name	Sensitive Information Exposure Error
CWE	200
SANS Category	SANS 04
Description	The product exposes sensitive information to an actor that is not explicitly authorized to have access to that information. **Below are some sensitive information that could be exposed:** • Personal information like personal messages, financial data, health status records, geographic location, or contact details • System configuration details and environment, **for example,** the operating system and installed packages • Business Record and intellectual property • Network configuration details • Internal application state • Metadata like the message headers
Business Impact	The impact of this vulnerability is significant because it can lead to unauthorized access to the system, resulting in the exposure of sensitive data. Attackers can use the information obtained from error messages to carry out further attacks, such as injection attacks or privilege escalation. When this sensitive data is accessed by an attacker as a result of a data breach, users are at risk for sensitive data exposure. Data breaches that result in the exposure of sensitive credentials can come with costs in the millions of dollars, destroying a company's reputation along with it.
References	https://cwe.mitre.org/data/definitions/200.html

5. Out-of-bounds Read Error

Vulnerability Name	Out-of-bounds Read Error
CWE	125
SANS Category	SANS 05

<table>
<tr><td>Description</td><td>This usually occurs when the application reads data past the normal level, either to the end or before the beginning of the buffer. This gives unprivileged access to an attacker to read sensitive information from other memory locations, which can as well leads to a system or application crash. A crash will certainly happen when the code reads data and thinks there is an indicator in place that stops the read operation like a NULL that is applied to a string.</td></tr>
<tr><td>Business Impact</td><td>Out-of-bounds reads can result in unexpected program behavior, crashes, and potential security vulnerabilities.
<table><tr><th>Scope</th><th>Impact</th></tr><tr><td>Confidentiality</td><td>Technical Impact: Bypass Protection Mechanism

By reading out-of-bounds memory, an attacker might be able to get secret values, such as memory addresses, which can be bypass protection mechanisms such as ASLR in order to improve the reliability and likelihood of exploiting a separate weakness to achieve code execution instead of just denial of service.</td></tr></table></td></tr>
<tr><td>References</td><td>https://cwe.mitre.org/data/definitions/125.html</td></tr>
</table>

6. SQL Injection

Vulnerability Name	SQL Injection
CWE	89
SANS Category	06
Description	SQL injection is a form of security vulnerability whereby the attacker injects a Structured Query Language (SQL) code to the Webform input box in order to gain access to resources or change data that is not authorized to access.This vulnerability can be introduced to the application during the design, implementation, and operation stages.

Business Impact	SQL injection attacks allow attackers to spoof identity, tamper with existing data, cause repudiation issues such as voiding transactions or changing balances, allow the complete disclosure of all data on the system, destroy the data or make it otherwise unavailable, and become administrators of the database server. Cybercriminals could gain unwanted or administrative access to private information and resources. Potential data breaches can occur as a result of unauthorized access to resources.
References	https://cwe.mitre.org/data/definitions/89.html

7. Previously Freed Memory

Vulnerability Name	Previously Freed Memory
CWE	416
SANS Category	07
Description	This issue is caused by the referencing of memory after it has been released, which can seriously lead to a program crash. When you use a previously freed memory, this can have adverse
	consequences, like corrupting of valid data, arbitrary code execution which is dependent on the flaw timing. **Two common causes are:** • Error conditions within the software and in some other exceptional cases. • No explanation as to which part of the program caused the free memory.
Business Impact	**Scope** / **Impact** Integrity — **Technical Impact:** *Modify Memory* The use of previously freed memory may corrupt valid data, if the memory area in question has been allocated and used properly elsewhere. Availability — **Technical Impact:** *DoS: Crash, Exit, or Restart* If chunk consolidation occurs after the use of previously freed data, the process may crash when invalid data is used as chunk information.

	Integrity Confidentiality Availability	**Technical Impact:** *Execute Unauthorized Code or Commands* If malicious data is entered before chunk consolidation can take place, it may be possible to take advantage of a writewhat-where primitive to execute arbitrary code.
References	https://cwe.mitre.org/data/definitions/416.html	

8. Integer Overflow Error

Vulnerability Name	Integer Overflow Error
CWE	190
SANS Category	08
Description	When a calculation is processed by an application and there is a logical assumption that the resulting value will be greater than the exact value, integer overflow happens. Here, an integer value increases to a value that cannot be stored in a location. When this happens, the value will usually wrap to become a very small or negative value. If the wrapping is expected, then it's fine,
	but there can be security consequences if the wrap is unexpected. When this scenario occurs, it could be termed critical as the result is used to manage looping, security decision, used to allocate memory, and many more
Business Impact	Integer overflow attacks involve exploiting bugs in software. When these integer overflow flaws are abused, it can lead to disastrous results, including infecting devices with spyware.

Scope	Impact
Availability	**Technical Impact:** *DoS: Crash, Exit, or Restart; DoS: Resource Consumption (CPU); DoS: Resource Consumption (Memory); DoS: Instability* This weakness will generally lead to undefined behavior and therefore crashes. In the case of overflows involving loop index variables, the likelihood of infinite loops is also high.

	Integrity	**Technical Impact:** *Modify Memory* If the value in question is important to data (as opposed to flow), simple data corruption has occurred. Also, if the wrap around results in other conditions such as buffer overflows, further memory corruption may occur.
	Confidentiality Availability Access Control	**Technical Impact:** *Execute Unauthorized Code or Commands; Bypass Protection Mechanism* This weakness can sometimes trigger buffer overflows which can be used to execute arbitrary code. This is usually outside the scope of a program's implicit security policy.
References	https://www.comparitech.com/blog/information-security/integeroverflow-attack/ https://cwe.mitre.org/data/definitions/190.html	

9. CSRF

Vulnerability Name	Cross Site Request Forgery
CWE	352
SANS Category	9
Description	When a web server is designed to receive a request from a client without any mechanism for verifying that it was intentionally sent, then it might be possible for an attacker to trick a client into making an unintentional request to the web server which will be treated as an authentic request. This can be done via a URL, image load, XMLHttpRequest, etc. and can result in exposure of data or unintended code execution.
Business Impact	A successful CSRF attack can be devastating for both the business and user. It can result in damaged client relationships, unauthorized fund transfers, changed passwords and data theft—including stolen session cookies.
References	https://cwe.mitre.org/data/definitions/352.html

10. Directory Traversal

Vulnerability Name	Directory Traversal

CWE	22
SANS Category	10
Description	Directory traversal or file path traversal is a web security vulnerability that allows an attacker to read arbitrary files on the server that is currently running an application. These files could be an application code, credentials for back-end systems, and the operating system files. In some other scenario, an attacker might be able to write to these arbitrary files on the server which could allow them to modify application data or behavior, and this will give them total control of the server.
Business Impact	Directory traversal can lead to unauthorized access of sensitive information stored in files outside of the web root directory. This could include: system files, configuration files, or even user data.
References	https://cwe.mitre.org/data/definitions/22.html

11. OS Command Injection

Vulnerability Name	OS Command Injection
CWE	78
SANS Category	11
Description	It is about the improper sanitization of special elements that may lead to the modification of the intended OS command that is sent to a downstream component. An attacker can execute these malicious commands on a target operating system and can access
	an environment to which they were not supposed to read or modify. This invariably would allow an attacker to execute dangerous commands directly into the operating system.

Business Impact	**Technical Impact:** *Execute Unauthorized Code or Commands; DoS: Crash, Exit, or Restart; Read Files or Directories; Modify Files or Directories; Read Application Data; Modify Application Data; Hide Activities* Attackers could execute unauthorized commands, which could then be used to disable the product, or read and modify data for which the attacker does not have permissions to access directly. Since the targeted application is directly executing the commands instead of the attacker, any malicious activities may appear to come from the application or the application's owner.
References	https://cwe.mitre.org/data/definitions/78.html

12. Out-of-Bounds Write Error

Vulnerability Name	Out-of-Bounds Write Error
CWE	787
SANS Category	12
Description	This happens when the application writes data past the end, or before the beginning of the designated buffer. When this happens, the end result is usually data corruption, system, or application crash. What the application does is some sort of pointer arithmetic that is used in referencing a memory location outside the buffer boundaries.
Business Impact	Typically, this can result in corruption of data, a crash, or code execution. The product may modify an index or perform pointer arithmetic that references a memory location that is outside of the boundaries of the buffer. A subsequent write operation then produces undefined or unexpected results. **Technical Impact:** *Modify Memory; DoS: Crash, Exit, or Restart; Execute Unauthorized Code or Commands*
References	https://cwe.mitre.org/data/definitions/787.html

13. Improper Authentication Error

Vulnerability Name	Improper Authentication Error
CWE	287

SANS Category	13
Description	This is when an attacker claims to have a valid identity but the software failed to verify or proves that the claim is correct. A software validates a user's login information wrongly and as a result, an attacker could gain certain privileges within the application or disclose sensitive information that allows them to access sensitive data and execute arbitrary code.
Business Impact	This vulnerability can allow unauthorized access to the system, sensitive information or data, or allow attackers to perform malicious actions. Improper authentication can have serious consequences, including data theft, unauthorized access to sensitive information, and account takeover attacks. **Technical Impact:** *Read Application Data; Gain Privileges or Assume Identity; Execute Unauthorized Code or Commands* This weakness can lead to the exposure of resources or functionality to unintended actors, possibly providing attackers with sensitive information or even execute arbitrary code
References	https://cwe.mitre.org/data/definitions/287.html

14. Dereferencing a Null Pointer

Vulnerability Name	Dereferencing a Null Pointer
CWE	476
SANS Category	14
Description	Dereferencing a null pointer is when the application dereferences a pointer that was supposed to return a valid result instead returns NULL and this leads to a crash. Dereferencing a null pointer can happen through many flaws like race conditions and some programming error. The processes that are performed with the help of the NULL pointer usually lead to failure, and the possibility of carrying out the process is very slim. This helps attackers to execute malicious code.

Business Impact	Availability	**Technical Impact:** *DoS: Crash, Exit, or Restart* NULL pointer dereferences usually result in the failure of the process unless exception handling (on some platforms) is available and implemented. Even when exception handling is being used, it can still be very difficult to return the software to a safe state of operation.
	Integrity Confidentiality Availability	**Technical Impact:** *Execute Unauthorized Code or Commands; Read Memory; Modify Memory* In rare circumstances, when NULL is equivalent to the 0x0 memory address and privileged code can access it, then writing or reading memory is possible, which may lead to code execution.
References	https://cwe.mitre.org/data/definitions/476.html	

15. Incorrect Permission Assignment

Vulnerability Name	Incorrect Permission Assignment	
CWE	732	
SANS Category	15	
Description	This vulnerability happens when an application assigns permissions to a very important and critical resource in such a manner that exposed the resource to be accessed by a malicious user. When you give many people permission to a resource, this could lead to sensitive information being exposed or modified by an attacker. If there are no checks in place against this kind of approach to permission assignment to resources, it can lead to a very disastrous end if a program configuration or some sensitive data gets into the wrong hand.	
Business Impact	**Scope**	**Impact**

	Confidentiality	**Technical Impact:** *Read Application Data; Read Files or Directories* An attacker may be able to read sensitive information from the associated resource, such as credentials
		or configuration information stored in a file.
	Access Control	**Technical Impact:** *Gain Privileges or Assume Identity* An attacker may be able to modify critical properties of the associated resource to gain privileges, such as replacing a world-writable executable with a Trojan horse.
	Integrity Other	**Technical Impact:** *Modify Application Data; Other* An attacker may be able to destroy or corrupt critical data in the associated resource, such as deletion of records from a database.
References	https://cwe.mitre.org/data/definitions/732.html	

16. Unrestricted File Upload

Vulnerability Name	Unrestricted File Upload
CWE	434
SANS Category	16
Description	This vulnerability occurs when the application does not validate the file types before uploading files to the application. This vulnerability is language independent but usually occurs in applications written in ASP and PHP language. A dangerous type of file is a file that can be automatically processed within the application environment.
Business Impact	The consequences of unrestricted file upload can vary, including complete system takeover, an overloaded file system or database, forwarding attacks to back-end systems, client-side attacks, or simple defacement. It depends on what the application does with the uploaded file and especially where it is stored.
References	https://cwe.mitre.org/data/definitions/434.html

17. Information Exposure through XML Entities

<table>
<tr><td>Vulnerability Name</td><td>Information Exposure through XML Entities</td></tr>
<tr><td>CWE</td><td>611</td></tr>
<tr><td>SANS Category</td><td>17</td></tr>
<tr><td>Description</td><td>When an XML document is uploaded into an application for processing and this document contains XML entities with uniform</td></tr>
<tr><td></td><td>resource identifier that resolves to another document in another location different from the intended location. This anomaly can make the application to attach incorrect documents into its output.

The XML documents sometimes contain a Document Type Definition (DTD), which is used to define the XML entities and other features. Through the DTD, the uniform resource identifier can serve as a form of substitution string. What the XML parser will do is access what is contained in the uniform resource identifier and input these contents back into the XML document for execution.</td></tr>
<tr><td>Business Impact</td><td>
<table>
<tr><th>Scope</th><th>Impact</th></tr>
<tr><td>Confidentiality</td><td>Technical Impact: Read Application Data; Read Files or Directories

If the attacker is able to include a crafted DTD and a default entity resolver is enabled, the attacker may be able to access arbitrary files on the system.</td></tr>
<tr><td>Integrity</td><td>Technical Impact: Bypass Protection Mechanism

The DTD may include arbitrary HTTP requests that the server may execute. This could lead to other attacks leveraging the server's trust relationship with other entities.</td></tr>
</table>
</td></tr>
</table>

	Availability	**Technical Impact:** *DoS: Resource Consumption (CPU); DoS: Resource Consumption (Memory)* The product could consume excessive CPU cycles or memory using a URI that points to a large file, or a device that always returns data such as /dev/random. Alternately, the URI could reference a file that contains many nested or recursive entity references to further slow down parsing.
References	https://cwe.mitre.org/data/definitions/611.html	

18. Code Injection

Vulnerability Name	Code Injection	
CWE	94	
SANS Category	18	
Description	This vulnerability depicts a scenario where software allows untrusted data into the code and does not perform validation of special characters which can negatively influence both the behavior of the code segment and the syntax. In short, an attacker would be able to inject some sort of arbitrary code and execute them within the application.	
Business Impact	**Scope**	**Impact**
	Access Control	**Technical Impact:** *Bypass Protection Mechanism* In some cases, injectable code controls authentication; this may lead to a remote vulnerability.
	Access Control	**Technical Impact:** *Gain Privileges or Assume Identity* Injected code can access resources that the attacker is directly prevented from accessing.

	Integrity Confidentiality Availability	**Technical Impact:** *Execute Unauthorized Code or Commands* Code injection attacks can lead to loss of data integrity in nearly all cases as the control-plane data injected is always incidental to data recall or writing. Additionally, code injection can often result in the execution of arbitrary code.
	Non-Repudiation	**Technical Impact:** *Hide Activities* Often the actions performed by injected control code are unlogged.
References	https://cwe.mitre.org/data/definitions/94.html	

19. Hard Coded Access Key

Vulnerability Name	Hard Coded Access Key	
CWE	798	
SANS Category	19	
Description	This is when the password and access key is hard coded into the application directly for inbound authentication purpose and outbound communication to some external components and for encryption of internal data. Hard-coded login details usually cause vulnerability that paves the way for an attacker to bypass the authentication that has been configured by the software administrator. The system administrator will always find it very hard to detect this vulnerability and fix it.	
Business Impact	**Scope**	**Impact**
	Access Control	**Technical Impact:** *Bypass Protection Mechanism* If hard-coded passwords are used, it is almost certain that malicious users will gain access to the account in question.

	Integrity Confidentiality Availability Access Control Other	**Technical Impact:** *Read Application Data; Gain Privileges or Assume Identity; Execute Unauthorized Code or Commands; Other* This weakness can lead to the exposure of resources or functionality to unintended actors, possibly providing attackers with sensitive information or even execute arbitrary code.
References	https://cwe.mitre.org/data/definitions/798.html	

20. Uncontrolled Resource Consumption

Vulnerability Name	Uncontrolled Resource Consumption	
CWE	400	
SANS Category	20	
Description	This vulnerability happens when the application does not control the allocation properly and maintenance of a limited resource, this allows an attacker to be able to influence the amount of resources consumed, which will eventually lead to the exhaustion of available resources.Part of the limited resources includes memory, file system storage, database connection pool entries, and CPU.	
Business Impact	**Scope**	**Impact**
	Availability	**Technical Impact:** *DoS: Crash, Exit, or Restart; DoS: Resource Consumption (CPU); DoS: Resource Consumption (Memory); DoS: Resource Consumption (Other)* The most common result of resource exhaustion is denial of service. The product may slow down, crash due to unhandled errors, or lock out legitimate users.
	Access Control Other	**Technical Impact:** *Bypass Protection Mechanism; Other* In some cases it may be possible to force the product to "fail open" in the event of resource exhaustion. The state of the

		product -- and possibly the security functionality - may then be compromised.
References	https://cwe.mitre.org/data/definitions/400.html	