**Top 10 Web Application Security Risks**

**Open Web Application Security Project (OWASP)**



**Table 1: OWSAP - Top 10 Vulnerabilities**

|  |  |  |  |
| --- | --- | --- | --- |
| **SNo** | **Vulnerability Name** | **CWE – No** | **OWASP Category** |
| 1 | Broken Access Control | 284 | A01:2021 |
| 2 | Cryptographic Failures | 347 | A02:2021 |
| 3 | Injection | 94 | A03:2021 |
| 4 | Insecure Design | 657 | A04:2021 |
| 5 | Security Misconfiguration | 260 | A05:2021 |
| 6 | Vulnerable and Outdated Components | 1395 | A06:2021 |
| 7 | Identification and Authentication Failures | 287 | A07:2021 |
| 8 | Software and Data Integrity Failures | 353 | A08:2021 |
| 9 | Security Logging and Monitoring Failures | 778 | A09:2021 |
| 10 | Server-Side Request Forgery (SSRF) | 918 | A10:2021 |

1. **Vulnerability Name:- Broken Access Control**

**CWE: - 284 – Improper Access Control**

**OWASP Category:- A01:2021**

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

Business Impact:-

This represents a security weakness where software systems inadequately restrict access to resources or functionalities based on user privileges. The business impact of this vulnerability can be significant, posing risks such as unauthorized access to sensitive information, data breaches, and potential compromise of critical systems. Improper access control undermines the confidentiality and integrity of data, leading to reputational damage, loss of customer trust, and regulatory compliance issues. The financial implications may include the costs associated with remediation efforts, legal consequences, and the aftermath of security incidents. Organizations failing to address CWE-284 may experience operational disruptions, putting their overall business continuity at risk. Implementing robust access control mechanisms and adhering to security best practices are essential for mitigating the business impact associated with improper access control vulnerabilities.

1. **Vulnerability Name:- Cryptographic Failures**

**CWE: - 347 – Improper Verification of Cryptographic Signature**

**OWASP Category:- A02:2021**

**Description:-** The product does not verify, or incorrectly verifies, the cryptographic signature for data.

**Business Impact:-**

CWE-347 denotes a security weakness where software fails to adequately verify cryptographic signatures, allowing malicious actors to manipulate or forge digital signatures. The business impact of this weakness can be severe, potentially leading to unauthorized access, data integrity compromises, and the acceptance of maliciously altered software or data. Improper verification of cryptographic signatures undermines the trust and security of digital transactions, potentially exposing sensitive information, leading to reputational damage, financial losses, and legal consequences. Organizations failing to robustly verify cryptographic signatures may face regulatory non-compliance and increased vulnerability to cyberattacks, highlighting the critical importance of secure cryptographic practices to safeguard sensitive data and maintain the integrity of digital systems.

1. **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.

**Business Impact:-**

CWE-94 represents a security weakness where software applications inadequately regulate the generation or execution of code, potentially allowing attackers to inject and execute arbitrary code. The business impact of this vulnerability is profound, encompassing the risk of unauthorized access, data breaches, and compromise of critical systems. Exploitation of CWE-94 can lead to the execution of malicious activities, such as the installation of malware or the initiation of denial-of-service attacks, posing serious threats to the confidentiality, integrity, and availability of data. The financial consequences include the costs associated with remediation efforts, potential legal liabilities, and damage to the organization's reputation. Businesses failing to address CWE-94 may face operational disruptions, loss of customer trust, and regulatory compliance issues, emphasizing the critical need for secure coding practices and robust controls to mitigate the impact of improper code generation vulnerabilities.

1. **Vulnerability Name:- Insecure Design**

**CWE: - 657 – Use of Unsynchronized Data**

**OWASP Category:- A04:2021**

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

**Business Impact:-**

CWE-657 points to a security weakness where software applications lack proper synchronization mechanisms for managing data concurrently, potentially leading to data corruption or inconsistent states. The business impact of this vulnerability is substantial, encompassing the risk of compromised data integrity, inaccurate decision making, and operational disruptions. Unsynchronized data access can result in security vulnerabilities, with attackers potentially exploiting race conditions to gain unauthorized access or disrupt systems. The financial implications include the costs associated with data recovery, system downtime, and potential legal consequences for non-compliance with data protection regulations. The organization's reputation may suffer due to perceived unreliability and inadequate data security practices. Mitigating the business impact of CWE-657 requires adopting secure coding practices, implementing effective synchronization mechanisms, and conducting regular assessments to identify and address unsynchronized data issues in software applications.

1. **Vulnerability Name:- Security Misconfiguration**

**CWE: - 260 - Direct Reconfiguration of Control Flow**

**OWASP Category:- A05:2021**

**Description:-** The product stores a password in a configuration file that might be accessible to actors who do not know the password.

**Business Impact:-**

CWE-260 represents a critical security weakness where software systems lack proper controls to prevent unauthorized changes to the control flow of an application, allowing attackers to manipulate program execution. The business impact of this vulnerability is profound, as it can lead to unauthorized access, data breaches, and the compromise of critical systems. Exploitation of CWE-260 can result in the execution of arbitrary code, potentially causing significant disruptions to system functionality and posing risks to the confidentiality and integrity of sensitive data. The financial consequences encompass the costs associated with remediation efforts, potential legal liabilities, and reputational damage. Businesses failing to address CWE-260 may face operational disruptions, loss of customer trust, and regulatory compliance issues, highlighting the urgent need for secure coding practices and robust controls to mitigate the impact of direct control flow manipulation vulnerabilities.

1. **Vulnerability Name:- Vulnerable and Outdated Components**

**CWE: - 1395 - Dependency on Vulnerable Third-Party Component**

**OWASP Category:- A06:2021**

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

**Business Impact:-**

CWE-1395 signifies a security weakness where a software project relies on a third-party component with known vulnerabilities. The business impact of this vulnerability can be severe, encompassing the risk of data breaches, unauthorized access, and compromised system integrity. Depending on the nature of the vulnerable component, exploitation could lead to financial losses, regulatory non-compliance, and reputational damage. The organization may incur significant costs associated with investigating and addressing the vulnerabilities, implementing security measures, and potentially compensating affected parties. Additionally, reliance on vulnerable third-party components may result in operational disruptions, legal consequences, and a loss of customer trust. Mitigating the business impact requires proactive monitoring of third-party components, timely updates, and thorough assessments to ensure the security resilience of the software ecosystem.

1. **Vulnerability Name:- Identification and authentication failures**

**CWE: - 287 - Improper Authentication**

**OWASP Category:- A07:2021**

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

**Business Impact:-**

CWE-287 represents a security weakness where software fails to implement robust authentication mechanisms. The business impact of this vulnerability is significant, as it can lead to unauthorized access, data breaches, and compromise of sensitive information. Insufficient authentication measures may result in financial losses, reputational damage, and regulatory compliance issues. Attackers exploiting this weakness may gain unauthorized privileges, posing risks to the confidentiality and integrity of critical data. The organization may face legal consequences and a loss of customer trust if authentication vulnerabilities are not promptly addressed. Mitigation involves implementing strong authentication practices, such as multi-factor authentication, and regularly updating and testing authentication mechanisms to ensure the security of access controls.

1. **Vulnerability Name:- Software and data integrity failures**

**CWE: - 353 - Missing Support for Integrity Check**

**OWASP Category:- A08:2021**

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

**Business Impact:-**

CWE-353 highlights a security weakness where a system lacks proper mechanisms to verify the integrity of data. The business impact of this vulnerability is substantial, encompassing the risk of data corruption, incorrect decision-making, and potential reputational damage. Without adequate integrity checks, organizations may face operational disruptions, delayed incident response, and increased dwell time for security breaches. Compliance violations, financial losses, and challenges in forensic analysis are also potential consequences. To mitigate the impact, organizations should implement comprehensive logging, regular audits, and automated alerting systems, ensuring the maintenance of data integrity and a proactive response to potential security incidents.

1. **Vulnerability Name:- Security logging and monitoring failures**

**CWE: - 778 - Insufficient Logging**

**OWASP Category:- A09:2021**

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

**Business Impact:-**

CWE-778 denotes a security weakness where software inadequately logs security-relevant events. The business impact of this vulnerability is considerable, leading to challenges in incident detection, delayed response times, and difficulties in forensic analysis. Insufficient logging can contribute to reputation damage, financial losses, and increased dwell time for security incidents. Compliance issues, ineffective auditing, and limitations in legal evidence may also arise. Organizations failing to address this weakness may experience prolonged recovery times and face regulatory non-compliance. To mitigate the impact, it is crucial to implement comprehensive logging, automated alerting systems, and regular log reviews for early detection and effective response to security events.

1. **Vulnerability Name:- Server side Request forgery**

**CWE: - 918 -** **Server-Side Request Forgery (SSRF)**

**OWASP Category:- A10:2021**

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

**Business Impact:-**

CWE-918 represents a security weakness where an attacker can manipulate a server into making unintended requests. The business impact of this vulnerability is significant, including unauthorized access to internal resources, data exposure, and potential service disruption through denial of service attacks. Successful exploitation can lead to reputation damage, financial losses, and regulatory compliance violations. SSRF can serve as a pivot point for further attacks within internal networks, increasing the overall risk. Mitigating the impact involves implementing input validation, network segmentation, and strict access controls to prevent unauthorized access and potential abuse of server-side request forgery vulnerabilities.

**SANS Top 20 Critical Vulnerabilities In Software**

The word**SANS :** **SysAdmin**, **Audit**, **Network**, and **Security**.

**Table 2: SANS - Top 20 Critical Vulnerabilities**

|  |  |  |
| --- | --- | --- |
| **SNo** | **Vulnerability Name** | **CWE – No** |
| **1** | Memory Buffer Error | **119** |
| **2** | Cross-site Scripting | **79** |
| **3** | Unvalidated Input Error | **20** |
| **4** | Sensitive Information Exposure Error | **200** |
| **5** | Out-of-bounds Read Error | **125** |
| **6** | SQL Injection | **89** |
| **7** | Free Memory Error | **416** |
| **8** | Integer Overflow Error | **190** |
| **9** | Cross-Site Request Forgery | **352** |
| **10** | Directory Traversal | **22** |
| **11** | OS Command Injection | **78** |
| **12** | Out-of-bounds Write Error | **787** |
| **13** | Improper Authentication Error | **287** |
| **14** | Dereferencing NULL Pointer | **476** |
| **15** | Incorrect Permission Assignment | **732** |
| **16** | Unrestricted File Upload | **434** |
| **17** | Information Exposure through XML Entities | **611** |
| **18** | Code Injection | **94** |
| **19** | Hard-coded Access Key | **798** |
| **20** | Uncontrolled Resource Consumption | **400** |

1. **Name:- Memory Buffer Error**

**CWE:- 119**

**SANS Category:- 1**

**Description:-** The product performs operations on a memory buffer, but it can read from or write to a memory location that is outside of the intended boundary of the buffer.

**Business Impact:-**

CWE-119, focusing on Memory Buffer Errors, poses significant business impact risks due to its potential to compromise the integrity, availability, and security of software systems. Memory buffer errors, such as buffer overflows or underflows, can lead to unexpected behaviours, system crashes, and, critically, security vulnerabilities that may be exploited by malicious actors. The exploitation of these vulnerabilities can result in unauthorized access, data breaches, and even remote code execution, placing sensitive business information at risk. In addition to the potential compromise of data confidentiality, the stability of applications may be compromised, leading to service disruptions and potential financial losses. Businesses face reputational damage as well, as customers and stakeholders may lose trust in the security of the affected systems.

1. **Name:- Cross-site Scripting**

**CWE:- 79**

**SANS Category:- 2**

**Description:-** The product does not neutralize or incorrectly neutralizes user-controllable input before it is placed in output that is used as a web page that is served to other users.

**Business Impact:-**

CWE-79, which addresses Cross-Site Scripting (XSS), has substantial business impact implications due to its potential to compromise the security and trustworthiness of web applications. XSS vulnerabilities can be exploited by attackers to inject malicious scripts into web pages, enabling them to steal sensitive user data, hijack sessions, deface websites, and propagate phishing attacks. The business consequences include reputational damage, loss of customer trust, and legal ramifications if sensitive information is compromised. Beyond the direct impact on users, XSS attacks can also disrupt normal business operations by causing service outages or making websites unreliable. To mitigate CWE-79, businesses must prioritize secure coding practices, implement effective input validation, and utilize security mechanisms like Content Security Policy (CSP) to prevent the execution of malicious scripts, thereby safeguarding both their users and the overall integrity of their web applications.

1. **Name: Unvalidated Input Error**

**CWE:- 20**

**SANS Category:- 3**

**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.

**Business Impact:-**

CWE-20, focusing on Unvalidated Input Errors, presents a substantial business impact by exposing systems to potential security vulnerabilities arising from inadequate input validation. This weakness allows attackers to input malicious data into applications, leading to various consequences such as unauthorized access, data manipulation, and potential compromise of sensitive information. Business operations may be significantly affected as unvalidated input errors can be exploited to execute arbitrary code, disrupt services, or compromise the confidentiality and integrity of critical data. Beyond the direct financial and operational implications, the reputational damage resulting from security breaches can erode customer trust and confidence in the affected systems. To mitigate CWE-20, businesses need to implement thorough input validation checks, employ secure coding practices, and conduct regular security assessments to identify and address unvalidated input errors, thereby fortifying their systems against potential exploits and safeguarding the overall security posture of the organization.

1. **Name:- Sensitive Information Exposure Error**

**CWE:- 200**

**SANS Category:- 4**

**Description:-** The product exposes sensitive information to an actor that is not explicitly authorized to have access to that information.

**Business Impact:-**

CWE-200, addressing Sensitive Information Exposure Errors, poses significant business impact risks by potentially exposing confidential data and compromising the privacy of users. This vulnerability occurs when sensitive information is unintentionally disclosed, either through inadequate data protection mechanisms or insecure coding practices. The consequences for businesses can be severe, including reputational damage, legal ramifications, and financial losses. Customer trust is particularly at stake, as the exposure of sensitive data can lead to breaches, identity theft, and regulatory non-compliance. Moreover, businesses may face operational disruptions as they work to contain and remediate the impact of the exposure. To mitigate CWE-200, organizations should adopt robust encryption practices, implement access controls, conduct regular security assessments, and follow privacy best practices to ensure the protection of sensitive information, ultimately safeguarding both their reputation and the trust of their user base.

1. **Name:- Out-of-bounds Read Error**

**CWE:- 125**

**SANS Category:- 5**

**Description:-** The product reads data past the end, or before the beginning, of the intended buffer.

**Business Impact:-**

CWE-125, focusing on Out-of-bounds Read Errors, presents a substantial business impact by potentially compromising the integrity, availability, and security of software systems. This vulnerability arises when a program reads data beyond the boundaries of an allocated memory buffer, leading to unintended consequences such as the exposure of sensitive information, system crashes, or unauthorized access. The business consequences include the potential compromise of critical data, loss of system availability, and reputational damage. Exploitation of out-of-bounds read errors can provide attackers with unauthorized insights into system memory, leading to security breaches and the potential execution of malicious code. Mitigating CWE-125 requires diligent coding practices, thorough input validation, and regular security testing to identify and address out-of-bounds read errors early in the development lifecycle, thereby fortifying the business against these serious security risks.**Top of Form**

1. **Name: SQL Injection**

**CWE:- 89**

**SANS Category:- 6**

**Description:-** The product reads data past the end, or before the beginning, of the intended buffer.

**Business Impact:-**

CWE-89, addressing SQL Injection vulnerabilities, carries significant business impact risks by exposing databases to unauthorized access and manipulation. This weakness occurs when user-supplied input is improperly sanitized, allowing attackers to inject malicious SQL code into queries. The consequences for businesses are severe, encompassing data breaches, unauthorized access to sensitive information, and potential manipulation or deletion of critical data. SQL Injection attacks can lead to the compromise of customer data, financial records, and other confidential information, resulting in reputational damage, legal liabilities, and financial losses. Beyond the immediate impact, organizations may face regulatory consequences for failing to protect sensitive data. Mitigating CWE-89 necessitates the implementation of secure coding practices, parameterized queries, and input validation to prevent the injection of malicious SQL code, thereby safeguarding the integrity and security of databases and mitigating the broader business risks associated with SQL Injection vulnerabilities.

1. **Name: Free Memory Error**

**CWE: -416**

**SANS Category:- 7**

**Description:-** Referencing memory after it has been freed can cause a program to crash, use unexpected values, or execute code.

**Business Impact:-**

CWE-416, which pertains to Free Memory Errors, presents a notable business impact risk by potentially compromising the stability, performance, and security of software systems. This vulnerability occurs when a program attempts to free memory that has already been deallocated or was never allocated, leading to unpredictable behavior, crashes, and potential exploitation by attackers. The business consequences include system instability, service disruptions, and the risk of unauthorized access or execution of malicious code. Free Memory Errors can result in application failures and downtime, affecting user experience and potentially leading to financial losses. Mitigating CWE-416 requires thorough memory management practices, proper handling of memory deallocation, and diligent code reviews to identify and rectify free memory errors early in the development process, thereby fortifying the business against the serious operational and security risks associated with this vulnerability.

1. **Name: Integer Overflow Error**

**CWE:- 190**

**SANS Category:- 8**

**Description:-** The product performs a calculation that can produce an integer overflow or wraparound, when the logic assumes that the resulting value will always be larger than the original value. This can introduce other weaknesses when the calculation is used for resource management or execution control.

**Business Impact:-**

CWE-190, addressing Integer Overflow Errors, poses a notable business impact risk by potentially compromising the integrity, functionality, and security of software systems. This vulnerability occurs when an arithmetic operation results in an integer value that exceeds the maximum representable value for its data type, leading to unexpected behavior, crashes, and potential security vulnerabilities. The business consequences include the potential for miscalculations, data corruption, and, critically, the risk of exploitation by attackers to execute arbitrary code or manipulate system behavior. Integer Overflow Errors can have broad implications, impacting financial transactions, data processing, and overall system reliability. To mitigate CWE-190, businesses need to employ secure coding practices, validate inputs, and conduct thorough code reviews to identify and address potential integer overflow vulnerabilities early in the development lifecycle, thereby fortifying the software against the serious operational and security risks associated with this type of error.

1. **Name: Cross-Site Request Forgery**

**CWE:- 352**

**SANS Category:- 9**

**Description:-** The web application does not, or can not, sufficiently verify whether a well-formed, valid, consistent request was intentionally provided by the user who submitted the request.

**Business Impact:-**

CWE-352, focusing on Cross-Site Request Forgery (CSRF), introduces a significant business impact risk by enabling attackers to force users into unknowingly executing unauthorized actions on a web application in which they are authenticated. This vulnerability arises when an attacker tricks a user's browser into making an unintended and unauthorized request on their behalf. The potential consequences for businesses include unauthorized transactions, data manipulation, and account compromise. CSRF attacks can lead to financial losses, reputational damage, and legal liabilities, as user trust in the security of the affected web application is undermined. Mitigating CWE-352 involves implementing anti-CSRF tokens, securing sensitive actions with additional authentication, and ensuring proper session management to prevent attackers from exploiting users' authenticated sessions for malicious purposes, thereby fortifying the business against the serious risks associated with CSRF vulnerabilities.

1. **Name: Directory Traversa**

**CWE:- 22**

**SANS Category:- 10**

**Description:-** 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.

**Business Impact:-**

CWE-22, addressing Directory Traversal vulnerabilities, poses a notable business impact risk by potentially exposing sensitive files and compromising the confidentiality and integrity of data within a system. This weakness occurs when an application allows an attacker to navigate outside the intended directory, accessing files and directories they should not be able to reach. The business consequences include unauthorized disclosure of sensitive information, potential data manipulation, and, in some cases, the execution of malicious code. Directory Traversal vulnerabilities can lead to reputational damage, regulatory non-compliance, and legal ramifications if confidential or proprietary information is compromised. To mitigate CWE-22, businesses should implement strict input validation, employ secure file access mechanisms, and configure proper access controls to prevent unauthorized directory traversals, thereby fortifying their systems against the serious security and operational risks associated with this vulnerability.

1. **Name: OS Command Injection**

**CWE:- 78**

**SANS Category:- 11**

**Description:-** The product constructs all or part of an OS command using externally-influenced input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could modify the intended OS command when it is sent to a downstream component.

**Business Impact:-**

CWE-78, which focuses on OS Command Injection vulnerabilities, introduces a significant business impact risk by enabling attackers to execute arbitrary commands on a system. This vulnerability arises when an application incorporates user-supplied input into a command that is sent to the operating system. The potential consequences for businesses include unauthorized access, data manipulation, and the execution of malicious operations. OS Command Injection attacks can lead to system compromise, unauthorized data disclosure, and disruption of critical operations. The business impact extends to reputational damage, financial losses, and legal liabilities, particularly if sensitive information is exposed. To mitigate CWE-78, organizations need to adopt secure coding practices, validate and sanitize user input, and implement proper command execution mechanisms to prevent attackers from injecting and executing arbitrary commands, thereby fortifying their systems against the serious risks associated with OS Command Injection vulnerabilities.

1. **Name: Out-of-bounds Write Error**

**CWE:- 787**

**SANS Category:- 12**

**Description:-** The product constructs all or part of an OS command using externally-influenced input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could modify the intended OS command when it is sent to a downstream component.

**Business Impact:-**

CWE-787, addressing Out-of-bounds Write Errors, presents a significant business impact risk by potentially compromising the stability, security, and functionality of software systems. This vulnerability occurs when a program writes data beyond the boundaries of an allocated memory buffer, leading to unintended consequences such as system crashes, data corruption, and potential exploitation by attackers. The business consequences include the compromise of critical data, service disruptions, and the risk of unauthorized access or execution of malicious code. Out-of-bounds Write Errors can result in application failures, impacting user experience and potentially leading to financial losses. To mitigate CWE-787, businesses need to enforce robust memory management practices, conduct thorough code reviews, and implement secure coding techniques to identify and rectify out-of-bounds write errors early in the development process. This helps fortify the software against the serious operational and security risks associated with this vulnerability, ensuring the overall reliability and security of the organization's systems.

1. **Name: Improper Authentication Error**

**CWE:- 287**

**SANS Category:- 13**

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

**Business Impact:-**

CWE-287, addressing Improper Authentication Errors, poses a significant business impact risk by exposing systems to unauthorized access and potentially compromising sensitive information. This vulnerability occurs when authentication mechanisms are not implemented or configured correctly, allowing attackers to gain unauthorized entry to applications, systems, or data. The potential consequences for businesses include unauthorized access to critical resources, data breaches, and the compromise of sensitive information. Improper Authentication Errors can lead to reputational damage, financial losses, and regulatory non-compliance, particularly if personally identifiable or confidential information is accessed illicitly. Mitigating CWE-287 involves implementing strong authentication practices, including multi-factor authentication, proper credential management, and rigorous access controls to prevent unauthorized access attempts, thereby fortifying systems against the serious operational and security risks associated with improper authentication vulnerabilities.

1. **Name: Dereferencing NULL Pointer**

**CWE:- 476**

**SANS Category:- 14**

**Description:-** A NULL pointer dereference occurs when the application dereferences a pointer that it expects to be valid, but is NULL, typically causing a crash or exit..

**Business Impact:-**

CWE-476, focused on Dereferencing NULL Pointer vulnerabilities, presents a substantial business impact risk by potentially causing system crashes, unexpected behavior, and security vulnerabilities. This weakness occurs when a program attempts to access or manipulate data through a pointer that has not been assigned a valid memory address (NULL pointer). The consequences for businesses include service disruptions, application failures, and potential exploitation by attackers to execute arbitrary code or disrupt critical operations. Dereferencing NULL Pointer errors can lead to downtime, impacting user experience and potentially resulting in financial losses. To mitigate CWE-476, businesses must employ robust coding practices, validate and initialize pointers appropriately, and conduct thorough code reviews to identify and rectify null pointer dereference issues early in the development process. This helps fortify the software against the serious operational and security risks associated with this vulnerability, ensuring the overall reliability and security of the organization's systems.

1. **Name: Incorrect Permission Assignment**

**CWE:- 732**

**SANS Category:- 15**

**Description:-** The product specifies permissions for a security-critical resource in a way that allows that resource to be read or modified by unintended actors.

**Business Impact:-**

CWE-732, addressing Incorrect Permission Assignment vulnerabilities, introduces a significant business impact risk by potentially exposing sensitive data and compromising the integrity and confidentiality of systems. This weakness occurs when permissions are improperly assigned, allowing unauthorized access to critical resources. The potential consequences for businesses include unauthorized disclosure or manipulation of sensitive information, data breaches, and the compromise of system functionality. Incorrect Permission Assignment can lead to reputational damage, regulatory non-compliance, and legal liabilities, particularly if customer or proprietary data is accessed illicitly. To mitigate CWE-732, organizations must implement robust access control mechanisms, follow the principle of least privilege, and conduct regular reviews of permission assignments to ensure that only authorized individuals or processes have the necessary access, thereby fortifying systems against the serious operational and security risks associated with this vulnerability.

1. **Name: Unrestricted File Upload**

**CWE:- 434**

**SANS Category:- 16**

**Description:-** The product allows the attacker to upload or transfer files of dangerous types that can be automatically processed within the product's environment.

**Business Impact:-**

CWE-434, focusing on Unrestricted File Upload vulnerabilities, presents a critical business impact risk by potentially allowing malicious actors to upload and execute arbitrary files on a system. This weakness arises when an application does not properly validate and restrict the types and locations of files that users can upload. The consequences for businesses include the potential for unauthorized execution of malicious code, data manipulation, and the compromise of system integrity. Unrestricted File Upload vulnerabilities can lead to service disruptions, unauthorized access, and reputational damage. Beyond the immediate operational impact, organizations may face legal liabilities and regulatory consequences, especially if sensitive or confidential information is affected. To mitigate CWE-434, businesses must implement strict file upload controls, validate file types and contents, and employ secure storage practices to prevent unauthorized uploads and execution of malicious files, thereby fortifying systems against the serious security and operational risks associated with this vulnerability.

1. **Name: Information Exposure through XML Entities**

**CWE:-611**

**SANS Category:- 17**

**Description:-** The product processes an XML document that can contain XML entities with URIs that resolve to documents outside of the intended sphere of control, causing the product to embed incorrect documents into its output.

**Business Impact:-**

CWE-611, addressing Information Exposure through XML Entities, introduces a substantial business impact risk by potentially exposing sensitive information through unintended access to external entities in XML documents. This vulnerability arises when an application processes XML data without proper validation, allowing attackers to include external entities that may disclose confidential data or perform unauthorized actions. The potential consequences for businesses include the compromise of sensitive information, data breaches, and unauthorized access to critical systems. Information Exposure through XML Entities can lead to reputational damage, legal consequences, and financial losses, particularly if personally identifiable information or proprietary data is exposed. To mitigate CWE-611, organizations must implement secure coding practices, use proper XML processing libraries, and validate and sanitize input to prevent unintended access to external entities, thereby fortifying systems against the serious operational and security risks associated with this vulnerability.

1. **Name:** Code Injection

**CWE:- 94**

**SANS Category:- 18**

**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.

**Business Impact:-**

CWE-94, addressing Code Injection vulnerabilities, presents a significant business impact risk by allowing attackers to inject and execute arbitrary code within a software application. This weakness arises when untrusted data is improperly processed by the application, leading to the execution of unintended commands. The potential consequences for businesses include unauthorized access to sensitive data, manipulation of system behavior, and the potential for executing malicious operations. Code Injection vulnerabilities can result in service disruptions, data breaches, and reputational damage. Beyond the immediate operational impact, organizations may face legal and regulatory consequences, particularly if customer data is compromised. To mitigate CWE-94, businesses must adopt secure coding practices, validate and sanitize input data rigorously, and employ proper input validation mechanisms to prevent the injection of arbitrary code, thereby fortifying systems against the serious security and operational risks associated with this vulnerability.

1. **Name:** **Hard-coded Access Key**

**CWE:- 798**

**SANS Category:- 19**

**Description:-** The product contains hard-coded credentials, such as a password or cryptographic key, which it uses for its own inbound authentication, outbound communication to external components, or encryption of internal data.

**Business Impact:-**

CWE-798, addressing Hard-coded Access Key vulnerabilities, presents a significant business impact risk by embedding access credentials directly into software code. This weakness occurs when sensitive authentication keys, tokens, or passwords are hardcoded, making them easily accessible to attackers. The potential consequences for businesses include unauthorized access to sensitive systems or data, leading to data breaches, unauthorized actions, and potential compromise of critical assets. Hard-coded Access Key vulnerabilities can result in reputational damage, financial losses, and legal liabilities, particularly if customer or proprietary information is compromised. To mitigate CWE-798, organizations must adopt secure coding practices, use secure credential management solutions, and avoid hardcoding sensitive access keys in software code to prevent unauthorized access, thereby fortifying systems against the serious operational and security risks associated with this vulnerability.

1. **Name: Uncontrolled Resource Consumption**

**CWE:- 400**

**SANS Category:- 20**

**Description:-** The product does not properly control the allocation and maintenance of a limited resource, thereby enabling an actor to influence the amount of resources consumed, eventually leading to the exhaustion of available resources.

**Business Impact:-**

CWE-400, addressing Uncontrolled Resource Consumption, presents a significant business impact risk by potentially allowing attackers to exhaust system resources, leading to service disruptions and denial-of-service (DoS) conditions. This vulnerability arises when an application or system does not properly limit the consumption of resources, such as CPU, memory, or network bandwidth. The potential consequences for businesses include degraded system performance, unresponsiveness, and potential downtime. Uncontrolled Resource Consumption vulnerabilities can result in the disruption of critical services, impacting user experience and potentially leading to financial losses. Beyond the immediate operational impact, organizations may face reputational damage and legal consequences if service-level agreements are violated. To mitigate CWE-400, businesses must implement resource usage controls, conduct thorough performance testing, and apply proper resource management practices to prevent abuse and ensure the stability and reliability of their systems, thereby fortifying against the serious operational and security risks associated with this vulnerability.