The OWASP (Open Web Application Security Project) Top 10 is a regularly updated list of the most critical web application security risks. As of my last knowledge update in September 2021, here are the OWASP Top 10 vulnerabilities along with a brief description and their risk points:

**A01:2021 - Broken Access Control**

* **Description:** Broken Access Control occurs when an application fails to properly enforce access restrictions, allowing users to perform actions or access data they should not have permission for. This can lead to unauthorized data exposure, account takeover, and other security breaches.
* **Examples:** Accessing another user's account, modifying data without proper authorization, and bypassing authentication mechanisms.
* **Mitigation:** Implement strong access controls, use proper authentication and authorization mechanisms, and regularly test for access control issues.
* **Business Impact:**
* Unauthorized access to sensitive data or functionality.
* Data breaches, leading to legal and regulatory consequences.
* Loss of customer trust and reputation damage.
* Financial losses due to data theft or fraud.

**A02:2021 - Cryptographic Failures**

* **Description:** Cryptographic Failures relate to issues in cryptographic implementations, which can lead to sensitive data exposure or system compromise. Weak encryption, improper key management, and insecure cryptographic practices fall into this category.
* **Examples:** Storing sensitive data without proper encryption, using weak encryption algorithms, and mishandling encryption keys.
* **Mitigation:** Use strong encryption algorithms, follow best practices for key management, and regularly review cryptographic implementations.
* **Business Impact:**
* Data confidentiality and integrity breaches.
* Loss of sensitive information.
* Legal and regulatory penalties for data protection violations.
* Reputation damage and loss of customer trust.
* Potential financial losses in case of data breaches.

**A03:2021 - Injection**

* **Description:** Injection vulnerabilities occur when untrusted data is included in a command or query, allowing attackers to execute arbitrary code or access unauthorized data. Common injection types include SQL Injection and Command Injection.
* **Examples:** Submitting malicious SQL queries to manipulate a database or injecting OS commands to gain control over the server.
* **Mitigation:** Implement input validation and proper parameterized queries, avoid dynamic code execution with user input, and use prepared statements.
* **Business Impact:**
* Data breaches or data manipulation.
* Unauthorized access to systems or data.
* Potential legal consequences for exposing customer data.
* Disruption of services due to system compromise.
* Damage to reputation and customer trust.

**A04:2021 - Insecure Design (New)**

* **Description:** Insecure Design focuses on design flaws in applications that can lead to vulnerabilities. These flaws can be architectural in nature and can significantly impact an application's security.
* **Examples:** Poorly designed authentication flows, lack of threat modeling, and insecure data flow.
* **Mitigation:** Implement secure design patterns, perform threat modelling, and follow secure architecture principles from the start of development.
* **Business Impact:**
* Increased security risks throughout the application's lifecycle.
* Costly security vulnerabilities discovered late in the development cycle.
* Delays in product release and increased development costs.
* Increased post-release maintenance and security patching.
* Reputation damage if security issues become public.

**A05:2021 - Security Misconfiguration**

* **Description:** Security Misconfiguration arises from improper configuration settings in applications or servers. This can expose sensitive data, APIs, and other critical resources to unauthorized access.
* **Examples:** Leaving default passwords, overly permissive permissions, and exposing unnecessary services to the internet.
* **Mitigation:** Regularly audit and review configurations, follow security best practices for server and application configurations, and use security hardening guides.
* **Business Impact:**
* Exposure of sensitive data or functionality due to misconfigured settings.
* Unauthorized access to systems or data.
* Legal and regulatory penalties for not securing sensitive information.
* Service disruptions and downtime due to misconfigurations.
* Reputational damage and loss of customer trust.

**A06:2021 - Vulnerable and Outdated Components**

* **Description:** This category focuses on the use of vulnerable or outdated third-party components in applications. It emphasizes the importance of keeping libraries and dependencies up to date.
* **Examples:** Using a third-party library with known vulnerabilities, failing to update components with security patches, and using deprecated software.
* **Mitigation:** Keep third-party components updated, use security monitoring tools to identify vulnerabilities, and have a process for patching or replacing vulnerable components.

**A07:2021 - Identification and Authentication Failures**

* **Description:** Identification and Authentication Failures pertain to issues related to user identification and authentication processes. This includes issues with weak or flawed authentication mechanisms.
* **Examples:** Weak password policies, inadequate account lockout mechanisms, and poor session management.
* **Mitigation:** Implement strong authentication mechanisms, use multi-factor authentication where applicable, and enforce proper session management practices.

**A08:2021 - Software and Data Integrity Failures (New)**

* **Description:** This new category emphasizes the importance of verifying the integrity of software updates, critical data, and CI/CD pipelines. It highlights the risks associated with making assumptions without validation.
* **Examples:** Trusting software updates without validation, assuming data integrity without verification, and relying on insecure CI/CD pipelines.
* **Mitigation:** Implement mechanisms to verify software and data integrity, use secure CI/CD practices, and avoid making unverified assumptions.

**A09:2021 - Security Logging and Monitoring Failures**

* **Description:** This category includes various types of failures related to security logging and monitoring practices. Failures in this category can impact an organization's ability to detect and respond to security incidents.
* **Examples:** Inadequate event logging, lack of monitoring for suspicious activities, and insufficient incident alerting.
* **Mitigation:** Implement comprehensive logging and monitoring solutions, define clear incident response procedures, and regularly review and analyze logs for security events.

**A10:2021 - Server-Side Request Forgery (New)**

* **Description:** Server-side request Forgery (SSRF) involves an attacker manipulating an application into making unauthorized requests to internal or external resources. This can lead to data exposure, service disruption, or remote code execution.
* **Examples:** Abusing an application's functionality to access internal resources, causing a server to make requests to malicious external sites, or bypassing security controls.
* **Mitigation:** Implement proper input validation and access controls, restrict access to sensitive internal resources, and validate and sanitize user input to prevent SSRF attacks.