GitProtect.io
by Xopero ONE

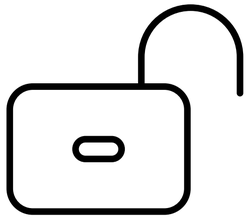
OWASP Top 10 Vulnerabilities List

Your handy security checklist

OWASP Top 10 Vulnerabilities List

A01:2021	**Broken Access Control**
A02:2021	**Cryptographic Failures**
A03:2021	**Injection**
A04:2021	**Insecure Design**
A05:2021	**Security Misconfiguration**
A06:2021	**Vulnerable and Outdated Components**
A07:2021	**Identification and Authentication Failures**
A08:2021	**Software and Data Integrity Failures**
A09:2021	**Security Logging and Monitoring Failures**
A10:2021	**Server-Side Request Forgery (SSRF)**

GitProtect.io
by Xopero ONE



A01:2021

Broken Access Control

How to prevent?

- [ ] Deny access by default
- [ ] Implement proper access controls throughout the application
- [ ] Restrict access to APIs and controllers

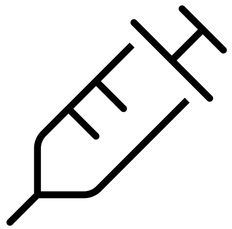
01
10

A02:2021

Cryptographic Failures

How to prevent?

- ☐ Categorize information that an application processes, stores, or transmits.
- ☐ Follow the necessary security measures based on the classification of the data
- ☐ Encrypt all sensitive data that is saved



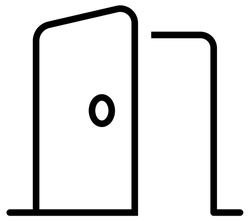
A03:2021

Injection

How to prevent?

- [ ] Harness the Strength of Secure APIs
- [ ] Enforce Whitelist Validation
- [ ] LIMIT and other SQL controls act as powerful guardians against injection attacks

GitProtect.io
by Xopero ONE



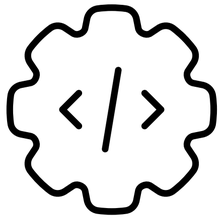
A04:2021

Insecure Design

How to prevent?

- [ ] Implement Secure SDLC
- [ ] Leverage Threat Modeling and Secure Patterns
- [ ] Integrate Security into User Stories

GitProtect.io
by Xopero ONE



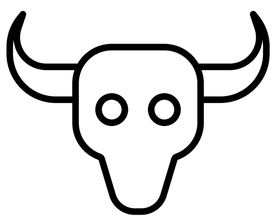
A05:2021

Security Misconfiguration

How to prevent?

- [ ] Establish a repeatable security hardening process, preferably automated
- [ ] Remove unused or unnecessary features, components, and files
- [ ] Implement an automated process to review and maintain security settings across environments

GitProtect.io
by Xopero ONE



A06:2021

Vulnerable and Outdated Components

GitProtect.io
by Xopero ONE

How to prevent?

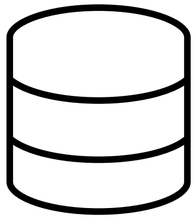
- [ ] Remove unused or unnecessary libraries, components, frameworks, documentation, and files from the application
- [ ] Maintain an inventory of both server-side and client-side components and regularly monitor for updates and vulnerabilities
- [ ] Use official libraries and sources through secure links
- [ ] Monitor for unsupported libraries and components that are no longer maintained or have reached the end of life

A07:2021

Identification and Authentication Failures

How to prevent?

- [ ] Implement multi-factor authentication to add an extra layer of security.
- [ ] Avoid using default credentials, especially for administrative accounts
- [ ] Take steps to limit account enumeration, making it difficult for attackers to determine valid usernames

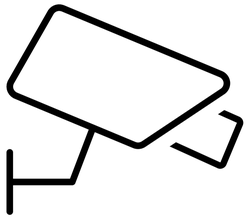


A08:2021

Software and Data Integrity Failures

How to prevent?

- [ ] Use digital signatures or other verification methods to ensure software updates originate from trusted sources and arrive intact
- [ ] Verify that third-party libraries and dependencies come from legitimate sources
- [ ] Use automated security tools designed for the software supply chain to scan for vulnerabilities in third-party resources
- [ ] Implement secure deserialization practices to prevent code execution vulnerabilities



A09:2021

Security Logging and Monitoring Failures

GitProtect.io
by Xopero ONE

How to prevent?

- ☐ Implement comprehensive security logging and monitoring across applications
- ☐ Log important events with user context to preserve evidence of malicious or suspicious activity
- ☐ Generate logs in a format compatible with log management tools.
- ☐ Enable monitoring and alerting for suspicious activities
- ☐ Develop an incident response and mitigation plan to respond effectively to security breaches

A10:2021

Server-Side Request Forgery (SSRF)

Preventing SSRF requires implementing protection measures at both the network and application levels.

Network-level prevention:

- [ ] Utilize network segmentation to separate remote resources and sensitive internal systems
- [ ] Adopt "deny-by-default" policies to block nonessential traffic and restrict access to trusted sources

Application-level prevention:

- ☐ Implement thorough data input sanitization, validation, and filtering to ensure the legitimacy of user-supplied URLs
- ☐ Disable HTTP redirection at the server level to prevent attackers from manipulating the destination of requests
- ☐ Ensure server responses conform to expected results and avoid exposing sensitive information. Raw server responses should never be directly sent to the client

MULTILAYERED APPROACH TO WEB APPS SECURITY

- Neutralizing risk factors and vulnerabilities
- Code protection: repositories and metadata backup
- Always-ready approach for data loss event: disaster recovery, ransomware protection, data migration, and more.

Read use cases


GitProtect.io
by Xopero ONE