

Question

What are the top 10 API security risks listed in the OWASCI
What is the first OWASP 2023 Top 10 API vulnerability?
What is the second OWASP 2023 Top 10 API vulnerability?
What is the third OWASP 2023 Top 10 API vulnerability?
What is the fourth OWASP 2023 Top 10 API vulnerability?
What is the fifth OWASP 2023 Top 10 API vulnerability?
What is the sixth OWASP 2023 Top 10 API vulnerability?
What is the seventh OWASP 2023 Top 10 API vulnerability?
What is the eighth OWASP 2023 Top 10 API vulnerability?
What is the ninth OWASP 2023 Top 10 API vulnerability?
What is the tenth OWASP 2023 Top 10 API vulnerability?
In the OWASP 2023 Top 10 list, what does the notation "/
What is OWASP and what does it stand for?
What types of resources does OWASP provide?
Are OWASP's resources accessible to everyone?
What is OWASP's approach to application security?
Is OWASP affiliated with any commercial technology com
What is the OWASP Foundation?
Who contributes to OWASP?
How does OWASP support security research?
What kind of events and community involvement does O'
What is the primary focus of the OWASP API Security Top
How does API security differ from broader web applicatio
Is there an example of a common API vulnerability?
Where can one contribute to the OWASP API Security Top
What should new readers of the OWASP Top 10 series do
What is the OWASP API Security Top 10 - 2023?
When was the first edition of the OWASP API Security Top
Why are APIs considered crucial in modern application ar
Can you provide a code example that shows a common A
What are some related OWASP top 10 projects recomme
What is the OWASP API Security Top 10 2023 and how is i
Why was there a combination of "Excessive Data Exposur
What is the new category "Unrestricted Access to Sensitiv
What does the "Unsafe Consumption of APIs" category ei
How was the OWASP API Security Top 10 2023 list compil
What methodology does OWASP use for risk analysis in A
How does the OWASP Risk Rating Methodology categoriz
What factors are considered in the OWASP Risk Rating M
Does the OWASP Risk Rating Methodology for API securit
Is the OWASP API Security Top 10 risk analysis data-drive
What is the purpose of the OWASP API Security Top 10?
What are the levels of Weakness Prevalence in the OWAS

How is Weakness Detectability classified in the OWASP Risk Index?

What does Technical Impact mean in the context of the OWASP Risk Rating?

How are Business Impacts evaluated in the OWASP Risk Rating?

What is the significance of categorizing Exploitability in the OWASP Risk Rating?

What is Broken Object Level Authorization (BOLA) in APIs?

How does Broken Object Level Authorization impact an application?

What are some example attack scenarios for BOLA?

How can Broken Object Level Authorization be prevented?

What are the threat agents and attack vectors for Broken Object Level Authorization?

What is the security weakness associated with BOLA?

What are the impacts of Broken Object Level Authorization?

Can you provide an attack scenario for BOLA involving an application?

How would BOLA manifest in an automobile manufacturer's API?

What is an example of BOLA in an online document storage system?

Discuss the exploitability, prevalence, and technical impact of BOLA.

What is API2:2023 Broken Authentication in the OWASP Top 10?

How easy is it to exploit the broken authentication vulnerability?

Why is broken authentication a common security weakness?

What is the technical and business impact of broken authentication?

Can you provide an example of a broken authentication vulnerability?

What are the signs that an API might be vulnerable to broken authentication?

How can broken authentication be prevented in APIs?

What is an example attack scenario involving broken authentication?

How can GraphQL batching be used as an attack vector in broken authentication?

What makes broken authentication a significant threat in APIs?

Can you explain the security weakness in broken authentication?

What are the long-term impacts of broken authentication?

What is API3:2023 Broken Object Property Level Authorization?

How can the exploitability of API3:2023 be described?

What is the prevalence and detectability of API3:2023?

What are the technical and business impacts of API3:2023?

Can you give an example of an attack scenario exploiting API3:2023?

What is a code example of an API request that could lead to API3:2023?

How can API3:2023 be prevented in API design?

What are some related references for understanding API3:2023?

What is API4:2023 Unrestricted Resource Consumption in APIs?

How can unrestricted resource consumption be exploited?

Can you provide a code example of an API vulnerable to unrestricted resource consumption?

What are some examples of attack scenarios for unrestricted resource consumption?

How can API4:2023 Unrestricted Resource Consumption be prevented?

What are some reference materials to understand API4:2023?

What are the threat agents and attack vectors associated with API4:2023?

What is the security weakness in APIs that leads to Unrestricted Resource Consumption?

What are the impacts of API4:2023 Unrestricted Resource Consumption?

How easy is it to exploit the Unrestricted Resource Consumption vulnerability?

How widespread is the issue of Unrestricted Resource Co
What is the technical severity of the Unrestricted Resourc
What is Broken Function Level Authorization in API securi
How can attackers exploit Broken Function Level Authoriz
Can you provide an example of an attack exploiting Broke
What are the best practices to prevent Broken Function L
What are some external resources for understanding and
Who are the typical threat agents in Broken Function Lev
What is the common security weakness in Broken Functic
What is the impact of Broken Function Level Authorizatio
How easy is it to exploit Broken Function Level Authorizat
How common are Broken Function Level Authorization vu
What is the technical severity of Broken Function Level Ai
What is API6:2023 Unrestricted Access to Sensitive Busin
How can attackers exploit Unrestricted Access to Sensitiv
What are some example attack scenarios for this vulnerab
How can businesses prevent Unrestricted Access to Sensi
Can you provide a code example of how to detect non-hu
Are there any references for further information on API se
What is the exploitability level of the API6:2023 vulnerabi
What does the prevalence metric indicate for API6:2023?
How would you describe the security weakness in API6:20
What is the detectability level for API6:2023?
Can you discuss the technical impact of API6:2023?
What are the business-specific impacts of API6:2023?
What is API7:2023 Server-Side Request Forgery (SSRF) in t
How exploitable is the SSRF vulnerability, and what are th
Can you provide an example of an SSRF vulnerability in ar
What's another scenario where SSRF can be exploited?
How can SSRF vulnerabilities be prevented?
Are there any additional references for understanding SSI
What does "Threat Agents/Attack Vectors" mean in the c
What are the security weaknesses associated with SSRF?
What impacts can SSRF have on a system or business?
How easy is it to exploit SSRF vulnerabilities?
How common are SSRF vulnerabilities?
What is the technical impact of SSRF vulnerabilities?
What is API8:2023 Security Misconfiguration in the OWAS
What are the common indicators that an API might be vu
Can you provide an example attack scenario demonstrati
How can API Security Misconfiguration be prevented?
What are the reference materials for understanding and p
What is the exploitability level of API8:2023 Security Misc
How prevalent is Security Misconfiguration in APIs as per
What is the detectability level of Security Misconfiguratio

What kind of security weakness does API8:2023 Security Issues represent?

What are the technical impacts of Security Misconfiguration?

How does Security Misconfiguration impact businesses?

What is API9:2023 in the OWASP Top 10 2023?

How can attackers exploit Improper Inventory Management?

What are some real-world attack scenarios for API9:2023?

How can organizations prevent the API9:2023 vulnerability?

Can you provide an example of how to document an API threat?

What are the threat agents associated with API9:2023 Impact?

What is the security weakness in API9:2023?

What impacts can result from exploiting API9:2023 vulnerabilities?

How easy is it to exploit the API9:2023 vulnerability?

How prevalent is the API9:2023 vulnerability?

What is the technical impact of the API9:2023 vulnerability?

What is API10:2023 in the OWASP Top 10 2023 list?

How can attackers exploit Unsafe Consumption of APIs vulnerability?

What are the common security weaknesses in Unsafe Consumption of APIs?

Can you provide an example of how an attacker might exploit Unsafe Consumption of APIs?

What is another attack scenario for Unsafe Consumption of APIs?

What is a preventive measure for Unsafe Consumption of APIs?

Can you list some relevant OWASP Cheat Sheets for preventing Unsafe Consumption of APIs?

What are the Threat Agents in the context of API10:2023?

What are the Attack Vectors for Unsafe Consumption of APIs?

How does the Security Weakness manifest in Unsafe Consumption of APIs?

What is the Impact of Unsafe Consumption of APIs on technical systems?

How would you rate the Exploitability of Unsafe Consumption of APIs?

What is the Prevalence of Unsafe Consumption of APIs in the wild?

What makes the Detectability of Unsafe Consumption of APIs challenging?

Can you discuss the Technical Severity of Unsafe Consumption of APIs?

Answer

The OWASP Top 10 API Security Risks for 2023 are as follows:

The first OWASP 2023 Top 10 API vulnerability is "Broken Object Level Authentication"

The second OWASP 2023 Top 10 API vulnerability is "Broken Access Control"

The third OWASP 2023 Top 10 API vulnerability is "Broken Authentication"

The fourth OWASP 2023 Top 10 API vulnerability is "Unrestricted Data Exposure"

The fifth OWASP 2023 Top 10 API vulnerability is "Broken Key Management"

The sixth OWASP 2023 Top 10 API vulnerability is "Unrestful APIs"

The seventh OWASP 2023 Top 10 API vulnerability is "Server-Side Request Forgery"

The eighth OWASP 2023 Top 10 API vulnerability is "Secure Data at Rest"

The ninth OWASP 2023 Top 10 API vulnerability is "Improper Protocol Usage"

The tenth OWASP 2023 Top 10 API vulnerability is "Unsafe Cryptographic Storage"

In the OWASP 2023 Top 10 list, the notation "API1:2023" indicates:

OWASP stands for the Open Web Application Security Project.

OWASP offers a variety of resources that are free and open source.

Yes, all OWASP tools, documents, videos, presentations, and research are open source.

OWASP advocates for a comprehensive approach to application security.

No, OWASP is not affiliated with any technology company.

The OWASP Foundation is the non-profit entity responsible for OWASP.

OWASP is primarily a volunteer-driven organization. This is why it is a community project.

OWASP supports security research by providing grants and funding for research projects.

OWASP hosts and participates in extensive conferences worldwide.

The primary focus of the OWASP API Security Top 10 is to **Foreword**.

API security differs from broader web application security.

A common API vulnerability is "Broken Object Level Authentication".

Contributions to the OWASP API Security Top 10 can be made through GitHub.

New readers of the OWASP Top 10 series are recommended to start with the **Introduction**.

The OWASP API Security Top 10 - 2023 is the second edition of the **Introduction**.

The first edition of the OWASP API Security Top 10 was published in 2019.

APIs are crucial in modern application architecture because they enable reuse and integration.

Certainly. One common API vulnerability is improper input validation.

Some related OWASP Top 10 projects include OWASP Cloud Security Top 10.

The OWASP API Security Top 10 2023 is an updated list of vulnerabilities.

In the OWASP API Security Top 10 2023, "Excessive Data Exposure" is a new category.

"Unrestricted Access to Sensitive Business Flows" is a new category.

The "Unsafe Consumption of APIs" category highlights the importance of secure API usage.

Despite a public call for data yielding no contributions, the OWASP community continues to work on improving API security.

OWASP utilizes the OWASP Risk Rating Methodology for a more detailed analysis of the risks.

The OWASP Risk Rating Methodology categorizes the exposure risk of each vulnerability.

The OWASP Risk Rating Methodology for API security considers various factors such as the impact and likelihood.

No, the OWASP Risk Rating Methodology for API security does not consider the severity of the vulnerability.

No, the OWASP API Security Top 10 risk analysis is not detailed enough to provide specific recommendations.

The purpose of the OWASP API Security Top 10 is to identify the most critical API security risks.

In the OWASP Risk Rating Methodology for API security, vulnerabilities are assigned a risk score based on their exposure risk.

Weakness Detectability in the OWASP Risk Rating Methodology

Technical Impact in the OWASP Risk Rating Methodology

Business Impacts in the OWASP Risk Rating Methodology

Categorizing Exploitability in the OWASP Risk Rating Methodology

Broken Object Level Authorization, featured as API1:2023 **API1:2023 Broken Object Level Authorization**

The impacts of BOLA can be significant, ranging from unauthorized access to data loss.

One scenario involves an e-commerce platform where an attacker can bypass authorization checks.

To prevent BOLA:1. Implement robust authorization mechanisms.

The threat agents for BOLA typically include attackers who exploit authorization flaws.

The primary security weakness in BOLA lies in the failure to properly validate object references.

The technical impact of BOLA is generally moderate but can lead to significant data loss.

Yes, in an e-commerce platform scenario, an attacker can identify products without proper authorization.

In this scenario, the automobile manufacturer's API allows unauthorized access to vehicle data.

In an online document storage service, users can perform unauthorized operations on files.

BOLA is considered to have high exploitability due to its straightforward nature.

API2:2023 Broken Authentication refers to security issues in API authentication mechanisms.

The exploitability of broken authentication in APIs is considered "Moderate."

Broken authentication is common due to misconceptions about best practices.

The technical impact is 'Severe', as attackers can gain control over user accounts.

Yes, consider a GraphQL API where login attempts are submitted without proper validation.

An API could be vulnerable if it allows:1. Credential stuffing attacks.

To prevent broken authentication:1. Understand all authentication flows.

In one scenario, attackers exploit the absence of a re-authentication mechanism.

Attackers can leverage GraphQL query batching to bypass rate limits.

The threat agents in broken authentication scenarios are typically malicious actors.

The security weakness often stems from misconceptions about secure password management.

The long-term impacts are severe for both the technical infrastructure and user privacy.

API3:2023 Broken Object Property Level Authorization is a vulnerability in how objects are handled.

The exploitability of API3:2023 is considered "Easy." Attackers can easily manipulate object properties.

API3:2023 is common in prevalence, indicating it's a frequent issue across various applications.

The technical impact is rated as "Moderate," which means it can lead to significant data loss.

Scenario #1: In a dating app, an API endpoint allows a user to report another user.

Json Code:```POST /graphql{ "operationName":"reportUser", "variables":{ "id": "123", "userId": "456" } }```

To prevent this vulnerability:1. Validate user access to specific resources.

Relevant references include:1. OWASP API Security Top 10 - Unrestricted Resource Consumption.

API4:2023 Unrestricted Resource Consumption is a vulnerability where an application consumes excessive resources.

Exploitation typically involves sending multiple or crafted requests to exhaust system resources.

Certainly. Consider an API endpoint that generates reports on demand.

One scenario involves a social network's "forgot password" feature being exploited.

Prevention strategies include:1. Using solutions like Content Delivery Networks (CDNs).

Reference materials include OWASP's own documentation on API security.

The threat agents in this context are individuals or systems that abuse resource consumption.

The primary security weakness is the lack of effective monitoring and mitigation.

Technically, this vulnerability can lead to Denial of Service (DoS) attacks.

The exploitability of this vulnerability is considered average.

The prevalence of this vulnerability is widespread. Many APIs are affected.

The technical severity of this vulnerability is considered severe.

API5:2023 Broken Function Level Authorization

Attackers can exploit these vulnerabilities by sending legitimate requests.

Certainly. Consider a registration process for an invite-only API.

To prevent these issues, your application should have a comprehensive access control system.

Useful external resources include the OWASP website, particularly the API Security section.

Threat agents in the context of Broken Function Level Authorization are typically automated tools.

The common security weakness in Broken Function Level Authorization is insufficient validation of user inputs.

The business impact of Broken Function Level Authorization is significant, leading to data breaches and loss of trust.

Broken Function Level Authorization vulnerabilities are generally easy to detect and exploit.

Broken Function Level Authorization vulnerabilities are quite common.

The technical severity of Broken Function Level Authorization is considered moderate.

API6:2023 Unrestricted Access to Sensitive Data

Attackers typically exploit this vulnerability by understanding the system's internal structures.

1. Scalping High-Demand Products: An attacker automates the process of buying products at a low price and reselling them at a higher price.

Prevention should be approached on two fronts: business logic and security measures.

Sure, here's a simple JavaScript example that might be useful for illustration.

Yes, for more information, you can refer to the OWASP API Security Guide.

The exploitability of the API6:2023 vulnerability is considered high.

The prevalence of the API6:2023 vulnerability is described as widespread.

The security weakness in API6:2023 is primarily due to a lack of proper validation.

The detectability of the API6:2023 vulnerability is considered moderate.

The technical impact of exploiting API6:2023 is generally revenue loss.

The business-specific impacts of API6:2023 can vary significantly.

API7:2023 Server Side Request Forgery (SSRF)

The exploitability of SSRF is considered easy, especially in web applications.

Sure, consider a social networking API that allows users to post links.

In a scenario involving webhooks, consider a security procedure.

To prevent SSRF vulnerabilities:
1. Isolate resource fetching logic from user input.

Yes, for further understanding of SSRF, you can refer to the OWASP API Security Guide.

In the context of SSRF (API7:2023), "Threat Agents/Attackers" refers to automated tools.

The primary security weakness in SSRF lies in the application's handling of URLs.

The impacts of SSRF can be significant, both technically and financially.

Exploiting SSRF vulnerabilities is generally considered easy.

SSRF vulnerabilities are quite common. This prevalence is due to the nature of the vulnerability.

The technical impact of SSRF vulnerabilities can range from minor to catastrophic.

API8:2023 Security Misconfiguration

An API might be vulnerable to Security Misconfiguration if it lacks proper configuration.

Scenario #1: An API server uses a logging utility with JNDI injection.

To prevent API Security Misconfiguration:
1. Implement a secure configuration management system.

Key references include:
1. OWASP Secure Headers Project.

The exploitability of API8:2023 Security Misconfiguration is considered high.

Security Misconfiguration is considered "Widespread" in the API Security Guide.

The detectability of Security Misconfiguration in APIs is also considered high.

API8:2023 Security Misconfiguration represents a security issue where APIs are not properly configured, leading to potential vulnerabilities.

The technical impact of Security Misconfiguration in APIs includes:

The impact of Security Misconfiguration on businesses can be significant.

API9:2023 refers to "Improper Inventory Management" or **API9:2023 Improper Inventory Management**.

Attackers can exploit this vulnerability by accessing old API documentation.

Two real-world scenarios include:

1. An attacker finds a bug in the API documentation.

To prevent this vulnerability, organizations should:

1. Implement strict access controls.

To mitigate the risk of this vulnerability, API documentation should be reviewed regularly.

The threat agents in API9:2023 mainly include external attackers and internal employees.

The core security weakness in API9:2023 stems from inadequate inventory management.

The exploitation of API9:2023 can lead to several impacts, such as data theft and system downtime.

The exploitability of API9:2023 is generally considered easy.

The prevalence of API9:2023 is widespread. Many organizations use legacy systems that are vulnerable.

The technical impact of API9:2023 can be moderate to severe, depending on the specific vulnerability.

API10:2023, in the OWASP Top 10 2023 list, refers to "Unsafe Consumption of APIs".

Attackers can exploit this vulnerability by identifying and abusing APIs.

The common security weaknesses in Unsafe Consumption of APIs include:

Sure, here's an example related to SQL injection:

Scenario: A user inputs a query string into an API endpoint.

Another scenario involves redirect vulnerabilities:

Scenario: A user is redirected to a malicious URL via an API.

A key preventive measure is to always validate and sanitize user input.

Yes, relevant OWASP Cheat Sheets include:

1. Web Service Security Cheat Sheet

The threat agents for API10:2023 primarily include attackers and internal employees.

Attack vectors for Unsafe Consumption of APIs include:

1. Direct API calls from web applications.

The primary security weakness in Unsafe Consumption of APIs is lack of proper validation and sanitization.

Technically, Unsafe Consumption of APIs can lead to severe data breaches and system downtime.

The exploitability of Unsafe Consumption of APIs is considered high.

The prevalence of Unsafe Consumption of APIs is common, especially in legacy systems.

Detectability is rated as average because while some instances are obvious, others may be more subtle.

The technical severity of Unsafe Consumption of APIs is considered high.

ion

uthorization

ption

ation

Business Flows

nt