

Vulnalyzer

DashboardScanReportsTeamRemediationSettings

Alex Morgan

Welcome to Vulnalyzer

Your comprehensive security vulnerability scanning platform. Start a scan to discover and remediate security issues.

Vulnerability Metrics

Total Vulnerabilities

61

↓ 12%

High

8

Critical

3

↓ 2

Medium

15

↑ 3

Critical

High

Medium

Low

Info

Recent Scans

View all

E-commerce Platform Scan

https://example-ecommerce.com

Completed Apr 15, 2025, 04:45 PM

30 Issues

DYNAMIC

Internal API Security Check

https://api.internal-example.com

Started Apr 18, 2025, 03:15 PM

19 Issues

68%

STATIC

Customer Portal Scan

https://portal.example.com

Started Apr 20, 2025, 07:30 PM

0 Issues

HYBRID

Marketing Website Security

https://www.example-marketing.com

Completed Apr 14, 2025, 10:05 PM

11 Issues

DYNAMIC

OWASP Top 10 Coverage

78% Avg. Coverage

Broken Access C.

Cryptographic F...

Injection

Insecure Design

Security Miscon.

Vulnerable Comp.

Auth Failures

Software Integr.

Logging Failure...

SSRF

Quick Scan

URL to scan

https://example.com

Scan Type

StaticDynamicHybrid

Start Scan

Advanced scan options

Team Activity

Alex Morgan

Started a new security scan E-commerce Platform

Alex Morgan

Assigned to Jordan Lee SQL Injection in Login...

Jordan Lee

Added a comment SQL Injection in Login...

Taylor Kim

Changed status to In Progress Cross-Site Scripting (...)

Jordan Lee

Started a new security scan Internal API

Vulnalyzer

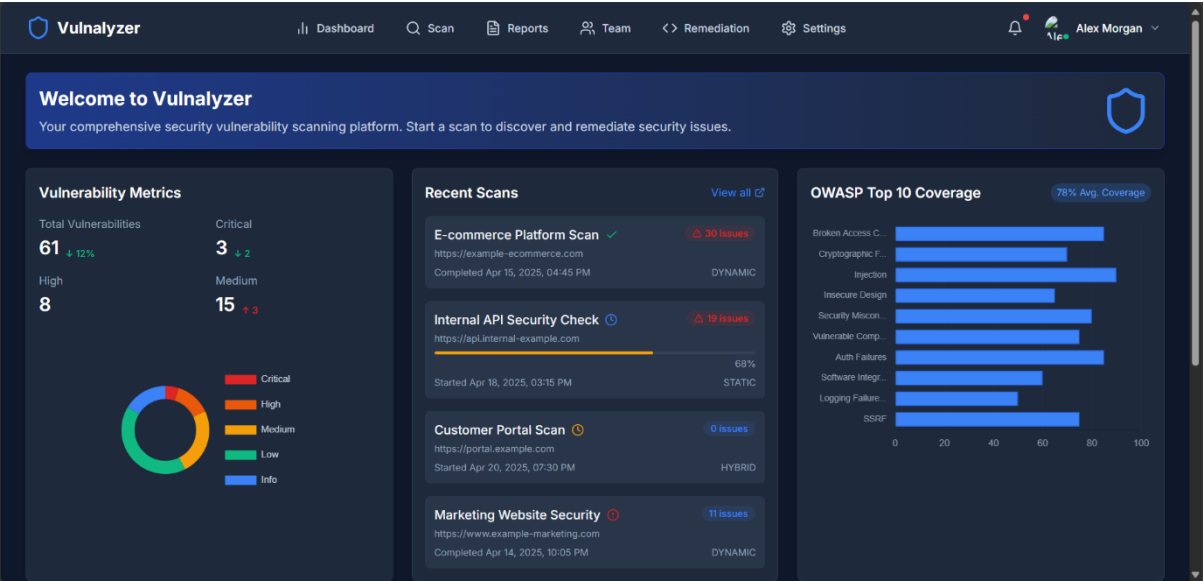
DashboardScanReportsTeamRemediationSettings

Alex Morgan

Search vulnerabilities...

5 vulnerabilities found

VULNERABILITY	SEVERITY	LOCATION	STATUS	ASSIGNED TO	ACTIONS
<div><div>SQL Injection in Login Form</div><div>Apr 15, 2025, 04:15 PM</div></div>	Critical	/login.php:42	Open	Jordan Lee	<div>Remediate Now</div> <div>Mark as False Positive</div>
<div><div>Description</div><div>The login form is vulnerable to SQL injection attacks, allowing attackers to bypass authentication.</div><div><div>Remediation</div><div>Use prepared statements or parameterized queries instead of string concatenation.</div></div><div><div>OWASP Reference</div><div>CWE</div><div>CVSS Score</div></div><div><div>A1:2021-Injection</div><div>CWE-89</div><div>9.8</div></div><div><div>Code Snippet</div><div>// login.php:42 function processUserInput(input) { const query = `SELECT * FROM users WHERE username = ` + input + `"; // Vulnerable to SQL injection return db.execute(query); }</div></div></div>					
<div><div>Broken Authentication in Password Reset</div><div>Apr 18, 2025, 03:45 PM</div></div>	Critical	/auth/reset-password.js:112	Open	Jamie Smith	



Vulnalyzer

Dashboard

Scan

Reports

Team

Remediation

Settings

Alex Morgan

Scan in Progress

Real-time monitoring of your security scan

Web Application Security Scan

Scan ID: new-scan

Pause

Stop

Progress

45%

Vulnerabilities Found

Critical: 0High: 0Medium: 1Low: 0Info: 2

Live Logs

[2025-05-31T04:08:19.426Z] Warning: weak password policy

[2025-05-31T04:08:20.974Z] Checking for unvalidated redirects and forwards...

[2025-05-31T04:08:24.547Z] Scanning for components with known vulnerabilities...

[2025-05-31T04:08:25.060Z] Checking for sensitive data exposure...

[2025-05-31T04:08:26.089Z] Scanning for insecure deserialization...

[2025-05-31T04:08:27.623Z] Low: Cookie without secure flag

[2025-05-31T04:08:30.172Z] Testing for insecure communications...

[2025-05-31T04:08:30.682Z] Testing API endpoints for security issues...

[2025-05-31T04:08:31.200Z] Scanning for security misconfigurations...

[2025-05-31T04:08:32.221Z] Low: Cookie without secure flag

[2025-05-31T04:08:34.778Z] No vulnerabilities found

[2025-05-31T04:08:36.313Z] Medium: Missing security headers

Vulnalyzer

Dashboard

Scan

Reports

Team

Remediation

Settings

Alex Morgan

Scan Results

Detailed findings from your security scan

Re-scan

Full Report

Export

Search vulnerabilities...

5 vulnerabilities found

Vulnerability	Severity	Location	Status	Assigned To	Actions
<div><div></div><div>SQL Injection in Login Form</div><div>Apr 15, 2025, 04:15 PM</div></div>	Critical	/login.php:42	Open	Jordan Lee	<div>Remediate Now</div> <div>Mark as False Positive</div> <div>Assign To</div> <div>Select team member</div> <div>Comments</div> <div>No comments yet</div> <div>Add a comment...</div> <div>Post</div>

Description

The login form is vulnerable to SQL Injection attacks, allowing attackers to bypass authentication.

Remediation

Use prepared statements or parameterized queries instead of string concatenation.

OWASP ReferenceCWECVSS Score

A1:2021-InjectionCWE-899.8

Code Snippet

// login.php:42
function processUserInput(input) {
 const query = "SELECT * FROM users WHERE username = '" + input + "'";
 // Vulnerable to SQL Injection
 return db.execute(query);
}

Vulnalyzer

Dashboard

Scan

Reports

Team

Remediation

Settings

Alex Morgan

Scan Results

Detailed findings from your security scan

Re-scan

Full Report

Export

Search vulnerabilities...

5 vulnerabilities found

Vulnerability	Severity	Location	Status	Assigned To	Actions
<div><div></div><div>SQL Injection in Login Form</div><div>Apr 15, 2025, 04:15 PM</div></div>	Critical	/login.php:42	Open	Jordan Lee	<div></div> <div></div>
<div><div></div><div>Broken Authentication in Password Reset</div><div>Apr 18, 2025, 03:45 PM</div></div>	Critical	/auth/reset-password.js:112	Open	Jamie Smith	<div></div> <div></div>
<div><div></div><div>Cross-Site Scripting (XSS) in Comment Section</div><div>Apr 15, 2025, 04:20 PM</div></div>	High	/products/view.php:157	In Progress	Taylor Kim	<div></div> <div></div>
<div><div></div><div>Insecure Direct Object Reference (IDOR)</div><div>Apr 15, 2025, 04:35 PM</div></div>	High	/account/details.php:88	Open	Alex Morgan	<div></div> <div></div>
<div><div></div><div>Missing Content Security Policy</div><div>Apr 15, 2025, 04:40 PM</div></div>	Medium	HTTP Headers	Open	Assign	<div></div> <div></div>

Vulnalyzer

Dashboard

Scan

Reports

Team

Remediation

Settings

Alex Morgan

Team Collaboration

Work together to identify and fix security vulnerabilities

Invite Team Member

Team Collaboration

3 online

Team Members

Alex Morgan

admin

Assign

Jordan Lee

member

Assign

Taylor Kim

member

Assign

Jamie Smith

member

Assign

Discussion

Hide

Jordan Lee

Apr 16, 2025, 03:00 PM

I've started working on this. Will implement prepared statements.

Alex Morgan

Apr 16, 2025, 03:15 PM

Great! Make sure to also add input validation as an extra layer of security.

Taylor Kim

Apr 16, 2025, 07:50 PM

This might be more complex than initially thought. The XSS vulnerability exists in multiple places.

Jamie Smith

Apr 18, 2025, 04:35 PM

I'll implement a secure token generation system with proper expiration.

Add your comments or notes...

Post Comment

Vulnalyzer

Dashboard

Scan

Reports

Team

Remediation

Settings

Alex Morgan

Code Remediation Sandbox

Fix security vulnerabilities in real-time with AI-powered suggestions

Light Mode

Settings

Share

History

Attack Path

AI Explorer

Test Payloads

Editor

Analysis

Collaborators (2)

Chat

```
1 // Example code with vulnerabilities
2 function displayUserData(userID) {
3   const element = document.getElementById('output');
4   element.innerHTML = userInput; // XSS vulnerability
5 }
6
7
8 function searchDatabase(query) {
9   const sql = "SELECT * FROM users WHERE name = '" + query + "'"; // SQL injection
10  return db.execute(sql);
11 }
12
13 function processCommand(input) {
14   return eval(input); // Command injection
15 }
16
```

AI Security Assistant

Welcome to the AI Security Assistant! I can help you with:

1. Security analysis of your code

2. Code enhancement suggestions

3. Best practices implementation

4. Vulnerability fixes

What would you like me to help you with?

9:13:46 am

Ask about security, optimization, or co

Vulnalyzer

Dashboard

Scan

Reports

Team

Remediation

Settings

Alex Morgan

Vulnerabilities

✓ No vulnerabilities found

SQL Injection

Target System

XSS Vulnerability

Entry Point

AI Security Assistant

Generate Suggestions

Team Members

Invite

JD

John Doe

Security Lead

JS

Jane Smith

Developer

MJ

Mike Johnson

QA Engineer

Security Patterns

XSS Prevention

Prevents XSS by usingtextContent instead of innerHTML

SQL Injection Prevention

Prevents SQL injection using parameterized queries

