Websocket Endpoint Analysis Report

Insecure WebSocket Implementations: Crawling Public Sites, Testing Endpoints for Vulnerabilities, and Reporting Impact Analysis

Report Generated on : July 10, 2025

PESU Center for
Information Security,
Forensics and Cyber Resilience

PESU RESEARCH
11TH FLOOR, BE BLOCK,
PES UNIVERSITY, RING ROAD CAMPUS,
BANGALORE, INDIA

WebSocket Security Scan Report

Executive Summary

Real-time apps increasingly rely on WebSocket connections, but insecure implementations—such as missing origin checks or weak authentication—can allow hijacking or sensitive data exposure.

To address this, we developed an automated scanner that crawls public web applications, detects vulnerable WebSocket endpoints, and analyzes their real-world impact.

- Crawl and detect active WebSocket endpoints from public websites.
- Apply origin-header enforcement and protocol fuzzing tests to assess security gaps.
- Generate structured PDF reports summarizing detected vulnerabilities and severity.

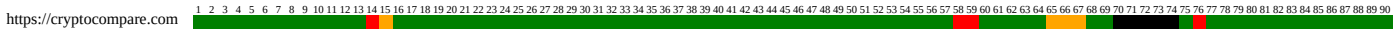
Scan Start Time:	2025-07-10 18:15:38
Scan End Time:	2025-07-10 18:19:11
Total Scan Duration:	214.9 seconds
Total URLs Scanned:	1
High Severity Vulnerabilities:	4
Medium Severity Vulnerabilities:	4
Low Severity Vulnerabilities:	0

All Scanned Websites

This section lists all scanned websites and summarizes the overall vulnerability distribution by severity. The bar graph below visualizes the number of High, Medium, and Low severity vulnerabilities identified across all scanned sites.

#	Website
1	https://cryptocompare.com

WebSocket vs. Attack Heatmap



Vulnerability Summary by Type

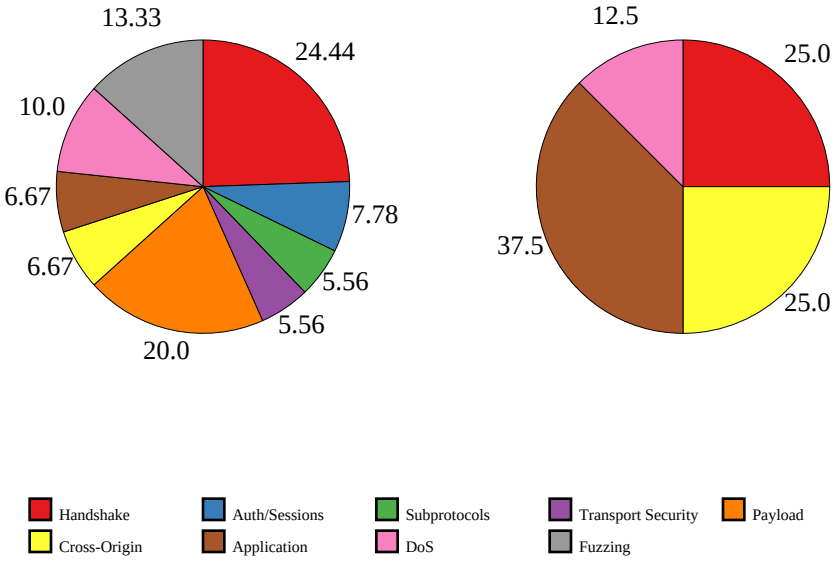
This section summarizes key categories of vulnerabilities found during the scan. It groups issues like missing origin checks, weak authentication, insecure handshakes, and over 80 other attack for test to highlight common WebSocket flaws.

The bar chart below visualizes how many vulnerabilities were found in each category. This helps quickly identify the most common and critical problem areas across scanned applications.

Type	Count
Handshake & Upgrade Validation	2
Authentication & Session Control	0
Subprotocols & Extension Handling	0
Transport Security & Encryption	0
Payload Framing & Messaging Semantics	0
Origin Policy & Cross-Origin Enforcement	2

Application-Layer Logic & Misconfigurations	3
DoS, Compression & Resource Limits	1
Protocol Fuzzing	0

Test Distribution vs Results



Detailed Scan Results

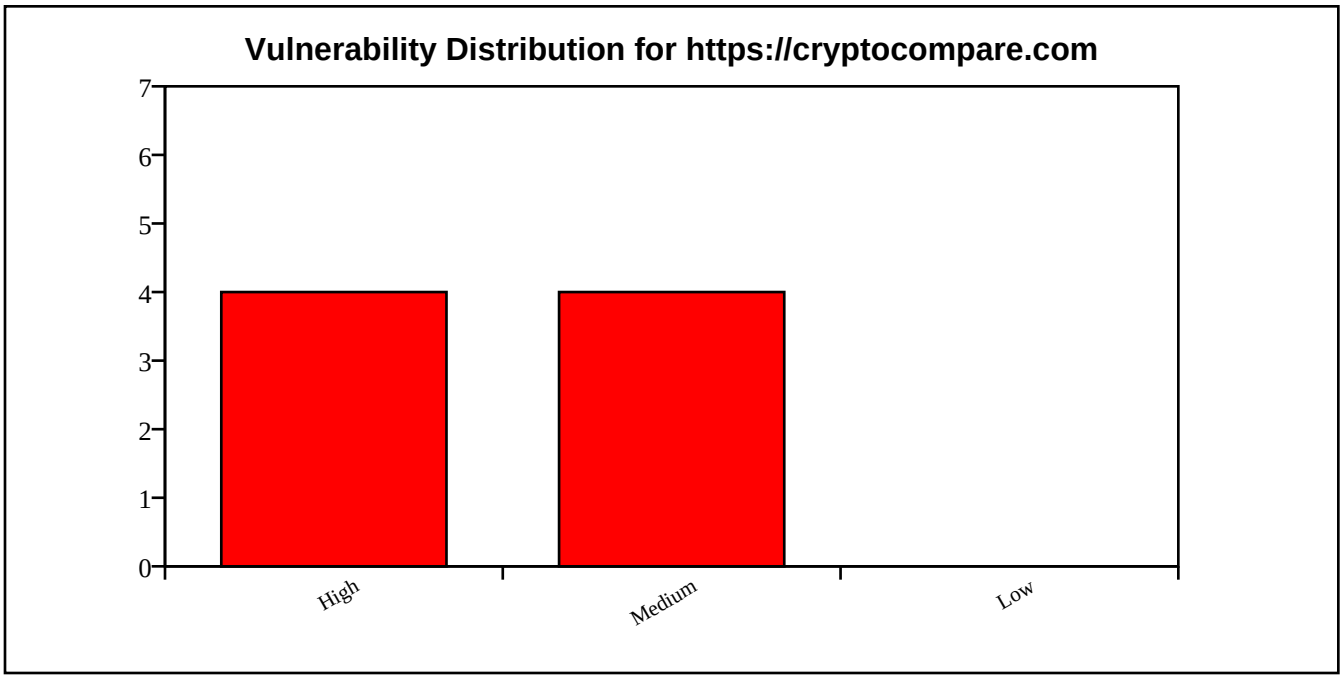
This section provides an in-depth breakdown of each scanned target. For every URL, it lists the scan duration, number of URLs crawled during reconnaissance, and the WebSocket endpoints discovered. It helps identify how many potential communication channels were exposed for testing. Each target's vulnerability distribution is summarized by severity (High, Medium, Low) using a bar chart, followed by a detailed list of detected vulnerabilities. The section also documents the types of attacks performed and the exact WebSocket endpoints and internal URLs involved in the scan. This allows for a thorough understanding of the security posture and exposure of each target.

Target URL: https://cryptocompare.com

Scan Duration:	198.23 seconds
URLs Crawled:	150
WebSocket Endpoints Found:	2
Attack Performed:	True
High Severity Findings:	4
Medium Severity Findings:	4
Low Severity Findings:	0

WebSocket Endpoints:

#	URL
1	wss://fews.stake.com.co/socket.io/?EIO=4 &transport=websocket
2	wss://streamer.cryptocompare.com/v2?form at=streamer



Detected Vulnerabilities:

This section lists all vulnerabilities identified during the scan of the target. Each entry includes the vulnerability name, its severity (High, Medium, or Low), a description of the issue, recommended solutions, and the affected WebSocket URL or host. This detailed information helps prioritize fixes and understand the exact flaws present in the WebSocket implementation of each target.

Affected WebSocket Endpoint: wss://fews.stake.com.co/socket.io

Name:	Fake HTTP Status
Risk Level:	High
Description:	Server at fews.stake.com.co:443 returned unexpected status: HTTP/1.1 502 Bad Gateway
Solution:	Ensure server returns "HTTP/1.1 101 Switching Protocols" for valid handshakes.

Name:	Wrong Sec-WebSocket-Accept
Risk Level:	Medium
Description:	Server at fews.stake.com.co:443 did not return a Sec-WebSocket-Accept header.
Solution:	Ensure server follows RFC 6455 and sends correct Sec-WebSocket-Accept header.

Name:	Missing CORS Headers
Risk Level:	High
Description:	WebSocket endpoint wss://fews.stake.com.co/socket.io (HTTP equivalent) lacks proper CORS headers.
Solution:	Implement proper CORS headers to restrict cross-origin access.

Name:	Cross-Origin Iframe
Risk Level:	High
Description:	wss://fews.stake.com.co/socket.io allows itself to be embedded in cross-origin iframes (missing X-Frame-Options / CSP).
Solution:	Set X-Frame-Options: DENY or SAMEORIGIN, or CSP frame-ancestors directive.

Name:	Server Disclosure
Risk Level:	Medium
Description:	WebSocket HTTP interface discloses: X-Powered-By: Express.
Solution:	Disable or obscure headers like Server, X-Powered-By, and X-AspNet-Version.

Name:	Invalid Content-Type
Risk Level:	Medium
Description:	WebSocket endpoint wss://fews.stake.com.co/socket.io (HTTP equivalent) serves invalid Content-Type: text/html; charset=utf-8.
Solution:	Ensure WebSocket endpoints return appropriate Content-Type or upgrade headers.

Name:	Missing Security Headers
Risk Level:	Medium
Description:	WebSocket endpoint wss://fews.stake.com.co/socket.io (HTTP equivalent) lacks the following headers: Strict-Transport-Security, X-Frame-Options.
Solution:	Add missing security headers such as Content-Security-Policy, X-Frame-Options, and Strict-Transport-Security.

Name:	TCP Half-Open Resource Leak
Risk Level:	High
Description:	WebSocket at wss://fews.stake.com.co/socket.io accepted hanging TCP connections without timeout.

Solution:	Use TCP keep-alive and server-side timeout policies.