WebSocket Insecurities Paper

Abstract

A WebSocket is a communication protocol that enables persistent, full-duplex communication between a client and a server over a single, long-lived TCP connection. WebSocket communication is a significantly faster means of communication compared to HTTP, enabling quick real-time communication. But this increase in speed of communication comes at a cost of minimal security. These vulnerabilities can lead to sensitive data exposure, unauthorized access, and many more problems. In this paper, we discuss our project, which covers extracting all WebSocket endpoints of a given website by means of crawling, and running various tests on the WebSocket, including tests on origin and authentication checks, protocol fuzzing using different payloads, handshaking, fragmentation, session management, subprotocols, encryption, Denial of Service (DOS), resource management, and cross-origin attacks. By running these tests, we obtain a complete and comprehensive report of all vulnerabilities.

Introduction

What is websocket

Websocket vs HTTP

Websocket vulnerabilities

Crawling

What is crawler

How does crawler work

Attacks

1. HTTP and Handshake Tests:

A WebSocket connection is obtained by upgrading the current http/ https connection. The following is the request message to the server

req = (

    f"GET {path} HTTP/1.1\r\n"

    f"Host: {host}\r\n"

    "Upgrade: websocket\r\n"

    "Connection: Upgrade\r\n"

    f"Sec-WebSocket-Key: {key}\r\n"

    "Sec-WebSocket-Version: 13\r\n"

    "\r\n"

)

And the response message from the server looks like

HTTP/1.1 101 Switching Protocols

Date: <date and time>

Connection: upgrade

Sec-WebSocket-Accept: <key>

Upgrade: websocket

…

The attacks under this category talk about modifying or omitting some lines from the request message to check if the server will still respond with the acceptance message and allow for switching of protocols.

**(Ask ma’am if we should elaborate on each test case here)**

2. Payload Handling and Fragmentation Tests:

Once the connection is established, the next batch of tests are executed. In these tests, we cover vulnerabilities during data transmission, mainly looking at opcodes, control frames, fragmentation, and others. We try to send illegal frames of different forms and see if the server returns a valid output.

3. Authentication and Session Management Tests:

4. Subprotocol and Extension Handling Tests

5. Transport Security and Encryption Tests

6. DoS and Resource Management Tests

7. Cross-Origin and Mixed Content Tests

8. Application-Layer Tests

Analysis and Results

Run the test for the 10 popular websites that make use of websockets  
Heatmap of 10 websites and all vulnerability categories  
2 pie charts: % of vulnerabilities, % of websites(which website ends up getting the most)

Use Case

There are certain tools that exist for the analysis of websockets like OWASP ZAP, Burp Suite.  
Explain what they do(They tend to go into certain specific domains. Also prior knowledge is needed to use the app properly. Our tool is very beginner friendly, as it requires user to only enter a website name and it tries to search for websocket endpoint on its own. Only if unable to find websockets it asks )

Conclusion

References

RFC 6455

<https://datatracker.ietf.org/doc/html/rfc6455>

Types of vulnerabilities:

1️⃣ Handshake & HTTP Request Tests

<https://core.ac.uk/download/pdf/45601062.pdf>

2️⃣ Payload Handling & Fragmentation

<https://www.openmymind.net/WebSocket-Framing-Masking-Fragmentation-and-More/?utm_source=chatgpt.com>

3️⃣ Authentication & Session Management

<https://arxiv.org/pdf/2104.05324>

4️⃣ Subprotocol & Extension Handling

[https://www.cyberchief.ai/2025/05/securing-websockets.html](https://www.cyberchief.ai/2025/05/securing-websockets.html?utm_source=chatgpt.com)

5️⃣ Security & Encryption

<https://www.ijnrd.org/papers/IJNRD2407274.pdf>

6️⃣ Denial of Service (DoS) & Resource Management

<https://cqr.company/web-vulnerabilities/denial-of-service-dos-via-websockets/>

8️⃣ Other Server Vulnerabilities

<https://arxiv.org/abs/1409.3367>

**Papers used at the start**

<https://www.researchgate.net/publication/384105299_Review_Analysis_of_Web_Socket_Security_Case_Study>

<https://juerkkil.iki.fi/files/websocket2012.pdf>

<https://oulurepo.oulu.fi/bitstream/handle/10024/6116/nbnfioulu-201603081281.pdf?sequence=1&utm_source=chatgpt.com>

<https://www.ijsr.net/getabstract.php?paperid=SUB152856>

<https://www.praetorian.com/blog/meshcentral-cross-site-websocket-hijacking-vulnerability/>

<https://ieeexplore.ieee.org/document/10479302>