23.12.2024

Nikhil Kumar

Learning Objectives

- Learn about WebSockets and their vulnerabilities.
- Learn how WebSocket Message Manipulation can be done.

Introduction to WebSocket

- ❖ WebSockets enable continuous communication between the browser and server.
- ❖ Unlike traditional methods, WebSockets keep the connection open for ongoing exchanges.
- ❖ Ideal for live chat, real-time games, or constant data updates.
- ❖ Uses a quick handshake to establish a connection for seamless two-way communication.
- ❖ Reduces overhead and ensures faster, real-time data flow.

Traditional HTTP Requests vs. WebSocket

HTTP Communication:

- Browser sends a request to the server.
- Server responds and closes the connection.
- New data requires a new request.
- Example: Like knocking on a door repeatedly for updates.

Polling with HTTP:

- Browser frequently asks, "Any new messages?"
- Inefficient and causes unnecessary work for both browser and server.

WebSocket Communication:

- Connection stays open after establishment.
- Server pushes updates directly when available.
- Example: Like leaving the door open for instant updates.
- Faster and resource-efficient.

WebSocket Vulnerabilities

While WebSockets can boost performance, they also come with security risks that developers need to monitor. Since WebSocket connections stay open and active, they can be taken advantage of if the proper security measures aren't in place. Here are some common vulnerabilities:

Weak Authentication and Authorization:

- WebSockets lack built-in authentication and session validation.
- Attackers can exploit improper controls to access sensitive data or disrupt connections.

Message Tampering:

- Data can be intercepted and altered if encryption isn't implemented.
- Attackers could inject harmful commands or manipulate the data.

Cross-Site WebSocket Hijacking (CSWSH):

- Attackers trick browsers into opening WebSocket connections to malicious sites.
- They may hijack connections or access sensitive data.

Denial of Service (DoS):

- Persistent WebSocket connections are vulnerable to DoS attacks.
- Attackers can flood servers with excessive messages, causing slowdowns or crashes.

What Is WebSocket Message Manipulation?

Definition:

- An attack where a hacker intercepts and modifies WebSocket messages exchanged between a web app and its server.

Cause:

- WebSockets keep an open, constant two-way communication, which lacks security if not properly configured.

Attack Scenario:

- Example: A hacker intercepts a message handling financial transactions, alters the amount, or reroutes funds to another account.

Consequences:

- **Bypassing Security**: Attackers can bypass checks, send unauthorized requests, and alter sensitive data (e.g., usernames or payment amounts).
- **Privilege Escalation**: Attackers may gain admin access or modify permissions by manipulating messages.
- **Data Corruption**: Invalid or tampered data can disrupt systems, causing user issues.
- **System Downtime**: Flooding servers with malicious messages can cause denial-of-service (DoS) attacks.

Risks:

- WebSocket connections often lack the robust security features (e.g., encryption) of HTTP connections.

Required Security Measures:

- Implement **message validation** and **encryption** to prevent tampering.
- Use **authentication** and **authorization checks** for all WebSocket interactions.

Impact:

- Leads to unauthorized actions, data manipulation, and potential system crashes if left unchecked.

The impact of changing WebSocket messages depends on how the app uses them and what kind of data is being sent. Here's a breakdown of what can happen:

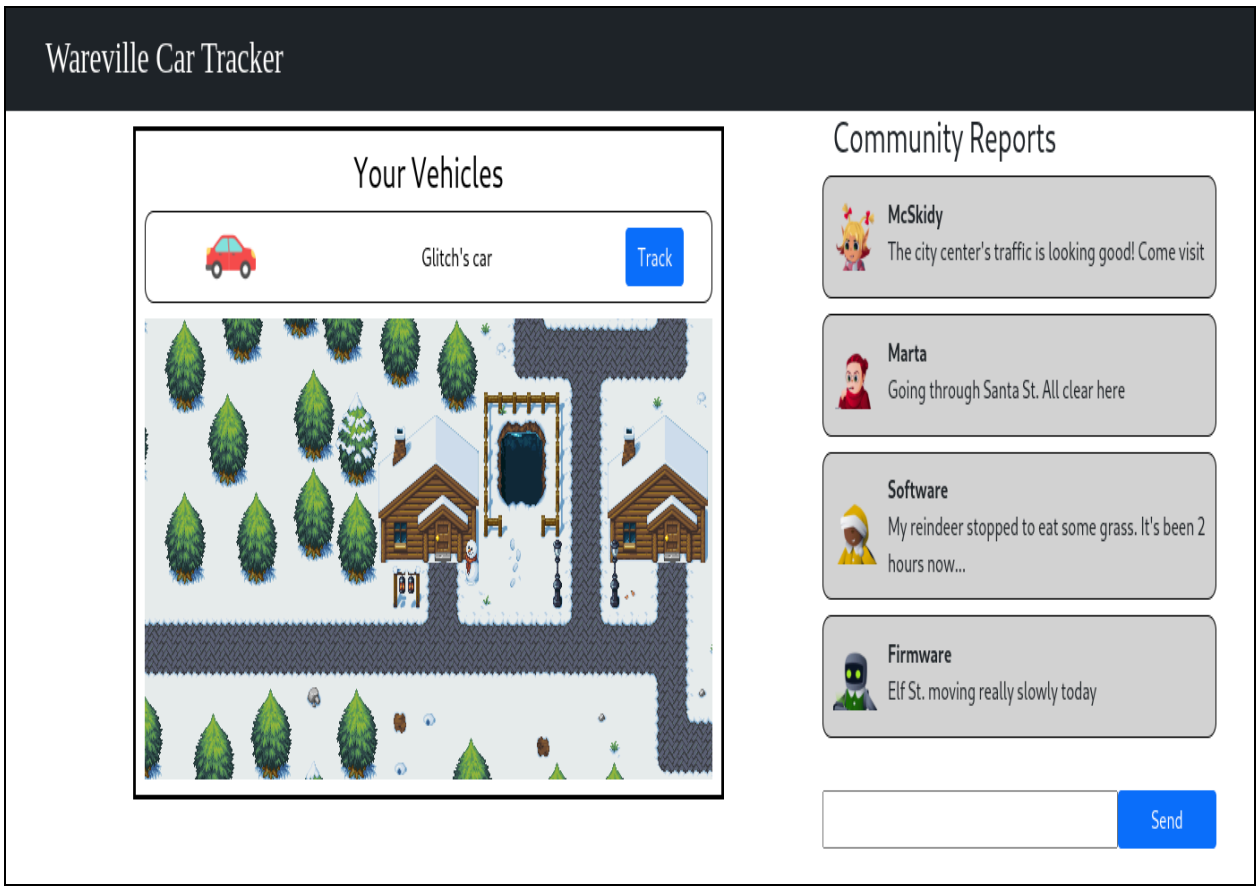
- **Doing Things Without Permission:** If someone can tamper with WebSocket messages, they could impersonate another user and carry out unauthorised actions such as making purchases, transferring funds, or changing account settings. For example, if a WebSocket manages payment transactions, an attacker could manipulate the transaction amount or reroute the payment to their own account.
- **Gaining Extra Privileges:** Attackers could also manipulate messages to make the system think they have more privileges than they actually do. This could let them access admin controls, change user data, view sensitive info, or mess with system settings.
- **Messing Up Data:** One of the significant risks is data corruption. If someone is changing the messages, they could feed bad data into the system. This could mess with user accounts, transactions, or anything else the app handles. They could change things in real-time and disrupt everyone's work in circumstances such as a shared document or tool.

- **Crashing the System:** An attacker could also spam the server with bad requests, causing it to slow down or crash. If this happens enough, the system could go offline, causing serious downtime for users and businesses.

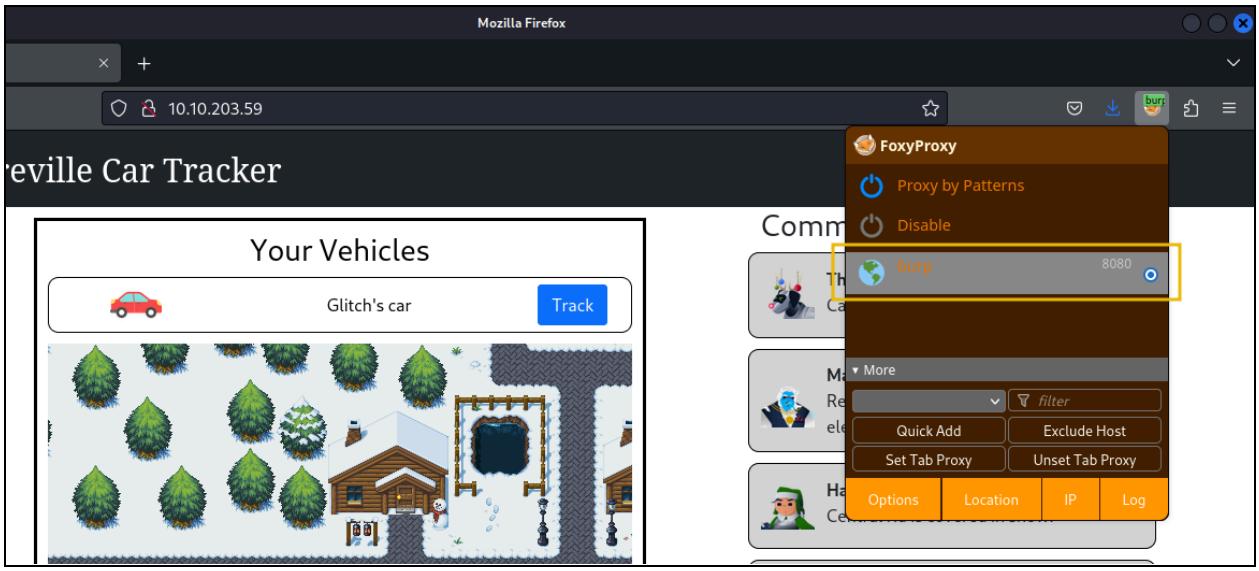
Without good security checks, this kind of message tampering can lead to anything from unauthorized actions to the downing of an entire service.

Exploitation

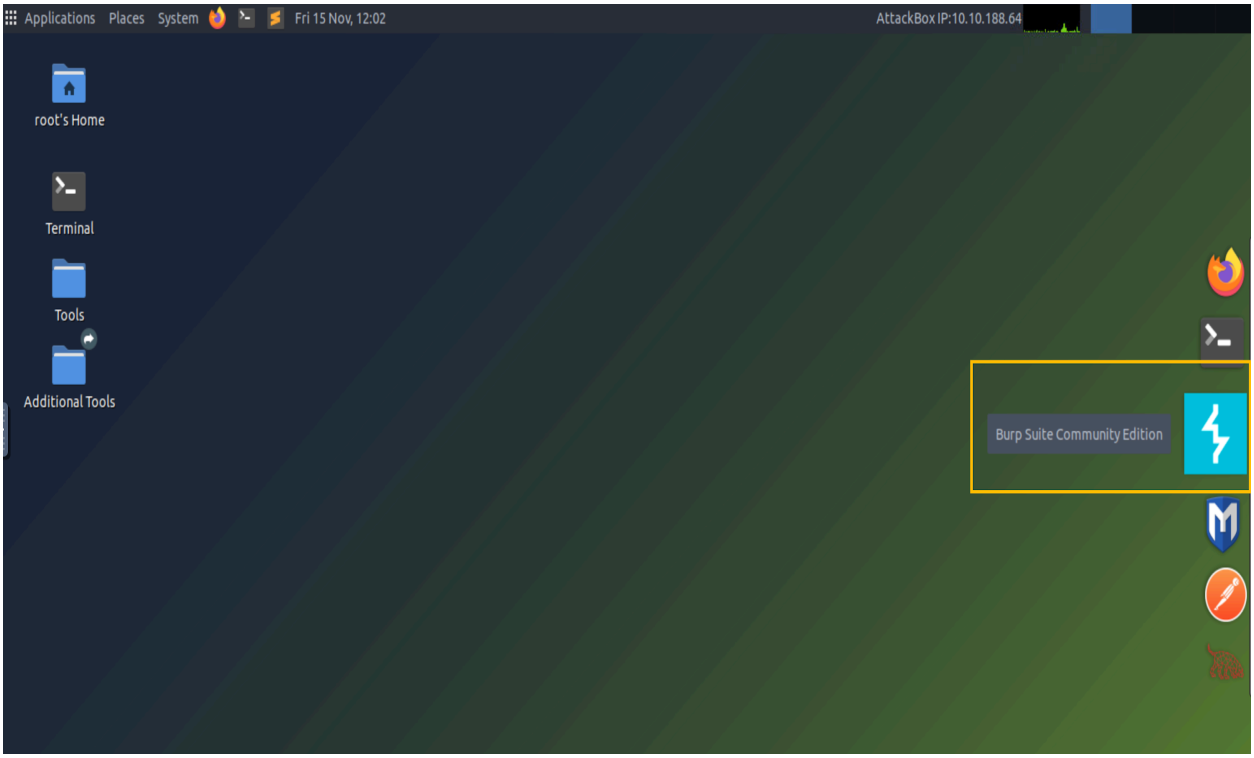
Navigate to http://10.10.226.241.

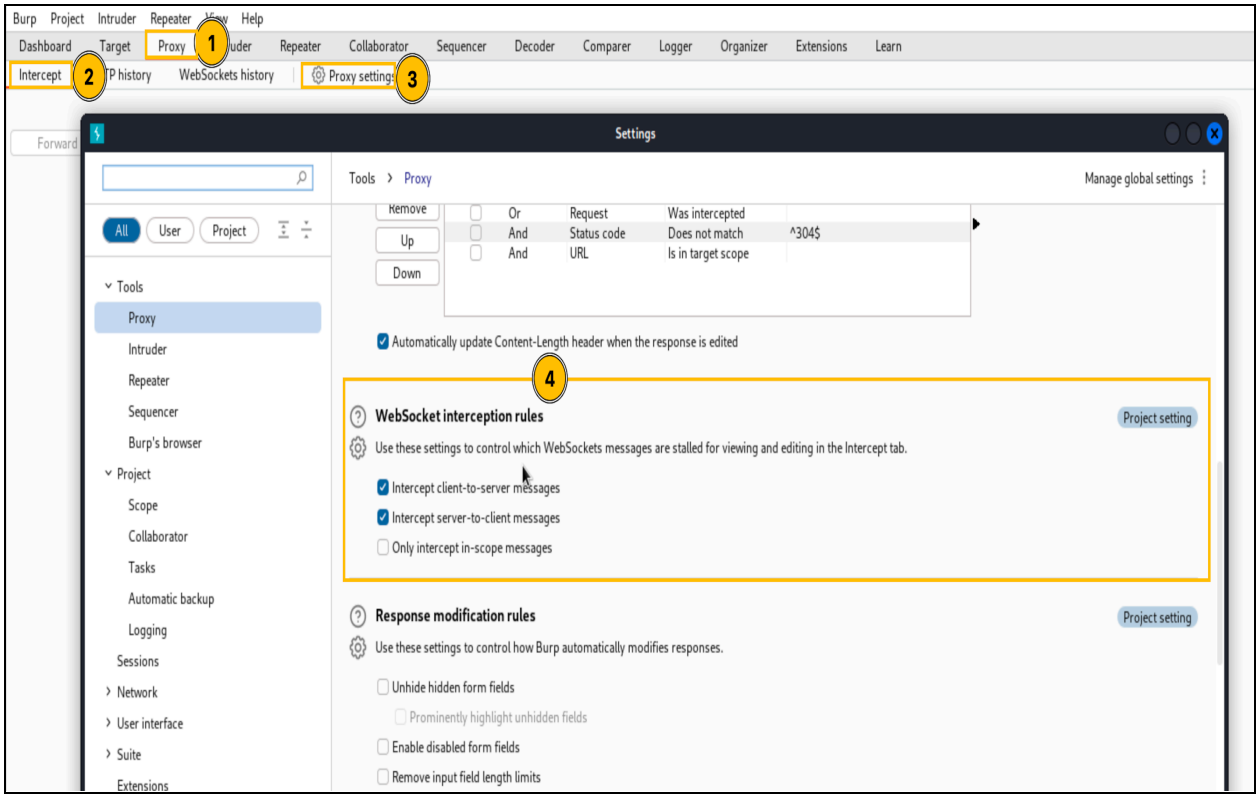


- If you're using the AttackBox, on your browser, make sure to proxy the traffic of the application, as shown below.

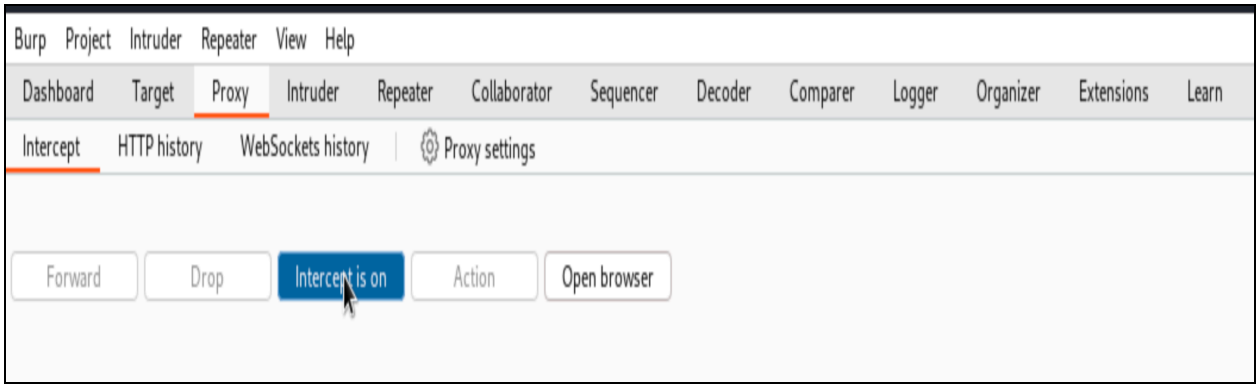


- Open Burp Suite, navigate to Proxy > Intercept > Proxy Settings and ensure the settings below are turned on.

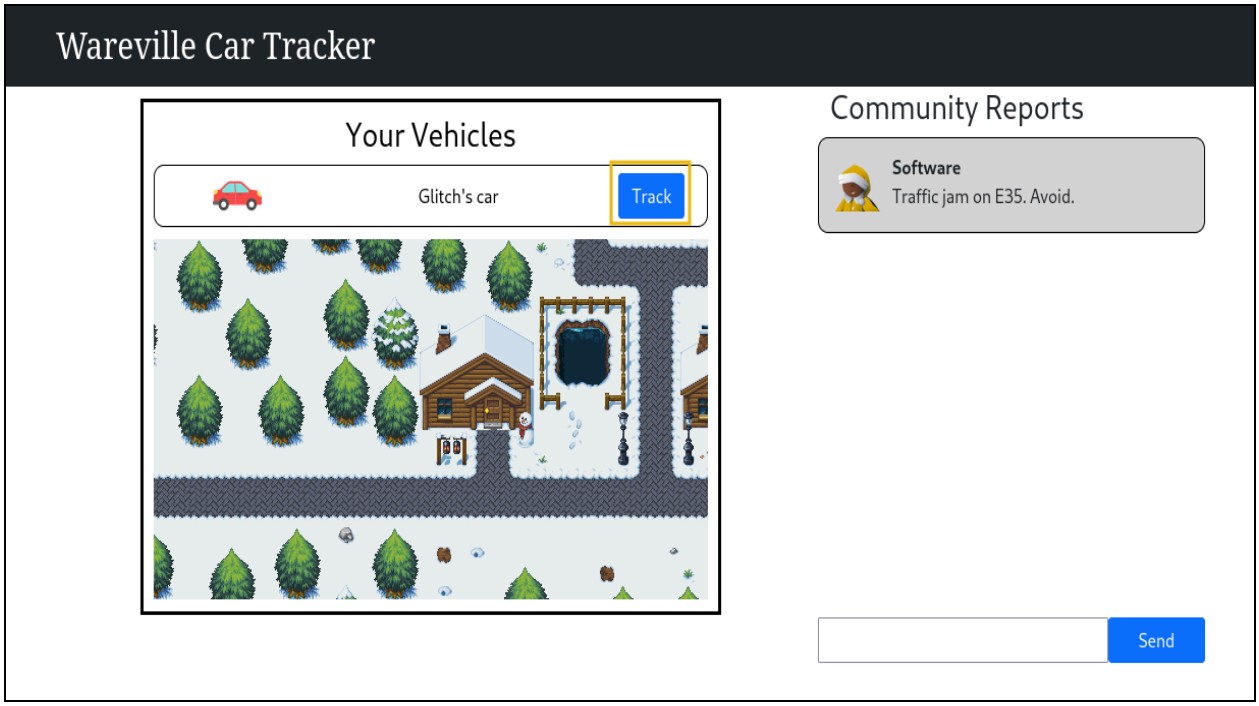




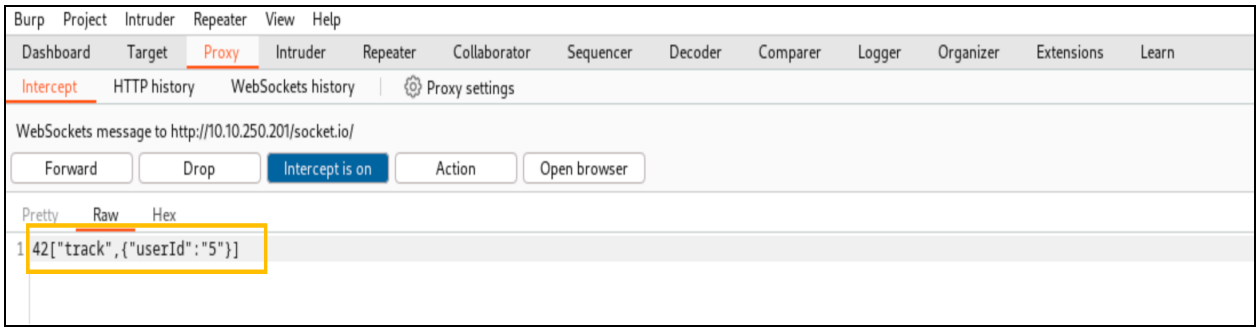
- Once done, close the window and enable the proxy intercept.



- Go back to your browser and click the Track button.

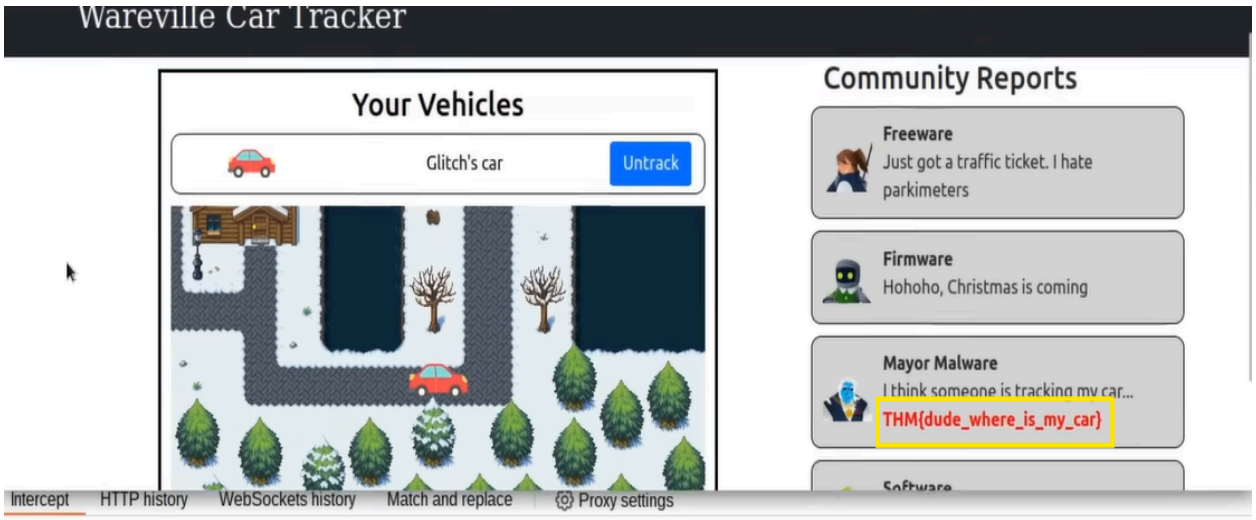


- Burp Proxy will intercept the WebSocket traffic, as shown below.



- Change the value of the userId parameter from **5 to 8** and click the Forward button.

Go back to your browser and check the community reports.



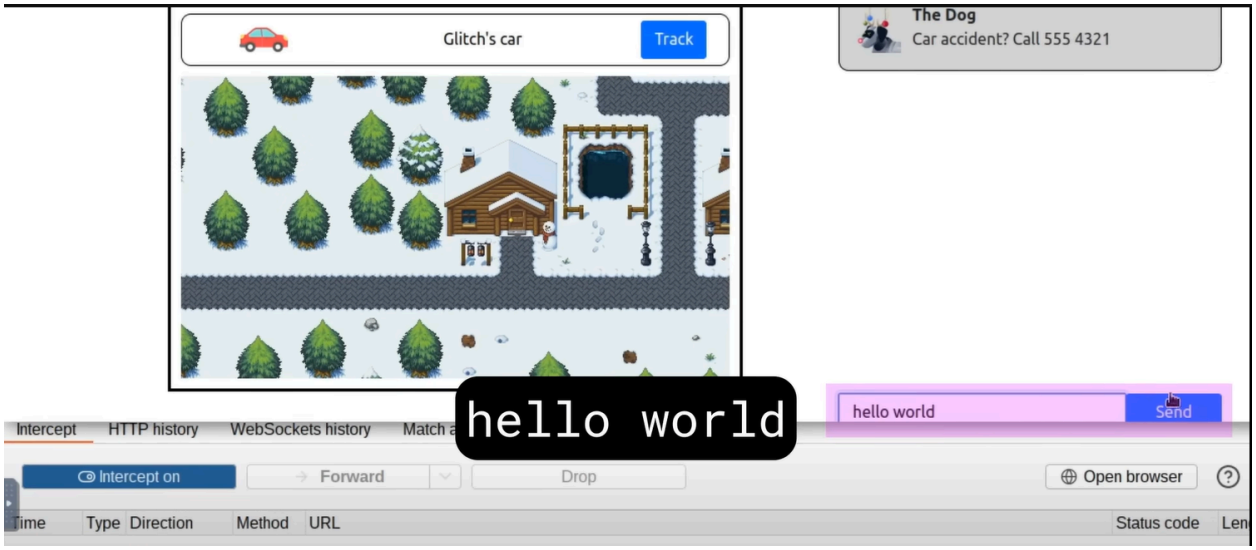
Note: If you don't see the traffic. Try to click the untrack button, refresh the page, and hit the track button again.

Manipulating the Messaging

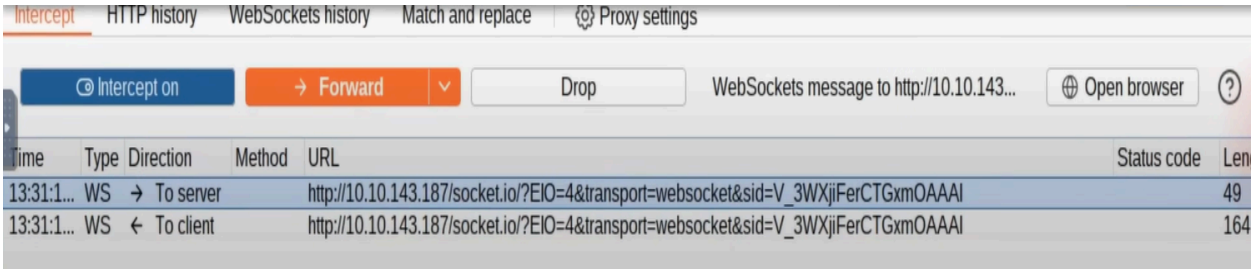
- ***Moving ahead in the event !!***

Following the successful identification of the WebSocket Message Manipulation vulnerability, Glitch continued testing for other ways to exploit the application. This time, he wanted to see if the messages posted on the app could be altered and manipulated. Is it possible to post using a different user ID?

- We try send a real time message and tried to exploit the message vulnerability by by changing the user id which does an impersonation message.



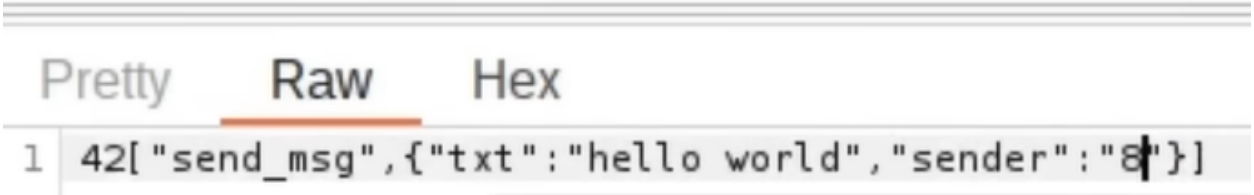
- So we can find the traffic on the burpsuite we find Type section holds - Websocket to server and vise-versa.



- Clicking on it .

- We can find that the message we have typed is shown and the sender's id -5 is shown so let's manipulate it .

- We changed the id 5 to 8 to impersonate the person and sent a message pretending to be someone else because of this vulnerability.



- After forwarding the message we were able to send the message and also got the flag we needed to solve this lab .