## Prevention

Different WebSocket vulnerabilities have different methods of prevention. Preventing the CSRF attack on the WebSocket handshake prevents CSWH attacks. Potential countermeasures include checking the [Origin header](https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Origin), implementing CSRF tokens, or secure configuration of the [SameSite](https://web.dev/i18n/en/samesite-cookies-explained/) cookie flag.

Furthermore, there are some general security considerations we should follow when implementing WebSocket connections:

* Always prefer the wss:// scheme over ws:// due to the security provided by TLS
* Sanitize data received over WebSocket connections accordingly, just like we sanitize data received in HTTP requests. The sanitization needs to correspond to the purpose of the data received, for instance, if used in SQL queries or inserted into the DOM to prevent XSS. In particular, the data needs to be treated as untrusted in both directions, i.e., the server should not trust data received by the client, and the client should not trust data received by the server