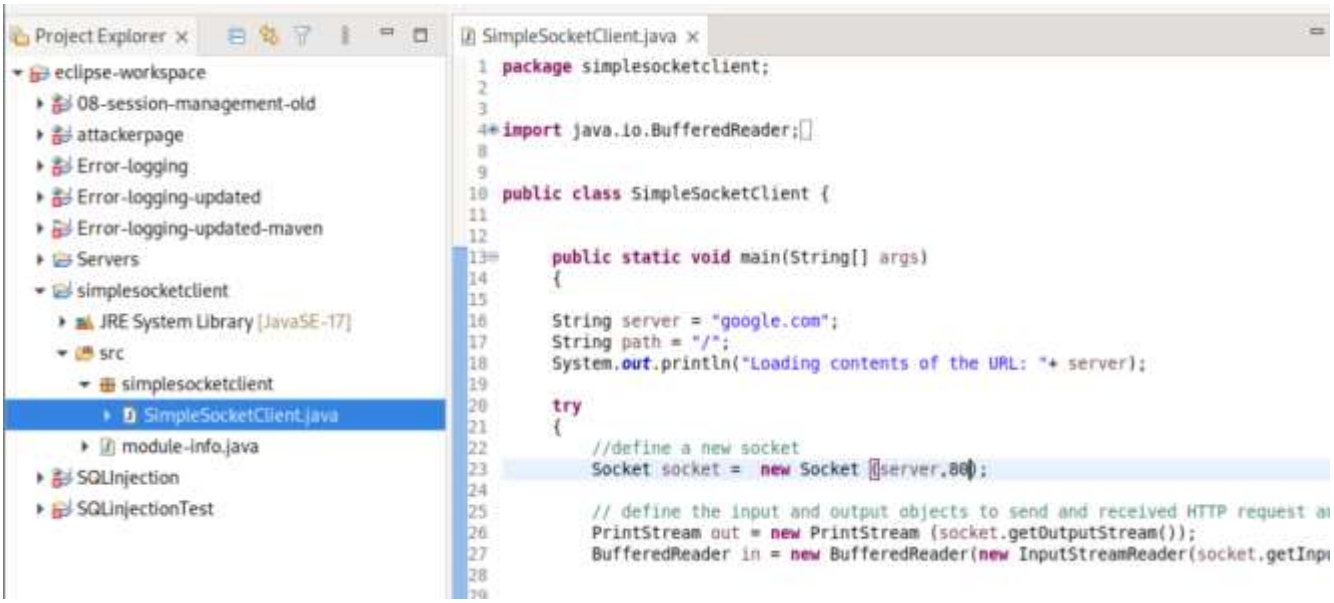
Pass task 2.1P: Web server security

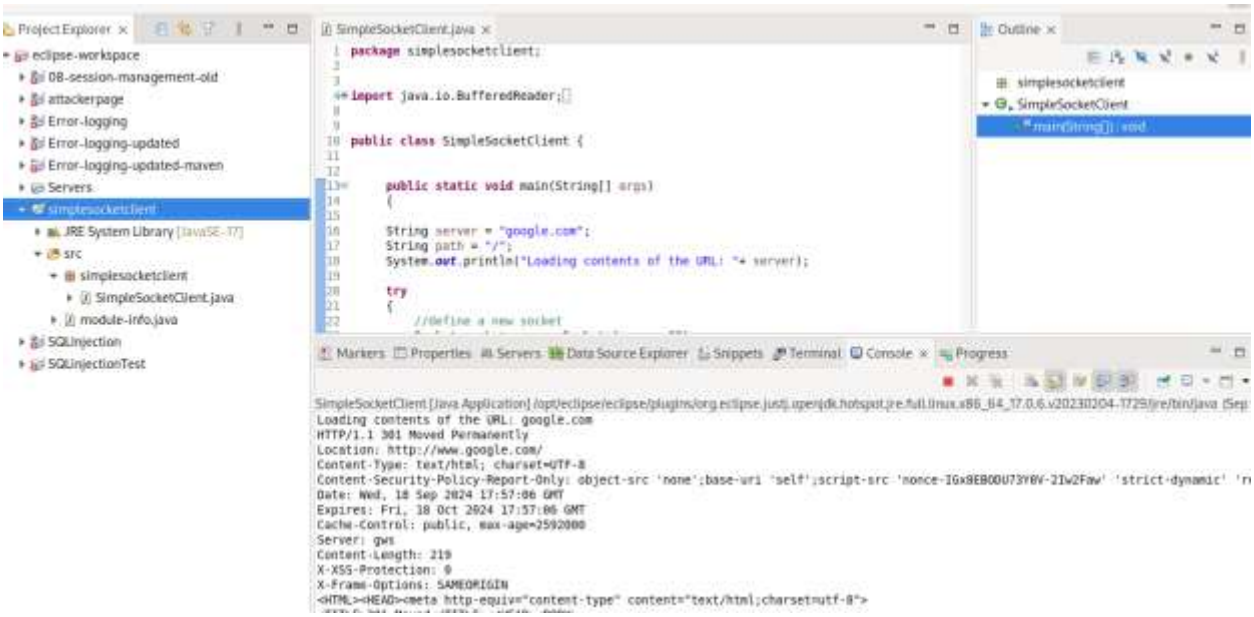
- The objective of the Task 2.1p is to secure the Tomcat server more efficiently. Web server security aims at minimizing the chances that somebody might find out that a specific version of the server is being used and exploits the vulnerabilities inherent in that version. One has to adjust a number of settings within the server to ensure that no other information, including the server banner or the version number, are provided in the answer that Tomcat returns. This way, efforts are made to keep the attacker from obtaining information which may be imperative for launching an attack.

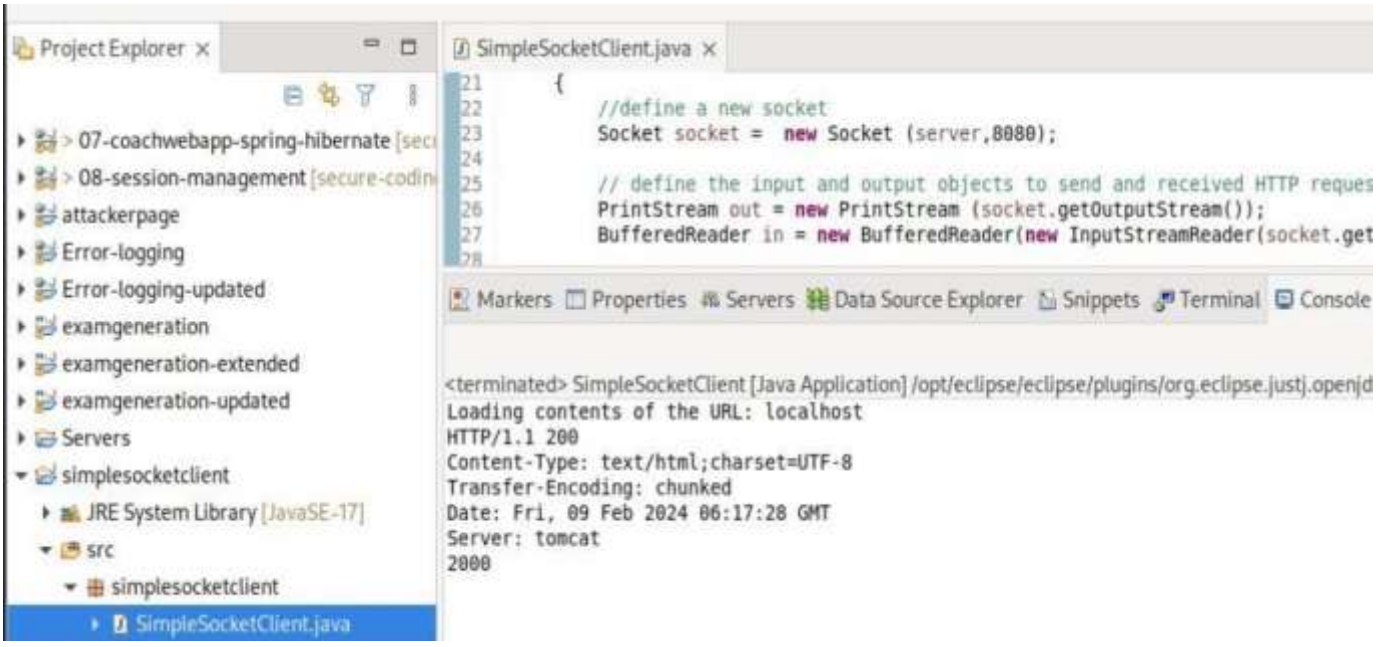
- To have more manageability of the server environment, I deployed the Tomcat server to start and stop via systemctl commands other than the IDE of Eclipse. Particularly, I made use of:

sudo systemctl start tomcat

- When I first sent my request to this server it gave me a detailed response containing the version of Tomcat which I may or may not be using for something and I'm not certain if this server is being used for malignant purposes or not. When I rerun the request after the decision to change some server parameters the version number and other details that in my opinion should not be sent disappeared from the answer. It was lessened as the attacker executed the command which yielded a simplistic, rudimentary reply from the server.




Project Explorer
eclipse-workspace
08-session-management-old
attackerpage
Error-logging
Error-logging-updated
Error-logging-updated-maven
Servers
simplesocketclient
JRE System Library [JavaSE-17]
src
simplesocketclient
SimpleSocketClient.java
module-info.java
SQLInjection
SQLInjectionTest
SimpleSocketClient.java
package simplesocketclient;
import java.io.BufferedReader;
public class SimpleSocketClient {
public static void main(String[] args)
{
String server = "google.com";
String path = "/";
System.out.println("Loading contents of the URL: "+ server);
try
{
//define a new socket
Outline
simplesocketclient
SimpleSocketClient
main(String[]) : void
Markers
Properties
Servers
Data Source Explorer
Snippets
Terminal
Console
Progress
Loading contents of the URL: google.com
HTTP/1.1 301 Moved Permanently
Location: http://www.google.com/
Content-Type: text/html; charset=UTF-8
Content-Security-Policy-Report-Only: object-src 'none';base-uri 'self';script-src 'nonce-IGx8EBODU73Y8V-2Iw2Faw' 'strict-dynamic' 'r
Date: Wed, 18 Sep 2024 17:57:06 GMT
Expires: Fri, 18 Oct 2024 17:57:06 GMT
Cache-Control: public, max-age=2592000
Server: gws
Content-Length: 219
X-XSS-Protection: 0
X-Frame-Options: SAMEORIGIN
<HTML><HEAD><meta http-equiv="content-type" content="text/html;charset=utf-8">
<TITLE>301 Moved</TITLE></HEAD><BODY>
<H1>301 Moved</H1>
The document has moved
<A HREF="http://www.google.com/">here</A>.
</BODY></HTML>


Project Explorer
07-coachwebapp-spring-hibernate
08-session-management
attackerpage
Error-logging
Error-logging-updated
examgeneration
examgeneration-extended
examgeneration-updated
Servers
simplesocketclient
JRE System Library [JavaSE-17]
src
simplesocketclient
SimpleSocketClient.java
SimpleSocketClient.java
{
//define a new socket
Socket socket = new Socket (server,8080);
// define the input and output objects to send and received HTTP reques
PrintStream out = new PrintStream (socket.getOutputStream());
BufferedReader in = new BufferedReader(new InputStreamReader(socket.get
Markers
Properties
Servers
Data Source Explorer
Snippets
Terminal
Console
<terminated> SimpleSocketClient [Java Application] /opt/eclipse/eclipse/plugins/org.eclipse.justj.openjd
Loading contents of the URL: localhost
HTTP/1.1 200
Content-Type: text/html;charset=UTF-8
Transfer-Encoding: chunked
Date: Fri, 09 Feb 2024 06:17:28 GMT
Server: tomcat
2000

1. Write a small paragraph reflecting on which secure coding principle we are applying by hardening the tomcat server?

- In this case, we are working under the assumption that attackable surface area must be minimized by shielding the Tomcat server, as well as removing as much versions and banners as we can. The basic purpose of this principle is to minimize the flow of information available about the potential attackers in the world. For instance, diagnostic data should be disclosed in small portions as it gradually decreases the Web server's vulnerability level.