Burp Suite: SQL Injection Vulnerability

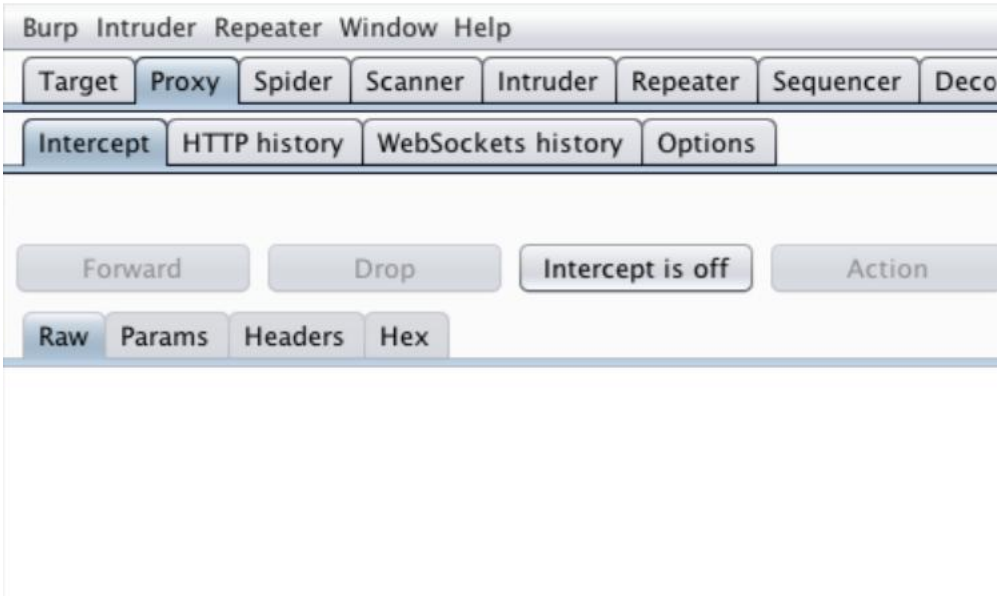
A SQL injection (SQLi) attack involves exploiting vulnerabilities in a web application's input fields to execute unauthorized SQL queries, thereby manipulating the backend database. The attack starts with identifying input fields where user data is submitted, such as login forms or search bars. An attacker inputs malicious SQL code, often crafted to alter the intended SQL query. For example, inputting `'; DROP TABLE users; --` in a vulnerable login field could terminate the original query and append a destructive command. Using tools like Burp Suite, the attacker intercepts the HTTP request containing the user input, and modifies it to include SQL injection payloads. The modified request is then sent to the server. If the application does not properly sanitize the input, the injected SQL code executes, potentially revealing data, bypassing authentication, or altering database structures. Analyzing the server's responses helps the attacker refine their payloads, incrementally gaining more control over the database. SQLi attacks highlight the critical need for proper input validation and parameterized queries to safeguard web applications against such exploits.

A SQL injection (SQLi) attack using Burp Suite involves exploiting vulnerabilities in a web application's input fields to

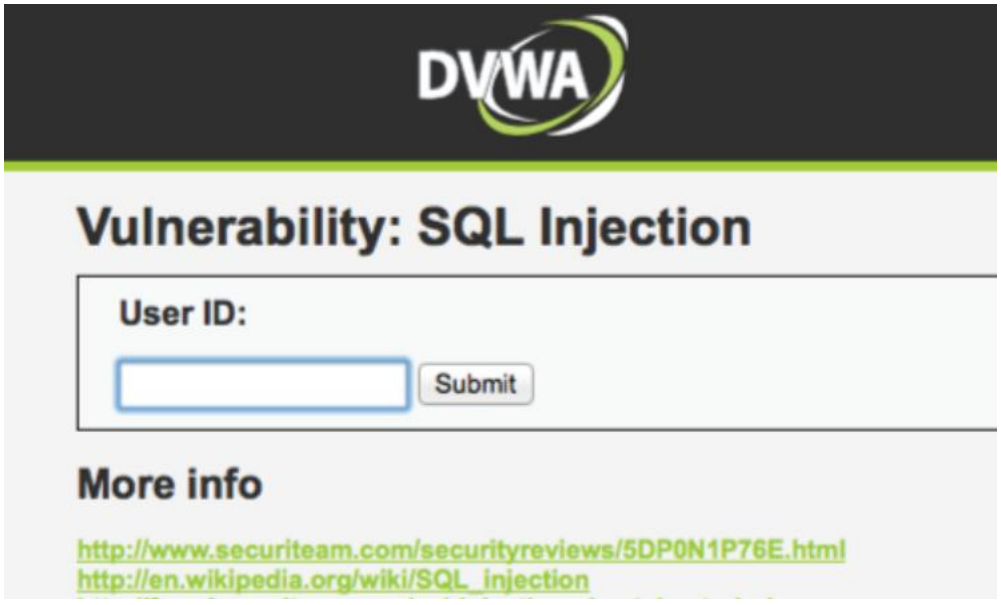
execute unauthorized SQL queries. To perform an SQLi attack, we will start by setting up Burp Suite as a proxy and configuring your browser to route traffic through it, enabling Burp Suite to capture and analyze HTTP/S requests and responses. Next, navigate to the target web application's input field (such as a login form or search bar) and submit a request with typical SQL injection payloads like `' OR '1'='1'`. Intercept the request using Burp Suite and send it to the Repeater tool for further manipulation. In Repeater, you can modify the input field to include various SQL payloads and resend the request to observe the application's response. Analyze the responses to identify any indications of successful SQL injection, such as error messages or unexpected data retrieval. By systematically testing different payloads and analyzing the server's responses, you can identify and exploit SQL injection vulnerabilities, potentially gaining unauthorized access to sensitive data or compromising the database.

Examples: -

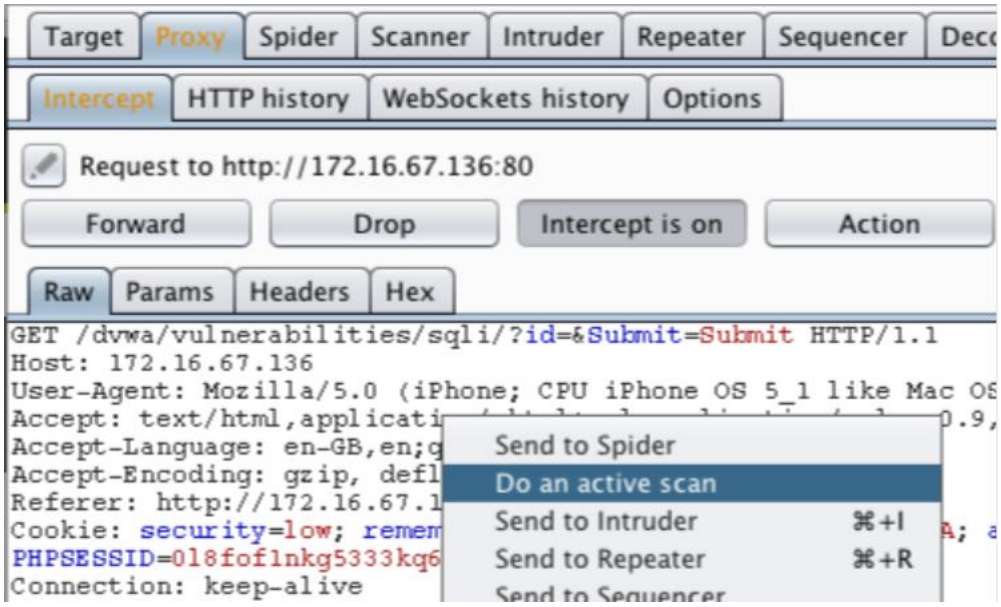
1. Firstly open the Burp Suite and turn on Burp Suite Proxy's Intercept and browser's proxy as well.



2. Now visit the test website and send a request to the server by clicking on submit button.



3. Now we have captured the request inside the proxy tab. Now right click and chose the Do an active scan option.



4. After the scanning is completed click on Advisory tab to have details of each vulnerability. Here we can see all the SQL Injection vulnerabilities.