Lab 2: Detect SQL Injection Vulnerabilities using Various SQL Injection Detection Tools

Lab Scenario

By now, you will be familiar with various types of SQL injection attacks and their possible impact. To recap, the different kinds of SQL injection attacks include authentication bypass, information disclosure, compromised data integrity, compromised availability of data and remote code execution (which allows identity spoofing), damage to existing data, and the execution of system-level commands to cause a denial of service from the application.

As an ethical hacker or pen tester, you need to test your organization's web applications and services against SQL injection and other vulnerabilities, using various approaches and multiple techniques to ensure that your assessments, and the applications and services themselves, are robust.

In the previous lab, you learned how to use SQL injection attacks on the MSSQL server database to test for website vulnerabilities.

In this lab, you will learn how to test for SQL injection vulnerabilities using various other SQL injection detection tools.

Lab Objectives

- Detect SQL injection vulnerabilities using OWASP ZAP

Overview of SQL Injection Detection Tools

SQL injection detection tools help to discover SQL injection attacks by monitoring HTTP traffic, SQL injection attack vectors, and determining if a web application or database code contains SQL injection vulnerabilities.

To defend against SQL injection, developers must take proper care in configuring and developing their applications in order to make them robust and secure. Developers should use best practices and countermeasures to prevent their applications from becoming vulnerable to SQL injection attacks.

Task 1: Detect SQL Injection Vulnerabilities using OWASP ZAP

OWASP Zed Attack Proxy (ZAP) is an integrated penetration testing tool for finding vulnerabilities in web applications. It offers automated scanners and a set of tools that allow you to find security vulnerabilities manually. It is designed to be used by people with a wide range of security experience, and as such is ideal for developers and functional testers who are new to penetration testing.

In this task, we will use OWASP ZAP to test a web application for SQL injection vulnerabilities.

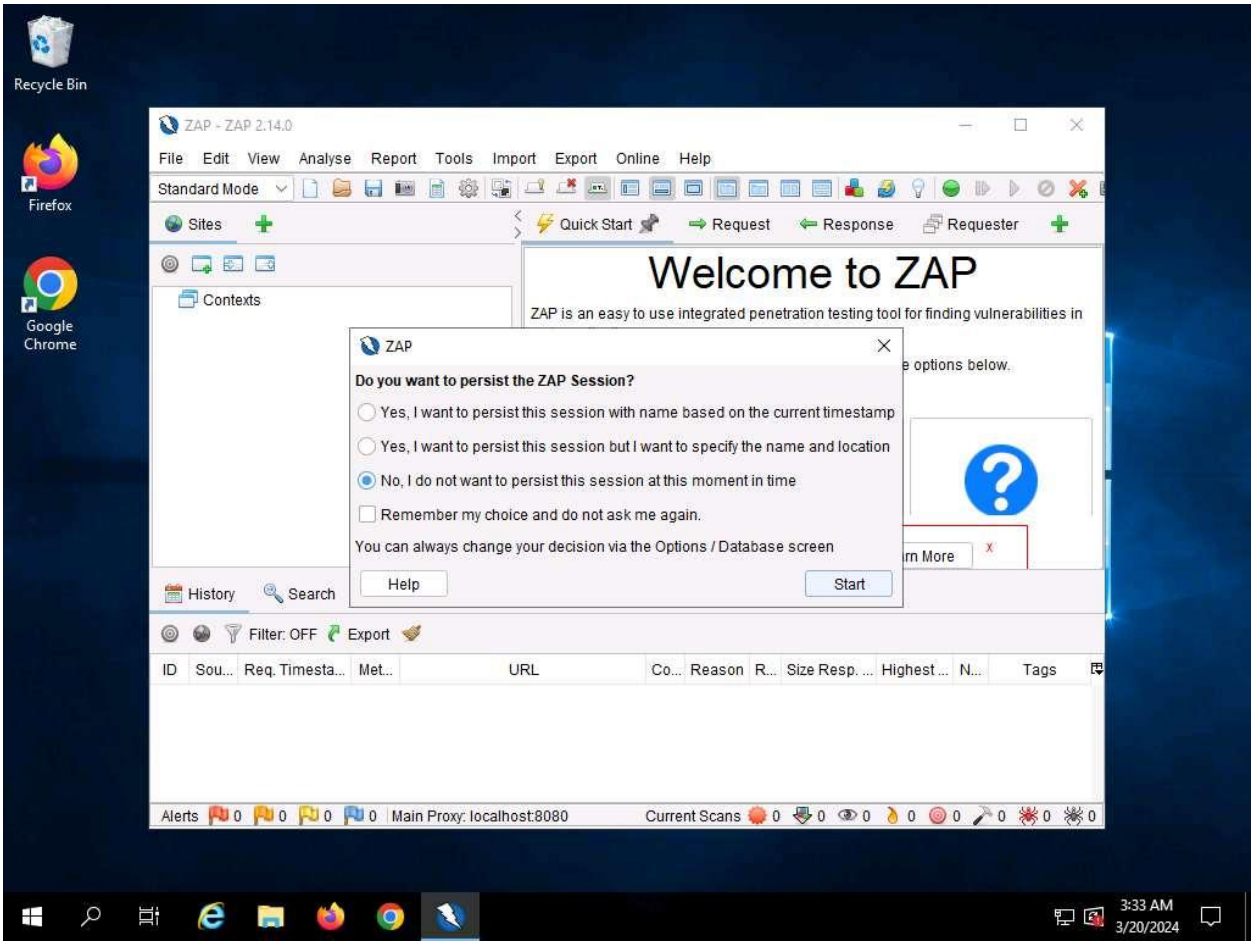
We will scan the **www.moviescope.com** website that is hosted on the **Windows Server 2019** machine.

1. Click Windows Server 2019 to switch to the **Windows Server 2019** machine.

If you are logged out of the **Windows Server 2019** machine, click Ctrl+Alt+Delete, and login with **Administrator/Pa$$w0rd**.

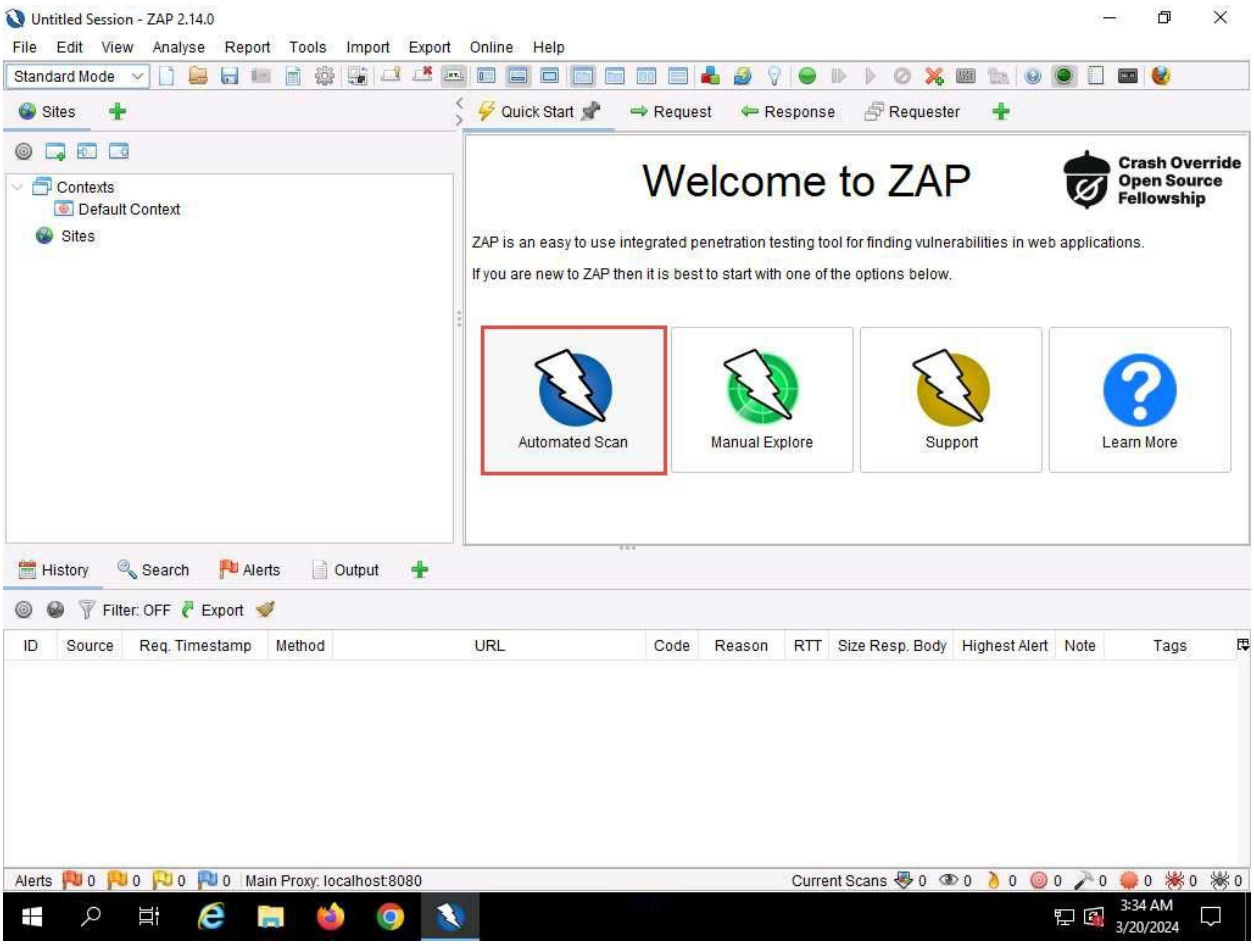
2. Click windows **Search** icon, search for **Zap 2.14.0** in the search bar and launch **ZAP**.
3. OWASP ZAP initialized and a prompt that reads **Do you want to persist the ZAP Session?** appears; select the **No, I do not want to persist this session at this moment in time** radio button, and click **Start**.

If a **Manage Add-ons** window appears, close it.

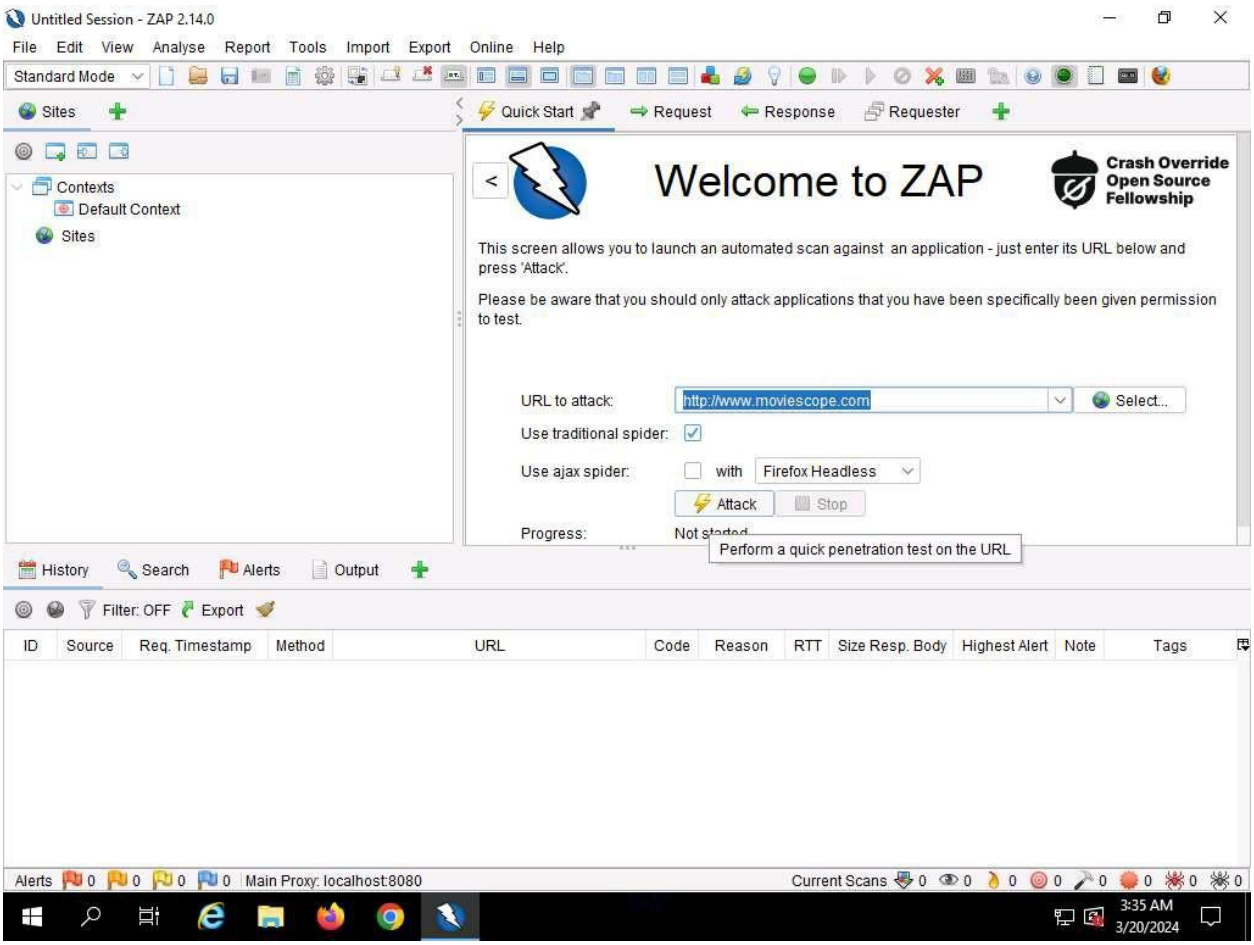


4. The **OWASP ZAP** main window appears; under the **Quick Start** tab, click the **Automated Scan** option.

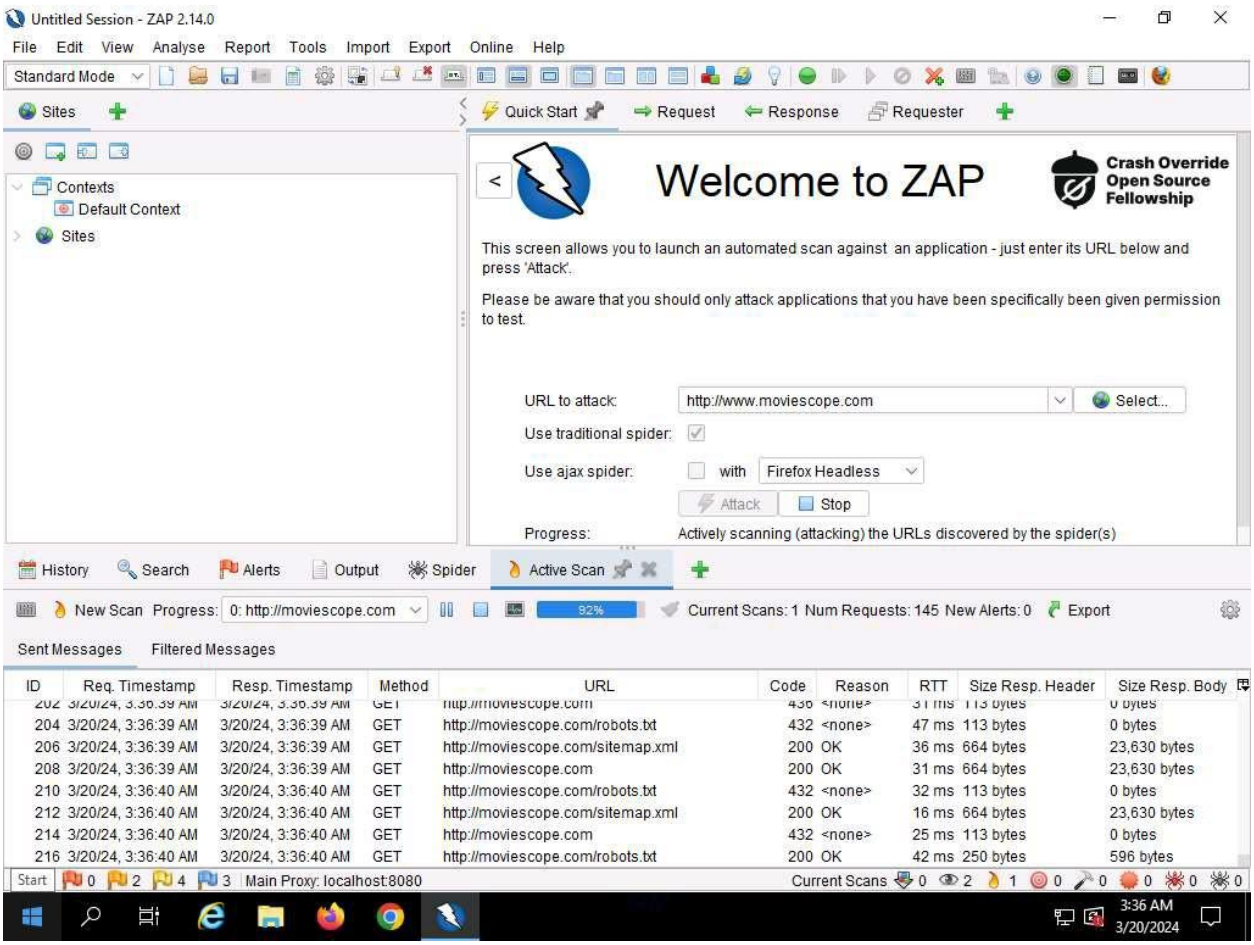
If OWASP ZAP alert pop-up appears, click **OK** in all the pop-ups.



5. The **Automated Scan** wizard appears, enter the target website in the **URL to attack** field (in this case, **http://www.moviescope.com**). Leave other options set to default, and then click the **Attack** button.

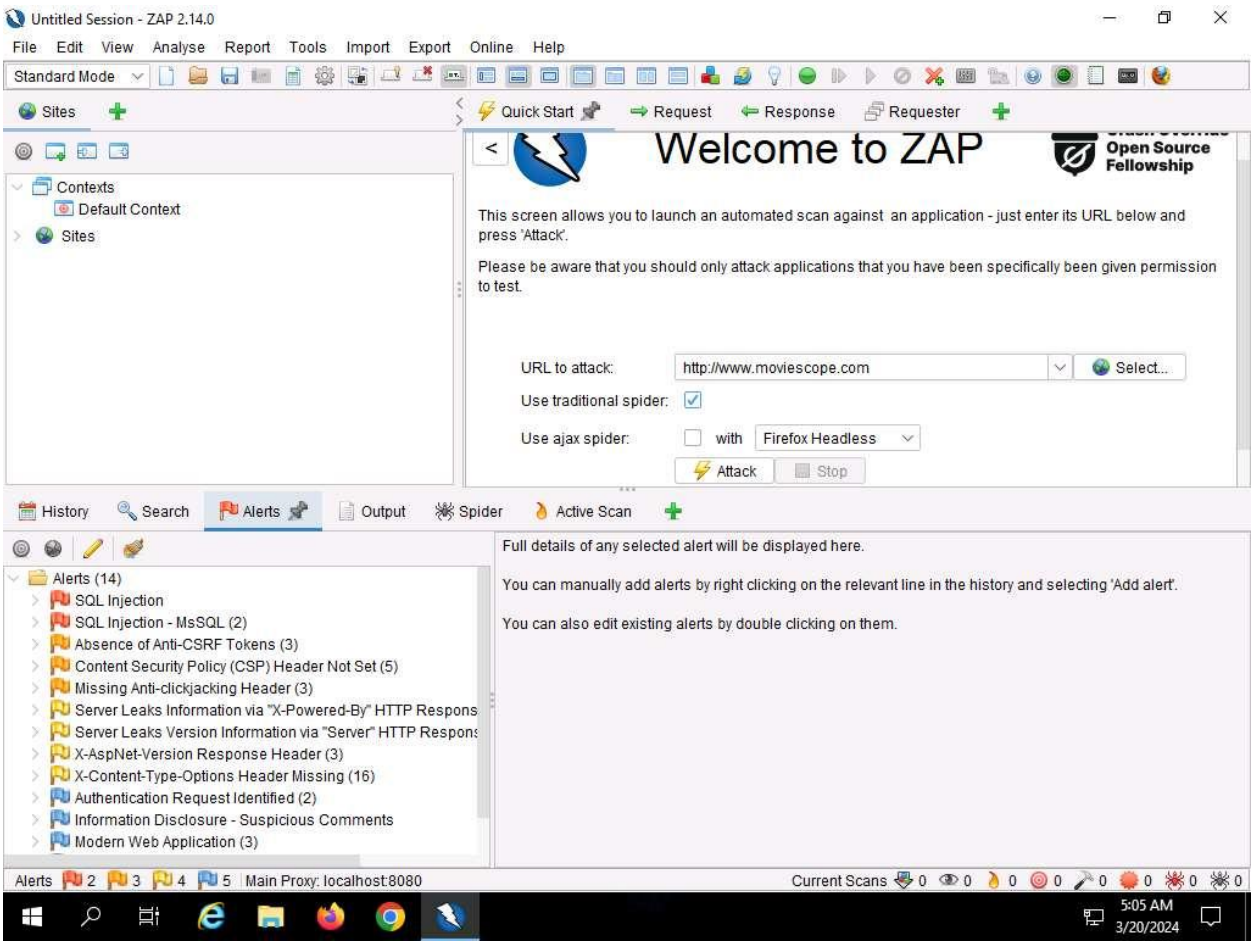


6. **OWASP ZAP** starts performing **Active Scan** on the target website, as shown in the screenshot.

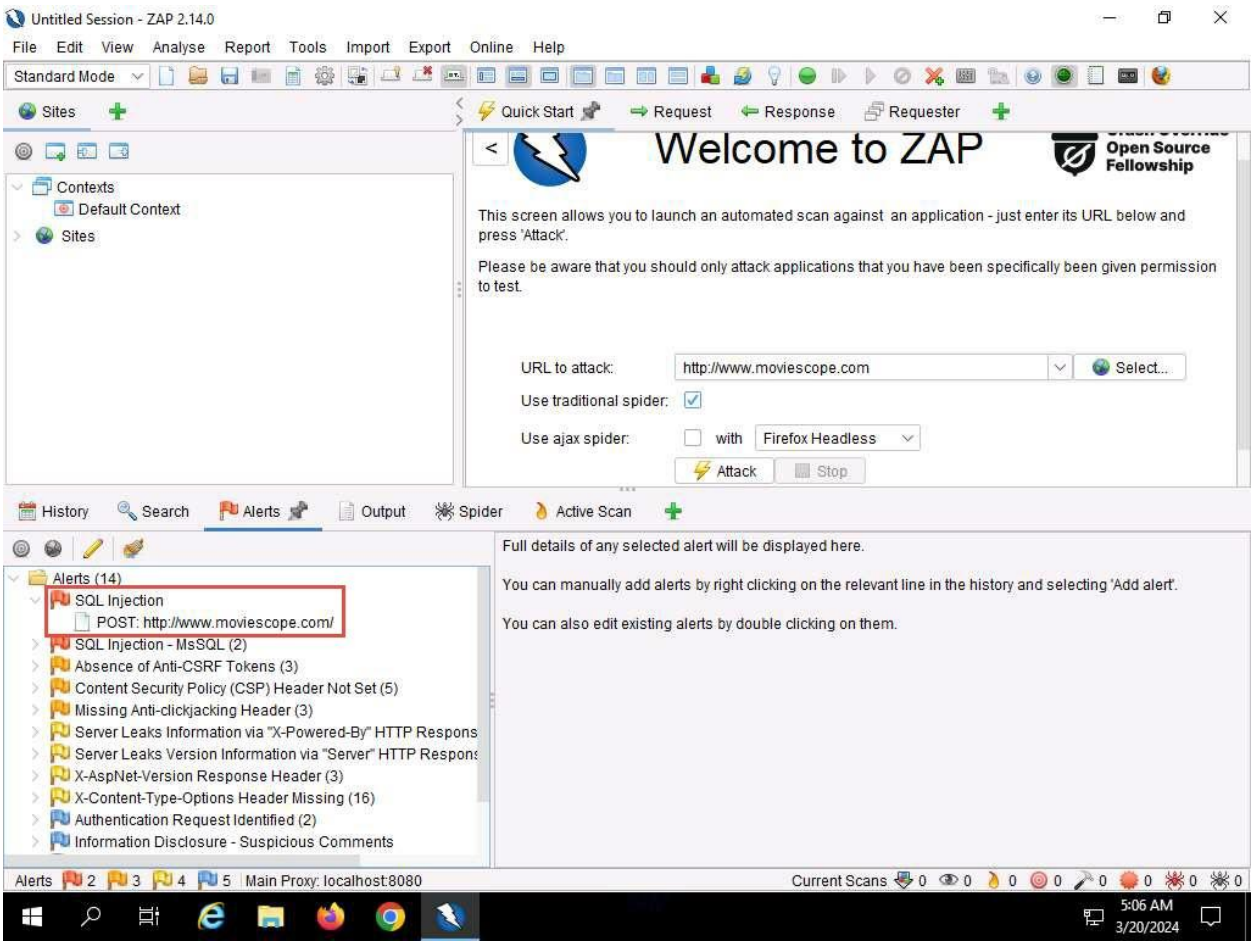


7. After the scan completes, **Alerts** tab appears. You can observe the vulnerabilities found on the website under the **Alerts** tab.

The discovered vulnerabilities might differ when you perform this task.



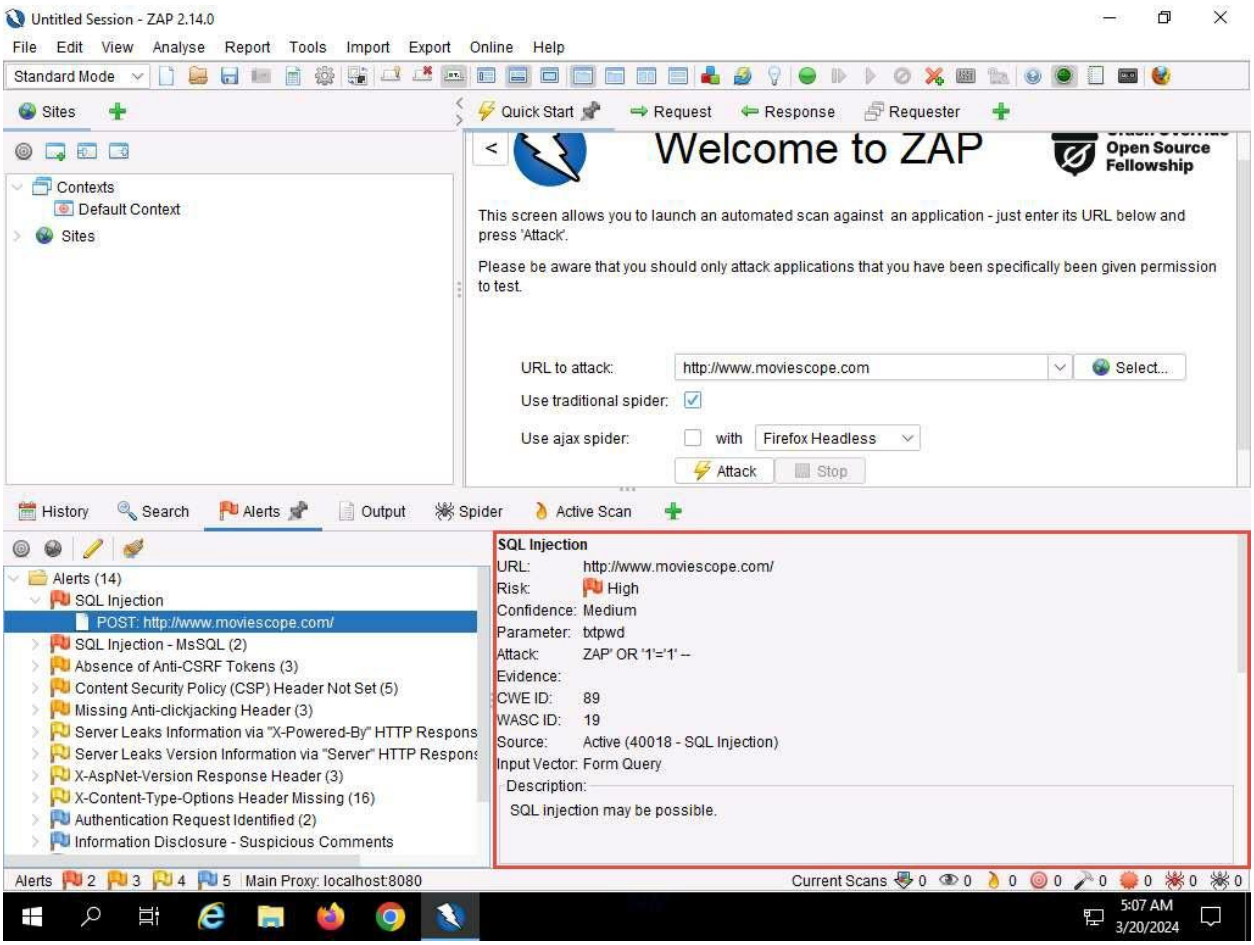
8. Now, expand the **SQL Injection** vulnerability node under the **Alerts** tab.



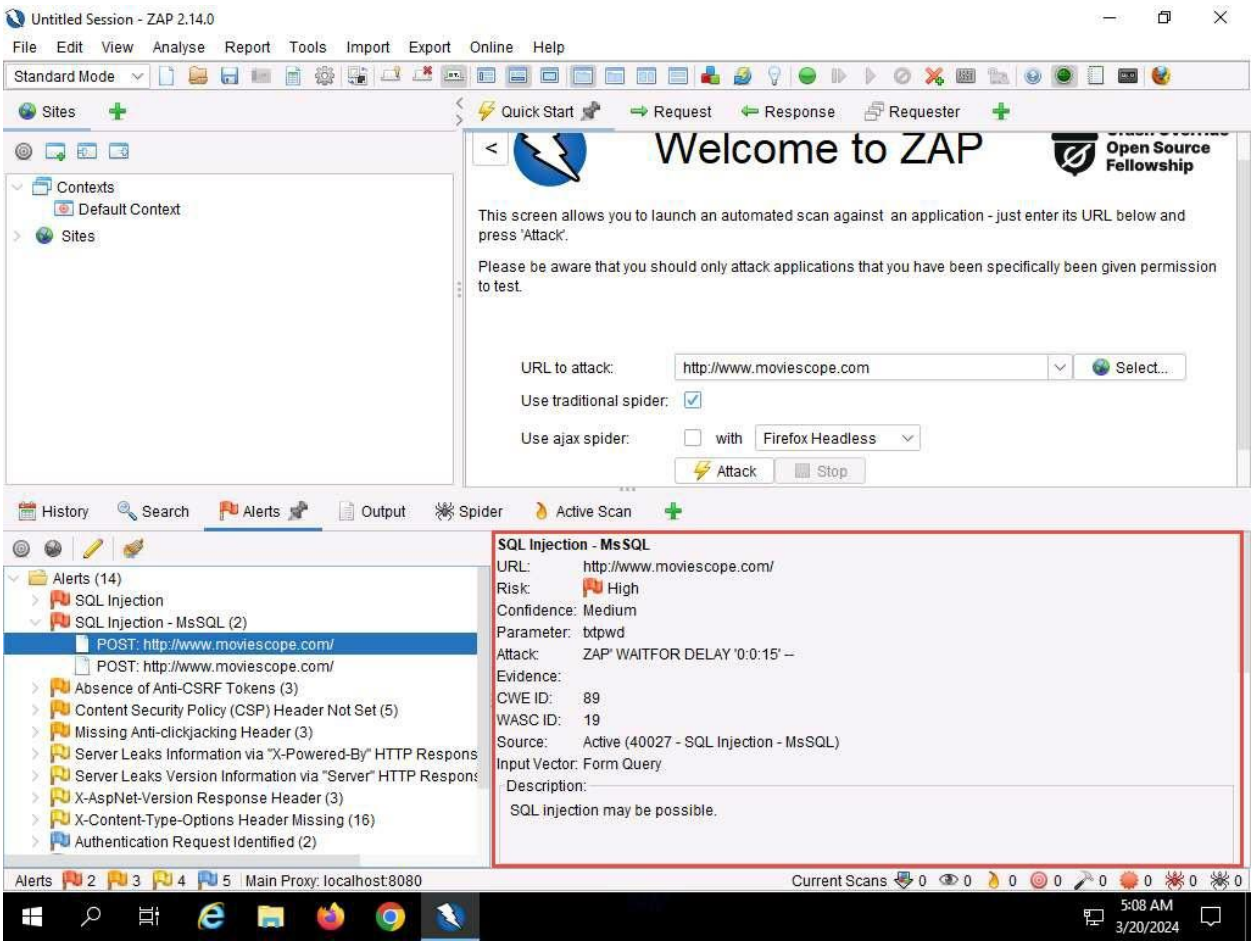
9. Click on the discovered **SQL Injection** vulnerability and further click on the vulnerable URL.
10. You can observe the information such as **Risk**, **Confidence**, **Parameter**, **Attack**, etc., regarding the discovered SQL Injection vulnerability in the lower right-bottom, as shown in the screenshot.

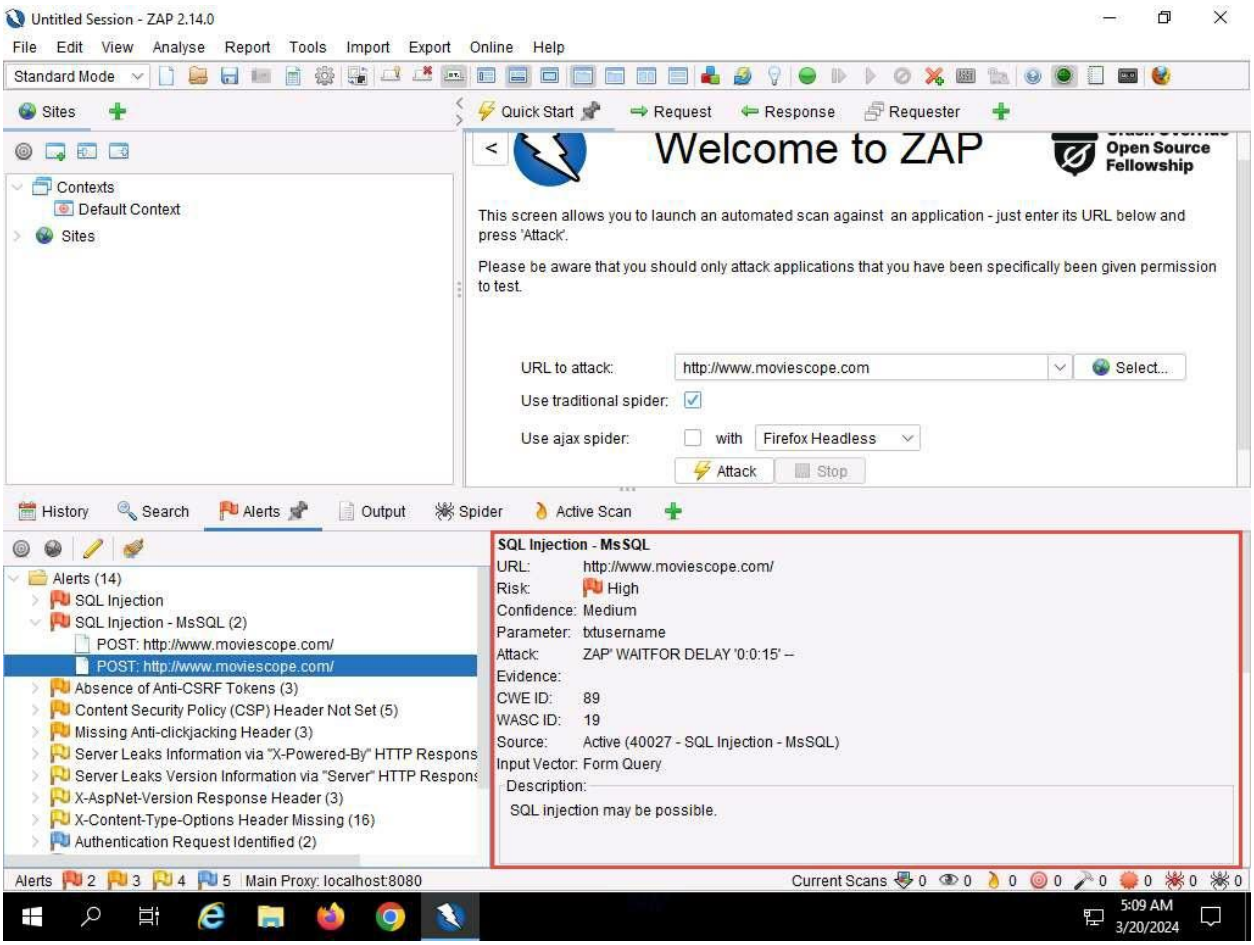
The risks associated with the vulnerability are categorized according to severity of risk as Low, Medium, High, and Informational alerts. Each level of risk is represented by a different flag color:

- **Red Flag**: High risk
- **Orange Flag**: Medium risk
- **Yellow Flag**: Low risk
- **Blue Flag**: Provides details about information disclosure vulnerabilities



11. Similarly, expand any other vulnerability (here, **SQL Injection-MsSQL**) node under the **Alerts** tab and further click on the vulnerable URLs.


Untitled Session - ZAP 2.14.0
File Edit View Analyse Report Tools Import Export Online Help
Standard Mode
Sites
Quick Start
Request
Response
Requester
Contexts
Default Context
Sites
Welcome to ZAP
Open Source Fellowship
This screen allows you to launch an automated scan against an application - just enter its URL below and press 'Attack'.
Please be aware that you should only attack applications that you have been specifically been given permission to test.
URL to attack:
http://www.moviescope.com
Select...
Use traditional spider:
Use ajax spider:
with
Firefox Headless
Attack
Stop
History
Search
Alerts
Output
Spider
Active Scan
Alerts (14)
SQL Injection
SQL Injection - MsSQL (2)
POST: http://www.moviescope.com/
POST: http://www.moviescope.com/
Absence of Anti-CSRF Tokens (3)
Content Security Policy (CSP) Header Not Set (5)
Missing Anti-clickjacking Header (3)
Server Leaks Information via "X-Powered-By" HTTP Respons
Server Leaks Version Information via "Server" HTTP Respons
X-AspNet-Version Response Header (3)
X-Content-Type-Options Header Missing (16)
Authentication Request Identified (2)
SQL Injection - MsSQL
URL: http://www.moviescope.com/
Risk: High
Confidence: Medium
Parameter: txtpwd
Attack: ZAP' WAITFOR DELAY '0:0:15' --
Evidence:
CWE ID: 89
WASC ID: 19
Source: Active (40027 - SQL Injection - MsSQL)
Input Vector: Form Query
Description:
SQL injection may be possible.
Alerts 2 3 4 5 Main Proxy: localhost:8080
Current Scans 0 0 0 0 0 0 0 0
5:08 AM
3/20/2024



12. This concludes the demonstration of how to detect SQL injection vulnerabilities using OWASP ZAP.
13. Close all open windows and document all the acquired information.
14. You can also use other SQL injection detection tools such as **Damn Small SQLi Scanner** (**DSSS**) (https://github.com), Snort (https://snort.org), **Burp Suite** (https://www.portswigger.net), **HCL AppScan** (https://www. hcl-software.com) etc. to detect SQL injection vulnerabilities.

Question 15.2.1.1

Use OWASP ZAP to test a web application (www.moviescope.com) for SQL injection vulnerabilities. Enter the CWE ID of the SQL injection vulnerability found in www.moviescope.com.

Question 15.2.1.2

Use OWASP ZAP to test a web application (www.moviescope.com) for SQL injection vulnerabilities. Enter the WASC ID of the SQL injection vulnerability found in www.moviescope.com.