Homework 9: Security Testing

Author: Syed Zain Raza

PART 1

SQL Injection Challenge:

Webgoat is server now changed so SQL string problem is at step 9 now.

Question:

What is the string that you entered into the name field to solve this challenge?

Answer:

I used the input as last_name = ‘ , or , ‘1’ = ‘1

Question:

What is Hershey Jolly’s AMEX Credit Card Number?

Answer:

Hershey Jolly’s AMEX credit card number is 33300003333

SQL Injection Advanced

Question:

What is the string that you entered into the user name field to solve this challenge?

Answer:

I wrote ‘; select * from user_system_data;-- in the username field

Question:

What is the password of user name dave?

Answer:

The password of dave is passW0rD

Explain How it Works

Question:

Describe in your own words how you attacked this site to reveal internal information?

Answer:

I attacked this website using SQL injection. As the website was using SQL queries behind the scenes. I was able to utilize and this scripts which will return required values. I learned this from SQL Injection tutorials on WebGoat website.

Question:

What would you do to prevent this attack in your website?

Answer:

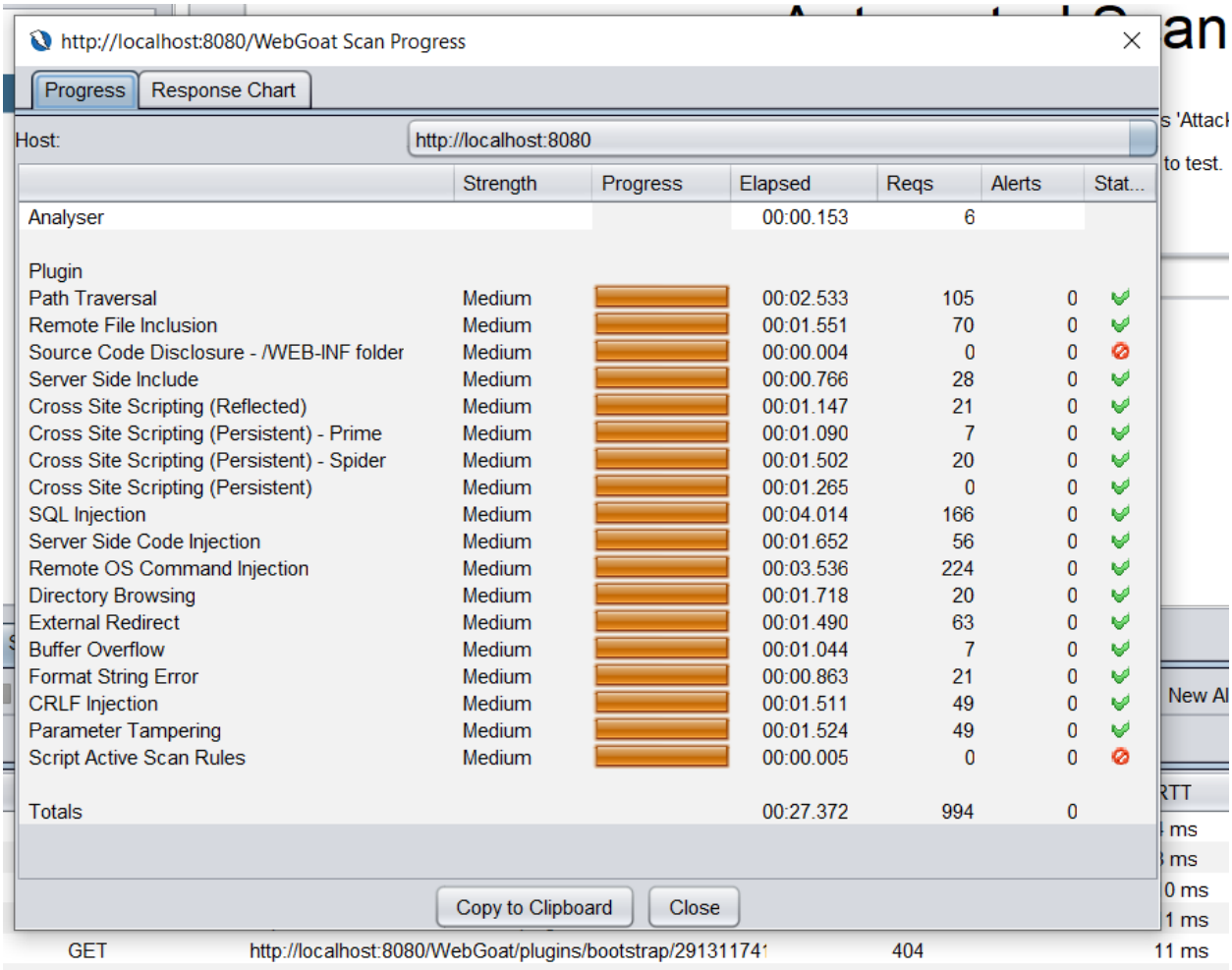
This type of attacks can be prevented by making SQL queries more structured and while writing these queries keep in mind the intentions of the hackers. This will help in resisting such attacks.

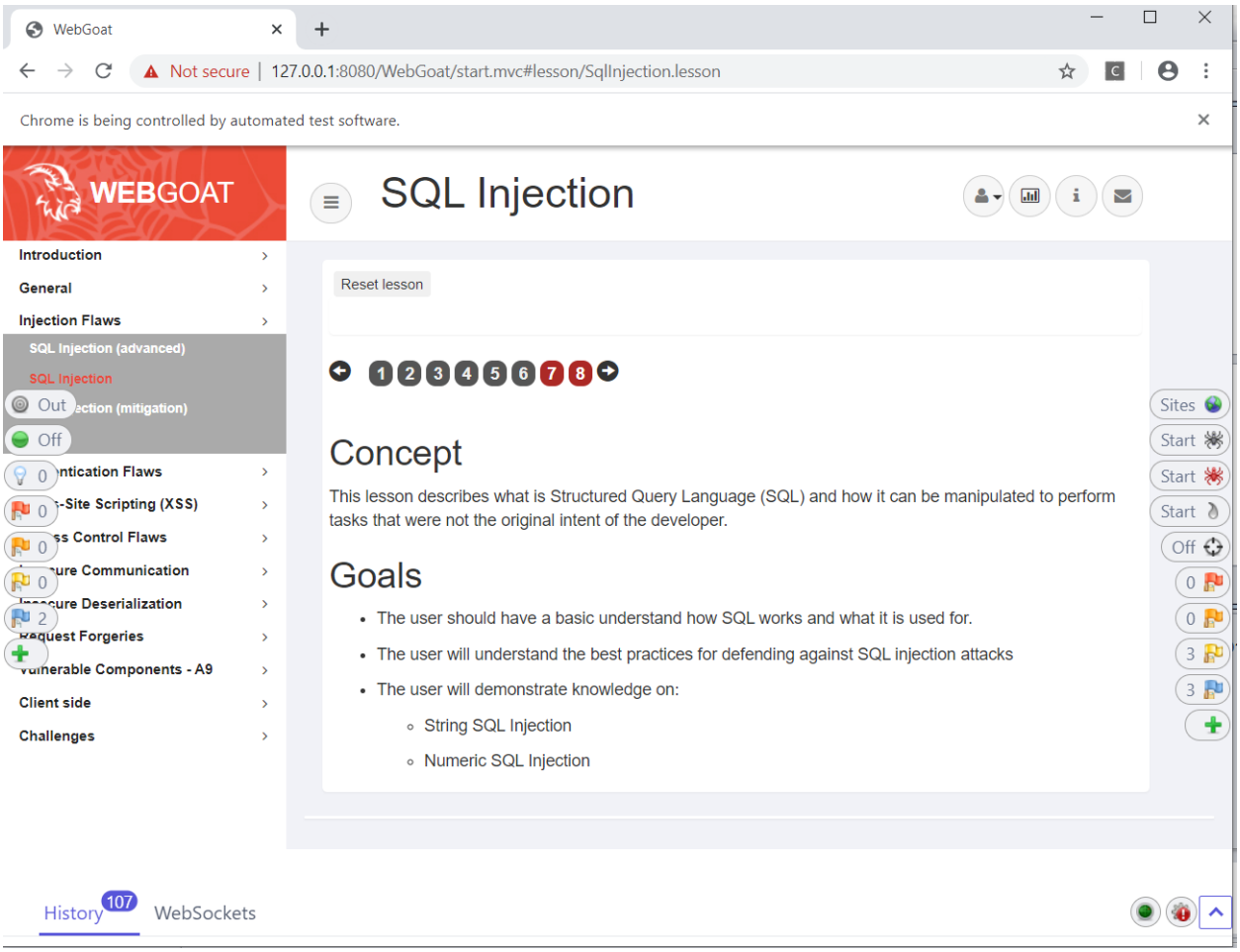
PART 2

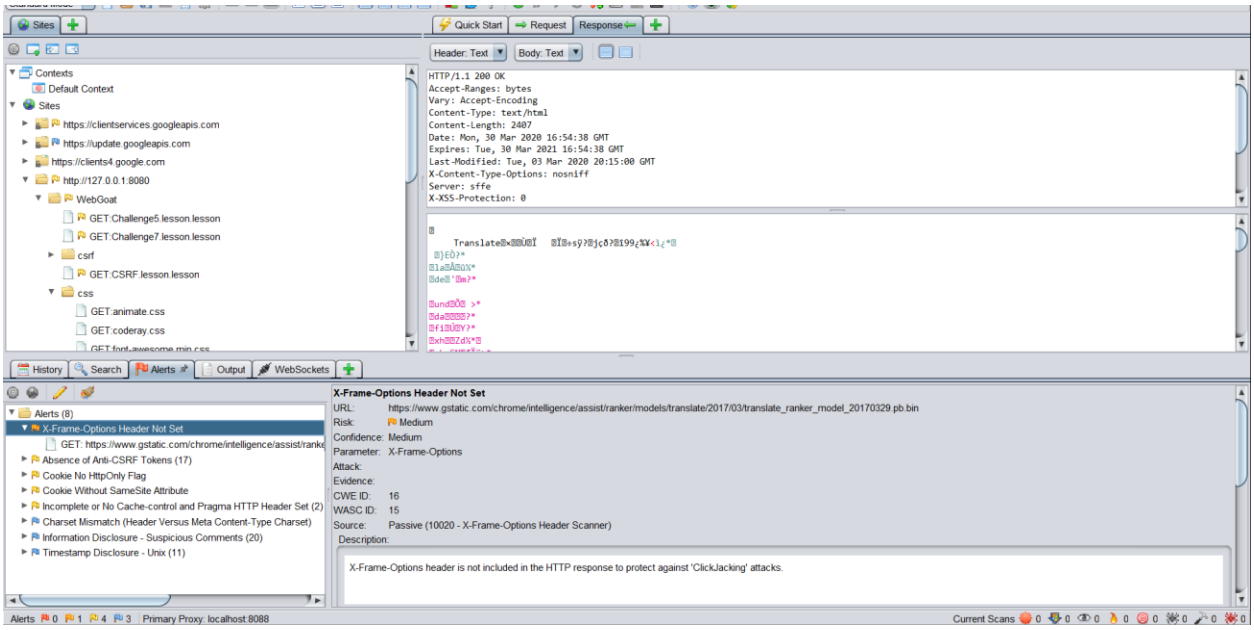
Question:

List and describe all of the high risk alerts that ZAP identifies with your local instance of WebGoat.

Answer:


http://localhost:8080/WebGoat Scan Progress
Progress
Response Chart
Host:
http://localhost:8080
Strength
Progress
Elapsed
Reqs
Alerts
Stat...
Analyser
00:00.153
6
Plugin
Path Traversal Medium 00:02.533 105 0
Remote File Inclusion Medium 00:01.551 70 0
Source Code Disclosure - /WEB-INF folder Medium 00:00.004 0 0
Server Side Include Medium 00:00.766 28 0
Cross Site Scripting (Reflected) Medium 00:01.147 21 0
Cross Site Scripting (Persistent) - Prime Medium 00:01.090 7 0
Cross Site Scripting (Persistent) - Spider Medium 00:01.502 20 0
Cross Site Scripting (Persistent) Medium 00:01.265 0 0
SQL Injection Medium 00:04.014 166 0
Server Side Code Injection Medium 00:01.652 56 0
Remote OS Command Injection Medium 00:03.536 224 0
Directory Browsing Medium 00:01.718 20 0
External Redirect Medium 00:01.490 63 0
Buffer Overflow Medium 00:01.044 7 0
Format String Error Medium 00:00.863 21 0
CRLF Injection Medium 00:01.511 49 0
Parameter Tampering Medium 00:01.524 49 0
Script Active Scan Rules Medium 00:00.005 0 0
Totals
00:27.372
994
0
Copy to Clipboard
Close
GET
http://localhost:8080/WebGoat/plugins/bootstrap/29131174
404
11 ms

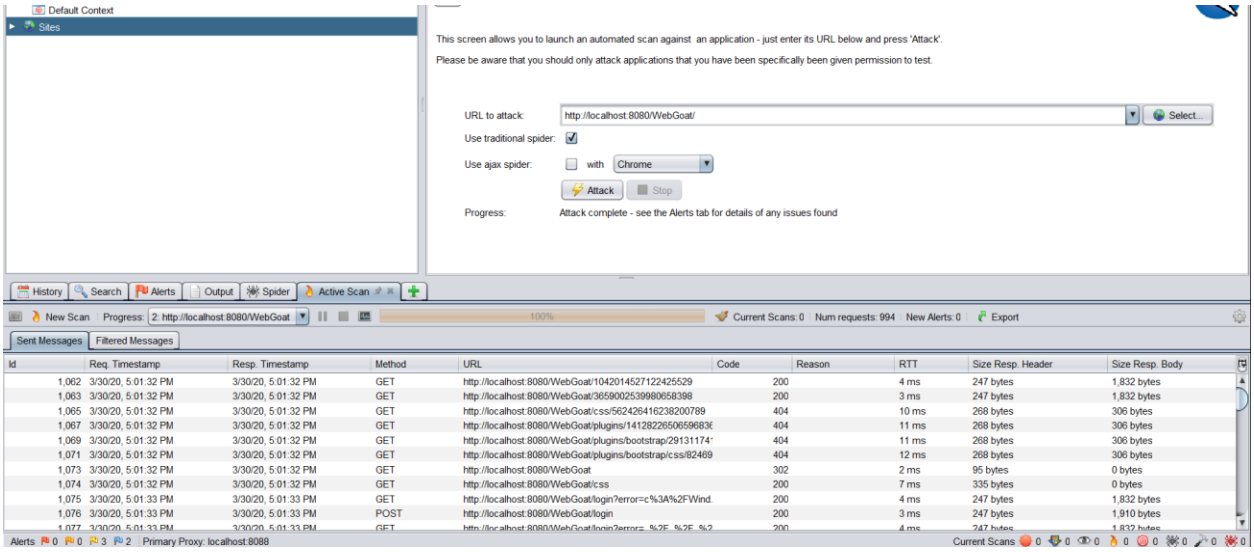

WebGoat
Not secure | 127.0.0.1:8080/WebGoat/start.mvc#lesson/SqlInjection.lesson
Chrome is being controlled by automated test software.
WEBGOAT
SQL Injection
Introduction
General
Injection Flaws
SQL Injection (advanced)
SQL Injection
Reset lesson
1 2 3 4 5 6 7 8
Concept
This lesson describes what is Structured Query Language (SQL) and how it can be manipulated to perform tasks that were not the original intent of the developer.
Goals
The user should have a basic understand how SQL works and what it is used for.
The user will understand the best practices for defending against SQL injection attacks
The user will demonstrate knowledge on:
String SQL Injection
Numeric SQL Injection
History 107
WebSockets


Sites
Quick Start
Request
Response
Header: Text
Body: Text
HTTP/1.1 200 OK
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Type: text/html
Content-Length: 2407
Date: Mon, 30 Mar 2020 16:54:38 GMT
Expires: Tue, 30 Mar 2021 16:54:38 GMT
Last-Modified: Tue, 03 Mar 2020 20:15:00 GMT
X-Content-Type-Options: nosniff
Server: sffe
X-XSS-Protection: 0
History
Search
Alerts
Output
WebSockets
X-Frame-Options Header Not Set
URL: https://www.gstatic.com/chrome/intelligence/assist/ranker/models/translate/2017/03/translate_ranker_model_20170329.pb.bin
Risk: Medium
Confidence: Medium
Parameter: X-Frame-Options
Attack:
Evidence:
CWE ID: 16
WASC ID: 15
Source: Passive (10020 - X-Frame-Options Header Scanner)
Description:
X-Frame-Options header is not included in the HTTP response to protect against 'ClickJacking' attacks.

Question:

Include a screen shot of ZAP window after you have completed the scan in your report.

Answer:



Reflection:

This assignment was informative as I was able to explore security testing using tools such as OWASP ZAP and OWASP Web Goat. WebGoat was helpful in learning more about SQL Injection. ZAP was used to attack WebGoat to know how many problems it has. Overall, it was a really useful experience.

Honor Pledge:

"I pledge my honor that I have abided by the Stevens Honor System."