WEB APPLICATION SECURITY TESTING REPORT

TABLE OF CONTENTS

ABTRACT

OWASP ZAP (ZAP) is one of the world's most popular free security testing tools and is actively maintained by hundreds of international volunteers. It can help to find security vulnerabilities in web applications. It's also a great tool for experienced pen testers and beginners.

As it is designed to be used by people with a wide range of pen testing experience, it was ideal for our team who were new to penetration testing. As ZAP spiders the web application, it constructs a map of the web applications' pages and the resources used to render those pages. Then it records the requests and responses sent to each page and creates alerts if there is something potentially wrong with a request or response.

OBJECTIVE

The prime objective of security testing is to find out how vulnerable a system may be and to determine whether its data and resources are protected from potential intruders. Online transactions have increased rapidly of late making security testing as one of the most critical areas of testing for such web applications. A web application security test focuses only on evaluating the security of a web application. The process involves an active analysis of the application for any weaknesses, technical flaws, or vulnerabilities.

INTRODUCTION

Open Web Application Security Project is an open project that focuses on Web Application vulnerabilities and everything related to them. They have testing guides, tools, references, explanations, how to mitigate them etc. OWASP is like the main reference for all webapp vulnerabilities. This project is about usually runs from the end point of the person inspecting the attack surface. Scanner would be intelligent enough to compare details about the target attack surface to a database of information about known security holes in services/ports, anomalies in packet construction, and potential paths to exploitable programs or scripts. E.g. *Burp suit*. Attack surface and possible attacks gets updated periodically as and when new security vulnerability detected.

METHODOLOGY

Setting up ZAP

To begin with, you need to download and install OWASP ZAP scanner and set it up correctly. ZAP is platform agnostic so you can install it on Windows, Linux or Mac OS. You need Java 8+ installed on your Windows or Linux system.

Starting ZAP

Once setup you can start ZAP by clicking the ZAP icon on your Windows desktop or from the start menu.

When the app launches, it asks you whether you want to save the session or not. If you want to use the current run configuration or test results later, you should save the session for later. For now let's select *"No, I do not want to persist this session at this moment in time"*.

Once you click the *"Start"* button, the ZAP UI will be launched.

Automated scan

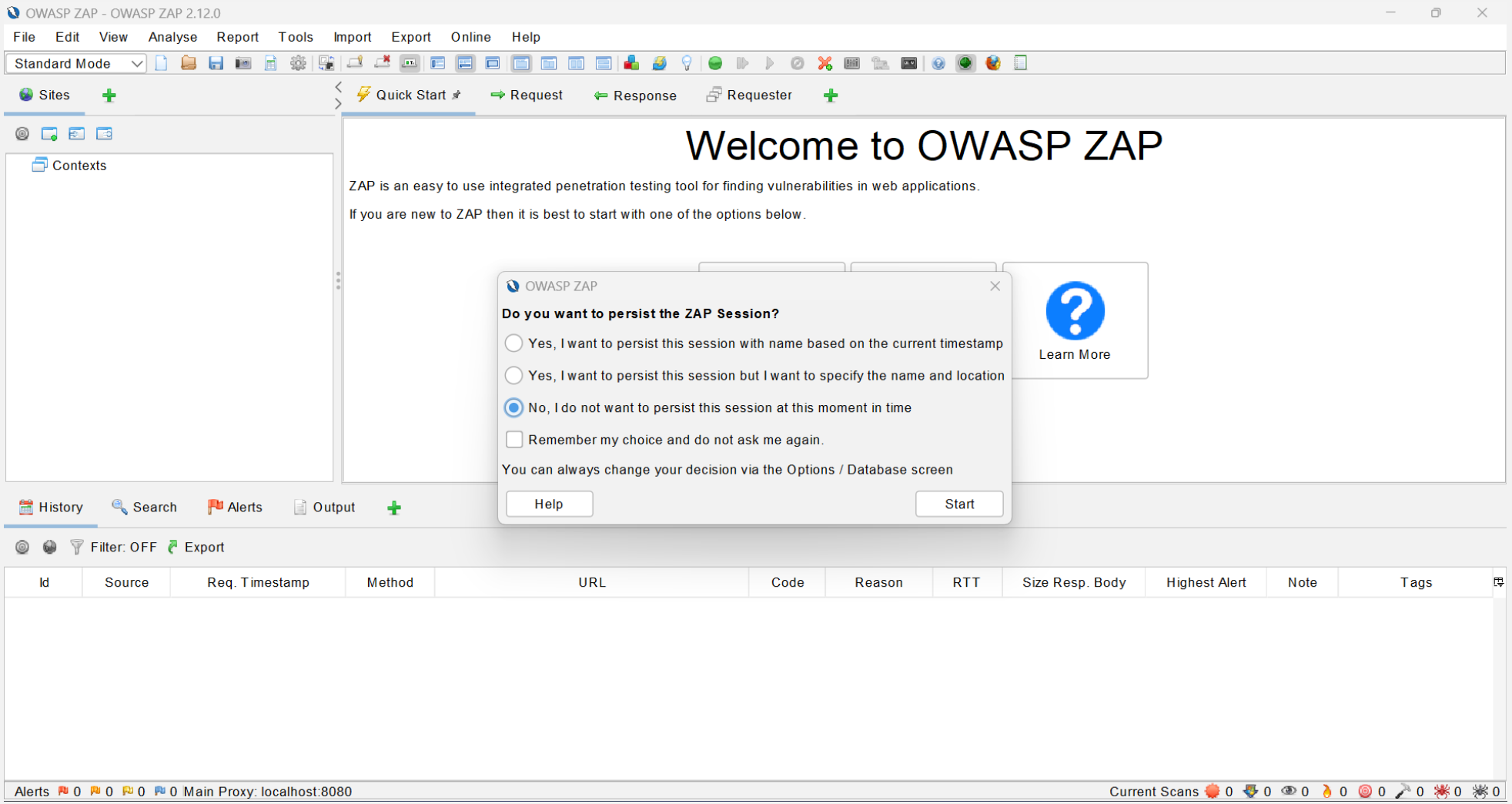
This option allows you to launch an automated scan against an application just by entering the URL. If you are new to ZAP, it is best to start with Automated Scan mode.

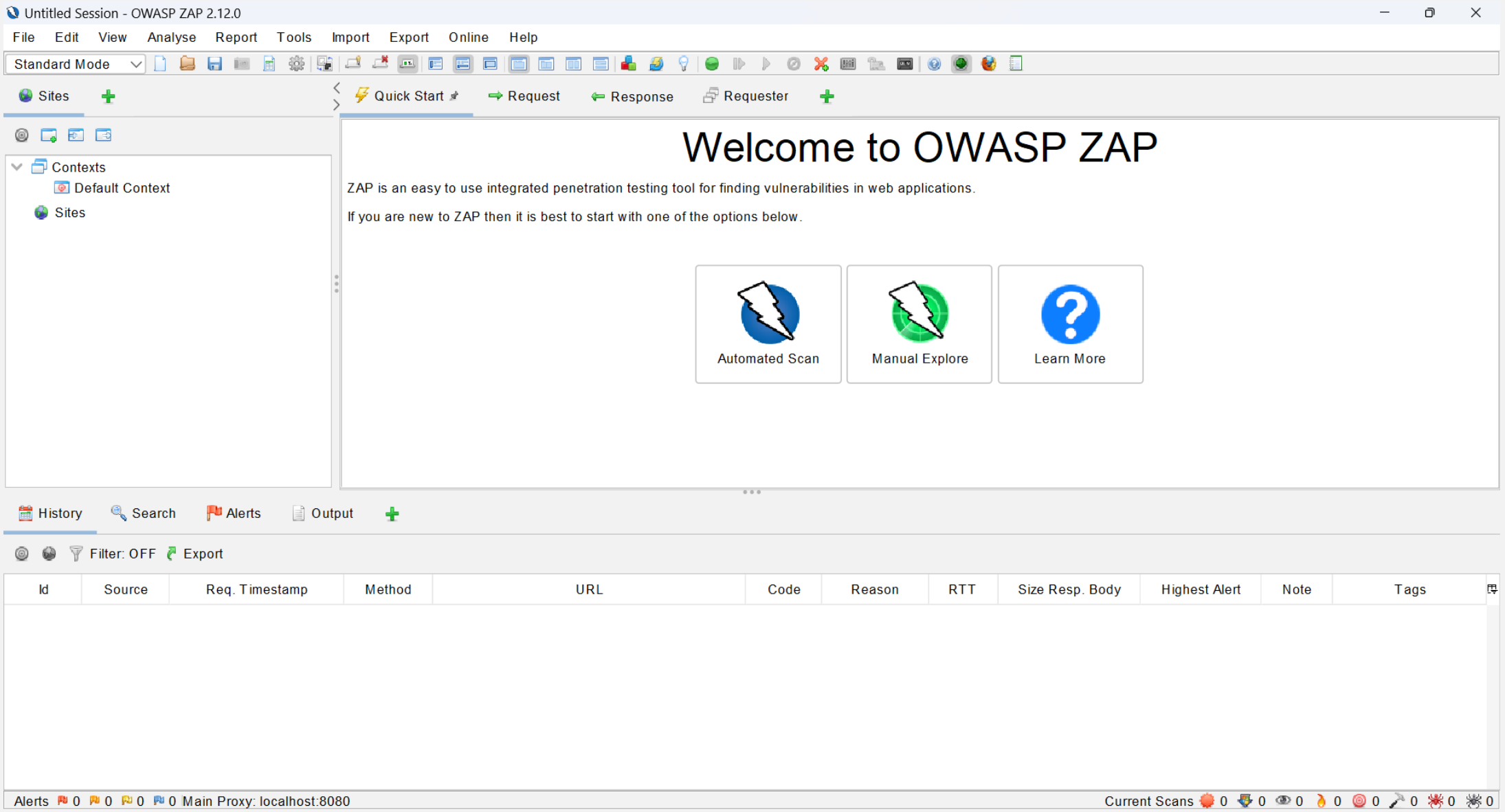
To run a Quick Start Automated Scan:
1. Start Zap and click the large 'Automated Scan' button in the 'Quick Start' tab.
2. Enter the full URL of the web application you want to attack in the 'URL to attack' text box.
3. Click the 'Attack' button.

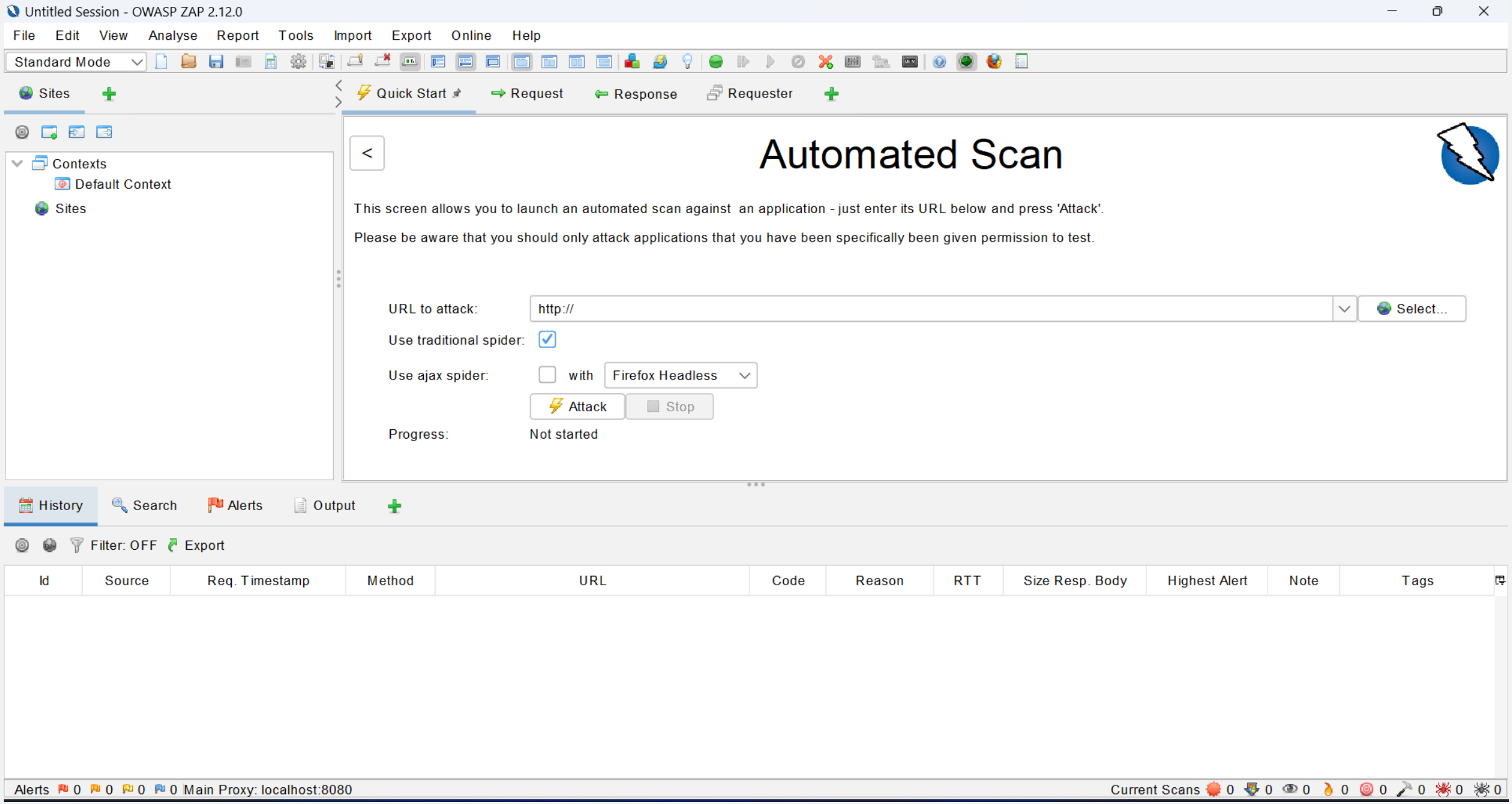
Inspecting the test results

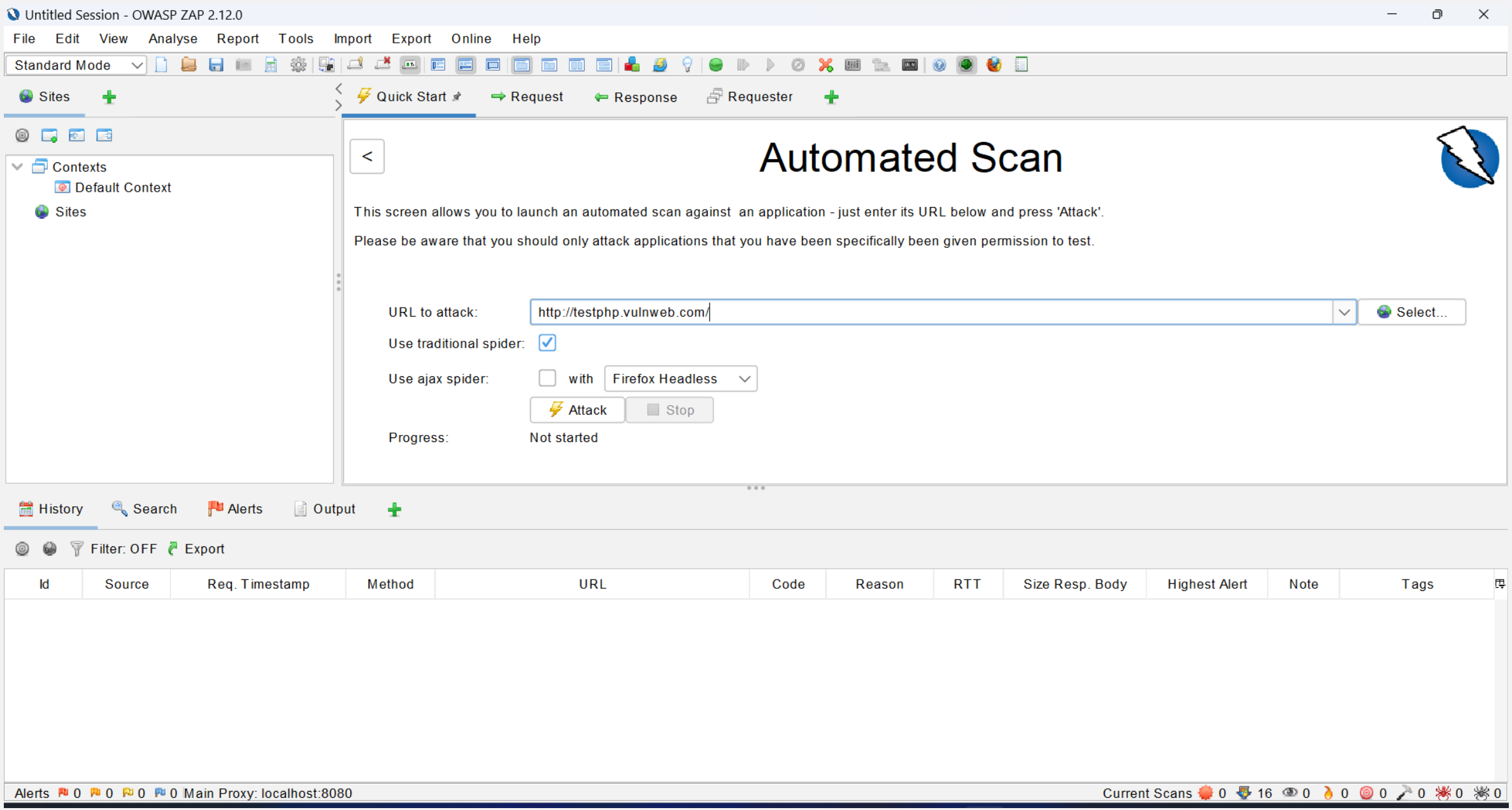
Once the scan is completed, ZAP generates a list of issues that are found during the scan. These issues can be seen on the *Alerts* tab that is located in the bottom pane. All the issues are marked with colour coded flags. You can also generate an HTML scan report through the 'Report' menu option on the top of the screen. You can further check the details and make improvement in that report if needed and also try to make things in sequence.

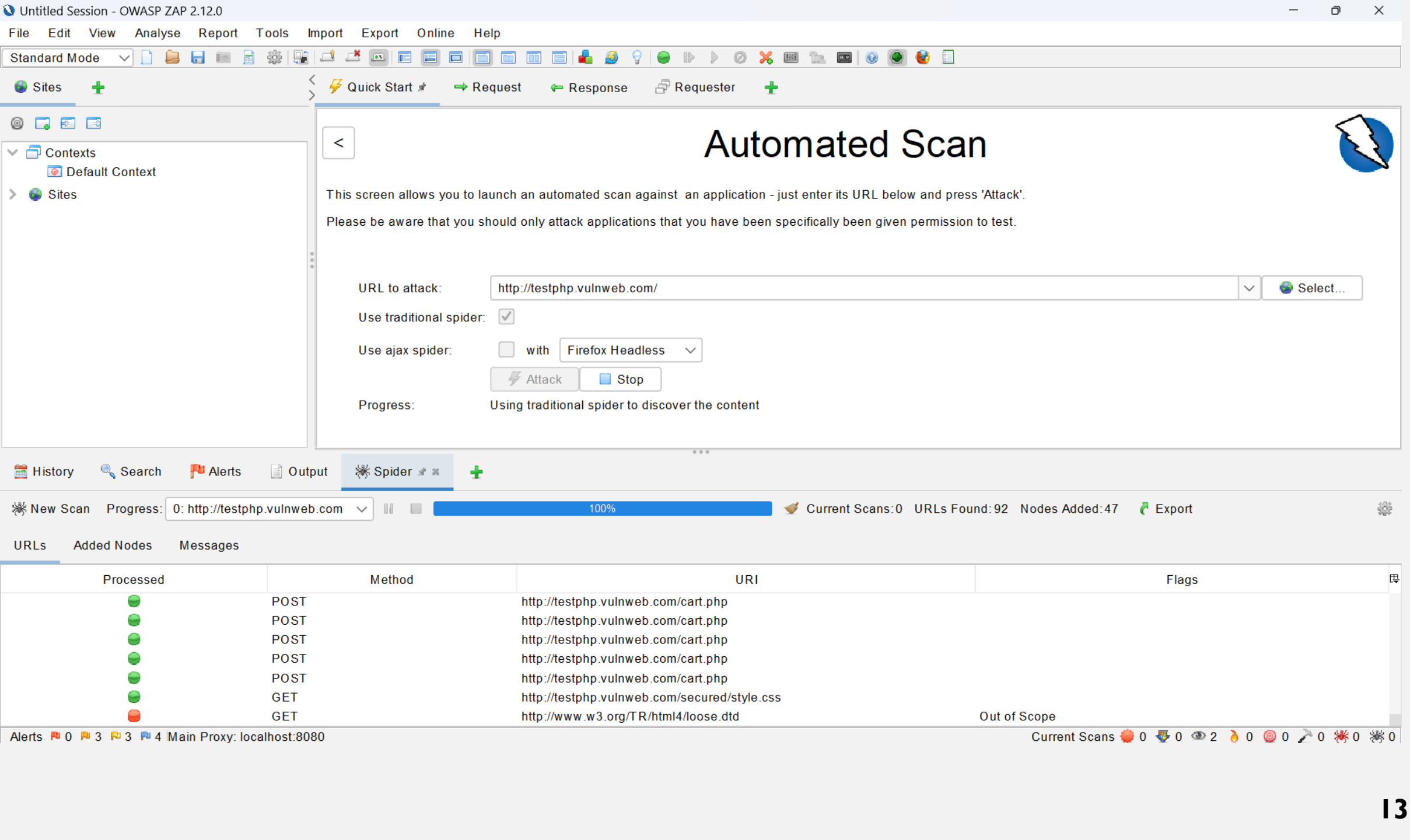
SCREENSHOTS




Untitled Session - OWASP ZAP 2.12.0
File Edit View Analyse Report Tools Import Export Online Help
Standard Mode
Sites
Quick Start
Request
Response
Requester
Contexts
Default Context
Sites
Welcome to OWASP ZAP
ZAP is an easy to use integrated penetration testing tool for finding vulnerabilities in web applications.
If you are new to ZAP then it is best to start with one of the options below.
Automated Scan
Manual Explore
Learn More
History
Search
Alerts
Output
Filter: OFF
Export
Id
Source
Req. Timestamp
Method
URL
Code
Reason
RTT
Size Resp. Body
Highest Alert
Note
Tags
Alerts 0 0 0 0 Main Proxy: localhost:8080
Current Scans 0 0 0 0 0 0 0 0


Untitled Session - OWASP ZAP 2.12.0
File Edit View Analyse Report Tools Import Export Online Help
Standard Mode
Sites
Quick Start
Request
Response
Requester
Contexts
Default Context
Sites
Automated Scan
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://
Select...
Use traditional spider:
Use ajax spider:
with
Firefox Headless
Attack
Stop
Progress:
Not started
History
Search
Alerts
Output
Filter: OFF
Export
Id
Source
Req. Timestamp
Method
URL
Code
Reason
RTT
Size Resp. Body
Highest Alert
Note
Tags
Alerts 0 0 0 0 Main Proxy: localhost:8080
Current Scans 0 0 0 0 0 0 0 0


Untitled Session - OWASP ZAP 2.12.0
File Edit View Analyse Report Tools Import Export Online Help
Standard Mode
Sites
Quick Start
Request
Response
Requester
Contexts
Default Context
Sites
Automated Scan
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://testphp.vulnweb.com/
Select...
Use traditional spider:
Use ajax spider:
with
Firefox Headless
Attack
Stop
Progress:
Not started
History
Search
Alerts
Output
Filter: OFF
Export
Id
Source
Req. Timestamp
Method
URL
Code
Reason
RTT
Size Resp. Body
Highest Alert
Note
Tags
Alerts 0 0 0 0 Main Proxy: localhost:8080
Current Scans 0 16 0 0 0 0 0 0


Untitled Session - OWASP ZAP 2.12.0
File Edit View Analyse Report Tools Import Export Online Help
Standard Mode
Sites
Quick Start
Request
Response
Requester
Contexts
Default Context
Sites
Automated Scan
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://testphp.vulnweb.com/
Select...
Use traditional spider:
Use ajax spider:
with
Firefox Headless
Attack
Stop
Progress:
Using traditional spider to discover the content
History
Search
Alerts
Output
Spider
New Scan
Progress:
0: http://testphp.vulnweb.com
100%
Current Scans: 0
URLs Found: 92
Nodes Added: 47
Export
URLs
Added Nodes
Messages
Processed
Method
URI
Flags
POST http://testphp.vulnweb.com/cart.php
POST http://testphp.vulnweb.com/cart.php
POST http://testphp.vulnweb.com/cart.php
POST http://testphp.vulnweb.com/cart.php
POST http://testphp.vulnweb.com/cart.php
GET http://testphp.vulnweb.com/secured/style.css
GET http://www.w3.org/TR/html4/loose.dtd Out of Scope
Alerts 0 3 3 4 Main Proxy: localhost:8080
Current Scans 0 0 2 0 0 0 0 0

Untitled Session - OWASP ZAP 2.12.0

File Edit View Analyse Report Tools Import Export Online Help

Standard Mode

Sites

Quick Start | Request | Response | Requester

Contexts
Default Context
Sites

Automated Scan

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://testphp.vulnweb.com/ Select...

Use traditional spider:

Use ajax spider: with Firefox Headless

Attack Stop

Progress: Actively scanning (attacking) the URLs discovered by the spider(s)

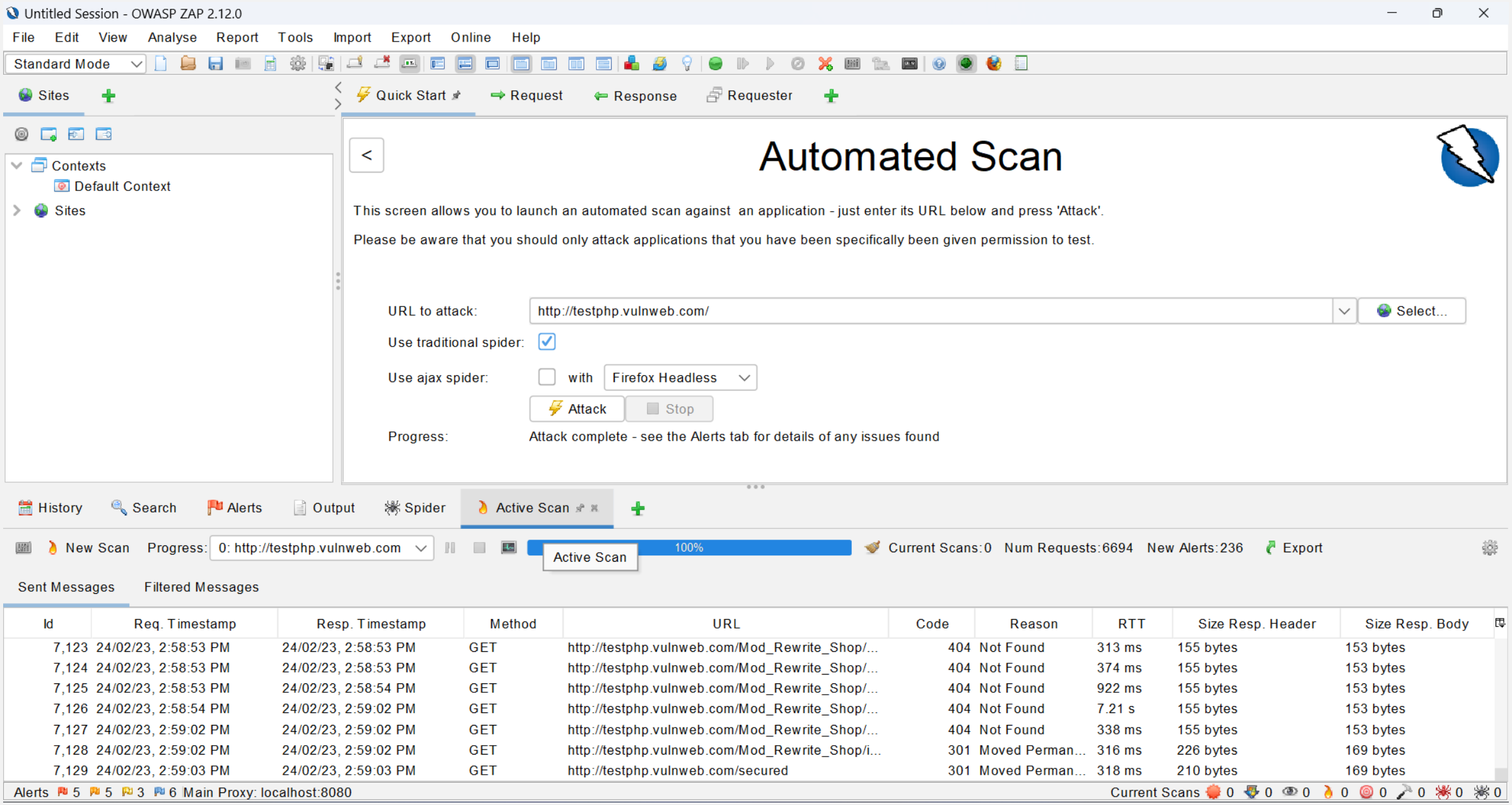
History Search Alerts Output Spider Active Scan

New Scan Progress: 0: http://testphp.vulnweb.com 0% Current Scans:1 Num Requests:24 New Alerts:0 Export

Sent Messages Filtered Messages

Id	Req. Timestamp	Resp. Timestamp	Method	URL	Code	Reason	RTT	Size Resp. Header	Size Resp. Body
234	24/02/23, 2:08:09 PM	24/02/23, 2:08:10 PM	POST	http://testphp.vulnweb.com/guestbook.php	200	OK	639 ms	222 bytes	5,448 bytes
235	24/02/23, 2:08:10 PM	24/02/23, 2:08:10 PM	POST	http://testphp.vulnweb.com/cart.php	200	OK	660 ms	222 bytes	4,903 bytes
236	24/02/23, 2:08:10 PM	24/02/23, 2:08:11 PM	POST	http://testphp.vulnweb.com/cart.php	200	OK	672 ms	222 bytes	4,903 bytes
237	24/02/23, 2:08:10 PM	24/02/23, 2:08:11 PM	POST	http://testphp.vulnweb.com/guestbook.php	200	OK	746 ms	222 bytes	5,393 bytes
238	24/02/23, 2:08:11 PM	24/02/23, 2:08:11 PM	POST	http://testphp.vulnweb.com/cart.php	200	OK	647 ms	222 bytes	4,903 bytes
239	24/02/23, 2:08:11 PM	24/02/23, 2:08:12 PM	POST	http://testphp.vulnweb.com/guestbook.php	200	OK	850 ms	222 bytes	5,391 bytes
240	24/02/23, 2:08:11 PM	24/02/23, 2:08:12 PM	POST	http://testphp.vulnweb.com/cart.php	200	OK	631 ms	222 bytes	4,903 bytes

Alerts 0 3 3 4 Main Proxy: localhost:8080 Current Scans 0 0 2 1 0 0 0 0


Untitled Session - OWASP ZAP 2.12.0
File Edit View Analyse Report Tools Import Export Online Help
Standard Mode
Sites
Quick Start
Request
Response
Requester
Contexts
Default Context
Sites
Automated Scan
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://testphp.vulnweb.com/
Select...
Use traditional spider:
Use ajax spider:
with
Firefox Headless
Attack
Stop
Progress:
Attack complete - see the Alerts tab for details of any issues found
History
Search
Alerts
Output
Spider
Active Scan
New Scan
Progress:
0: http://testphp.vulnweb.com
Active Scan
100%
Current Scans:0
Num Requests:6694
New Alerts:236
Export
Sent Messages
Filtered Messages
Id
Req. Timestamp
Resp. Timestamp
Method
URL
Code
Reason
RTT
Size Resp. Header
Size Resp. Body
7,123 24/02/23, 2:58:53 PM 24/02/23, 2:58:53 PM GET http://testphp.vulnweb.com/Mod_Rewrite_Shop/... 404 Not Found 313 ms 155 bytes 153 bytes
7,124 24/02/23, 2:58:53 PM 24/02/23, 2:58:53 PM GET http://testphp.vulnweb.com/Mod_Rewrite_Shop/... 404 Not Found 374 ms 155 bytes 153 bytes
7,125 24/02/23, 2:58:53 PM 24/02/23, 2:58:54 PM GET http://testphp.vulnweb.com/Mod_Rewrite_Shop/... 404 Not Found 922 ms 155 bytes 153 bytes
7,126 24/02/23, 2:58:54 PM 24/02/23, 2:59:02 PM GET http://testphp.vulnweb.com/Mod_Rewrite_Shop/... 404 Not Found 7.21 s 155 bytes 153 bytes
7,127 24/02/23, 2:59:02 PM 24/02/23, 2:59:02 PM GET http://testphp.vulnweb.com/Mod_Rewrite_Shop/... 404 Not Found 338 ms 155 bytes 153 bytes
7,128 24/02/23, 2:59:02 PM 24/02/23, 2:59:02 PM GET http://testphp.vulnweb.com/Mod_Rewrite_Shop/i... 301 Moved Perman... 316 ms 226 bytes 169 bytes
7,129 24/02/23, 2:59:03 PM 24/02/23, 2:59:03 PM GET http://testphp.vulnweb.com/secured 301 Moved Perman... 318 ms 210 bytes 169 bytes
Alerts 5 5 3 6 Main Proxy: localhost:8080
Current Scans 0 0 0 0 0 0 0 0

Summary

ZAP is a free open source platform-agnostic security testing tool that scans through your web application to identity any security vulnerabilities as possible. It is a great tool for experienced pen testers, as well as beginners. ZAP spiders the web application under test and scan for any known vulnerabilities.

For beginners, it is easy to start with Automated Scan that will crawl the given URL with spider and passively scan each page it finds. You can do a more in-depth scanning by exploring the web application manually.

ZAP generates the scan report in the form of Alerts that are marked with colour coded flags. You can even download HTML reports from the "Report" menu option.

ZAP Scanning Report

Generated with ZAP on Fri 24 Feb 2023, at 15:14:00

Contents

About this report

Report parameters

Contexts

No contexts were selected, so all contexts were included by default.

Sites

The following sites were included:

- http://testphp.vulnweb.com

(If no sites were selected, all sites were included by default.)

An included site must also be within one of the included contexts for its data to be included in the report.

Risk levels

Included: High, Medium, Low, Informational

Excluded: None

Confidence levels

Included: User Confirmed, High, Medium, Low

Excluded: User Confirmed, High, Medium, Low, False Positive

Summaries

Alert counts by risk and confidence

This table shows the number of alerts for each level of risk and confidence included in the report.

(The percentages in brackets represent the count as a percentage of the total number of alerts included in the report, rounded to one decimal place.)

	Confidence				
	User Confirmed	**High**	**Medium**	**Low**	**Total**
High	0 (0.0%)	0 (0.0%)	4 (21.1%)	1 (5.3%)	5 (26.3%)

		Confidence				
		User Confirmed	High	Medium	Low	Total
Risk	Medium	0 (0.0%)	1 (5.3%)	3 (15.8%)	1 (5.3%)	5 (26.3%)
	Low	0 (0.0%)	1 (5.3%)	2 (10.5%)	0 (0.0%)	3 (15.8%)
	Informational	0 (0.0%)	1 (5.3%)	2 (10.5%)	3 (15.8%)	6 (31.6%)
	Total	0 (0.0%)	3 (15.8%)	11 (57.9%)	5 (26.3%)	19 (100%)

Alert counts by site and risk

This table shows, for each site for which one or more alerts were raised, the number of alerts raised at each risk level.

Alerts with a confidence level of "False Positive" have been excluded from these counts.

(The numbers in brackets are the number of alerts raised for the site at or above that risk level.)

		Risk			
		High (= High)	Medium (>= Medium)	Low (>= Low)	Informational (>= Informational)
Site	http://testphp.vulnweb.com	5 (5)	5 (10)	3 (13)	6 (19)

Alert counts by alert type

This table shows the number of alerts of each alert type, together with the alert type's risk level.

(The percentages in brackets represent each count as a percentage, rounded to one decimal place, of the total number of alerts included in this report.)

Alert type	Risk	Count
Cross Site Scripting (Reflected)	High	14 (73.7%)
Path Traversal	High	1 (5.3%)
SQL Injection	High	8 (42.1%)
SQL Injection - MySQL	High	6 (31.6%)
SQL Injection - SQLite	High	1 (5.3%)
.htaccess Information Leak	Medium	7 (36.8%)
Absence of Anti-CSRF Tokens	Medium	41 (215.8%)
Content Security Policy (CSP) Header Not Set	Medium	49 (257.9%)
Missing Anti-clickjacking Header	Medium	45 (236.8%)
XSLT Injection	Medium	2 (10.5%)
Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)	Low	63 (331.6%)
Server Leaks Version Information via "Server" HTTP Response Header Field	Low	75 (394.7%)
X-Content-Type-Options Header Missing	Low	69 (363.2%)
Charset Mismatch (Header Versus Meta Content-Type Charset)	Informational	32 (168.4%)
GET for POST	Informational	1 (5.3%)
Information Disclosure - Suspicious Comments	Informational	1 (5.3%)
Total		19

Alert type	Risk	Count
Modern Web Application	Informational	9 (47.4%)
User Agent Fuzzer	Informational	196 (1,031.6%)
User Controllable HTML Element Attribute (Potential XSS)	Informational	3 (15.8%)
Total		19

Alerts

1. **Risk=High, Confidence=Medium (4)**
    1. **http://testphp.vulnweb.com (4)**
        1. Cross Site Scripting (Reflected) **(1)**
            1. ▶ POST http://testphp.vulnweb.com/guestbook.php
        2. SQL Injection **(1)**
            1. ▶ POST http://testphp.vulnweb.com/search.php?test=query%27+AND+%271%27%3D%271%27+--+
        3. SQL Injection - MySQL **(1)**
            1. ▶ POST http://testphp.vulnweb.com/secured/newuser.php
        4. SQL Injection - SQLite **(1)**
            1. ▶ POST http://testphp.vulnweb.com/secured/newuser.php
2. **Risk=High, Confidence=Low (1)**
    1. **http://testphp.vulnweb.com (1)**
        1. Path Traversal **(1)**
            1. ▶ POST http://testphp.vulnweb.com/guestbook.php
3. **Risk=Medium, Confidence=High (1)**
    1. **http://testphp.vulnweb.com (1)**
        1. Content Security Policy (CSP) Header Not Set **(1)**
            1. ▶ GET http://testphp.vulnweb.com/
4. **Risk=Medium, Confidence=Medium (3)**
    1. **http://testphp.vulnweb.com (3)**
        1. .htaccess Information Leak **(1)**
            1. ▶ GET http://testphp.vulnweb.com/Mod_Rewrite_Shop/.htaccess
        2. Missing Anti-clickjacking Header **(1)**
            1. ▶ GET http://testphp.vulnweb.com/
        3. XSLT Injection **(1)**
            1. ▶ GET http://testphp.vulnweb.com/showimage.php?file=%3Cxsl%3Avalue-of+select%3D%22document%28%27http%3A%2F%2Ftestphp.vulnweb.com%3A22%27%29%22%2F%3E
5. **Risk=Medium, Confidence=Low (1)**
    1. **http://testphp.vulnweb.com (1)**

1. Absence of Anti-CSRF Tokens (1)
    1. ▶ GET http://testphp.vulnweb.com/

6. **Risk=Low, Confidence=High (1)**
    1. **http://testphp.vulnweb.com (1)**
        1. Server Leaks Version Information via "Server" HTTP Response Header Field (1)
            1. ▶ GET http://testphp.vulnweb.com/

7. **Risk=Low, Confidence=Medium (2)**
    1. **http://testphp.vulnweb.com (2)**
        1. Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s) (1)
            1. ▶ GET http://testphp.vulnweb.com/
        2. X-Content-Type-Options Header Missing (1)
            1. ▶ GET http://testphp.vulnweb.com/

8. **Risk=Informational, Confidence=High (1)**
    1. **http://testphp.vulnweb.com (1)**
        1. GET for POST (1)
            1. ▶ GET http://testphp.vulnweb.com/cart.php

9. **Risk=Informational, Confidence=Medium (2)**
    1. **http://testphp.vulnweb.com (2)**
        1. Modern Web Application (1)
            1. ▶ GET http://testphp.vulnweb.com/AJAX/index.php
        2. User Agent Fuzzer (1)
            1. ▶ POST http://testphp.vulnweb.com/guestbook.php

10. **Risk=Informational, Confidence=Low (3)**
    1. **http://testphp.vulnweb.com (3)**
        1. Charset Mismatch (Header Versus Meta Content-Type Charset) (1)
            1. ▶ GET http://testphp.vulnweb.com/
        2. Information Disclosure - Suspicious Comments (1)
            1. ▶ GET http://testphp.vulnweb.com/AJAX/index.php
        3. User Controllable HTML Element Attribute (Potential XSS) (1)
            1. ▶ POST http://testphp.vulnweb.com/search.php?test=query

Appendix

Alert types

This section contains additional information on the types of alerts in the report.

1. **Cross Site Scripting (Reflected)**

Source raised by an active scanner (Cross Site Scripting (Reflected))

CWE ID 79
WASC ID 8
Reference
1. http://projects.webappsec.org/Cross-Site-Scripting
2. http://cwe.mitre.org/data/definitions/79.html

2. **Path Traversal**

Source raised by an active scanner (Path Traversal)
CWE ID 22
WASC ID 33
Reference
1. http://projects.webappsec.org/Path-Traversal
2. http://cwe.mitre.org/data/definitions/22.html

3. **SQL Injection**

Source raised by an active scanner (SQL Injection)
CWE ID 89
WASC ID 19
Reference
1. https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html

4. **SQL Injection - MySQL**

Source raised by an active scanner (SQL Injection - MySQL)
CWE ID 89
WASC ID 19
Reference
1. https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html

5. **SQL Injection - SQLite**

Source raised by an active scanner (SQL Injection - SQLite)
CWE ID 89
WASC ID 19
Reference
1. https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html

6. **.htaccess Information Leak**

Source raised by an active scanner (.htaccess Information Leak)
CWE ID 94
WASC ID 14
Reference
1. http://www.htaccess-guide.com/

7. **Absence of Anti-CSRF Tokens**

Source raised by a passive scanner (Absence of Anti-CSRF Tokens)
CWE ID 352
WASC ID 9
Reference
1. http://projects.webappsec.org/Cross-Site-Request-Forgery
2. http://cwe.mitre.org/data/definitions/352.html

8. **Content Security Policy (CSP) Header Not Set**

Source raised by a passive scanner (Content Security Policy (CSP) Header Not Set)
CWE ID 693
WASC ID 15
Reference
1. https://developer.mozilla.org/en-US/docs/Web/Security/CSP/Introducing_Content_Security_Policy
2. https://cheatsheetseries.owasp.org/cheatsheets/Content_Security_Policy_Cheat_Sheet.html
3. http://www.w3.org/TR/CSP/
4. http://w3c.github.io/webappsec/specs/content-security-policy/csp-specification.dev.html
5. http://www.html5rocks.com/en/tutorials/security/content-security-policy/
6. http://caniuse.com/#feat=contentsecuritypolicy
7. http://content-security-policy.com/

9. **Missing Anti-clickjacking Header**

Source	raised by a passive scanner (Anti-clickjacking Header)
CWE ID	1021
WASC ID	15
Reference	1. https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options

10. **XSLT Injection**

Source	raised by an active scanner (XSLT Injection)
CWE ID	91
WASC ID	23
Reference	1. https://www.contextis.com/blog/xslt-server-side-injection-attacks

11. **Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)**

Source	raised by a passive scanner (Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s))
CWE ID	200
WASC ID	13
Reference	1. http://blogs.msdn.com/b/varunm/archive/2013/04/23/remove-unwanted-http-response-headers.aspx 2. http://www.troyhunt.com/2012/02/shhh-dont-let-your-response-headers.html

12. **Server Leaks Version Information via "Server" HTTP Response Header Field**

Source	raised by a passive scanner (HTTP Server Response Header)
CWE ID	200
WASC ID	13
Reference	1. http://httpd.apache.org/docs/current/mod/core.html#servertokens 2. http://msdn.microsoft.com/en-us/library/ff648552.aspx#ht_urlscan_007 3. http://blogs.msdn.com/b/varunm/archive/2013/04/23/remove-unwanted-http-response-headers.aspx 4. http://www.troyhunt.com/2012/02/shhh-dont-let-your-response-headers.html

13. **X-Content-Type-Options Header Missing**

Source	raised by a passive scanner (X-Content-Type-Options Header Missing)
CWE ID	693
WASC ID	15
Reference	1. http://msdn.microsoft.com/en-us/library/ie/gg622941%28v=vs.85%29.aspx 2. https://owasp.org/www-community/Security_Headers

14. **Charset Mismatch (Header Versus Meta Content-Type Charset)**

Source	raised by a passive scanner (Charset Mismatch)
CWE ID	436
WASC ID	15
Reference	1. http://code.google.com/p/browsersec/wiki/Part2#Character_set_handling_and_detection

15. **GET for POST**

Source	raised by an active scanner (GET for POST)
CWE ID	16
WASC ID	20

16. **Information Disclosure - Suspicious Comments**

Source	raised by a passive scanner (Information Disclosure - Suspicious Comments)
CWE ID	200
WASC ID	13

17. **Modern Web Application**

Source raised by a passive scanner (Modern Web Application)

18. **User Agent Fuzzer**

Source raised by an active scanner (User Agent Fuzzer)
Reference 1. https://owasp.org/wstg

19. **User Controllable HTML Element Attribute (Potential XSS)**

Source raised by a passive scanner (User Controllable HTML Element Attribute (Potential XSS))
CWE ID 20
WASC ID 20
Reference 1. http://websecuritytool.codeplex.com/wikipage?title=Checks#user-controlled-html-attribute