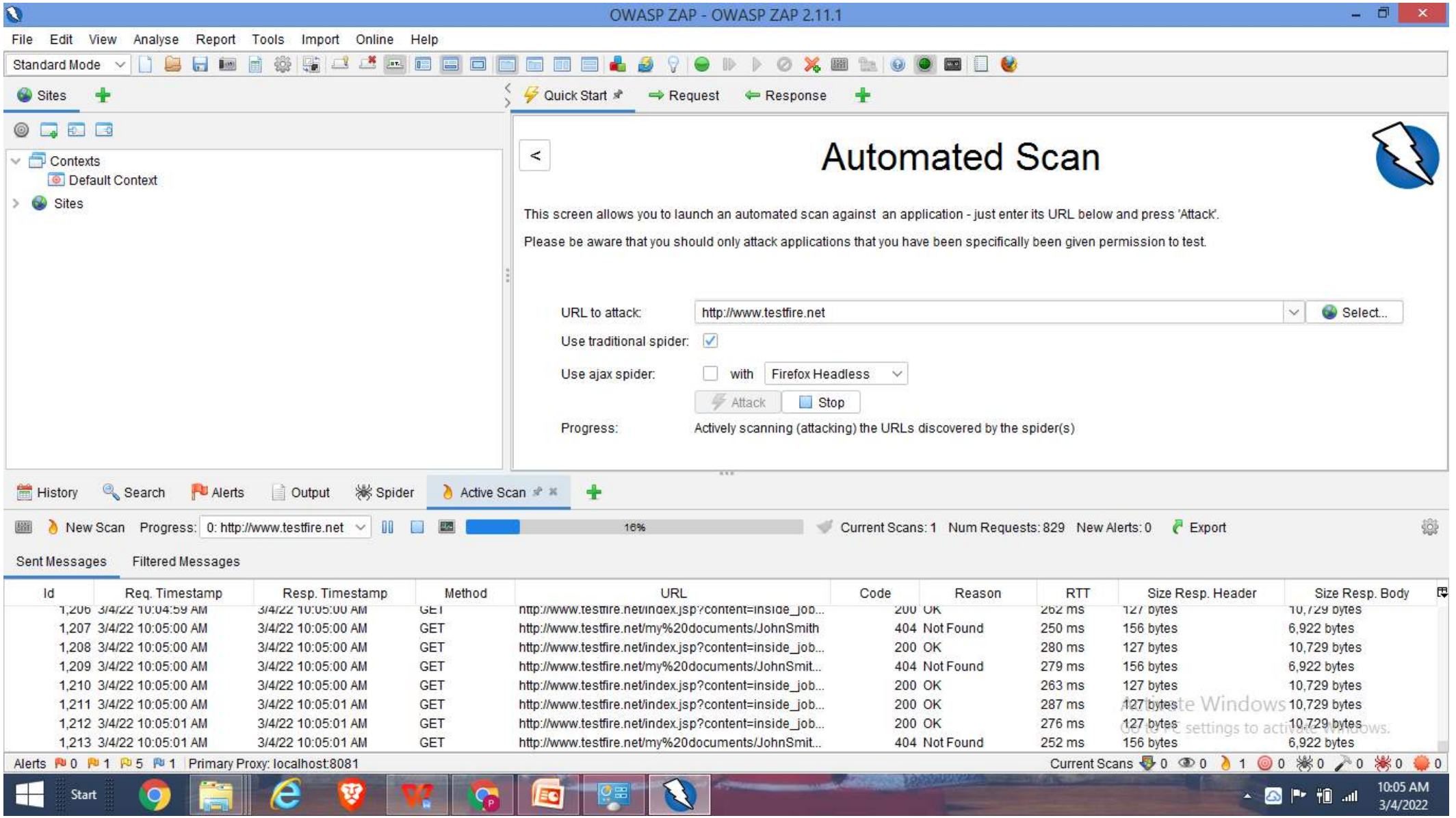
TASK2

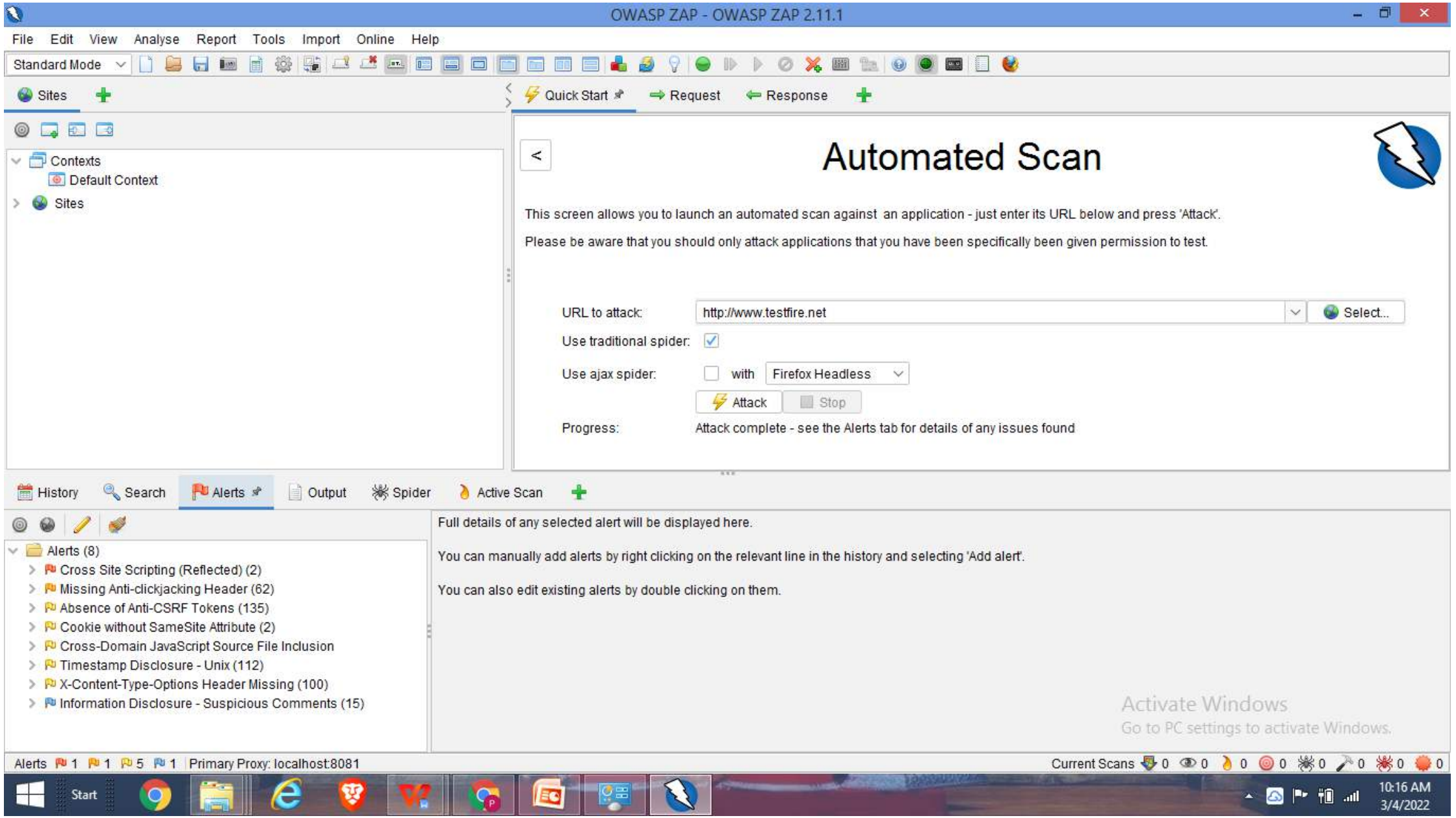
AIM:- FINDING VULNERABILITIES OF A GIVEN WEBSITE.

STEPS:-

STEP 1: OPEN VIRTUAL BOX AND LOGIN TO LINUX
STEP 2: OPEN A VIRTUAL BANK WEBSITE , for example ., www.testfire.net
STEP 3: OPEN NETSPARKER , IF IT IS NOT AVAILABLE YOU CAN DO THIS USING OWASP ZAP APP FOR VULNERABILITY FINDING
STEP 4: OPEN OWASP ZAP THEN GIVE THE DETAILS OF THE WEBSITE EITHER THE **URL OR IP ADDRESS**
STEP 5: THEN START ATTACKING
STEP 6: U CAN SEE THE PROCESS , AFTER THE COMPLETION CLICK ON REPORT THEN SAVE IT AS .html FILE NOW THE REQUIRED REPORT IS SAVED
STEP 7 : IN THE REPORT ,ALL THE REQUIRED WILL BE OBTAINED like., **vulnerabilities ,url's and the solution.**

OUTPUT:-


OWASP ZAP - OWASP ZAP 2.11.1
File Edit View Analyse Report Tools Import Online Help
Standard Mode
Sites
Quick Start
Request
Response
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://www.testfire.net
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://www.testfire.net
16%
Current Scans: 1
Num Requests: 829
New Alerts: 0
Export
Sent Messages
Filtered Messages
Id | Req. Timestamp | Resp. Timestamp | Method | URL | Code | Reason | RTT | Size Resp. Header | Size Resp. Body
1,207 3/4/22 10:05:00 AM 3/4/22 10:05:00 AM GET http://www.testfire.net/my%20documents/JohnSmith 404 Not Found 250 ms 156 bytes 6,922 bytes
1,208 3/4/22 10:05:00 AM 3/4/22 10:05:00 AM GET http://www.testfire.net/index.jsp?content=inside_job... 200 OK 280 ms 127 bytes 10,729 bytes
1,209 3/4/22 10:05:00 AM 3/4/22 10:05:00 AM GET http://www.testfire.net/my%20documents/JohnSmit... 404 Not Found 279 ms 156 bytes 6,922 bytes
1,210 3/4/22 10:05:00 AM 3/4/22 10:05:00 AM GET http://www.testfire.net/index.jsp?content=inside_job... 200 OK 263 ms 127 bytes 10,729 bytes
1,211 3/4/22 10:05:00 AM 3/4/22 10:05:01 AM GET http://www.testfire.net/index.jsp?content=inside_job... 200 OK 287 ms 127 bytes 10,729 bytes
1,212 3/4/22 10:05:01 AM 3/4/22 10:05:01 AM GET http://www.testfire.net/index.jsp?content=inside_job... 200 OK 276 ms 127 bytes 10,729 bytes
1,213 3/4/22 10:05:01 AM 3/4/22 10:05:01 AM GET http://www.testfire.net/my%20documents/JohnSmit... 404 Not Found 252 ms 156 bytes 6,922 bytes
Alerts 0 1 5 1 Primary Proxy: localhost:8081
Current Scans 0 0 1 0 0 0 0 0
Start
10:05 AM
3/4/2022


OWASP ZAP - OWASP ZAP 2.11.1
File Edit View Analyse Report Tools Import Online Help
Standard Mode
Sites
Quick Start
Request
Response
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://www.testfire.net
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
Alerts (8)
Cross Site Scripting (Reflected) (2)
Missing Anti-clickjacking Header (62)
Absence of Anti-CSRF Tokens (135)
Cookie without SameSite Attribute (2)
Cross-Domain JavaScript Source File Inclusion
Timestamp Disclosure - Unix (112)
X-Content-Type-Options Header Missing (100)
Information Disclosure - Suspicious Comments (15)
Full details of any selected alert will be displayed here.
You can manually add alerts by right clicking on the relevant line in the history and selecting 'Add alert'.
You can also edit existing alerts by double clicking on them.
Activate Windows
Go to PC settings to activate Windows.
Alerts 1 1 5 1
Primary Proxy: localhost:8081
Current Scans 0 0 0 0 0 0 0 0
Start
10:16 AM
3/4/2022

Edit View Analyse Report Tools Import Online Help

Quick Start Request Response

Header: Text Body: Text

HTTP/1.1 200 OK

Contexts
Default Context

<scrIpt>alert(1);</scRipt>

story Search Alerts Output Spider Active Scan

Alerts (8)
- Cross Site Scripting (Reflected) (2)
  - GET: http://www.testfire.net/search.jsp?query=%3C%2Fp%
  - POST: http://www.testfire.net/sendFeedback
- Missing Anti-clickjacking Header (62)
- Absence of Anti-CSRF Tokens (135)
- Cookie without SameSite Attribute (2)
- Cross-Domain JavaScript Source File Inclusion
- Timestamp Disclosure - Unix (112)
- X-Content-Type-Options Header Missing (100)
- Information Disclosure - Suspicious Comments (15)

Cross Site Scripting (Reflected)

URL: http://www.testfire.net/search.jsp?query=%3C%2Fp%3E%3Cscrlpt%3Ealert%281%29%3B%3C%2FscRipt%3E%3Cp%3E
Risk: High
Confidence: Medium
Parameter: query
Attack: <scrlpt>alert(1);</scRipt>
Evidence: <scrlpt>alert(1);</scRipt>
CWE ID: 79
WASC ID: 8
Source: Active (40012 - Cross Site Scripting (Reflected))

Description:

Cross-site Scripting (XSS) is an attack technique that involves echoing attacker-supplied code into a user's browser instance. A browser instance can be a standard we browser client, or a browser object embedded in a software product such as the browser within WinAmp, an RSS reader, or an email client. The code itself is usually written in HTML/JavaScript, but may also extend to VBScript, ActiveX, Java, Flash, or any other browser-supported technology.

Other Info:

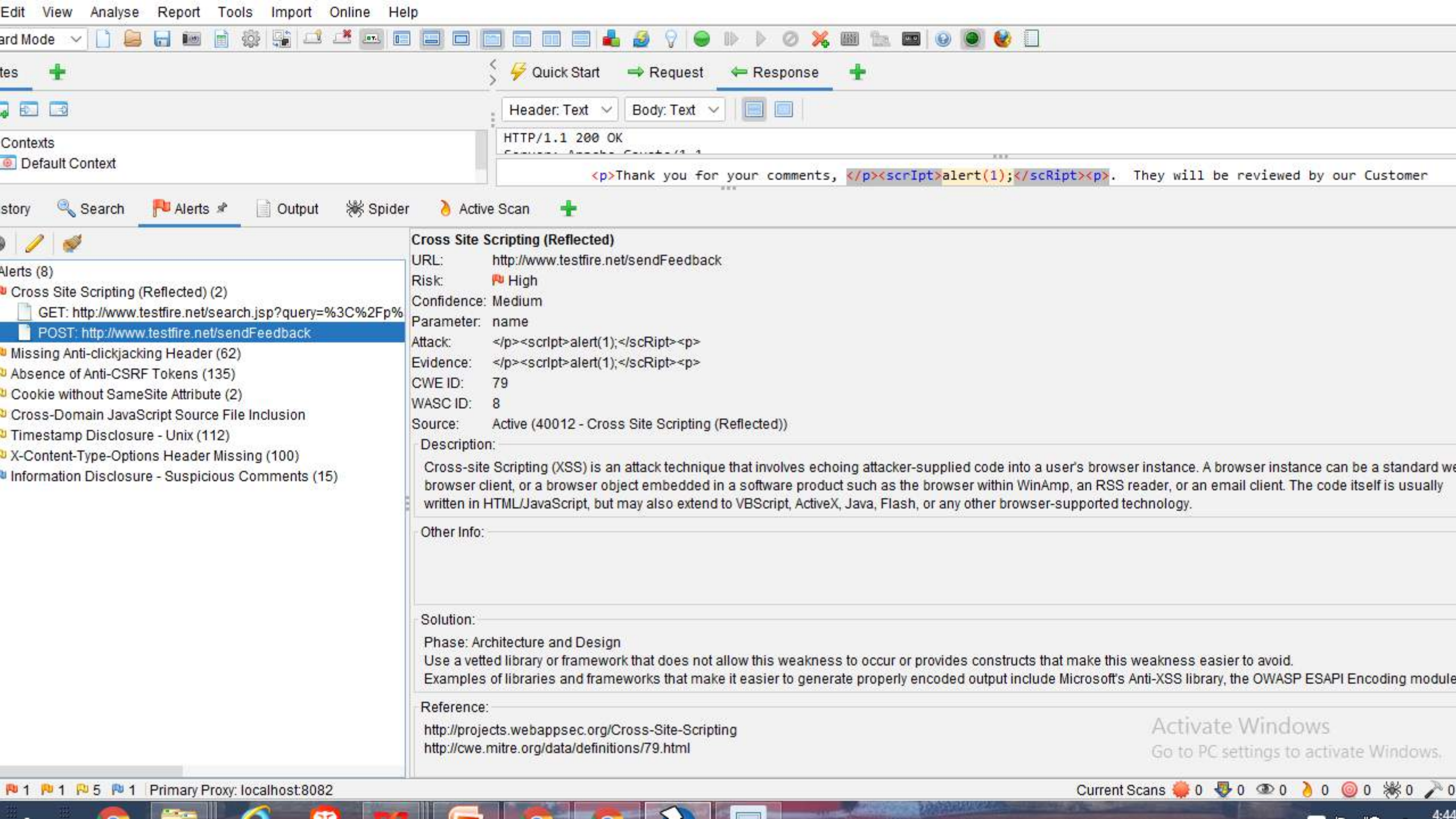
Solution:

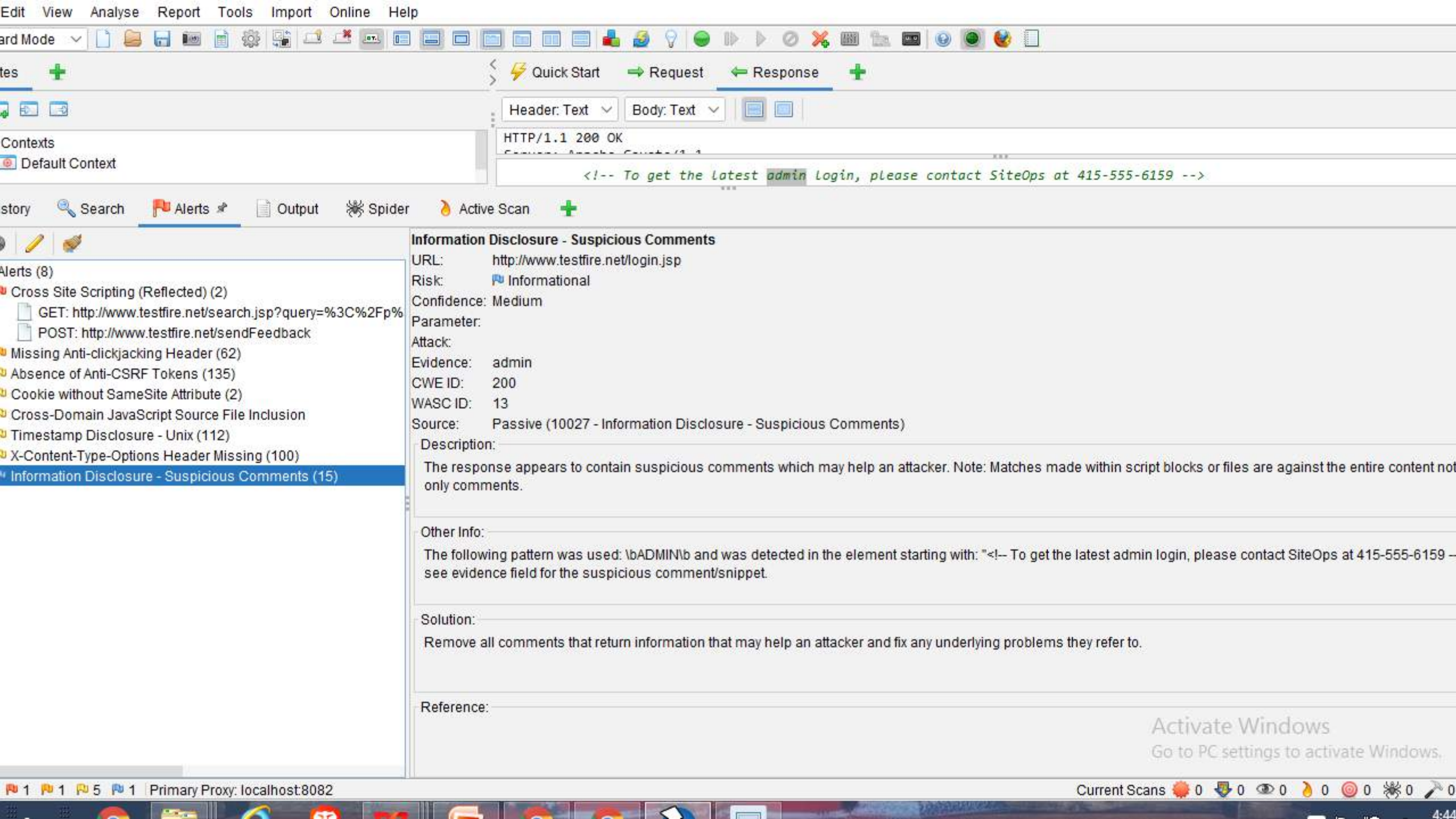
Phase: Architecture and Design
Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid.
Examples of libraries and frameworks that make it easier to generate properly encoded output include Microsoft's Anti-XSS library, the OWASP ESAPI Encoding module

Reference:

http://projects.webappsec.org/Cross-Site-Scripting
http://cwe.mitre.org/data/definitions/79.html

1 1 5 1 Primary Proxy: localhost:8082 Current Scans 0 0 0 0 0 0 0


Edit View Analyse Report Tools Import Online Help
Quick Start
Request
Response
Header: Text
Body: Text
HTTP/1.1 200 OK
Thank you for your comments, <scrIpt>alert(1);</scRipt> . They will be reviewed by our Customer
Contexts
Default Context
Search
Alerts
Output
Spider
Active Scan
Alerts (8)
Cross Site Scripting (Reflected) (2)
GET: http://www.testfire.net/search.jsp?query=%3C%2Fp%
POST: http://www.testfire.net/sendFeedback
Missing Anti-clickjacking Header (62)
Absence of Anti-CSRF Tokens (135)
Cookie without SameSite Attribute (2)
Cross-Domain JavaScript Source File Inclusion
Timestamp Disclosure - Unix (112)
X-Content-Type-Options Header Missing (100)
Information Disclosure - Suspicious Comments (15)
Cross Site Scripting (Reflected)
URL: http://www.testfire.net/sendFeedback
Risk: High
Confidence: Medium
Parameter: name
Attack: <scrIpt>alert(1);</scRipt>
Evidence: <scrIpt>alert(1);</scRipt>
CWE ID: 79
WASC ID: 8
Source: Active (40012 - Cross Site Scripting (Reflected))
Description:
Cross-site Scripting (XSS) is an attack technique that involves echoing attacker-supplied code into a user's browser instance. A browser instance can be a standard we
browser client, or a browser object embedded in a software product such as the browser within WinAmp, an RSS reader, or an email client. The code itself is usually
written in HTML/JavaScript, but may also extend to VBScript, ActiveX, Java, Flash, or any other browser-supported technology.
Other Info:
Solution:
Phase: Architecture and Design
Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid.
Examples of libraries and frameworks that make it easier to generate properly encoded output include Microsoft's Anti-XSS library, the OWASP ESAPI Encoding module
Reference:
http://projects.webappsec.org/Cross-Site-Scripting
http://cwe.mitre.org/data/definitions/79.html
Activate Windows
Go to PC settings to activate Windows.
Primary Proxy: localhost:8082
Current Scans


Edit View Analyse Report Tools Import Online Help
Quick Start
Request
Response
Header: Text
Body: Text
HTTP/1.1 200 OK
<!-- To get the latest admin login, please contact SiteOps at 415-555-6159 -->
Contexts
Default Context
Search
Alerts
Output
Spider
Active Scan
Alerts (8)
Cross Site Scripting (Reflected) (2)
GET: http://www.testfire.net/search.jsp?query=%3C%2Fp%
POST: http://www.testfire.net/sendFeedback
Missing Anti-clickjacking Header (62)
Absence of Anti-CSRF Tokens (135)
Cookie without SameSite Attribute (2)
Cross-Domain JavaScript Source File Inclusion
Timestamp Disclosure - Unix (112)
X-Content-Type-Options Header Missing (100)
Information Disclosure - Suspicious Comments (15)
Information Disclosure - Suspicious Comments
URL: http://www.testfire.net/login.jsp
Risk: Informational
Confidence: Medium
Parameter:
Attack:
Evidence: admin
CWE ID: 200
WASC ID: 13
Source: Passive (10027 - Information Disclosure - Suspicious Comments)
Description:
The response appears to contain suspicious comments which may help an attacker. Note: Matches made within script blocks or files are against the entire content not only comments.
Other Info:
The following pattern was used: \bADMIN\b and was detected in the element starting with: "<!-- To get the latest admin login, please contact SiteOps at 415-555-6159 -- see evidence field for the suspicious comment/snippet.
Solution:
Remove all comments that return information that may help an attacker and fix any underlying problems they refer to.
Reference:
Activate Windows
Go to PC settings to activate Windows.
Primary Proxy: localhost:8082
Current Scans

Edit View Analyse Report Tools Import Online Help

Quick Start Request Response

Header: Text Body: Text

HTTP/1.1 200 OK

<form id="frmSearch" method="get" action="/search.jsp">

Contexts
Default Context

History Search Alerts Output Spider Active Scan

Alerts (8)
- Cross Site Scripting (Reflected) (2)
  - GET: http://www.testfire.net/search.jsp?query=%3C%2Fp%
  - POST: http://www.testfire.net/sendFeedback
- Missing Anti-clickjacking Header (62)
- Absence of Anti-CSRF Tokens (135)
- Cookie without SameSite Attribute (2)
- Cross-Domain JavaScript Source File Inclusion
- Timestamp Disclosure - Unix (112)
- X-Content-Type-Options Header Missing (100)
- Information Disclosure - Suspicious Comments (15)

Absence of Anti-CSRF Tokens

URL: http://www.testfire.net
Risk: Low
Confidence: Medium
Parameter:
Attack:
Evidence: <form id="frmSearch" method="get" action="/search.jsp">
CWE ID: 352
WASC ID: 9
Source: Passive (10202 - Absence of Anti-CSRF Tokens)

Description:

No Anti-CSRF tokens were found in a HTML submission form.
A cross-site request forgery is an attack that involves forcing a victim to send an HTTP request to a target destination without their knowledge or intent in order to perform an action as the victim. The underlying cause is application functionality using predictable URL/form actions in a repeatable way. The nature of the attack is that CSRF

Other Info:

No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token, _csrf, _csrfSecret, __csrf_magic, CSRF, _token, _csrf_token] was found in the following HTML form: [Form 1: "query"].

Solution:

Phase: Architecture and Design
Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid.
For example, use anti-CSRF packages such as the OWASP CSRFGuard.

Reference:

http://projects.webappsec.org/Cross-Site-Request-Forgery
http://cwe.mitre.org/data/definitions/352.html

1 1 5 1 Primary Proxy: localhost:8082

Current Scans 0 0 0 0 0 0 0

Edit View Analyse Report Tools Import Online Help

Quick Start Request Response

Header: Text Body: Text

HTTP/1.1 200 OK

Contexts
Default Context

History Search Alerts Output Spider Active Scan

Alerts (8)
- Cross Site Scripting (Reflected) (2)
  - GET: http://www.testfire.net/search.jsp?query=%3C%2Fp%
  - POST: http://www.testfire.net/sendFeedback
- Missing Anti-clickjacking Header (62)
- Absence of Anti-CSRF Tokens (135)
- Cookie without SameSite Attribute (2)
- Cross-Domain JavaScript Source File Inclusion
- Timestamp Disclosure - Unix (112)
- X-Content-Type-Options Header Missing (100)
- Information Disclosure - Suspicious Comments (15)

Missing Anti-clickjacking Header

URL: http://www.testfire.net
Risk: Medium
Confidence: Medium
Parameter: X-Frame-Options
Attack:
Evidence:
CWE ID: 1021
WASC ID: 15
Source: Passive (10020 - Anti-clickjacking Header)

Description:

The response does not include either Content-Security-Policy with 'frame-ancestors' directive or X-Frame-Options to protect against 'ClickJacking' attacks.

Other Info:

Solution:

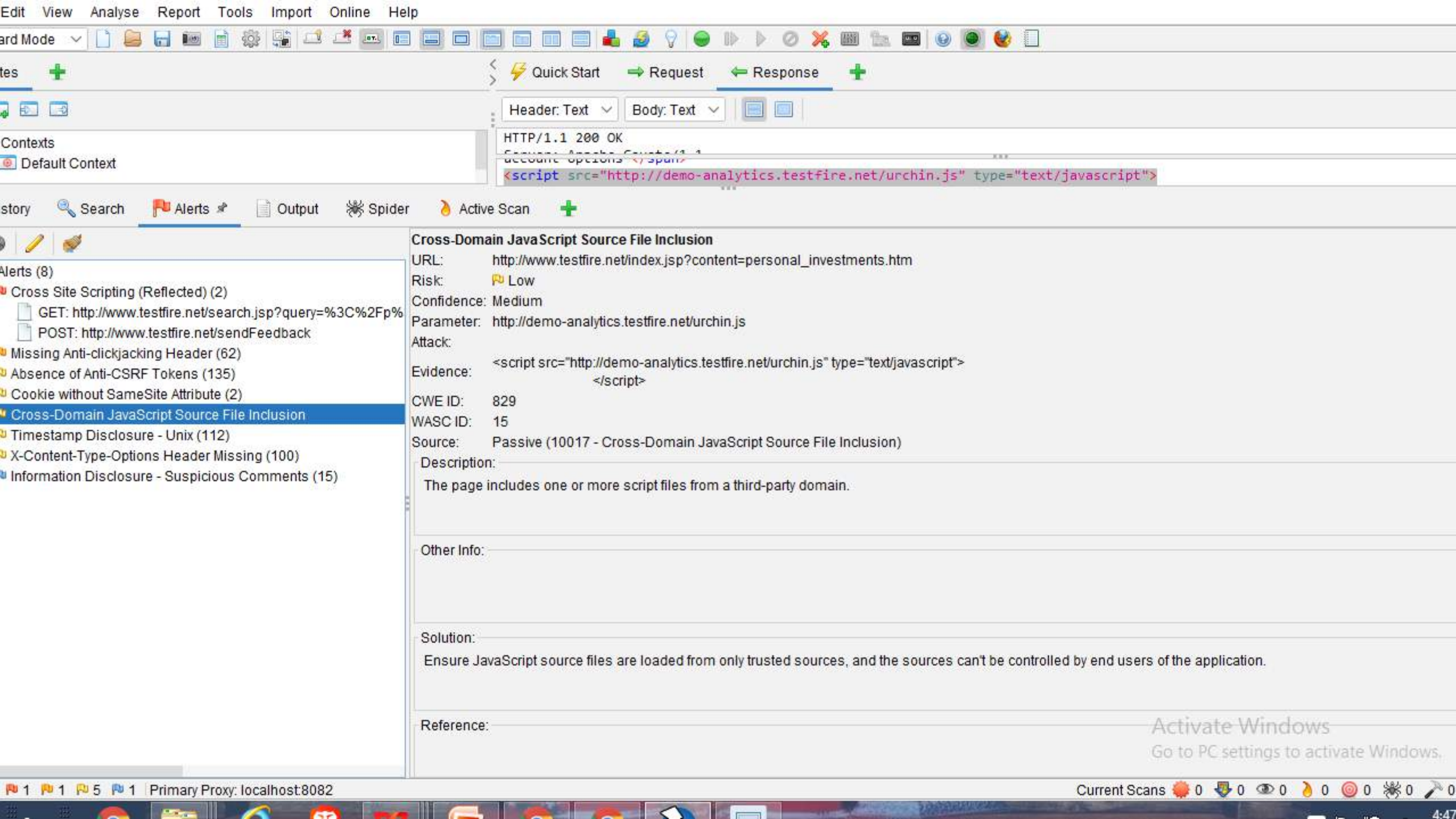
Modern Web browsers support the Content-Security-Policy and X-Frame-Options HTTP headers. Ensure one of them is set on all web pages returned by your site/app.
If you expect the page to be framed only by pages on your server (e.g. it's part of a FRAMESET) then you'll want to use SAMEORIGIN, otherwise if you never expect the pa
to be framed, you should use DENY. Alternatively consider implementing Content Security Policy's "frame-ancestors" directive.

Reference:

https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options

1 1 5 1 Primary Proxy: localhost:8082

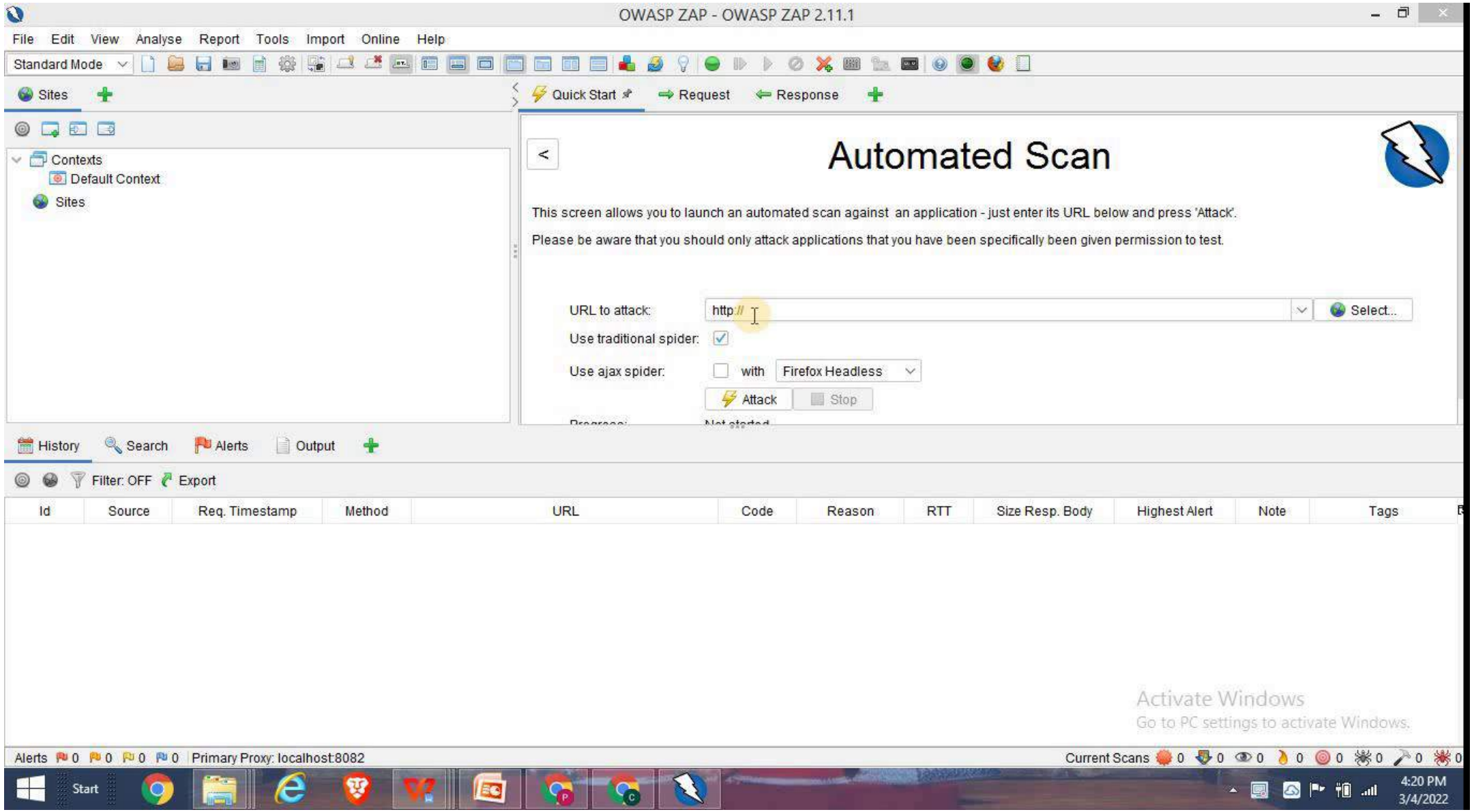
Current Scans 0 0 0 0 0 0 0


Edit View Analyse Report Tools Import Online Help
Quick Start
Request
Response
Header: Text
Body: Text
HTTP/1.1 200 OK
<script src="http://demo-analytics.testfire.net/urchin.js" type="text/javascript">
Contexts
Default Context
Search
Alerts
Output
Spider
Active Scan
Alerts (8)
Cross Site Scripting (Reflected) (2)
GET: http://www.testfire.net/search.jsp?query=%3C%2Fp%
POST: http://www.testfire.net/sendFeedback
Missing Anti-clickjacking Header (62)
Absence of Anti-CSRF Tokens (135)
Cookie without SameSite Attribute (2)
Cross-Domain JavaScript Source File Inclusion
Timestamp Disclosure - Unix (112)
X-Content-Type-Options Header Missing (100)
Information Disclosure - Suspicious Comments (15)
Cross-Domain JavaScript Source File Inclusion
URL: http://www.testfire.net/index.jsp?content=personal_investments.htm
Risk: Low
Confidence: Medium
Parameter: http://demo-analytics.testfire.net/urchin.js
Attack:
Evidence: <script src="http://demo-analytics.testfire.net/urchin.js" type="text/javascript">
</script>
CWE ID: 829
WASC ID: 15
Source: Passive (10017 - Cross-Domain JavaScript Source File Inclusion)
Description:
The page includes one or more script files from a third-party domain.
Other Info:
Solution:
Ensure JavaScript source files are loaded from only trusted sources, and the sources can't be controlled by end users of the application.
Reference:
Activate Windows
Go to PC settings to activate Windows.
Primary Proxy: localhost:8082
Current Scans

REPORT:-

2022-03-04-ZAP-Report-.html

VIDEO:-


OWASP ZAP - OWASP ZAP 2.11.1
File Edit View Analyse Report Tools Import Online Help
Standard Mode
Sites
Quick Start
Request
Response
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
History
Search
Alerts
Output
Filter: OFF
Export
Id
Source
Req. Timestamp
Method
URL
Code
Reason
RTT
Size Resp. Body
Highest Alert
Note
Tags
Activate Windows
Go to PC settings to activate Windows.
Alerts
Primary Proxy: localhost:8082
Current Scans
Start
4:20 PM
3/4/2022