Dynamic Analysis for TuneStore

Dynamic Analysis for Tunestore Report

Shashank Mondrati

ITIS 4221

April, 2022

DYNAMIC ANALYSIS FOR XSS DEMOS REPORT

TABLE OF CONTENTS

Section Page #

1.0 General Information 3

1.1 Purpose 3

1.2 Overview 3

1.0 Cross Site Scripting- DOM Based (High) 4

2.0 Session ID in URL Rewrite (Medium) 5

3.0 X-Frame-Options Header Not Set (Low) 6

4.0 Absence of Anti-CSRF Tokens (Low) 7

5.0 Cookie NoHttpOnlyFlag (Low) 8

6.0 Cookie without SameSite Attribute (Low) 8

7.0 TimeStamp Disclosure (Low) 9

8.0 X-Content Type-Options Header Missing(Low) 10

9.0 Information Disclosure- Suspicious Comments (Medium) 11

10.0 Information Disclosure - URL (Low) 12

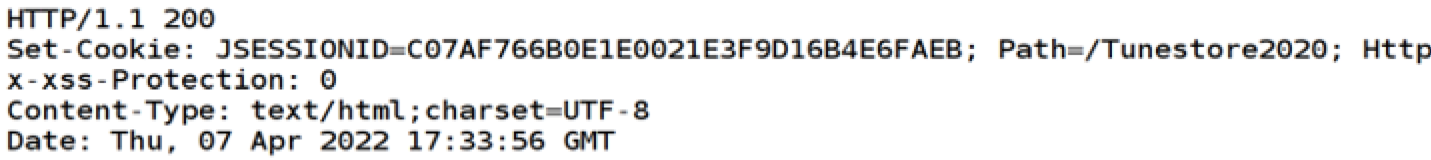
11.0 Loosely Scooped Cookie (LOW) 13

1.0 General Information

1.1 Purpose

The purpose of this Dynamic Analysis report is to examine all vulnerabilities present in the XSS webpage that have a severity level of low or higher, according to ZAP. Each of these vulnerabilities will be categorized as one of the following: true positive or false positive. Additionally, false negatives, which are vulnerabilities that were found in my own pentesting, but not discovered by ZAP, will be discussed.

As you can see below are session cookie and the date for this ZAP Report



```
HTTP/1.1 200
Set-Cookie: JSESSIONID=C07AF766B0E1E0021E3F9D16B4E6FAEB; Path=/Tunestore2020; Http
x-xss-Protection: 0
Content-Type: text/html;charset=UTF-8
Date: Thu, 07 Apr 2022 17:33:56 GMT
```

1.2 Overview

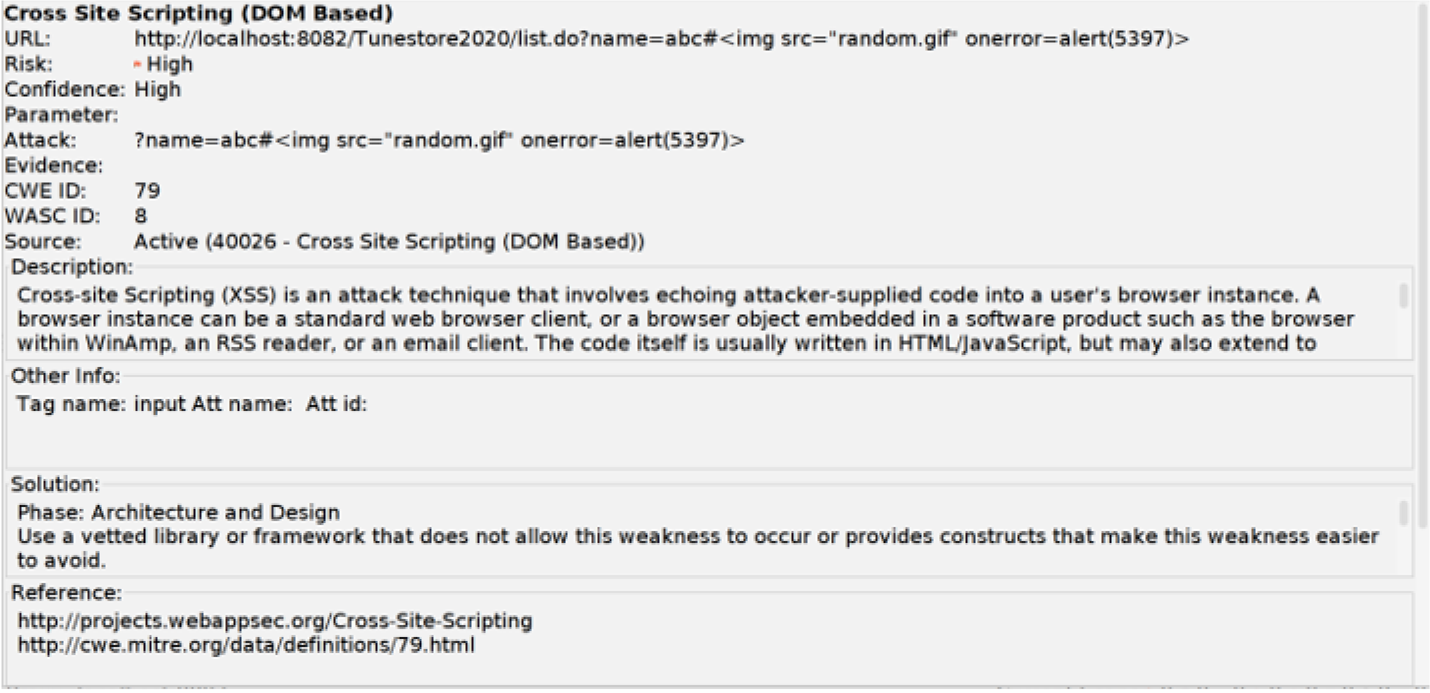
After running a ZAP automated scan on the XSS demos application, a total of 11 vulnerabilities were found to have a security level of low or higher. The following is a list of all vulnerabilities that met this criteria:

- Cross Site Scripting - DOM Based(High)
- Session ID in URL Rewrite (High)
- X-Frame-Options Header Not Set (Medium)
- Absence of Anti-CSRF Tokens (Low)
- Cookie NoHttpOnlyFlag (Low)
- Cookie without SameSite Attribute (Low)
- TimeStamp Disclosure (Low)
- X-Content Type-Options Header Missing(Low)
- Information Disclosure- Suspicious Comments (Medium)
- Information Disclosure - URL (Low)
- Loosely Scooped Cookie (LOW)

1.0 Cross Site Scripting - (DOM Based)

The first vulnerability that ZAP discovered with a security level of low or higher was a reflected cross site scripting vulnerability (high). Below is a screenshot of the ZAP Scanning Report that contains the information regarding the Reflected Cross Site Scripting vulnerability that may be present on the XSS Demos webpage.

Dynamic Analysis for XSS Demos Report 1



Cross Site Scripting (DOM Based)

URL: http://localhost:8082/Tunestore2020/list.do?name=abc#<img src="random.gif" onerror=alert(5397)>

Risk: High

Confidence: High

Parameter:

Attack: ?name=abc#<img src="random.gif" onerror=alert(5397)>

Evidence:

CWE ID: 79

WASC ID: 8

Source: Active (40026 - Cross Site Scripting (DOM Based))

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 web 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

Other Info:

Tag name: input Att name: Att id:

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.

Reference:

http://projects.webappsec.org/Cross-Site-Scripting

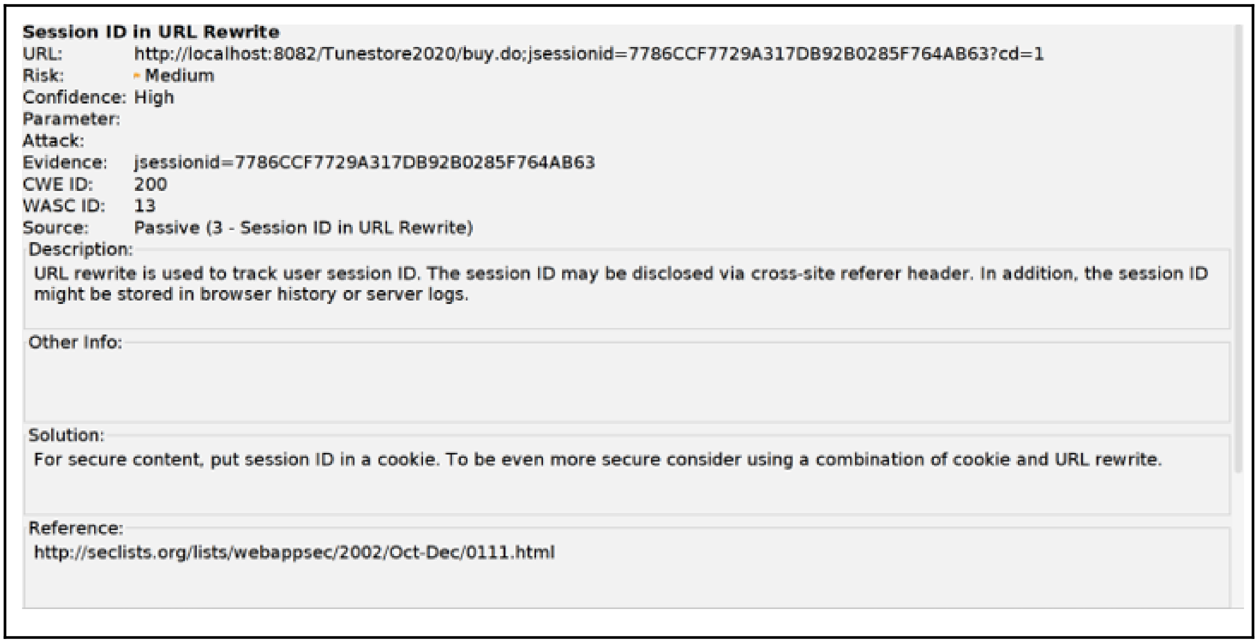
http://cwe.mitre.org/data/definitions/79.html

This Reflected Cross Site Scripting vulnerability is a *true positive*, which means that it is actually present on the XSS Demos webpage, and that it can be maliciously exploited. After ZAP completed the automated scan, the following URL was generated for a Reflected Cross Site

2.0 Session ID URL in Rewrite

The next vulnerability that ZAP discovered with a security level of medium or higher was a Session ID URL in rewrite (High). Below is a screenshot of the ZAP Scanning Report that contains the information regarding the vulnerability that may be present on the XSS Demos webpage.

Dynamic Analysis for XSS Demos Report 2



Session ID in URL Rewrite
URL: http://localhost:8082/Tunestore2020/buy.do;jsessionid=7786CCF7729A317DB92B0285F764AB63?cd=1
Risk: Medium
Confidence: High
Parameter:
Attack:
Evidence: jsessionid=7786CCF7729A317DB92B0285F764AB63
CWE ID: 200
WASC ID: 13
Source: Passive (3 - Session ID in URL Rewrite)

Description:
URL rewrite is used to track user session ID. The session ID may be disclosed via cross-site referer header. In addition, the session ID might be stored in browser history or server logs.

Other Info:

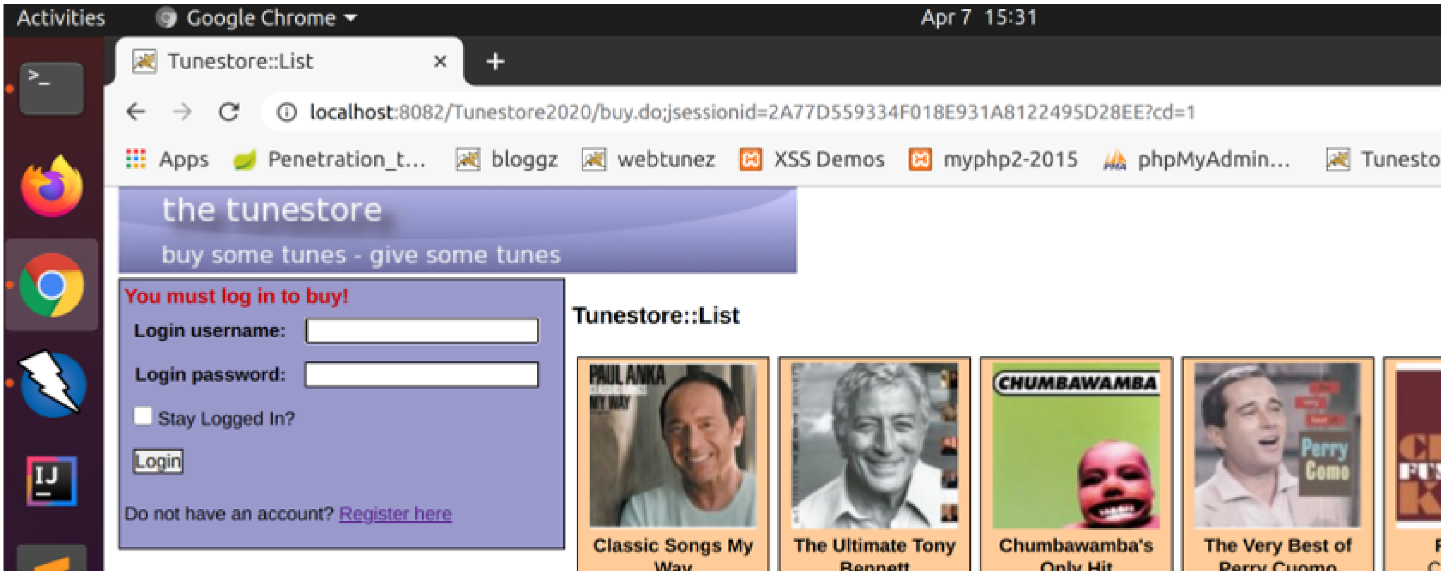
Solution:
For secure content, put session ID in a cookie. To be even more secure consider using a combination of cookie and URL rewrite.

Reference:
http://seclists.org/lists/webappsec/2002/Oct-Dec/0111.html

This CSP Scanner: script-src unsafe-inline vulnerability is a *false positive*, which means that it cannot be maliciously exploited. We can tell that this is a *false positive* by looking at some of the URLs where the ZAP scanning report found the vulnerability:
From looking at these URLs, we can tell that this vulnerability is not one that is contained on the webpage that is being analyzed:

http://localhost:8082/TuneStore2020/buy.do:jsessionid=7786CCF7792A317DB92B0285764AB63?cd=1

Intead, this vulnerability is located on the login page. Since this is not the application that is being reviewed and it requires you to login and cannot be exploited, this would be considered a *false positive*. Below is the screenshot showing the vulnerability.

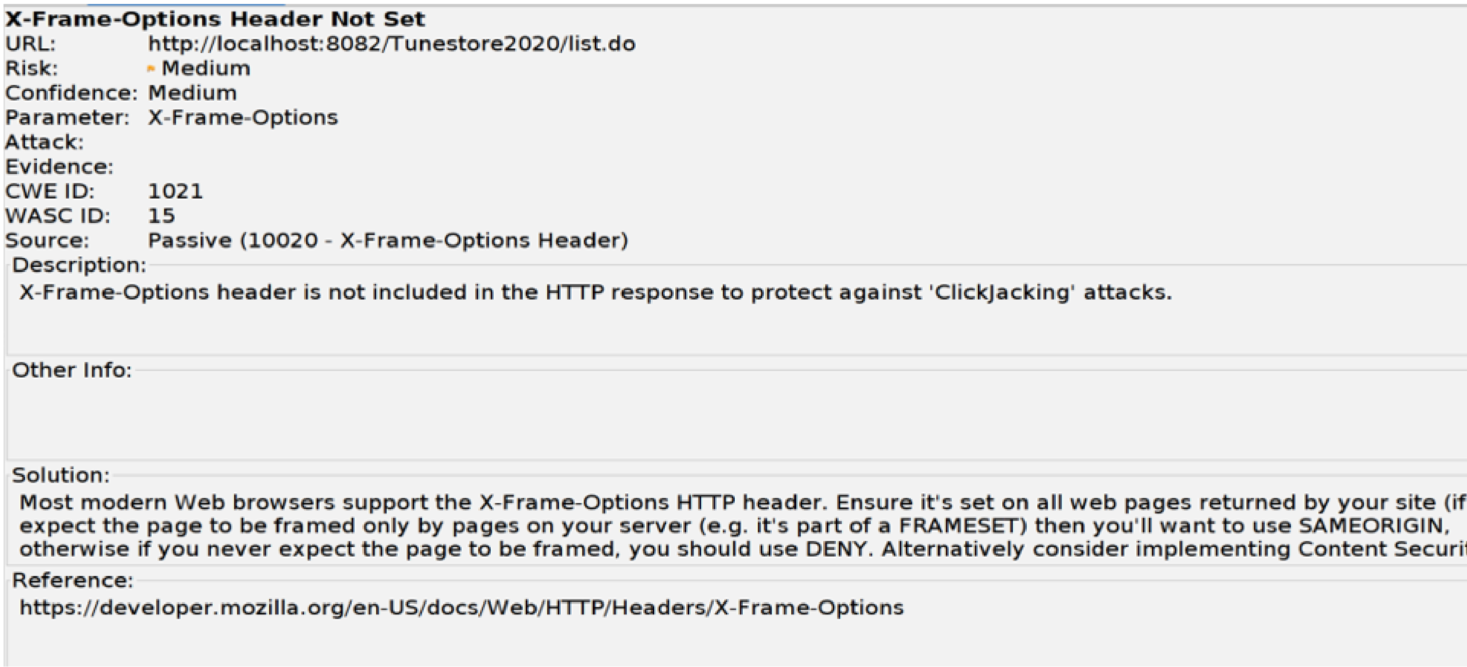


Dynamic Analysis for XSS Demos Report 6

3.0 X-Frame-Options Header Not Set (Medium)

The next vulnerability that ZAP discovered with a security level of medium or higher was a X-Frame-Options Header Not Set (Medium). Below is a screenshot of the ZAP Scanning Report that contains the information regarding the vulnerability that may be present on the XSS Demos webpage

Dynamic Analysis for XSS Demos Report 3



X-Frame-Options Header Not Set

URL: http://localhost:8082/Tunestore2020/list.do
Risk: Medium
Confidence: Medium
Parameter: X-Frame-Options
Attack:
Evidence:
CWE ID: 1021
WASC ID: 15
Source: Passive (10020 - X-Frame-Options Header)

Description:
X-Frame-Options header is not included in the HTTP response to protect against 'ClickJacking' attacks.

Other Info:

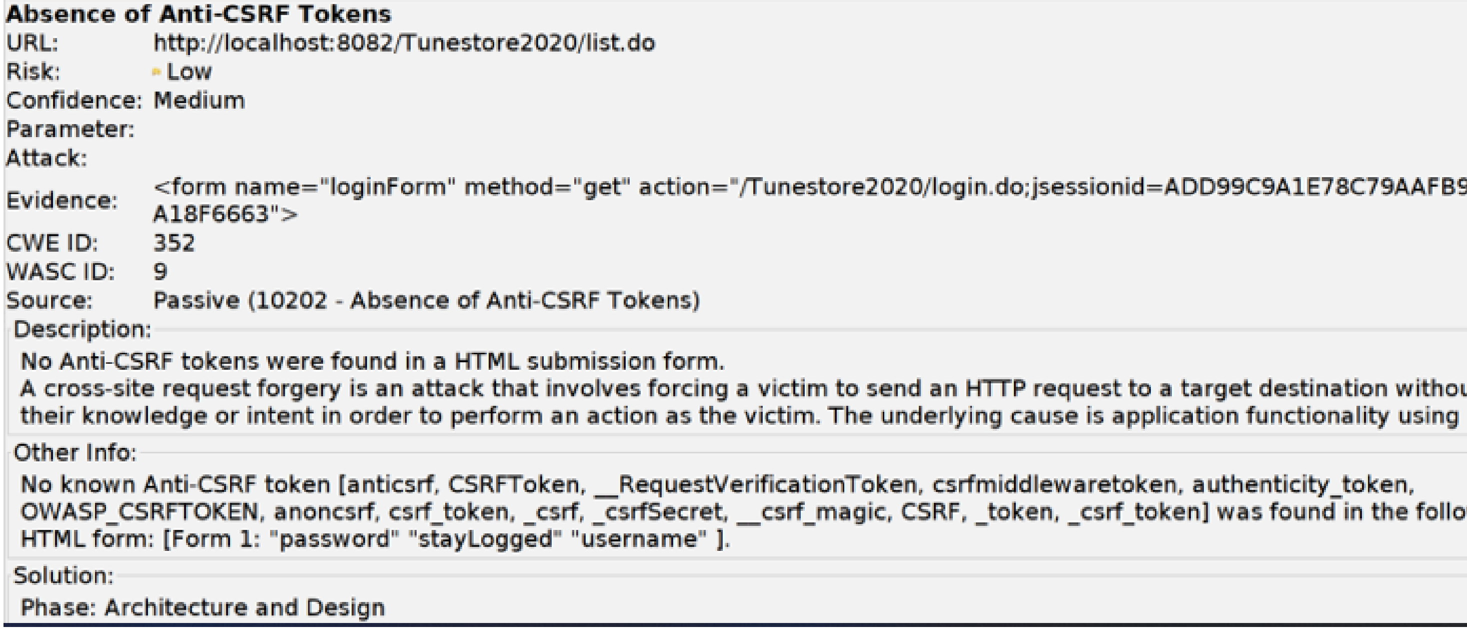
Solution:
Most modern Web browsers support the X-Frame-Options HTTP header. Ensure it's set on all web pages returned by your site (if expect the page to be framed only by pages on your server (e.g. it's part of a FRAMESET) then you'll want to use SAMEORIGIN, otherwise if you never expect the page to be framed, you should use DENY. Alternatively consider implementing Content Securi

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

4.0 Absence of Anti-CSRF Tokens

The next vulnerability that ZAP discovered with a security level of low or medium was an Absence of Anti CSRF Tokens (Low- Medium). Below is a screenshot of the ZAP Scanning Report that contains the information regarding the vulnerability that may be present on the XSS Demos webpage.

Dynamic Analysis for XSS Demos Report 4



Absence of Anti-CSRF Tokens

URL: http://localhost:8082/Tunestore2020/list.do
Risk: Low
Confidence: Medium
Parameter:
Attack:
Evidence: <form name="loginForm" method="get" action="/Tunestore2020/login.do;jsessionid=ADD99C9A1E78C79AAFB9
A18F6663">
CWE ID: 352
WASC ID: 9
Source: Passive (10202 - Absence of Anti-CSRF Tokens)

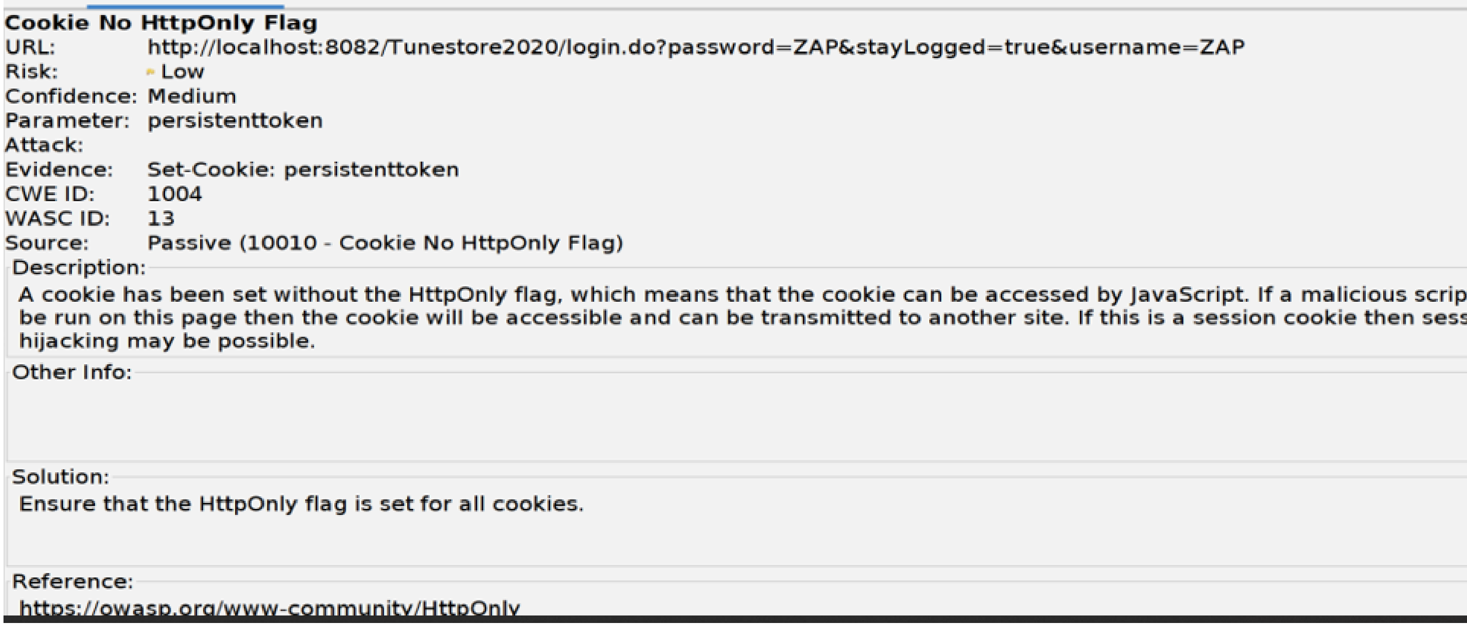
Description:
No Anti-CSRF tokens were found in a HTML submission form.
A cross-site request forgery is an attack that involves forcing a victim to send an HTTP request to a target destination withou
their knowledge or intent in order to perform an action as the victim. The underlying cause is application functionality using

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 follo
HTML form: [Form 1: "password" "stayLogged" "username"].

Solution:
Phase: Architecture and Design

5.0 Cookie NoHttpOnly Flag

The next vulnerability that ZAP discovered with a security level of medium or lower was a Cookie No HttpOnly Flag (Medium). Below is a screenshot of the ZAP Scanning Report that contains the information regarding the vulnerability that may be present on the XSS Demos webpage.



Cookie No HttpOnly Flag

URL: http://localhost:8082/Tunestore2020/login.do?password=ZAP&stayLogged=true&username=ZAP
Risk: Low
Confidence: Medium
Parameter: persistenttoken
Attack:
Evidence: Set-Cookie: persistenttoken
CWE ID: 1004
WASC ID: 13
Source: Passive (10010 - Cookie No HttpOnly Flag)

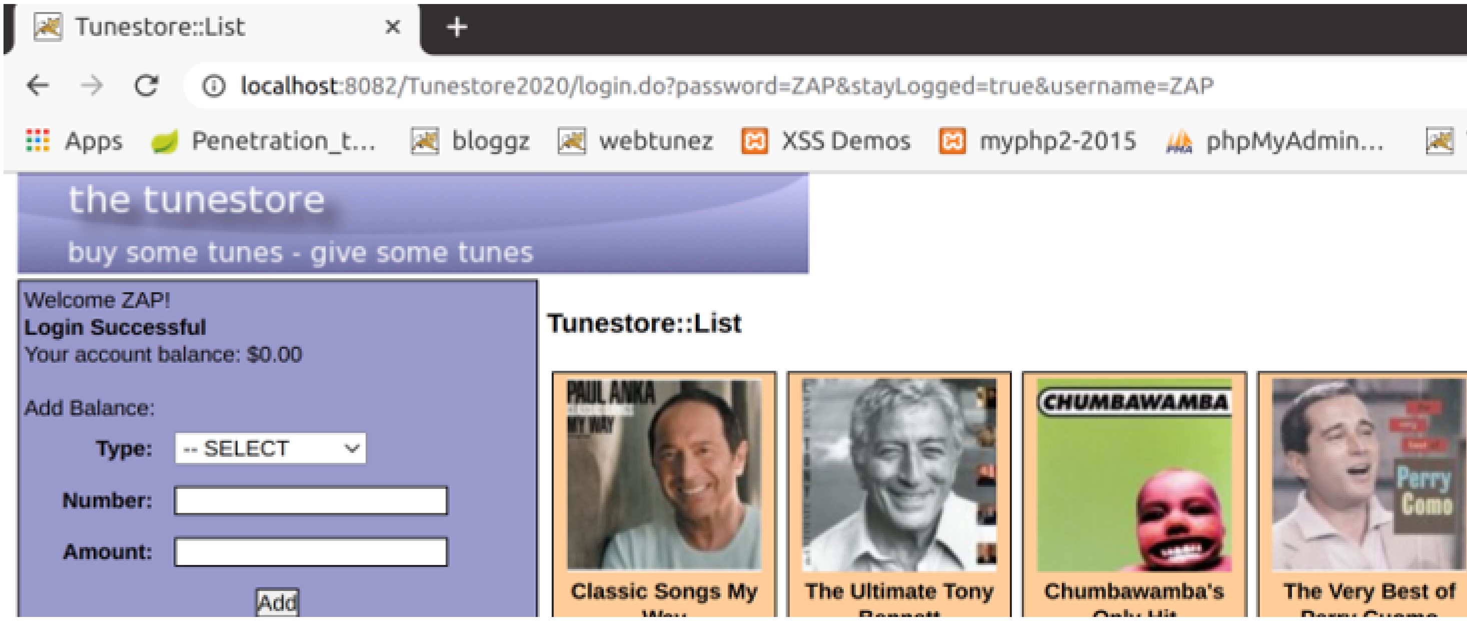
Description:
A cookie has been set without the HttpOnly flag, which means that the cookie can be accessed by JavaScript. If a malicious scrip
be run on this page then the cookie will be accessible and can be transmitted to another site. If this is a session cookie then sess
hijacking may be possible.

Other Info:

Solution:
Ensure that the HttpOnly flag is set for all cookies.

Reference:
https://owasp.org/www-community/HttpOnly

This reflected vulnerability is *true positive* because it can be maliciously exploited, and it is actually present on the XSS demo pages. After ZAP cleared its automated scan. The Link in the above screenshot directed me to the vulnerability. Below is the screenshot where you can see it. As you can see below, the user can automatically login. And most of the alerts in this report share the same risk and the vulnerability, where the users can login with the URL.



Tunestore::List
localhost:8082/Tunestore2020/login.do?password=ZAP&stayLogged=true&username=ZAP
Apps
Penetration_t...
bloggz
webtunez
XSS Demos
myphp2-2015
phpMyAdmin...
the tunestore
buy some tunes - give some tunes

Welcome ZAP!
Login Successful
Your account balance: $0.00

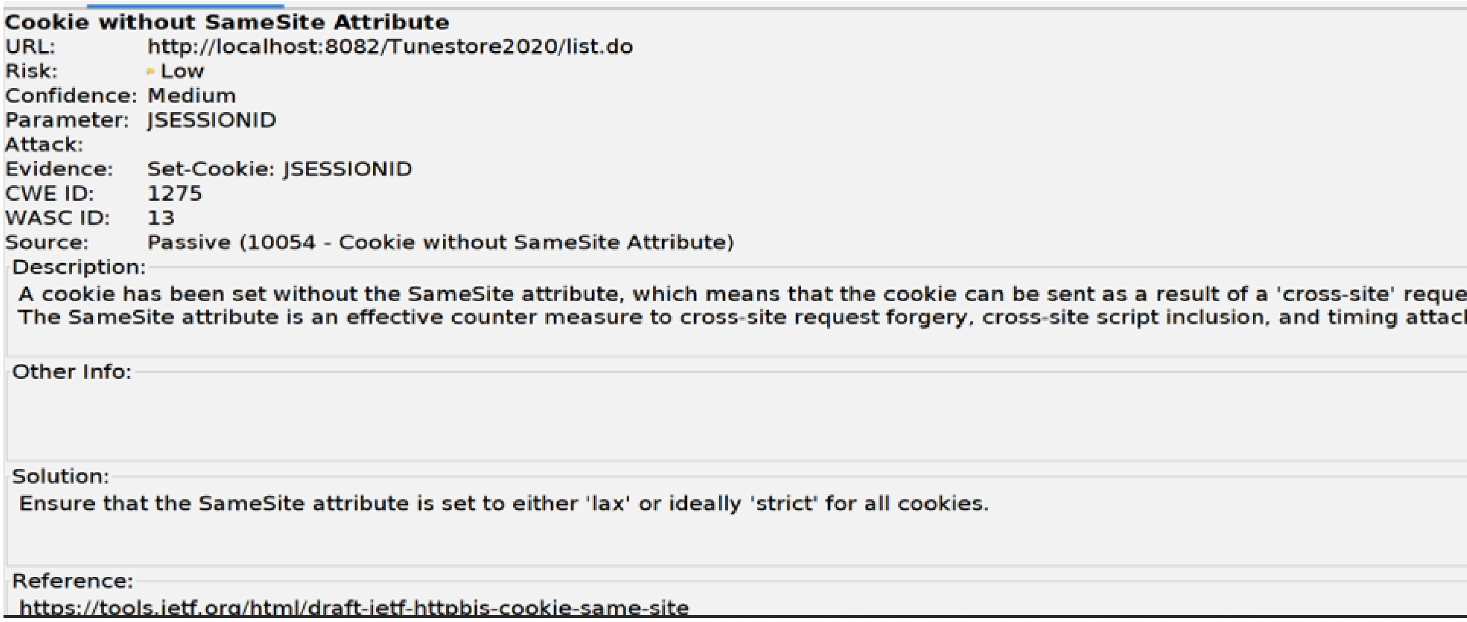
Add Balance:
Type: -- SELECT
Number:
Amount:
Add

Tunestore::List

6.0 Cookie Without SameSite Attribute

The next vulnerability that ZAP discovered with a security level of medium or lower was a Cookie without SameSite Attribute (High). Below is a screenshot of the ZAP Scanning Report that contains the information regarding the vulnerability that may be present on the XSS Demos webpage.

Dynamic Analysis for XSS Demos Report 2



Cookie without SameSite Attribute

```
URL:          http://localhost:8082/Tunestore2020/list.do
Risk:         Low
Confidence:   Medium
Parameter:    JSESSIONID
Attack:
Evidence:     Set-Cookie: JSESSIONID
CWE ID:       1275
WASC ID:      13
Source:       Passive (10054 - Cookie without SameSite Attribute)
```

Description:

A cookie has been set without the SameSite attribute, which means that the cookie can be sent as a result of a 'cross-site' reque
The SameSite attribute is an effective counter measure to cross-site request forgery, cross-site script inclusion, and timing attac

Other Info:

Solution:

Ensure that the SameSite attribute is set to either 'lax' or ideally 'strict' for all cookies.

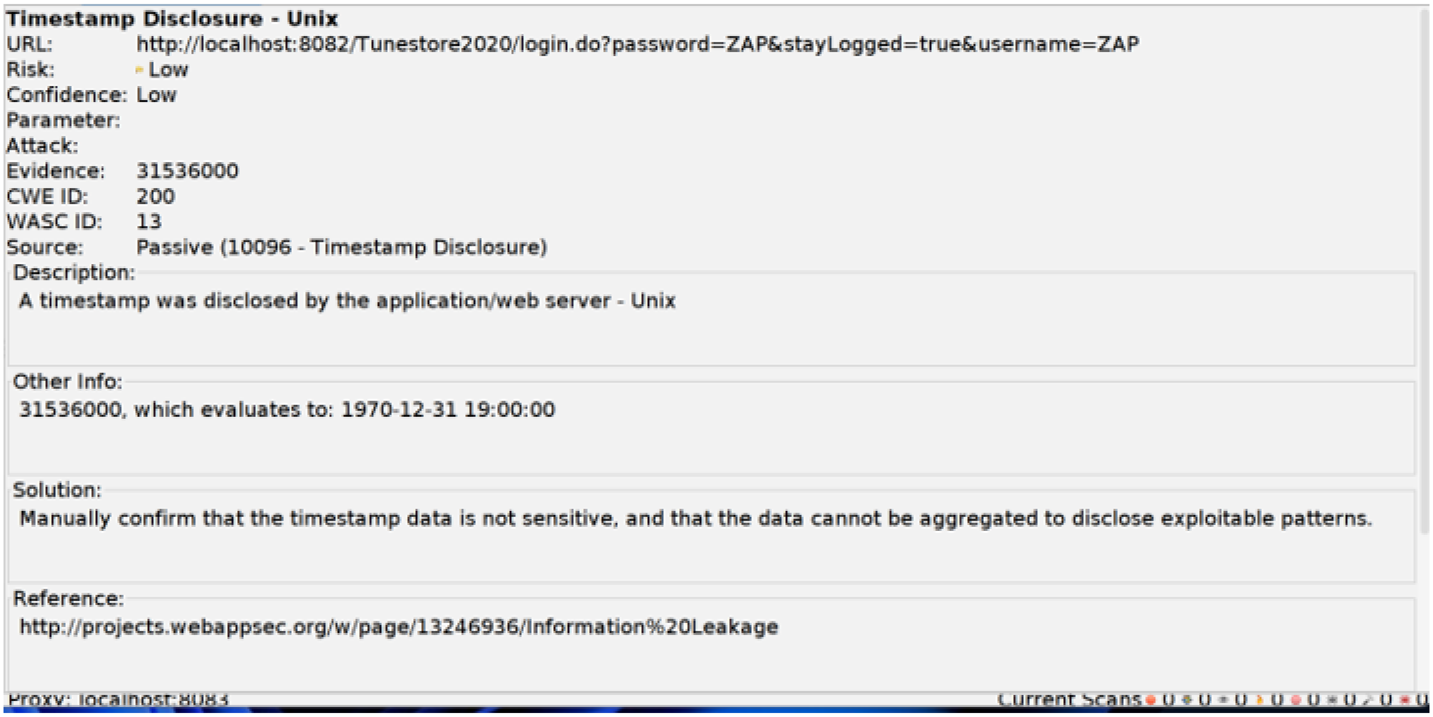
Reference:

https://tools.ietf.org/html/draft-ietf-httpbis-cookie-same-site

7.0 TimeStamp Disclosure -Unix (Low)

The next vulnerability that ZAP discovered with a security level of lower was a TimeStamp Disclosure (Low). Below is a screenshot of the ZAP Scanning Report that contains the information regarding the vulnerability that may be present on the XSS Demos webpage.

Dynamic Analysis for XSS Demos Report 7



Timestamp Disclosure - Unix
URL: http://localhost:8082/Tunestore2020/login.do?password=ZAP&stayLogged=true&username=ZAP
Risk: Low
Confidence: Low
Parameter:
Attack:
Evidence: 31536000
CWE ID: 200
WASC ID: 13
Source: Passive (10096 - Timestamp Disclosure)

Description:
A timestamp was disclosed by the application/web server - Unix

Other Info:
31536000, which evaluates to: 1970-12-31 19:00:00

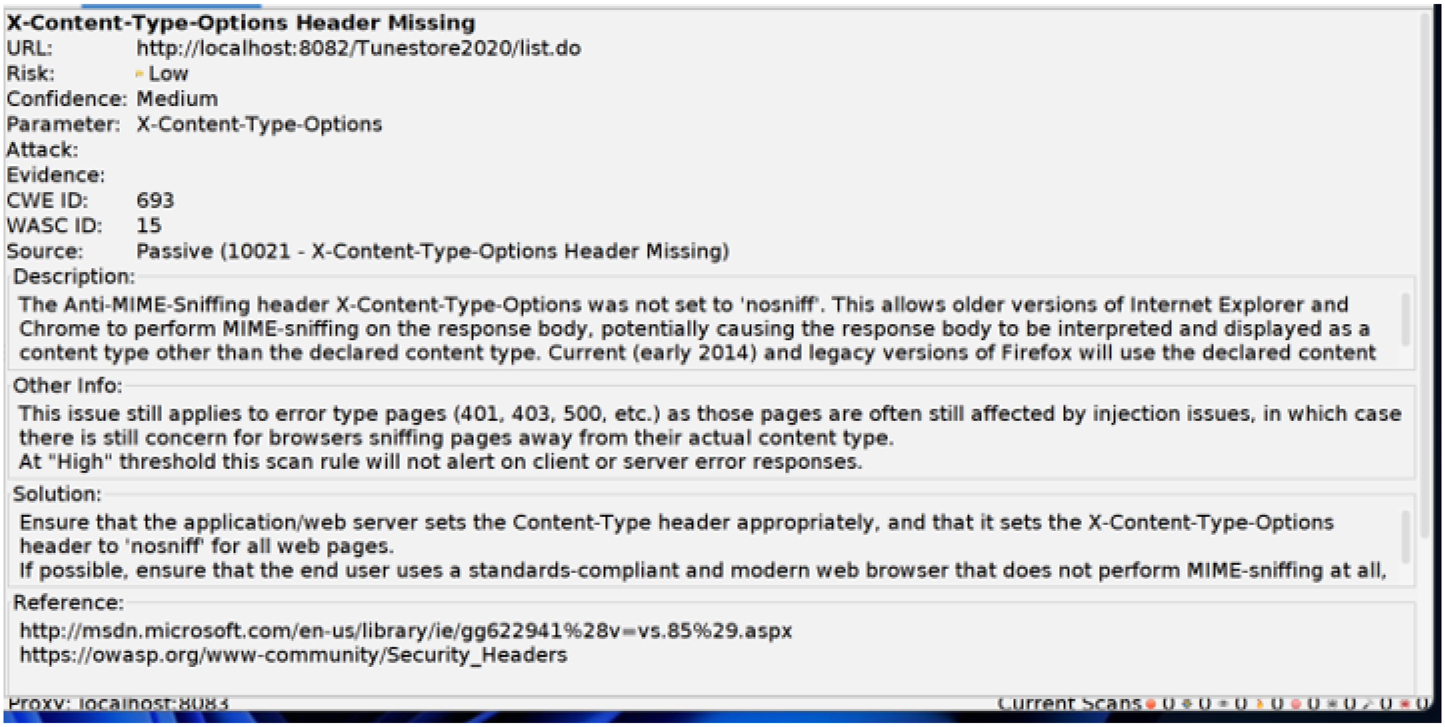
Solution:
Manually confirm that the timestamp data is not sensitive, and that the data cannot be aggregated to disclose exploitable patterns.

Reference:
http://projects.webappsec.org/w/page/13246936/Information%20Leakage

This reflected vulnerability is *true positive* because it can be maliciously exploited, and it is actually present on the XSS demo pages. After ZAP cleared its automated scan. The Link in the above screenshot directed me to the vulnerability. Below is the screenshot where you can see it. As you can see below, the user can automatically login. And most of the alerts in this report share the same risk and the vulnerability, where the users can login with the URL.
This vulnerability is shared with *5.0* where the users can login automatically by altering the link's URL.

8.0 X-Content Type-Options Header Missing(Low)

The next vulnerability that ZAP discovered with a security level of **Lower** was a **X-Content Type-Options Header Missing**. Below is a screenshot of the ZAP Scanning Report that contains the information regarding the vulnerability that may be present on the XSS Demos webpage.
Dynamic Analysis for XSS Demos Report 8



X-Content-Type-Options Header Missing

URL:	http://localhost:8082/Tunestore2020/list.do
Risk:	Low
Confidence:	Medium
Parameter:	X-Content-Type-Options
Attack:	
Evidence:	
CWE ID:	693
WASC ID:	15
Source:	Passive (10021 - X-Content-Type-Options Header Missing)

Description:

The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'. This allows older versions of Internet Explorer and Chrome to perform MIME-sniffing on the response body, potentially causing the response body to be interpreted and displayed as a content type other than the declared content type. Current (early 2014) and legacy versions of Firefox will use the declared content

Other Info:

This issue still applies to error type pages (401, 403, 500, etc.) as those pages are often still affected by injection issues, in which case there is still concern for browsers sniffing pages away from their actual content type.
At "High" threshold this scan rule will not alert on client or server error responses.

Solution:

Ensure that the application/web server sets the Content-Type header appropriately, and that it sets the X-Content-Type-Options header to 'nosniff' for all web pages.
If possible, ensure that the end user uses a standards-compliant and modern web browser that does not perform MIME-sniffing at all,

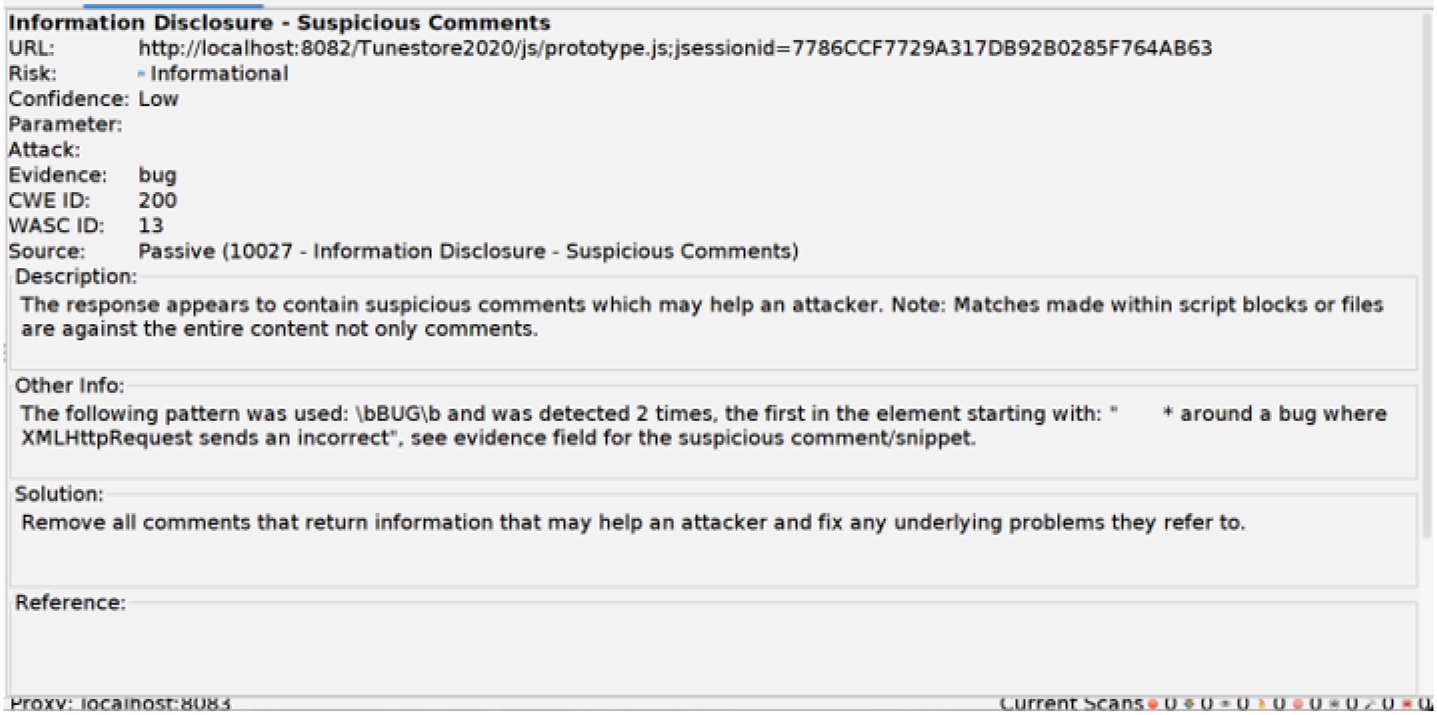
Reference:

http://msdn.microsoft.com/en-us/library/ie/gg622941%28v=vs.85%29.aspx
https://owasp.org/www-community/Security_Headers

9.0 Information Disclosure - Suspicious Comments (Informational)

The next vulnerability that ZAP discovered with a security level of **Informational** was an Informational Disclosure. Below is a screenshot of the ZAP Scanning Report that may be present on the XSS Demos webpage.

Dynamic Analysis for XSS Demos Report 9



Information Disclosure - Suspicious Comments

URL:	http://localhost:8082/Tunestore2020/js/prototype.js;jsessionid=7786CCF7729A317DB92B0285F764AB63
Risk:	Informational
Confidence:	Low
Parameter:	
Attack:	
Evidence:	bug
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: \bBUG\b and was detected 2 times, the first in the element starting with: " * around a bug where XMLHttpRequest sends an incorrect", 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:

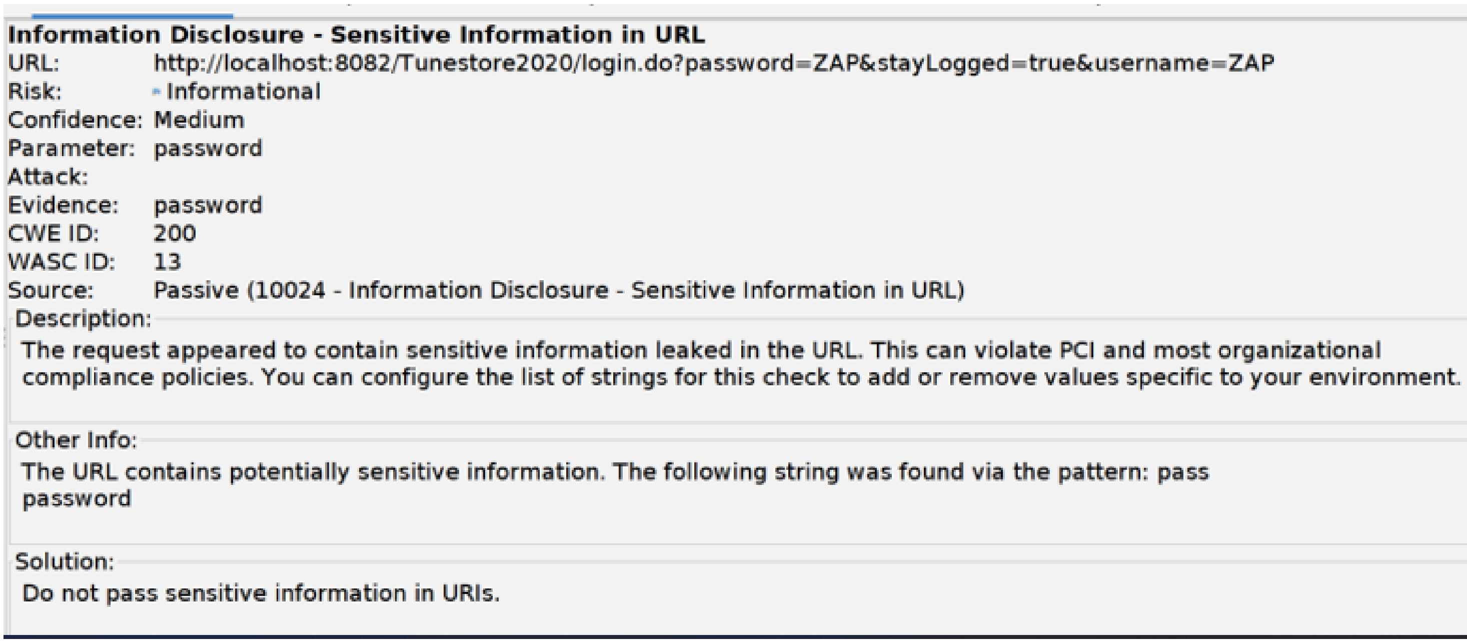
This vulnerability is shared with 1.0 where the user has to login to buy something, henceforth a false

positive.

10.0 Information Disclosure- Sensitive Information in URL (Informational)

The next vulnerability that ZAP discovered with a security level of **Informational** was an Informational Disclosure - Sensitive Information in URL. Below is a screenshot of the ZAP Scanning Report that may be present on the XSS Demos webpage.

Dynamic Analysis for XSS Demos Report 10



Information Disclosure - Sensitive Information in URL
URL: http://localhost:8082/Tunestore2020/login.do?password=ZAP&stayLogged=true&username=ZAP
Risk: Informational
Confidence: Medium
Parameter: password
Attack:
Evidence: password
CWE ID: 200
WASC ID: 13
Source: Passive (10024 - Information Disclosure - Sensitive Information in URL)

Description:
The request appeared to contain sensitive information leaked in the URL. This can violate PCI and most organizational compliance policies. You can configure the list of strings for this check to add or remove values specific to your environment.

Other Info:
The URL contains potentially sensitive information. The following string was found via the pattern: pass
password

Solution:
Do not pass sensitive information in URIs.

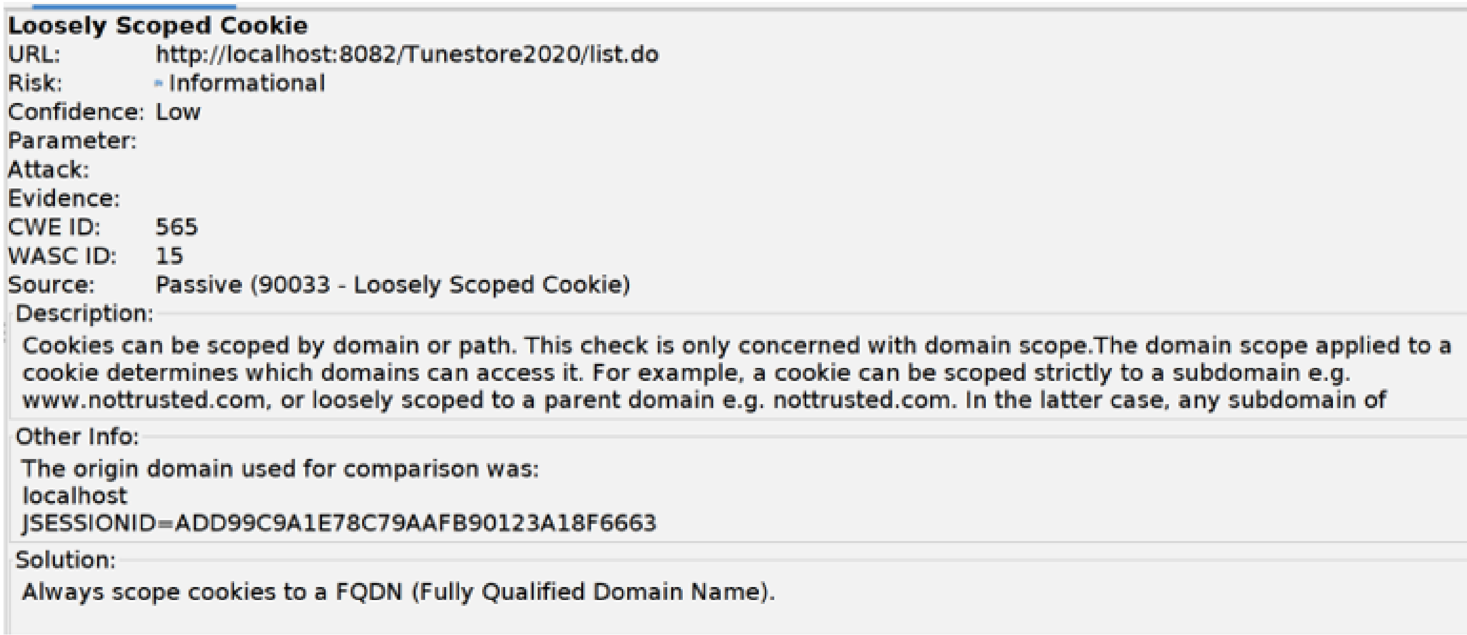
This reflected vulnerability is *true positive* because it can be maliciously exploited, and it is actually present on the XSS demo pages. After ZAP cleared its automated scan. The Link in the above screenshot directed me to the vulnerability. Below is the screenshot where you can see it. As you can see below, the user can automatically login. And most of the alerts in this report share the same risk and the vulnerability, where the users can login with the URL.

This vulnerability is shared with *5.0* where the user is already logged in and they can access the users funds.

11.0 Loosely Scooped Cookie (Informational)

The next vulnerability that ZAP discovered with a security level of **Informational** was a Loosely Scooped Cookie. Below is a screenshot of the ZAP Scanning Report that may be present on the XSS Demos webpage.

Dynamic Analysis for XSS Demos Report 11



Loosely Scoped Cookie

URL: http://localhost:8082/Tunestore2020/list.do
Risk: Informational
Confidence: Low
Parameter:
Attack:
Evidence:
CWE ID: 565
WASC ID: 15
Source: Passive (90033 - Loosely Scoped Cookie)

Description:

Cookies can be scoped by domain or path. This check is only concerned with domain scope.The domain scope applied to a cookie determines which domains can access it. For example, a cookie can be scoped strictly to a subdomain e.g. www.nottrusted.com, or loosely scoped to a parent domain e.g. nottrusted.com. In the latter case, any subdomain of

Other Info:

The origin domain used for comparison was:
localhost
JSESSIONID=ADD99C9A1E78C79AAFB90123A18F6663

Solution:

Always scope cookies to a FQDN (Fully Qualified Domain Name).