# SECURITY ASSESSMENT

## THIORO FALL

**Submitted to: Udajuicer Development Team**
**Security Analyst: Security Analyst Team**

**Date of Testing: 7/5/2021**
**Date of Report Delivery: 7/5/2021**

# Table of Contents

## Contents


**SECURITY ENGAGEMENT SUMMARY ..... 2**

ENGAGEMENT OVERVIEW ..... 2

SCOPE ..... 2

EXECUTIVE RISK ANALYSIS ..... 2

ENGAGEMENT SUMMARY FORM ..... 4

EXECUTIVE RECOMMENDATION ..... 5

**SIGNIFICANT VULNERABILITY SUMMARY ..... 6**

High Risk Vulnerabilities ..... 6
Medium Risk Vulnerabilities ..... 6
Low Risk Vulnerabilities ..... 6

**SIGNIFICANT VULNERABILITY DETAIL ..... 8**

CROSS-DOMAIN MISCONFIGURATION ..... 8

CROSS-DOMAIN JAVASCRIPT SOURCE FILE INCLUSION ..... 9

WEB BROWSER XSS PROTECTION NOT ENABLED ..... 10

INFORMATION DISCLOSURE – SUSPICIOUS COMMENTS ..... 11

TIMESTAMP DISCLOSURE -UNIX ..... 12

THIS MAY BE INTERESTING… ..... 13

**METHODOLOGY ..... 14**

ASSESSMENT TOOLSET SELECTION ..... 14

ASSESSMENT METHODOLOGY DETAIL ..... 14

REFERENCES ..... 19

# Security Engagement Summary

## Engagement Overview

The Udajuicer Development team that is responsible for the organization's web-applications has requested a vulnerability assessment of a legacy web-application to help them understand what security risk the web-application is posing to the organization

The goal of this engagement is to have a better understanding of the security risk associated with the web application and what mitigations are possible to increase the security posture and reduce the risk to the organization. The assessment evaluates the security of the application against best practice criteria to validate security mechanisms and identify application-level vulnerabilities

A security analyst working in the Information Security department at Udajuicer will be in charge of the engagement.

The assessment should be completed once every quarter.

## Scope

The purpose of the engagement is to utilize assessment techniques in order to identify and validate potential vulnerabilities across all systems within scope.

The Udajuicer Development Team defined the following IP address and port number:

- 127.0.0.1
- 3000

## Executive Risk Analysis

The risk of vulnerabilities listed in the Executive Summary Form below give the overall risk the Juice Shop is facing. This form identifies some security risks that could have significant impact on Juice shop for their day-to-day business operations.

The graph below shows a summary of the number of vulnerabilities found for the Web Application Security Assessment.

No high impact vulnerabilities were found. But a significant number of medium impact vulnerabilities were found.

| HIGH SEVERITY | MEDIUM SEVERITY | LOW SEVERITY | INFORMATIOMAL |
|---|---|---|---|
| 0 | 30 | 17 | 29 |

# Executive Summary Form

The Udajuicer Development Team that is responsible for the organization's web-applications has requested a vulnerability assessment of a legacy web-application to help them understand what security risk the web-application is posing to the organization, and what mitigations are possible to increase the security posture and reduce the risk to the organization.

The purpose of the engagement was to utilize vulnerability assessment techniques in order to evaluate the security of the application against best practice criteria and to validate its security.

The Udajuicer Development Team has provided a way to be able to access the Juice Shop application at the given IP address and port:127.0.0.1 or localhost at port 3000.

This report details the scope of testing conducted, all significant findings along with detailed remedial advice.

The summary below is addressed to a non-technical audience. The first section provides a significant vulnerability summary of the key findings and relates these back to business impacts. The second section of this report highlights a summary details of each vulnerability found.

We explained:

- The assessment risk level of the vulnerability
- How the vulnerability was identified and validated
- The probability of exploit (attack)
- Who would be impacted if the attack was exploited?
- Potential remediation.

The last section of this report provides detailed assessment methodology and toolset used.

Below is a summary of all the vulnerability found during the assessment and their remediations:

| Vulnerability | Severity Level | Remediations |
|---|---|---|
| Cross-Domain Misconfiguration | Medium | A solution is to ensure sensitive data is unavailable in an unauthenticated manner by using IP address white listing. |
| Cross-Domain JavaScript Source File Inclusion | Low | A solution is to ensure JavaScript files are loaded from only trusted sources and the sources can't be controlled by end users of the application. |

| | | |
|---|---|---|
| Web Browser XSS Protection Not Enabled | Low | For the solution, ensure that the web browser's X-XSS filter is enabled by setting the X-XSS protection HTTP response header on the web server to 'I'. |
| Information Disclosure – Suspicious Comments | Informational | The solution is to remove all comments that return information for the attacker and fix any underlying problems they refer to. |
| Timestamp Disclosure -Unix | Informational | The solution is to manually confirm that the timestamp data is not sensitive, and that the data cannot be aggregated to disclose exploitable patterns. |
| This may be interesting… | Informational | The solution is to modify the Apache configuration. |

# Executive Recommendation

Recommendations for this form are based on the available findings from the scanner. We have a medium risk which should be remediated first.

The solution is to ensure sensitive data is unavailable in an unauthenticated manner by using IP address white listing.

The results should not be interpreted as definitive measurement of the security posture of the Juice Shop network.

Also other elements used to assess the current security posture should include policy review, a review of internal security controls and procedures, or internal red teaming/penetration testing.

# Significant Vulnerability Summary

## High Risk Vulnerabilities

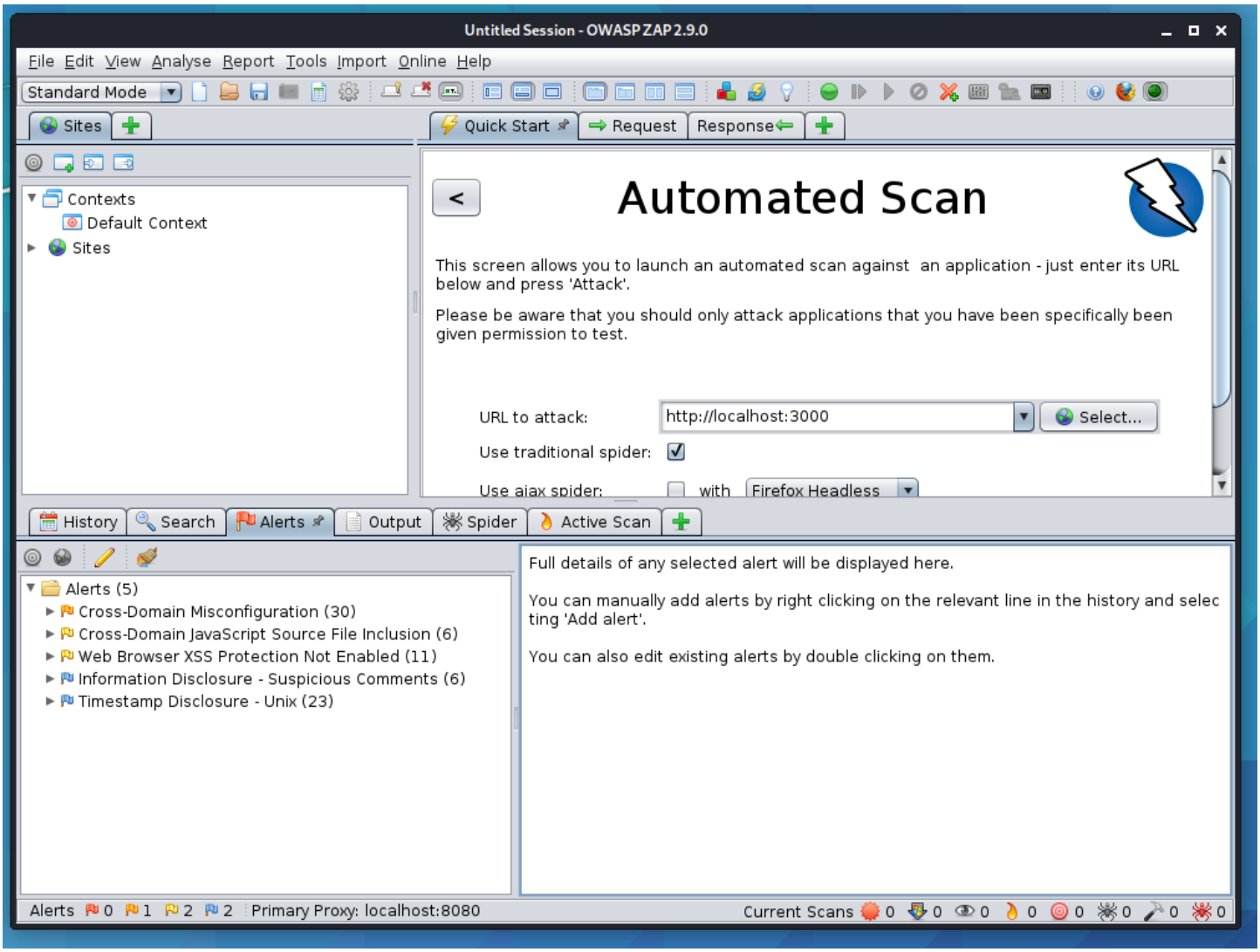
- None

## Medium Risk Vulnerabilities

- Cross-Domain Misconfiguration (30)

## Low Risk Vulnerabilities

- Cross-Domain JavaScript Source File Inclusion (6)
- Web Browser XSS protection Not Enabled (11)


Untitled Session - OWASP ZAP 2.9.0
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://localhost:3000
Select...
Use traditional spider:
Use ajax spider:
with
Firefox Headless
History
Search
Alerts
Output
Spider
Active Scan
Alerts (5)
Cross-Domain Misconfiguration (30)
Cross-Domain JavaScript Source File Inclusion (6)
Web Browser XSS Protection Not Enabled (11)
Information Disclosure - Suspicious Comments (6)
Timestamp Disclosure - Unix (23)
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.
Alerts 0 1 2 2
Primary Proxy: localhost:8080
Current Scans 0 0 0 0 0 0 0 0

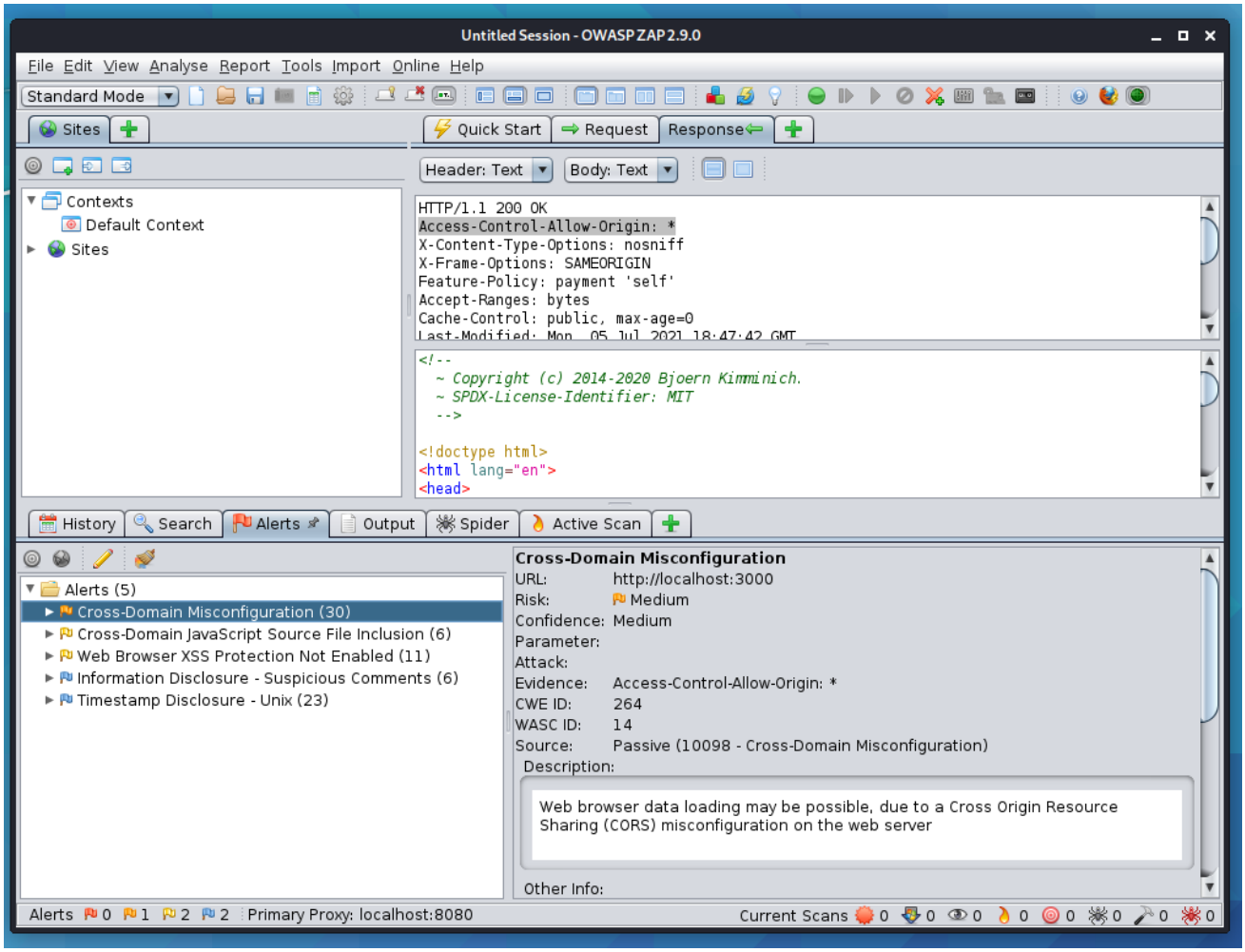
# Significant Vulnerability Details:

## ❖ Cross-Domain Misconfiguration

**RISK LEVEL: MEDIUM**

Vulnerability details:

- The assessed risk level of the vulnerability is Medium.
- The CORS misconfiguration on the webserver permits cross domain read requests from arbitrary third-party domains, using unauthenticated APIs on this domain. Web server implementations don't permit arbitrary third parties to read the response from authenticated APIs.
- The probability of exploit is medium. This misconfiguration could be used by an attacker to access data that is available in an unauthenticated manner.
- This type of attack may impact users-groups, departments, business-continuity/revenue
- A solution is to ensure sensitive data is unavailable in an unauthenticated manner by using IP address white listing.

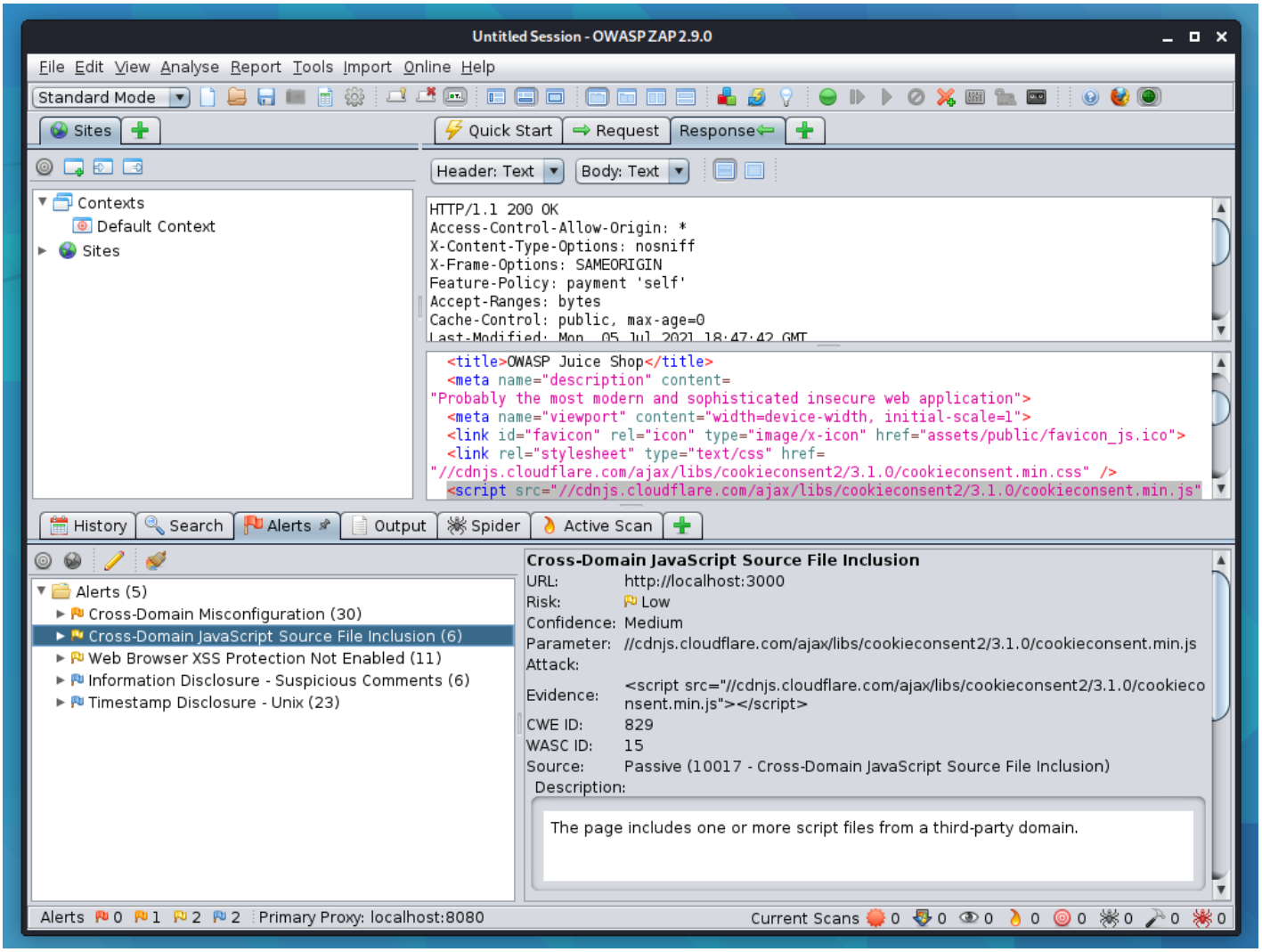


## ❖ Cross-Domain JavaScript Source File Inclusion

**RISK LEVEL: LOW**

Vulnerability details:

- The vulnerability has Low risk.
- The page had one or more script files from a third-party domain.
- The probability of exploit is low as the exploit might not give important information to the user.
- The departments will be affected since they need to fixe bugs to the files that are infected.
- A solution is to ensure JavaScript files are loaded from only trusted sources and the sources can't be controlled by end users of the application.



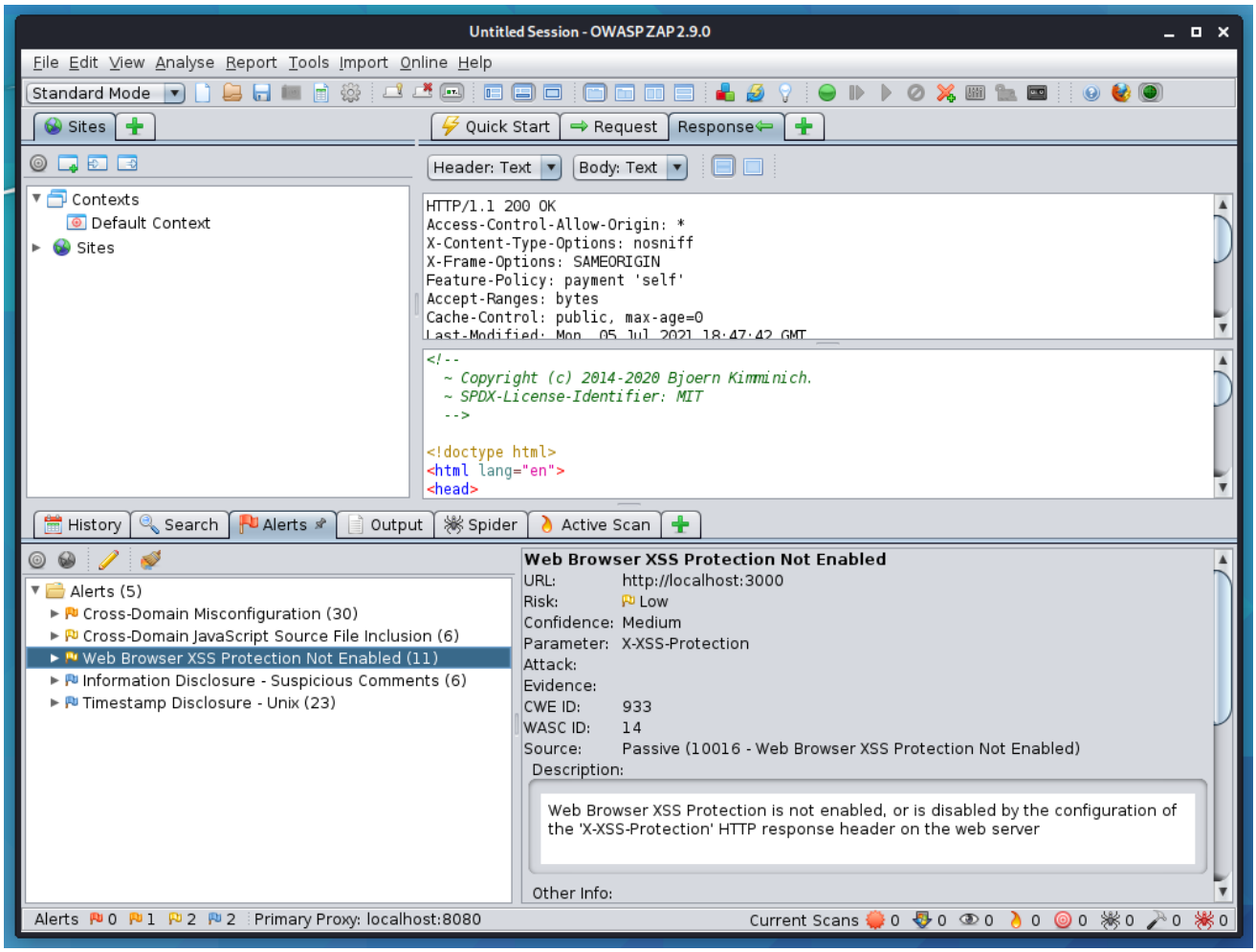
## ❖ Web Browser XSS Protection Not Enabled

**RISK LEVEL: LOW**

Vulnerability details:

- The assessed risk level of the vulnerability is Low.
- Web browser XSS protection is not enabled or is disabled by the configuration of the X-XSS protection HTTP response header on the web server.
- The probability of exploit/attacks is low. An attacker can still take advantage of this insecure web browser.
- If the attack was exploited the whole company and clients will be impacted since it is a web browser.
- For the solution, ensure that the web browser's X-XSS filter is enabled by setting the X-XSS protection HTTP response header on the web server to 'l'.

-

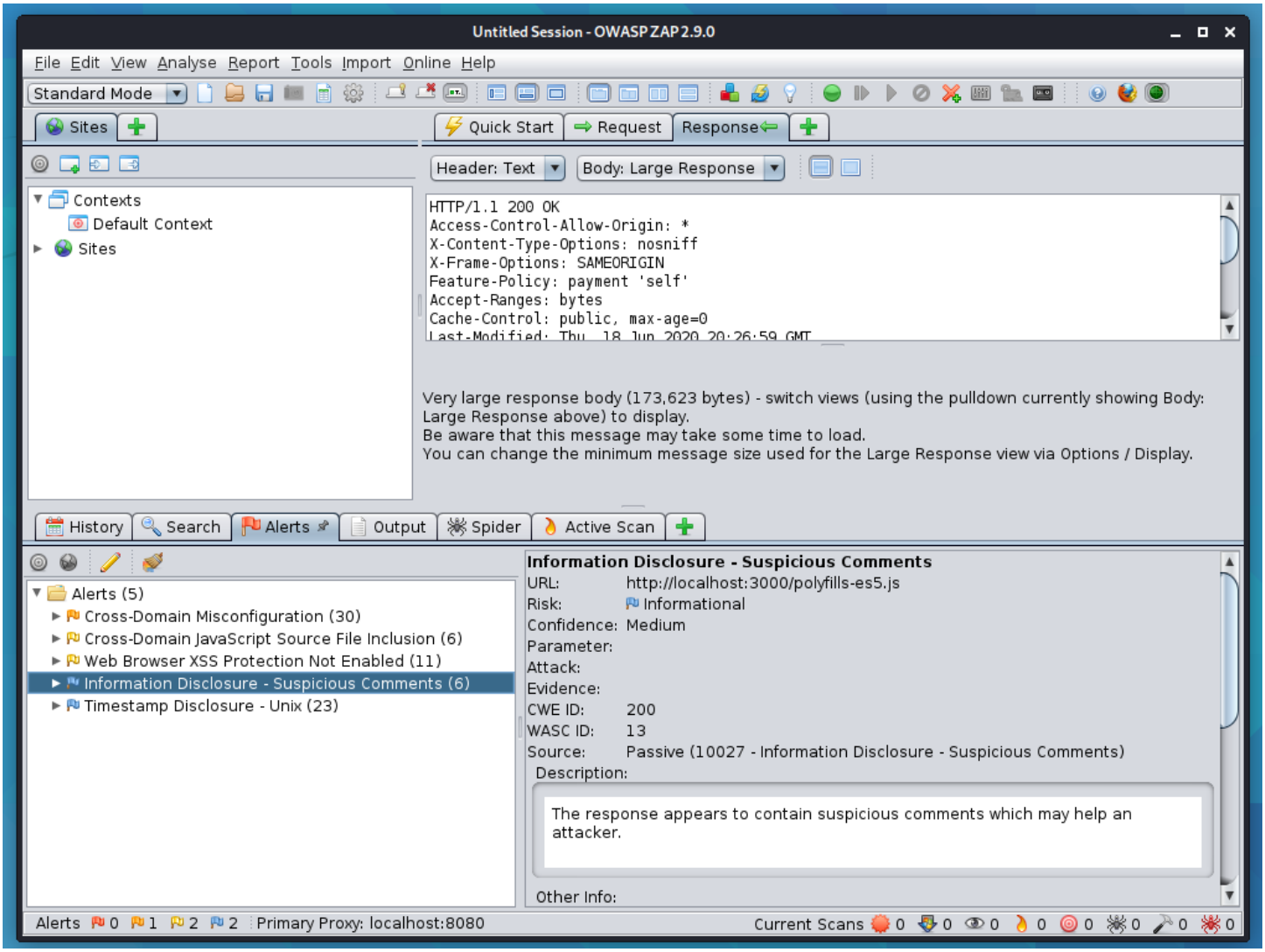


## ❖ Information Disclosure - Suspicious Comments

**RISK LEVEL: Informational**

Vulnerability details:

- The Vulnerability has Informational risk.
- The response appears to contain suspicious comments which may help an attacker.
- The probability of exploit is insignificant. We can remove the comments which leads information for the attacker.
- This won't affect anyone as this is just informational.
- A solution is to remove all comments that return information for the attacker and fix any underlying problems they refer to.

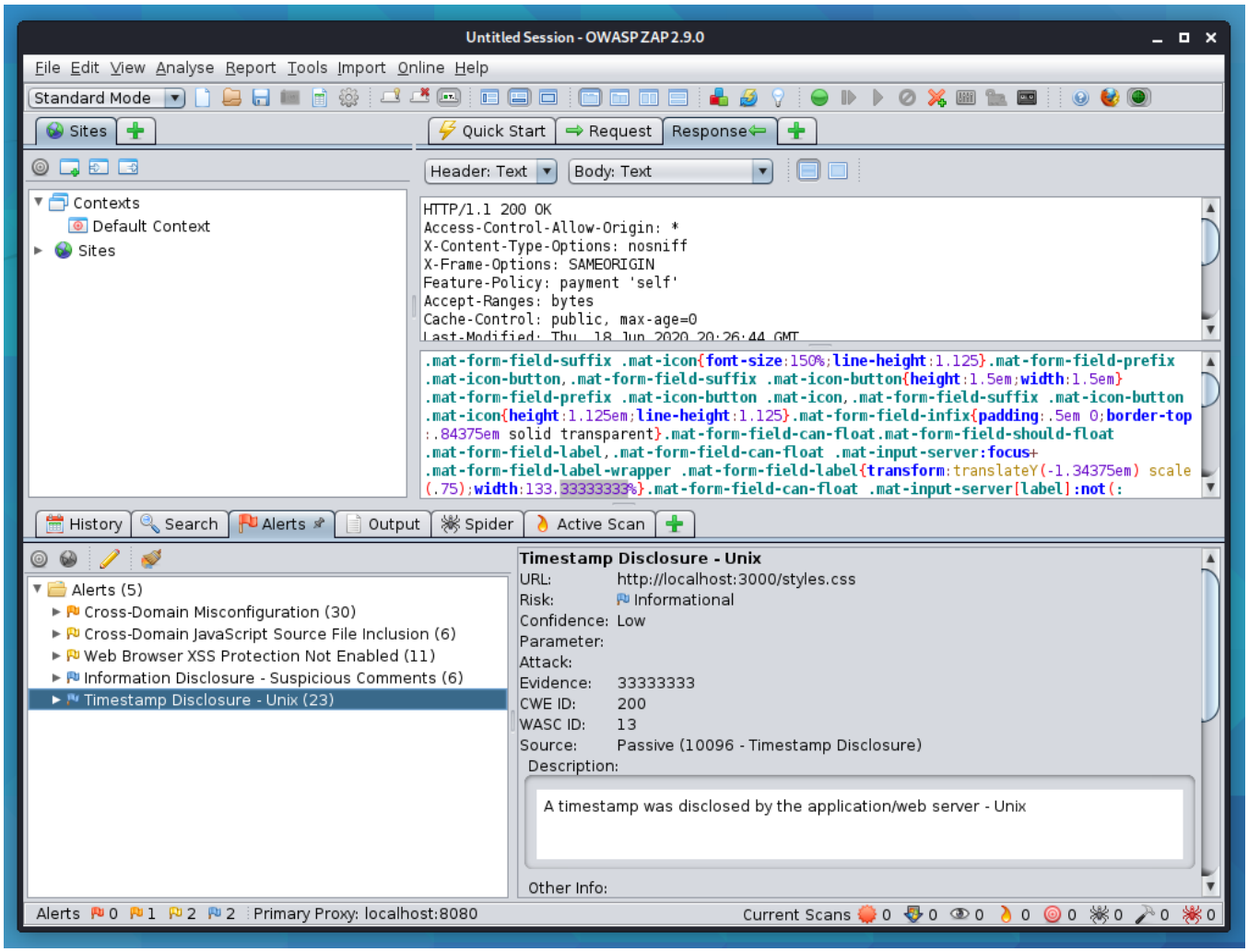


## ❖ Timestamp Disclosure -Unix

**RISK LEVEL: Informational**

Vulnerability details

- The Vulnerability has Informational risk.
- The timestamp was disclosed by application/web server – Unix. For evidence we have 33333333, which evaluates to: 1971-01-21 14:15:33.
- The probability of exploit is insignificant.
- This informational risk level should not affect anybody.
- The solution is to manually confirm that the timestamp data is not sensitive, and that the datacannot be aggregated to disclose exploitable patterns.

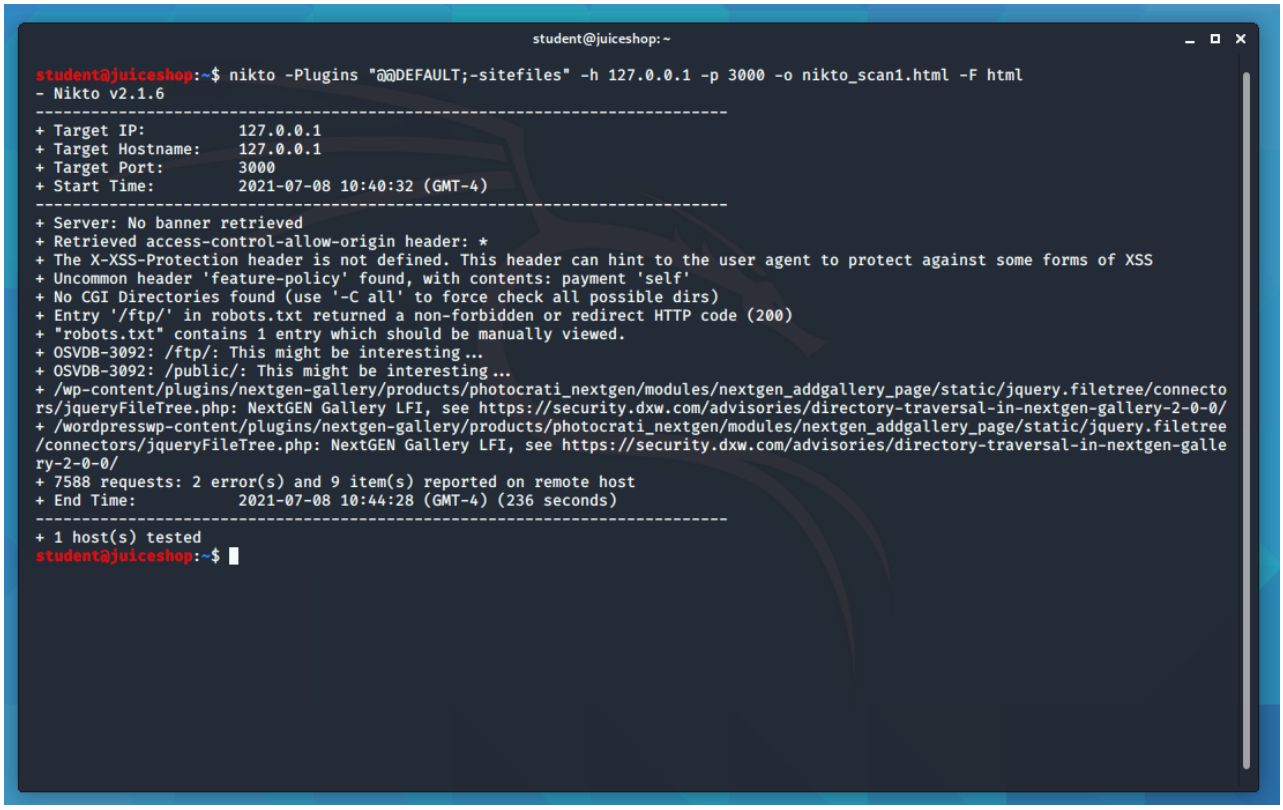


## ❖ This may be interesting …

**RISK LEVEL: Informational**

Vulnerability details:

- The Vulnerability has Informational risk.
- The response appears to contain suspicious comments which may help an attacker.
- The probability of exploit is insignificant.
- This won't affect anyone as this is just informational.
- A solution is to modify the Apache configuration.



# Methodology

## Assessment Toolset Selection

To perform a vulnerability assessment for the Udajuicer Development team we used:

- Virtual box – Environment setup
- Kali Linux - Platform
- OWAS ZAP – Web Application Vulnerability scanning tool
- Nikto – Web Application Vulnerability scanning tool

## Assessment Methodology Detail

First of all, we import into a VirtualBox an ova file provided by the Udajuicer Development team. We started the virtual machine, check for the application URL on the console screen after it is fully booted, and add port forwarding for the application.

We run the following command at a terminal in order to have the Juice Shop up and running:

**# bash startup.sh**

Then, we performed an Automated Scan with the web application vulnerability assessment OWASP ZAP at

http://localhost:3000

The result of the scan shows 5 alerts:

- Cross-Domain Misconfiguration (30)
- Cross-Domain JavaScript Source File Inclusion (11)
- Web Browser XSS Protection Not Enabled (6)
- Information Disclosure -Suspicious Comments (6)
- Timestamp Disclosure – Unix (23)

The Juice Shop web application was found to be vulnerable to a number of attacks related to Cross-Domain Misconfiguration. Web browser data loading may be possible, due to a Cross Origin Resource Sharing (CORS) misconfiguration on the web server.

A solution for this type of vulnerability is to ensure that sensitive data is not available in an unauthenticated manner (using IP address white-listing, for instance). Configure the 'Access-Control-Allow-Origin' HTTP header to a more restrictive set of domains, or remove all CORS headers entirely, to allow the web browser to enforce the Same Origin Policy (SOP) in a more restrictive manner.

According to https://www.zaproxy.org/docs/alerts:

**Alert ID**: 10098

**Alert type**: Passive Scan Rule

**Status**: Release

According to the MITRE organization:

**CWE-264: Permissions, Privileges, and Access Controls**

The application is vulnerable to a number of exploitable issues related to Cross-Domain JavaScript Source File Inclusion. The Cross-Domain JavaScript Source File Inclusion alert means that the given page includes and potentially runs one or multiple JavaScript files from a third-party domain.

If the external script location is not owned and managed by you there is a risk that the JavaScript file used by your application can be replaced with a malicious content that e.g. includes dangerous code or steals sensitive information/resources from your application users.

When some of your application JavaScript files are located on a third-party domain not managed by you the attacker may try to hijack that domain or access that third-party server to modify the files, so that your application will include a modified version that will be executed in web browsers of your users. This can be done without accessing your physical servers.

The solution for this vulnerability is to ensure JavaScript source files are loaded from only trusted sources, and the sources can't be controlled by end users of the application. The general principle is to always host all your application files on server locations managed by you or a publicly trusted and recognized third-party service, e.g. CDN.

According to https://www.zaproxy.org/docs/alerts:

**Alert ID**: 10017

**Alert type**: Passive Scan Rule

**Status**: Release

According to the MITRE organization:

**CWE-829: Inclusion of Functionality from Untrusted Control Sphere**

Many instances of Web Browser XSS Protection Not Enabled were identified. The Web Browser XSS Protection is not enabled, or is disabled by the configuration of the 'X-XSS-Protection' HTTP response header on the web server.

According to https://www.zaproxy.org/docs/alerts:

**Alert ID**: 10016

**Alert type**: Passive Scan Rule

**Status**: Deprecated since 2020-02-11(no longer widely supported by browsers)

According to the MITRE organization:

**CWE-933: OWASP Top Ten 2013 Category A5 – Security Misconfiguration**

Some Information Disclosure -Suspicious Comments were also identified. This term is frequently used in vulnerability advisories to describe a consequence or technical impact, for any vulnerability that has a loss of confidentiality.

According to the MITRE organization:

**CWE-200: Information Exposure**

Often, Information Exposure can be misused to represent the loss of confidentiality, even when the weakness is not directly related to the mishandling of the information itself. In addition, Information Exposure is also used frequently in policies and legal documents, but it does not refer to any disclosure of security-relevant information.

Many instances of Timestamp Disclosure – Unix were also identified. A timestamp was disclosed by the application/web server. A solution is to manually confirm that the timestamp data is not sensitive, and that the data cannot be aggregated to disclose exploitable patterns.

According to https://www.zaproxy.org/docs/alerts:

**Alert ID**: 10096

**Alert type**: Passive Scan Rule

**Status**: Release

We also run another web application vulnerability assessment tool call nikto by the following command:

**# nikto - h 127.0.0.1 -p 3000 -o nikto_scan1.html – Format html**

The scanner found 249 items but majority of them are false positives. For example, most of the items in the scan result refer to a vulnerable file in the application, like this:

```
Potentially interesting archive/cert file found.
Potentially interesting archive/cert file found. (NOTE: requested by IP address).
```

We were able to filter out the false positives by running the following command:

**# nikto -Plugins "@@DEFAULT;-sitefiles" - h 127.0.0.1 -p 3000 -o nikto_scan.html - Format html**

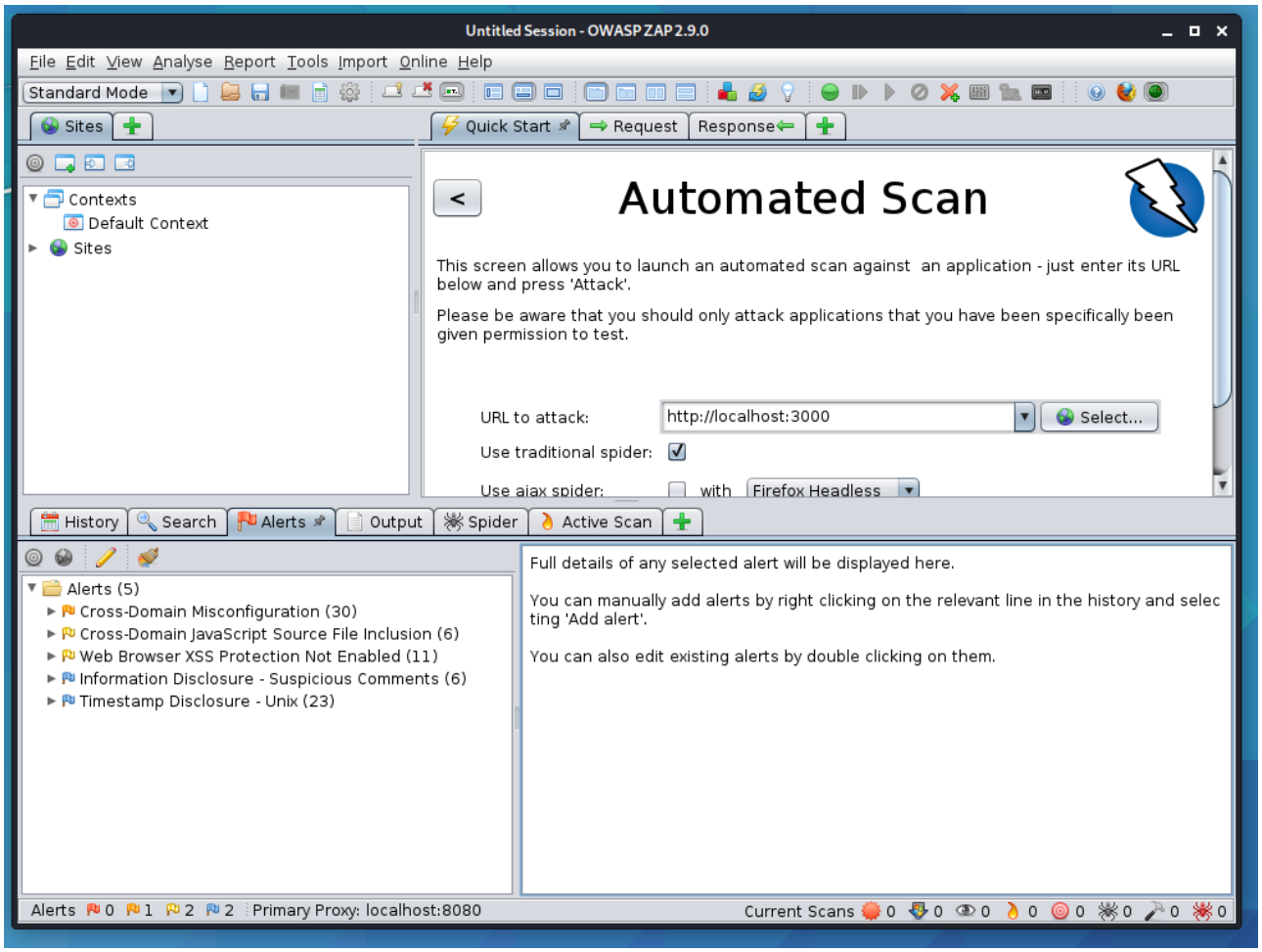
The nikto scanner now found 9 items which are pretty much close to the OWASP ZAP. We still did not find any high severity vulnerability but another vulnerability listed:

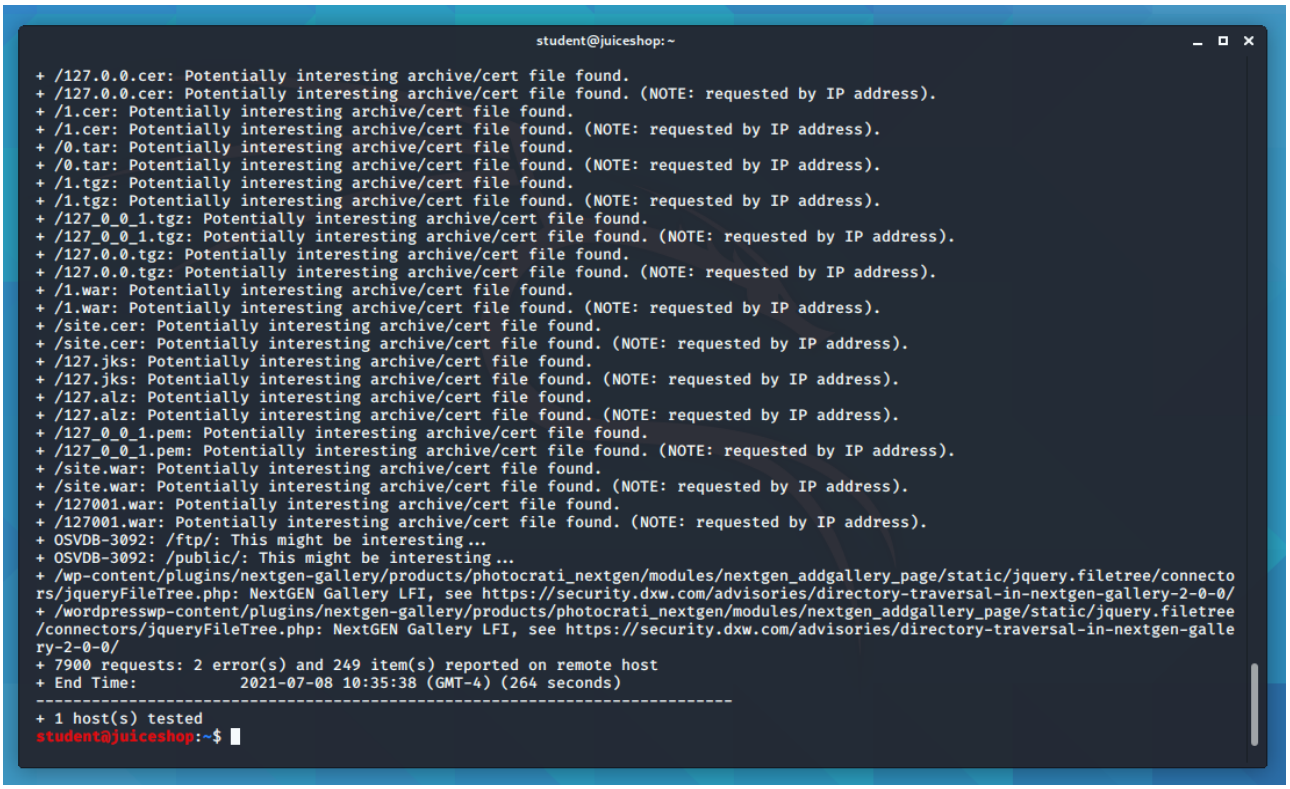
**OSVDB-3092: /ftp/: This might be interesting...**

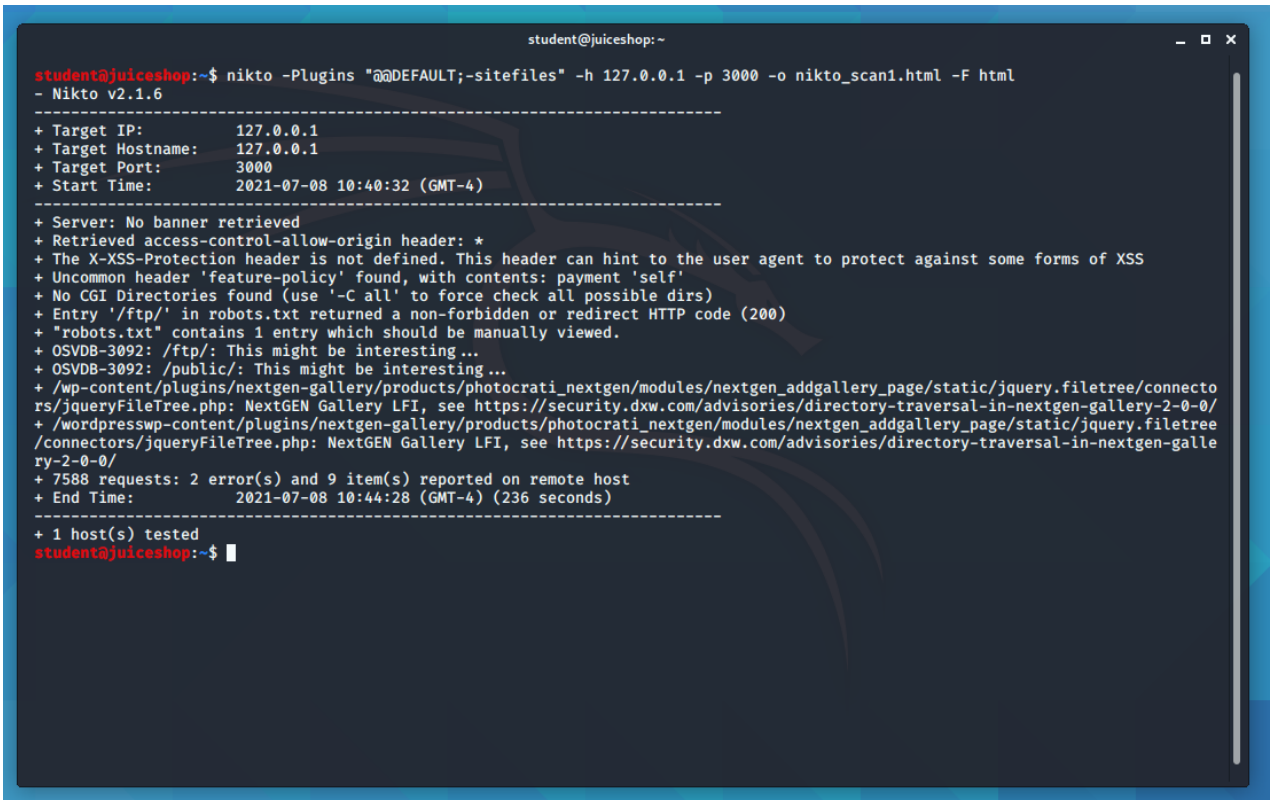
**OSVDB-3092: /public/: This might be interesting...**

This is considered to be a minor information disclosure vulnerability. To patch this vulnerability, we can modify the Apache configuration.

Below are the screen-shots of our findings.


Untitled Session - OWASP ZAP 2.9.0
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://localhost:3000
Select...
Use traditional spider:
Use ajax spider:
with
Firefox Headless
History
Search
Alerts
Output
Spider
Active Scan
Alerts (5)
Cross-Domain Misconfiguration (30)
Cross-Domain JavaScript Source File Inclusion (6)
Web Browser XSS Protection Not Enabled (11)
Information Disclosure - Suspicious Comments (6)
Timestamp Disclosure - Unix (23)
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.
Alerts 0 1 2 2 Primary Proxy: localhost:8080
Current Scans 0 0 0 0 0 0 0 0


student@juiceshop: ~
+ /127.0.0.cer: Potentially interesting archive/cert file found.
+ /127.0.0.cer: Potentially interesting archive/cert file found. (NOTE: requested by IP address).
+ /1.cer: Potentially interesting archive/cert file found.
+ /1.cer: Potentially interesting archive/cert file found. (NOTE: requested by IP address).
+ /0.tar: Potentially interesting archive/cert file found.
+ /0.tar: Potentially interesting archive/cert file found. (NOTE: requested by IP address).
+ /1.tgz: Potentially interesting archive/cert file found.
+ /1.tgz: Potentially interesting archive/cert file found. (NOTE: requested by IP address).
+ /127_0_0_1.tgz: Potentially interesting archive/cert file found.
+ /127_0_0_1.tgz: Potentially interesting archive/cert file found. (NOTE: requested by IP address).
+ /127.0.0.tgz: Potentially interesting archive/cert file found.
+ /127.0.0.tgz: Potentially interesting archive/cert file found. (NOTE: requested by IP address).
+ /1.war: Potentially interesting archive/cert file found.
+ /1.war: Potentially interesting archive/cert file found. (NOTE: requested by IP address).
+ /site.cer: Potentially interesting archive/cert file found.
+ /site.cer: Potentially interesting archive/cert file found. (NOTE: requested by IP address).
+ /127.jks: Potentially interesting archive/cert file found.
+ /127.jks: Potentially interesting archive/cert file found. (NOTE: requested by IP address).
+ /127.alz: Potentially interesting archive/cert file found.
+ /127.alz: Potentially interesting archive/cert file found. (NOTE: requested by IP address).
+ /127_0_0_1.pem: Potentially interesting archive/cert file found.
+ /127_0_0_1.pem: Potentially interesting archive/cert file found. (NOTE: requested by IP address).
+ /site.war: Potentially interesting archive/cert file found.
+ /site.war: Potentially interesting archive/cert file found. (NOTE: requested by IP address).
+ /127001.war: Potentially interesting archive/cert file found.
+ /127001.war: Potentially interesting archive/cert file found. (NOTE: requested by IP address).
+ OSVDB-3092: /ftp/: This might be interesting...
+ OSVDB-3092: /public/: This might be interesting...
+ /wp-content/plugins/nextgen-gallery/products/photocrati_nextgen/modules/nextgen_addgallery_page/static/jquery.filetree/connectors/jqueryFileTree.php: NextGEN Gallery LFI, see https://security.dxw.com/advisories/directory-traversal-in-nextgen-gallery-2-0-0/
+ /wordpresswp-content/plugins/nextgen-gallery/products/photocrati_nextgen/modules/nextgen_addgallery_page/static/jquery.filetree/connectors/jqueryFileTree.php: NextGEN Gallery LFI, see https://security.dxw.com/advisories/directory-traversal-in-nextgen-gallery-2-0-0/
+ 7900 requests: 2 error(s) and 249 item(s) reported on remote host
+ End Time: 2021-07-08 10:35:38 (GMT-4) (264 seconds)
+ 1 host(s) tested
student@juiceshop:~$


student@juiceshop: ~
student@juiceshop:~$ nikto -Plugins "@@DEFAULT;-sitefiles" -h 127.0.0.1 -p 3000 -o nikto_scan1.html -F html
- Nikto v2.1.6
+ Target IP: 127.0.0.1
+ Target Hostname: 127.0.0.1
+ Target Port: 3000
+ Start Time: 2021-07-08 10:40:32 (GMT-4)
+ Server: No banner retrieved
+ Retrieved access-control-allow-origin header: *
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ Uncommon header 'feature-policy' found, with contents: payment 'self'
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Entry '/ftp/' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ "robots.txt" contains 1 entry which should be manually viewed.
+ OSVDB-3092: /ftp/: This might be interesting...
+ OSVDB-3092: /public/: This might be interesting...
+ /wp-content/plugins/nextgen-gallery/products/photocrati_nextgen/modules/nextgen_addgallery_page/static/jquery.filetree/connectors/jqueryFileTree.php: NextGEN Gallery LFI, see https://security.dxw.com/advisories/directory-traversal-in-nextgen-gallery-2-0-0/
+ /wordpresswp-content/plugins/nextgen-gallery/products/photocrati_nextgen/modules/nextgen_addgallery_page/static/jquery.filetree/connectors/jqueryFileTree.php: NextGEN Gallery LFI, see https://security.dxw.com/advisories/directory-traversal-in-nextgen-gallery-2-0-0/
+ 7588 requests: 2 error(s) and 9 item(s) reported on remote host
+ End Time: 2021-07-08 10:44:28 (GMT-4) (236 seconds)
+ 1 host(s) tested
student@juiceshop:~$

**References:**

https://materials.rangeforce.com/tutorial/2019/12/05/Nikto/

https://www.zaproxy.org/docs/alerts/

https://www.netsparker.com

https://scanrepeat.com/

https://cwe.mitre.org/data/published/cwe_v2.6.pdf

https://cve.mitre.org/data/refs/refmap/source-OSVDB.html