2024

Report on the security of the College E-commerce site – Juice Shop

CSM060 INFORMATION SECURITY

ANDY DAVIS, STUDENT NUMBER 220491901

Table of Contents

List of Tables

Executive Summary

This report represents the results of the security review performed by the Lead Security Consultant from ACME Networks for the client Birkbeck College and their newly implemented e-commerce application to sell College insignia-branded merchandise through https://infosec.updrs.net:443. This assessment harnessed basic penetration testing techniques to provide an understanding of the risks and security of e-commerce site.

Assessment Scope

Testing was performed using industry-standard penetration testing tools and frameworks which include Open Web Application Security Project's Zed Attack Proxy (OWASP ZAP), SSL Lab, and Njsscan. Since the focus was only on one website, there was deep dive into other aspects of the client, such as scanning the wireless network, internal network, any other web applications, or any mobile applications the client may or may not have created. The testing began with using ZAP to look at any vulnerabilities found within the website. For the ease of reporting, False Positives have not been included in the scan report. ZAP reports helpfully provide links to the CWE site in which the details of these findings are provided with recommendations. After that, the site was scanned with Njsscan which showed cryptography risks and their references. Finally, SSL Lab was used to look at the certificates the site was using for the cryptography employed.

Summary of Findings

Table 1 shows the results from all the methods used to probe the site. ZAP returned with 11 alerts, Njsscan returned with 2 warnings, and SSL Lab returned with an overall rating of A. Table 2 shows the risk level of each alert returned from the ZAP report.

		Result	Type
Method	**ZAP**	11	Alerts
	Njsscan	2	Warnings
	SSL	A	Rating

Table 1: Summary of Findings

	High	Medium	Low	Informational	Total
Risk	1	3	3	4	11

Table 2: ZAP Alert Counts by Type

Summary of Recommendations

Based on the information detailed below, the following is a list of recommendations to provide a direction of improvement for securing the e-commerce site:

1. Address the Cloud Metadata Exposure issue urgently.
2. Implement Content Security Policy headers properly.
3. Investigate and remediate Cross-Domain Misconfigurations.
4. Replace MD5 usage with a stronger has function (such as SHA-256) within “Gruntfile.js”.
5. Replace the usage of ‘Math.random()’ with a cryptographically secure random number generator at the specified line numbers within “three.js”.
6. Consider implementing security features such as HSTS and OCSP stapling to enhance the overall security posture.

Relevant Policies and Procedures

The Birkbeck Policies and Procedures can be found here: https://www.bbk.ac.uk/about-us/policies/birkbeck-it-regulations.

The main two that are relevant are:

- Main policy: Birkbeck information security policy
- Supporting policy 1: Birkbeck data protection policy

Detailed Analysis

ZAP Alerts Overview

There are two types of criteria when looking into ZAP reports: Risk level and Confidence level. Risk level denotes how much of a risk the alert will cause if not solved, and confidence level denotes how strong the program believes it to be a threat. Only risk levels have been considered when looking at this report, which means that even though some alerts were raised, they may not actually have anything exposed, such as the case with the Timestamp Disclosure alert.

Table 3 shows how many alerts of each type were reported, next to their risk level. As a quick overview of solution recommendations, along with references to describe those solutions in detail, are shown on Table 4.

Alert Type	Risk	Count
Cloud Metadata Potentially Exposed	High	1

Content Security Policy (CSP) Header Not Set	Medium	548
CSP: Wildcard Directive	Medium	2
Cross-Domain Misconfiguration	Medium	565
Cross-Domain JavaScript Source File Inclusion	Low	1080
Strict-Transport-Security Header Not Set	Low	563
Timestamp Disclosure - Unix	Low	1
Information Disclosure - Suspicious Comments	Informational	2
Modern Web Application	Informational	541
Re-examine Cache-control Directives	Informational	43
User Agent Fuzzer	Informational	1249

Table 3: ZAP Alert Counts by Alert

Alert Type	Recommendation	Reference
Cloud Metadata Potentially Exposed	Do not trust any user data in NGINX configs.	https://www.nginx.com/blog/trust-no-one-perils-of-trusting-user-input/
Content Security Policy (CSP) Header Not Set	Ensure that your web server, application server, load balancer, etc. is	https://developer.mozilla.org/en-US/docs/Web/Security/CSP/Introducing_Content_Security_Policy https://cheatsheetseries.owasp.org/

	properly configured to set the CSP header.	cheatsheets/Content_Security_Policy_Cheat_Sheet.html http://www.w3.org/TR/CSP/
CSP: Wildcard Directive	Ensure that your web server, application server, load balancer, etc. is properly configured to set the CSP header.	http://www.w3.org/TR/CSP2/ http://www.w3.org/TR/CSP/ http://caniuse.com/#search=content+security+policy
Cross-Domain Misconfiguration	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	https://vulncat.fortify.com/en/detail?id=desc.config.dotnet.html5_overly_permissive_cors_policy
Cross-Domain JavaScript Source File Inclusion	Ensure JavaScript source files are loaded from only trusted sources, and the sources can't be controlled by end users of the application.	
Strict-Transport-Security Header Not Set	Ensure that your web server, application server, load balancer, etc. is configured to enforce Strict-Transport-Security.	https://cheatsheetseries.owasp.org/cheatsheets/HTTP_Strict_Transport_Security_Cheat_Sheet.html https://owasp.org/www-community/Security_Headers http://en.wikipedia.org/wiki/HTTP_Strict_Transport_Security http://caniuse.com/stricttransportsecurity http://tools.ietf.org/html/rfc6797
Timestamp Disclosure - Unix	Manually confirm that the timestamp data is not sensitive, and that the data cannot be aggregated to disclose exploitable patterns.	http://projects.webappsec.org/w/page/13246936/Information%20Leakage
Information Disclosure -	Remove all comments that return information that may help an attacker and fix any	

Suspicious Comments	underlying problems they refer to.	
Modern Web Application	This is an informational alert and so no changes are required.	
Re-examine Cache-control Directives	For secure content, ensure the cache-control HTTP header is set with "no-cache, no-store, must-revalidate". If an asset should be cached consider setting the directives "public, max-age, immutable".	https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html#web-content-caching https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Cache-Control https://grayduck.mn/2021/09/13/cache-control-recommendations/
User Agent Fuzzer		https://owasp.org/wstg

Table 4: ZAP Alert Recommendations

ZAP Alerts Details

Cloud Metadata Potentially Exposed

Description:

The Cloud Metadata Attack attempts to abuse a misconfigured NGINX server to access the instance metadata maintained by cloud service providers such as AWS, GCP and Azure. These providers provide metadata via an internal unrouteable IP address '169.254.169.254' - this can be exposed by incorrectly configured NGINX servers and accessed by using this IP address in the Host header field.

Other Information:

Metadata may have been included in the response, which is what triggered the alert. Check the response to make sure no metadata was included and/or there was no information an attacker can use to compromise the system.

Affected Response:

GET https://infosec.updrs.net:443/latest/meta-data/

OWASP Tag(s):

OWASP_2017_A06

OWASP_2021_A05

Content Security Policy (CSP) Header Not Set

Description:

Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content that browsers should be allowed to load on that page — covered types are JavaScript, CSS, HTML frames, fonts, images, and embeddable objects such as Java applets, ActiveX, audio, and video files.

Affected Response:

GET https://infosec.updrs.net

CWE ID:

CWE-693: Protection Mechanism Failure

OWASP Tag(s):

OWASP_2017_A06

OWASP_2021_A05

CSP: Wildcard Directive

Description:

A different type of alert from the CSP (mentioned above). This one comes from the GET response of "Content-Security-Policy: default-src 'none'".

Other Information:

The directives "form-action" and "frame-ancestors" either will allow wildcard sources, are not defined, or are broadly defined.

Affected Response:

GET https://infosec.updrs.net/assets/public

CWE ID:

CWE-693: Protection Mechanism Failure

OWASP Tag(s):

OWASP_2017_A06

OWASP_2021_A05

Cross-Domain Misconfiguration

Description:

Web browser data loading may be possible, due to a Cross Origin Resource Sharing (CORS) misconfiguration on the web server.

Other Information:

There may be some unauthorized APIs used on the site. This may cause an attacker to access data in an unauthenticated way which could use a different form of security.

Affected Response:

GET https://infosec.updrs.net/assets/public/favicon_js.ico

CWE ID:

CWE-264: Permissions, Privileges, and Access Controls

OWASP Tag(s):

OWASP_2017_A05

OWASP_2021_A01

Cross-Domain JavaScript Source File Inclusion

Description:

The page includes one or more script files from a third-party domain.

Affect Response:

GET https://infosec.updrs.net/sitemap.xml

CWE ID:

CWE-829: Inclusion of Functionality from Untrusted Control Sphere

OWASP Tag(s):

OWASP_2021_A08

Code Snippet:

```
<meta charset="utf-8">
<title>OWASP Juice Shop</title>
<meta name="description" content="Probably the most modern and sophisticated insecure
web application">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link id="favicon" rel="icon" type="image/x-icon"
href="assets/public/favicon_js.ico">
<link rel="stylesheet" type="text/css"
href="//cdnjs.cloudflare.com/ajax/libs/cookieconsent2/3.1.0/cookieconsent.min.css">
<script
src="//cdnjs.cloudflare.com/ajax/libs/cookieconsent2/3.1.0/cookieconsent.min.js"></sc
ript>
<script src="//cdnjs.cloudflare.com/ajax/libs/jquery/2.2.4/jquery.min.js"></script>
```

Strict-Transport-Security Header Not Set

Description:

HTTP Strict Transport Security (HSTS) is a web security policy mechanism whereby a web server declares that complying user agents (such as a web browser) are to interact with it using only secure HTTPS connections (i.e., HTTP layered over TLS/SSL). HSTS is an IETF standards track protocol and is specified in RFC 6797.

Affect Response:

GET https://infosec.updrs.net/assets/public/favicon_js.ico

CWE ID:

CWE-319: Cleartext Transmission of Sensitive Information

OWASP Tag(s):
OWASP_2017_A06

OWASP_2021_A05

Timestamp Disclosure – Unix

Description:

A timestamp was disclosed by the application/web server – Unix.

Other Information:

The Unix code 1734944650 was returned, which represents 2024-12-23 01:04:10. It appears in a link to a Google Form for a Coding Challenge, thus does not seem like it is a threat.

Affected Response:

GET https://infosec.updrs.net/main.js

Code Snippet:

```
Q6J("href","https://docs.google.com/forms/d/e/1FAIpQLSdaNEuz0dzFA2sexCa0AJ4QOb2OYdEL0
4eQOLFD2Y4T-
BW6ag/viewform?usp=pp_url&entry.384948954="+e.dialogData.name+"&entry.435235279=Codin
g+Challenge&entry.1734944650=No",t.LSH)}
```

CWE ID:

CWE-200: Exposure of Sensitive Information to an Unauthorized Actor

OWASP Tag(s):

OWASP_2017_A03

OWASP_2021_A01

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 Information:

The pattern “query” was found within the files. Comments can be used to SQL inject into the database using that pattern and find information not meant to be shared.

Affected Response:

GET https://infosec.updrs.net/main.js

CWE ID:

CWE-200: Exposure of Sensitive Information to an Unauthorized Actor

OWASP Tag(s):

OWASP_2017_A03

OWASP_2021_A01

WSTG-v42-INFO-05

Modern Web Application

Description:

The application appears to be a modern web application. Nothing else to report about it, just an informational alert.

Affected Response:

GET https://infosec.updrs.net

CWE ID:

None

OWASP Tag(s):

None

Re-examine Cache-control Directives

Description:

The cache-control header has not been set properly or is missing, allowing the browser and proxies to cache content. For static assets like CSS, JS, or image files this might be intended, however, the resources should be reviewed to ensure that no sensitive content will be cached.

Affected Response:

GET https://infosec.updrs.net/robots.txt

CWE ID:

CWE-525: Use of Web Browser Cache Containing Sensitive Information

OWASP Tag(s):

WSTG-v42-ATHN-06

User Agent Fuzzer

Description:

Check for differences in response based on fuzzed User Agent (e.g., mobile sites, access as a Search Engine Crawler). Compares the response status code and the hash code of the response body with the original response.

Affected Response:

GET https://infosec.updrs.net/assets

CWE ID:

None

OWASP Tag(s):

None

Njsscan Warnings

Rule ID: node_md5

Description:

MD5 is a weak hash function which is known to have collisions. Use a strong hash function.

CWE ID:

CWE-327: Use of a broken or Risky Cryptographic Algorithm

Affected File(s):

BBKJuiceShop/bbk-shop-njsscan/juice-master-2/Gruntfile.js

Line Number(s):

73

Match String:

const md5 = crypto.createHash('md5')

Rule ID: node_insecure_random_generator

Description:

Math.random() is a cryptographically weak random number generator.

CWE ID:

CWE-327: Use of a broken or Risky Cryptographic Algorithm

Affected File(s):

BBKJuiceShop/bbk-shop-njsscan/juice-master-2/frontend/src/assets/private/three.js

Line Number(s):

6359, 6426, 6434, 6442, 6450, 15504, 15534, 15567

Match String:

Any instance of Math.random()

Recommendations/Remediations:

The issue is that whoever created the site decided to use a weak cryptography method to storing sensitive information. Also, Math.random() is not truly a random number generator and is not recommended for use with cryptography. It is recommended to use a stronger hash function (such as

SHA-256). On top of that, the site designer and/or maintainers should study up on stronger forms of cryptography, or even cryptography in general if they do not have any previous knowledge on the subject.

SSL Labs Results

There really isn't much to say on this since the results came back very positive. Certificate and Protocol Support came back 100% and Key Exchange and Cipher Strength came back 90%. The DROWN test did not get performed due to an internal error, but that is issue known with the SSL Labs site (see the blog post here). I was also a little confused as to what that part of the assignment was really doing. I know it must do something with the certificates on the site (http vs https), but it was confusing this whole time as to which one to type in when doing the coursework (conflicting answers in the brief vs announcements/forum posts). I think I did the https version, which is inherently stronger than http and would explain the A rating.

References

Most reference links are included in the appropriate sections. I wish I had a good way to aggregate them, but it might get a bit confusing as to where they belong and what they are referencing.

Not quite sure if this is properly made, I was just following the example reports provided and mimicked their style. They did not have a final Reference page like book reports (or other assignments) normally do, so I figured at least having the proper links in the proper locations will do for references.

Final Thoughts

I am not the best at writing reports, and it seems like this assessment requires a ton of training prior that the module just does not provide (very base level topics). I am sure the TurnItIn score might be a little high, but that is all just the descriptions and everything in the tables copied over from the ZAP report, which I am sure most to all people will have. Same with the section headers. Honestly, my attitude towards everything has really soured since I have started last year due to false accusations of plagiarism on another module (which made me retake the class, after trying to appeal as well as submitting a complaint). It is tough for me to fully commit because now I am worried if that course gets a zero again then all of this will be for nothing, and I will have to end my journey and not finish getting my Master's. This thought has really been weighting on me hard and wrecking me mentally, but I am still trying the best I can to finish this out strong (or at least get a passing grade on the final modules). This one may seem light to everyone, but there was no minimum requirement, only max of 3000 words, so I think this will be good enough for me for now. It has been a rough two weeks, and this is a very poor effort from me which I apologize for, but at least you know why it is very poor.

Final Word Count

1981