EzMealz Security Assessment

Oregon State University

March 2023

Prepared by:

Dam Security

Table of Contents

Report Introduction

This Report will outline

1. The description of EzMealz
2. Users of the application
3. Document Owner, participants of the threat modeling process, and reviewer
4. External dependencies
5. Entry and exit points of EzMealz
6. Assets of the application
7. Trust levels
8. Data flow diagrams
9. Threat Analysis
10. Definition of Risk
11. List of Threats
12. Threat Mitigation and Techniques

EzMealz Threat Model Information

Application Version: 1.0

Description: EzMealz is a web application to provide recipe links that have been verified to be easy to cook and are beginner friendly. Many online recipes that claim to be quick and easy take longer than expected or are more difficult than anticipated. EzMealz aims to provide recipes that have been verified to truly be time-efficient and easy to make. There will be two users:

1. External users
2. Internal application administrators

EzMealz External Dependencies

ID	Description
1	EzMealz runs on an Azure Web App service instance on a Linux cloud server. The cloud server adheres to the Azure security protocols and standards, such as Microsoft Azure TLS certificate issues and activity monitoring and health checks.
2	The database storing all information is MongoDB. It is a document-based, non-SQL open-source database. The database adheres to MongoDB's standards such as encryption, authentication, and access control.
3	The connection between the web application and the MongoDB server is done through the Node.js driver named mongodb. The connection string from the database is used to establish the connection and make and manipulate schemas.
4	Payment processing for subscription service in EzMealz will be done through PayPal.
5	Shipping of merchandise will be done through FedEx.
6	Source code is stored in a GitHub repository.

EzMealz Entry Points

ID	Name	Description	Trust Levels
1	HTTP Port	An entry port into the web application. Accessing through HTTP is not secure and not recommended, but it is currently available.	(1) Anonymous Web User (2) User with invalid login credentials (3) User with valid login credentials (4) Web application administrator
1.1	HTTPS Port	Secure entry point into the web application. A future improvement to the web application will include exclusive access to EzMealz through HTTPS only.	(1) Anonymous Web User (2) User with invalid login credentials (3) User with valid login credentials (4) Web application administrator
1.3	EzMealz Home Page	The home page for EzMealz and the entry point for all users of the website.	(1) Anonymous Web User (2) User with invalid login credentials (3) User with valid login credentials (4) Web application administrator
1.3.1	Search Bar	Allows the user to search for a recipe based on keywords. Ex: “chicken” will return recipes that have chicken.	(1) Anonymous Web User (2) User with valid login credentials (3) User with invalid login credentials (4) Web application administrator
1.4	Login Page	Page for users to log in to their account. Logging in is not mandatory to use the website, but future implementations will allow the user to save recipes.	(1) Anonymous Web User (2) User with invalid login credentials (3) User with valid login credentials (4) Web application administrator

1.4.1	Login Function	Takes in user-supplied input and compares it with information in the database to verify if the user has an account.	(2) User with invalid login credentials (3) User with valid login credentials (4) Web application administrator
1.5	Sign Up Page	Page to create an account. The future release of EzMealz will allow people with accounts to save recipes.	(1) Anonymous Web User (2) User with invalid login credentials (3) User with valid login credentials (4) Web application administrator
1.5.1	Sign Up Function	Takes in user input to create an account and save it to the database.	(2) User with invalid login credentials (3) User with valid login credentials (4) Web application administrator
1.6	Subscribe Page	Page to subscribe to EzMealz that gives the user perks such as ad-free browsing, monthly merchandise, and exclusive recipes sent to their email address.	(1) Anonymous Web User (2) User with invalid login credentials (3) User with valid login credentials (4) Web application administrator
1.6.1	Subscribe function	Takes in user input for subscription payment and saves it to the database.	(1) Anonymous Web User (2) User with invalid login credentials (3) User with valid login credentials (4) Web application administrator
1.7	View Subscribers Page	Allows the administrator to view subscribers and their information.	(4) Web application administrator (5) Database administrator

EzMealz Assets

ID	Name	Description	Trust Levels
1	**EzMealz Users and Administrators**	**Assets relating to EzMealz users and administrators**	
1.1	User login credentials	The username and password that EzMealz users and administrators will use to log into the application.	(3) User with valid login credentials (4) Web application administrator (5) Database administrator
1.2	Subscriber Information	Personal information of EzMealz subscribers. Information includes name, physical address, email address, and credit card information.	(4) Web application administrator (5) Database administrator
2	**System**	**Assets relating to the EzMealz system functionality**	
2.1	Availability of EzMealz	EzMealz should be available 24 hours, seven days a week unless maintenance is being done on the application.	(4) Web application administrator (5) Database administrator
2.2	Ability to change source code to the website	The ability to change the source code to the website and change the structure or look of the web application.	(4) Web application administrator
2.3	Ability to modify GitHub repository and make push/pull requests	The ability to change code through GitHub and deploy the code through GitHub Actions.	(4) Web application administrator
2.4	Ability to modify the MongoDB Database	The ability to change the contents of the database. This includes adding and deleting.	(5) Database administrators
3	**Website**	**Assets relating to the EzMealz Website**	
3.1	Login Session	Login session of the EzMealz user. This includes regular users and administrators.	(3) User with valid login credentials (4) Web application

			administrator
3.2	MongoDB Database Access	Access to the MongoDB database has full ability to create, read, update, and delete contents from the database.	(5) Database administrator
3.3	Ability to create users	Ability to create a profile on EzMealz.	(1) Anonymous Web User (2) User with invalid login credentials (3) User with valid login credentials
3.4	Ability to subscribe to EzMealz	Ability to subscribe to EzMealz for exclusive content and monthly merchandise.	(1) Anonymous Web User (2) User with invalid login credentials (3) User with valid login credentials

Trust Levels

ID	Name	Description
1	Anonymous Web User	The user who is browsing EzMealz but does not have a profile or is subscribed to the web application.
2	User with Invalid Login Credentials	The user who is on EzMealz and is trying to log into the website using invalid credentials.
3	User with Valid Login Credentials	The user who is browsing EzMealz and is logged into the website with valid credentials.
4	Web Application Administrator	The administrator of EzMealz who can change contents and view subscriber information.
5	Database Administrator	The administrator of the EzMealz MongoDB database who has read and write access.

Data Flow Diagrams

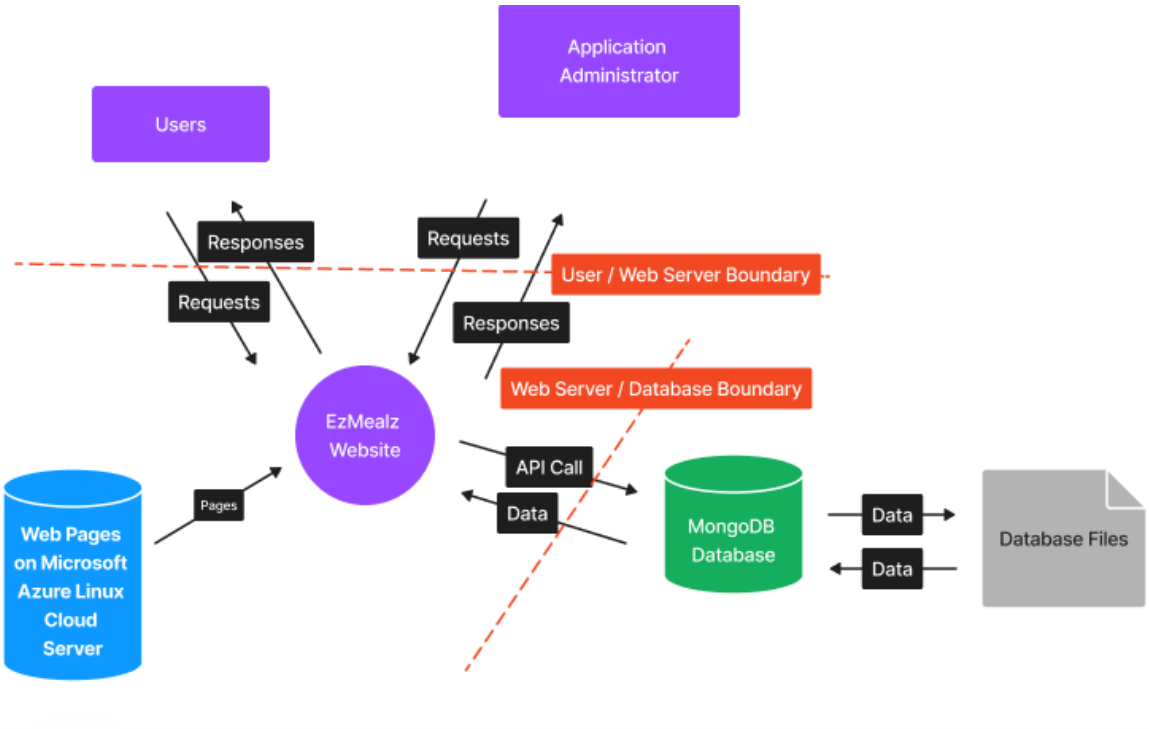

Figure 1: High-Level data flow diagram of EzMealz.

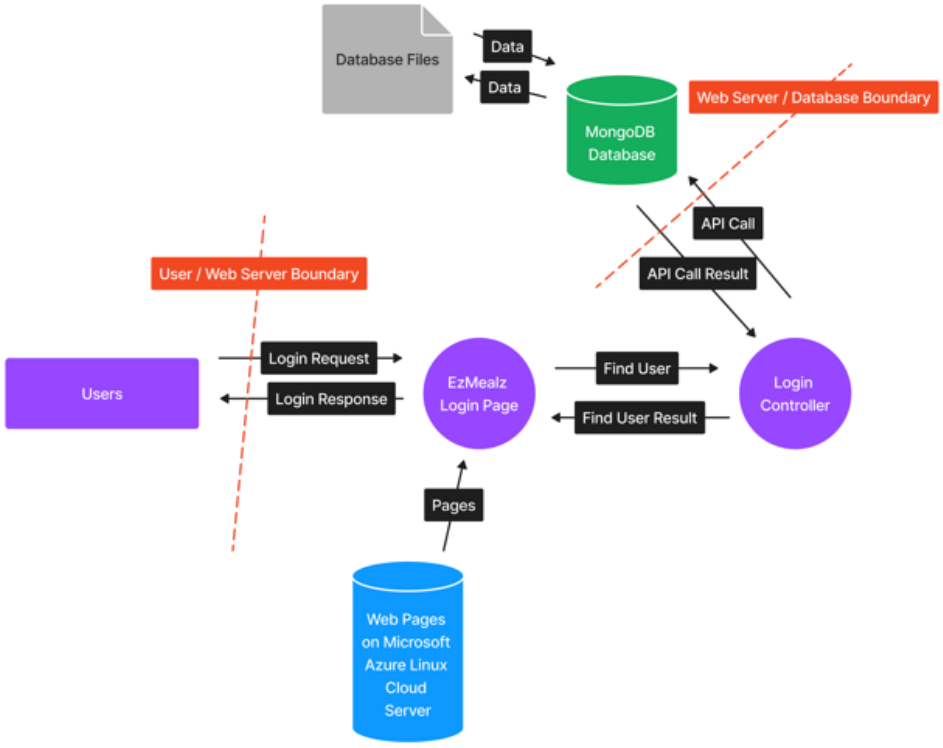

Figure 2: Data flow diagram of login page.

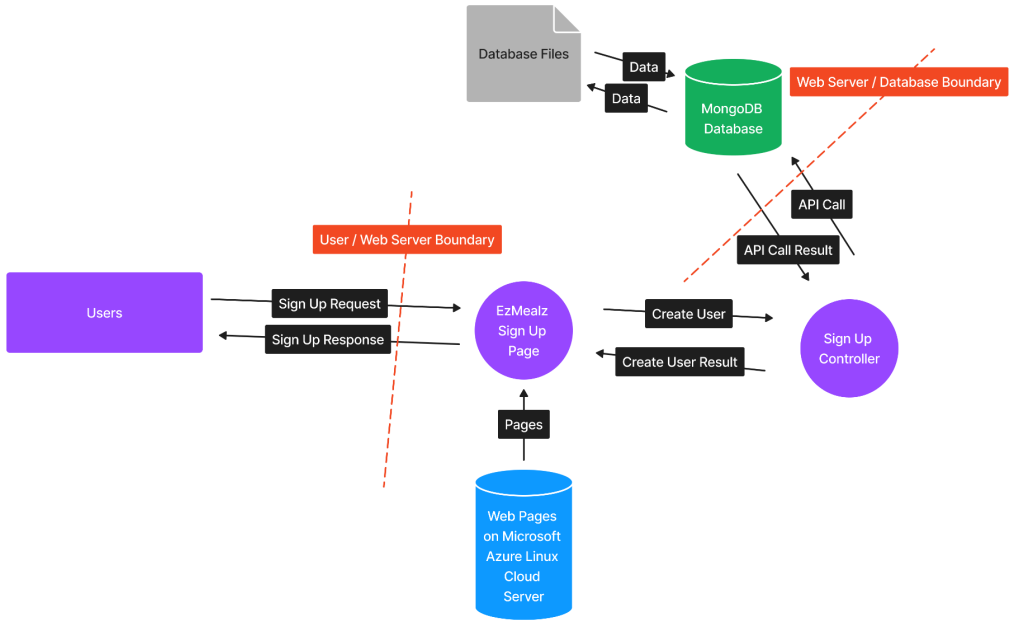

Figure 3: Data flow diagram of sign-up page.

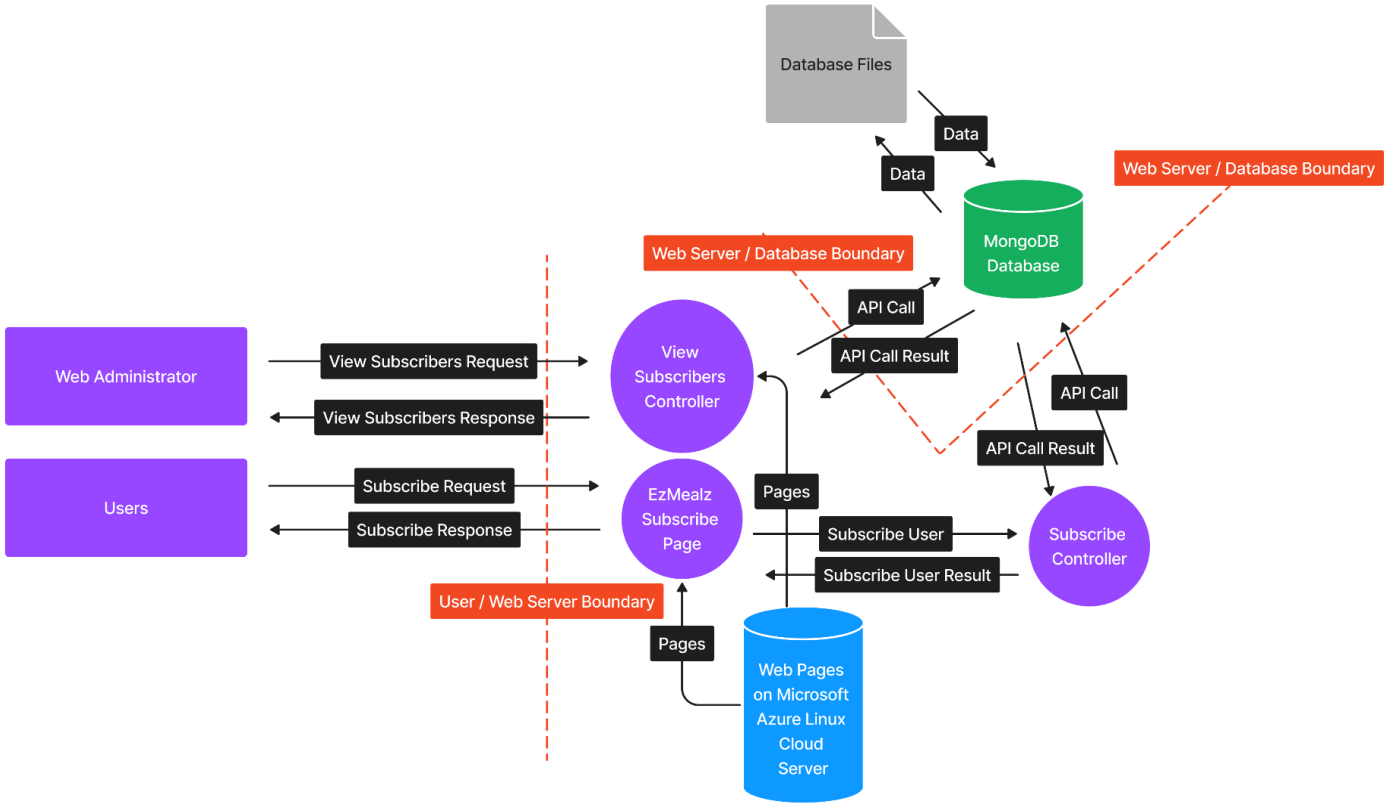

Figure 4: Data flow diagram of subscriber function.

Threat Analysis

Threat Categorization - STRIDE

- Spoofing - When a malicious actor attempts to use other users' credentials to act on their behalf. Having a strong authentication protocol will minimize spoofing.
- Tampering - When a malicious actor attempts to modify data in a database or in between sending and receiving data over the network. Integrity security measures will minimize tampering.
- Repudiation - Performing untraceable, unauthorized actions to a system. Non-repudiation security measures will minimize this threat. All actions must be associated with a unique user and logged.
- Information Disclosure - When someone attempts to read a file that they have not been granted permission to access or attempts to read data in transit. Proper access controls, and making sure not to hardcode sensitive data will help reduce the chance of a threat.
- Denial of Service - Creating disturbance and inability to access a service, such as rendering a webpage unavailable. Security measures to maintain the availability of the website will minimize the threat.
- Elevation of Privilege - Trying to gain administrative access for the purpose of seeing information or crashing the system.

Definition of Risk

A generic definition of the risk of a threat is as follows:

(Probability that threat occurs) x (Cost to organization)

The probability that any threat could occur is between 0 and 1. Some threats have infinitesimal odds of occurring, so their probabilities are essentially 0. The likelihood of a threat could increase over time, so we will use a one year timeframe to determine the rough probabilities of each threat.

The cost to an organization includes monetary, reputational and infrastructure losses. We will be ranking costs of each threat from very low to very high.

List of Threats

Spoofing

1. Website and/or URL spoofing

   Assets at risk: user's personal information (login, credit card, address, email)

   How threat occurs: Attackers impersonate EzMealz's email newsletters or their website by cloning their website and getting it higher up in Google search results. Next, users click the fake link in the email or on Google and may be prompted to enter login or payment information

2. Man-in-the-middle attacks

   Assets at risk: (Login credentials, credit card information, address information, email, IP address)

   How threat occurs: An adversary intercepts the connection between the user and EzMealz to steal data being transferred to and from the user and the website. The

attacker can either redirect traffic to a different host or record information and redirect traffic back to the original destination.

3. IP spoofing

   Assets at risk: user's personal information (credit card, address, email, login credentials), system functionality.

   How threat occurs: The attacker will change their IP address in the packet headers so that the destination host will validate the incoming traffic and accept the packets. This allows the attacker to conceal their identity and bypass firewalls. Some attacks that involve IP spoofing are DDos attacks, Man in the Middle attacks, and Botnet attacks.

Tampering

4. Cookies

   Assets at risk: user login info and session tokens (can be used to gain access to user's account)

   How threat occurs: Cookie tampering can be done in several different ways -

   - Cookies can be manipulated directly through the browser from the developer tools. The client-side cookies are manipulated with this technique, and an attacker can change the cookies from a site and send that URL to a target. The attacker can then take the response from the server and gather information about the target.
   - An attacker can send malicious data to a vulnerable web application so that the application's responses will contain a harmful payload to clients waiting for a response.

- Cookies can be manipulated by reordering small bits of data to see if any results appear. Changing key-value pairs and adding duplicate keys can also bypass validation.

5. HTML Form Fields

   Assets at risk: users' purchased merchandise and newsletter subscriptions

   How threat occurs: The attacker can edit the values entered for payment, address, and email. They could set the email and physical addresses to ones accessible to them so they could steal the merchandise and newsletter subscriptions

6. URL Query Strings

   Assets at risk: payment information, purchased merchandise

   How threat occurs: An adversary can change the information in the parameters of communication between client and server. This can change the data in the application such as user permissions and credentials. Parameter modifications can be done in several ways, including -

   - Manipulating input in form fields
   - Changing the HTML parameters on hidden items to change the value,
   - Manipulating the URL parameters directly by changing specific parameters in the URL. If a website renders certain pages or performs a particular action based on the URL parameters, the parameters can be changed to perform an unauthorized action.

7. HTTP Headers

   Assets at risk: user's personal information (credit card, address, email, login credentials), system functionality.

   How threat occurs: Poor configurations when resolving domain names or unexpected input in the HTTP headers. Malicious code can be included within the HTTP headers that can lead to an attacker taking over the system.

8. Password Cracking

   Assets at risk: user's personal information (credit card, address, email)

   How threat occurs: weak or lacking guidelines for users' passwords

Repudiation

9. Non-repudiation is a legal requirement for transactions on a website. A user can claim repudiation of origin, submission, delivery, or receipt, and if there is no evidence in the shop's favor, the customer is legally entitled to a refund.
   a. repudiation of origin: customer denies that this order was theirs
   b. repudiation of submission: customer denies ordering anything
   c. repudiation of delivery: customer denies receiving the goods
   d. repudiation of receipt: customer claims not getting the correct goods in their order
10. Non-repudiation often uses TTP (time stamping) and has a major vulnerability. The time key and clock could be used to crash the system.

Information Disclosure

11. Without proper access controls, an attacker could find sensitive information with basic testing. Misconfigurations, hardcoded sensitive data and published source code could also open a website up to this threat.

Denial of Service

12. Flooding the server is one type of DoS attack, after which users and admins will be unable to access the server. DoS attacks generally target higher-profile businesses, so this threat would have a low probability.
13. Exploiting particular vulnerabilities that use more of the server's resources is another way to conduct a DoS attack.

Elevation of Privilege

Assets at risk: user's login info (particularly admins) and personal information (name, credit card, address, email)

14. Credential Exploitation

    How threat occurs: The attacker gains access to the user's login info or username from another source or attack. With the username, they can crack the user's password. This is particularly problematic if they get access to an admin account.

15. Misconfiguration

    How threat occurs: Having poorly configured security such as blank or weak passwords for administrators. Unknown backdoor access.

    Assets at risk: user's login info (particularly admins) and personal information (name, credit card, address, email). System functionality.

16. Malware

How threat occurs: Malware is loaded onto the system through a variety of different ways, such as social engineering, vulnerabilities, exploits, or supply chain weaknesses. The malware can then do various functions to elevate privileges, such as logging keystrokes to gather user credential information or gather password information from memory.

Assets at risk: user's login info (particularly admins) and personal information (name, credit card, address, email). System functionality.

Severity						
	Very Severe 5	Medium	Medium High 10	High	Very High 14,16	Very High 3,7,11,15
	Severe 4	Low 2	Medium	Medium High	High 5	Very High 1,6
	Moderate 3	Low	Medium 12,13	Medium 8	Medium High 4	High
	Minor 2	Low	Low	Medium 9	Medium	Medium High
	Negligible 1	Low	Low	Low	Low	Medium
		Rare 1	Unlikely 2	Possible 3	Likely 4	Highly Likely 5
Probability						

Figure 5: Table showing the probability and severity of the 16 aforementioned risks.

Threat and Mitigation Techniques

Threat	Mitigation Techniques
Spoofing	1. Maintain a good password protocol 2. Hash and salt passwords 3. Implement Two-Factor Authentication 4. Maintain good monitoring protocol for suspicious network activity 5. Packet filtering to block packets with an invalid source address information 6. Use protocols with encryption such as HTTPS or SSH 7. Install firewalls on Network 8. Access Control List to block private IP addresses
Tampering	1. Validate Input so that invalid characters are denied. Have a list of allowed regular expressions that exclude the following characters: & | ; $ > < \ \ ! 2. Separate the data from commands in parameterization 3. Encrypt sensitive data-at-rest and data-in-transit. Use HTTPS and SSH to protect data-in-transit. 4. Ensure development frameworks and dependencies are up to date 5. Encode output when user input is displayed as output 6. Utilize JavaScript methods such as .textContent() to treat user input as text
Repudiation	1. Monitor user activity through logs 2. Have strong authentication measures such as good password hygiene 3. Encrypt sensitive data, both data-at-rest and data-in-transit
Information Disclosure	1. Implement strong authorization 2. Have robust encryption protocols for data

	3. Protect sensitive information 4. Avoid using specific error messages 5. Disable debugging or diagnostic features in the production build
Denial of Service	1. Avoid using unnecessary ports for communication 2. Make use of Content Distribution Networks or Load Balancers 3. Make use of firewalls, and Access Control Lists to control traffic reaching the website
Elevation of Privilege	1. Run the application with the Principle of Least Privilege 2. Secure user credentials, especially for the web administer

Vulnerability Assessment

Using Skipfish

Vulnerabilities Found -

- Incorrect Caching Directives (Higher Risk):
    - The set.cookie response in the HTTP header is implicitly cached, making the web application vulnerable to possible cache poisoning attacks. The lack of instructions for how to handle the cache can make the web application vulnerable to other attacks, such as cross-site scripting attacks.
        - Users who visit the site can be injected with malicious code stored in the cache (Stored Cross Site Scripting)
- External Content Embedded on a Page (Higher Risk) -
    - The application utilizes the Bootstrap library for web application styling. This can pose a problem if Bootstrap becomes compromised and malicious code is inserted into EzMealz through Bootstrap.
- XSS Vector in Document Body (Higher Risk) -
    - Skipfish was able to inject a <sfi000151v888863> tag within a GET request for a recipe. The response from the server was 500 (Internal Server Error), but a more malicious injection could yield devastating results.
- Incorrect Caching Directives (lower risk) -
    - The Cache-Control directive for some responses was set to “public, max-age=0” meaning that no cache was saved. This gives conflicting instructions to save responses to a shared cache despite the cache not being saved.

- HTML Form with No Apparent XSRF Protection (lower risk) -
    - The search field in EzMealz was flagged to have no XSRF (cross-site request forgery) protection. This means that an authenticated user can become vulnerable to making unauthorized requests to the web application if they fall prey to a trap such as phishing. This can be more devastating if the user is the administrator.
- Directory Listing Restrictions Bypassed (lower risk) -
    - The vulnerability scanner was able to get a valid response after sending a request with an invalid directory listing. This shows the lack of restrictions for directory listings that can be used to access unauthorized directories.

Using Nikto

Vulnerabilities found -

- The anti-clickjacking X-Frame-Options header is not present
    - There are no configurations in the x-frame header of the HTTP requests, which can render EzMealz vulnerable to clickjacking. Content-security-policy has a frame-ancestors header which can make this header useless, but it is still safe to have x-frame options on.
- The site uses TLS, and the Strict-Transport-Security HTTP header is not defined
    - The site can be accessed through both HTTPS and HTTP. Accessing the website through HTTP can be dangerous if an adversary is sniffing incoming and outgoing traffic, as they can read the unencrypted data being sent back and forth.
    - Configuring the “strict-transport-security” header option will make it so that the site can only be accessed through an HTTPS connection.
- The X-Content-Type-Options header is not set -
    - By not having the x-content-type-option header set, the browser can render the content in a different fashion than it was intended. This can cause a cross-site-scripting vulnerability if the website allows users to submit files such as a photo, and if another user clicks on that file, malicious code can be executed.
- Cookie connect.sid created without the secure flag -
    - There is no secure flag for the cookie headers. Therefore, an adversary may intercept sensitive information stored in the cookies.
- The server is using a wildcard certificate: *.azurewebsites.net -

- The vulnerability arises when one server that is hosting a wildcard is compromised, and all the applications that use the wildcard certificate are in danger. If an adversary has a private key, they can impersonate any web application that utilizes the wildcard certificate and trick users into giving their credentials.

Using OWASP Zap

- Absence of Anti-CSRF Tokens (medium risk)
    - HTML submission forms (search and login/sign up) have no protection against cross-site request forgery attacks meaning that an adversary can send HTTP requests on the user's behalf without the user knowing. The application believes that the adversary is the user and thus fulfills the requests sent by the adversary on the user's behalf.
    - Redesigning the submission forms with a library that minimizes the CSRF weakness or utilizing CSRF tokens is recommended.

- Content Security Policy (CSP) Header Not Set (medium risk)
    - CSP gives a set of HTTP headers that will allow only particular content to be rendered or displayed on the website. This helps prevent attacks such as Cross-Site Scripting or SQL Injection Attacks.
    - It is recommended to have the web server put in place the Content-Security-Policy header to have browsers render the intended content and not unexpected files.
- Cross-Domain Misconfiguration (medium risk)
    - The Access-Control-Allow-Origin header is set to all. This allows third parties to read responses from unauthenticated APIs and could allow attackers to access unauthorized data.
    - It is recommended that the Access-Control-Allow-Origin header is set to a more restrictive set of domains or remove the header entirely to enforce the Same

Origin Policy. This will allow only authorized APIs to access the resources of the website.

- Cookie without SameSite Attribute (low risk)
    - Cookies are not set to have the SameSite attribute, which means that a cookie can be sent as a request to other sites. This poses the risk of Cross-Site Request Forgery attacks, as cookies can be sent and used to have authenticated users send unintentional HTTP requests.
    - It is recommended to set the SameSite attribute to ‘strict’ for all cookies.

- The server shows information about the backend framework through the "X-Powered-By" HTTP Response Header Field (low risk)
    - The “X-Powered-By” header in the HTTP response shows that the web application backend is run with Express JS. This can tip off an adversary to see if Express JS has any vulnerabilities that can be exploited to attack the web application.
    - It is recommended that the application server is configured to hide the “X-Powered-By” header.
- X-Content-Type-Options Header Missing (low risk)
    - The X-Content-Type-Option header is not set to “nosniff” which means that previous versions of Chrome and Internet Explorer can do MIME-sniffing (Multipurpose Internet Mail Extensions) on the responses, which poses the risk of the responses being intercepted and rendered as a content type that is not intended.

- It is recommended to set the X-Content-Type-Option header to “nosniff” and set the Content-Type headers appropriately to render the intended content. If possible, have users utilize modern web browsers that cannot perform MIME-sniffing.

References

"Avoid Dangers of Wildcard TLS Certificates, the Alpaca Technique." National Security Agency/Central Security Service, 7 Oct. 2021, www.nsa.gov/Press-Room/News-Highlights/Article/Article/2804293/avoid-dangers-of-wildcard-tls-certificates-the-alpaca-technique/.

Banach, Zbigniew. "Information Disclosure Attacks in Web Applications." Invicti, 30 Aug. 2022, www.invicti.com/blog/web-security/information-disclosure-issues-attacks/. Information disclosure mitigation

Blixt, Karl-Fredrik, and Åsa Hagström. "Adding Non-Repudiation to Web Transactions." Non Repudiation for Web Transactions, June 1999, www.cse.msu.edu/~cse870/Public/Lectures/Security/NonRepudiation.html. Understanding non-repudiation in the context of transactions on websites.

Conklin, Larry, et al. "Threat Modeling Process." OWASP, owasp.org/www-community/Threat_Modeling_Process#determine-and-rank-threats. Accessed 10 May 2023. Threat model structure.

"Cross Site Scripting Prevention Cheat Sheet." Cheat Sheet Series, 2021, cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html. Tampering prevention.

Eaton, Ben. "What Is Mime Sniffing? - Keycdn Support." KeyCDN, 14 Apr. 2023, www.keycdn.com/support/what-is-mime-sniffing.

Francis, Abey. "Data Tampering – Meaning, Types and Countermeasures." MBA Knowledge Base, 3 Sept. 2018, www.mbaknol.com/information-systems-management/data-tampering-meaning-types-

and-countermeasures/#:~:text=Data%20tampering%20or%20data%20manipulation%20can%20usually%20be%20done%20through,other%20data%20including%20session%20tokens. List of types of data tampering and risks associated with each.

Furman, Michael. “Samesite Cookies - Everything You Need to Know.” Ultimate Security Professional Blog, 16 Feb. 2020, ultimatesecurity.pro/post/same-site-cookie/.

Haber, Morey J. “Privilege Escalation Attack & Defense Explained.” BeyondTrust, 2 Mar. 2021, www.beyondtrust.com/blog/entry/privilege-escalation-attack-defense-explained.

Haber, Morey J. “Privilege Escalation Attack & Defense Explained.” BeyondTrust, 2 Mar. 2021, www.beyondtrust.com/blog/entry/privilege-escalation-attack-defense-explained. Privilege elevation.

Haber, Morey J. “What Is Least Privilege & Why Do You Need It?” BeyondTrust, 2 Mar. 2021, www.beyondtrust.com/blog/entry/what-is-least-privilege. Privilege elevation.

“How to Identify and Exploit HTTP Host Header Vulnerabilities.” Web Security Academy, portswigger.net/web-security/host-header/exploiting. Accessed 10 May 2023. Understanding attacks on HTTP headers.

“How to Prevent Data Tampering in Your Business.” Cypress Data Defense, 24 June 2020, www.cypressdatadefense.com/blog/data-tampering-prevention/.

“How to Prevent Data Tampering in Your Business.” Cypress Data Defense, 24 June 2020, www.cypressdatadefense.com/blog/data-tampering-prevention/. Tampering prevention.

“Information Disclosure Vulnerabilities.” Web Security Academy, 2023, portswigger.net/web-security/information-disclosure. Information disclosure.

"Injection Prevention Cheat Sheet." Cheat Sheets Series, 2021, cheatsheetseries.owasp.org/cheatsheets/Injection_Prevention_Cheat_Sheet.html. Tampering injection prevention.

Irish, Paul, and Franklin Yu. "Strict Transport Security." MDN Web Docs, 10 May 2023, developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Strict-Transport-Security.

Klippert, Joni. "Node.Js CSRF Protection Guide: Examples and How to Enable It." StackHawk, 11 Oct. 2021, www.stackhawk.com/blog/node-js-csrf-protection-guide-examples-and-how-to-enable-it/.

Long, Nathan, et al. "X-Frame Options." MDN Web Docs, 10 Apr. 2023, developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options.

Martens, Ben. "What Is Spoofing & How to Prevent Spoofing Attacks in 2023?" SafetyDetectives, 25 Sep. 2021, www.safetydetectives.com/blog/what-is-spoofing-and-how-to-mitigate-an-under-the-radar-threat/. Spoofing Mitigation Techniques.

"Missing Content-Type Header." Invicti, 13 Jan. 2023, www.invicti.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/.

"Nikto: Kali Linux Tools." Kali Linux, 8 Mar. 2023, www.kali.org/tools/nikto/.

"OWASP® Zed Attack Proxy (ZAP)." OWASP ZAP, 27 Oct. 2022, www.zaproxy.org/.

Richardson, Stephen. "Repudiation Attacks - Firewall Security." Cisco Certified Expert, 9 May 2023, www.ccexpert.us/firewall-security/repudiation-attacks.html. Repudiation prevention.

"Skipfish: Kali Linux Tools." Kali Linux, 8 Mar. 2023, www.kali.org/tools/skipfish/.

"Spoofing: What Is a Spoofing Attack?" Malwarebytes, www.malwarebytes.com/spoofing. Accessed 10 May 2023. List of types of spoofing.

Swinhoe, Dan. "Man-in-the-Middle (MITM) Attack Definition and Examples." CSO Online, 25 Mar. 2022, www.csoonline.com/article/3340117/man-in-the-middle-attack-definition-and-examples.html. Understanding Man in the Middle attacks.

Tagade, Kanishk, et al. "Clickjacking." MDN Web Docs, 21 Feb. 2023, developer.mozilla.org/en-US/docs/Glossary/Clickjacking.

Tiwari, Yash. "Cookie Tampering Techniques." GeeksforGeeks, 2 Nov. 2022, www.geeksforgeeks.org/cookie-tampering-techniques/. Understanding cookie tampering.

"What Is a Denial of Service Attack (Dos) ?" Palo Alto Networks, www.paloaltonetworks.com/cyberpedia/what-is-a-denial-of-service-attack-dos. Accessed 10 May 2023. DoS attack information

"What Is an IP Address Spoofing Attack?" Microsoft 365, 29 Dec. 2022, www.microsoft.com/en-us/microsoft-365-life-hacks/privacy-and-safety/what-is-ip-address-spoofing-attack. Understanding IP spoofing attacks.

"What Is IP Spoofing?" Cloudflare, www.cloudflare.com/learning/ddos/glossary/ip-spoofing/. Accessed 10 May 2023. understanding how IP spoofing works.

Willee, Hamish, et al. "Using HTTP Cookies." MDN Web Docs, 10 Apr. 2023, developer.mozilla.org/en-US/docs/Web/HTTP/Cookies.