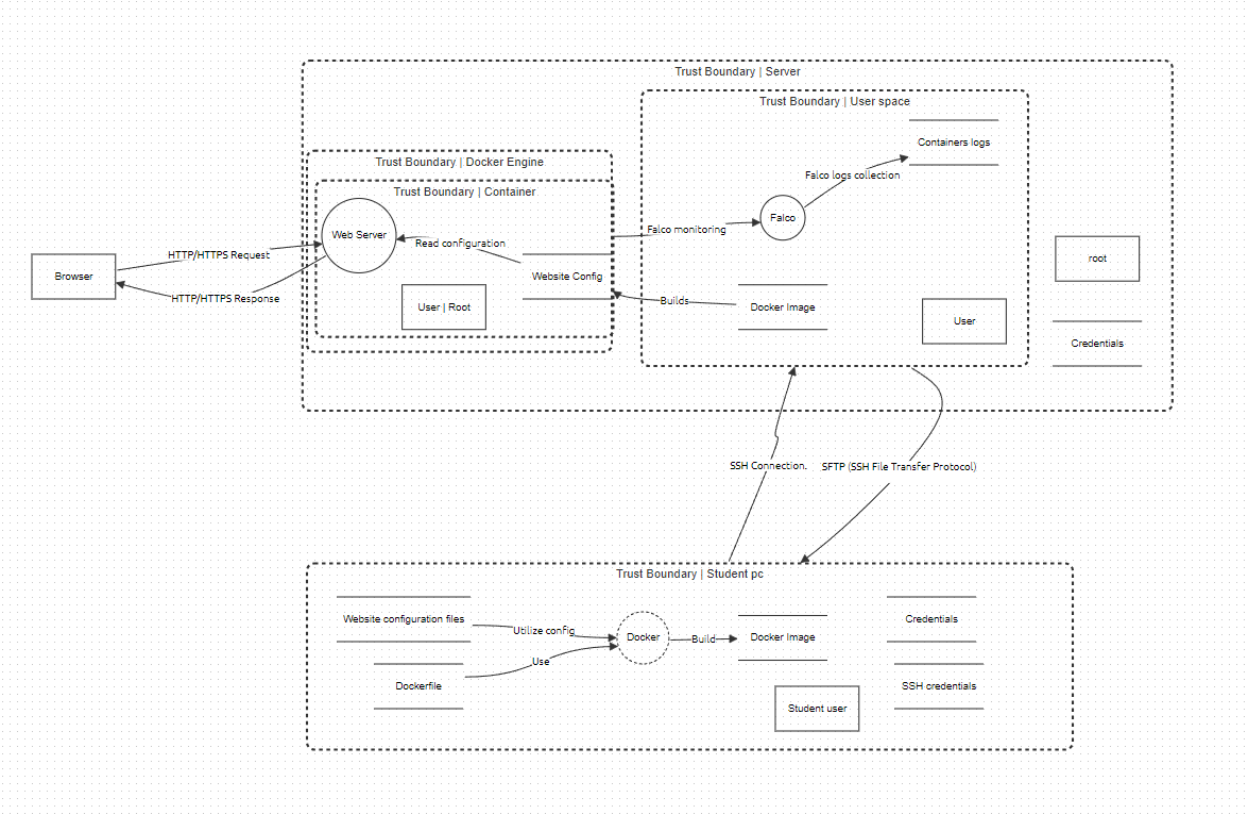
Student Website Threat Model

Owner: Teacher

Reviewer: Sohaib Saeed

Contributors:

Date Generated: Sept 15 2024



Executive Summary

High-Level System Description:

The system is a containerized website hosted on a cloud node, designed for a student portfolio. The website is built using HTML5 and CSS and is managed using Docker with Nginx as the web server. The student rents the server and develops the website locally, connecting to the cloud server via SSH.

Summary

Description	Count
Total Threats	10
Total Mitigated	5
Not Mitigated	5
Open / High Priority	2
Open / Medium Priority	3
Open / Low Priority	5
Open / Unknown Priority	0

Identified Threats

1. **Spoofing (S) - Web Server:**
   - **Description:** An attacker could create a fake server to intercept HTTP/HTTPS requests.
   - **Mitigation:** Enforce HTTPS with SSL/TLS certificates to authenticate the server and encrypt communications.
2. **Tampering (T) - Configuration Files:**
   - **Description:** Unauthorized users could alter the website configuration files.
   - **Mitigation:** Implement access controls and file integrity monitoring.
3. **Repudiation (R) - User Actions:**
   - **Description:** Users might deny deploying Docker images or modifying configurations.
   - **Mitigation:** Enable detailed logging and auditing of user actions.
4. **Information Disclosure (I) - Sensitive Data in Logs:**
   - **Description:** Sensitive information such as SSH credentials might be exposed if logs are not secured.
   - **Mitigation:** Encrypt sensitive data and restrict access to logs.
5. **Denial of Service (D) - Web Server Overload:**

- **Description:** An attacker could overwhelm the web server, leading to service disruptions.
- **Mitigation:** Implement rate limiting and use load balancers to distribute traffic.

6. **Elevation of Privilege (E) - Root Access via SSH:**
   - **Description:** Exploiting SSH vulnerabilities could grant an attacker root access.
   - **Mitigation:** Regular software updates and least privilege principles.
7. **Tampering (T) - Docker Image from Student PC:**
   - **Description:** Malicious code could be introduced into the Docker image during development.
   - **Mitigation:** Use security scanning tools and enforce code reviews.
8. **Spoofing (S) - SSH Credential Theft:**
   - **Description:** An attacker could impersonate the student user by stealing SSH credentials.
   - **Mitigation:** Multi-factor authentication (MFA) and secure SSH key management.
9. **Information Disclosure (I) - Falco Logs:**
   - **Description:** Unauthorized access to Falco logs could reveal sensitive information.
   - **Mitigation:** Encrypt logs and restrict access to authorized personnel.
10. **Repudiation (R) - Denying Faulty Docker Deployment:**
    - **Description:** A student might deny deploying a faulty Docker image that caused a service disruption.
    - **Mitigation:** Detailed logging and secure audit trails for deployment actions.

System STRIDE

Component	Type	Number	Title	Priority	Status	Score	Description	Mitigations
Browser (Actor)	Spoofing (S)	1	Spoofing Browser Requests	High	Open	8	An attacker could send spoofed requests to the server	Use HTTPS, CSRF tokens, and strong

							pretending to be the browser.	server-side validation.
Web Server (Process)	Tampering (T)	2	Tampering with Web Server Config	High	Open	7	Unauthorized user modification of web server configurations.	Enforce access controls and use file integrity monitoring.
Website Config (Store)	Info Disclosure (I)	3	Information Disclosure in Config	Medium	Open	6	Sensitive data could be exposed in configuration files.	Encrypt sensitive data and restrict access.
Read Configuration (Data Flow)	Elevation of Privilege (E)	4	Unauthorized Access to Config	Medium	Open	5	Unauthorized access during the read process.	Encrypt data, enforce access control policies.
HTTP/HTTPS Response (Data Flow)	Tampering (T)	5	Response Manipulation	High	Open	8	An attacker might intercept and modify server responses.	Use HTTPS, implement CSP, validate data integrity on the client side.
Use (Data Flow)	Tampering (T)	6	Misuse of Docker Image	Medium	Open	7	An attacker could tamper with the Docker image during its use.	Implement image signing and verification.

Utilize Config (Data Flow)	Tampering (T)	7	Configuration Data Misuse	Medium	Open	6	Misuse of configuration data.	Use version control, restrict access based on roles.
SFTP (Data Flow)	Info Disclosure (I)	8	Data Interception during SFTP Transfer	High	Open	7	Interception of sensitive data during transfer.	Use encryption for SFTP transfers, enforce strong authentication.
Docker Image (Store)	Tampering (T)	9	Deployment of Malicious Docker Image	High	Open	9	A malicious Docker image could be deployed.	Implement image scanning, enforce code reviews.
Containers Logs (Store)	Repudiation (R)	10	Log Tampering	Low	Open	5	An attacker could tamper with container logs to hide activities.	Use immutable logs, secure backups.
Falco (Process)	Denial of Service (D)	11	Overloading Falco Monitoring	Medium	Open	6	Excessive logging could overwhelm Falco, missing critical events.	Implement rate limiting, prioritize critical alerts.

Additional System Components

Component	Type	Number	Title	Priority	Status	Score	Description	Mitigations
Dockerfile (Store)	N/A	12	-	N/A	N/A	N/A	Dockerfile used to build Docker images.	N/A
SSH Credentials (Store)	Info Disclosure (I)	13	SSH Credential Theft	High	Open	8	Theft of SSH credentials leading to unauthorized access.	Use MFA, secure SSH keys with passphrases.
Credentials (Store)	N/A	14	-	N/A	N/A	N/A	Credentials stored securely for access control.	N/A
Student User (Actor)	N/A	15	-	N/A	N/A	N/A	Actor responsible for local development.	N/A
Root (Actor)	N/A	16	-	N/A	N/A	N/A	Root user with elevated privileges.	N/A
User (Actor)	N/A	17	-	N/A	N/A	N/A	Regular user with standard privileges.	N/A