Building a secure website

6005CEM

Saad Iftikhar

9789180

Introduction

The goal of this project is to provide the client with a secure messaging platform to encourage student involvement. The programme will attempt to mimic more traditional social media by allowing students to create and comment on a feed.

This website will have the features listed below.

1. Users will have varying levels of access.
   - The system will only be accessible to students and employees.
   - Students will be able to publish, comment to, and read other students' postings.
   - Lecture materials will be able to be added to and updated by staff.

   The admin account will have the following capabilities:
   - Add and delete people from the system
   - Users' access levels can be changed.
   - Course materials and posts can be viewed and edited.

2. API, there will also be a small mobile app that attempts to replicate the website's functionality using a simple REST API that should allow:
   - Authentication, the authentication/authorisation will be the same as for the web application
   - Create, Read, and Respond to posts. Posts will be seen in the same way that they are on the web app.

3. Analytics and logging
   - It was also determined that the system would collect logging and analytics data.

   Users will not be allowed to opt out since this will be used to measure student participation. There is no requirement for users to examine the analytics at this time, a dashboard has been planned. The following sorts of data will be collected:
     - Username
     - The browser, and the operating system details
     - Details of page clicks, and interaction with site elements
     - Geo-Location Data.

Design

Potential security risks and their solutions for each design aspect

Users have varying levels of access.

What a person can do directly, as well as what programmes running on their behalf, are restricted by access control. Access control works in this way to prohibit activities that might lead to a security breach (Sandhu et al., 1994, pp 13-48). To steal data or user credentials, access control attacks often overcome or bypass access control mechanisms. Adversaries can securely circumvent access control by signing in as the authorised user and accessing his or her resources by gathering the latter. The attackers then try to change the information in the system such that it is no longer accurate (CISSP Prep: Mitigating access control attacks - Infosec Resources, 2019).

Risks

Broken access control is number 1 in the OWASP Top 10 security risks 2021(OWASP Top 10:2021, 2021). Vertical privilege escalation occurs when a user gains access to functionality that they are not authorised to utilise (Academy, 2021). For example, a staff member can edit the material of other staff members. Metadata could be manipulated, by an attacker by replaying or tampering with a JSON Web Token (JWT) access control token, or a cookie or hidden field manipulated to elevate privileges or abusing JWT invalidation (A01 Broken Access Control - OWASP Top 10:2021, 2021). If passwords are not required to be long and have varying characters, they are not salted randomly or hashed potential hackers can attack and gain control of accounts on the website (L, 2020). SQL injections can be used to gain access to user accounts (Using SQL Injection to Bypass Authentication, 2021).

Solution

A Log should be kept for access control failures, alert admins when appropriate. Rate limit API and controller access should be implemented to minimize the harm from automated attack tooling. After logging out, stateful session IDs should be invalidated on the server. Stateless JWT tokens should be short-lived to reduce an attacker's window of opportunity (A01 Broken Access Control - OWASP Top 10:2021, 2021). Access rights should be approved/documented when changing roles, and they should be removed across all systems for any departures as soon as possible. Finally, as part of a security governance process, access rights should be evaluated and updated regularly (McGregor, 2016). Access should be denied by default unless a resource is designed to be publicly accessible (Academy, 2021). SQL or XSS injections may be prevented by Using LIMIT and other SQL controls, as well as affirmative server-side input validation, escaping special characters utilising the interpreter's unique escape syntax (A03 Injection - OWASP Top 10:2021, 2021).

Rest API

An API, or application programming interface, is a collection of rules that specify how programmes and devices may communicate with one another (rest-apis, 2021). Major data breaches are caused by APIs that are broken, exposed, or hacked (API security, 2019).

Risks

Rest APIs are venerable to MiTM (Man-in-the-Middle) Attacks a hostile third-party intercepts mobile API traffic, data scraping could be applied on rest API's as API gateways are typically ineffective against API scraping bots, which imitate legitimate app traffic and employ the proper API protocols, keys, and authentication. Malicious actors try to access another service or mobile application using login credentials (username/password combinations) acquired from a data breach against an organisation by credential surfing, a copy or App impersonation could be used to target a person for their information and DoS and DDoS Attacks can be used to bring the targeted application down, or at least make it unusable (Stewart, 2020). Incorrectly using HTTPS can cause a connection to be intercepted (Schöne, 2017).

Solutions

Traffic can be authenticated, as well as regulate and monitor how an APIs is utilised, by using API gateway and setting limitations for how frequently an API may be used and tracking its usage over time can prevent it from being abused or overrun (API security, 2019) and (Schöne, 2017). The mentioned being used against DoS and DDoS Attacks (Schöne, 2017). Encryption could be used to make sure the correct users are using modifying data and tokens could be used to limit access to services (API security, 2019). Thus, encryption would cause a hindrance to data scrapers and MiTM attacks.

Analytics and logging

Applications, operating systems, networks, and other components of a technological stack generate log files in real-time. Depending on the extent of technology involved in the review, log analysis might entail a huge quantity of data. Analysts can search for trends or irregularities in log entries that could suggest possible problems with a web server or websites, such as system-related difficulties or security vulnerabilities (What is Log Analysis? Uses Cases For Digital Performance, 2020).

Risks

Inadequate logging and monitoring affect the entire IT infrastructure, not simply the web application that is visible to the public. While insufficient logging and monitoring is too abstract to constitute a direct attack vector, it has an impact on breach discovery and reaction (Security Logging and Monitoring Failures Practical Overview | OWASP Top 10 | Exploits and Solutions, 2021). A website can be venerable if Logins, unsuccessful logins, and high-value transactions are not reported as auditable events. Warnings and errors provide no, insufficient, or ambiguous log messages. Suspicious behaviour is not tracked in the logs of apps and APIs. Only local logs are kept. There are no appropriate alerting levels in place, and response escalation systems are ineffective. The application cannot detect, escalate, or warn for active assaults in real-time or near real-time, therefore penetration testing and scans by dynamic application security testing tools do not generate alarms (A09 Security Logging and Monitoring Failures - OWASP Top 10:2021, 2021).

Solutions

These hazards can be reduced by implementing any or all of the following controls. Ensure that all login, access control, and server-side input validation failures can be logged with enough user context to identify suspicious or malicious accounts, that log data is encoded correctly to prevent injections or attacks on the logging or monitoring systems, and that log data is encoded correctly to prevent injections or attacks on the logging or monitoring systems. Ensure that high-value operations, such as append-only database tables or equivalent, have an audit trail with integrity controls to prevent tampering or deletion, and that suspicious activity is noticed and responded to immediately. Create or establish a strategy for responding to and recovering from an incident (A09 Security Logging and Monitoring Failures - OWASP Top 10:2021, 2021).

Implementation details

The Website was designed for the client by keeping the client's requirements, mentioned security issues and the OWASP top ten web application security risks (OWASP Top 10:2021, 2021) (see Table 1 OWASP Top Ten Web Application Security Risks) in mind to deliver a secure and reliable website to the client.

Ranking 2021	Security Risk
A01	Broken Access Control
A02	Cryptographic Failures
A03	Injection
A04	Insecure design
A05	Security Misconfiguration
A06	Venerable and outdated components
A07	Identification and Authentication Failures
A08	Software and data integrity failures
A09	Security logging and monitoring failures
A010	Server-side requests Forgery

Table 1 OWASP Top Ten Web Application Security Risks

The Website will be made using python3 using flask and SQL will be used on the website. The website will avoid using any outdated components as they are a high-security risk (A06 Vulnerable and Outdated Components - OWASP Top 10:2021, 2021) and avoid using insecure design (A04 Insecure Design - OWASP Top 10:2021, 2021). Digital signatures have been used to verify that the software used is from an expected source and has not been altered to avoid Software and data integrity failures (A08 Software and Data Integrity Failures - OWASP Top 10:2021, 2021).

On the website users have different access levels, logs are kept for access control failures so admins can be alerted in the face of a potential threat. Harm from automated attacks via tools is minimized by the use of Rate limit API. The system is only accessible to students and employees. Students can publish, comment, read other students posting and lecture material have been added and updated by the staff. Stateless sessions IDs are invalidated on the server and Stateless JWT tokens are short-lived. There is a page where access rights can be approved/documented when changing roles, and they can be removed across all systems for any departures. When a user requests access to special resources or privileges it will be rejected by default unless it is manually approved. The website will use LIMIT and other SQL controls, as well as affirmative server-side input validation, escaping special characters utilising the interpreter's unique escape syntax to avoid SQL and XSS injections (A03 Injection - OWASP Top 10:2021, 2021).

The user passwords have been salted and hashed using Argon2, all data has been encrypted HTTP and TLS and caching containing sensitive data is disabled to avoid Cryptographic Failures (A02 Cryptographic Failures - OWASP Top 10:2021, 2021). The website avoids using any features, components, documentation, samples and has very generic and non-descriptive error messages (A05 Security Misconfiguration - OWASP Top 10:2021, 2021). Multi-factor authentication has been implemented and admin accounts have not been deployed with default credentials. Login attempts have been limited and login failures will be logged. Weak passwords are checked and not allowed to avoid Identification and Authentication Failures (A07 Identification and Authentication Failures - OWASP Top 10:2021, 2021).

The website has a mobile app that replicates the website's functionality using a Rest API and will allow the users to perform their required tasks on the mobile app. The API with its usage over time and traffic on the App will be regulated so it will avoid being overrun. Tokens will be used to limit access to services.

The website will collect the username, the browser and the operating system of the user, Details of page clicks, and interaction with site elements of a user and Geo-Location data of a user in a log file. Users will not be allowed to opt-out student participation will be tracked. A dashboard has been made to display the mentioned data.

All the access control and server-side input validation failures are logged. The log data is encoded and a backup is created for responding and recovering from incidents so Security logging and failure monitoring can be done (A09 Security Logging and Monitoring Failures - OWASP Top 10:2021, 2021).

To lessen the impact of SSRF, the remote resource access functions have been divided into distinct networks. To prevent all except critical internet traffic, "deny by default" firewall settings or network access control rules have been implemented. Firewalls will keep track of all permitted and banned network flows All client-supplied input will be sanitised and validated. With a positive allow list, the URL schema, port, and destination will be enforced. HTTP redirections have been disabled. To prevent Server-side requests Forgery (A10 Server Side Request Forgery (SSRF) - OWASP Top 10:2021, 2021).

Summary

The website was made with all the client’s requirements fulfilled. When developing the website, the main issues looked at were Broken Access Control, Injections and Security logging and monitoring failures which affected the client’s requirements directly but developing the website all the OWASP top ten web application security risks (OWASP Top 10:2021, 2021) have been used to secure the client's website. It is built so it can deal with Broken Access Control, Cryptographic Failures, injections, Security Misconfiguration, Identification and Authentication Failures, Software and data integrity failures, Security logging and monitoring failures, Server-side requests Forgery, it has been made sure that Venerable and outdated components are not used on the website, it has also been made sure that Security Misconfiguration and Insecure design are not present on the website.

References

Academy, W., 2021. *Access control vulnerabilities and privilege escalation | Web Security Academy*. [online] Portswigger.net. Available at: <https://portswigger.net/web-security/access-control> [Accessed 9 December 2021].

AppDynamics. 2020. *What is Log Analysis? Uses Cases For Digital Performance*. [online] Available at: <https://www.appdynamics.com/topics/what-is-log-analysis#~1-how-does-log-analysis-work> [Accessed 10 December 2021].

Ibm.com. 2021. *rest-apis*. [online] Available at: <https://www.ibm.com/uk-en/cloud/learn/rest-apis> [Accessed 10 December 2021].

Immuniweb.com. 2021. *Security Logging and Monitoring Failures Practical Overview | OWASP Top 10 | Exploits and Solutions*. [online] Available at: <https://www.immuniweb.com/blog/OWASP-security-logging-and-monitoring-failures.html> [Accessed 10 December 2021].

Infosec Resources. 2019. *CISSP Prep: Mitigating access control attacks - Infosec Resources*. [online] Available at: <https://resources.infosecinstitute.com/certification/mitigating-access-control-attacks/> [Accessed 9 December 2021].

L, K., 2020. *Password Cracking Is Easy: Here's How to Do It*. [online] Medium. Available at: <https://kennymuli.medium.com/password-cracking-is-easy-heres-how-to-do-it-875806a1e42a> [Accessed 9 December 2021].

McGregor, G., 2016. *The Three Most Common Access Control Issues*. [online] Blog.grantmcgregor.co.uk. Available at: <https://blog.grantmcgregor.co.uk/2016/the-three-most-common-access-control-issues> [Accessed 9 December 2021].

Owasp.org. 2021. *A01 Broken Access Control - OWASP Top 10:2021*. [online] Available at: <https://owasp.org/Top10/A01_2021-Broken_Access_Control/> [Accessed 9 December 2021].

Owasp.org. 2021. *A02 Cryptographic Failures - OWASP Top 10:2021*. [online] Available at: <https://owasp.org/Top10/A02_2021-Cryptographic_Failures/> [Accessed 10 December 2021].

Owasp.org. 2021. *A03 Injection - OWASP Top 10:2021*. [online] Available at: <https://owasp.org/Top10/A03_2021-Injection/> [Accessed 10 December 2021].

Owasp.org. 2021. *A04 Insecure Design - OWASP Top 10:2021*. [online] Available at: <https://owasp.org/Top10/A04_2021-Insecure_Design/> [Accessed 10 December 2021].

Owasp.org. 2021. *A05 Security Misconfiguration - OWASP Top 10:2021*. [online] Available at: <https://owasp.org/Top10/A05_2021-Security_Misconfiguration/> [Accessed 10 December 2021].

Owasp.org. 2021. *A06 Vulnerable and Outdated Components - OWASP Top 10:2021*. [online] Available at: <https://owasp.org/Top10/A06_2021-Vulnerable_and_Outdated_Components/> [Accessed 10 December 2021].

Owasp.org. 2021. *A07 Identification and Authentication Failures - OWASP Top 10:2021*. [online] Available at: <https://owasp.org/Top10/A07_2021-Identification_and_Authentication_Failures/> [Accessed 10 December 2021].

Owasp.org. 2021. *A08 Software and Data Integrity Failures - OWASP Top 10:2021*. [online] Available at: <https://owasp.org/Top10/A08_2021-Software_and_Data_Integrity_Failures/> [Accessed 10 December 2021].

Owasp.org. 2021. *A09 Security Logging and Monitoring Failures - OWASP Top 10:2021*. [online] Available at: <https://owasp.org/Top10/A09_2021-Security_Logging_and_Monitoring_Failures/> [Accessed 10 December 2021].

Owasp.org. 2021. *A10 Server Side Request Forgery (SSRF) - OWASP Top 10:2021*. [online] Available at: <https://owasp.org/Top10/A10_2021-Server-Side_Request_Forgery_%28SSRF%29/> [Accessed 10 December 2021].

Owasp.org. 2021. *OWASP Top 10:2021*. [online] Available at: <https://owasp.org/Top10/> [Accessed 9 December 2021].

Portswigger.net. 2021. *Using SQL Injection to Bypass Authentication*. [online] Available at: <https://portswigger.net/support/using-sql-injection-to-bypass-authentication> [Accessed 10 December 2021].

redhat. 2019. *API security*. [online] Available at: <https://www.redhat.com/en/topics/security/api-security> [Accessed 10 December 2021].

Sandhu, R. S., & Samarati, P. (1994). Access control: principle and practice. *IEEE communications magazine*, *32*(9), 40-48.

Schöne, P., 2017. *List of REST API security risks*. [online] Axway Blog. Available at: <https://blog.axway.com/amplify-products/api-management/rest-api-security> [Accessed 10 December 2021].

Stewart, D., 2020. *Top 5 Threats to APIs Servicing Mobile Apps*. [online] Blog.approov.io. Available at: <https://blog.approov.io/top-5-threats-to-apis-servicing-mobile-apps> [Accessed 10 December 2021].