Dylan Whiteford
19214597

Cloud computing security

What is cloud computing security

Cloud computing security is a group of policies and technologies created to protect and prevent attacks from security threats from within or outside an organisation. It serves to protect the data, infrastructure and applications on the cloud services.

What does cloud security entail

The cloud computing threats can in some cases be similar to normal computing threats however there are some unique threats it faces too. Some include Lack of visibility, multitenancy, access management, misconfiguration and compliance. These threats are related to the fact that the data, application or infrastructure is on the cloud. The most used approaches to cloud security are preventative(stop before it happens), detective(spot while it's happening), automated(responding if spotted or prevented) and administrative(looking at what you have in place) controls.

Latest trends in cloud computing security

1. Zero trust approach to security - no predefined trust given to user and users authorisation is constantly checked(Arora, 2024). Not checked once and then allowing them to use the system forever more.
2. Intelligent security - Making use of AI and ML to help prevent and protect against threats(Arora, 2024). This allows for large amounts of data and captivity to be monitored.
3. Secure devops - Pace of development is only increasing and therefore implementing security within the CI/CD process will provide more protection(Arora, 2024). It also prevents harmful or unsecure code from being deployed which will put your application at risk.

Dylan Whiteford
19214597

Tasks

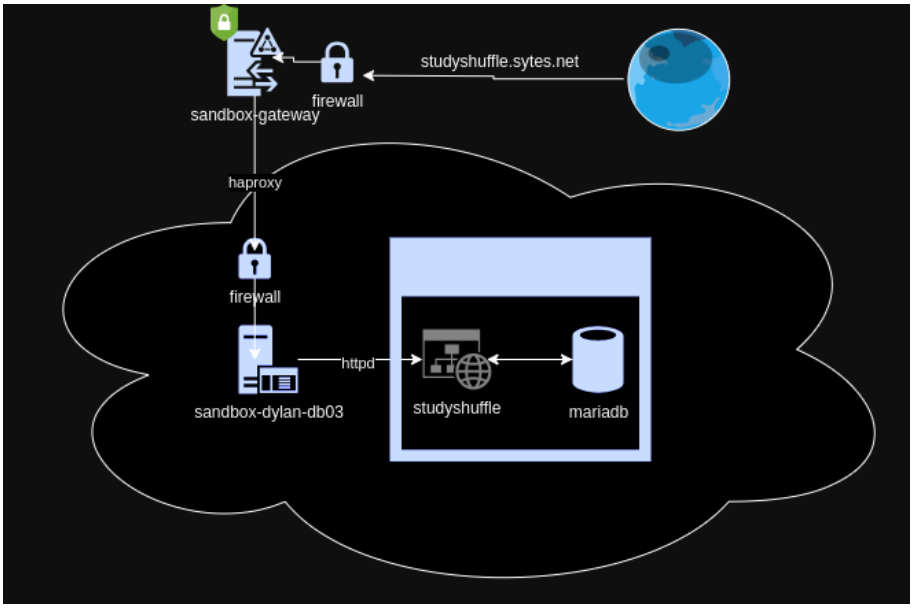
1. Setup cloud environment

I made use of my work's cloud infrastructure for this project. Southern Cross solution is a Iaas company which supplies dev-sec-ops services.

The production version of this application is running on a rocky linux VM in a cloud environment in a software defined network. Both a spring app and mariadb db instance are running on this machine. The machine's name is sandbox-dylan-db03. Ideally the spring app and database would be running on two separate machines but due to resource and time constraints i decided to setup both services on the same machine, this is a security risk.

sandbox-dylan-db03 is only accessible via a VPN. Only authorised users can connect to this machine via ssh through a vpn. The website is exposed to the public through the use of public proxy, please see the gateway diagram for more context.

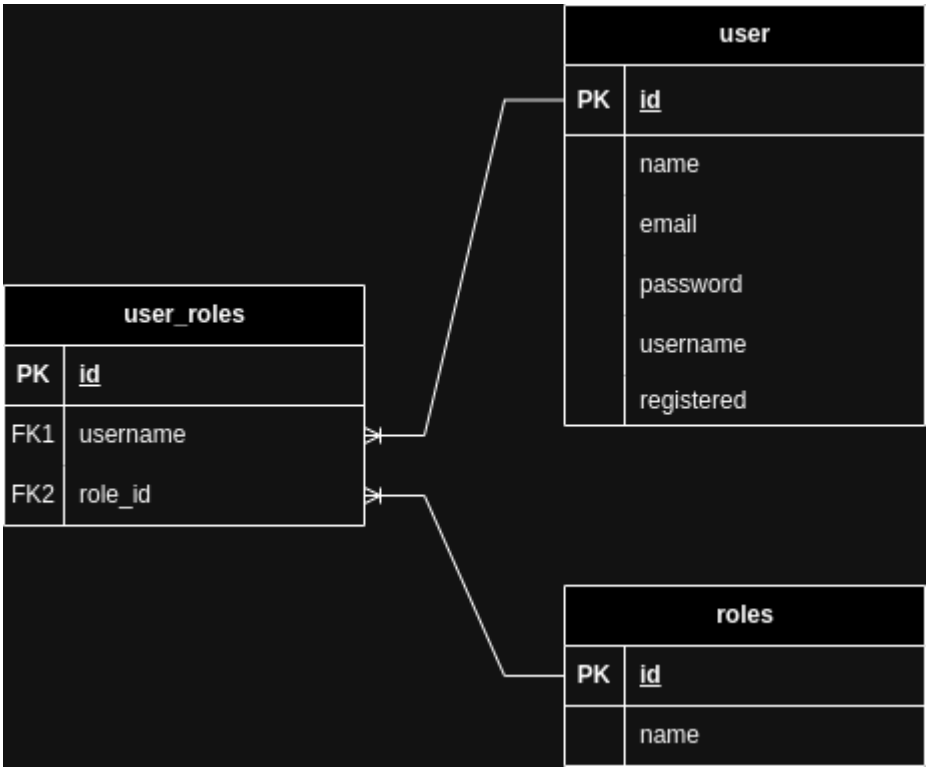
This proxy's name is sandbox-gateway-0x. sandbox-gateway-0x is running haproxy, which proxies tcp requests to the desired endpoints. For my project I have a public domain which points to sandbox-gateway-0x, any tcp request from this domain is proxied through to sandbox-dylan-db03. Apache is running on sandbox-dylan-db03. Apache serves the ssl certificate and proxies request through to a instance of the spring app running locally on non-specific port (8081).



Dylan Whiteford
19214597

2. Application design interface

I have created a springboot web application. This application acts as both a rest api as well as webclient. The use of thymeleaf was employed to create all html frontend pages.
I am making use of springboot's JPA framework to manage and persist database records.



3. Security measure

- Https has been setup. SSL certificates were generated with certbot/letsencrypt.
- Access to the cloud environment is hidden behind a vpn. Only user's on the VPN have visibility to the web application and database server. Identity management is achieved with ssh and private/public key pairs. Only authorised users using the private keys can ssh into the sandbox server.
- All user data is transferred via http post requests
- All sensitive user data (such as passwords) is bcrypted before it is stored in the database.

Dylan Whiteford
19214597

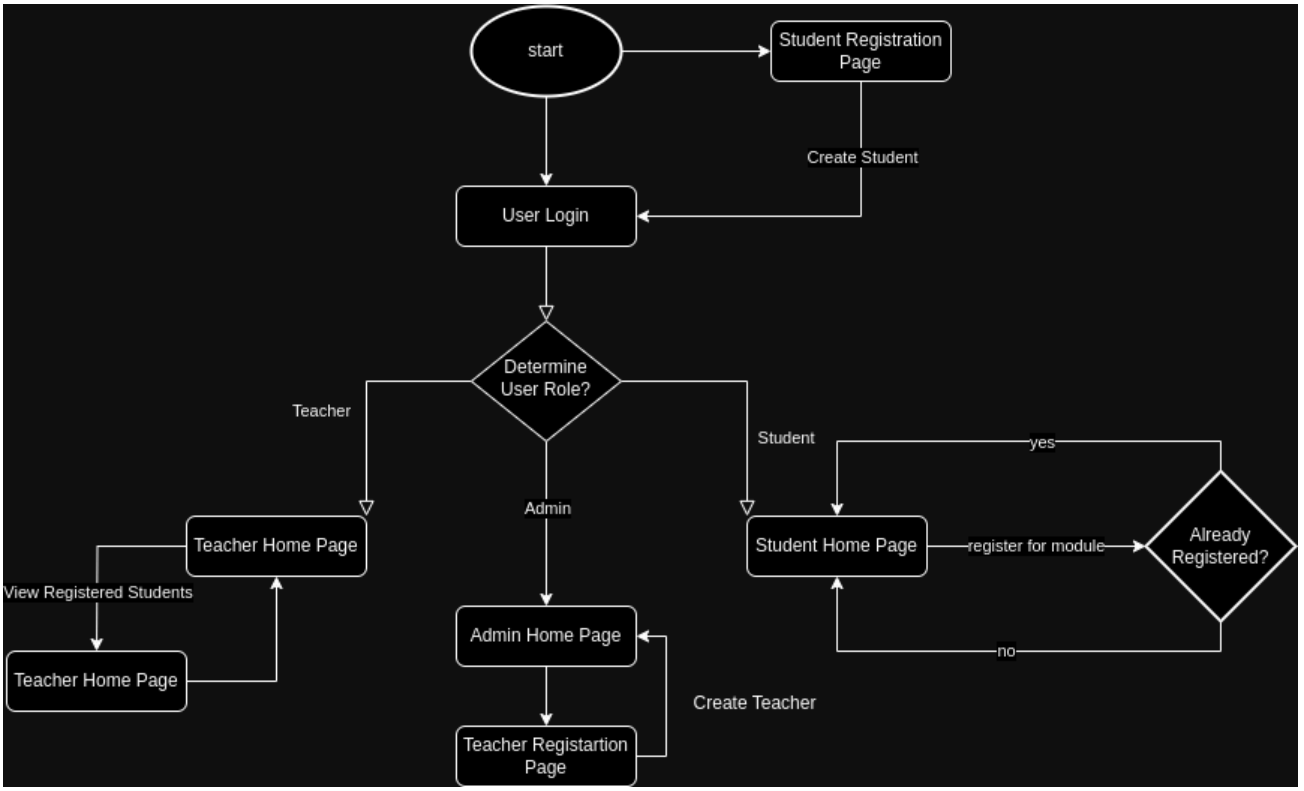
4. Authentication and authorizations

I have created 3 roles :

a. Student - can register for a module
b. Teacher- can view registered students for a module
c. Admin - can create teacher accounts

All the roles make use of password authentication, giving the various user authorization according to their role. A user can only access content that is assigned to their role.

Session management and role identification is achieved through the use of JWT(Java Web Tokens). This is a special token that is generated, stored in the user's cookies. This token can be used to identify, authorise and determine a user's role.



5. Logging and Monitoring

I make use of spring boot logback framework to log key application events.
I wanted to make use of my company's monitoring platform. I ran out of time. The goal was to make use of elastic's elk stack. The use of elastic would have allowed me to implement application program monitoring as well as availability monitoring. Elastic provides a java application which auto instruments spring boot web applications.

6. Security testing

No testing

Dylan Whiteford
19214597

7. Documentation

Security features

a. Encrypted DB passwords - this means that if the DB is compromised the passwords of users are not at risk as
b. DevSecOps/monitoring - Deployment is automated and takes security into consideration and is protected on a private companies cloud resources
c. Managing access - implemented login based off username and password
d. Understanding security and compliance
e. Error handling
f. Database backup - cronjob that run daily and does a mysqldump of the database

Deployment

sandbox-dylan-db03 is responsible for building and deploying the project. There is a cron job that periodically checks for changes on the remote git repository on github. If it detects a change, it will trigger a ‘mvn package’, then copy that project to the deployment directory and start it by using systemd. A systemd service is defined to manage the starting and stopping of the java application. The production environment (application.properties) only exists on sandbox-dylan-db03. This environment is loaded in when the java app starts. See more details in the readme of the project.

Dylan Whiteford
19214597

Reference

(No date) *Top 5 cloud security trends*. Available at: https://www.oracle.com/a/ocom/docs/top-five-cloud-security-trends-en.pdf (Accessed: 26 May 2024).

Arora, A.A. (2024) *Top 6 cloud security trends in 2024 (+best practices)*, *CloudDefense.AI*. Available at: https://www.clouddefense.ai/cloud-security-trends/ (Accessed: 26 May 2024).

Bezkoder (2023) *Spring boot login example: Rest api with mysql and JWT*, *BezKoder*. Available at: https://www.bezkoder.com/spring-boot-login-example-mysql/ (Accessed: 26 May 2024).

Top 6 cloud security trends in 2024 (2024) *Astra Security Blog*. Available at: https://www.getastra.com/blog/cloud/cloud-security-trends/ (Accessed: 26 May 2024).

What is cloud security? (2021) *What is cloud secuirty*. Available at: https://www.ibm.com/topics/cloud-security (Accessed: 26 May 2024).

What is cloud security? (no date) *What Is Cloud Security? | Oracle South Africa*. Available at: https://www.oracle.com/za/security/cloud-security/what-is-cloud-security/ (Accessed: 26 May 2024).

What is cloud security? | google cloud (no date) *Google*. Available at: https://cloud.google.com/learn/what-is-cloud-security#:~:text=Cloud%20security%20defined,-Cloud%20security%20is&text=This%20includes%20applying%20security%20policies,online%20attacks%2C%20and%20insider%20threats (Accessed: 26 May 2024).

Moyle, E. (2024) *5-step iaas security checklist for cloud customers*, *Security*. Available at: https://www.techtarget.com/searchsecurity/tip/5-step-IaaS-security-checklist-for-cloud-customers (Accessed: 26 May 2024).

Campbell, N. (no date) *7 cloud security best practices*, *7 Cloud Security Best Practices*. Available at: https://www.liquidweb.com/blog/cloud-security-best-practices/ (Accessed: 26 May 2024).