Faculdade de Design, Tecnologia e Comunicação
Universidade Europeia

Relatório Final

Sistema Two Factor Authentication

(integração com Sistema Anti-Ransomware do grupo G04)

Engenharia Informática

Sistemas Distribuídos

Professor Pedro Rosa

Realizado por:

Mafalda Quintas - 50036446

Rafael Pilré - 20190877

Tiago Silva - 20190878

Repositório de github: https://github.com/mrquintas98/2FA

Composição do Grupo			
Número / Nome	Esforço (Horas)		
	Pesqui. Web	Reunião	Implementação
50036446 – Mafalda Quintas	30	50	50
20190877 – Rafael Pilré	30	50	50
20190878 – Tiago Silva	30	50	50

Enquadramento

A Autenticação de Dois Fatores (2FA) é um processo de verificação em duas etapas que visa proporcionar um nível adicional de segurança ao exigir que o utilizador se autentique a si próprio utilizando um meio secundário. Sem a utilização do 2FA, um hacker poderia obter acesso aos dispositivos ou contas de uma pessoa apenas conhecendo a palavra-passe da vítima, enquanto que com o 2FA sabendo apenas esta palavra-passe é insuficiente para passar a verificação de autenticação. O objetivo deste trabalho consiste em implementar uma solução de autenticação de dois fatores em conjunto com outro grupo de trabalho para que consigamos reforçar a segurança do sistema. Esperamos com este projeto, beneficiar a sociedade a aumentar o conhecimento geral do 2FA conduzindo a serviços mais seguros, como também ganhar experiência em trabalhar com equipas de projetos diferentes mas que se ligam ao mesmo tempo.

Casos de uso

1. O utilizador, após efetuar o primeiro login corretamente, pede o envio de um Token de autenticação.
2. A aplicação envia um pedido de criação de Token e envio do mesmo para o email do utilizador.
3. O utilizador recebe o Token de autenticação no seu email.
4. Inserção do código Token na janela do browser criada para esse efeito.
5. É feito um pedido de validação de Token.
    a. Caso esteja certo, o login é efetuado.
    b. Caso esteja errado, uma mensagem de erro é mostrada ao utilizador e o mesmo, caso assim o queira, faz um pedido de novo Token.

Solução a implementar

i. Descrição genérica

Após o utilizador se registar com sucesso no *website*, os dados inseridos serão guardados na Base de Dados (utilizando um algoritmo de encriptação chamado de bcrypt) da aplicação e, ao mesmo tempo, é gerado um *Token* aleatoriamente, encriptado com o algoritmo Galois Counter Mode (*aes-256-gcm*[1]) e guardado na mesma.

Ao efetuar um início de sessão, o *Token* presente na Base de Dados é desencriptado e enviado, através de *email*, ao utilizador para este utilizar como segundo fator de autenticação (após inserir o *email* e *password* da conta). Caso o *Token* não seja aceite (quer por ser o *Token* errado ou por o tempo limite de 30 segundos para inserção do mesmo ter expirado) o mesmo terá que ser gerado novamente e enviado ao utilizador. Caso contrário, será barrado, na sua totalidade, o acesso do utilizador às informações presentes após o início de sessão.

Como solução para evitar faltas, é configurado uma máquina virtual como *Load Balancer* e *Reverse Proxy* que, ao receber os pedidos vindos dos utilizadores, distribui os mesmos pelas máquinas disponíveis para que as mesmas não sejam sobrecarregadas e que não tenham acesso direto aos web servers. No entanto, tendo em conta a escala desta solução, irá, inicialmente, haver apenas uma máquina a trabalhar enquanto que a segunda irá estar, em intervalos de tempo pré-definidos, a fazer *pings* de modo a perceber se a máquina principal se encontra a funcionar ou não. Deste modo não estão as duas máquinas a gerar códigos

automaticamente para os mesmos utilizadores mas, caso uma vá abaixo, a outra "percebe" rapidamente e entra em funcionamento.

No que toca à integração com o trabalho do grupo *G04 - Sistema Anti-Ransomware*, após o utilizador efetuar o *login* na aplicação de repositório de ficheiros, irá haver um pedido por parte desta aplicação à nossa, para que um novo *Token* seja gerado e enviado para o *email* utilizado para efetuar o *login.*

ii. Enquadramento nas áreas da unidade curricular

Quando se trata de segurança on-line, o fator de autenticação mais comum, de longe, é a combinação nome de utilizador/senha. Isso significa que a maioria dos sistemas está a usar apenas a autenticação de fator único. Também é importante lembrar que as pessoas têm mais contas on-line do que quando as senhas foram introduzidas pela primeira vez, o que significa que, muitas vezes, há muitas senhas para serem lembradas. Isso pode levar à "reciclagem de senha", que é quando a mesma senha é usada para várias contas, facilitando o acesso a terceiros.Um sistema distribuído vai permitir também que tanto no servidor ou aplicação caso haja algum tipo de erro, este não comprometa o risco da operacionalidade do sistema.

iii. Requisitos técnicos

O sistema 2FA irá utilizar uma máquina virtual onde estará o nginx instalado funcionando como reverse proxy e Load Balancer e pelo menos 2 Web Servers sendo que um estará responsável por fazer a geração de Tokens.

Inicialmente, aquando da proposta do projeto, referimos que:

- "O sistema 2FA irá utilizar 1 HAProxy Load Balancer e pelo menos 2 *Web Servers* para geração de Tokens. Irá ter acesso a uma Base de Dados onde irá estar presente qual a aplicação que está a pedir os Tokens, os utilizadores para os quais os Tokens vão ser gerados e a string do último Token gerado (em Hash). Irá também ser construída uma aplicação mobile para apresentação do Token ao cliente que efetua Login no Web Site.".

Apesar de, grande parte daquilo que foi proposto ser o que foi implementado, houve ainda algumas mudanças significativas, nomeadamente:

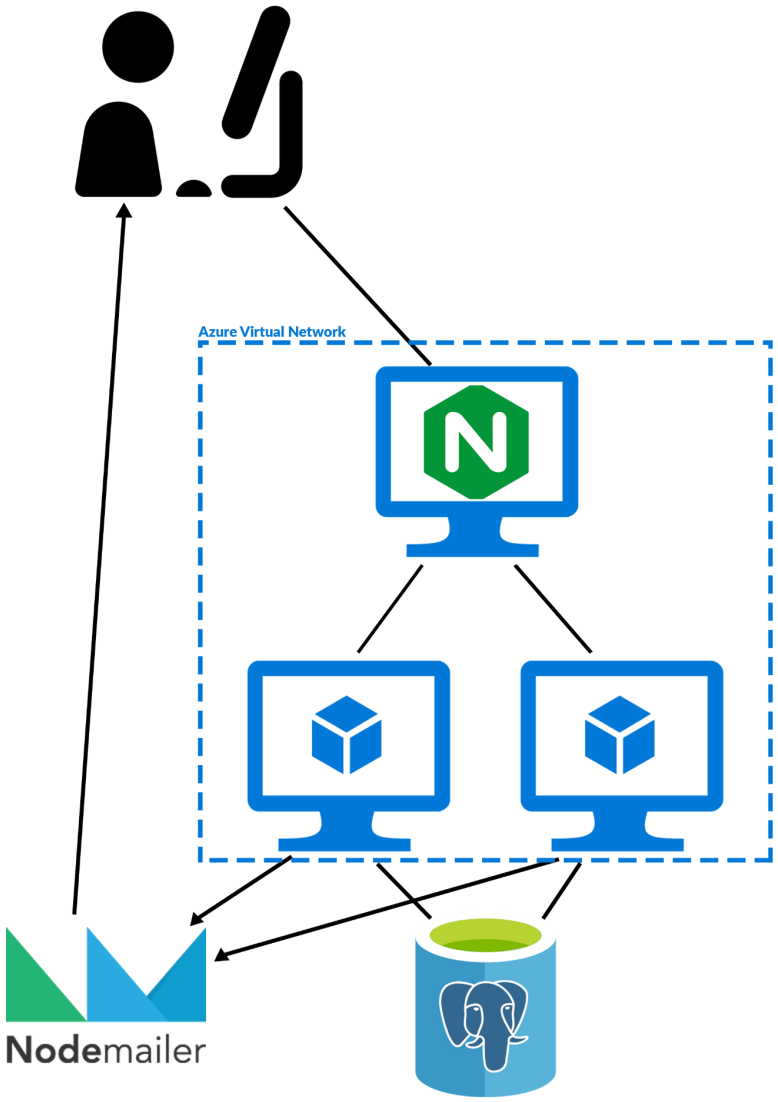
- Na Base de Dados, visto que, atualmente, o nosso projeto trabalha, exclusivamente, com o projeto do grupo *G04*, não é guardado nenhuma

indicação para qual aplicação os *Tokens* estão a ser criados, apenas os seus dados de *login* (*email* e *password* devidamente encriptados);

- Visto que, a própria aplicação do grupo G04 está construída em formato de *Website*, decidimos fazer o mesmo e optar por uma abordagem que utiliza o envio de um *email* ao utilizador com o *Token* para segundo fator de autenticação.

Sendo assim, o projeto de 2FA acaba por utilizar 3 máquinas virtuais (1 Reverse Proxy/Load Balancer - NGINX, e 2 *Web Servers - Node.js*) e um serviço de envio de *emails* por *Node.js* (*Nodemailer).* Foi também construído um *Website* sediado na aplicação do *Heroku*, que é, também, a aplicação que faz a gestão da nossa Base de Dados online.

iv. Arquitetura da solução



Tendo em conta a integração do projeto com o grupo G04, a solução irá funcionar da seguinte forma:

1. O utilizador irá efetuar o seu *login* na aplicação de repositório de ficheiros.
    a. Se o *login* for efetuado com sucesso, é feito um pedido à nossa aplicação para que seja gerado um *Token* e o mesmo enviado para o *Email* com o qual o utilizador efetuou o *login.*
    b. Se o login não for efetuado com sucesso, nada acontece e é barrado o acesso ao utilizador.
2. Após o utilizador receber o *Token* no seu *email*, o mesmo terá que ser inserido no local do *Site* preparado para este passo.
    a. Caso o *Token* esteja correto, é dado ao utilizador acesso à informação do sistema de repositório de ficheiros.

b. Caso o *Token* ou não esteja correto ou a janela de tempo disponível para a verificação do mesmo (30 segundos) expire, uma mensagem de erro é mostrada ao utilizador e o mesmo terá que repetir o seu *login* para que seja gerado e enviado um novo *Token.*

v. Tecnologias utilizadas

As tecnologias que utilizamos como meios para atingir o nosso objetivo foram as seguintes:

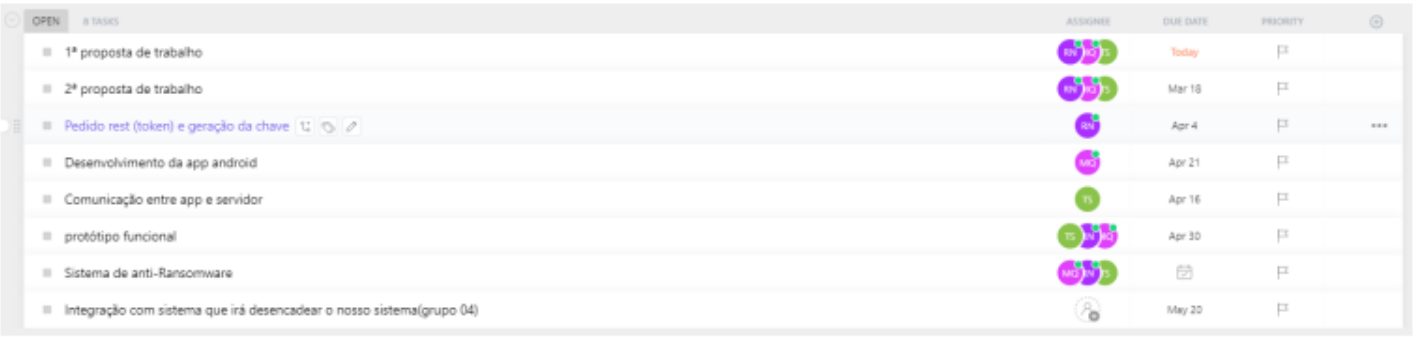
Todas as tecnologias foram escolhidas por serem de fácil aprendizagem e implementação.

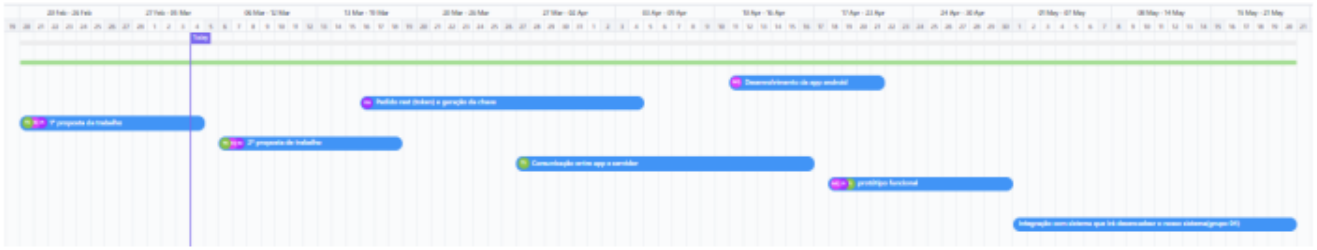
Node.js, Heroku e PostgreSQL foram as únicas tecnologias escolhidas que, para além de serem de fácil aprendizagem e implementação, são também tecnologias que os elementos do grupo já tinham conhecimento e experiência com.

No que toca à virtualização, foi escolhida a plataforma de Máquinas Virtuais da Microsoft, o Azure, por motivos técnicos (nenhum dos portáteis de trabalho do grupo tinha a potência suficiente para aguentar com as máquinas virtuais necessárias para o projeto). Neste mesmo ponto, o NGINX foi uma questão de pesquisa sobre tecnologias existentes e aquela que mais se adequa ao projeto a desenvolver.

Por fim, ao haver uma necessidade de enviar um *email* ao utilizador, fomos à procura de um serviço que nos possibilitasse fazer a ligação entre aquilo que já tínhamos e a funcionalidade pretendida e daí termos escolhido o Nodemailer.

Planeamento e calendarização





Resultados

Durante a implementação do projeto, houve algumas mudanças relativas ao mesmo, no entanto, a maior delas foi, sem dúvida, mudar de um cliente *Mobile* para receber o *Token* por *email*.

Relativamente aos pontos menos fortes do projeto, temos que apontar ao facto de existir a possibilidade que, mais do que um utilizador possa ter o mesmo *Token* de verificação. Este ponto pode, apesar de a nível do sistema em si não apresentar qualquer tipo de problema, apresentar problemas a nível da segurança visto que pode ser "explorado" por alguém que se dê ao trabalho de programar algo extremamente eficiente (visto que cada *Token* tem um tempo de vida de apenas 30 segundos). Mesmo assim, visto que todas as informações relativamente a utilizadores são encriptadas (utilizando dois algoritmos diferentes), isto acaba por nos fazer ganhar tempo e segurança perante estes ataques.

Sendo assim, como trabalho futuro, o nosso foco seguinte seria fazer uma verificação de *Tokens* já criados para termos a certeza que não existem *Tokens* repetidos para nenhum utilizador, de forma a mitigar qualquer tipo de possibilidade de ataque. Esta solução obrigaria à utilização tanto de máquinas virtuais mais potentes como de algoritmos e código extremamente bem construído e com uma performance e tempo de processamento bastante baixo. Como consequência positiva fazia com que, no entanto, houvesse a possibilidade de haver mais utilizadores a utilizarem a nossa aplicação de 2FA e dar-lhes um maior nível de segurança quando o fizessem.

Em suma, o objetivo do projeto foi cumprido, aprendemos bastante com o mesmo (especialmente no que toca a algoritmos de encriptação e à utilização de *TOTPs - Time-based One Time Passwords*) e acabou por nos motivar a, caso haja essa oportunidade, a acabar este projeto e torná-lo em algo real visto que, hoje em dia, aquilo que é mais valorizado na Internet é a segurança dos nossos próprios dados.

Bibliografia

1. Ahmad, N., Wei, L. M., & Jabbar, M. H. (2018, June 1). *IOPscience*. Journal of Physics: Conference Series. Retrieved May 20, 2022, from https://iopscience.iop.org/article/10.1088/1742-6596/1019/1/012008