Segurança de Sistemas – 2023-2024

Relatório do Trabalho de grupo D6

Duarte Bento Batista (2211089)
João Miguel Pereira Lopes (2212561)
Manuel José Antunes Eusébio (2211049)
Rafael Moreira Nunes (2212939)

Introdução

Desde a Revolução Digital, as empresas e as instituições de vários países começaram a necessitar de guardar os seus dados em formato digital, para isso necessitaram de sistemas que guardassem a informação de um modo seguro, de maneira a não perder esses dados. Infelizmente devido à falta de segurança que existe nos equipamentos que tratam estes dados, nomeadamente servidores e computadores, os dados eram facilmente acedidos por utilizadores sem autorização, que manipulavam, roubavam ou até destruíam esses mesmos dados. Neste projeto pretendemos mostrar como podemos deixar os sistemas mais seguros, para isso utilizámos um servidor com o Sistema Operativo Ubuntu 22.04.3 LTS, no qual implementámos várias soluções de segurança.

A nível de implementação, o nosso servidor Ubuntu está alojado numa instância da *Google Cloud*, com 1 Virtual CPU partilhado, 4Gb de memória RAM, e 10 Gb de disco. Tudo isto foi possível graças ao crédito de 300 dólares que a *Google Cloud* oferece durante 3 meses. Uma máquina virtual com estas características apresenta um custo de 30 dólares por mês, aproximadamente.

Para a criação do domínio foi utilizado o serviço da dominios.pt, onde é possível, durante um ano, obter um domínio '.pt' de forma gratuita. O domínio que foi criado pelo nosso grupo tem o nome de "fourkings.pt".

Com a criação do domínio já foi possível, de forma gratuita, obter um certificado TLS assinado e verificado, pela entidade "*let's encrypt*", permitindo-nos assim a criação de um Website que utiliza o protocolo HTTPS.

Para criar o Website foi utilizado como serviço Web o *Apache2* com *Wordpress*, permitindo com maior facilidade criar um Website Demo.

Para proteger a máquina Ubuntu foi configurada uma firewall utilizando o serviço iptables, as regras de modo a não serem aplicadas uma a uma pela *Command Line,* foram escritas num script *Bash* para trazer mais organização, flexibilidade e rapidez nas alterações que foram surgindo.

No sentido de testar a firewall utilizámos a ferramenta *Network MAP* que permitiu de forma eficaz verificar se as regras que foram implementadas se encontravam a funcionar.

De forma a trazer mais segurança para a máquina Ubuntu decidimos instalar o serviço *Unattended Upgrades*, que permite à máquina fazer os upgrades de segurança automaticamente assim que estes sejam disponibilizados.

Para que fosse possível aceder à máquina de forma mais segura foi instalado o serviço *fail2ban*, que permite bloquear um endereço IP, de um dispositivo, que tente entrar na máquina via SSH, durante algum tempo se o utilizador errar a palavra-passe mais que 5 vezes, também foram criadas chaves assimétricas SHA-256 que permitiram a partir de uma chave pública e privada aceder à máquina vis SSH sem a necessidade de utilizar uma password.

Ainda na vertente da segurança instalámos um serviço de um repositório *GIT* designado por *sshttps* onde é possível a partir de um mesmo porto aceder ao serviço *SSH* e *HTTP/HTTPS*.

Desenvolvimento

Setup do Servidor Ubuntu 22.04.3 LTS

Para criar uma instância na *Google Cloud* foi necessário criar uma conta de email Google, com a conta criada já é dada a possibilidade de usar o voucher de 300 dólares durante 3 meses.

Após isso, para criar uma instância, é necessário preencher um formulário com as características que pretendemos para a instância. Algumas das características são: o local físico onde a instância vai ser criada, o hardware virtual que a máquina vai utilizar, e também a configuração das regras da firewall externa da Google à nossa máquina. Com o formulário preenchido a máquina é criada e está pronta para começar a trabalhar

AVISO: Devido ao uso de uma firewall externa por parte da Google, pode não ser possível verificar algumas regras da firewall do Ubuntu *(iptables*). Um exemplo pode ser a sobrecarga de pacotes, que a google impede de imediato na sua firewall externa.

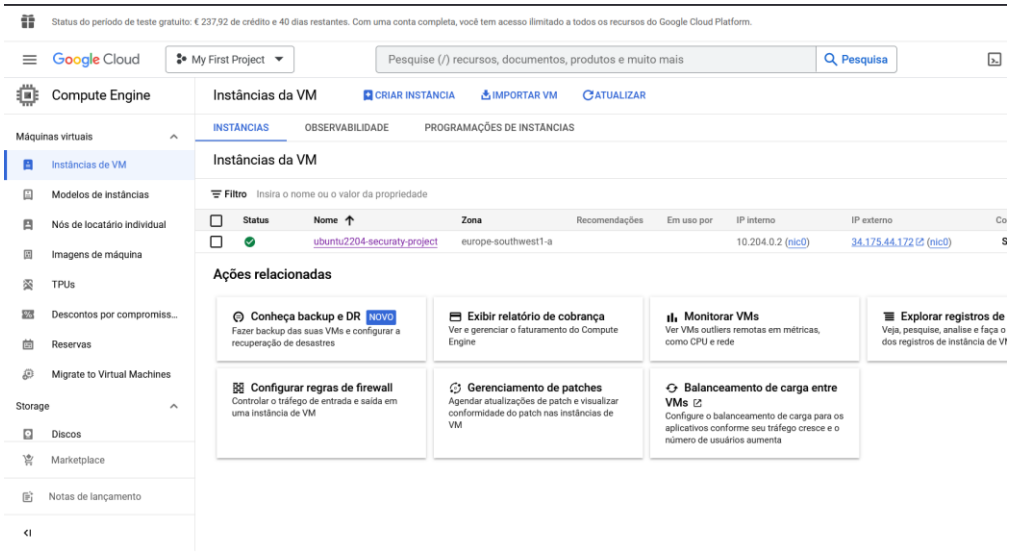

Fig. 1 Dashboard da Google Cloud para Controlo das Instâncias

Registo do Domínio

De modo a registar um domínio para o nosso site, recorremos ao domínios.pt que trabalha na área dos registos de domínios, alojamentos de Websites, e *housing* de servidores. Graças à sua oferta de um ano de domínio grátis foi possível criar o domínio “fourkings.pt” de forma gratuita.

Para criar um domínio nesta empresa é necessário criar uma conta e escolher o nome para o nosso domínio.

De forma a fazer alterações, ao nosso domínio de *DNS,* o domínios.pt apresenta uma *dashboard* interativa onde o utilizador pode fazer a gestão dos seus nomes e endereços IP’s. Após definir o nome e o IP, os servidores DNS começam a ser informados sobre a que IP corresponde aquele nome.

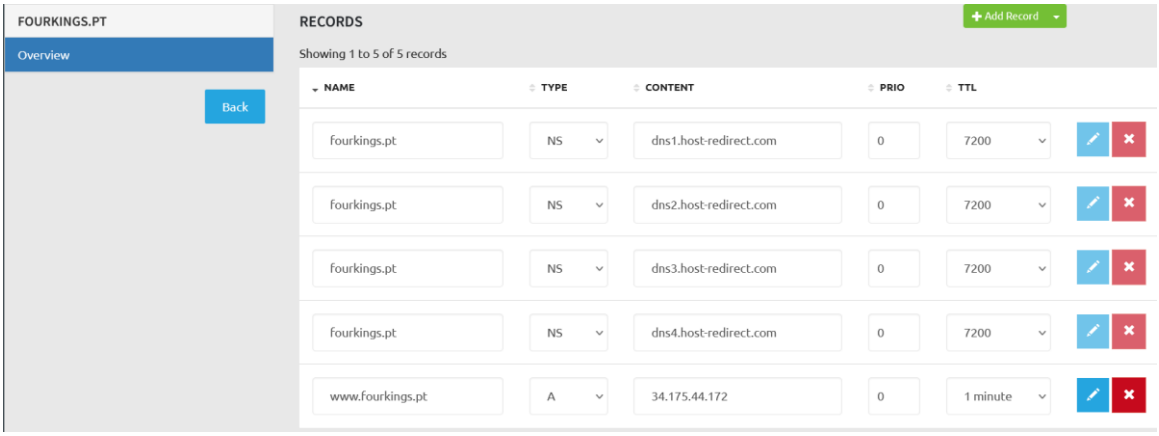

Fig. 2 Dashboard com os registos do domínio FOURKINGS.PT

Criação do certificado TLS

A fim de colocar o nosso Website a funcionar com o protocolo HTTPS, foi necessário pedir a uma entidade certificadora um certificado TLS válido. Para isso utilizamos o certbot para criar e instalar um certificado do let's Encrypt.

Foi necessário realizar várias etapas para permitir a criação e configuração do certificado.

1. Instalar o certbot e o script em python do certbot para facilitar a instalação

```
root@duarte:/home/duarte# apt install certbot python3-certbot-apache
```

Fig. 3 Comando para instalar o Certbot e o script para apache

2. Correr o seguinte comando para começar a criação do certificado

```
root@duarte:/home/duarte# certbot --apache -d www.fourkings.pt
```

Fig. 4 Comando para correr o script

3. Responder as questões do script

```
root@duarte:/home/duarte# certbot --apache -d www.fourkings.pt
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Enter email address (used for urgent renewal and security notices)
 (Enter 'c' to cancel): duarte@teste.pt

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.3-September-21-2022.pdf. You must
agree in order to register with the ACME server. Do you agree?
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(Y)es/(N)o: Y

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Would you be willing, once your first certificate is successfully issued, to
share your email address with the Electronic Frontier Foundation, a founding
partner of the Let's Encrypt project and the non-profit organization that
develops Certbot? We'd like to send you email about our work encrypting the web,
EFF news, campaigns, and ways to support digital freedom.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(Y)es/(N)o: N
Account registered.
Requesting a certificate for www.fourkings.pt
```

Fig. 5 Questões do script

4. Após respondidas as questões o certificado é criado e guardado no servidor. Para além disto, o script em python cria de imediato um ficheiro de configuração default para a diretoria /etc/apache2/sites-available e assim já é possível utilizar o ficheiro de configuração para um site do apache 2.

```
adminuser@ubuntu2204-securaty-project:/etc/apache2/sites-available$ ls -l | grep 000-default-le-ssl.conf
-rw-r--r-- 1 root root 1544 Oct 25 08:17 000-default-le-ssl.conf
```

Fig. 6 Ficheiro Default para sites HTTPS

```
<IfModule mod_ssl.c>
<VirtualHost *:8443>
        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html/wordpress

        <Directory /var/www/html/wordpress>
        Options FollowSymlinks
        AllowOverride All
        DirectoryIndex index.php
        Require all granted
        </Directory>

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        ServerName www.fourkings.pt
        SSLCertificateFile /etc/letsencrypt/live/www.fourkings.pt/fullchain.pem
        SSLCertificateKeyFile /etc/letsencrypt/live/www.fourkings.pt/privkey.pem
        Include /etc/letsencrypt/options-ssl-apache.conf
</VirtualHost>
</IfModule>
```

Fig. 7 Exemplo de um site com as configurações para HTTPS

Criação do Web Server

Na criação do Web Server, foi necessário instalar dois serviços, o Apache 2 e o Wordpress. O Apache é o serviço que vai lidar com os pedidos HTTP/HTTPS e vai responder aos clientes. O Wordpress é uma framework de código livre que permite de uma forma muito mais facilitada a criação de sites web.

Para instalar o Apache basta pela linha de comandos do Ubuntu usar a ferramenta apt para o Apache ser instalado.

.

```
duarte@duarte:~$ sudo apt install apache2
```

Fig. 8 Comando para instalar o Apache2

Depois da instalação foi necessário fazer alterações nos ficheiros principais do Apache, ports.conf, sites-available, etc.

```
<IfModule mod_ssl.c>
<VirtualHost *:8443>
        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html/wordpress

        <Directory /var/www/html/wordpress>
        Options FollowSymlinks
        AllowOverride All
        DirectoryIndex index.php
        Require all granted
        </Directory>

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        ServerName www.fourkings.pt
        SSLCertificateFile /etc/letsencrypt/live/www.fourkings.pt/fullchain.pem
        SSLCertificateKeyFile /etc/letsencrypt/live/www.fourkings.pt/privkey.pem
        Include /etc/letsencrypt/options-ssl-apache.conf
</VirtualHost>
</IfModule>
```

```
Listen 80

<IfModule ssl_module>
        #Listen 443
        Listen 8443 https
</IfModule>

<IfModule mod_gnutls.c>
        #Listen 443
        Listen 8443 https
</IfModule>
```

Fig. 9 Ficheiro Wordpress.Conf (ESQ) e ports.conf (DIR)

Para realizar a instalação do Wordpress é necessário antes, instalar a linguagem interpretada PHP versão 8 e também uma base de dados MySQL, no nosso caso utilizamos o MariaDB que é uma base de dados que vem do antigo MySQL.

Fig. 10 Comando para instalar o PHP (EM CIMA) e comando para instalar a Base de Dados MariaDB (EM BAIXO)

Para instalar o Wordpress é necessário transferir a partir do site oficial o seu ficheiro zip, e após isso deve-se colocar a pasta descompactada na diretoria /var/www/html/wordpress para que possa ser usada para o Apache.

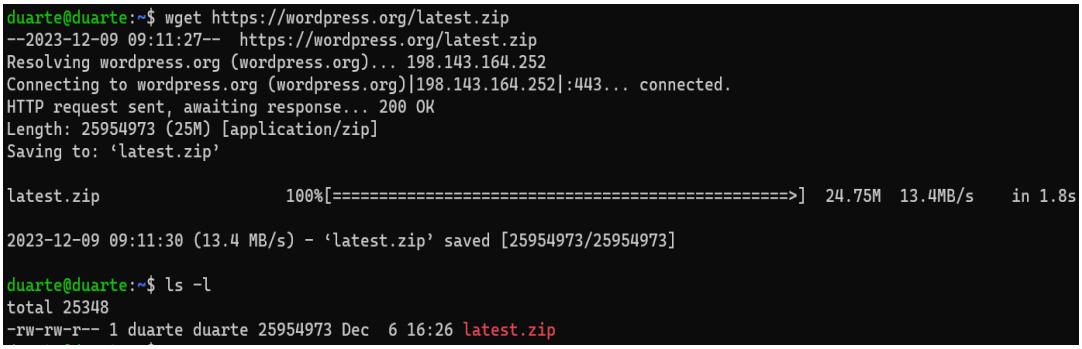

Fig. 11 Comando para transferir o ZIP do Wordpress

Após os seguintes passos, se não existirem erros já é possível ver a partir de um Browser a página Web a funcionar já a pedir para configurar o Wordpress.

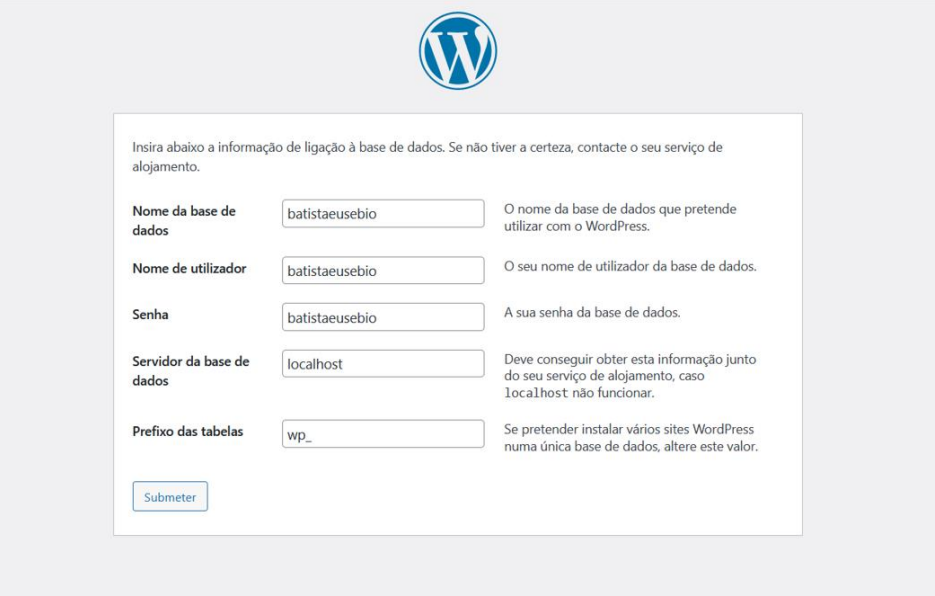

Fig. 12 Configurações pedidas pelo wordpress, nas primeiras interações

Dado que, a instalação tem mais pormenores, junto ao relatório vai um ficheiro com todos os passos que devem ser feitos para instalar o Wordpress.

Configuração da firewall

De modo a evitarmos potenciais ataques online, críamos uma firewall que apenas deixa entrar e sair alguns pacotes pelos seus portos abertos, para fazermos essa configuração usámos o IPTABLES que faz a filtragem de todos os pacotes que entravam e saiam da máquina descartando assim os pacotes que não cumpriam as regras.

Ao se executar a firewall, criámos uma regra que faz a limpeza de todas as regras anteriormente existentes do iptables e também uma regra que elimina as listas personalizadas do iptables, logo no incio do script , fizemos também a criação de uma política por omissão que define a regra principal do iptable, que caso os pacotes recebidos não estejam conforme as regras declaradas todos estes são descartados. Definimos também a firewall para:

- Permitir a interface loopback, para que esta seja capaz de funcionar.
- Criámos também uma lista personalizada que faz a proteção de ataques por flood, isto é, evita que haja uma sobrecarga de comunicações ICMP, TCP e UDP.
- Criámos uma regra que faz a criação de logs para todos os pacotes inválidos que entram e que saem e faz também a sua respetiva negação.
- Criámos uma regra que faz o log para todo o tráfego input, que já tenham uma ligação iniciada com o servidor.
- Elaborámos regras que permitem a entrada e saída de pacotes que já tenham uma ligação estabelecida com o servidor.
- Elaborámos uma lista personalizada que realiza um LOG de todo o tráfego INPUT, e também faz a autorização dos pacotes ICMP, HTTP, HTTPS e SSHTTP.

- Por fim elaborámos uma lista personalizada que aceita os pacotes de saída (OUTPUT) que tenham o estado NOVO, para os pacotes de saída dos seguintes protocolos PING, DNS,DNS over TLS, SSH, Docker, whois, http e https.

Configuração do sshttps

O sshttp é um programa que serve para, a partir de um porto do nosso sistema operativo, realizar comunicações de 2 protocolos diferentes nomeadamente o HTTPS e o SSH.

Isto é possível devido ao código que é executado, que verifica no pacote que entra qual é o cabeçalho layer 7 do modelo OSI, para perceber se a comunicação é HTTPS ou SSH.

A razão pela qual o programa funciona é devido ao reencaminhamento interno que é feito aos pacotes, estes ao entrarem no porto 443 são processados e são reencaminhados internamente para o porto do serviço a que corresponde o pacote.

AVISO: Para que seja possível executar este programa é necessário primeiro alterar o porto em que o Apache escuta comunicações HTTPS pois por default o sshttps vai escutar comunicações no porto 443

Para instalar e configurar o sshttp foi necessário em primeiro lugar realizar o git clone do repositório no nosso servidor.

```
duarte@duarte:~$ git clone https://github.com/stealth/sshttp.git
Cloning into 'sshttp'...
remote: Enumerating objects: 255, done.
remote: Counting objects: 100% (16/16), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 255 (delta 5), reused 16 (delta 5), pack-reused 239
Receiving objects: 100% (255/255), 114.55 KiB | 1.10 MiB/s, done.
Resolving deltas: 100% (151/151), done.
```

Fig. 13 Git clone do repositorio sshttp

De modo a colocar o código do sshttps a funcionar foi também necessário instalar 2 packages extras, o libcap-dev e o build-essential.

```
duarte@duarte:~$ sudo apt install libcap-dev build-essential
```

Fig. 14 Instalação das dependências

Com a instalação feita destes 2 pacotes, foi necessário compilar o programa e alterar algumas configurações do programa, nomeadamente o ficheiro sshttp/iptables/nf-setup onde indicámos qual é a interface que vai ouvir os pedidos e os portos dos serviços SSH e HTTPS, e também no Apache o ficheiro ports.conf onde indicámos o novo porto que vai escutar ligações HTTPS e ainda no ficheiro do site para indicar o novo porto em que o HTTPS vai ouvir.

Fig. 15 Compilação do programa na esquerda e modificação no ficheiro nf-setup na direita

Fig. 16 Alteração dos ficheiros do Apache, na esquerda o ficheiro ports.conf, e na direita o ficheiro do sites-available

De seguida para colocar o programa a funcionar devemos executar o ficheiro sshttpd que vai ser o dameon que vai verificar qual é o tipo de tráfego enviado pelo pacote (SSH ou HTTPS). E por fim executar o ficheiro nf-setup para este configurar e instalar as regras proxy.

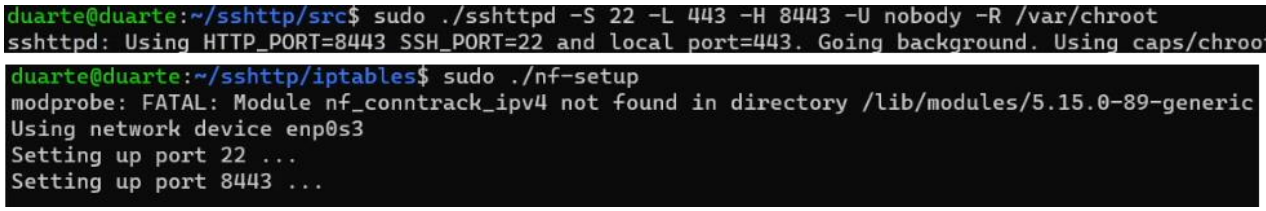

Fig. 17 Execução do programa sshttpd em cima e execução do nf-setup em baixo

A partir do momento em que estes comandos forem executados já é possível estabelecer comunicação SSH e HTTPS pelo porto 443.

Configuração SSH

Na configuração do SSH foram efetuadas várias alterações. No ficheiro de configuração base do SSH foram feitas as seguintes alterações: o SSH apenas comunica por IPv4, para entrar na máquina por login são dados apenas 60 segundos para colocar a password, é possível fazer login na máquina com a conta root, e ainda é possível realizar a autenticação no servidor a partir do uso de uma palavra-passe.

```
# Configuracoes Enderecos
#Port 22
#AddressFamily any
AddressFamily inet
ListenAddress 0.0.0.0
#ListenAddress ::
```

```
# To disable tunneled clear text passwords, change to no here!
#Permite Autenticacao com Password
PasswordAuthentication yes
#PermitEmptyPasswords no
```

```
#Tempo que o servidor permite para colocar a password antes de desconectar
LoginGraceTime 1m
#Permite o uso de password para a conta Root
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

Fig. 18 Configurações feitas no ficheiro sshd_config

De modo a trazer mais segurança também foram criadas chaves assimétricas para permitir não só um login mais seguro como também um login mais rápido.

A criação de chaves foi um processo simples, usamos Powershell do Windows para criar o par de chaves, a chave privada foi colocada na pasta .ssh do utilizador do Windows, e a chave pública foi guardada na pasta .ssh do utilizador do servidor Ubuntu.

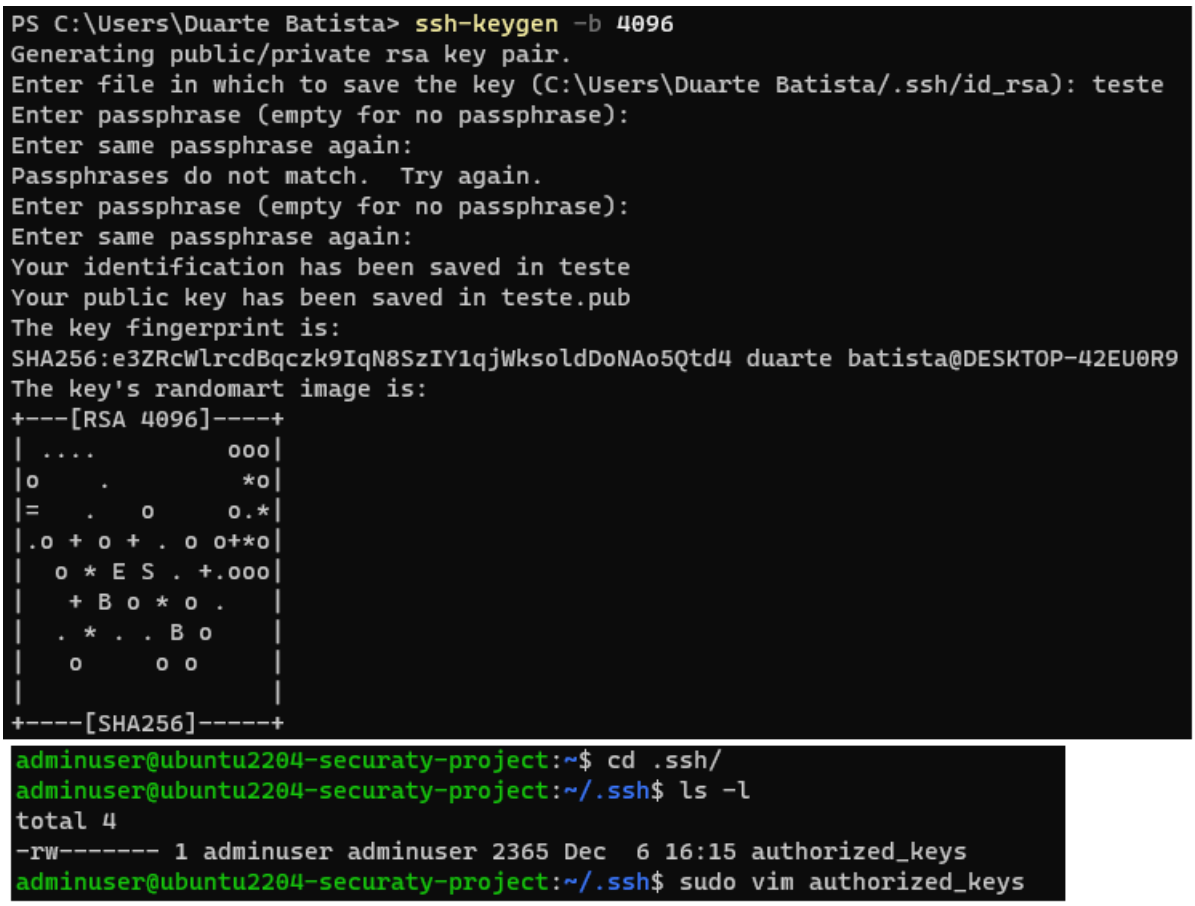

Fig. 19 Comando para criar as chaves no Windows em cima e ficheiro no servidor onde deve ser colocada a chave publica do par de chave em baixo

Para além desta configuração no SSH, também instalámos o serviço fail2ban que permite bloquear o endereço IP de um dispositivo que tente se conectar por SSH ao nosso servidor, e erre a palavra passe mais do que 5 vezes, o tempo que fica bloqueado são 10 minutos.

Fig. 20 Comando para Instalar o fail2ban

Instalação do Unattended-Upgrades

Uma das principais razões pelo qual os servidores e os computadores são atacados deve-se ao esquecimento de instalar as atualizações mais recentes de segurança nesses dispositivos, o que deixa uma porta aberta para Hackers atacarem esses dispositivos. Devido a isto, o unattended-upgrades é

um serviço que de forma automática, assim que sai uma atualização de segurança estável, instala-a de imediato no nosso servidor. Para instalar basta usar a ferramenta apt do Ubuntu.

```
duarte@duarte:~$ sudo apt install unattended-upgrades
```

Fig. 21 Comando para instalar o Unattended-upgrades

Testes

Verificação da resolução do nome do Domínio “www.fourkings.pt”

Para testar se a resolução do nome “www.fourkings.pt” estava a funcionar foi efetuado o comando nslookup.

```
Non-authoritative answer:
Name:    www.fourkings.pt
Address:  34.175.44.172
```

Fig. 22 Output do comando nslookup

Verificação do fail2ban

Para testar se o fail2ban estava a funcionar, erramos propositadamente a palavra-passe várias vezes ao tentarmos se conectar por ssh ao servidor, no final o servidor não deu resposta.

```
duarte@duarte-VirtualBox:~$ ssh adminuser@34.175.44.172 -p 443
(adminuser@34.175.44.172) Password:
(adminuser@34.175.44.172) Password:
(adminuser@34.175.44.172) Password:
adminuser@34.175.44.172's password:
Permission denied, please try again.
adminuser@34.175.44.172's password:
Permission denied, please try again.
adminuser@34.175.44.172's password:
Received disconnect from 34.175.44.172 port 443:2: Too many authentication failures
Disconnected from 34.175.44.172 port 443
duarte@duarte-VirtualBox:~$ ssh adminuser@34.175.44.172 -p 443
```

Fig. 23 Tentativas de login falhadas por ssh ao servidor

Verificação do certificado Let’s Encrypt

Para verificar se o certificado estava a funcionar foi usado o Browser, e nele verificamos o certificado do site.

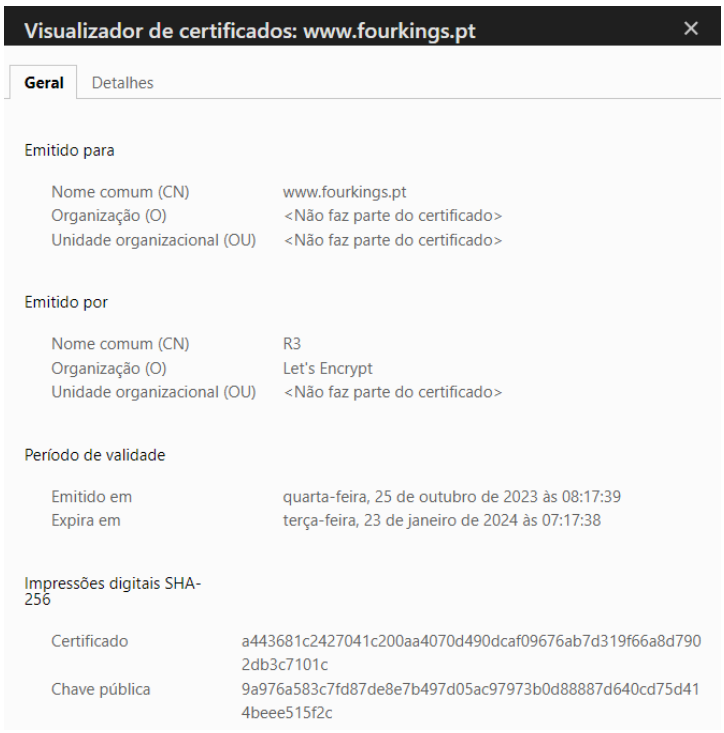

Fig. 24 Detalhes do Certificado www.fourkings.pt

Verificação do site Apache em Wordpress

Para verificar se o site estava a funcionar foi usado o Browser, e nele navegamos pela página Web

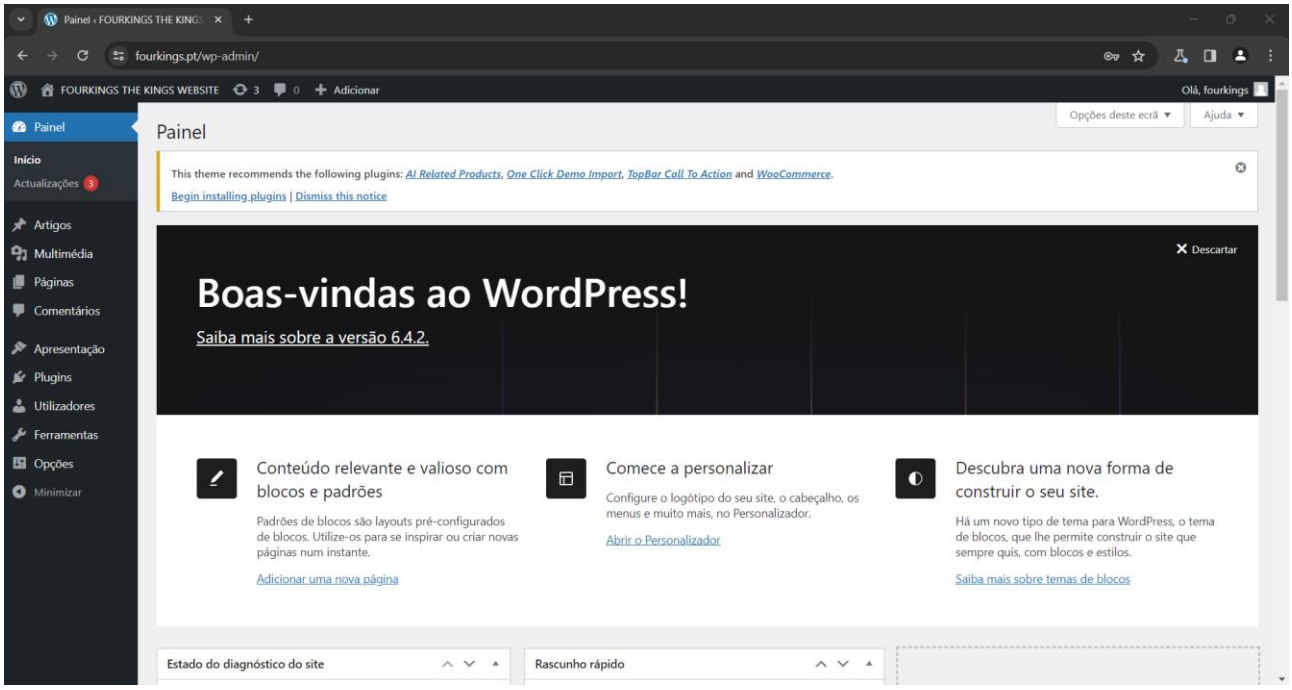

Fig. 25 Página de administração do site "www.fourkings.pt"

Verificação do sshttps

Para verificar se o sshttps estava a funcionar usamos o browser para aceder ao site pelo porto default do protocolo HTTPS 443 e acedemos ao servidor por SSH pelo porto 443.

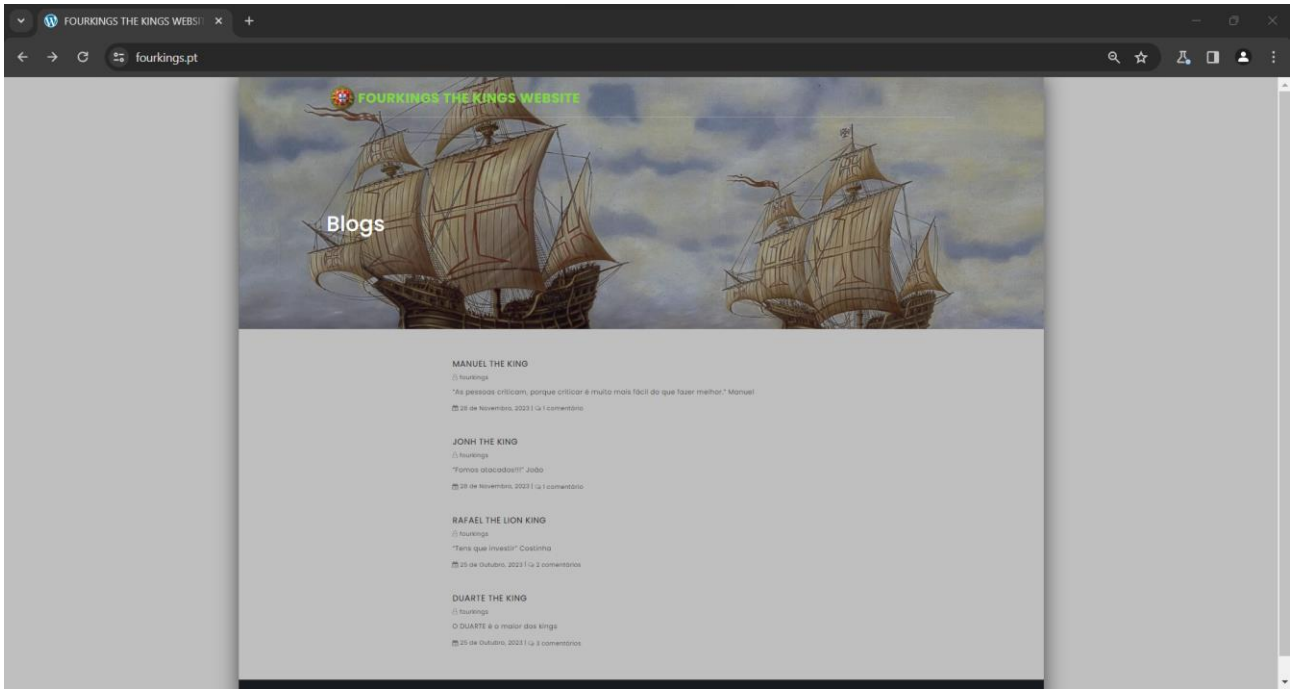

Fig. 26 Página principal do site "www.fourkings.pt"

```
PS C:\Users\Duarte Batista> ssh adminuser@34.175.44.172 -p 443
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1019-gcp x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sat Dec  9 16:36:33 UTC 2023

  System load:  0.0               Processes:             113
  Usage of /:   57.0% of 9.51GB   Users logged in:       0
  Memory usage: 11%               IPv4 address for ens4: 10.204.0.2
  Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

3 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm


*** System restart required ***
Last login: Sat Dec  9 15:09:46 2023 from 82.155.239.79
adminuser@ubuntu2204-securaty-project:~$
```

Fig. 27 Acesso por SSH ao servidor pelo porto 443

Limitações

Ao longo do projeto existiram várias dificuldades, mas todas elas foram ultrapassadas. Conseguimos implementar tudo o que era pedido no projeto. A dificuldade que mais nos deu dores de cabeça foi no serviço sshttps, a funcionar em conjunto com o Wordpress. O Wodpress guarda na sua base de dados o URL do site e do domínio onde é configurado, e ao fazer as alterações no porto em que o HTTPS escutava, fez com que o site não trabalhasse corretamente, após várias tentativas descobrimos que tínhamos que alterar determinados parâmetros nas definições do Wordpress. Para além disto o guia de passos a ter para configurar o sshttp não se encontrava explícito o que tornou ainda mais difícil conseguir executar e configurar este serviço.

Conclusão

Para concluir, queremos em primeiro lugar agradecer ao professor Miguel Monteiro de Sousa Frade pela oportunidade de estudar os conceitos da disciplina de uma forma mais aprofundada e prática. Esperamos ter estado à prova do desafio proposto pelo professor. Saímos deste projeto orgulhosos e com o sentimento de dever cumprido. Este projeto ensinou-nos não só a como deixar uma máquina mais segura, mas também nos ensinou a superar as dificuldades colocadas pela interligação destes vários mecanismos de segurança. A partir deste momento já conseguimos deixar os sistemas um “bocadinho” mais seguros.