

SEGURANÇA DE SISTEMAS

2020/2021

Fábio Henriques¹, Gonçalo Vicente², Matias Luna³, Miguel Reis⁴, Steven Lincango⁵

¹ 2181359@my.ipleiria.pt, Engenharia Informática, Diurno

² 2172131@my.ipleiria.pt, Engenharia Informática, Diurno

³ 2182082@my.ipleiria.pt, Engenharia Informática, Diurno

⁴ 2160804@my.ipleiria.pt, Engenharia Informática, Diurno

⁵ 2180962@my.ipleiria.pt, Engenharia Informática, Diurno

Resumo: O presente documento descreve os trabalhos desenvolvidos no âmbito da unidade curricular Segurança de Sistemas pertence à licenciatura de Engenharia Informática e descreve a configuração de serviços, utilizando maioritariamente versões TLS. Estes serviços encontram-se configurados em 5 máquinas diferentes, utilizando o Virtual Box e encontram-se interligados através de uma VPN.

Palavras-chave: Segurança, TLS, VPN, VirtualBox.

1. Introdução

A segurança dos sistemas é um dos tópicos mais importantes na atualidade. Sem ela todos os sistemas dos quais somos dependentes não existiam. Com o evoluir das tecnologias, a segurança aumenta, mas a capacidade de quebrá-la também, sendo que, por vezes, acontece a um ritmo superior.

Neste projeto pretendemos demonstrar configurações seguras dos serviços mais utilizados (DNS, HTTP, Mail) e também implementar ligações seguras (VPN) tanto entre clientes-servidores como entre os próprios servidores, fisicamente distantes.

Temos definido implementar os serviços de DNS over TLS, HTTPS com certificado *Let's Encrypt*, um servidor para backups automatizados e centralizados com o “bacula”, uma *firewall* de filtro de pacotes (pfSense) e uma de *proxy* (squid), uma base de dados MySQL e um servidor de email. Para a interligação dos cenários de forma segura, configuramos uma VPN “Wireguard” e utilizamos uma máquina “ubuntu” para servir de *gateway* do tráfego da VPN para a “pfSense”.

Em termos de rede idealizamos possuir duas: uma interna e uma DMZ que contém os serviços de email e HTTPS.

2. Desenho da rede

Como não é possível implementar o cenário fisicamente, decidimos utilizar uma máquina na cloud para servir de servidor VPN, utilizando o “Wireguard”. Nela foram configuradas duas interfaces para a VPN: wg0, que interliga a rede interna (LAN) e a wg1 para a rede DMZ.

Todos os servidores ligam-se ao servidor VPN na respetiva interface e o seu tráfego é encaminhado sempre para o endereço .2 da rede a que se ligam. Este está ligado a uma *firewall* “pfSense” que filtra todo o tráfego e faz, respetivo *port forwarding* para os servidores da DMZ e para o serviço SSH de cada servidor (esteja na DMZ ou na LAN).

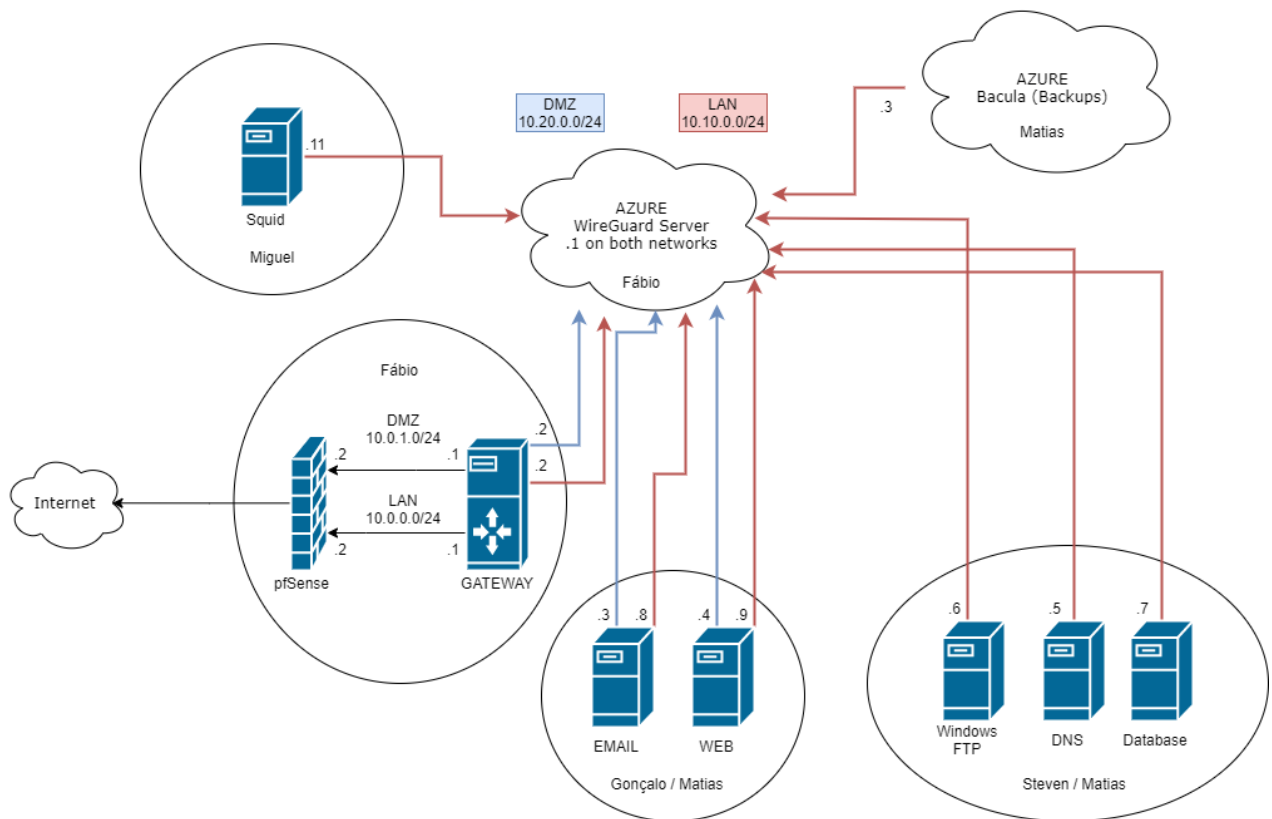


Figura 1 - Rede

Sempre que não for indicado o SO da máquina, significa que esse SO é “Ubuntu”.

Serviços de segurança

Na nossa implementação pretendemos garantir os seguintes serviços de segurança:

- **Disponibilidade**, através do *port forwarding* para os serviços acessíveis do exterior e da aplicação “fail2ban”, que impede ataques de força bruta nos serviços configurados;
- **Confidencialidade, integridade e autenticação** através da VPN e do uso de um certificado digital no servidor Web;
- **Controlo de acessos** através da firewall.

3. Configuração das firewalls

A firewall “pfSense” está configurada da seguinte forma:

- Três interfaces: uma WAN ligada por *bridge*, uma para a rede LAN e outra para a rede DMZ;
- Apenas permite tráfego de saída;
- Contém regras de *port forwarding* para permitir o tráfego de entrada HTTP, HTTPS, SMTP, POP3, IMAP e SSH para os servidores da LAN e DMZ.

Port Forward

1:1

Outbound

NPt

Rules

































<input type="checkbox"/>		Interface	Protocol	Source Address	Source Ports	Dest. Address	Dest. Ports	NAT IP	NAT Ports	Description	Actions	
<input type="checkbox"/>	<input checked="" type="checkbox"/>		WAN	TCP	*	*	WAN address	2020	10.0.1.1	22 (SSH)	SSH to Default Gateway	  
<input type="checkbox"/>	<input checked="" type="checkbox"/>		WAN	TCP	*	*	WAN address	2021	10.0.1.1	2021	SSH to EMAIL	  
<input type="checkbox"/>	<input checked="" type="checkbox"/>		WAN	TCP	*	*	WAN address	2022	10.0.1.1	2022	SSH to WEB	  
<input type="checkbox"/>	<input checked="" type="checkbox"/>		WAN	TCP	*	*	WAN address	2023	10.0.1.1	2023	SSH to BACULA	  
<input type="checkbox"/>	<input checked="" type="checkbox"/>		WAN	TCP	*	*	WAN address	80 (HTTP)	10.0.1.1	80 (HTTP)	Port Forward HTTP	  
<input type="checkbox"/>	<input checked="" type="checkbox"/>		WAN	TCP	*	*	WAN address	443 (HTTPS)	10.0.1.1	443 (HTTPS)	Port Forward HTTPS	  
<input type="checkbox"/>	<input checked="" type="checkbox"/>		WAN	TCP	*	*	WAN address	25 (SMTP)	10.0.1.1	25 (SMTP)	Port Forward SMTP	  
<input type="checkbox"/>	<input checked="" type="checkbox"/>		WAN	TCP	*	*	WAN address	110 (POP3)	10.0.1.1	110 (POP3)	Port Forward SMTP	  

Figura 2 - Algumas regras de port forwarding na pfSense

Como utilizamos o “Wireguard” como VPN e o “pfSense” não possibilita a ligação com essa VPN, foi necessário implementar um servidor “Ubuntu” para servidor de *router* entre a “pfSense” e as redes na VPN. Nesse servidor foi implementado o *port forwarding* com recurso à *firewall* “iptables”.

Com essa ferramenta, fizemos uso da tabela *nat* na *chain* PREROUTING para alterar o endereço de destino do pacote proveniente da “pfSense”. Para ser possível enviar o tráfego da VPN para a “pfSense”, utilizamos a função MASQUERADE no tráfego que sai das interfaces de ligação.

Também foi necessário criar duas tabelas de encaminhamento novas para encaminhar o tráfego proveniente da VPN.

```
#!/bin/bash
#SSH - Gonçalo
iptables -t nat -A PREROUTING -i enp0s3 -p tcp -m tcp --dport 2021 -j DNAT --to-destination 10.20.0.3:22
iptables -t nat -A PREROUTING -i enp0s3 -p tcp -m tcp --dport 2022 -j DNAT --to-destination 10.20.0.4:2221
#SSH - Matias
iptables -t nat -A PREROUTING -i enp0s3 -p tcp -m tcp --dport 2023 -j DNAT --to-destination 10.10.0.3:22
```

Figura 3 - Parte da script para o port forwarding no servidor Ubuntu

```
PostUp = iptables -t nat -A POSTROUTING -o enp0s8 -j MASQUERADE
PostUp = ip rule add from 10.10.0.0/24 table lan
PostUp = ip route add default via 10.0.0.2 table lan
PostUp = ip route add to 10.10.0.0/24 via 10.10.0.1 table lan
```

Figura 4 - MASQUERADE e criação das rotas para o encaminhamento da rede LAN da VPN

As tabelas de encaminhamento de todos os servidores tiveram de ser alteradas para encaminhar todo o tráfego pela VPN.

```
PostUp = ip route add 51.124.106.163 via 10.0.2.2
PostUp = ip route del default via 10.0.2.2
PostUp = ip route add default via 10.20.0.1
```

Figura 5 - Alteração da tabela de encaminhamento

4. Serviços instalados

Backups Automáticos e Centralizados

De forma a ser possível implementar um serviço de recuperação em caso de falhas foi configurado um servidor Linux Ubuntu Server 18.04 com recurso à ferramenta Bacula. O servidor faz parte da rede interna 10.0.0.0/24 que depois responde ao endereço público 20.67.35.13, tem 2 CPU's e 8 GB de RAM.

Neste servidor foi configurada a versão cliente do Wireguard para que o servidor possa ter acesso ao resto dos servidores da rede.

O báculo é uma solução muito útil quando se têm vários servidores espalhados pela rede que costumam estar ligados a maior parte do tempo e assim a utilização desta ferramenta poupa trabalho aos administradores dos servidores já que o servidor bacula encarrega-se de realizar os backups dos dados de forma remota.

Algumas das regras configuradas para o serviço foram:

- Backups automáticos feitos a cada meia hora

A primeira vez que um backup é feito a uma máquina, realiza-se uma cópia total dos dados, e depois cada vez que é executado, é feito um backup incremental.

Por motivos de testes, configurou-se para que o servidor realize backups cada 30 minutos, embora este valor possa ser outro, como por exemplo o primeiro domingo de cada mês um backup Full e incremental o resto dos dias, sempre às 3:00). Estes são feitos de forma automática porque são programados pela própria ferramenta baseada nas nossas configurações.

Quando um dos jobs não é realizado, por qualquer razão o servidor volta a tentar depois de 5 minutos, faz 2 tentativas após a primeira falha e depois descarta o backup. Quando é executado corretamente, o backup é sempre feito da pasta /home, incluindo a pasta do próprio servidor de backup's, sendo que todos os backup's executados manter-se-ão guardados durante 365 dias, sendo depois reciclados.

Infelizmente não existe uma opção no bacula que permita fazer um trabalho (job) de backup para diferentes clientes, mas sim podemos efetuar vários jobs à mesma hora e portanto, teremos a mesma experiência, pelo que foi configurado um job para cada cliente e como respondem ao mesmo horário, são todos efetuados à mesma hora, garante-se que os backups de todos os clientes sejam realizados com normalidade pelo bacula.

```
Terminal Shell Edição Visualização Janela Ajuda
Projeto — root@backup: /bacula/backups — ssh -i backupkeys.pem backupadmin@
BackupServerSS-dir Version: 9.0.6 (20 November 2017) x86_64-pc-linux-gnu ubuntu 18.04
Daemon started 16-Jan-21 21:35, conf reloaded 16-Jan-2021 21:35:56
Jobs: run=1, running=0 mode=0,0
Heap: heap=278,528 smbytes=116,652 max_bytes=146,113 bufs=401 max_bufs=444
Res: njobs=10 nclients=7 nstores=2 npools=4 ncats=1 nfsets=4 nscheds=3

Scheduled Jobs:
=====
Level      Type      Pri  Scheduled      Job Name      Volume
=====
Incremental Backup    15  16-Jan-21 22:30  BackupGatewayServer VolumeSS-0003
Incremental Backup    15  16-Jan-21 22:30  BackupFTPServer   VolumeSS-0003
Incremental Backup    15  16-Jan-21 22:30  BackupDBServer    VolumeSS-0003
Incremental Backup    15  16-Jan-21 22:30  BackupDNSServer   VolumeSS-0003
Incremental Backup    15  16-Jan-21 22:30  BackupMailServer  VolumeSS-0003
Incremental Backup    15  16-Jan-21 22:30  BackupWebServer   VolumeSS-0003
Full        Backup    11  16-Jan-21 23:10  BackupCatalog     Local-0001
Differential Backup    10  17-Jan-21 03:00  BackupLocalFiles  Local-0001
=====

Running Jobs:
Console connected at 16-Jan-21 21:49
No Jobs running.
=====

Terminated Jobs:
=====
JobId Level  Files  Bytes  Status  Finished      Name
=====
245  Incr      0      0    OK      16-Jan-21 17:30 BackupDBServer
246  Incr      0      0  Error   16-Jan-21 17:30 BackupFTPServer
247  Incr      9  5.692 K OK      16-Jan-21 21:30 BackupGatewayServer
248  Incr      3  1.329 K OK      16-Jan-21 21:30 BackupWebServer
249  Incr     15  3.594 K OK      16-Jan-21 21:30 BackupMailServer
253  Full     22  79.10 K OK      16-Jan-21 21:30 BackupLocalFilesU
250  Incr      2  1.031 K OK      16-Jan-21 21:30 BackupDNSServer
251  Incr      3   200 K OK      16-Jan-21 21:30 BackupDBServer
252  Incr      1      0 OK      16-Jan-21 21:30 BackupFTPServer
254  Incr      2  2.416 K OK      16-Jan-21 21:36 BackupLocalFilesU
=====
*
```

Figura 6 - Trabalhos (Jobs/Backups) da ferramenta bacula

- Integração com Windows.

Não existe uma versão servidor da ferramenta para o sistema Windows, segundo a página oficial do Bacula.

Quando instalada e configurada a versão cliente do Bacula para Windows está começa a funcionar sempre que a máquina é iniciada.

Foi necessária esta configuração para o servidor FTP da rede.

- Ligações encriptadas

Para a configuração do TLS no bacula foi referida à documentação na sua página oficial.

Apesar dos esforços para tornar as comunicações mais seguras não se chegou a por funcional, porque as ligações não resultavam com um certificado auto-assinado e por tanto era preciso um certificado de uma CA válida, pelo que com certificados gerados pelo próprio servidor não era possível.

Portanto, pensou-se na possibilidade de pedir um certificado válido com recurso ao serviço LetsEncrypt, mas como era necessário um domínio para o servidor, a configuração não se chegou a fazer.

```
TLS CA Certificate File = /etc/ssl/certs/bacula_ca.crt
TLS Certificate = /etc/ssl/certs/bacula_server.crt
TLS Key = /etc/ssl/private/bacula_server.key
```

Figura 7 - Configurações adicionais para ligações seguras no Bacula

- Encriptação dos dados

Visto que as configurações de TLS só encriptam as comunicações, os dados continuam desprotegidos quando chegam ao servidor já que são guardados sem encriptação, para isso a ferramenta têm um mecanismo de encriptação, baseado em chaves, a configuração é feita no cliente, ou seja, antes de enviar os dados ao servidor, estes são encriptados, pelo que chegam ao servidor encriptados e são salvaguardados de forma segura.

É importante referir que o conteúdo é o cifrado, sendo que os metadados (nomes, permissões, dono, etc.) mantem-se visíveis ao utilizador.

```
Build OS: x86_64-pc-linux-gnu ubuntu 18.04
JobId: 254
Job: BackupLocalFilesU.2021-01-16_21.36.08_03
Backup Level: Incremental, since=2021-01-16 21:30:04
Client: "BackupServerSS-fd" 9.0.6 (20Nov17) x86_64-pc
linux-gnu,ubuntu,18.04
FileSet: "FilesTest" 2021-01-16 21:30:00
Pool: "FileUnenc" (From Job resource)
Catalog: "MyCatalog" (From Client resource)
Storage: "File1" (From Job resource)
Scheduled time: 16-Jan-2021 21:36:07
Start time: 16-Jan-2021 21:36:10
End time: 16-Jan-2021 21:36:10
Elapsed time: 1 sec
Priority: 10
FD Files Written: 2
SD Files Written: 2
FD Bytes Written: 2,416 (2.416 KB)
SD Bytes Written: 3,179 (3.179 KB)
Rate: 2.4 KB/s
Software Compression: None
Comm Line Compression: 0.9% 1.0:1
Snapshot/VSS: no
Encryption: yes
Accurate: no
Volume name(s): file-unenc0004
Volume Session Id: 8
Volume Session Time: 1610830932
Last Volume Bytes: 3,930 (3.930 KB)
Non-fatal FD errors: 0
SD Errors: 0
FD termination status: OK
SD termination status: OK
Termination: Backup OK
```

Figura 8 - Estado final de um trabalho (job/backup) com encriptação

Serviço DNS

Para a resolução de nomes foi configurado o serviço DNS numa máquina Ubuntu Server 18.04 com o serviço bind9 e, de forma a manter a integridade dos dados da resposta do DNS foi configurado uma extensão de segurança do DNS, o DNSSEC.

DNSSEC junta todos os registos da zona usando uma chave publica e para gerar as chaves foi usado o ZSK e KSK, logo assinaremos a zona de nosso domínio com as chaves criadas.

Configurações:

Depois de termos configurado o bind9 temos de ativar o DNSEC, no ficheiro `/etc/bind/named.conf.options` na zona de options.

```
dnssec-enable yes;  
dnssec-validation yes;
```

Figura 9 - Ativar o DNSEC

Para gerir as chaves do ZSK utilizamos o comando `dnssec-keygen -a <ALGORITHM> -b <BITS> -n ZONE <ZONENAME>`, onde o algoritmo escolhido para a encriptação da chave podem ser dois **RSASHA256** e **ECDSAP256SHA256**, sendo gerado dois ficheiros um `.key` que é a chave publica e outro `.private` que é a chave privada, sendo a opção `-b` o tamanho e `zonename` é a nossa zona DNS.

Para gerar o KSK é muito similar ao ZSK, sendo necessário utilizar o comando `dnssec-keygen -f KSK -a <ALGORITHM> -b <BITS> -n ZONE <ZONENAME>`.

```
root@projetoss:/etc/bind/keys# ls -l  
total 16  
-rw----- 1 root bind 3316 Jan 16 12:21 Kprojetoss.pt.+008+33811.private  
-rw-r--r-- 1 root bind 953 Jan 16 12:21 Kprojetoss.pt.+008+33811.key  
-rw----- 1 root bind 1776 Jan 16 12:23 Kprojetoss.pt.+008+03615.private  
-rw-r--r-- 1 root bind 607 Jan 16 12:23 Kprojetoss.pt.+008+03615.key  
root@projetoss:/etc/bind/keys#
```

Figura 10 - Ficheiros criados com as chaves ZSK e KSK

Por fim, resta fazer a assinatura manual de nossa zona com o comando `dnssec-signzone -o <nome de zona> -N INCREMENT -t -k <KSK> <ZSK>`, e é gerado um novo ficheiro `.signed` que temos que modificar nas zonas do bind.

```
zone "projetoss.pt" {  
    type master;  
    file "/etc/bind/zones/db.projetoss.pt.signed";  
};
```

Figura 11 - Modificação das zonas do bind

Ao fazermos um dig a ftp.projetoss.pt utilizando o dnssec, obtivemos a respostas que podemos visualizar na figura 9.

```
;; ANSWER SECTION:  
ftp.projetoss.pt.      604800 IN      A      10.10.0.6  
ftp.projetoss.pt.      604800 IN      RRSIG   A 8 3 604800 20210215113028 2  
0210116113028 3615 projetoss.pt. g87RQLJ4N1qPg4OQL7JJwsLB1ch4B5yi2jALQK76430s  
DwJFU6Q12A0T W2A0iO/j6yhHScslw5JmlitLXUS+RefnZ+qXmCTDhJ97Y7Lk2ZJ/IDD4 J1/9UiF  
eM6yC9RUht7tfNgo4lreJlnkYRNxkiM7tAXe0/576bN3bk58f kjloEgGvnc0/O9IDrW6EYEu5VwI  
5sjCVxvGNDGxJcqFg3o4LdSGUswet g3liOjNCv3qEF2/jlyRQZw3nFlo6/MOrC4ygUV4gOvtQmpr  
la7zN0hYq ZI77Wr8PprHVAzy2zIiu2aW75DeYt42mrJC3/wdF3ZjwulqWRAaSglek hoEEtg==
```

Figura 12 - Resposta a dig ftp.projetoss.pt +dnssec

Durante a configuração do DNS over TLS existiu alguns problemas. Foi tentado configurar com o serviço de stubby, o stub resolved stubby que é um pequeno cliente de DNS que recebe e reenvia pedidos DNS cifrados. Outra forma de dar segurança ao serviço DNS é com o HTTP, com os serviços de nginx e do apache2, porém essas configurações não foram exploradas, mas é uma boa opção para ter uma maior segurança no nosso servidor de DNS.

Serviço FTP

Para o servidor de FTP foi usado um Windows Server 2019, com o serviço de Internet Information Service (IIS) Manager. O FTP é um protocolo de rede para transferência de ficheiros baseado numa arquitetura de cliente-servidor, o cliente pode-se ligar ao servidor para a descarregar ou carregar ficheiros. Para a implementação do SFTP, a versão segura do serviço FTP, é necessário o serviço de OpenSSH-Server já que é uma extensão do protocolo SSH e todos os pacotes que são enviados estão protegidos por este protocolo.

De forma a instalarmos o serviço é necessário aceder ao Windows Manager e adicionar um novo “roles and features”, logo em tools podemos ingressar ao servidor

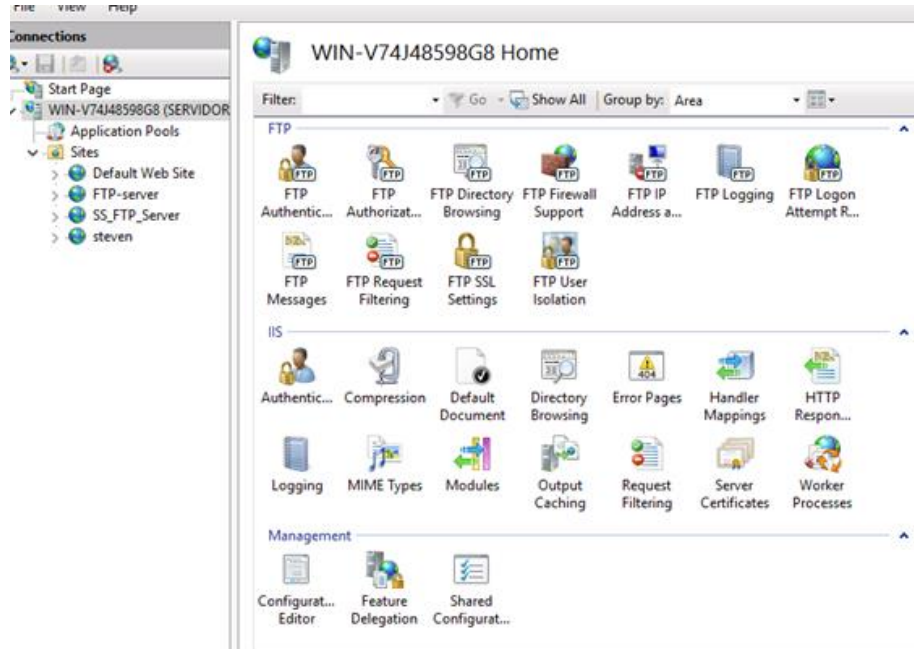


Figura 13 – Serviço FTP

Adicionamos um novo site FTP e escolhemos a pasta que queremos partilhar, colocamos a ip de nosso servidor e porto em que vai ser ficar à espera de pedidos e também temos a opção de colocar um certificado SSL.

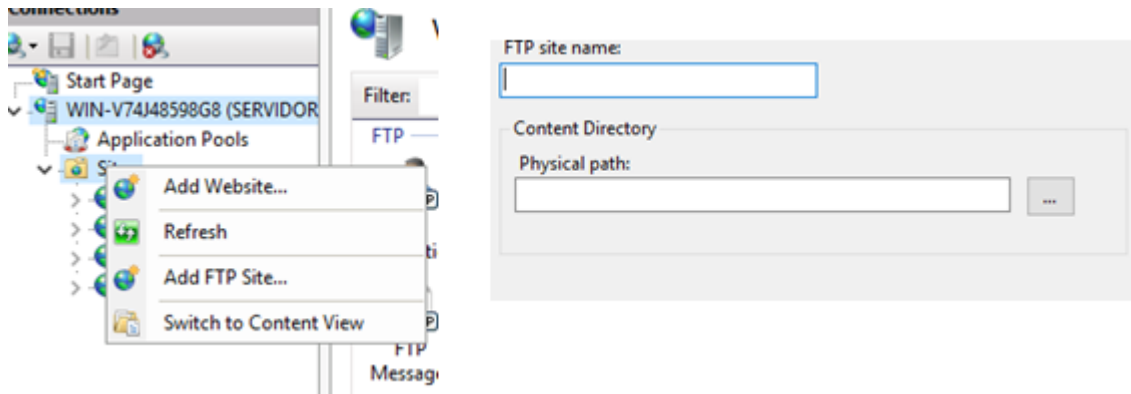


Figura 14 - Configuração do serviço FTP

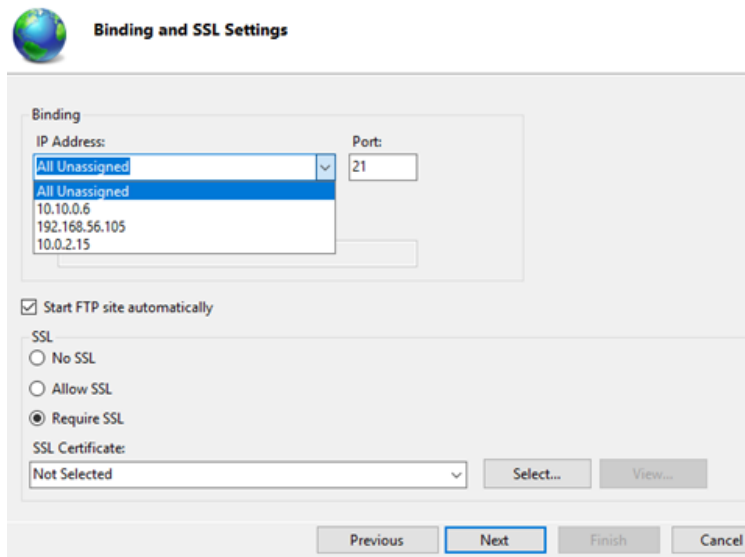


Figura 15 - Configurações adicionais do serviço FTP

Por último temos de escolher se queremos uma autenticação quando um utilizador tente aceder ao servidor FTP e também se pode especificar os usuários que tem permissões para ver os ficheiros.

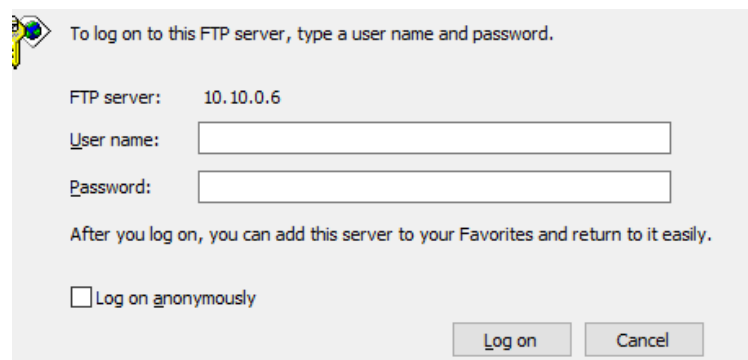


Figura 16 - Acesso ao servidor FTP

FTP root at 10.10.0.6

To view this FTP site in File Explorer: press Alt, click View, and then click **Open FTP Site in File Explorer**.

01/15/2021 07:23AM	Directory	admin
01/15/2021 07:41AM	Directory	users
01/15/2021 09:33AM	81,448	wireguard-installer (1).exe

Figura 17 - Pastas e permissões no servidor FTP

Para o SFTP foi instalado o serviço OpenSSH Server e foi testado a ligação com uma das VM do cenário.

```

root@dbadmin:~# sftp userFTP@10.10.0.6
The authenticity of host '10.10.0.6 (10.10.0.6)' can't be established.
ECDSA key fingerprint is SHA256:vdPNsmYyDJAZX7s8/ibBcTzHWj/m8dUAGGL3o1Ocpj
Are you sure you want to continue connecting (yes/no)? y
Please type 'yes' or 'no': yes
Warning: Permanently added '10.10.0.6' (ECDSA) to the list of known hosts
userFTP@10.10.0.6's password:
Connected to 10.10.0.6.
sftp>
sftp> ls -ltr

```

Figura 18 - Ligação à máquina 10.10.0.6


```
sftp> put /etc/ssh/sshd_config /C:/FTP
Uploading /etc/ssh/sshd_config to /C:/FTP/sshd_config
/etc/ssh/sshd_config 100% 2576 26.9KB/s 00:00
```

Figura 19 - Cópia do ficheiro sshd_config para o servidor FTP

Name	Date modified	Type
admin	1/15/2021 7:23 AM	File folder
users	1/15/2021 7:41 AM	File folder
sshd_config	1/16/2021 10:57 AM	File

Figura 20 - Ficheiro copiado no servidor de FTP

Um dos desafios na configuração do FTP foi a implementação de certificados SSL, sendo que o servidor Windows Server apresenta a opção de gerir certificados.

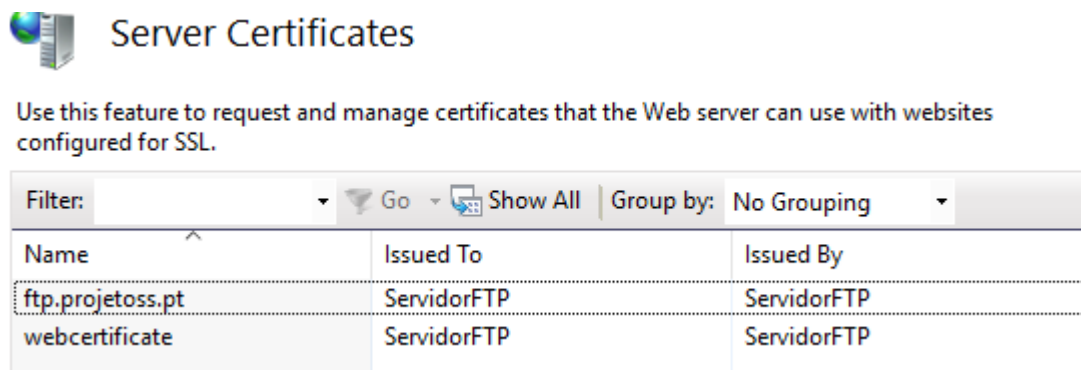


Figura 21 - Geração de certificados

Serviço Database

Para a implementação do serviço de base de dados existem muitas opções a ter em conta, como servidores PostgreSQL Server, Microsoft SQL Server, Oracle-XE -- Versão "lite" e MariaDB. Um dos mais utilizados é o MySQL que é o serviço que foi instalado na máquina Ubuntu 18.04. Durante a instalação do MySQL server existe a opção de o fazer de forma segura, com as credencias e um utilizador para aceder à base de dados.

Outra das recomendações foi que o servidor só seja possível aceder através de ligações ssh para a alteração e atualização de dados e para isso tem de ser alterado o ficheiro `/etc/mysql/mysql.conf.d/mysqld.cnf`.

```
# localhost which is more compatible and is
bind-address          = 127.0.0.1
local-infile = 0
#
# * Fine Tuning
```

Figura 22 - Ficheiro `/etc/mysql/mysql.conf.d/mysqld.cnf`

Caso se pertenda aceder ao servidor MySQL sem fazer uso de SSH é necessário alterar o endereço IP de destino.

Serviço HTTP

De forma a existir um website, foi utilizada uma máquina virtual com o sistema operativo linux Ubuntu Server 18.04. Neste servidor, foi configurado o Wireguard cliente de forma a ligar o servidor à VPN. Foi também configurado o bacula, o serviço SSH bem como Fail2Ban.

Numa prestação de segurança, foi configurada a versão segura do protocolo HTTP, o protocolo HTTPS. Assim, foi necessário criar um certificado, neste caso configurado através do certbot. De forma a ser possível gerar um certificado, foi necessário associar o endereço IP a um domínio específico.

Para associar o endereço IP público a um domínio foi utilizado o Freenom, onde foi criado o domínio `segurancasistemas.tk`.

De forma a utilizar o `certbot`, foi necessário executar o comando `sudo certbot --nginx -d segurancasistemas.tk` e escolher a opção de redirecionar todo o tráfego para HTTPS, de forma a tornar mais seguro. De seguida, e de forma a testar o certificado, foi necessário aceder ao website <https://www.ssllabs.com/ssltest/analyze.html?d=segurancasistemas.tk> (Figura 23)

De forma a verificar o correto funcionamento deste serviço, basta colocar no navegador o endereço `segurancasistemas.tk` e, podemos visualizar, como mostra na **Erro! A origem da referência não foi encontrada.**⁴, que o website contém um certificado válido.

SSL Report: segurancasistemas.tk (79.169.163.184)

Assessed on: Sat, 16 Jan 2021 21:22:57 UTC | [Hide](#) | [Clear cache](#)

[Scan Another »](#)

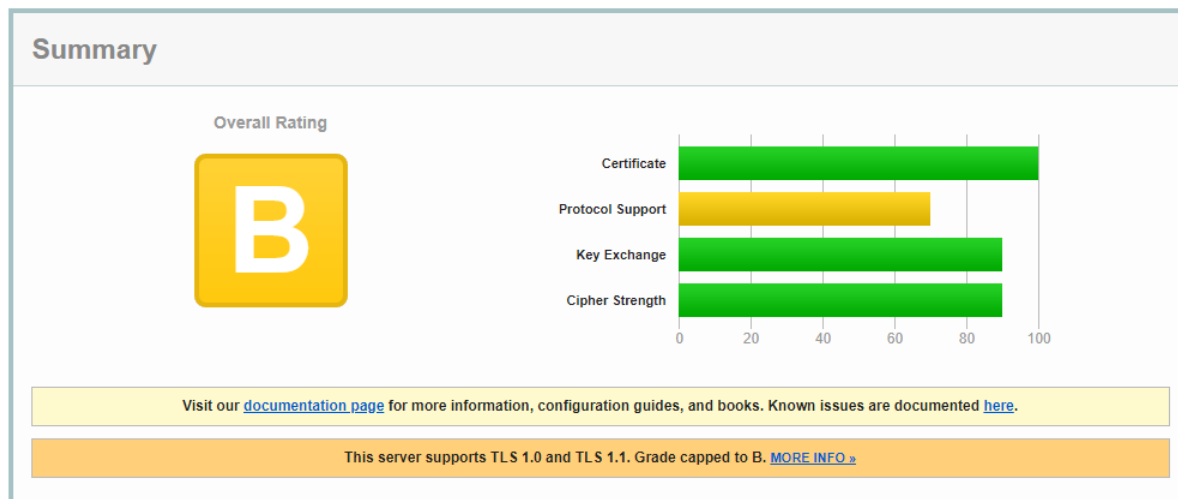


Figura 24 - Teste ao certificado SSL



Figura 23- Página Web com certificado

Serviço Mail

De forma a existir um servidor de email, foi utilizada uma máquina virtual com o sistema operativo linux Ubuntu Server 18.04. Neste servidor, foi configurado o Wireguard cliente de forma a ligar o servidor à VPN. Foi também configurado o bacula, o serviço SSH bem como Fail2Ban.

Em especial para o correto funcionamento do servidor de mail, foram instalados e configurados os serviços postfix e dovecot, bem como configuradas as versões TLS.

Assim foram criadas várias contas e na figura 25, podemos visualizar, através do cliente de email Mozilla Thunderbird o acesso ao às respetivas contas de email.

Configurar o seu endereço de e-mail existente

Utilize o seu endereço de e-mail atual

O seu nome: User 2

Endereço de e-mail: user2@projetoss.pt

Palavra-passe: •••••

☒ Memorizar palavra-passe

✓ Foram encontradas as seguintes definições ao testar o servidor

	A RECEBER	A ENVIAR
Protocolo:	POP3	SMTP
Servidor:	79.169.163.184	79.169.163.184
Porta:	110	587
SSL:	STARTTLS	STARTTLS
Autenticação:	Palavra-passe normal	Palavra-passe normal
Nome de utilizador:	user2	user2

[Configuração avançada](#)

Cancelar Testar novamente Feito

Figura 25- Configuração de e-mail

Proxy Squid

No nosso cenário, criamos uma máquina virtual com a versão servidor do Ubuntu 18.04 a correr uma Proxy Squid, a qual foi usada para permitir mais controlo e segurança nas pesquisas web, tendo sido implementado um modelo de autenticação no browser, através do proxy, para que antes de poder aceder ao mesmo e realizar qualquer pesquisa o utilizador entre com as suas credenciais. Além disso, foi negado o acesso a sites especificados nas configurações do Squid.

Também foi configurado o fail2ban para garantir que quando os utilizadores falham repetitivamente a autenticação via SSH tenham o acesso negado e por sua vez o seu IP de acesso banido. Esta configuração serve para garantir a segurança do servidor.

Por último foi configurado também o acesso da máquina virtual à VPN via Wireguard cliente.

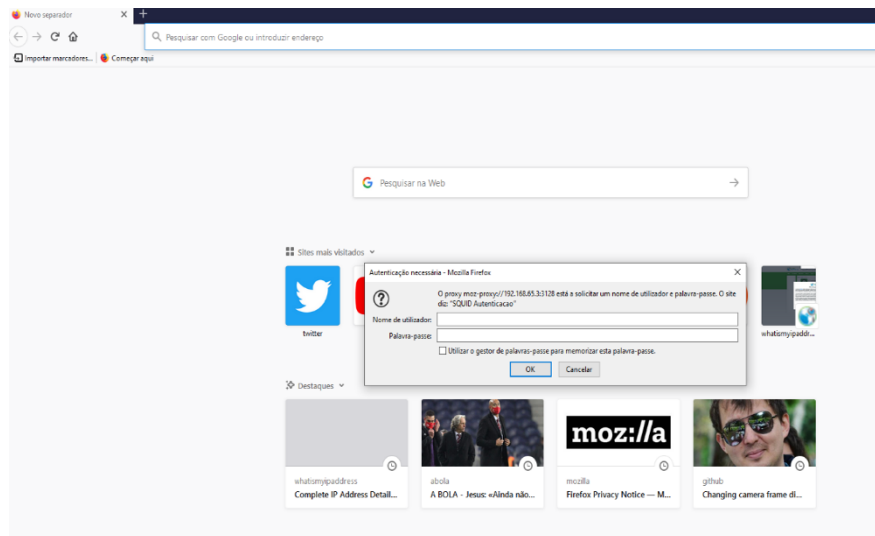


Figura 26 - Autenticação ao proxy Squid

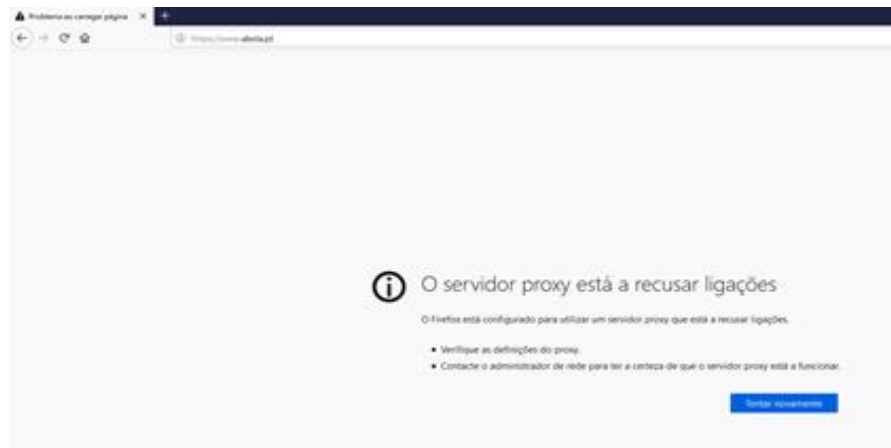


Figura 27 - Site bloqueado pela Squid

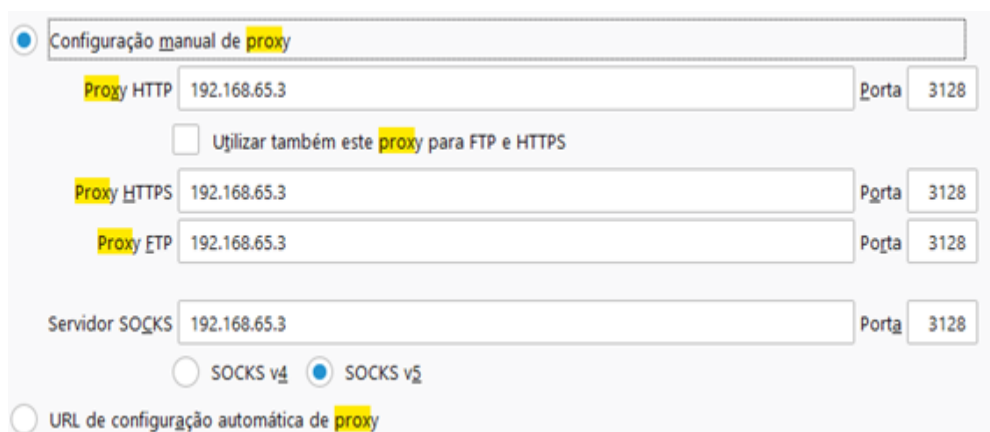


Figura 28 - Configuração do proxy no Firefox

5. Conclusão

Este trabalho permitiu-os pôr em prática o conteúdo lecionado nas aulas num cenário parecido com o mundo real. Conseguimos implementar uma grande parte dos serviços de segurança necessários para a realização de comunicações seguras, bem como proteger os serviços disponibilizados.

Gostaríamos de ter implementado uma rede de clientes fictícios para tornar o cenário ainda mais real, mas investimos bastante tempo na resolução de problemas que foram aparecendo.

6. Referências

- [1] CraigMckenna, “Waikato Linux Users Group,” 16 12 2005. [Online]. Available: <http://wiki.wlug.org.nz/SourceBasedRouting>.
- [2] Linode, “Linode,” 11 01 2021. [Online]. Available: <https://www.linode.com/docs/guides/using-fail2ban-to-secure-your-server-a-tutorial/>.
- [3] S. Hermoso, “APNIC,” 23 05 2019. [Online]. Available: <https://blog.apnic.net/2019/05/23/how-to-deploying-dnssec-with-bind-and-ubuntu-server/>.
- [4] 04 05 2015. [Online]. Available: <https://blog.inittab.org/administracion-sistemas/dnssec-asegurando-las-respuestas-de-nuestro-dominio-la-practica-i/>.
- [5] Martin, “WinSCP,” 11 01 2021. [Online]. Available: https://winscp.net/eng/docs/guide_windows_ftps_server.
- [6] S. Portillo, “Tutoriales IT,” 14 01 2020. [Online]. Available: <https://tutorialesit.com/windows-server-configurar-ftp-conexiones-ftp-modo-pasivo/>.
- [7] K. Sibbald, “Bacula,” 18 08 2013. [Online]. Available: https://www.bacula.org/5.2.x-manuals/en/main/main/Configuring_Director.html.
- [8] “Ubuntu,” 4 2019. [Online]. Available: <https://ubuntu.com/server/docs/backups-bacula>.
- [9] Eric, “Bacula,” 08 06 2011. [Online]. Available: https://www.bacula.org/5.0.x-manuals/en/main/main/Bacula_Security_Issues.html.
- [10] K. Sibbald, “Bacula,” 18 08 2013. [Online]. Available: https://www.bacula.org/5.2.x-manuals/en/main/main/Bacula_TLS_Communications.html.
- [11] “Bacula Enterprise Documentation,” [Online]. Available: https://www.baculasystems.com/dl/trial_experience/manual/html/main/Bacula_TLS_Communications_E.html.
- [12] D. Lukan, “Infosec,” 2 10 2014. [Online]. Available: <https://resources.infosecinstitute.com/topic/data-backups-bacula-backup-encryption/>.
- [13] Ethand, “Ionos,” 04 12 2014. [Online]. Available: <https://devops.ionos.com/tutorials/use-ssh-keys-with-putty-on-windows/>.
- [14] K. Azza, “restoreBin,” 21 11 2020. [Online]. Available: <https://restorebin.com/letsencrypt-nginx-certbot/>.
- [15] [Online]. Available: https://www.server-world.info/en/note?os=Ubuntu_18.04&p=mail&f=5.
- [16] B. S. L. Videos, “Youtube,” 10 10 2014. [Online]. Available: <https://www.youtube.com/watch?v=2nkAQ9M6ZF8>.
- [17] 20 01 2019. [Online]. Available: <https://linuxize.com/post/secure-nginx-with-let-s-encrypt-on-ubuntu-18-04/>.
- [18] J. Ellingwood, “Digital Ocean,” 23 07 2013. [Online]. Available: <https://www.digitalocean.com/community/tutorials/how-to-secure-mysql-and-mariadb-databases-in-a-linux-vps>.