

**Trabalho prático Cibersegurança**

**Membros:** Adilson Alves Neves, Gustavo Ribeiro, Fredson Gomes

**Professor:** Rui Alexandre Coelho Alves

Brangança, 2023

**Introdução**

Neste relatório abordaremos algumas configurações de marquinhas virtuais de forma semelhante como as que são feitas em grande parte da internet, por grandes empresas e instituições.  
 O objetivo deste projeto é para entendermos como funcionam as comunicações entre as máquinas para que estas tenham conexões seguras, evitando ataques maliciosos que ataquem a integridade, a confidencialidade e a disponibilidade da informação de sistemas.

# **Integração entre as máquinas**

Primeiramente, as VMs devem estar conectadas na mesma rede, para que exista a comunicação entre os clientes com os servidores, e também entre os servidores em si.

Todas as máquinas envolvidas precisam se comunicar com um Certification Authority (CA) em comum, para que possam validar os certificados de segurança (SSL).

Os servidores WEB e FTP, assim como os clientes (no caso deste relatório, o cliente é a VM Ubuntu) devem confiar em alguma Certification Authority em comum, para que os servidores WEB e FTP, com as configurações corretas, possam estabelecer conexões seguras (HTTPS, no caso do servidor WEB, e FTPS, no caso do servidor FTP).

Para isto, é necessário que todas as máquinas configurem a CA criada por nós para ser confiável. Isso é possível da seguinte forma (Debian / Ubuntu):

1. Transferir o certificado gerado pela CA para a máquina;
2. Copiar ou mover este certificado para a pasta de certificados confiáveis com o seguinte comando:

*sudo cp <caminho\_do\_certificado> /usr/local/share/ca-certificates/*

1. Atualizar a lista de certificados do sistema com o seguinte comando:

*sudo update-ca-certificates*

# **VM Certification Authority (CA)**

Para configurar a nossa *Certification Authority*, seguimos os seguintes passos:

1. Instalamos o *OpenSSL* com o seguinte comando:  
   *sudo apt install openssl*
2. Geramos a chave privada da CA com o seguinte comando:  
   *sudo openssl genpkey -algorithm RSA -out ca.key*
3. Geramos o certificado da CA com o seguinte comando:  
   *sudo openssl req -x509 -new -key ca.key -out ca.crt*
4. Instalamos o certificado da CA como foi descrita no *item* *2* deste relatório;
5. Certificamos de que a CA foi configurada corretamente com o seguinte comando:  
   *openssl x509 -in ca.crt -text -noout*

Caso tudo tenha corrido corretamente, este comando deve retornar informações sobre o certificado da CA configurada.

# **VM Ubuntu** **Workstation**

Para realizar a configuração dessa máquina primeiramente criei a mesma no virtual box, utilizando a versão do Ubunto do 22.04.2 64 bits. Após a instalação, realizei a confirmação do SSH, seguindo para configuração do sudo e depois a configuração do firewall. Está é uma maquina relativamente simples porque trabalha basicamente como um sysadmin woksstation.

# **VM web server**

Para a realização da configuraçao dessa maquina virtual, para trabalhar como Web Server, utilizamos o Apache, juntamente o banco de dados maria db, php e por final o wordpress. Além de realziar as devidas configurações de sengurança.

**Passos realizados:**

1. Criação da VM no Virtual Box
2. Configurações do sudo
3. Instalação e configuração do SSH
4. Configuração do Firewall
5. Instalação/configuração do Apache
6. Instalação php
7. Instalação/configuração do Mariadb
8. Instalação e configuração do wordpress
9. Deploy do site pelo wordpress

# **VM FTP Server**

Esta etapa descreve passo a passo a instalação e configuração do vsftpd (Very Secure FTP Daemon) no sistema operacional Debian.

Passo 1: Instalando o vsftpd:

O comando para tal é a seguinte: *sudo apt install vsftpd*

O APT irá baixar e instalar o vsftpd, juntamente com suas dependências.

Para verificar se o *vsftpd* foi instalado corretamente, usei o comando abaixo:

*sudo systemctl status vsftpd*

Passo 2: Criando um usuário FTP exclusivo

Criei uma conta de usuário FTP usando o comando *"adduser"* através do comando abaixo:

*sudo adduser <nome>*

Isso criará um usuário FTP chamado "nome" no sistema.

Desabilitei o acesso ao shell para esse usuário, pois ele será usado exclusivamente para fins de FTP.

Passo 3: Adicionando o usuário FTP à lista de usuários permitidos

Adicionei o usuário FTP recém-criado ao arquivo vsftpd.userlist para permitir o login:

*echo "ftpuser" | sudo tee -a /etc/vsftpd.userlist*

Isso garante que o usuário "ftpuser" tenha permissão para fazer login no servidor FTP.

Passo 4: Criando diretório para o usuário FTP

Criei um diretório FTP para o usuário FTP e defini as seguintes permissões:

*sudo mkdir -p /home/ftpuser/ftp\_dir/upload*

*sudo chmod 550 /home/ftpuser/ftp\_dir*

*sudo chmod -R 750 /home/ftpuser/ftp\_dir/upload*

*sudo chown -R ftpuser: /home/ftpuser/ftp\_dir*

Esses comandos criam a estrutura de diretórios para o usuário FTP e definem as permissões corretas para garantir a segurança e a privacidade dos arquivos.

Passo 5: Configurando o vsftpd

Editei o arquivo de configuração principal do vsftpd:

*sudo vim /etc/vsftpd.conf*

O arquivo de configuração do vsftpd controla o comportamento do servidor FTP.

Defini as seguintes configurações no arquivo:

anonymous\_enable=NO

local\_enable=YES

write\_enable=YES

chroot\_local\_user=YES

allow\_writeable\_chroot=YES

userlist\_enable=YES

userlist\_file=/etc/vsftpd.userlist

userlist\_deny=NO

Essas configurações desativam o acesso anônimo, permitem o login de usuários locais, permitem a gravação de arquivos, restringe os usuários locais a seus diretórios pessoais, possibilitam usar uma lista de usuários específica para controlar o acesso ao servidor FTP, especificam o caminho do arquivo que contém a lista de usuários permitidos ou negados, especificam uma lista de usuários permitidos.

# 

# **CONCLUSÃO**

Conseguimos completar nossos objetivos, que foi configurar máquinas para que haja uma comunicação segura entre o cliente e os servidores. Uma dificuldade comum entre os parcipantes do grupo foi na configuração do SSH para comunicação entre maquinas.

E acreditamos que esse projeto pode ser melhorado adicionando tecnologias mais recentes, como a containerização com Docker. Apenas para exemplificar algumas ideias superficialmente: poderíamos utilizar um container com Servidor WEB e FTP, para que fosse possível atualizar um website, enviando os novos arquivos utilizando o servidor FTP. Poderíamos também criar um container para o servidor de email e outro para a CA, evitando assim utilizar mais máquinas virtuais, otimizando a utilização de recursos da máquina host.

# 