**Configuração da infiniband após formatação e instalação do Proxmox VE:**

**Passo 1:**

Atualizar sistema e instalar bibliotecas necessárias com o comando:

**apt update**

**apt install infiniband-diags ibutils rdmacm-utils libmlx4-1 libmlx5-1 libibverbs1 ibverbs-utils**

**Passo 2:** Verificar se a interface da infiniband foi criada:

**ibv\_devinfo**

**ip a**

**Passo 3:** Criar a interface no proxmox pra configurar a infiniband:

entrando nas interfaces:

**nano /etc/network/interfaces**

Adicione a interface ibp5s0 (OBS.: Alterar o “x” para o ip desejado):

**auto ibp5s0**

**iface ibp5s0 inet static**

**address 192.168.1.x/24**

**mtu 65507**

**pre-up echo connected > /sys/class/net/ibp5s0/mode**

Para aplicar -> “CTRL + O” + “CTRL + X”

**Passo 4:** Atualizar alterações:

**ifdown ibp5s0 && ifup ibp5s0**

**Passo 5:** Testar a comunicação:

**ping 150.168.1.x**

**Passo 6:** Adicionar o nó configurado no HOST (OBS.: Alterar o “x”):

Com isso os nós estarão conectados

**192.168.1.x coiotex.cer.ufpe.br coiotex**

**Instalação do hardenning:**

sudo apt update && sudo apt upgrade -y

**Passo 1 :**

nano /root/[harden-ubuntu.sh](http://harden-ubuntu.sh)

**Passo 2 - Colar o script:**

#!/bin/bash

set -e

cat << 'EOF'

██████╗███████╗██████╗ ██╗ ██╗███████╗██████╗ ███████╗

██╔════╝██╔════╝██╔══██╗ ██║ ██║██╔════╝██╔══██╗██╔════╝

██║ █████╗ ██████╔╝█████╗██║ ██║█████╗ ██████╔╝█████╗

██║ ██╔══╝ ██╔══██╗╚════╝██║ ██║██╔══╝ ██╔═══╝ ██╔══╝

╚██████╗███████╗██║ ██║ ╚██████╔╝██║ ██║ ███████╗

╚═════╝╚══════╝╚═╝ ╚═╝ ╚═════╝ ╚═╝ ╚═╝ ╚══════╝

Hardening & FreeIPA Ready Template for Rocky/Ubuntu

EOF

#───────────────────────────────────────────────#

# Verificações básicas

#───────────────────────────────────────────────#

if [ "$EUID" -ne 0 ]; then

echo "❌ Execute como root ou via sudo."

exit 1

fi

if [ -z "${SUDO\_USER:-}" ]; then

echo "❌ Execute com: sudo $0 [opções]"

exit 1

fi

#───────────────────────────────────────────────#

# Parâmetros

#───────────────────────────────────────────────#

JOIN\_DOMAIN=0

DOMAIN="cer.ufpe.br"

SERVER="iam.cer.ufpe.br"

REALM="CER.UFPE.BR"

usage() {

echo "Uso: sudo $0 [--join-domain]"

exit 1

}

while [[ $# -gt 0 ]]; do

case $1 in

--join-domain)

JOIN\_DOMAIN=1

shift

;;

-h|--help)

usage

;;

\*)

echo "Opção desconhecida: $1" >&2

usage

;;

esac

done

#───────────────────────────────────────────────#

# Etapas do hardening

#───────────────────────────────────────────────#

echo "[1/10] Atualizando o sistema..."

dnf update -y || apt update -y && apt upgrade -y

echo "[2/10] Instalando pacotes essenciais..."

if command -v dnf &>/dev/null; then

dnf install -y openssh-server firewalld epel-release fail2ban dnf-automatic ipa-client

else

apt install -y openssh-server firewalld fail2ban unattended-upgrades freeipa-client

fi

echo "[3/10] Configurando SSH seguro..."

sed -i 's/^#\*PermitRootLogin.\*/PermitRootLogin no/' /etc/ssh/sshd\_config

sed -i 's/^#\*PasswordAuthentication.\*/PasswordAuthentication yes/' /etc/ssh/sshd\_config

sed -i 's/^#\*X11Forwarding.\*/X11Forwarding no/' /etc/ssh/sshd\_config

sed -i 's/^#\*AllowTcpForwarding.\*/AllowTcpForwarding no/' /etc/ssh/sshd\_config

sed -i 's/^#\*MaxAuthTries.\*/MaxAuthTries 3/' /etc/ssh/sshd\_config

systemctl restart sshd

echo "[4/10] Configurando firewall..."

systemctl enable --now firewalld

firewall-cmd --set-default-zone=drop

firewall-cmd --permanent --add-service=ssh

firewall-cmd --permanent --add-port=80/tcp

firewall-cmd --permanent --add-port=443/tcp

firewall-cmd --permanent --add-rich-rule='rule family="ipv4" service name="ssh" limit value="3/m" accept'

firewall-cmd --reload

echo "[5/10] Configurando Fail2Ban..."

cat <<EOL >/etc/fail2ban/jail.local

[sshd]

enabled = true

port = ssh

maxretry = 3

bantime = 30m

findtime = 10m

EOL

systemctl enable --now fail2ban

echo "[6/10] Protegendo permissões sensíveis..."

chmod 644 /etc/passwd

chmod 600 /etc/shadow

echo "[7/10] Ativando atualizações automáticas..."

if command -v dnf-automatic &>/dev/null; then

systemctl enable --now dnf-automatic.timer

sed -i 's/^upgrade\_type = .\*/upgrade\_type = security/' /etc/dnf/automatic.conf

else

dpkg-reconfigure -plow unattended-upgrades

fi

echo "[8/10] Limpando pacotes desnecessários..."

if command -v dnf &>/dev/null; then

dnf autoremove -y && dnf clean all

else

apt autoremove -y && apt clean

fi

echo "[9/10] Limpando dados sensíveis e históricos..."

history -c || true

rm -rf /tmp/\* /var/tmp/\* /root/.bash\_history /home/\*/.bash\_history

find /root -name ".\*history" -exec rm -f {} \;

#───────────────────────────────────────────────#

# FreeIPA Join opcional

#───────────────────────────────────────────────#

if [[ "$JOIN\_DOMAIN" -eq 1 ]]; then

echo "[10/10] Iniciando integração com FreeIPA..."

echo "➡️ Dominio: $DOMAIN"

echo "➡️ Servidor: $SERVER"

echo "➡️ Realm: $REALM"

ipa-client-install --domain="$DOMAIN" --server="$SERVER" --realm="$REALM" --mkhomedir --force-join -U

echo "✅ FreeIPA join concluído com sucesso."

else

echo "[10/10] Pular integração FreeIPA (use --join-domain para habilitar)"

fi

echo "✅ Hardening finalizado. Pronto para uso como TEMPLATE."

**Passo 3 - Execute o script dentro do CT Ubuntu:**

chmod +x harden-ubuntu.sh

sudo ./harden-ubuntu.sh --join-domain

**Passo 4 - Apertar “Enter” na primeira tela:**

**Passo 5 - Segunda e terceira tela, Digitar:**

[**iam.cer.ufpe.br**](http://iam.cer.ufpe.br)

**e “Enter”:**

**Passo 6 - Verificar se está tudo certo:**

systemctl status fail2ban

systemctl status firewalld

grep PermitRootLogin /etc/ssh/sshd\_config

ipa host-show $(hostname) # se tiver integrado ao FreeIPA

**Passo 7 - Configurar o acesso ao freeipa-client com acesso ADM:**

sudo ipa-client-install \

--domain=cer.ufpe.br \

--server=iam.cer.ufpe.br \

--realm=CER.UFPE.BR \

--principal=admin \

--password='SENHA\_DO\_ADMIN' \

--mkhomedir --force-join -U