#!/bin/bash

# The following requirements will be met by the system after running this script:

# Hostname: autosrv

# Network setup: static IP address, gateway, DNS server, and search domains

# Software installed: ssh server, apache2 web server, squid web proxy

# Firewall configuration: allow SSH, HTTP, HTTPS, and web proxy

# User accounts created with home directories, authorized keys, and bash shell

# Make sure the hostname has autosrv chosen.

if [ "$(hostname)" != "autosrv" ]; then

echo "Setting hostname to autosrv..."

sudo hostnamectl set-hostname autosrv

fi

# Make sure the network settings are correct.

interface="ens34"

# Make sure the interface already exists before creating it.

if ! ip link show $interface >/dev/null 2>&1; then

echo "Creating interface $interface..."

sudo nmcli con add con-name $interface ifname $interface type ethernet ipv4.method manual ipv4.addresses 192.168.16.21/24 ipv4.gateway 192.168.16.1 ipv4.dns 192.168.16.1

fi

# Configure the interface.

echo "Setting interface $interface up..."

sudo ip link set $interface up

# DNS search domains should be set to "home.arpa" and "localdomain".

echo "Setting DNS search domains to home.arpa and localdomain..."

echo 'search home.arpa localdomain' | sudo tee /etc/resolv.conf >/dev/null

echo "Network configuration is set."

# Check that the programme was installed correctly.

packages=("openssh-server" "apache2" "squid")

for package in "${packages[@]}"; do

if ! dpkg -s $package >/dev/null 2>&1; then

echo "Installing $package..."

sudo apt update

sudo apt install -y $package

echo "Installed $package."

else

echo "$package is already installed."

fi

done

# Verify that the SSH server is set up to only accept authentication using SSH keys and not passwords.

if ! grep -q '^PasswordAuthentication no' /etc/ssh/sshd\_config; then

echo "Configuring SSH server to allow SSH key authentication and not allow password authentication..."

sudo sed -i 's/#PasswordAuthentication yes/PasswordAuthentication no/' /etc/ssh/sshd\_config

sudo systemctl restart sshd

fi

# Inspect the Apache web server's ports 443 and 80 to make sure they are open and taking incoming HTTP traffic.

if ! ss -tlnp | grep -q -E ':(80|443)\s'; then

echo "Configuring Apache web server to listen on port 80 and 443..."

sudo sed -i 's/Listen 80/Listen 0.0.0.0:80/' /etc/apache2/ports.conf

sudo sed -i 's/Listen 443/Listen 0.0.0.0:443/' /etc/apache2/ports.conf

sudo systemctl restart apache2

fi

# Verify that port 3128 on the Squid web proxy is open for traffic.

if ! ss -tlnp | grep -q ':3128\s'; then

echo "Configuring Squid web proxy to listen on port 3128..."

sudo sed -i 's/http\_port 3128/http\_port 0.0.0.0:3128/' /etc/squid/squid.conf

sudo systemctl restart squid

fi

# Verify that the firewall is set up properly.

echo "Configuring firewall with UFW..."

allowed\_services=("OpenSSH" "Apache" "HTTPS" "Squid")

sudo ufw default deny incoming

sudo ufw default allow outgoing

for service in "${allowed\_services[@]}"; do

if ! sudo ufw status verbose | grep -q "$service"; then

echo "Allowing $service..."

sudo ufw allow "$service"

else

echo "$service is already allowed."

fi

done

sudo ufw --force enable

echo "Firewall configured successfully."

# Verify that the user accounts were created properly..

echo "Creating user accounts..."

# Functionality for adding authorized\_keys files, creating users, and creating SSH keys.

create\_user() {

username=$1

sudo useradd -m -s /bin/bash $username

echo "Generating SSH keys for $username..."

sudo su - $username -c "ssh-keygen -t rsa -f /home/$username/.ssh/id\_rsa -N ''"

sudo su - $username -c "ssh-keygen -t ed25519 -f /home/$username/.ssh/id\_ed25519 -N ''"

echo "Adding SSH keys to authorized\_keys file for $username..."

sudo cat /home/$username/.ssh/id\_rsa.pub | sudo tee -a /home/$username/.ssh/authorized\_keys >/dev/null

sudo cat /home/$username/.ssh/id\_ed25519.pub | sudo tee -a /home/$username/.ssh/authorized\_keys >/dev/null

}

# Make a user named dennis with sudo privileges and an extra SSH key.

if ! id "dennis" >/dev/null 2>&1; then

create\_user "dennis"

sudo usermod -aG sudo dennis

echo "ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIG4rT3vTt99Ox5kndS4HmgTrKBT8SKzhK4rhGkEVGlCI student@generic-vm" | sudo tee -a /home/dennis/.ssh/authorized\_keys >/dev/null

fi

# Add extra users to your accounts.

usernames=("aubrey" "captain" "nibbles" "brownie" "scooter" "sandy" "perrier" "cindy" "tiger" "yoda")

for username in "${usernames[@]}"; do

if ! id "$username" >/dev/null 2>&1; then

create\_user "$username"

else

echo "User $username already exists."

fi

done

echo "Successfully created user accounts."

# Verify the system setup

echo "Verifying system setup..."

# Verify hostname

if [ "$(hostname)" = "autosrv" ]; then

echo "Hostname is set to autosrv."

else

echo "Hostname is not set to autosrv."

fi

# Verify network configuration

if ip addr show dev $interface | grep -q "192.168.16.21/24" && ip route show dev $interface | grep -q "default via 192.168.16.1"; then

echo "Network configuration is correct."

else

echo "Network configuration is incorrect."

fi

# Verify software installation

packages=("openssh-server" "apache2" "squid")

missing\_packages=()

for package in "${packages[@]}"; do

if ! dpkg -s $package >/dev/null 2>&1; then

missing\_packages+=("$package")

fi

done

if [ ${#missing\_packages[@]} -eq 0 ]; then

echo "All required packages are installed."

else

echo "Missing packages: ${missing\_packages[\*]}"

fi

# Verify SSH server configuration

if grep -q '^PasswordAuthentication no' /etc/ssh/sshd\_config; then

echo "SSH server is configured to allow SSH key authentication only."

else

echo "SSH server configuration is incorrect."

fi

# Verify Apache web server configuration

if ss -tlnp | grep -q -E ':(80|443)\s'; then

echo "Apache web server is configured to listen on ports 80 and 443."

else

echo "Apache web server configuration is incorrect."

fi

# Verify Squid web proxy configuration

if ss -tlnp | grep -q ':3128\s'; then

echo "Squid web proxy is configured to listen on port 3128."

else

echo "Squid web proxy configuration is incorrect."

fi

# Verify firewall configuration

allowed\_services=("OpenSSH" "Apache" "HTTPS" "Squid")

missing\_services=()

for service in "${allowed\_services[@]}"; do

if ! sudo ufw status verbose | grep -q "$service"; then

missing\_services+=("$service")

fi

done

if [ ${#missing\_services[@]} -eq 0 ]; then

echo "Firewall is configured to allow all required services."

else

echo "Missing services in firewall configuration: ${missing\_services[\*]}"

fi

# Verify user accounts and SSH keys

usernames=("dennis" "aubrey" "captain" "nibbles" "brownie" "scooter" "sandy" "perrier" "cindy" "tiger" "yoda")

missing\_users=()

for username in "${usernames[@]}"; do

if ! id "$username" >/dev/null 2>&1; then

missing\_users+=("$username")

else

if [ ! -f "/home/$username/.ssh/authorized\_keys" ]; then

missing\_users+=("$username (missing authorized\_keys file)")

fi

fi

done

if [ ${#missing\_users[@]} -eq 0 ]; then

echo "All user accounts and SSH keys are properly configured."

else

echo "Missing or misconfigured user accounts or SSH keys: ${missing\_users[\*]}"

fi

echo "System setup verification complete."