Bisevac Nina

nina.bisevac@noseryoung.com

VPN

VPN-Server und VPN-Client erstellen

Inhaltsverzeichnis

[1 Einleitung 2](#_Toc168479062)

[2 Namensschema 2](#_Toc168479063)

[3 Netwerkplan 2](#_Toc168479064)

[4 IP-Konzept 2](#_Toc168479065)

[5 Ports 2](#_Toc168479066)

[6 Gerätebeschreibung 2](#_Toc168479067)

[7 Installation / Konfiguration 3](#_Toc168479068)

[8 Testen 3](#_Toc168479069)

[8.1 Testkonzept 3](#_Toc168479070)

[8.2 Testprotokoll 4](#_Toc168479071)

Abildungsverzeichnis

[Abbildung 1Netzwerkplan 2](#_Toc168671041)

[Abbildung 2 ProxMox 3](#_Toc168671042)

[Abbildung 3 Konfiguration Server 4](#_Toc168671043)

[Abbildung 4 Konfiguration Client 5](#_Toc168671044)

Tabellenverzeichnis

[Tabelle 1 Namensschema 2](#_Toc168671046)

[Tabelle 2 IP-Konzept 2](#_Toc168671047)

[Tabelle 3 Ports 3](#_Toc168671048)

[Tabelle 4 Gerätebeschreibung 3](#_Toc168671049)

[Tabelle 5 Testfall 1 5](#_Toc168671050)

[Tabelle 6 Testfall 2 5](#_Toc168671051)

[Tabelle 7 Testprotokoll 6](#_Toc168671052)

# Einleitung

Dieser Auftrag umfasst die Einrichtung eines VPN-Servers und eines VPN-Clients. Auf dem Server wird ein Wireguard-Server eingerichtet. Lokal wird dann eine neue Linux-VM als VPN-Client erstellt.

Die Aufgabe besteht darin, eine Verbindung zwischen dem Client und dem Server herzustellen und zu überprüfen, ob ein Handshake stattgefunden hat und der neue Netzwerkadapter aktiv ist.

# Namensschema

Tabelle 1 Namensschema

|  |  |  |
| --- | --- | --- |
| **Gerät** | **Name** | **Lägende** |
| Linux Server | wghostXX | XX: zugewiesene Nummer |
| Linux Client | linux-client-XX | XX: Aufsteigende Zahl |

# Netzwerkplan

Ein Bild, das Screenshot, Reihe, Diagramm, Text enthält.

Automatisch generierte Beschreibung

Abbildung 1Netzwerkplan

# IP-Konzept

Tabelle 2 IP-Konzept

|  |  |  |
| --- | --- | --- |
| **Gerätetyp** | **Art** |  |
| Linux Server | Statische | 192.168.5.118 |
| Linux Client | DHCP | 172.31.80.8 |

# Ports

Tabelle 3 Ports

|  |  |
| --- | --- |
| **SSH-Port** | **VPN-Port** |
| 830 | 51828 |

# Gerätebeschreibung

Tabelle 4 Gerätebeschreibung

|  |  |
| --- | --- |
| **Gerätetyp** | **Service** |
| Linux Server | Wireguard |
| Linux Client | Ubuntu |

# 

# Installation / Konfiguration

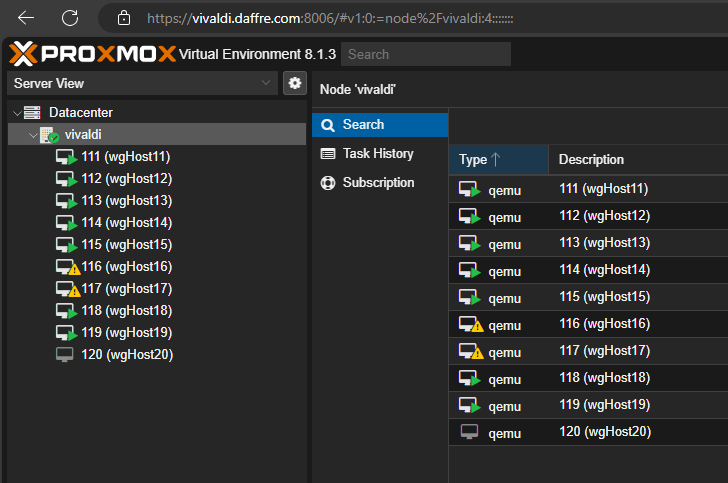


Abbildung 2 ProxMox

In Proxmox mit dem URL [https://vivaldi.daffre.com:8006](https://vivaldi.daffre.com:8006/) anmelden und den zugewiesenen Server starten. Danach eine SSH-Verbindung herstellen. Dazu folgenden den Befehl eingeben:

*ssh-p 830 sysadmin@ vivaldi.daffre.com*

Mit der IP des Servers wird es nicht funktionieren, da es eine private IP ist und man somit den zugewiesenen Port und die URL vom Proxmox angeben muss.

Auf diesem Server dann Wireguard installieren.

Lokal wird noch eine Linux VM als Client erstellt.

## Konfiguration Server

WireGuard mit folgendem Befehl installieren:

sudo apt install wireguard

Danach einen private- und publickey erstellen:

wg genkey | tee privatekey | wg pubkey > publickey

Mit dem Befehl cat kann man den private- oder publickey anzeigen lassen:

cat privatekey/publickey

Wireguard Konfigurations-Datei erstellen:

sudo nano /etc/wireguard/wg0.conf

Diese Datei so konfigurieren:

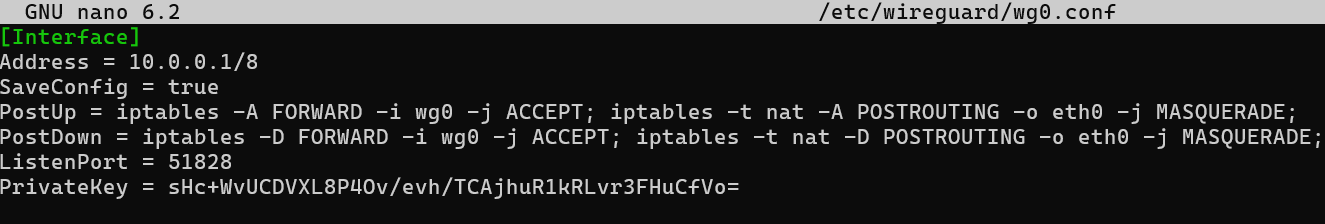


Abbildung 3 Konfiguration Server

Mit diesem befehl kann man das Konfigurations-Datei aktivieren:

wg-quick up wg0

und mit diesem hier deaktivieren:

wg-quick down wg0

Bei Änderungen zuerst wg-quick down wg0 anwenden, sonst werden diese nicht übernommen.

Zum Schluss muss man noch diesen befehl angeben damit setzt man die erlaubten IPs:

sudo wg set wg0 peer <publickey of client> allowed-ips 10.0.0.2/32

sudo wg kann benutz werden, um zu schauen das alles richtig konfiguriert ist.

## Konfiguration Client

Wie beim Sever alles bis zum Konfigurations-Datei befolgen. Danach wie angezeigt konfigurieren:

Ein Bild, das Text, Screenshot, Schrift, Software enthält.

Automatisch generierte Beschreibung

Abbildung 4 Konfiguration Client

Wg-quick up wg0

Bei Änderungen gilt das gleiche wie bei dem Server.

Die erlaubten IPs sind hier bereits in der Datei angegeben, im Gegensatz zum Server.

# Testen

## Testkonzept

Tabelle 5 Testfall 1

|  |  |
| --- | --- |
| **ID** | **T01** |
| **Testfall** | VPN Tunnel Test zwischen Client und Server |
| **Host** | wghost16 / linux-client-01 |
| **Beschreibung** | Es wird überprüft, ob ein VPN-Tunnel zwischen Client und Server aufgebaut wurde. Damit diese miteinander kommunizieren können. |
| **Testmethoden/ Testschritte** | Der Client muss die IP des Servers anpingen. |
| **Erwartetes Ergebnis** | Das Anpingen des Servers sollte funktionierten und eine Antwort soll zurückkommen. |

Tabelle 6 Testfall 2

|  |  |
| --- | --- |
| **ID** | **T02** |
| **Testfall** | Verwendung eines VPN-Tunnels für die gesamte Internetverbindung |
| **Host** | wghost16 / linux-client-01 |
| **Beschreibung** | Es wird überprüft, ob die gesamte Internetverbindung durch den VPN-Tunnel zwischen Client und Server gehen kann. |
| **Testmethoden/ Testschritte** | Der Client muss z.B. Google (8.8.8.8) anpingen. |
| **Erwartetes Ergebnis** | Das Anpingen von Google (8.8.8.8) sollte funktionierten und eine Antwort soll zurückkommen. |

## Testprotokoll

Tabelle 7 Testprotokoll

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Person** | **Datum** | **Test Name** | **Ergebnis** |
| **T01** | NB | 06.06.2024 | Verbindung von VPN-Tunnel zwischen Client und Server getestet | Erfolgreich |
| **T02** | NB | 06.06.2024 | Verbindung von Ganzen Internetverbindung über VPN-Tunnel getestet | Nicht erfolgreich |