

Projekt AE – SS 22

Entwicklung eines Konfigurationsgenerators für WireGuard VPN

2. Projektpräsentation

Jonas Berger,
Fachhochschule Südwestfalen



Agenda

- Wiederholung der Aufgabenstellung
- Umsetzung
- Bemerkenswerte Funktionen
- Funktionsweise

Wiederholung: Zielsetzung

- Vereinfachte Übersicht und Verwaltung der Konfigurationsdateien
 - Erweiterung
 - Änderung aller Konfigurationen (z.B. bei Anpassung Netzwerkdefinition)
- Verhindern von Syntaxfehlern
- Verwaltung sämtlicher Parameter
- Verwaltung von Client IP-Adressen
- Generierung von QR-Codes

Wiederholung: Konfiguration von WireGuard

[Interface]

PrivateKey = aCNV\

Address = 10.10.11.3/32

[Peer]

PublicKey = wS

AllowedIPs = 10.10.11.1/32

Endpoint = !49.137:1337

INI-Syntax

[Interface]

Name = node1.example.tld

Address = 192.0.2.3/32

ListenPort = 51820

PrivateKey = localPrivateKeyAbcAbcAbc=

DNS = 1.1.1.1,8.8.8.8

Table = 12345

MTU = 1500

PreUp = /bin/example arg1 arg2 %i

PostUp = /bin/example arg1 arg2 %i

PreDown = /bin/example arg1 arg2 %i

PostDown = /bin/example arg1 arg2 %i

[Peer]

Name = node2-node.example.tld

AllowedIPs = 192.0.2.1/24

Endpoint = node1.example.tld:51820

PublicKey = remotePublicKeyAbcAbcAbc=

PersistentKeepalive = 25

Konfigurationsparameterreferenz

Umsetzung

- Bedienung per Kommandozeile mit interaktiver Menüführung
- Plattformunabhängigkeit durch Containervirtualisierung
- Umfangreiche Ausgaben für Fehleranalyse
- Erweiterung für vermaschtes VPN möglich

Bemerkenswertes

- Konfigurationsdateien unterstützen INI-Syntax nicht vollständig
- Dateisystemzugriffe unterscheiden sich unter Windows und Unix
- Datenstruktur Klasse ist der beste Kompromiss

- Vorführung der Programmfunktionen anhand des Hauptmenüs
- Übersicht über Konfigurationsvariablen
- Erstellung einer neuen Konfiguration auf einem vorbereiteten Server bei AWS
- Herstellen einer Verbindung von unterschiedlichen Endgeräten, u.A. per QR-Code