trutz.io

SPICKZETTEL · WEGWERF-SERVER

Der Wegwerf-Server

Firewall-Minimalregeln, WireGuard-Vorlage, encryption-Block, Wegwerf-Rezept und die Nachweis-Kommandos. Begleitmaterial zum Video: ein Hetzner-Server aus OpenTofu, dessen Admin-Zugang in keinem Port-Scan auftaucht.

DIE EINE REGEL, AUS DER ALLES FOLGT

Der sicherste Admin-Zugang ist der, den ein Port-Scanner nicht einmal finden kann.

01 Die komplette Firewall: zwei Regeln

Video 02:32

```
resource "hcloud_firewall" "lab" {
  rule {  # ping: die Maschine lebt
    direction  = "in"
    protocol   = "icmp"
    source_ips = ["0.0.0.0/0", "::/0"]
  }
  rule {  # die einzige Tür
    direction  = "in"
    protocol   = "udp"
    port       = "51820"
    source_ips = ["0.0.0.0/0", "::/0"]
  }
}
```

Kein 22, kein 80, kein 443, kein 6443. SSH läuft nur durch den Tunnel. HTTP und HTTPS erst öffnen, wenn dort wirklich etwas lauscht: **nur öffnen, was läuft.**

02 WireGuard: ein Interface, ein Peer

Video 05:31

```
# Server: /etc/wireguard/wg0.conf
[Interface]
Address    = 10.101.0.1/24
ListenPort = 51820
PrivateKey = <server-private-key>
[Peer]  # dein Rechner, der einzige Peer
PublicKey  = <client-public-key>
AllowedIPs = 10.101.0.2/32
# Client: das Spiegelbild
[Interface]
Address    = 10.101.0.2/24
PrivateKey = <client-private-key>
[Peer]
PublicKey  = <server-public-key>
Endpoint   = <server-ip>:51820
AllowedIPs = 10.101.0.1/32
```

`AllowedIPs` bewusst eng: Nur der Verkehr zum Server geht durch den Tunnel. SSH spricht nur die Tunnel-Adresse an.

03 State und Plan verschlüsseln, ab Tag 1

Video 07:28

```
terraform {
  encryption {
    key_provider "pbkdf2" "passphrase" {
      passphrase = var.state_passphrase
    }
    method "aes_gcm" "default" {
      keys = key_provider.pbkdf2.passphrase
    }
    state {
      method   = method.aes_gcm.default
      enforced = true
    }
    plan {
      method   = method.aes_gcm.default
      enforced = true
    }
  }
}
```

Gespeicherte **Plan-Dateien** tragen sensitive Variablenwerte im Klartext, deshalb beides verschlüsseln. `enforced` ist die Ratsche: nie wieder ein unverschlüsselter State. Kann OpenTofu seit 1.7, Terraform nicht.

04 Das Wegwerf-Rezept

Video 10:50

- [ ] Die öffentliche IP als **eigene Ressource** anlegen: `hcloud_primary_ip` mit `auto_delete = false`. Sie überlebt den Server, der WireGuard-Endpoint bleibt stabil.
- [ ] Abbauen mit `tofu destroy -target=hcloud_server.lab`: bewusst nur der Server, nie die IP. Davor warnt auch die Provider-Doku wörtlich.
- [ ] Wiederaufbau mit einem einzigen `tofu apply`. Die WireGuard-Keys bleiben dieselben, der Tunnel verbindet sich von selbst neu.
- [ ] Die SSH-Warnung danach ist kein Fehler, sondern Trust On First Use: `ssh-keygen -R 10.101.0.1`, dann den neuen Fingerprint einmal bewusst bestätigen.

05 Der Nachweis von außen

Video 02:32

```
ping -c 2 <server-ip>
# antwortet: die Maschine lebt
nc -z -v -w3 <server-ip> 22
# Timeout, nicht refused
sudo nmap -sU -p 51820,33333 -Pn <server-ip>
# beide open|filtered: ununterscheidbar
```

Timeout heißt: Die Firewall verwirft still, bevor der Server das Paket sieht. Ein **connection refused** auf Port 22 wäre das Alarmsignal. Und der WireGuard-Port? Antwortet ohne gültigen Schlüssel gar nicht.

Von außen existiert dieser Server nicht. Für dich, mit dem richtigen Schlüssel, vollständig.

Dieser Spickzettel gehört zum Video über den Server, den ein Port-Scanner nicht findet. Jede Woche praxisnahe Antworten auf Docker- und Kubernetes-Probleme: der Newsletter.

trutz.io

Newsletter