BBSakura Networks

WireGuardで始める VPNパケットキャプチャ入門

naoki matsushita

JANOG58 LT

BBSakura Networks

自己紹介とやったこと

- 松下 尚樹
- ネットワークエンジニア
- 普段はクラウド事業者との接続回りを担当
- VPN初心者
- 自分のLab環境へのアクセスを **WireGuard** で構築
- 実際に動作しているのか**パケットレベルで確認**

BBSakura Networks

WireGuardとは

- UDPベースのVPN
- 設定がシンプル・高速

```
# 設定ファイルを作成し、以下のコマンドで起動するだけ
sudo wg-quick up wg0
```

wireguard VPNの仕組み① カプセル化

元のパケット（内側）：

```
[ IP ヘッダ ] [ ICMP / TCP ] [ データ ]
```

WireGuardで処理（外側）：

```
[ IPヘッダ ] [ UDPヘッダ(51820) ] [ WireGuard ヘッダ ] [ 暗号化ペイロード ] [ MAC ]
```

→ 外からはUDPパケットとして見える

BBSakura Networks

wireguard VPNの仕組み② 暗号化

- 暗号化アルゴリズム: **ChaCha20-Poly1305**
  → 軽量・高速。復号には正しい鍵が必要
- 鍵管理: **公開鍵・秘密鍵のペア**（Curve25519）
  → 設定ファイルに相手の公開鍵を記載するだけ
- ハンドシェイク: **Noise Protocol**
  → 接続前に鍵を合意（pingの前から流れるUDP通信）

```
# wg0.conf(抜粋)
[Peer]
PublicKey = abc123xyz...     ← 相手の公開鍵
AllowedIPs = 10.8.0.0/24
```

Lab構成

```
Internet
      ↓
Yamaha RTX1300  ← UDP 51820 を Port Forward
      ↓  10.128.1.0/24
Ubuntu
  ens18  ← インターネット側
  wg0    ← WireGuardトンネル側
      ↓
Lab
```

- Ubuntu に **WireGuard** を構築
- RTX1300 で UDP 51820 を NAT → Ubuntu へ転送

実験内容

MacからWireGuard接続。Ubuntuで**2か所同時**にキャプチャ。

```
# 外側(インターネット境界)
sudo tcpdump -ni ens18 udp port 51820
```

```
# 内側(WireGuardトンネル内部)
sudo tcpdump -ni wg0 icmp
```

```
Mac → Internet → ens18 → [WireGuard] → wg0 → Lab
                   ↑                     ↑
                外側を見る             内側を見る
```

実験結果① 外側（ens18）

```
12:34:56.123456 IP 203.0.113.1.54321 > 10.128.1.10.51820: UDP, length 148
12:34:56.234567 IP 203.0.113.1.54321 > 10.128.1.10.51820: UDP, length 92
12:34:57.345678 IP 203.0.113.1.54321 > 10.128.1.10.51820: UDP, length 128
12:34:57.456789 IP 10.128.1.10.51820 > 203.0.113.1.54321: UDP, length 96
```

- **UDP 51820 しか見えない**
  → WireGuardがすべてを暗号化してUDPに包むため、他のプロトコルは現れない
- **パケットサイズがバラバラ**（96, 128, 148…）
  → 長さだけでは Handshake か Keepalive か判別できない
  → **通信内容は一切わからないようになっている**

実験結果② 内側（wg0）

```
12:34:57.400000 IP 10.8.0.2 > 10.8.0.1: ICMP echo request, id 1, seq 1, length 64
12:34:57.400123 IP 10.8.0.1 > 10.8.0.2: ICMP echo reply,   id 1, seq 1, length 64
12:34:58.400000 IP 10.8.0.2 > 10.8.0.1: ICMP echo request, id 1, seq 2, length 64
12:34:58.400087 IP 10.8.0.1 > 10.8.0.2: ICMP echo reply,   id 1, seq 2, length 64
```

普通のICMPパケットとして見える ＝ WireGuardが復号した後

- IPアドレスはトンネル用のもの（10.8.0.x）が見える
  → 外側で見えた IP は出てこない
- wg0 はすでに復号済みのインターフェース
  → Pingが応答している様子がわかる

実験結果③ バイトで比べてみる① wg0側（復号済み）

```
sudo tcpdump -ni wg0 -XX icmp
```

```
05:12:33.223520 IP 10.128.1.31 > 10.0.1.2: ICMP echo reply, id 28373, seq 245, length 64
    0x0000:  4500 0054 4ac2 0000 4001 1947 0a80 011f  E..TJ...@..G....
    0x0010:  0a00 0102 0000 e6ca 6ed5 00f5 6a3e 0a41  ........n...j>.A
    0x0020:  0003 49e5 0809 0a0b 0c0d 0e0f 1011 1213  ..I.............
    ...（ICMPデータ 連番続く）
```

- `0x0000` 先頭が `45` → IPv4ヘッダ（バージョン4・ヘッダ長20B）
- `0x0000` の `0a80 011f` → 送信元IP 10.128.1.31 がそのまま読める
- `0x0020` 以降の `0809 0a0b 0c0d...` → ICMPデータ部の連番パターンが見える

→ **構造が丸見え。ヘッダもデータも人間が解読できる**

実験結果③ バイトで比べてみる② ens18側（暗号化済み）

```
sudo tcpdump -ni ens18 -XX udp port 51820
```

```
    0x0000:  aabb ccdd eeff 1122 3344 5566 0800  Ethernet（MACアドレス）
    0x000E:  4500 xxxx xxxx 4011 xxxx cb00 7101  E.......
             ↑ 45=IPv4               ↑ 203.0.113.1（送信元IP 読める！）
    0x001E:  0a80 010a xxxx ca6c xxxx xxxx
             ↑ 10.128.1.10（宛先IP 読める！）   ↑ ca6c=51820（port 読める！）
    0x002A:  0400 0000 xxxx xxxx xxxx xxxx xxxx  WireGuardヘッダ（Type=4）
    0x003A:  ////// 暗号化ペイロード ここから↓ //////
    ...
    0x04c0:  5a13 7574 edb3 b867 ff55 6288 b99a  Z.ut...g.Ub.....
    ...
    0x05d0:  eece 93c4 bc74  ← 末尾16B = Poly1305 MAC
```

- IP・UDP ヘッダは**平文**
- **0x002A（WireGuardヘッダ）以降**は完全ランダム → 内側のIPもデータも解読できない

まとめ

- WireGuard を構築してパケットキャプチャした
- **外側（ens18）**：UDP 51820 のみ。中身は見えない
- **内側（wg0）**：復号後の普通のIPパケット
- パケットキャプチャから VPN の動きの理解を深めることができた