インターネットVPNの選択肢にWireGuardはいかがですか？

株式会社Ping-t
谷岡 英治(CCIE #52766)

自己紹介

- 谷岡 英治（CCIE #52766）
- 株式会社Ping-t
  （ https://ping-t.com/ ）

谷岡 英治

株式会社Ping-t

好きなポート番号：179, 646, 5432, 51820

Agenda

- インターネット VPN のおさらい
- WireGuard とは
- VPN サーバの例
- Site-to-Siteの例

インターネット VPN のおさらい

- インターネットを経由する仮想プライベートネットワーク
- 接続形態
  - Point-to-Point, Point-to-Multipoint
  - Site-to-Site
- よく使われる暗号化トランスポート
  - IPsec
  - SSL/TLS
  - **WireGuard**

WireGuard とは

- Jason A. Donenfeld によって開発
- Layer 3 暗号化トンネル for IPv4 and IPv6
  - UDP ベース
  - 4in6 / 6in4 対応
- カーネルツリーにマージ
  - Linux >= 5.6.0, OpenBSD >= 6.8
  - Compat Linux kernel module for 3.10 ≤ kernel ≤ 5.5.y もあり
  - 対応していない環境には Go / Rust 言語による実装あり
- いわゆる「アプリケーション」ではない
  - OS からは単なるインターフェースに見える（スキトオル。）

WireGuardの魅力

- モダンな暗号化
  - Curve25519(ECDH鍵交換), ChaCha20(暗号化), Poly1305(認証), BLAKE2s(ハッシュ), SipHash24(ハッシュテーブル鍵), HKDF(鍵導出)
- シンプル
  - ソースコード：約 6,000 行なので、監査が容易 = 脆弱性の入る余地が少ない
    - OpenVPN: 約 12 万行 （別途 OpenSSL の分もある）
    - StrongSwan: 約 58 万行
    - SoftEther: 約 42 万行
  - コンフィグレーション：1ピアあたり2行（必須定義：ピアの公開鍵、Allowed IPs）
  - 起動プロセス：カーネル（モジュール）有効化、インターフェース有効化、設定反映
- 高速
  - カーネル内で処理
  - コネクションレス（UDPベース）

ソースコード行数（Linux 5.7.6）

```
$ wc -l linux-5.7.6/drivers/net/wireguard/*.(c|h)
     377 allowedips.c
      59 allowedips.h
     236 cookie.c
      59 cookie.h
     459 device.c
      65 device.h
      63 main.c
     128 messages.h
     636 netlink.c
      12 netlink.h
     831 noise.c
     135 noise.h
     237 peer.c
      83 peer.h
     (続く→)
```

```
(→続き)
     221 peerlookup.c
      64 peerlookup.h
      55 queueing.c
     208 queueing.h
     223 ratelimiter.c
      19 ratelimiter.h
     596 receive.c
     422 send.c
     425 socket.c
      44 socket.h
     243 timers.c
      31 timers.h
       1 version.h
    5932 total
```

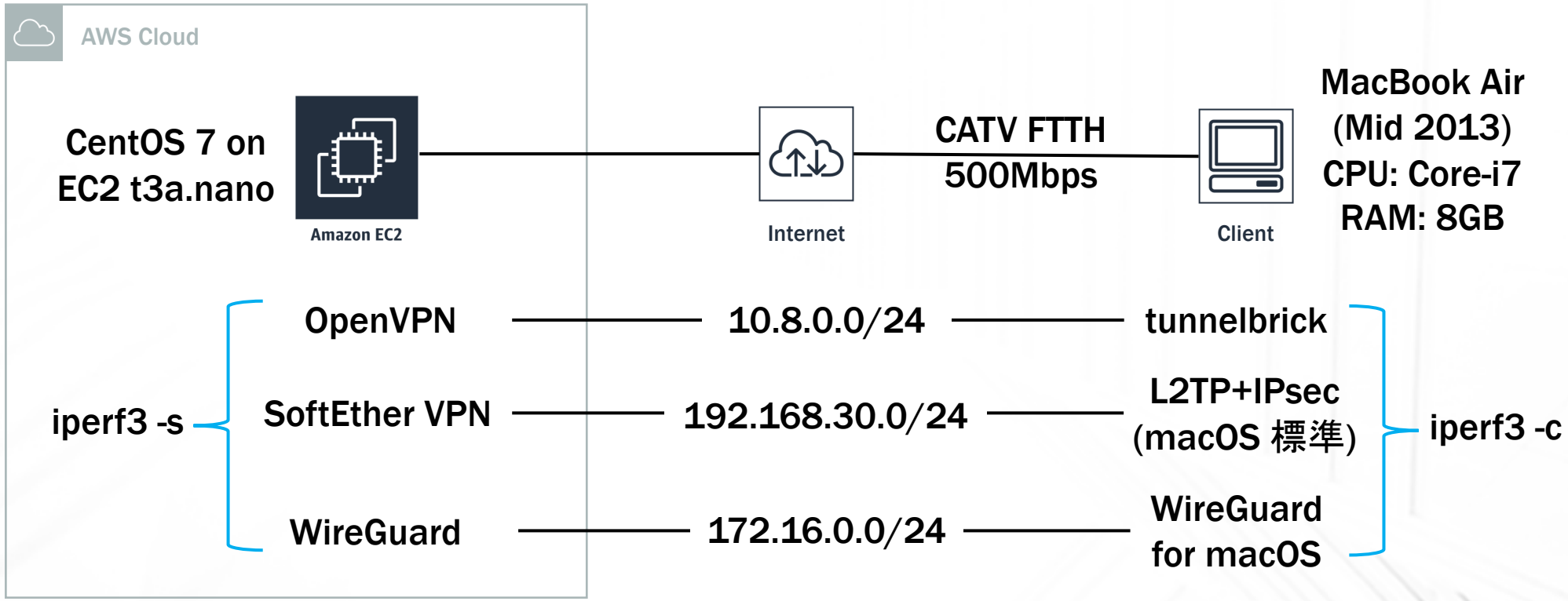
ソースコード行数（OpenBSD-current）

```
$ cd openbsd/src/sys
$ (cd net; wc -l if_wg.(c|h) wg_noise.(c|h) wg_cookie.(c|h))
    2736 if_wg.c
     109 if_wg.h
    1344 wg_noise.c
     195 wg_noise.h
     697 wg_cookie.c
     131 wg_cookie.h
    5212 total
$ (cd crypto; wc -l  blake2s.(c|h) curve25519.(c|h))
     241 blake2s.c
      68 blake2s.h
     890 curve25519.c
      43 curve25519.h
    1242 total
```

転送速度の実測

- 環境
  - サーバとクライアントはそれぞれ同一マシン上で各ソフトウェアをインストール。
    単独で接続して、転送速度とその際のCPU利用率を測定
    - 測定には iperf3 と dstat を使用
  - サーバ：AWS EC2 t3a.nano インスタンス
    - OS: CentOS 7
    - サーバソフトウェアは yum でインストールできるものはそちらを利用。個別のインストールが必要なものは公式の手順に従ってインストールした。
  - クライアント：Macbook Air(Mid 2013) Core i7 / RAM 8GBモデル
    - OS: macOS Catalina 10.15.4
    - OpenVPN クライアント: tunnelbrick（バージョン2.4.8、OpenSSL バージョン1.1.1e）
    - IPsec クライアント: OS 標準の L2TP over IPsec
    - WireGuard クライアント: AppStore よりインストール（0.0.20191105）
- 接続回線：自宅の CATV インターネット（光ファイバー 500Mbps 契約）

構成イメージ


AWS Cloud
CentOS 7 on
EC2 t3a.nano
Amazon EC2
Internet
CATV FTTH
500Mbps
Client
MacBook Air
(Mid 2013)
CPU: Core-i7
RAM: 8GB
iperf3 -s
OpenVPN
10.8.0.0/24
tunnelbrick
SoftEther VPN
192.168.30.0/24
L2TP+IPsec
(macOS 標準)
WireGuard
172.16.0.0/24
WireGuard
for macOS
iperf3 -c

測定例：OpenVPN

```
[centos@ip-172-31-34-71 ~]$ openvpn --version
OpenVPN 2.4.8 x86_64-redhat-linux-gnu [Fedora EPEL patched]
[SSL (OpenSSL)] [LZO] [LZ4] [EPOLL] [PKCS11] [MH/PKTINFO]
[AEAD] built on Nov  1 2019
library versions: OpenSSL 1.0.2k-fips  26 Jan 2017, LZO 2.06
（略）
```

測定例：OpenVPN

```
$ iperf3 -c 10.8.0.1
Connecting to host 10.8.0.1, port 5201
[  5] local 10.8.0.6 port 64550 connected to 10.8.0.1 port 5201
[ ID] Interval           Transfer     Bitrate
[  5]   0.00-1.00   sec  7.17 MBytes  60.1 Mbits/sec
[  5]   1.00-2.00   sec  4.47 MBytes  37.5 Mbits/sec
[  5]   2.00-3.00   sec  2.91 MBytes  24.4 Mbits/sec
[  5]   3.00-4.00   sec  3.52 MBytes  29.6 Mbits/sec
[  5]   4.00-5.00   sec  4.32 MBytes  36.3 Mbits/sec
[  5]   5.00-6.00   sec  3.14 MBytes  26.3 Mbits/sec
[  5]   6.00-7.00   sec  2.86 MBytes  24.0 Mbits/sec
[  5]   7.00-8.00   sec  1.51 MBytes  12.7 Mbits/sec
[  5]   8.00-9.00   sec  2.49 MBytes  20.8 Mbits/sec
[  5]   9.00-10.00  sec  2.70 MBytes  22.6 Mbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate
[  5]   0.00-10.00  sec  35.1 MBytes  29.4 Mbits/sec                  sender
[  5]   0.00-10.00  sec  33.2 MBytes  27.8 Mbits/sec                  receiver

iperf Done.
```

測定例：OpenVPN

```
----system---- ----total-cpu-usage---- -net/total- --io/total- ---system--
     time     |usr sys idl wai hiq siq|  recv  send|  read  writ|  int   csw
02-04 13:13:49|  0   0 100   0   0   0|2006B  948B|    0     0 |  240   413
02-04 13:13:50|  6  10  83   0   0   1|  10M  403k|    0     0 |8253    15k
02-04 13:13:51|  6  11  82   0   0   1|9984k  346k|    0     0 |8968    18k
02-04 13:13:52|  4   7  88   0   0   1|6567k  236k|    0     0 |5730    11k
02-04 13:13:53|  4   8  87   0   0   1|7477k  238k|    0     0 |7436    16k
02-04 13:13:54|  5  12  82   0   0   1|9308k  298k|    0     0 |8697    18k
02-04 13:13:55|  5   7  87   0   0   1|7288k  264k|    0     0 |5912    12k
02-04 13:13:56|  4   7  89   0   0   1|5969k  194k|    0     0 |5747    12k
02-04 13:13:57|  2   4  93   0   0   1|3385k  126k|    0     0 |3521  6914
02-04 13:13:58|  4   7  88   0   0   1|5919k  193k|    0  6.00 |5048  9769
02-04 13:13:59|  3   7  90   0   0   1|5229k  172k|    0     0 |5200    11k
02-04 13:14:00|  1   2  97   0   0   0|1016k   37k|    0     0 |1402  2706
02-04 13:14:01|  1   0 100   0   0   0|2462B  385B|    0     0 |  280   455
```

idl からみて、全体的に7～18%程度のCPU利用が見える

測定例：SoftEther VPN

バージョンは `v4.34-9744-beta-2020.03.20` を使用

測定例：SoftEther VPN

```
$ iperf3 -c 192.168.30.200
Connecting to host 192.168.30.200, port 5201
[  5] local 192.168.30.10 port 64681 connected to 192.168.30.200 port 5201
[ ID] Interval           Transfer     Bitrate
[  5]   0.00-1.00   sec  14.2 MBytes   119 Mbits/sec
[  5]   1.00-2.00   sec  8.20 MBytes  68.8 Mbits/sec
[  5]   2.00-3.00   sec  12.1 MBytes   102 Mbits/sec
[  5]   3.00-4.00   sec  12.3 MBytes   103 Mbits/sec
[  5]   4.00-5.00   sec  11.8 MBytes  99.2 Mbits/sec
[  5]   5.00-6.00   sec  12.3 MBytes   103 Mbits/sec
[  5]   6.00-7.00   sec  11.9 MBytes   100 Mbits/sec
[  5]   7.00-8.00   sec  12.4 MBytes   104 Mbits/sec
[  5]   8.00-9.00   sec  11.6 MBytes  97.0 Mbits/sec
[  5]   9.00-10.00  sec  11.6 MBytes  97.5 Mbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate
[  5]   0.00-10.00  sec   118 MBytes  99.4 Mbits/sec                  sender
[  5]   0.00-10.00  sec   118 MBytes  99.0 Mbits/sec                  receiver

iperf Done.
```

測定例：SoftEther VPN

```
----system---- ----total-cpu-usage---- -net/total- --io/total- ---system--
     time     |usr sys idl wai hiq siq|  recv   send|  read  writ|  int   csw
02-04 13:15:58|  0   0 100   0   0   0|  66B   428B|    0     0 |  204   384
02-04 13:15:59|  4   5  90   0   0   1|4402k   211k|    0  14.0 |3233  6999
02-04 13:16:00| 23  22  53   0   0   2|  26M   517k|    0  8.00 |  13k   23k
02-04 13:16:01| 23  22  52   0   0   2|  23M   273k|    0     0 |  13k   28k
02-04 13:16:02| 29  25  43   0   0   2|  27M   291k|    0     0 |  15k   32k
02-04 13:16:03| 27  29  42   0   0   2|  27M   311k|    0     0 |  15k   33k
02-04 13:16:04| 27  26  45   0   0   2|  26M   620k|    0     0 |  15k   32k
02-04 13:16:05| 28  26  44   0   0   2|  26M   320k|    0     0 |  14k   33k
02-04 13:16:06| 29  25  43   0   0   2|  27M   322k|    0     0 |  14k   32k
02-04 13:16:07| 28  28  42   0   0   2|  27M   307k|    0     0 |  14k   32k
02-04 13:16:08| 26  25  47   0   0   2|  25M   303k|    0     0 |  13k   29k
02-04 13:16:09| 25  21  53   0   0   2|  22M   320k|    0     0 |  12k   26k
02-04 13:16:10|  0   0 100   0   0   0| 220B   496B|    0     0 |  207   398
```

idl からみて、全体的に50%以上のCPU利用が見える

測定例：WireGuard

```
[centos@ip-172-31-34-71 ~]$ rpm -q wireguard-dkms
wireguard-dkms-0.0.20200318-1.el7.noarch
```

測定例：WireGuard

```
$ iperf3 -c 172.16.0.1
Connecting to host 172.16.0.1, port 5201
[  5] local 172.16.0.2 port 64754 connected to 172.16.0.1 port 5201
[ ID] Interval           Transfer     Bitrate
[  5]   0.00-1.00   sec  14.0 MBytes   117 Mbits/sec
[  5]   1.00-2.00   sec  15.8 MBytes   133 Mbits/sec
[  5]   2.00-3.00   sec  16.2 MBytes   136 Mbits/sec
[  5]   3.00-4.00   sec  17.6 MBytes   148 Mbits/sec
[  5]   4.00-5.00   sec  14.0 MBytes   117 Mbits/sec
[  5]   5.00-6.00   sec  13.5 MBytes   113 Mbits/sec
[  5]   6.00-7.00   sec  6.99 MBytes  58.8 Mbits/sec
[  5]   7.00-8.00   sec  11.6 MBytes  97.4 Mbits/sec
[  5]   8.00-9.00   sec  11.2 MBytes  93.6 Mbits/sec
[  5]   9.00-10.00  sec  11.5 MBytes  96.2 Mbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate
[  5]   0.00-10.00  sec   132 MBytes   111 Mbits/sec                  sender
[  5]   0.00-10.00  sec   131 MBytes   110 Mbits/sec                  receiver

iperf Done.
```

測定例：WireGuard

```
----system---- ----total-cpu-usage---- -net/total- --io/total- ---system--
     time     |usr sys idl wai hiq siq|  recv  send|  read  writ|  int   csw
02-04 13:17:20|  0   0 100   0   0   0|  66B  428B|    0  10.0 |  209   385
02-04 13:17:21|  3   4  91   0   0   2|9981k  430k|    0     0 |5189  7402
02-04 13:17:22|  9  13  72   0   0   5|  27M 1088k|    0     0 |  15k   21k
02-04 13:17:23| 11  16  65   0   0   8|  35M 1403k|    0     0 |  19k   27k
02-04 13:17:24| 12  17  61   0   0  10|  38M 1541k|    0     0 |  20k   30k
02-04 13:17:25| 13  16  65   0   0   6|  33M 1343k|    0     0 |  18k   27k
02-04 13:17:26|  8  12  72   0   0   7|  29M 1159k|    0     0 |  14k   19k
02-04 13:17:27|  8  10  77   0   0   5|  25M 1067k|    0     0 |  13k   17k
02-04 13:17:28|  7  12  76   0   0   5|  22M  906k|    0  6.00 |  14k   25k
02-04 13:17:29|  8  12  74   0   0   7|  24M  969k|    0  14.0 |  15k   26k
02-04 13:17:30|  9  13  72   0   0   6|  26M 1043k|    0     0 |  17k   28k
02-04 13:17:31|  4   7  86   0   0   3|  13M  563k|    0     0 |8152    13k
02-04 13:17:32|  0   0 100   0   0   0| 136B  904B|    0     0 |  161   311
```

idl からみて、全体的に20～40%程度のCPU利用が見える

測定結果から

	OpenVPN	SoftEther VPN	WireGuard
CPU利用率	7～18%	50%前後	20～40%
転送速度（Bitrate）	約 30 Mbps	約 100 Mbps	約 110 Mbps
アドバンテージ	低 CPU 利用率	高速転送	中 CPU 利用率 (OpenVPNの 2 倍程度) 高速転送 (ほぼ SoftEther と同等)

VPNサーバとして

- Point-to-Point, Point-to-Multipoint
  - よくある「リモートVPN」
- サーバ：各種 Linux distro, (Free|Open)BSD など
  - Linux の場合、5.6 以前のカーネル用に kmod / dkms が提供されている
  - Go 言語での実装により、ユーザスペースでも動作する
  - Rust によるユーザスペース実装(Boring Tun)もある
- クライアント：Windows, macOS, Android, iOS(iPhone, iPad) など
  - Windows, Androidアプリは日本語化済み
  - macOS, iOSアプリは翻訳済み（翻訳版ビルド未提供）

サーバ構築

- ソフトウェア環境構築
- WireGuard 有効化（カーネルモジュールなど）
- インターフェース作成、アドレス割当
- サーバ用の秘密鍵、公開鍵の作成
- 待受ポート、秘密鍵の設定
- ピア（クライアント）の秘密鍵、公開鍵の作成
- ピアの登録
- （必要であれば）NA(P)Tの設定

ソフトウェア環境構築（CentOS 7の場合）

```
$ sudo yum install epel-release elrepo-release
$ sudo yum install yum-plugin-elrepo
$ sudo yum install kmod-wireguard wireguard-tools
```

以上

各環境向けのインストール手順は https://www.wireguard.com/install/

サーバ構築例

長い…

WireGuard 有効化

```
# modprobe wireguard
```

インターフェース作成、アドレス割当

```
# ip link add dev wg0 type wireguard
# ip addr add dev wg0 10.0.0.1/24
```

サーバ用の秘密鍵、公開鍵の作成

```
# touch privkey; chmod 400 privkey; wg genkey > privkey
# wg pubkey < privkey > pubkey
```

秘密鍵、待受ポートの設定

```
# wg set wg0 private-key privkey listen-port 51820
```

ピア用の秘密鍵、公開鍵の作成

```
# touch cl1-privkey; chmod 400 cl1-privkey; wg genkey > cl1-privkey
# wg pubkey < cl1-privkey > cl1-pubkey
```

ピアの登録

```
# wg set wg0 peer $(cat cl1-pubkey) allowed-ips 10.0.0.11/32
```

WireGuard 起動

```
# ip link set up wg0
```

wg showconf/syncconf/addconf

- 手で設定したものを書き出しておく

```
# wg showconf wg0 > wg0.conf
```

```
[Interface]
ListenPort = 51820
PrivateKey = qJ3Nn1IBjeAxTBMRaZ1Opk0CPuK3uFGnVNIwl0+hH2M=

[Peer]
PublicKey = 3jfR5BnhNvoymNwzEkVsxr8JU0L+J9oeK7FyW6SV2Sc=
AllowedIPs = 10.0.0.11/32
```

- wg0.conf を適宜修正
- 設定ファイルを適用する
  - `# wg syncconf wg0 wg0.conf`
  - `# wg addconf wg0 wg0-addpeer.conf`
- wgインターフェースの作成、アドレス設定、ルーティング編集まではやってくれない…

wg-quick

- 設定ファイルに拡張パラメータを追加して、wg インターフェースの作成、アドレス設定、ルーティングの登録、DNSリゾルバ指定までやってくれる。
- PreUp/PostUp/PreDown/PostDown パラメータで、iptables ルールの書き換えなども可能（bash が理解できるコマンドラインなら何でも実行可能）

```
# wg-quick up wg0
[#] ip link add wg0 type wireguard
[#] wg setconf wg0 /dev/fd/63
[#] ip -4 address add 10.0.0.1/24 dev wg0
[#] ip link set mtu 1420 up dev wg0
[#] ip -4 route add 10.0.0.11/32 dev wg0

# wg-quick down wg0
[#] ip link delete dev wg0
```

Allowed-IPs

- フィルタリング＋ルーティング
  - 送信時は、パケットの宛先となるピアを特定するのに使用する
  - 受信時は、パケットの送信元の判断に使用する
- AllowedIPs = 192.168.0.0/24, 10.0.0.1/32
  →192.168.0.0/24（トンネルの向こうのネットワーク） および
  10.0.0.1/32（トンネルのエンドポイントアドレス）へ（から）のパケットを許可
- AllowedIPs = 0.0.0.0/0
  →全てのパケットを許可
  主にVPNクライアントが全てのトラフィックをトンネル経由にしたい場合に設定する
  （同時に複数のピアに設定すると宛先が決定できなくなる！）
- 特定トラフィックだけVPN経由にしたいときは適切に！

クライアント（ピア）の設定

- サーバからもらった情報を登録
  - 自身の秘密鍵
  - サーバの公開鍵
  - サーバの待受アドレス、ポート
  - 通信可能とする宛先（Allowed IPs）

クライアント 設定例

```
[Interface]
PrivateKey = （cl1-privkey の内容）
ListenPort = 51820
Address = 10.0.0.11/32

[Peer]
PublicKey = （サーバの pubkey の内容）
AllowedIPs = 0.0.0.0/0
Endpoint = （サーバの物理IPアドレス）:51820
```

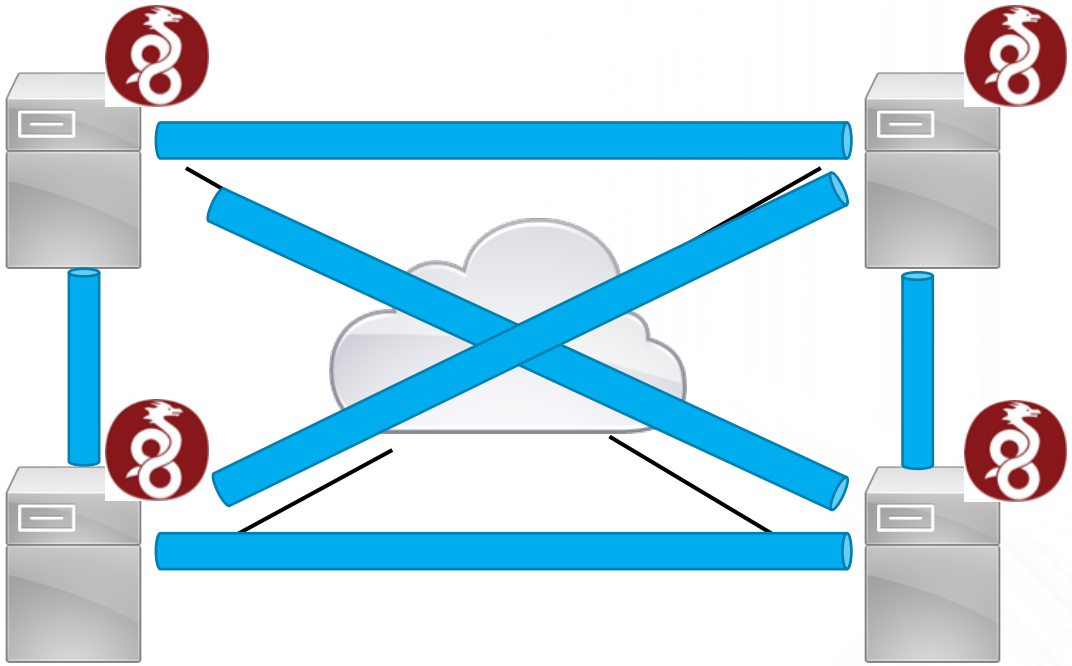
QR コードにすることで、Android / iOS アプリで読み込んで登録できる

Site-to-Site

Eth0:
- 192.0.2.1/24
- 2001:DB8:1234:1::1/64

Eth1:
- 10.1.1.0/24
- 2001:DB8:1:1::0/64

Wg0:
- 192.168.0.1/24
- 2001:DB8:12:34::1/64



Eth0:
- 198.51.100.1/24
- 2001:DB8:1234:2::1/64

Eth1:
- 10.2.2.0/24
- 2001:DB8:2:2::0/64

Wg0:
- 192.168.0.2/24
- 2001:DB8:12:34::2/64

Eth0:
- 2001:DB8:1234:3::1/64

Eth1:
- 10.3.3.0/24
- 2001:DB8:3:3::0/64

Wg0:
- 192.168.0.3/24
- 2001:DB8:12:34::3/64

Eth0:
- 2001:DB8:1234:4::1/64

Eth1:
- 10.4.4.0/24
- 2001:DB8:4:4::0/64

Wg0:
- 192.168.0.4/24
- 2001:DB8:12:34::4/64

まとめ

- インターネット VPN の新たな選択肢として WireGuard が出てきた
  - UDPベースのレイヤ3トンネル
  - モダンな暗号化
  - カーネルに取り込まれている Linux, OpenBSD では高速に動作する
  - 設定はとてもシンプル
- クライアント側アプリも主要なものは揃っている
  - Android
  - iOS, macOS
  - Windows
- Point-to-Point, Point-to-Multipoint, Multipoint-to-Multipoint 構成可

おまけ

Eiji Tanioka

taniokaeiji

Preferred Languages

Japanese (native)

Projects | Activity | To Do

Search projects

All | Own | Manage | Translation

WireGuard — Languages — proofreader — •••

参考資料

- 本家サイト https://www.wireguard.com/
- [CB16] WireGuard:次世代耐乱用性カーネルネットワークトンネル by Jason Donenfeld
  https://www.slideshare.net/codeblue_jp/cb16-donenfeld-ja
- Linux カーネルツリーへのマージ
  https://github.com/torvalds/linux/commit/bd2463ac7d7ec51d432f23bf0e893fb371a908cd
- OpenBSD へマージされることが確定したメール
  https://marc.info/?l=openbsd-cvs&m=159274150512676&w=2
- SpeakerDeck: 作って理解する WireGuard
  https://speakerdeck.com/fadis/zuo-tuteli-jie-suruwireguard
- BoringTun
  https://github.com/cloudflare/boringtun

ソースコード行数（Linux 5.7.6 crypto周り）

```
$ find linux-5.7.6 -name "blake2s*" -exec wc -l {} ¥;
     169 linux-5.7.6/crypto/blake2s_generic.c
     106 linux-5.7.6/include/crypto/blake2s.h
      24 linux-5.7.6/include/crypto/internal/blake2s.h
     256 linux-5.7.6/arch/x86/crypto/blake2s-core.S
     231 linux-5.7.6/arch/x86/crypto/blake2s-glue.c
     622 linux-5.7.6/lib/crypto/blake2s-selftest.c
     111 linux-5.7.6/lib/crypto/blake2s-generic.c
     126 linux-5.7.6/lib/crypto/blake2s.c
$ find linux-5.7.6 -name "curve25519*" -exec wc -l {} ¥;
      90 linux-5.7.6/crypto/curve25519-generic.c
      71 linux-5.7.6/include/crypto/curve25519.h
    1514 linux-5.7.6/arch/x86/crypto/curve25519-x86_64.c
     135 linux-5.7.6/arch/arm/crypto/curve25519-glue.c
    2062 linux-5.7.6/arch/arm/crypto/curve25519-core.S
      24 linux-5.7.6/lib/crypto/curve25519-generic.c
      35 linux-5.7.6/lib/crypto/curve25519.c
     864 linux-5.7.6/lib/crypto/curve25519-fiat32.c
    1321 linux-5.7.6/lib/crypto/curve25519-selftest.c
     788 linux-5.7.6/lib/crypto/curve25519-hacl64.c
```

ソースコードの行数（OpenVPN）

```
$ git clone https://github.com/OpenVPN/openvpn.git
$ cd openvpn
$ git log -1 | head -n 3
commit ec33bae311e7f3549b05de4a4c92fa7bc7144d29
Author: Gert Doering <gert@greenie.muc.de>
Date:   Fri Jun 26 10:27:43 2020 +0200
$ cd src
$ find . -name "*.[ch]" -exec wc -l {} ¥; | awk '{n += $1}; END{print n}'
121337
```

ソースコードの行数（StrongSwan）

```
$ git clone https://github.com/strongswan/strongswan.git
$ cd strongswan
$ git log -1 | head -n 3
commit feda4a3d37728bc84f12a95f86f034cf835e6919
Author: Tobias Brunner <tobias@strongswan.org>
Date:   Wed Jul 1 13:49:58 2020 +0200
$ cd src
$ find . -name "*.[ch]" -exec wc -l {} ¥; | awk '{n += $1}; END{print n}'
580868
```

ソースコードの行数（SoftEther VPN）

```
$ git clone https://github.com/SoftEtherVPN/SoftEtherVPN_Stable.git
$ cd SoftEtherVPN_Stable
$ git log -1 | head -n 3
commit ec3d052e218281dc1aa734f1bf84cf42127744aa
Author: dnobori <da.git@softether.co.jp>
Date:   Mon Apr 6 00:18:10 2020 +0900
$ cd src
$ find . -name "*.[ch]" -exec wc -l {} ¥; | awk '{n += $1}; END{print n}'
424482
```