

# Linux ネットワーク、踏み台 (ssh トンネル)

## 目次

はじめに .....	1
1. ネットワークの前提知識 .....	1
1.1. プロトコル (OSI の 7 階層 L1~L7) .....	1
1.2. ネットワーク…MAC アドレスと IP アドレス .....	2
1.3. ハブ、ブリッジ、スイッチ、ルータ、ゲートウェイ (コリジョン、ブロードキャスト) 2 .....	
1.4. ルーティング .....	3
1.4.1. IP アドレスの構成 .....	3
1.4.2. サブネットマスクとネットワークアドレス .....	3
1.4.3. サブネットとルータ .....	3
1.4.4. デフォルトゲートウェイ .....	4
1.4.5. ネットワーク内の情報伝播 (盗聴の可能性) .....	6
1.4.6. ネットワーク間のホップ (デフォルトゲートウェイの先) .....	6
1.5. ポート番号 .....	6
1.6. エンド・ツー・エンド .....	7
2. ネットワークとドメイン名 .....	8
2.1. DNS (Domain Name System) .....	8
2.2. DNS 以外の名前解決 .....	9
2.3. ネットワークと IP アドレス .....	9
2.3.1. グローバル IP アドレス .....	9
2.3.2. プライベート IP アドレス .....	10
2.3.3. 特殊な IP アドレス .....	10
3. NAT/PAT (NAPT) .....	10
4. ネットワーク・セキュリティ .....	11
5. 経路の暗号化 (SSL/TLS、SSH、IPSec) .....	11
6. 組織内ネットワークの分界/境界 .....	12
7. ネットワークの接続 .....	13
7.1. リスク .....	13
7.2. 対策 .....	13
7.3. 個人識別・認証の方法 .....	13
7.4. 公開鍵/秘密鍵の作り方 .....	14
7.5. 踏み台サーバ (ssh ポートフォワーディング/トンネリング) .....	15

# Linux ネットワーク、踏み台 (ssh トンネル)

はじめに

Linux (Unix) サーバを使うとネットワークを越えた作業環境が簡単に構築できます。少々危険な匂いがしますが (実際、設定や運用方法によってかなり危険になります)、適切な環境を作ることによってネットワーク全体のセキュリティレベルを向上させることができます。

例えば顧客情報や営業秘密の情報流出を防ぐために情報を保管しているサーバをスタンドアロンで運用すれば安全でしょうか?…サーバから外部媒体や紙に出力して使うのではセキュリティレベルはむしろ下がります。内容によってはメモ書きですらマズイでしょう。

リスクを重く考えるのであれば要員を含むシステムの全てを外部と切り離せばよさそうですが、それは不可能です。外部からの入力情報 (日々のトランザクション、制御シグナル、時刻、etc.) を必要としないシステムは恐らく存在しません。ましてやクラウドが当たり前の環境にもなっています。これらの入出力データの多くはネットワーク経由で行き来しています。また、開発の現場では利用したいソフトウェアやそのマニュアルがネット公開になっていてネットワーク接続なしには仕事ができない状況も発生しています。日常業務ではフィッシングメールやランサムウェアの被害がしばしばニュースになりますが、今から FAX を使った取引だけに戻すのは困難です (通信会社等の統計資料によると FAX の設置数は逡減傾向にあり、EC 取引は拡大しています)。

上の例でいえば、①システムの運用、②開発、③日常業務のそれぞれ異なる用途に応じたリスク対策・セキュリティレベルの環境が必要になります。そして、その相互接続をセキュアに行うために ssh(Secure SHell)を使ったトンネリング、IPSec や https を使った VPN(Virtual Private Network) 等のツールがあります。

## 1. ネットワークの前提知識

ネットワーク間を接続したり逆に境界を設けるために必要な前提知識を確認しておきます。

### 1.1. プロトコル (OSI の 7 階層 L1~L7)

ネットワーク上の各種の装置 (PC、サーバ、ルータ etc.) が信号をやり取りできるのは届いた電気信号 (以下、パケット) の意味を理解できるからです。通信を成立させるためには決められたフォーマットのパケットを決められた手順で交換する必要があり、このフォーマット・手順がプロトコル (Protocol) です。

プロトコルは階層になっていて、OSI 参照モデルでは物理層 (第 1 層: Layer1)、データリンク層 (第 2 層: L2)、ネットワーク層 (第 3 層: L3)、トランスポート層 (第 4 層: L4) …等の 7 階層で表現されます<sup>1</sup>。ssh、http/https は最上位階層のアプリケーション層にあたり<sup>2</sup>ます。

---

<sup>1</sup> 各層はスタック (Stack) とかプロトコル・スタックと呼ばれます。実装としての OSI はメインフレームで試行された (異なるメーカー間では仕様の解釈相違等により接続は困難でした) ことがある程度で、ネットワーク層より上は TCP/IP プロトコルが主流です

<sup>2</sup> 実際は TCP (OSI での第 4 層) /IP (OSI での第 3 層) の上で動作するアプリケーションなので OSI モデルの 5 層~7 層に相当します

# Linux ネットワーク、踏み台 (ssh トンネル)

## 1.2. ネットワーク…MAC アドレスと IP アドレス

ネットワークに接続している機器<sup>3</sup>を OSI 第 2 層 (L2) で識別するのが MAC (Media Access Control) アドレスで、MAC アドレスで通信する範囲がローカルネットワーク (LAN、以下単にネットワークと記述した場合はこの範囲) です。通信相手がネットワーク内で見つからない場合に、IP アドレスを判定してパケットの送り先のネットワークを決定します。

IP アドレスは OSI 第 3 層 (L3) のパケットが持っていて、L3 のパケットは L2 のパケット (フレーム<sup>4</sup>) のデータ (ペイロード) として運ばれます (L2 パケット=L2 ヘッダ+L3 パケットです)。

有線 (Ethernet)、無線 (Wi-Fi) 等で L2 のプロトコルが異なりますが、プロトコル・スタックが階層になっているので L3(IP)から上の階層は有線・無線を意識せず通信することができます。

## 1.3. ハブ、ブリッジ、スイッチ、ルータ、ゲートウェイ (コリジョン、ブロードキャスト)


ハブ、ブリッジ、スイッチ、ルータ等は通信を接続するための機器ですが、違いは、主にパケットの振分を行う層の違いにあります。以下、有線 LAN を使って説明します。

### (1) ハブ、ブリッジ、スイッチ (L2)

ハブはリピータとも呼ばれ、届いたパケットを全てのポートに配布します。この接続はパケットの衝突 (コリジョン) の検知と再送が必要な環境なので**コリジョン・ドメイン**と呼ばれます。

ブリッジと L2 スイッチ (スイッチングハブ) はパケットの送信先 MAC アドレスでブリッジ要否と送出先スイッチ・ポートを決定します。この接続はブロードキャスト (一斉同報) パケットが届く範囲<sup>5</sup>なので**ブロードキャスト・ドメイン**と呼ばれます。

L2 スイッチの中にはポートを論理的に分けてブロードキャストを遮断し、VLAN(Virtual LAN)を構成できるものもあります。一つの VLAN をセグメントと呼ぶこともあります。

 **個人利用の Windows 上で LAN アダプタ同士に「ブリッジ接続」の設定が簡単にできますが、これによりネットワークにループを作ってしまうことがあります。ループができると環境によってはフラットニングによる輻輳でネットワークの通信停止を引き起こす場合があります**

### (2) スイッチ (L3)、ルータ、ゲートウェイ

L3 スイッチはネットワークを接続し、IP アドレスでパケットの振分先ネットワークを決定します。スイッチはハードウェア/ファームウェアで作られているためルータよりも高速で動作します。ルータはソフトウェアで処理を行い、IP パケットの振分だけでなくプロトコル変換 (Ethernet、Wi-Fi、DSL/FTTH 等の相互接続) も行うことができます。

ゲートウェイはネットワークの接続を行っているルータ等の装置で、振分先が分からないパケットの送信先のルータをデフォルトゲートウェイ (default gateway) として設定する等で使います。

---

<sup>3</sup> 実際は NIC(Network Interface Card)が一意的 MAC アドレスを持ち、一つの機器が複数の NIC を持つこともあります。NIC は LAN アダプタやネットワークアダプタとも呼ばれます

<sup>4</sup> 層によってパケットに別の呼び方があり、L2 は MAC フレーム、L4 (セッション) はセグメント

<sup>5</sup> **【フラットニング】** ブリッジや L2 スイッチは送信先 MAC アドレスの所在が分からないパケットが届くと接続している全ネットワークにパケットを送出します

# Linux ネットワーク、踏み台 (ssh トンネル)

## 1.4. ルーティング

LAN に接続するためのインタフェースを持っている機器は以下の 2 点を満たす IP アドレスを割り振られることでネットワークに参加することができるようになります。

- ① 重複が無い
- ② ネットワークアドレス (サブネット) が正しい

以下の情報は NIC/LAN アダプタ毎に固定または DHCP を使って設定します。

### 1.4.1. IPアドレスの構成

IP アドレスはネットワークアドレスとホストアドレスを合成したものです。IPv4 の場合は IP アドレスを 32 ビットで表します。例として 192.168.11.21/28 と表記すると、末尾の"/28"で先頭 28 ビットがネットワークアドレス、残りの 4 ビットがホストアドレスであることを示します。

### 1.4.2. サブネットマスクとネットワークアドレス

サブネットマスクは単に“ネットマスク”とも呼び、ネットワークアドレスを表すビット数を全て 1、残りを 0 として 10 進数表示にしたものです。

192.168.11.21/28 のサブネットマスクは先頭 28 ビットを 1 残り 4 ビットを 0 にした 11111111 11111111 11111111 11110000 の 10 進数 255 . 255 . 255 . 240 になります。

[IP アドレス] 192.168.11.21 [ネットマスク] 255 . 255 . 255 . 240 の場合

IP アドレス	1100 0000 . 1010 1000 . 0000 1011 . 0001 0101
サブネットマスク	1111 1111 . 1111 1111 . 1111 1111 . 1111 0000
上記の論理積	1100 0000 . 1010 1000 . 0000 1011 . 0001 0000
10 進数表記に変換	192 . 168 . 11 . 16 …ネットワークアドレス

### 1.4.3. サブネットとルータ

LAN で接続できる距離やハブの階層等の物理的な上限、パケットの衝突 (コリジョン) 発生量、接続機器の管理のし易さから一定の規模を超えるネットワークはサブネットワークに分割します。

サブネットワーク (ブロードキャスト・ドメイン) のネットワークアドレスが同一の相手とは MAC アドレスで通信<sup>6</sup>を行い、ネットワークアドレスが異なる通信先は<sup>7</sup>デフォルトゲートウェイ (ルータ等の IP で振分を行う機器) の MAC アドレスに向けてパケットを送信します。

同一ネットワークに接続されていて、ARP テーブルにキャッシュ済の NIC は arp コマンド (Linux の場合は ip neigh コマンド<sup>8</sup>) で確認することができます。

---

<sup>6</sup> ARP (Address Resolution Protocol-ブロードキャストで実行) によりネットワーク内の IP アドレスと MAC アドレスの対応表 (ARP Table) が作られ、各装置に一時保存 (キャッシュ) されます

<sup>7</sup> 送信先の IP アドレスと自分のネットマスクを使ってネットワークアドレスを計算します。また、同一 LAN 上に MAC アドレスが存在するか否かを確認 (ARP) せずにデフォルト・ゲートウェイに送るという動作は実装依存で OS が変わると変化する可能性があります

参考文献 (RFC) : <https://www.rfc-editor.org/info/rfc0950>

(wiki) : <https://en.wikipedia.org/wiki/Subnetwork>

<sup>8</sup> Linux の arp,ifconfig,netstat 等のネットワーク系コマンドは非推奨になり ip コマンドに置き換わ

# Linux ネットワーク、踏み台 (ssh トンネル)

下記実行例の“物理アドレス”が MAC アドレスで ff-ff-ff-ff-ff-ff はブロードキャストアドレスです。

```
C:\Users\User>arp -a
```

```
インターフェイス: 192.168.11.2 --- 0x3 ...netstat -r コマンドで表示されるインタフェース一覧の 3
```

インターネット アドレス	物理アドレス	種類
192.168.11.1	18-c2-bf-12-cf-ee	動的
192.168.11.255	ff-ff-ff-ff-ff-ff	静的
(中略)		
239.255.255.253	01-00-5e-7f-ff-fd	静的
255.255.255.255	ff-ff-ff-ff-ff-ff	静的

```
インターフェイス: 172.27.16.1 --- 0x26 ...netstat -r コマンドで表示されるインタフェース一覧の 38 (十進表示)
```

インターネット アドレス	物理アドレス	種類
172.27.25.67	00-15-5d-0b-05-07	静的
172.27.31.255	ff-ff-ff-ff-ff-ff	静的
(中略)		
239.255.255.253	01-00-5e-7f-ff-fd	静的
255.255.255.255	ff-ff-ff-ff-ff-ff	静的

## 1.4.4. デフォルトゲートウェイ

デフォルトゲートウェイは送られたきたパケットの IP アドレス (L2 パケットのデータ部から L3 ヘッダを参照) を使って送り出し先のネットワークを決めます。

PC/サーバはデフォルトゲートウェイは一つだけ設定することができます。NIC を複数持ってもデフォルトゲートウェイを複数有効にすることはできません<sup>9</sup>。

デフォルトゲートウェイは、ipconfig(Linux の場合は ip route コマンド)で確認できます。

```
C:\Users\User>ipconfig
```

Windows IP 構成

イーサネット アダプター vEthernet (Default Switch):

```
接続固有の DNS サフィックス . . . . . :  
リンクローカル IPv6 アドレス. . . . . : fe80::f000:2e65:2fe4:7993%38  
IPv4 アドレス . . . . . : 172.17.16.1  
サブネット マスク . . . . . : 255.255.240.0  
デフォルト ゲートウェイ . . . . . :
```

Wireless LAN adapter Wi-Fi:

```
接続固有の DNS サフィックス . . . . . :  
IPv6 アドレス . . . . . : 2404:7a80:2d00:b210:793b:1403:da1b:3bf  
一時 IPv6 アドレス. . . . . : 2404:7a80:2d00:b210:3577:f8de:ce5c:926a  
リンクローカル IPv6 アドレス. . . . . : fe80::793b:1403:da1b:3bf%3  
IPv4 アドレス . . . . . : 192.168.11.2  
サブネット マスク . . . . . : 255.255.255.0  
デフォルト ゲートウェイ . . . . . : fe80::1ac2:bfff:fe12:cfee%3  
192.168.11.1
```

(以下、略)

---

りました。下記に Red Hat Enterprise Linux の ip コマンドチートシートが公開されています  
<https://access.redhat.com/ja/articles/1361373>

<sup>9</sup> 【デフォルトゲートウェイ】 送り出し先を予め決めておけるのであれば NIC 毎に「固定ルート」として設定し、予め決められないものだけデフォルトゲートウェイに送られるようにします

Copyright(C)2022 Future Office Coordinate Service Corporation All Rights Reserved. p. 4

# Linux ネットワーク、踏み台 (ssh トンネル)

ネットワーク間はインターネットプロトコル (IP) で接続され、インターネット上の機器は IP アドレスで識別されます。アプリケーションが特定の IP アドレス宛にパケットを送出したとき、ルーティング・テーブルを参照して送り出す LAN アダプタ (インタフェース) が決定されます。

ルーティングテーブルは自動的に作成され、`netstat -r` か `route print`(Linux の場合は `ip route` コマンド)で確認できます。

```
C:\Users\User>netstat -r
```

```
=====
```

インターフェイス一覧

```
38...00 15 5d b3 a0 2a .....Hyper-V Virtual Ethernet Adapter
3...50 3e aa e9 4f 80 .....Realtek RTL8188EU Wireless LAN 802.11n USB 2.0 Network Adapter
(中略)
```

```
=====
```

IPv4 ルート テーブル

```
=====
```

アクティブ ルート:

ネットワーク宛先	ネットマスク	ゲートウェイ	インターフェイス	メトリック
0.0.0.0	0.0.0.0	192.168.11.1	192.168.11.2	55
(中略)				
172.27.16.0	255.255.240.0	リンク上	172.27.16.1	271
172.27.16.1	255.255.255.255	リンク上	172.27.16.1	271
172.27.31.255	255.255.255.255	リンク上	172.27.16.1	271
192.168.11.0	255.255.255.0	リンク上	192.168.11.2	311
192.168.11.2	255.255.255.255	リンク上	192.168.11.2	311
192.168.11.255	255.255.255.255	リンク上	192.168.11.2	311
255.255.255.255	255.255.255.255	リンク上	192.168.11.2	311
255.255.255.255	255.255.255.255	リンク上	172.27.16.1	271

```
=====
```

固定ルート:

なし

```
=====
```

IPv6 ルート テーブル

```
=====
```

アクティブ ルート:

If	メトリック	ネットワーク宛先	ゲートウェイ
3	311	::/0	fe80::1ac2:bfff:fe12:cfee

(中略)

```
=====
```

固定ルート:

なし

※アクティブ ルートは RIP (Routing Information Protocol) 等で動的に収集された経路情報で...

- ・ ネットワーク宛先とネットマスクが 0.0.0.0 はデフォルトゲートウェイ
  - …固定ルートにもその他のアクティブ ルートにも該当しない場合に、インタフェース欄の LAN アダプタを使って、ゲートウェイ欄の IP アドレスにパケットを送ります
- ・ ネットマスクが 255.255.255.255 になっているネットワーク宛先はホストアドレス
- ・ ネットマスクが 255.255.255.255 以外(255.255.255.0 等)はネットワークアドレス
- ・ ネットワーク宛先のホスト部全ビット 1or 全ビット 0 はブロードキャストアドレス
  - <例> 192.168.11.255 …ネットワークアドレス指定のブロードキャスト (ルータ越えはできません)
  - 255.255.255.255 …ローカル (発信元) ネットワーク向けのブロードキャスト
- ・ 固定ルートはデフォルトゲートウェイに送りたくないアドレスを管理者が手動で設定します
  - `root -p add ネットワーク宛先 mask ネットマスク ゲートウェイ if インタフェース …`のように指定する

# Linux ネットワーク、踏み台 (ssh トンネル)

## 1.4.5. ネットワーク内の情報伝播 (盗聴の可能性)

NIC から送り出されたパケットは同一 LAN に接続されている全ての NIC で受信可能で、パケットキャプチャ、パケットスニファアと呼ばれる LAN 監視用フリーソフトが公開されています。

また、スイッチやルータにはミラーポート、モニターポートというパケットを監視するためのポートがあり、流れている情報を観察することができます。

## 1.4.6. ネットワーク間のホップ (デフォルトゲートウェイの先)

デフォルトゲートウェイから先の動作は、tracert コマンド(Linux の場合は ip route コマンド) にホスト名等を指定することで経路 (ルータ) を確認できます。

```
C:\Users\User>tracert -4 xxxxxx.com
```

```
xxxxxx.com [142.251.nn.nnn] へのルートをトレースしています
```

```
経由するホップ数は最大 30 です:
```

```
 1    3 ms    2 ms    2 ms  AP18C2BF12CFEE [192.168.11.1]
 2   13 ms    7 ms    8 ms  203.136.xx.xx
 3    8 ms    9 ms    6 ms  203.136.yy.yy
 4    6 ms    7 ms    5 ms  142.250.zzz.zz
```

(以下、略)

## 1.5. ポート番号

ポートという言葉はハブやルータ等の差込口でも使われますが、ネットワークでいうポート番号はサービスやアプリケーションを指します。ポート番号は送信元ポートと宛先ポートが L4 パケット (TCP はセグメント、UDP はダイアグラムとも呼ぶ) のヘッダ部分に書かれています。

ポート番号は 16 ビット、0~65535 を使います。サーバ側で使われるポート番号は、Unix の時代からあるようなサービス/アプリケーションは 1023 以下の番号 (システムポート番号と呼ぶ) が振られており、49151 以下は IANA が管理<sup>10</sup>しています。クライアント側のポート番号は DHCP 等の固定ポートのサービスを除き、動的に (TCP の場合であればコネクション確立の都度) 一意な番号が振られていき 65535 に達すると再循環します (開始番号は IANA が 49152 を推奨)。

以下にサービスとしてよく使われるポート番号を IANA のホームページから抜粋します。

Service Name	Port Number	Transport Protocol	Description
ftp-daa	20	tcp	File Transfer [Default Data]
ftp-data	20	sctp	FTP
ftp	21	tcp	File Transfer Protocol [Control]
ssh	22	tcp	The Secure Shell (SSH) Protocol
ssh	22	udp	The Secure Shell (SSH) Protocol
domain	53	tcp	Domain Name Server
domain	53	udp	Domain Name Server
http	80	tcp	
http	80	udp	
https	443	tcp	http protocol over TLS/SSL
https	443	udp	http protocol over TLS/SSL

※ポート番号は tcp と udp の間では競合せず、同一の番号を同時に使うことができます (例:22,53)

※サービス名とポート番号の対応は、/etc/services ファイルに書かれています

<sup>10</sup> IANA-Service Name and Transport Protocol Port Number Registry

<https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml>

## Linux ネットワーク、踏み台 (ssh トンネル)

サーバで実際に開いているポート (サービス/アプリケーション) は `ss` コマンドで確認できます (Windows の場合は `netstat -a` コマンド)

```
adminuser@ubuntu2004:~$ ss -atu
Netid   State    Recv-Q   Send-Q   Local Address:Port      Peer Address:Port      Process
udp     UNCONN   0         0         172.27.31.255:netbios-ns 0.0.0.0:*
udp     UNCONN   0         0         172.27.25.67:netbios-dgm 0.0.0.0:*
udp     UNCONN   0         0         127.0.0.53%lo:domain    0.0.0.0:*
udp     UNCONN   0         0         172.27.25.67%eth0:bootpc 0.0.0.0:*
tcp     LISTEN   0         4096      127.0.0.53%lo:domain    0.0.0.0:*
tcp     LISTEN   0         128      0.0.0.0:ssh             0.0.0.0:*
tcp     LISTEN   0         50       0.0.0.0:microsoft-ds    0.0.0.0:*
tcp     LISTEN   0         50       0.0.0.0:netbios-ssn        0.0.0.0:*
tcp     ESTAB    0         0         172.27.25.67:ssh        172.27.16.1:54878
tcp     LISTEN   0         128      [::]:ssh                [::]:*
tcp     LISTEN   0         50       [::]:microsoft-ds      [::]:*
tcp     LISTEN   0         50       [::]:netbios-ssn       [::]:*
```

※State の LISTEN は接続要求を待ち受けている状態、ESTAB はクライアントと接続確立済です

`ss` コマンドのオプションに `n` を指定するとポート番号を数字で表示できます。

```
adminuser@ubuntu2004:~$ ss -atun
Netid   State    Recv-Q   Send-Q   Local Address:Port      Peer Address:Port      Process
udp     UNCONN   0         0         172.27.31.255:137       0.0.0.0:*
udp     UNCONN   0         0         172.27.25.67:138       0.0.0.0:*
udp     UNCONN   0         0         127.0.0.53%lo:53       0.0.0.0:*
udp     UNCONN   0         0         72.27.25.67%eth0:68    0.0.0.0:*
tcp     LISTEN   0         4096      127.0.0.53%lo:53       0.0.0.0:*
tcp     LISTEN   0         128      0.0.0.0:22             0.0.0.0:*
tcp     LISTEN   0         50       0.0.0.0:445            0.0.0.0:*
tcp     LISTEN   0         50       0.0.0.0:139           0.0.0.0:*
tcp     ESTAB    0         36       172.27.25.67:22        172.27.16.1:54878
tcp     LISTEN   0         128      [::]:22                [::]:*
tcp     LISTEN   0         50       [::]:445                [::]:*
tcp     LISTEN   0         50       [::]:139                [::]:*
```

### 1.6. エンド・ツー・エンド

情報収集で使われるブロードキャスト<sup>11</sup>と専用アプリケーションに同報を送るマルチキャスト<sup>12</sup>を除き、クライアントとサービスはお互いの IP アドレス+ポート番号を固定した 1 : 1 の関係を作り (一対の関係がコネクション/セッション…サーバ側は複数開設できる) 通信を行います。

クライアントアプリ (一方のエンド) はサーバの IP アドレスとサービスのポート番号を宛先に、自分のポート番号と IP アドレスを送信元にしたリクエストを送ります。受信したサーバは宛先に書いてあるポート番号を開いているプロセス (もう一方のエンド) にリクエストの処理を依頼します。

サーバプロセスは処理結果をレスポンスにして、リクエストに書かれていた送信元の IP アドレスとポート番号を宛先にしたパケットを送信します。受信したホストは、宛先のポート番号を開いているプロセスにこのレスポンスを渡します。エンド・ツー・エンドの間の伝送路を構成する機器は自律的に動作するのでクライアントとサーバは経路上の機器の存在を意識する必要がありません。

<sup>11</sup> MAC アドレスを収集 (arp) したり IP アドレスを貰うために DHCP サーバを探すときに使う

<sup>12</sup> ビデオ映像・音声を流すとき等に使う。IP アドレス 224.0.0.0~239.255.255.255 を使い、専用のクライアントアプリが回線上のパケットの宛先 MAC アドレスを判定して受信するか否かを定める



# Linux ネットワークと踏み台 (ssh トンネル)

## 2. ネットワークとドメイン名

ネットワークに接続しているホスト (PC やサーバ) は IP アドレスで識別<sup>13</sup>しますが、DNS サーバでドメイン名として管理することもできます。DNS で管理するドメインはネットワークと一致している必要はありません。

### 2.1. DNS (Domain Name System)

DNS はホストにインストールされているクライアントから DNS のサーバ (ネームサーバ) に問合せを行い、ドメイン名と IP アドレスを相互 (逆引き機能は設定による) に変換します。

#### (1) ドメインの階層

DNS はドメイン名を階層で管理します。

<例>

www.example.co.jp の場合は

第 4 レベルドメイン . 第 3 レベルドメイン . 第 2 レベルドメイン . トップレベルドメイン

www                      example                      co                      jp

…の階層になっています。もっと大きな組織であれば下の例のように階層を増やすこともできます。

(ドメイン名の長さは全体で 253 文字以下という制限があります)

www.develop.example.co.jp

mail.eigyo.example.co.jp

それぞれの階層の各ドメインには「ネームサーバー」が配置され、ドメイン内の名前の管理を行っています。ネームサーバーは配下にあるドメイン名と IP アドレスの対応関係を管理し、その下の階層のドメイン(サブドメイン)の問い合わせには、そのドメインを管理しているネームサーバーの位置を返します。該当のドメインに辿り着くまでの再帰問い合わせは DNS の仕組みの一部として行われるので DNS に問い合わせた側 (アプリ等) は意識する必要がありません。

#### (2) 内部と外部のネームサーバ

問い合わせ先として設定できるネームサーバは 1 カ所<sup>14</sup>なので、自組織内にネームサーバを立てかつインターネット上のサーバも検索する場合はキャッシュサーバを立てて組織内部の名前解決を行い、インターネット上の名前解決は外部の DNS にフォワードするようにします ([為念] キャッシュサーバは攻撃方法が知られていて何度か注意が呼びかけられて<sup>15</sup>います)。

---

<sup>13</sup> ホストは NIC を増設して複数の MAC アドレスと IP アドレスを持つことができます。更に一つの NIC に複数の IP アドレスを設定することもできます

<sup>14</sup> Windows、Linux のどちらも DNS のクライアントは DNS サーバが複数指定されていた場合、先頭のサーバに問い合わせを行いタイムアウトになった場合にだけ次のサーバに問合せを行います。対象が「登録されていない」という応答が返された場合、そこで問い合わせは終了です

<sup>15</sup> JPCERT <https://www.jpccert.or.jp/tips/2017/wr172801.html>

インターネット 10 分講座 : DNS キャッシュポイズニング <https://www.nic.ad.jp/ja/newsletter/No40/0800.html>

# Linux ネットワークと踏み台 (ssh トンネル)

## (3) Windows の DNS サーバ設定

Windows の場合、NIC 毎に問合せ先の DNS サーバを設定できますが、問合せに使われるネームサーバはメトリックが一番小さい NIC に設定されたものだけです。メトリックは手動でも設定できますが、自動設定の場合はインターフェースとゲートウェイの性能からメトリックを算出<sup>16</sup>します。

DNS サーバの動作は、nslookup コマンドで確認することができます。

```
adminuser@ubuntu2004:~$ nslookup focs.co.jp
Server:          127.0.0.53
Address:         127.0.0.53#53
```

```
Non-authoritative answer:
Name:   focs.co.jp
Address: 59.106.171.42
```

※上記の Server が DNS サーバです (この例ではキャッシュサーバが使われている)

## 2.2. DNS 以外の名前解決

ホストの名前解決の方法は Linux でも Windows でもデフォルトで hosts ファイルの優先順位が高くなっています (Linux の場合は/etc/nsswitch.conf に優先順指定)。

hosts ファイルはドメイン名を指定することができ、別名をつけることもできます。

<hosts の記述例> 192.168.11.6 hostname hostname.example.co.jp

## 2.3. ネットワークと IP アドレス

ネットワークは階層を持ちません。DNS と組み合わせるとドメイン名の階層でホストが識別できますが、これは電話帳と同じで索引が業種や名前前で細分化し体系づけされているだけです。

IP アドレスはいくつかの RFC に定義があります。

### 2.3.1. グローバル IP アドレス

IP アドレス体系は RFC791 に記述があり IPv4 では以下のように定義されています。

<RFC 791 抜粋>

上位ビット	フォーマット	クラス	
0	7 ビットがネットワーク、24 ビットがホスト	a	... 0.0.0.0 - 127.255.255.255
10	14 ビットがネットワーク、16 ビットがホスト	b	... 128.0.0.0 - 191.255.255.255
110	21 ビットがネットワーク、8 ビットがホスト	c	... 192.0.0.0 - 223.255.255.255
111	拡張アドレッシングモードへのエスケープ		

※日本では JPNIC が管理<sup>17</sup>。IP アドレスの割り当てでは上記のクラス分けは意識されていません

<sup>16</sup> リモートアクセス利用時におけるメトリックの変動について

<https://jpwinsup.github.io/blog/2021/04/11/Networking/TCPIP/About-metric-fluctuations/>

IPv4 ルートの自動メトリック機能の説明

<https://docs.microsoft.com/ja-JP/troubleshoot/windows-server/networking/automatic-metric-for-ipv4-routes>

<sup>17</sup> JPNIC <https://www.nic.ad.jp/ja/ip/list.html>

世界で使われている IP アドレス(オープンソースライセンス) <https://ipv4.fetus.jp/>

# Linux ネットワークと踏み台 (ssh トンネル)

## 2.3.2. プライベート IP アドレス

組織内だけで使うのはプライベートアドレスで、「RFC1918 プライベートインターネットのアドレス割り当て 1996年2月」に以下のように記述されています。

『プライベートインターネット用の IP アドレススペースの次の 3 つのブロック

10.0.0.0 - 10.255.255.255 (10/8 prefix)  
172.16.0.0 - 172.31.255.255 (172.16/12 prefix)  
192.168.0.0 - 192.168.255.255 (192.168/16 prefix)  
(中略)

ここに定義されたアドレス空間は組織が IANA 他誰とも調整せずに使うことができる (意識)』

※組織内だけで使う分にはこのアドレス空間に限らず勝手に使っても問題は起こりませんが、「この範囲の IP アドレスを使っておけばグローバル IP アドレスと重複することは決して無い」そうです

## 2.3.3. 特殊な IP アドレス

RFC1122 3.2.1.3 Addressing には、以下が定義されています。

- ① 0.0.0.0 …ローカルネットワーク上の自ホスト (This host on this network)  
…4.2 BSD Unix とその派生の場合、ブロードキャストアドレスに使われる
- ② <ネットワークアドレス=0>、<ホストアドレス> …ローカルネットワーク上の<ホスト>
- ③ ホストアドレス=255 …ブロードキャスト
- ④ 127.0.0.0 - 127.255.255.255 …ホストのループバックアドレス (Internal host loopback address)
- ⑤ クラス D (224.0.0.0 - 239.255.255.255) …マルチキャスト

※127.xxx.xxx.xxx は全てループバックアドレスに該当しますが、自ホストを表すアドレスとしては通常、127.0.0.1 か”localhost”を使います

## 3. NAT/PAT (NAPT)

組織内で使う IP アドレスは通常プライベートアドレスになりますが、外部に公開 (ホームページやメール) しているサイトと外部のサイトにリクエストを出す (インターネットを利用する) ホストはグローバル IP アドレスを持つ必要があります。

このうち、外部のサイトにリクエストを出すホストは組織の大きさに比例して大量になるため、グローバルアドレスを割り当てるのは困難です。そこで利用者にはプライベートアドレスだけ割り当て、ルータ等のインターネットに直接接続する機器でリクエストの送信元 IP アドレスをプライベートアドレスからグローバルアドレスに (NAT: Network Address Translation)、リクエスト元ポート番号はレスポンスの戻り先が識別できるように未使用の番号と変換 (PAT: Port Address Translation) します。逆にレスポンスの宛先 IP アドレスとポート番号はリクエストのアドレスとポート番号に戻します。NAT と PAT を合わせて NAPT (Network Address Port Translation) と呼びます。この操作は内部のアドレスを外に見せないでインターネット経由で内部のネットワークに侵入されるリスクも減らします。

外部に公開している複数のサイトのグローバルアドレスを NAT で内部アドレスに変換することもできます。この場合は URL のドメイン名/パスの文字列で処理 AP を振り分けます。

# Linux ネットワークと踏み台 (ssh トンネル)

## 4. ネットワーク・セキュリティ

セキュリティ全般では、DDoS のような物量を伴う攻撃に対しては送信元の監視や対策が必要になるので、公開サイトはクラウド業者に依頼する等のリスク回避が必要です。組織内部に設置したネットワークでは、侵入（破壊／改竄）や盗聴等に対してはスイッチやルータの監視ポートやハブの空ポートが自由に使えるようになっていないか等の設備の物理的な管理と、認証サーバやストレージ等の重要設備がある部屋への入出監視やセキュリティアップデートの適用（配布直後に全ホストへ一斉に適用するのは別の危険があります）漏れや独断でソフトをインストールしていないかをチェックする等の運用の管理が重要<sup>18</sup>になります。

しかし、設備、運用上の対策だけでは安全にはなりません。「はじめに」に記述したように、ネットワークはインターネットと直接／間接に接続せざるを得ませんが、侵入手法を公開するサイトがあるうえに OS の新版に新しいセキュリティホールが見つかることが珍しくありません。これ等はソフト／システムによる対策が必要で、代表的なものとして以下のものがあります。

- ① アンチウイルスソフトや侵入検知ソフト
- ② ファイアーウォールによる侵入抑止
- ③ アプリケーションゲートウェイ<sup>19</sup>による L7 レベルの制御

## 5. 経路の暗号化 (SSL/TLS、SSH、IPSec)

ネットワークに流れるパケットは同一のネットワークに接続しているホストからは隠しようがないので、以下のような環境・ツールを使った暗号化により盗聴や改竄から守る必要があります。

### (1) SSL/TLS

SSL (Secure Sockets Layer) はエンドツーエンド間のパケット<sup>20</sup>を暗号化するプロトコルです。公開鍵暗号で作った証明書によるサーバ（ドメイン）の認証も行います。ネットスケープ社が 1990 年代にブラウザの Netscape Navigator に組み込み、その後 IETF(Internet Engineering Task Force) の下で TLS という名前に変えて脆弱性の対策が行われています。ブラウザで https を使うと、背後で URL に指定したドメインの証明書確認<sup>21</sup>と、通信データの暗号化が行われます。

---

<sup>18</sup> 「経済犯罪実態調査 2020 —グローバル翻訳版・日本分析版—PwC」によると、過去 2 年間に経済犯罪の被害報告をした企業の割合が 21%、内サイバー犯罪を報告 36%で 1 位（2016 年 6%）世界では 47%が被害を報告し、サイバー犯罪を報告したのは 34%で 2 位（2016 年 32%）

<sup>19</sup> 通信パケットの内容も調べるファイアーウォール的一种。一般的なファイアーウォールよりも前から研究されているようですが、2022 年時点で一般企業の適用事例はあまり見当たりません  
[https://en.wikipedia.org/wiki/Guard\\_\(information\\_security\)](https://en.wikipedia.org/wiki/Guard_(information_security))

<sup>20</sup> 具体的にはアプリケーション層（ブラウザ／Web サーバ）のパケットを暗号化して TCP（セッション層：L4）のデータにします…TCP ヘッダと IP(L3)以下の層は暗号化されません

<sup>21</sup> インターネット上で公開されているサイトの証明書はルート証明局に認証を受けた証明局(CA)に認証を受けています。ブラウザにインストールされているルート証明書から辿って認証が確認できないサイトはブラウザが警告を出します

# Linux ネットワークと踏み台 (ssh トンネル)

## (2) SSH

SSH (Secure SHell) は暗号化した経路を作りサーバにログインするツールです。ファイル転送コマンドの scp と sftp も同一環境 (同一ポート) で動作し、SSL/TLS と同様に公開鍵方式の認証と共通鍵方式のセッションキーで通信データを暗号化します。ssh が特に重要なのは「ポート転送」とか「ssh トンネリング」と呼ぶ機能で、他のアプリケーションのパケットを ssh のパケットとして暗号化して運ぶことができます。

## (3) IPsec

IPsec (Security Architecture for Internet Protocol) は IP パケット全体を暗号化する L3 レベルのプロトコルです。別の IP ヘッダを付けて元のパケットをデータにして送るトンネルモードと元の IP ヘッダを残して送るトランスポートモードがあります。トンネルモードの場合は各拠点 (VPN のルータ等) に、トランスポートモードの場合はエンドツーエンドのホストに認証や鍵交換、パケットの暗号化/複合化を行う装置や専用ソフトが必要です。L3 よりも上層、アプリケーションからは暗号化を意識せず利用できます。

VPN の機能を持つルータは L2TP (Layer 2 Tunneling Protocol) という L2 プロトコルと組み合わせた L2TP/IPsec で拠点間にブロードキャストドメインを構成 (ブリッジで繋いだようになる) し、ファイル共有やプリンタ共有ができる (リスクも高い) ものが普及しています。

## 6. 組織内ネットワークの分界/境界

組織の中のネットワークは用途によりセキュリティのレベルや維持の方法が異なります。

例えば、重要なデータを管理しているネットワークは侵入の防止に重点を置く必要がありますが、長時間のサービス停止も困ります。機器の二重化や負荷分散装置はハード障害には対応できますがソフト障害 (特に OS 周辺) は機器の冗長化では防ぐことができないので、セキュリティアップデートは他所で動作確認とウイルススキャンが終わった後にします。また、明らかに必要なポート以外はファイアウォールで閉じておきます。

営業で使うネットワークは外部との継続的な接続が重要なので、0 day 攻撃を想定してセキュリティアップデートやアンチウイルスは常に最新にしておきます。データを保管しているネットワークとのやり取りはアプリケーションサーバだけが行うようにします (この通信の往復で使うポートだけ開ける)。開発用のネットワークは情報収集のためにインターネットに接続が必要で、営業のネットワークとデータ管理用のネットワークとはアプリケーションサーバやデータベースサーバのメンテナンス用に個人の識別・認証を行った上で接続できるようにする等のネットワークの切り分けが必要になります。その他に、外部のクラウド環境を使っている場合や在宅勤務のリモート接続を行うのであれば接続プロトコル (プロバイダや通信会社との契約内容) やリモート側に置く装置の物理的なセキュリティも考える必要があります。

日常業務では一人の要員が各ネットワークで作業することがありますが、各ネットワーク専用にクライアント PC を用意して使い分けるのは現実的ではありません。利用者は使い辛いシステムは使わないか、使い易くするためにセキュリティの穴を利用したくなります。

# Linux ネットワークと踏み台 (ssh トンネル)

## 7. ネットワークの接続

ネットワークを分界・分離しただけでは使い勝手が悪いだけで安全でもありません。一般的なリスク対策を施した以下の環境で考えると...

<環境条件>

- ① 公開サイトはクラウドに置き、在宅勤務含むリモート接続は通信会社等の VPN を使う
- ② 用途（リスクの種類が異なる）やセキュリティレベル異なるネットワークを分離している
- ③ ネットワーク間にファイアーウォールを設置している

### 7.1. リスク

この環境でも、以下のリスクが残ります。

- ① エンドツーエンド間での盗聴
- ② ルートアカウントを含む利用者の資格情報の乗っ取り

※他に正規利用者による破壊、改竄、漏洩のリスクがありますが、利用者の権限内のアクセスはシステムでは防げません。運用による対処が必要です。

### 7.2. 対策

漏れたり改竄されてはいけない情報が存在する部分に対策を行います。比較的すぐできる対策としては以下があります<sup>22</sup>。

- ① 日常業務では ssh や https を使いエンドツーエンド間を暗号化する
- ② ファイアーウォールでセキュリティレベルの低い側からの ssh と https リクエストだけを許可
  - ・情報参照はセキュリティレベルの高いネットワーク内で加工した結果を https で参照する
  - ・作業を行う場合は ssh でログインする
- ③ システム利用者はアカウント分けを個人毎の識別・認証をする

### 7.3. 個人識別・認証の方法

エンドツーエンドの経路は暗号化を、エンドポイントでは認証でなりすましを防止します。

ssh を使う場合は、以下の認証方法があります。

- ① パスワードによる OS 認証<sup>23</sup> ...サーバに利用者登録があればログイン可能です
- ② 公開鍵／秘密鍵による認証 ...秘密鍵をクライアントに保存し公開鍵をサーバに登録します
- ③ 公開鍵とパスワードの二段階認証 ...①+②の手順で認証します

※②の場合秘密鍵はファイルとして保存されている機器からはパスワードなしでログインができるようになります。しかしこのファイルは通常のテキストファイルなので複写が可能です

---

<sup>22</sup> 情報の内容によってはより高度な手段（アプリケーションゲートウェイ、暗号化装置等）をとったり、逆にコスト（使い勝手含む）と効果を比べたうえで「何もしない」ことも選択肢に入ります

<sup>23</sup> 厳密にはサーバ OS と同一の機構=PAM (Pluggable Authentication Modules) による認証

# Linux ネットワークと踏み台 (ssh トンネル)

## 7.4. 公開鍵／秘密鍵の作り方

公開鍵／秘密鍵はペアになっていて、相互検証ができるようになっています。公開鍵は認証局 (CA) ソフトで電子署名を付けて「証明書」にすることもできます。サーバ用に構成した Linux には通常 オープンソース (BSD ライセンス) の OpenSSH がインストールされています。

Windows 10 以降は OpenSSH がインストールされている (%SystemRoot%\System32\OpenSSH) ので、Linux (/usr/bin/ssh-keygen) と同様の方法で鍵ペアが作れます。(以下、コマンドパスは略)

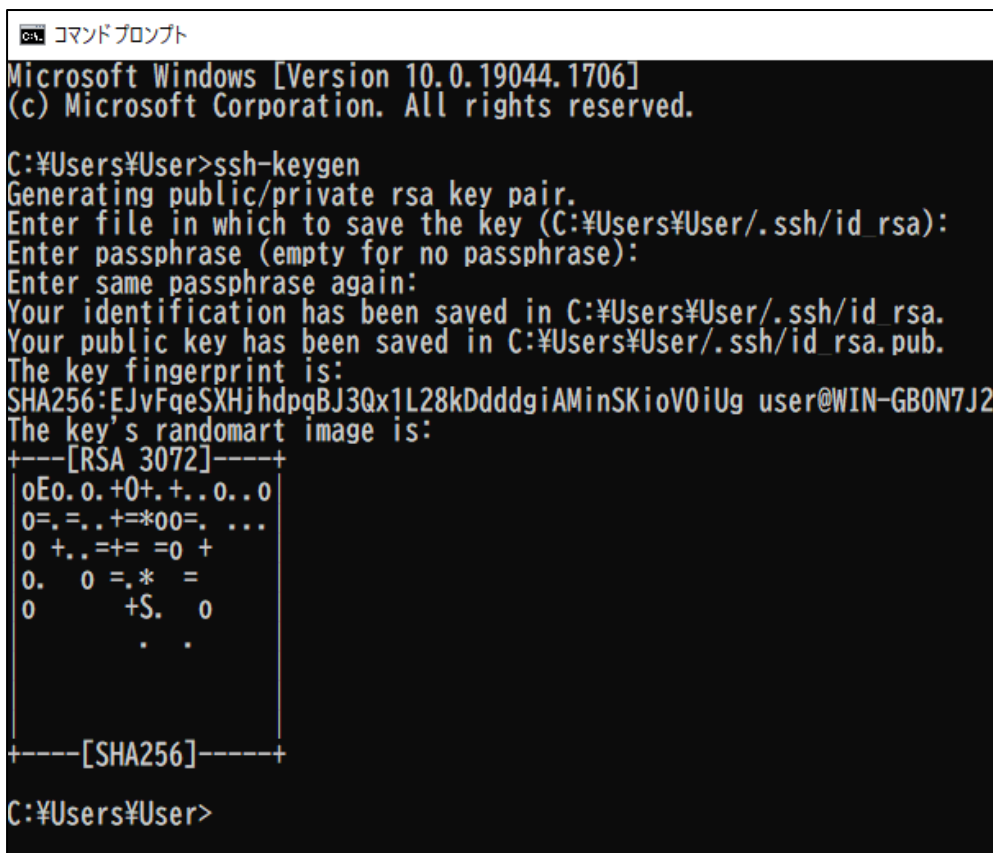
コマンド：ssh-keygen

<Windows 10 のコマンドプロンプトからの実行例>

デフォルト (パラメータなし) では以下のように実行されます。

- ・暗号化方式：rsa
- ・鍵の出力先：実行した利用者のホームディレクトリ配下 **.ssh** フォルダに以下が出力されます  
公開鍵：**id\_rsa.pub**  
秘密鍵：**id\_rsa**

※ Enter passphrase (empty for no passphrase): には秘密鍵を暗号化するフレーズ (キー) を指定します。フレーズを指定しなかった場合、このテキストファイルが盗まれたときノーガードになります。Windows をクライアントに使う場合は比較的簡単に接近できるので必ず指定すべきです



```
Microsoft Windows [Version 10.0.19044.1706]
(c) Microsoft Corporation. All rights reserved.

C:\Users\User>ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\User\.ssh\id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:\Users\User\.ssh\id_rsa.
Your public key has been saved in C:\Users\User\.ssh\id_rsa.pub.
The key fingerprint is:
SHA256:EJvFqeSXHjhdpgBJ3Qx1L28kDdddgiAMinSKioV0iUg user@WIN-GBON7J2
The key's randomart image is:
+----[RSA 3072]-----+
|oEo.o.+0+.+.o.o.o|
|0=.+.+=*00=. ...|
|0+.+=+=0+|
|o. 0 =.* =|
|0      +S. 0|
|          . .|
+-----[SHA256]-----+

C:\Users\User>
```

## Linux ネットワークと踏み台 (ssh トンネル)

.ssh フォルダにできた **id\_rsa.pub** (暗号化方式が rsa のとき) を ssh の接続先のサーバに送り、自分のアカウントの承認済キーのファイルに追加します。

・承認済キーのファイル: `~/.ssh/authorized_keys`

このファイルに既に別の公開鍵が書かれていたら後ろにつなげます。エディターで貼り付けても問題ありませんが、Linux には一連の作業を行うコマンドがあります。

<Linux の追加コマンド>

`ssh-copy-id 利用者名@サーバ`

<Windows の場合>

Windows 用のコマンドはないため、以下のようにしてファイルに追記することができます。

```
> type %HOMEPATH%\%.ssh\id_rsa.pub | ssh 利用者名@サーバ "cat >> ~/.ssh/authorized_keys"
利用者名@サーバ's password: ...利用者名のサーバアカウントのパスワードを入力
```

※実行している内容は、公開鍵ファイルの内容を ssh コマンドにパイプで渡し、サーバ側の cat コマンドでホームディレクトリの **.ssh/authorized\_keys** ファイル (なければ新規作成) の最後に追加

<公開鍵登録後の ssh 接続>

```
C:\Users\User>ssh adminuser@ubuntu2004
```

```
Enter passphrase for key 'C:\Users\User/.ssh/id_rsa': ...秘密鍵の暗号化フレーズを入力します
```

```
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.4.0-109-generic x86_64)
```

※以上で公開鍵と秘密鍵による認証方法で ssh 接続ができるようになりました

### 7.5. 踏み台サーバ (ssh ポートフォワーディング/トンネリング)

ssh で接続 (デフォルトのポート番号 22) したホスト間の経路は、ssh 以外 (22 以外のポート) のプロトコルを乗せることができ、これを ssh ポートフォワーディング、ssh トンネリングと呼びます。この機能により ssh でログインしたサーバを踏み台にすれば、直接はファイアウォールで遮られている全ての機器の接続待ちポートを使うことができるようになります。

このトンネルは、ssh クライアントからの以下のコマンドで開設できます。

```
ssh -L <ローカルポート>:<目的のサーバ>:<目的のポート> <利用者名>@<踏み台サーバ> -N
```

#### ●オプション/パラメータの意味

ローカルポート : ssh コマンドを実行したホストのポート

…このポートが *目的のポート* に繋がります

目的のサーバ : 使いたいアプリケーションが動作しているサーバの名前または IP アドレス

目的のポート : 使いたいアプリケーションが接続待ちしているポート

利用者名@踏み台サーバ : ssh で接続する (sshd が動作している) サーバ

…このサーバから目的のサーバ : ポートに対してリクエストが出されます

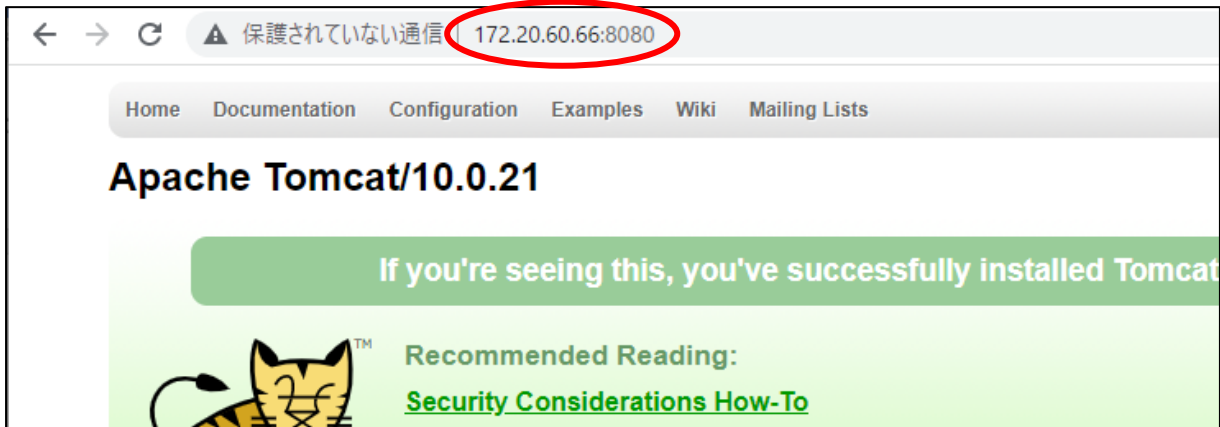
-N : このオプションを付けると ctrl + C で ssh を打ち切るまでターミナルが後続のコマンドを受け付けなくなります。付けないと、他のコマンドを受け付けます



# Linux ネットワークと踏み台 (ssh トンネル)

ssh -L の実行例

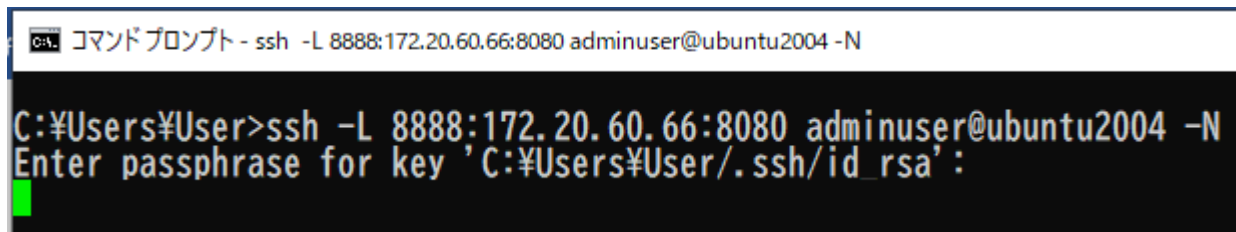
別のネットワーク (172.20.60.66/20) で、以下のような Web 画面を出すアプリケーションが稼働しているとき



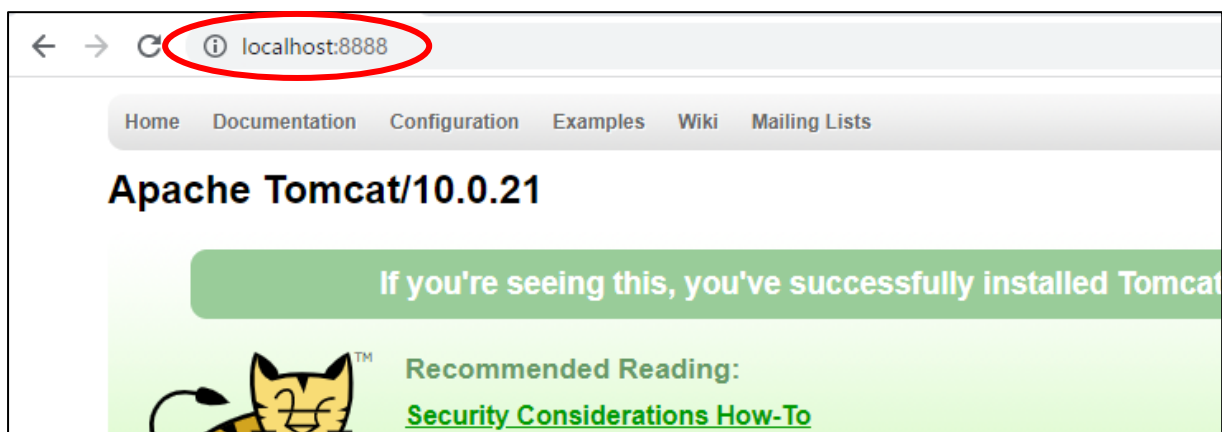
以下のコマンドで ubuntu2004 と同一のネットワークにある 172.20.60.66 の Web サーバに接続します。

```
C:¥Users¥User>ssh -L 8888:172.20.60.66:8080 adminuser@ubuntu2004 -N
```

秘密鍵のパスフレーズに答えると、以下のようにコマンドプロンプトは待機状態になります。



このコマンドが ctrl+C 押下で打ち切られるまで、ローカルホストの 8888 ポートで 172.20.60.66 のポート 8080 に接続することができます。



※このようにネットワーク内のサーバにログインできるアカウントがあれば、ファイアウォールで遮断されている IP アドレスやポートでも利用することができます。

アカウントの管理に注意してください

以上