.Deb

銀河系唯一のDebian専門誌

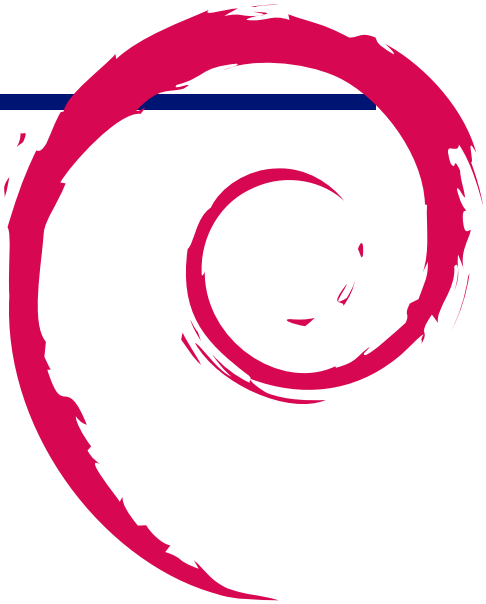
2013 年 8 月 17 日

特集1：ＯｐｅｎＶＰＮを使ってみた

特集2：デビアン勉強会資料のｅＰＵＢ化

デビアン勉強会

目次



1 事前課題

上川 純一

今回の事前課題は以下です:

1. 自分では vpn をどう利用していますか?

この課題に対して提出いただいた内容は以下です。

1.1 Yoshida Shin

VPN を構築した事は有りません。また、プライベートでも使用した事は有りません。

VPN は仕事で以下の 2 個の使い方をしています。

1. データセンターのサーバーにログインする
2. 緊急対応のため、自宅から職場につなぐ

でも、緊急対応は行わないで済むようにするべきだし、(緊急対応が無いと仮定すれば、) 多くの場合はデータセンターへの接続も職場からの IP 制限だけで十分だと考えています。

VPN は万一の為に用意するものであり、あまり積極的に使いたいと思わないです。

1.2 吉野 (yy_y_ja_jp)

個人的には使ってません.

1.3 sakai

今のところ VPN は利用していない。そのうち、外出した時用に VPN で自宅とつなごうかなー、ということを考える程度。

1.4 野島 貴英

以前、苦肉の策で 2 つの拠点間をインターネット経由で openvpn 使って一時的に LAN を組み、複数の WEB サイトの WEB-DB 間のアクセスを遠方の DB に常時流し込みつづけてそのまま半月ぐらいサービス維持した事があります。使った結果ですが、予想外にも、簡単／大変タフ／非常に安定した VPN 経路が組めた記憶があります。ただ、ちゃんとしたデータ取ってないので、感想以上の事がいえない状況ではあります。

1.5 dictoss(杉本 典充)

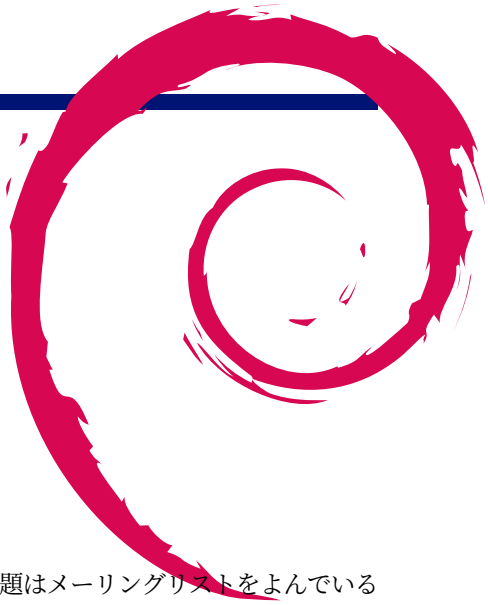
最近自宅サーバに openvpn を入れて、出先でテザリングをしながらサーバにアクセスしている。それまでは ssh とポートフォワードでがんばっていたが、サーバの台数が増えるとポートフォワードの数が増えるので設定が疲れました。 VPS から自宅サーバに VPN セッションを張ろうかと思っているが自宅サーバは DynamicDNS を利用しているので切れずにつながっていられるか不明。

会社だと拠点間をつなぐために L2TP/IPsec で VPN を張る場合や、グロバール IP を持たない (=NAT 配下) に設置されるが管理上サーバから ssh する必要がある PC にはクライアントから VPN を張らせることで ssh できるようにする、といった使い方をしている。

1.6 まえだこうへい

会社環境に繋ぐのに、 OpenVPN と、 Cisco Anycoonect を使ってますが、後者の環境では、 Anyconnect クライアントではなく、 OpenConnect を使ってます。 RSA OneTimePassword のモジュールを併用して二要素認証にしていますが、これにも対応してます。

個人環境では SSH の Proxy 機能を使っているので、接続先ノード数増えても特に困らず VPN って面倒だよなぁとふと思い出しながら仕事してます。



2 Debian Trivia Quiz

上川純一

ところで、みなさん Debian 関連の話題においついていますか? Debian 関連の話題はメーリングリストをよんでいると追跡できます。ただよんでいるだけではりあいがないので、理解度のテストをします。特に一人だけでは意味がわからないところもあるかも知れません。みんなで一緒に読んでみましょう。

今回の出題範囲は `debian-devel-announce@lists.debian.org` や `debian-devel@lists.debian.org` に投稿された内容と Debian Project News からです。

問題 1. Sylvestre Ledru が JDK についてアナウンスしたのは

☐ A OpenJDK7 にきりかえ
☐ B JDK6 の削除
☐ C JDK8 への移行

問題 2. OpenJDK 7でサポートされていないアーキテクチャはどれか

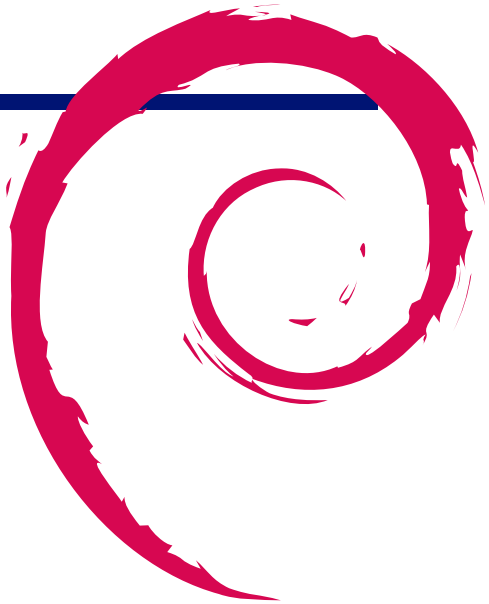
☐ A mipsel
☐ B amd64
☐ C i386

問題 3. Summer Of Code のコーディネーションメンバーでないのは誰か

☐ A David Bremner
☐ B Nicolas Dandrimont
☐ C Nobuhiro Iwamatsu

問題 4. Brian Gupta が Debian のトレードマークとして USPTO に追加登録しようと提案したのは何か

☐ A ロゴ
☐ B Debian の文字列
☐ C DD



3 最近の Debian 関連のミーティング報告

上川純一

3.1 東京エリア Debian 勉強会 102 回目報告

東京エリア Debian 勉強会１０２回めは杉並区荻窪で開催されました。

岩松さんが Linux Kernel の armmp フレーバーについて紹介しました。デバイスツリー情報を利用することで複数のボードをサポートする仕組みのようです。

野島さんが dh_strip について紹介しました。デバッグ情報について深いお話でした。

上川が raspberry pi を使ってみたのでその紹介をしました。

4 OpenVPN を使ってみた

上川純一

4.1 はじめに

VPN（ 仮想プライベートネットワーク）とはインターネット上にある (WAN) に閉域ネットワーク (LAN) を仮想的に構築する方法です。

OpenVPN は認証と接続方式の部分で TLS[5] [*1]を活用しています。 TLS は HTTPS などで利用されている仕組みで、公開鍵認証をつかって最初の鍵交換をおこない、そこで交換した対象鍵をつかって実際の通信をすることで効率良く暗号通信ができるようになっています。

4.2 今回実現したいこと

今回想定する例としてとある個人の開発環境ネットワーク構成図を見てみましょう（ 図 1）。

自宅にはごく一般的なプロバイダ契約でネットワークをひいていて、ラップトップと携帯電話はどういうネットワークを経由してどういう IP アドレスでつながるのかよくわからないという構成です。

携帯電話とラップトップから自宅にあるサーバ Raspberry Pi にアクセスできるようにしたい、そう思った時にプライベート IP で接続しているプロバイダー接続がネックになります。

開発や実験などに便利な環境を実現したい、それは Raspberry pi とラップトップと携帯電話が同じ LAN にいるような環境です。

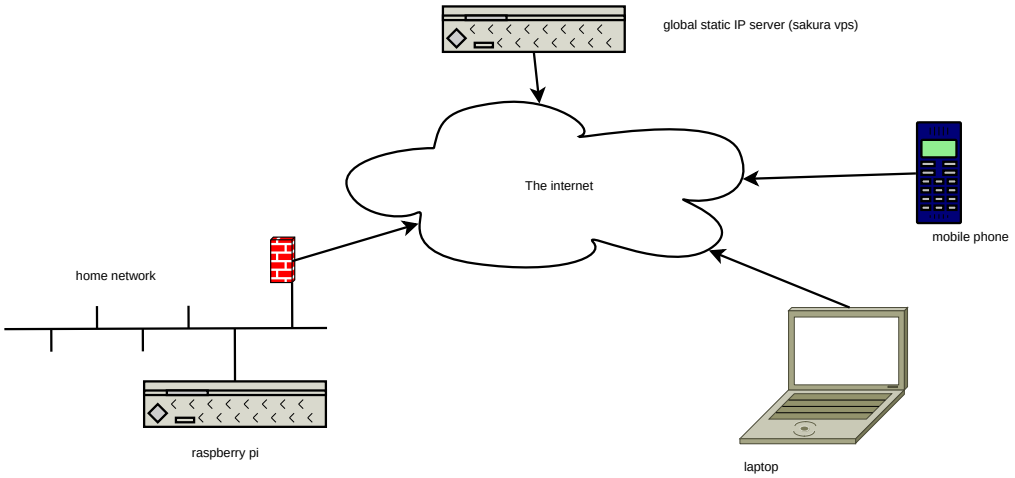


図 1 とある個人の開発環境ネットワーク構成図

[*1] 規格が TLS になるまえは SSL という規格でした

4.3 OpenSSL, CA の仕組み

OpenVPN で利用できる認証のしくみは複数あり、一番簡単な方法は共有鍵方式だと思われます[*2]。ここでは自分の直接の管理下にあるマシンを複数台管理する場合に便利そうな方式、自前で CA を立てて PKI(x509) で構築する方法について紹介します。

今回紹介する方式を実現する副作用として、 openvpn に脆弱性があったり、認証情報がもれると外部のユーザが仮想のプライベートネットワークに接続できるようになります。また、ネットワークにはユーザ認証がないので、 VPN 接続しているホストにログインできればプライベートネットワークにアクセスできるようになります。

4.4 インストール

Debian では openvpn パッケージとして提供されています。また、 CA 関連は openssl に依存しているので openssl パッケージもインストールしましょう。

```
# apt-get install openvpn openssl
```

/etc/openvpn/*.conf に設定ファイルを配置します。 Debian での openvpn の設定は、/etc/openvpn ディレクトリにある conf という拡張子になっているファイルを設定ファイルとして認識して起動時に利用するようです。デフォルトの状態では、最初は何も設定ファイルがないので何も起動していません。

クライアント側も openvpn パッケージをインストールします。

4.5 OpenVPN の接続性試験

まず、サーバとクライアント間で接続できるかどうかを確認する方法について紹介します。サーバとクライアントで openvpn を起動して相互に接続し、 openvpn が接続できる状態であることを確認します。 ping とかうってみるのがよいでしょう。

openVPN は UDP もしくは TCP 接続を利用することができます。どちらかはつながるでしょう。

```
server$ sudo openvpn --dev tun1 --ifconfig 10.1.1.1 10.1.1.2
client$ sudo openvpn --dev tun1 --remote サーバホスト名 --ifconfig 10.1.1.2 10.1.1.1

client$ ifconfig
tun1      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:10.1.1.2  P-t-P:10.1.1.1  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:2 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:168 (168.0 B)  TX bytes:168 (168.0 B)
client$ ping 10.1.1.1
```

4.6 CA の作成

まず最初に認証局を作成します。とりあえずは easy-rsa を使うとよいでしょう。 wheezy の openvpn パッケージでは /usr/share/doc/openvpn/examples/easy-rsa/2.0/ にファイルがおいてありそれをコピーしてつかえということのようです。

[*2] 認証なし暗号化なしという方法もあってそっちのほうが簡単ですが openvpn の接続性テスト以外ではあまりつかわないと思います

```
 # cd /etc/openvpn
 # sudo cp -R /usr/share/doc/openvpn/examples/easy-rsa/2.0/ easy-rsa/
 # cd easy-rsa/
 # vi vars

export KEY_COUNTRY="JP"
export KEY_PROVINCE="TOKYO"
export KEY_CITY="Suginami-ku"
export KEY_ORG="uekawa"
export KEY_EMAIL="dancerj@gmail.com"
#export KEY_CN=changeme
#export KEY_NAME=changeme
#export KEY_OU=changeme
#export PKCS11_MODULE_PATH=changeme
#export PKCS11_PIN=1234

 # . ./vars
 # ./clean-all
 # ./build-ca
```

いろいろと質問されますが、あまり重要な質問はない気がします。 “Common Name” が名前で、ここで聞かれている名前は CA の名前です。が、この名前が重要な場面はあまりない気がします。

注意事項としては何も設定しないと全員ファイルが読めるようになっているのですが、認証情報関連のファイルは秘密であることが重要なので、 root のみが読めるようにしましょう。

4.7 各種証明書の作成

サーバはクライアントが信頼できる CA に署名された証明書を使っていることを確認することになります。クライアントはサーバの証明書が信頼できる CA にに署名されていることを確認することになります。

正式な手順はクライアント・サーバ各ノードで Certificate Signing Request (CSR) を作って CA 側で署名して CRT ファイルを返してあげるということになります。今回は略式でサーバを CA として運用し発行機関としても併用します。OpenVPN サーバが乗っ取られたら CA としても乗っ取られるのですが OpenVPN サーバが乗っ取られた時点で全ノードの再設定が必要になると思うので、新しく自己署名 CA を作りなおしてしまうという方針でいきます。

まずサーバの証明書を作成します。今回 sakura という名前のサーバの証明書を作ってみましょう。

```
 # ./build-key-server sakura
```

CN(common name) のところにはホスト名をいれる慣習になっているようで、その値は後々ログなどで表示されます。A challenge password と an optional company name は CSR に追加ではいる情報みたいですが、 CSR を読むことはないのでこの場合は何も入力しなくてもよいようです。証明書を作成して証明書の署名と登録を連続しておこなうのでちょっとわかりにくいです。

クライアントの証明書を作成します。

```
 # ./build-key client1
 # ./build-key nexus4
```

4.8 HMAC 鍵の作成

TLS をのハンドシェークは CPU 負荷の高い処理なので DoS される可能性もあります。そこで、 HMAC 鍵を使ってその前段でフィルタリングをかけることができるようです。ついでにつくってしまいましょう。

```
 # openvpn --genkey --secret ta.key
```

4.9 OpenVPN のサーバ設定

TLS 接続用にサーバ側で必要となる Diffie Hellman パラメータ(なにそれ?) を生成します。

```
 # ./build-dh
```

/etc/openvpn/server.conf に設定ファイルを作成します。

```
port 1194
proto udp
dev tun

user nobody
group nogroup

tls-auth      /etc/openvpn/easy-rsa/keys/ta.key 0 # server is 0.
ca      /etc/openvpn/easy-rsa/keys/ca.crt
cert    /etc/openvpn/easy-rsa/keys/sakura.crt
key     /etc/openvpn/easy-rsa/keys/sakura.key  # keep secret
dh      /etc/openvpn/easy-rsa/keys/dh1024.pem

server 10.55.2.0 255.255.255.0  # internal tun0 connection IP
ifconfig-pool-persist ipp.txt

keepalive 10 120

comp-lzo         # Compression - must be turned on at both end
persist-key
persist-tun

status log/openvpn-status.log

verb 3  # verbose mode
client-to-client
```

この設定ファイルは log/ディレクトリにステータスログを出力する設定なので、ディレクトリ /etc/openvpn/log/ をつくっておきます。

/etc/openvpn/ipp.txt に IP アドレスの割り当て設定が記録されます。デフォルトでは 60 秒に一回更新されるようになっています。

keepalive 10 120 の設定では、サーバとクライアントはお互いに 10 秒に一回 ping で生死監視をして、 2 分間接続がなかったら再起動します。

/etc/openvpn/*.conf にファイルがあると起動時にサーバを起動してくれるようになるのでこれで設定は完了です。

4.9.1 トポロジー

デフォルトの TUN デバイスのトポロジーは net30 です。これはクライアント毎にサブネットマスク /30 の ipv4 アドレス(4 個づつ) を割り当てることになります。ブリッジモードで TAP を使う、もしくは p2p・ subnet トポロジを使うとクライアントあたり一つの IP アドレスになるみたいですが、どうせクラス A のプライベートアドレスを大量に使えるので特に必要もないので検証してません。

4.9.2 tun or tap

仮想ネットワークデバイスは TUN と TAP の二択あります。 TAP はイーサネットレベル(L2) の仮想デバイスで、TUN は IP レベル(L3) の仮想デバイスです。今回は iOS と Android の openvpn クライアントが TUN しかサポートしていないので TUN を使うことになります。

	ネットワークレイヤー	機能	つかえる OS
tap	L2	ブリッジ(ブロードキャストパケットが到達する)	linux
tun	L3	ルータ	linux, iOS, Android など

4.10 OpenVPN のクライアント設定

CA からいくつかのファイルをコピーしてくる必要があります。 Debian の場合はサーバと大体ディレクトリ構成をあわせておけばよいでしょう。 root でしか読み込めないように権限を設定しておくのをお忘れなく。

ca.crt, client.crt, client.key, ta.key が必要になります。

Android の OpenVPN アプリケーションの場合 [6] は、 /sdcard 以下に適当な名前でディレクトリを作成してそこに必要なファイルを配置します。設定ファイルは拡張子を conf ではなく ovpn にしておかないと認識してくれないようです。一回 OpenVPN アプリケーションで Import したらその SD カード上のディレクトリの中身は必要なくなるので消しましょう。

```
client
dev tun
port 1194
proto udp

remote xyz.sakura.ne.jp 1194             # vpn server ip : port
nobind

tls-auth      ta.key 1 # client is 1.
ca ca.crt
cert android-nexus4.crt
key android-nexus4.key

remote-cert-tls server
comp-lzo
persist-key
persist-tun

verb 3
```

remote-cert-tls server で証明書がサーバー鍵であることを確認します。./build-key-server で作られた証明書は “key usage” に サーバ証明書であることが記述されています。これを指定しないと有効な CA をに署名された証明書であればどれでもつなぎに行きます。たとえば流出したクライアントの証明書でサーバを偽装することが可能になります。

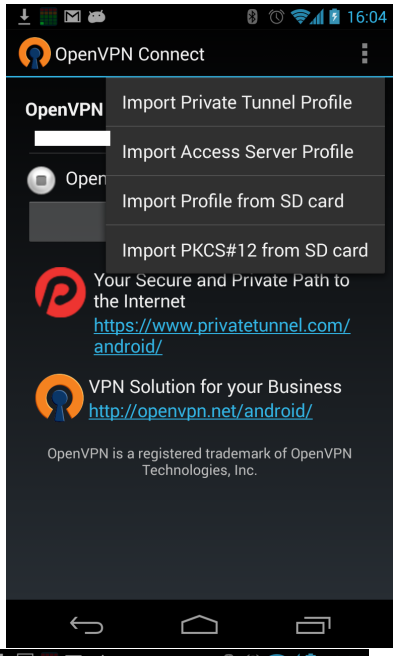
起動方法は、 `/etc/network/interfaces` に記述することもできますが [3] デフォルトの状態でネットワークの設定が変更になったら勝手に OpenVPN 接続を試行してくれる設定になっているようです。[*3]

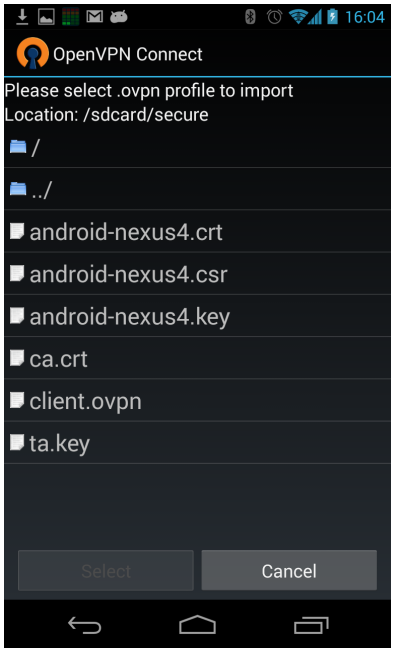
4.10.1 Android UI

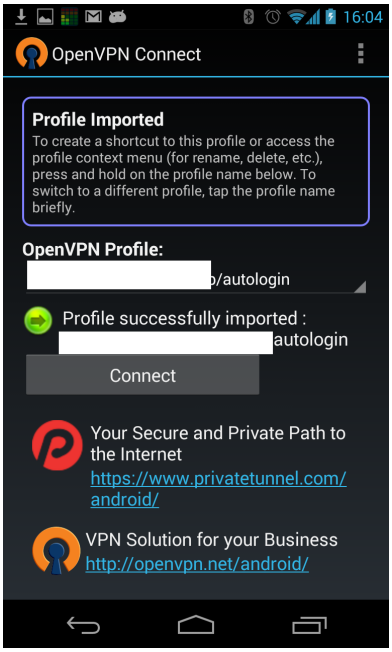
Android デバイスに必要なファイルをコピーします。

```
$ ls -1
android-nexus4.crt
android-nexus4.csr
android-nexus4.key
ca.crt
client.ovpn
ta.key
$ adb push . /sdcard/secure
push: ./ca.crt -> /sdcard/secure/ca.crt
push: ./android-nexus4.key -> /sdcard/secure/android-nexus4.key
push: ./ta.key -> /sdcard/secure/ta.key
push: ./android-nexus4.crt -> /sdcard/secure/android-nexus4.crt
push: ./client.ovpn -> /sdcard/secure/client.ovpn
push: ./android-nexus4.csr -> /sdcard/secure/android-nexus4.csr
```

[*3] 多分 `/etc/network/if-up.d/openvpn` の設定による







インポートしたら sdcard からファイルを削除します。

4.11 応用

特に今回必要でなかったので設定しなかったのですが違う設定も可能なのでそれも紹介します。

ルーティングなどは openvpn サーバ側で設定すればクライアントに設定を配布(push) してくれるようです。

4.11.1 ローカルネットワークへのゲートウェイを設定したい

redirect-private オプションで VPN 経由でのルーティングを設定します。たとえば、 VPN 経由のゲートウェイ(たとえば 10.0.0.1) 経由で 自宅のネットワーク(例えば 192.168.1.0/24) にアクセスできるようにしたいときに使えます。

`/usr/share/doc/openvpn/examples/sample-config-files/firewall.sh` に iptables の設定例が掲載されています。

4.11.2 暗号化経路のためにすべてのパケットを VPN 経由にしたい

redirect-gateway オプションで指定できます。

公共の WiFi スポット経由での通信などが安心できない人向けのオプションです。しかし DHCP の再リースや DNS の名前解決なども VPN 経由になってしまうと困る場合もあるので bypass-dhcp, bypass-dns オプションなどがあるようです。

4.12 おわりに

openvpn 2.3 で git リポジトリの再編を行ったようで、 easy-rsa とかが分離されたっぽいです。

マニュアルを読んでいると Windows 版の専用オプションとかが複雑にからみあっていて実際どれをつかえばよいのか読み解きにくいです。最低限動かすまではそこまで難しくないのですがルーティングとかデフォルトゲートウェイを変更しようかと考え始めると考えることがいくつかあるようです。しかし一般論で考えても、ネットワーク 3 つ以上を接続する場合のファイアウォールとかルーターの設定はそれなりに複雑ですから、それを考えると妥当なのかもしれませんね。

4.12.1 ssh の各機能との比較

開発者・運用者愛用のツール ssh とどう違うのかとおもってみてみたのですが、 ssh にも VPN 機能もあるので比較しにくいですね。ただ、通常の運用では root 同士で ssh することはあまりないかと思いますが、 VPN を利用するには root で ssh できる必要があるようです。

openvpn のメリットとしては openvpn には接続が切れた時を検出して再接続する仕組みがあること、 TCP だけでなく UDP も使えること、 root 権限で ssh ログインを許可しないでよいことなどがあげられると思います。

表 1　ssh の機能の対応

	ssh オプション	機能
ポートフォワーディング	-L, -R	特定のローカルの TCP ポート番号をリモートのポート番号とマッピングする
プロキシ	-D	SOCKS に対応しているアプリケーションのプロキシを提供
VPN	-w	IP レベルで相互に見えるよう仮想的にネットワークを構築する

参考文献

[1] `http://wiki.debian.org/OpenVPN`

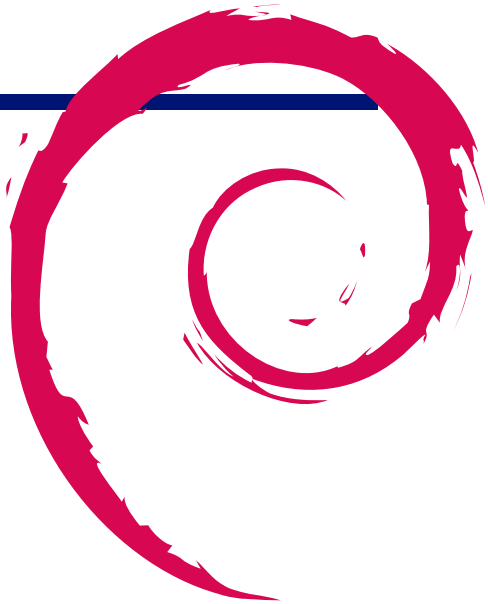
[2] openvpn(8) manpage

[3] OpenVPN README.Debian `/usr/share/doc/openvpn/README.Debian.gz`

[4] OpenVPN Community Software `http://openvpn.net/index.php/open-source.html`

[5] The Transport Layer Security (TLS) Protocol Version 1.2 `https://tools.ietf.org/html/rfc5246`

[6] OpenVPN Connect Android app `https://play.google.com/store/apps/details?id=net.openvpn.openvpn`



5 Debian 勉強会の資料の ePUB 化を試みた

まえだこうへい

5.1 ePUB 化の動機

最近はかなりスマートフォンやタブレットが普及し、それ以外の eBook reader も東京周辺の電車内でも見かけるようになりました。 Debian 勉強会の参加者もそれらのデバイスを持っている人も今までちらほら見かけています。 Debian 勉強会の事前配布資料や年 2 回の「 あんどきゅめんてっどでびあん[*4]」をそれらのデバイスで読む機会も増えているのではないでしょうか。

ところで、 Debian 勉強会の資料は LaTeX をソースとして PDF として配布されています。 PDF は印刷物と同じレイアウトになるので、本勉強会のように印刷した資料を頒布する上では最適です。しかし、画面で閲覧するのには向いていないと思いませんか。その一番の理由は多くの PDF リーダーは、画面に最適な大きさで自動的にリサイズされないためだと考えています。[*5]読みたい部分を拡大表示することはできますが、画面サイズに合わせた文字が巻き返しはされず、画面外にでた続きの文章は横方向にスクロールしなくてはなりません。これは面倒です。

そこで ePUB です。 O'Reilly, 達人出版会から出版されている Ebook の多くは ePUB が採用されているので、これらを既に購入、読んでいる人もいるでしょう。ページあたりの構成が画面サイズに最適化され、かつ、フォントサイズの変更でもページが自動的にリサイズされるので、これらのデバイスで読むのにはうってつけでしょう。

今回は Debian 勉強会の資料を、 LaTeX のソースコードには極力手を加えずに[*6]ePUB 生成できないか検証しました。

5.2 LaTeX からの ePUB 生成の方法

LaTeX から ePUB を生成するにはいくかの方法があります。取りうる手段としては 4 パターンです。

1) LaTeX から直接 ePUB を生成する
2) a. TeX もしくは b. DVI から, XML もしくは HTML を生成し、 2)-2. ePUB に変換する
3) PDF から ePUB に変換する

[*4] 現在名前は違いますが、これが一番通りは良いと思うのでこのまま表記します。
[*5] ここでは印刷物に比べて読みにくい、という観点では論じません。
[*6] PDF は今まで通り印刷物用に必要なので。

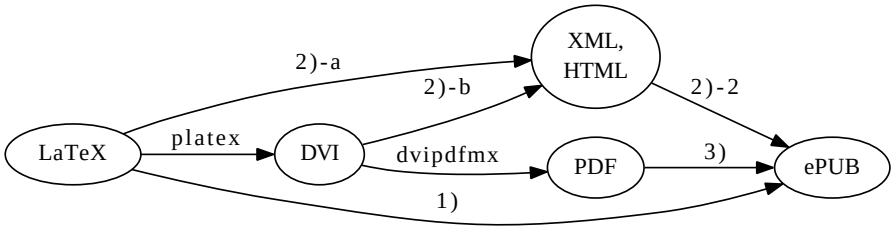


図 2　LaTeX から ePUB への変換

これらを行うためのツールについて検証しました。

5.3　検証結果

基本的に Debian パッケージになっているツールで検証を行いました。対象は次の 2 つのファイルです。

1. Debian 勉強会の資料 (debianmeetingresume2013007.tex)
2. latex2epub のサンプル TEX ファイル (sample.tex)

その結果は下記の通りです。

パターン	ツール名	入力	出力	結果	
				Debian 勉強会の資料	latex2pub のサンプル
1)	Pandoc	LATEX	ePUB	NG	OK
1)	latex2epub	LATEX	ePUB	NG	OK
2)-a	LATEXML	LATEX	XML	NG	OK
2)-b	TEX4ht	DVI	HTML	NG	NG
2)-b	htplatex(TEX4ht)	TEX	HTML	OK	NG
3)	Pandoc	HTML	ePUB	OK	N/A
4)	Calibre	PDF	ePUB	OK	OK

Debian 勉強会の資料と、 latex2epup のサンプル (以後、サンプルファイル) とで結果に大きな差がありますが、エラーメッセージを見る限りでは、その主な原因は使用している documentclass の違いと、マクロの有無によるものではないかと考えられます。が、ここはあまり深堀できていません。

5.4　変換できたケースでの課題

Debian 勉強会の資料で変換できたケースでの課題を挙げます。

5.4.1　htplatex & pandoc

現状では次の問題がありますが、今回検証した中では一番まともに読める形で変換することができました。

- tabular が table に変換されず、表にならない
- 表紙の画像が追加されない
- あんどきゅめんてっどでびあん夏号、冬号をそれが含まれる月よりも先に変換すると、その中で使われている画像が html ディレクトリにコピーされず、ファイルが無いため pandoc 実行時に失敗する
- TEX4ht で HTML 変換時に自動生成される画像のファイル名が異なり、同じく pandoc 実行時に失敗する月もある (2013 年 4 月など)
- PDF を生成する場合 (= make を実行する場合)、 htplatex が依存するパッケージ dvi2ps-fontdata-a2n をアンインストールする必要がある。これをアンインストールしないと、 dvipdfmx での DVI から PDF 変換時に、”`** ERROR ** Virtual fonts nested too deeply`”というエラーが出て失敗する

なお、 TEX4ht は、実は上川さんが既に通った道でした。[*7] Debian 勉強会の資料のリポジトリの中に、 htplatex というシェルスクリプトがあります。このスクリプトは Debian 勉強会の資料の LATEX のファイルを TEX4ht を使って HTML 化するものです。

このスクリプトを使った手順は次のとおりです。

このスクリプトの２行目に使い方、３行目に依存パッケージのインストールについて記載されています。まず、必要なパッケージをインストールします。

```
$ apt-get install dvi2ps-fontdata-a2n dvi2dvi dvipng
```

次にスクリプトを実行します。

```
$ ./htplatex debianmeetingresume200708.tex jp,2,sections+
```

すると、./html/ディレクトリ下に LATEX から変換された HTML が生成されます。これを pandoc を使って ePUB に変更します。

```
$ cd html
$ pandoc -o debianmeetingresume200708.epub debianmeetingresume200708*.html
```

なお、このスクリプトは実行時に、”-e”オプションをつけることで、 ePUB を生成することができるようにしました。

```
$ ./htplatex -e debianmeetingresume200708.tex jp,2,sections+
```

変更内容は下記リンク先をご参照下さい。

http://goo.gl/2KshN0

5.4.2 calibre

calibre では次の課題があります。

- 目次のレイアウトが崩れる
- デフォルトでは行間が広すぎる
- 図が表示されない場合もある
- tabular が表として表示されない

変換時に、”ヒューリスティック処理を有効にする”にチェックを入れ、”外観”の”段落の間の間隔を削除する”にチェックを入れると、一部は多少改善されます。

5.5 結論

現時点では、 LATEX からの ePUB 生成は、不完全ながらも一応可能なようです。どれを選択するかはユーザ次第ではありますが、いずれにしても更に使えるようにするには、各ツールでのパターンマッチングの機能を充実させる必要があるのではないかと思います。

しかし、 iPad mini くらいの大きさのデバイスでは、 Debian 勉強会の資料の PDF 版でもギリギリ拡大させなくても読めてしまいます。ですので、 PDF をそのまま読むのが一番読みやすい、という状況を解消できるようにしていくのが、我々の今後のアクションとなるに違いないでしょう。

参考文献

[1] What ’ s the best TEX-to-HTML or TEX-to-ePUB converter? http://boolesrings.org/krautzberger/2013/01/05/whats-the-best-tex-to-html-or-tex-to-epub-converter/

[*7] http://lists.debian.or.jp/debian-users/200708/msg00110.html

[2] Tools for Converting LaTeXto XML http://jblevins.org/log/xml-tools

[3] Pandoc http://johnmacfarlane.net/pandoc/README.html

[4] LXir http://www.lxir-latex.org/

[5] Hermes http://hermes.roua.org/

[6] TeXWiki http://oku.edu.mie-u.ac.jp/~okumura/texwiki/

[7] LaTeX をウエッブに載せよう http://osksn2.hep.sci.osaka-u.ac.jp/~naga/miscellaneous/tex4ht/tex4ht-howto.html

[8] LaTeX2EPUB http://kmuto.jp/d/index.cgi/computer/latex2epub.htm

Debian 勉強会資料

2013 年 8 月 17 日　　初版第 1 刷発行

東京エリア Debian 勉強会（ 編集・ 印刷・ 発行）