

LPIC レベル3 303 Security 技術解説無料セミナー

株式会社びぎねっと

宮原 徹

tmiyahar@Begi.net

本日のゴール

1. LPIC Level3 303 Securityは出題範囲に具体的なツールの使い方が多く含まれている
2. 1つ1つがなかなかヘビー
3. 実践でも役立つ内容をデモを交えて解説
4. 暗号化や認証局など基本をしっかりと理解
5. 残りの出題範囲もしっかり押さえて、303試験合格を目指す



本日のアジェンダ

1. 出題範囲の確認と学習対策
2. ポイント解説(デモ付き)
 - I. OpenSSL
 - II. OpenVPN



303試験の出題範囲(v1.0)

主題 320:暗号化

320.1 OpenSSL	4
320.2 高度な GPG	4
320.3 暗号化ファイルシステム	3

主題 321:アクセス制御

321.1 ホストベースのアクセス制御	2
321.2 拡張属性とACL	5
321.3 SELinux	6
321.4 その他の強制アクセス制御システム	2

主題 322:アプリケーションセキュリティ

322.1 BIND/DNS	2
322.2 メールサービス	2
322.3 Apache/HTTP/HTTPS	2
322.4 FTP	1
322.5 OpenSSH	3
322.6 NFSv4	1
322.7 syslog	1

主題 323:操作のセキュリティ

323.1 ホスト構成管理	2
---------------	---

主題 324:ネットワークセキュリティ

324.1 侵入検出	4
324.2 ネットワークセキュリティスキャン機能	5
324.3 ネットワークの監視	3
324.4 netfilter/iptables	5
324.5 OpenVPN	3



学習の方法

- セキュリティ概論について把握
 - どのような脅威があるのか？
 - どのような対策があるのか？
- セキュリティを高める技術
 - 暗号化、アクセス制御、侵入検知など
- 各種ツールの利用方法の確認
 - マニュアルをよく読む（英語でも）



オススメ書籍（暗号関係）

- 『新版暗号技術入門 秘密の国のアリス』
 - － 詳細ですが、比較的読みやすい
 - － 『暗号技術大全』と構成が似ているので、これで十分な場合も
- 『暗号技術大全』
 - － 網羅的だが、読むのが大変
 - － 絶版なので、中古で買うしかない



ポイント解説 I

OpenSSL



HTTPSの3つの役割

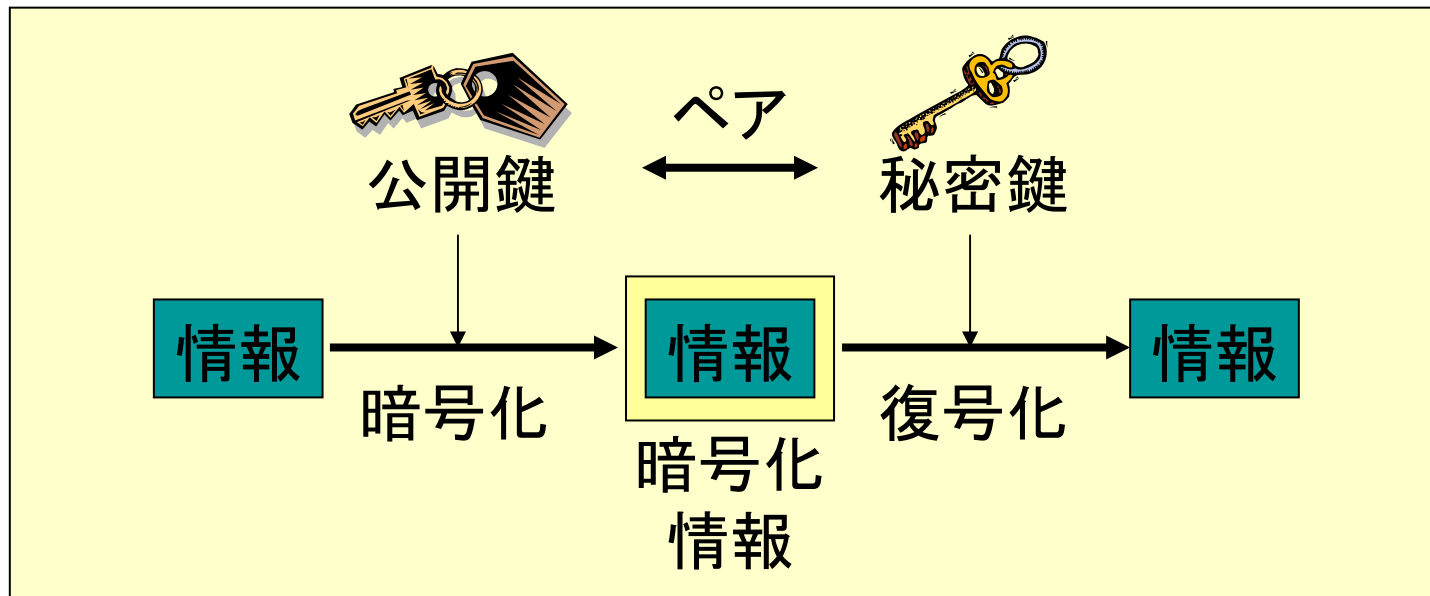
SSLを介したHTTPをHTTPSと呼ぶ

1. サーバーのなりすましを防ぐ
 - 認証局(が電子署名したサーバー証明書)でサーバー自体の信頼性を保証する
2. 送受信されるデータの盗み見を防ぐ
 - 共通鍵暗号で通信内容を暗号化する
3. データの改ざんを防ぐ
 - メッセージ認証コードでデータが改ざんされていないことを確認する



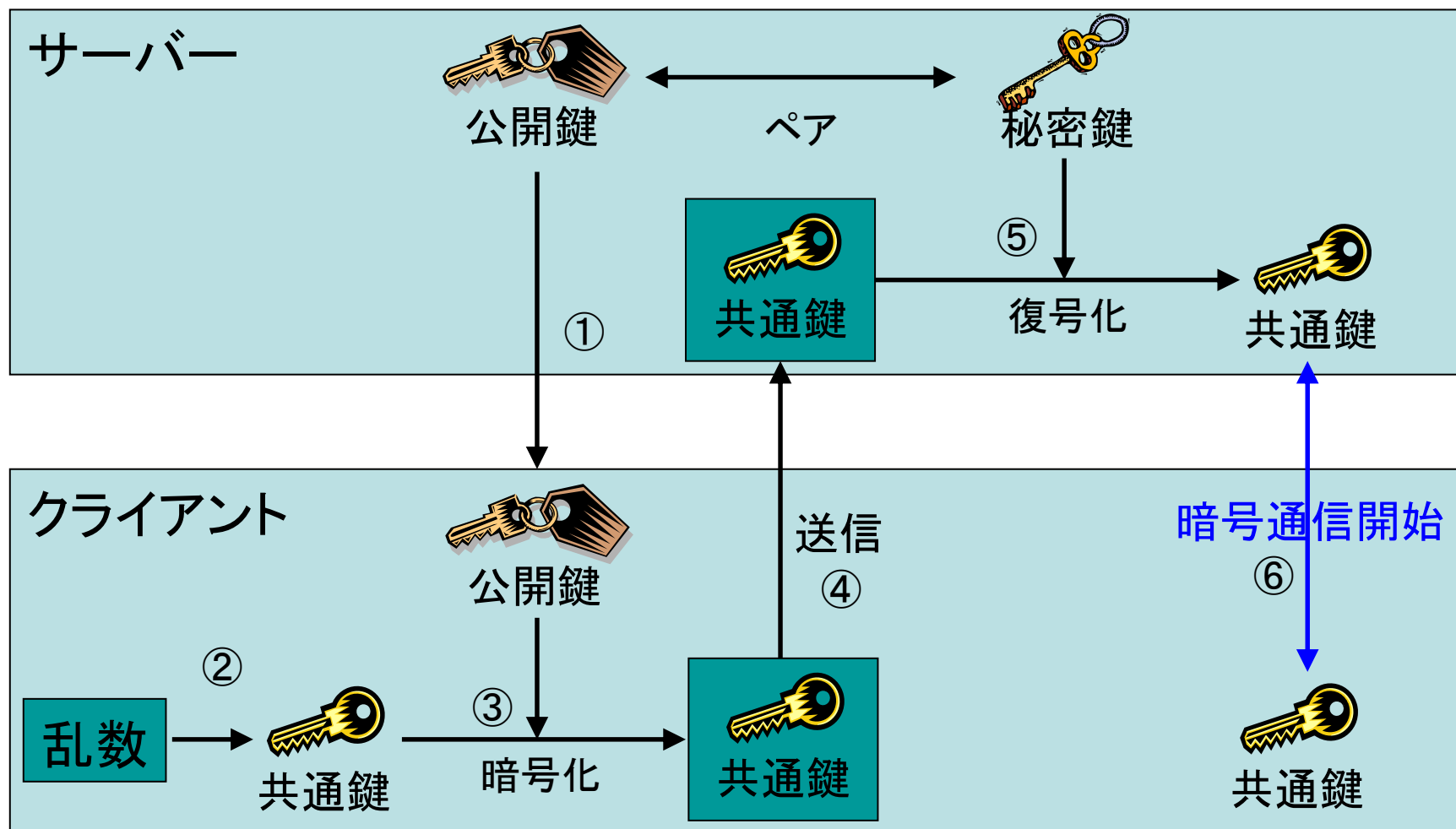
秘密鍵/公開鍵暗号の仕組み

- 秘密鍵と公開鍵はペアで生成される
- 公開鍵で暗号化された情報は、秘密鍵だけが復号化(元の情報を取り出す)できる

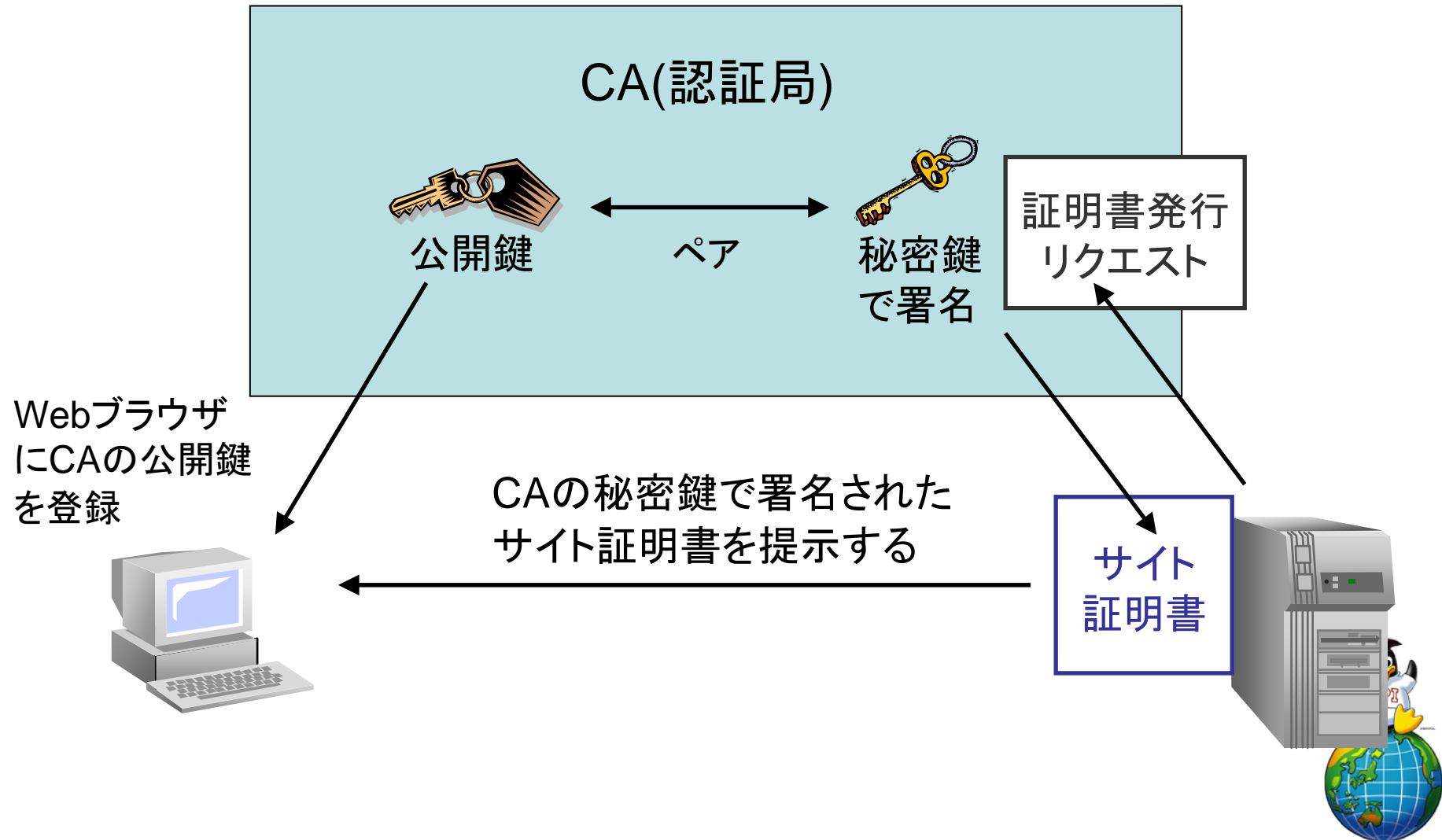


共通鍵を送る手順

～公開鍵暗号通信～



CAを介した信頼性の保証



SSLの設定手順(参考)



HTTPSの提供に必要なもの

- Apacheのsslモジュール
- サーバーの公開鍵と秘密鍵
 - セッション開始時にクライアントから受信した共通鍵を復号化するため
- 認証局(CA)が電子署名したサイト証明書
 - 信頼できるサーバーである(なりすましではない)ことを確認できる
 - サーバーの運営主体やコンテンツの内容が信頼できるわけではない
 - 本人確認が甘いCAもあるので、信頼性が低い場合もある



HTTPSの設定手順

1. サーバーの秘密鍵の作成
2. CSR (Certificate Signing Request) の作成
3. CAにサイト証明書へ電子署名してもらう
4. Apacheの設定
 - ssl.confにヴァーチャルホストとして設定
 - ssl.confはデフォルトでhttpd.confから読み込まれるように設定されている
 - httpd.confそのものに記述する場合もある



サーバーの秘密鍵の作成

1. OpenSSLでサーバーの秘密鍵を作成する
2. 擬似乱数ファイルを生成
 - # openssl md5 * > rand.dat
3. 秘密鍵ファイルを作成
 - # openssl genrsa -rand rand.dat -des3 1024 > key.pem
 - パスフレーズを2回入力
 - Apache起動時にパスフレーズ入力が必要となる



CSRを作成

- CSR (Certificate Signing Request) でCAにサイト証明書の発行を依頼
 - CSR、サイト証明書にはサーバーの公開鍵が含まれる
- 作成したサーバーの秘密鍵を使ってCSR作成
 1. # openssl req -new -key key.pem -out csr.pem
 2. 証明書用の入力情報は適宜入力
 3. カレントディレクトリにCSRファイルcsr.pemが作成される



テスト用CAの構築

- CSRはCAに送付して電子署名してもらおう
 - 2週間お試し証明書サービスもある(日本ベリサイン)
 - <http://www.verisign.co.jp/server/trialserver/>
- HTTPSのテストを行うために自分でCAを構築し、署名をすることができる(自己署名)
- テスト用CAの作成方法
 1. CA用ディレクトリ作成・移動
 2. /etc/pki/tls/misc/CAスクリプトを利用して、CAの公開鍵・秘密鍵を作成
 3. # /etc/pki/tls/misc/CA -newca
 4. CA用の入力情報は適宜入力



署名済みサイト証明書を発行

1. CSRのファイル名をnewreq.pemに変更して/root/CA/ディレクトリにコピー
2. CAスクリプトで署名し、サイト証明書を発行
 - CA用ディレクトリで作業
 - # /etc/pki/tls/misc/CA -sign
3. /root/CA/ディレクトリ内に署名済みサイト証明書ファイルnewcert.pemができる



ssl.confの設定

以下のディレクティブを確認・編集する

- SSLCertificateFileディレクティブ
 - 署名済みサイト証明書のファイルを指定する
- SSLCertificateKeyFileディレクティブ
 - サーバーの秘密鍵のファイルを指定する
- その他のディレクティブは通常のバーチャルホストと同様に設定



Apacheのインストール

1. yumコマンドを使用してインストール

- # yum install httpd mod_ssl

- httpd: Apache本体
- mod_ssl: SSLモジュール
- distcacheモジュールも同時にインストールされる



サーバー秘密鍵の作成

1. 擬似乱数を生成

```
- # openssl md5 * > rand.dat
```

2. サーバー秘密鍵の作成

```
- # openssl genrsa -rand rand.dat -des3 1024 >  
key.pem
```

- パスフレーズを2回入力



CSRの作成

1. CSRを作成

- # openssl req -new -key key.pem -out csr.pem
- 必要情報を入力

Country Name (2 letter code) [GB]:**JP**
State or Province Name (full name) [Berkshire]:**Tokyo**
Locality Name (eg, city) [Newbury]:**Chiyodaku**
Organization Name (eg, company) [My Company Ltd]:**JPI-Japan**
Organizational Unit Name (eg, section) []:(**無入力**)
Common Name (eg, your name or your server's hostname) []:**www.lpi.or.jp**
Email Address []:**info@lpi.or.jp**

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:(**無入力**)
An optional company name []:(**無入力**)



openssl.cnfの編集

1. /etc/pki/tls/openssl.cnfを編集

– 念のためバックアップを作成すると良い

- # cd /etc/pki/tls/
- # cp openssl.cnf openssl.cnf.bak

2. 以下の行を変更

– dir

- ../../CA から .に変更

– basicConstraints

- CA:FALSEになっているものをすべてCA:trueeに変更



CAスクリプトの編集

1. /etc/pki/tls/misc/CAを編集

– 念のためバックアップを作成すると良い

- # cd /etc/pki/tls/misc
- # cp CA CA.bak

2. 以下の行を変更

– CATOP

- ../../ca から . に変更



CAの構築

1. /root/CAディレクトリを作る
 - # mkdir /root/CA
2. /root/CAディレクトリに移動する
 - # cd /root/CA
3. 独自の認証局を構築する
 - # /etc/pki/tls/misc/CA -newca
 - CA秘密鍵のパスフレーズ入力と、CA情報の入力が必要



CA情報の入力

CAの情報入力例

Country Name (2 letter code) [GB]:**JP**
State or Province Name (full name) [Berkshire]:**Tokyo**
Locality Name (eg, city) [Newbury]:**Chiyodaku**
Organization Name (eg, company) [My Company Ltd]:**LPI-Japan**
Organizational Unit Name (eg, section) []:**(無入力)**
Common Name (eg, your name or your server's hostname) []:**ca.lpi.or.jp**
Email Address []:**info@lpi.or.jp**

Please enter the following 'extra' attributes
to be sent with your certificate request

A challenge password []:**(無入力)**

An optional company name []:**(無入力)**

Using configuration from /etc/pki/tls/openssl.cnf

Enter pass phrase for ./private/.cakey.pem:**(CAのパスフレーズを入力)**

Check that the request matches the signature

Signature ok



サイト証明書の発行

1. CSRを/root/CAディレクトリにコピー
 - ファイル名をnewreq.pemに変更すること
 - # cp ../csr.pem newreq.pem
2. 署名済みサイト証明書を生成する
 - # /etc/pki/tls/misc/CA -sign
 - CA秘密鍵のパスフレーズ入力が必要
 - newcert.pemが生成される



秘密鍵とサイト証明書の設置

1. 作成されたファイルを適切な場所に配置
 - 配置用ディレクトリがないので作成すること
2. サーバー秘密鍵key.pemの設置
 - /etc/httpd/conf/ssl.key/server.keyとして設置
 - パーMISSIONは厳重に設定する
 - ディレクトリは700、ファイルは600にを設定
3. サイト証明書newcert.pemの設置
 - /etc/httpd/conf/ssl.crt/server.crtとして設置



Apacheの設定と接続確認

1. ssl.confを編集

- # vi /etc/httpd/conf.d/ssl.conf

2. ドキュメントルートの設定

- DocumentRoot "/var/www/html"

3. サイト証明書とサーバー秘密鍵の設定

- SSLCertificateFile

 - /etc/httpd/conf/ssl.crt/server.crt

- SSLCertificateKeyFile

 - /etc/httpd/conf/ssl.key/server.key



動作テスト

1. Apacheを再起動
 - # service httpd restart
 - うまく設定が行われていれば、サーバー秘密鍵のパスフレーズ入力を要求される
2. WebブラウザにCAの公開鍵をインポート
 - /root/CA/cacert.pemをインポートする
3. https://サーバー名/にアクセス
 - サーバー名はサーバー証明書に記載された名前でないといけない
 - 検証レベルであれば、DNSか/etc/hostsに記述
4. SSL証明書を確認



ポイント解説 Ⅱ

OpenVPN

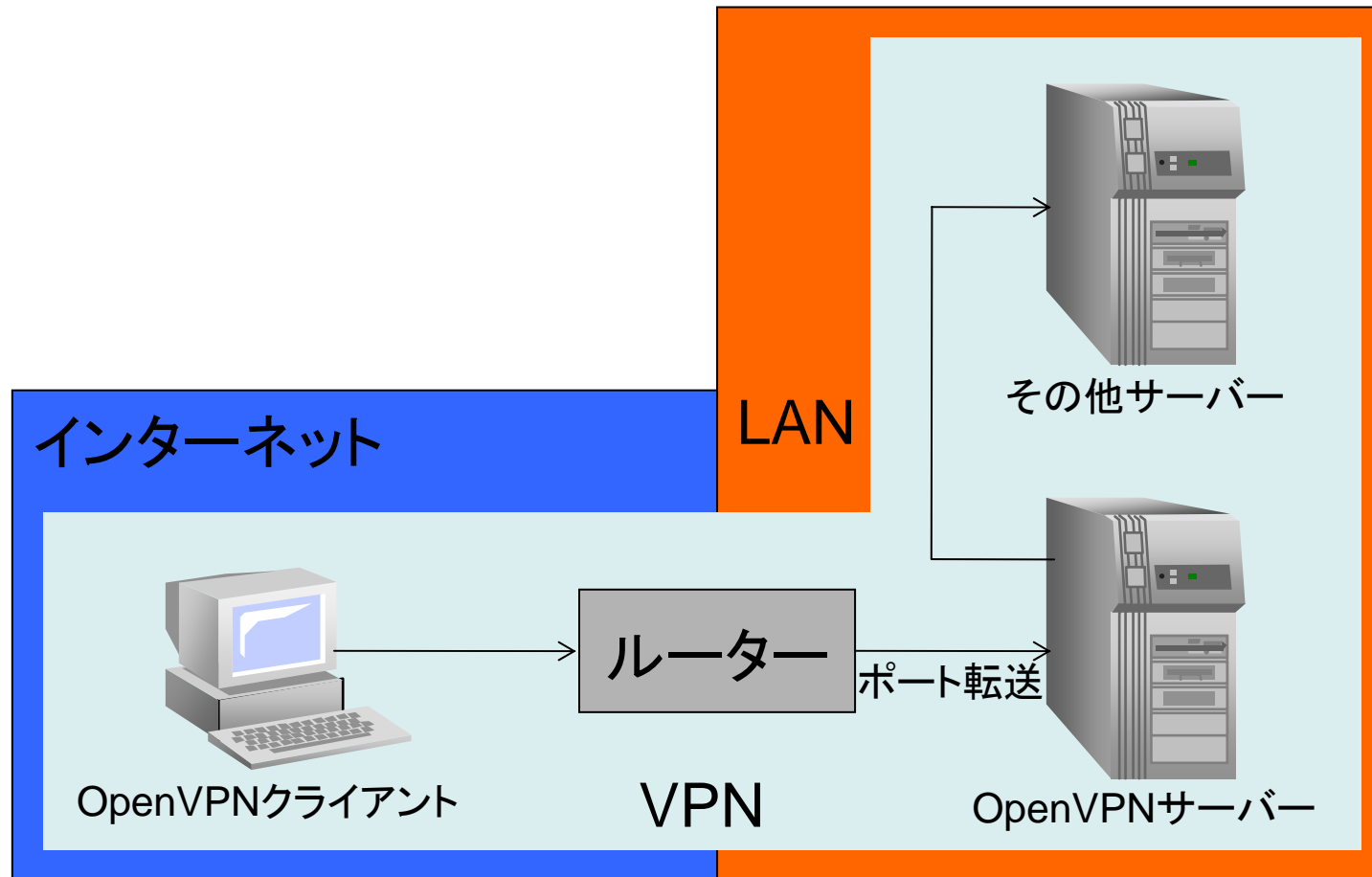


OpenVPN

- SSLを利用したSSL VPN
- GPLで提供されている
 - 非オープンな商用ライセンスも選択可能
- マルチOSサポート
 - Linux、Windows、各種BSD系OS、Solaris等
- 公開鍵認証やパスワード認証が可能



OpenVPNによるVPN構築例



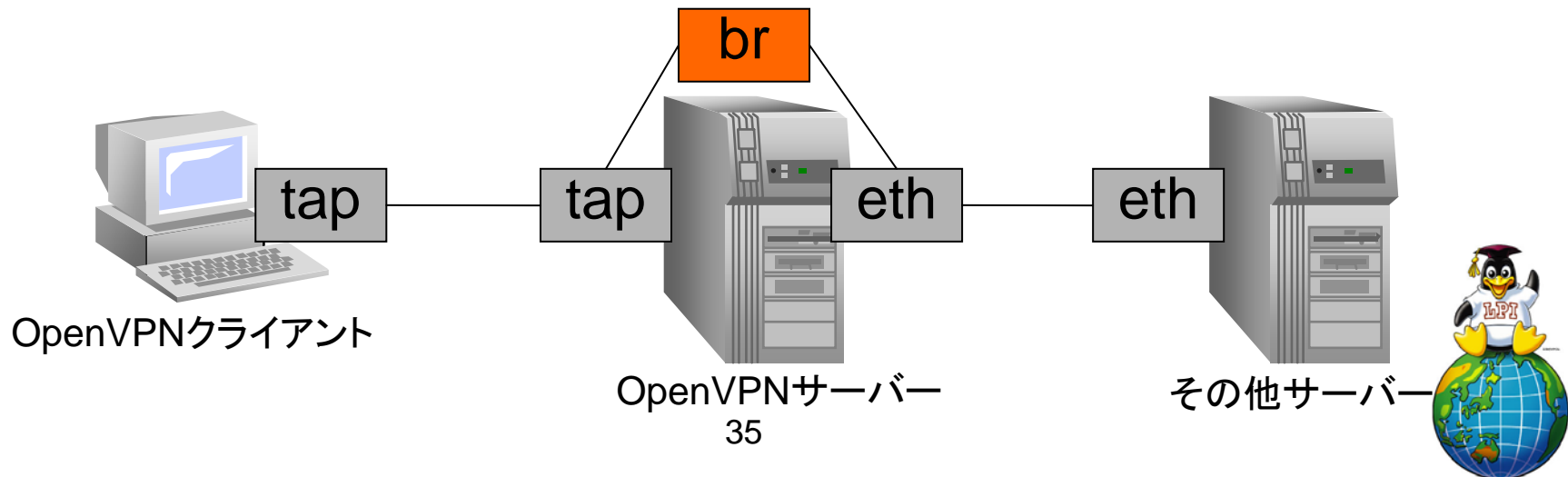
OpenVPN設定時のポイント

- ルーティングモードとブリッジモードがある
 - Windowsファイル共有のようにブロードキャストを使う場合はブリッジモードが楽
 - 今回はブリッジモードで設定
- OpenVPNサーバーとクライアントは別セグメント
 - 間にルーター等を入れてポート転送などを行う
- OpenVPNサーバーのNICは1つで良い
 - もちろん別NICとの間でのルーティングやNATも可能



tun/tapデバイスとブリッジ

- 仮想NIC tun/tapデバイス
 - カーネルモジュールが必要
 - tunはL3、tapはL2で動作
- tapデバイスとethデバイスをブリッジ接続



OpenVPNの設定手順(参考)

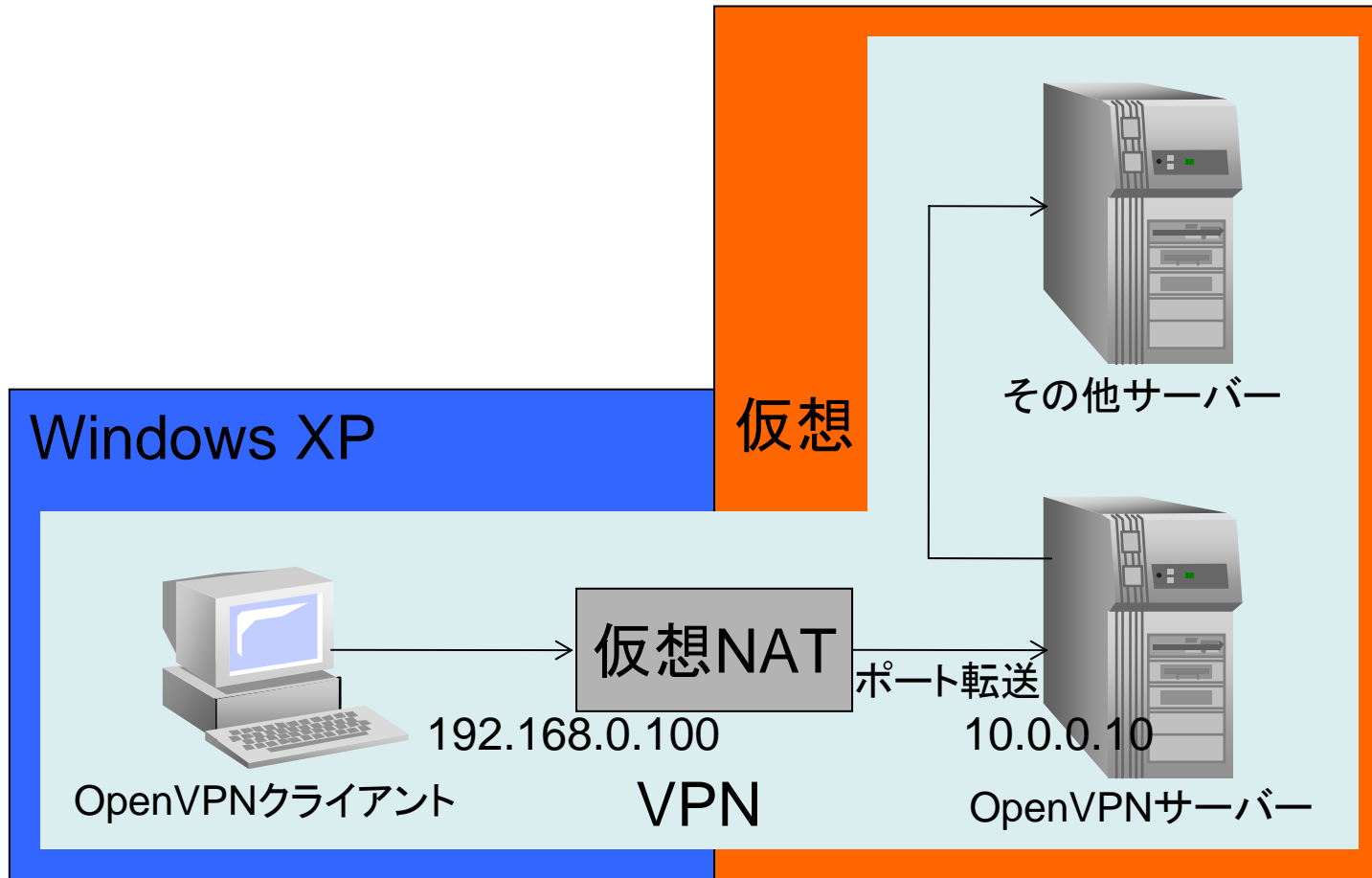


事前準備

- デモ環境はVMware Workstationを利用
 - ルーターの代わりに仮想NATを使用
- クライアントの設定
 - OSはWindows XP Professional
 - NICを192.168.0.100/24に設定
- サーバーの導入
 - OSはCentOS 5.3
 - 仮想NATネットワークに接続
 - eth0を10.0.0.10/8に設定
- 仮想NATのポート転送設定
 - クライアント→サーバーにポート1194を転送



デモ環境



作業上の注意

- 時刻を合わせておく
 - CA作成前にNTPなどで時間を合わせておく
 - 時間がずれた状態で作業をするとやり直しになる
- OpenVPNサーバーの自動起動に注意
 - openvpnパッケージはOpenVPNサーバーを自動起動に設定する
 - ブリッジモードで使用する場合、ブリッジ設定スクリプトが別なので、ブリッジ無しで起動してしまう
 - 作業途中でシステムの再起動などを行った場合には要注意



OpenVPN サーバー構築手順

- 必要なパッケージの導入
- 各種認証関係ファイルの作成
 - CAの作成
 - サーバー証明書の作成
 - DHパラメータの作成
 - 証明書廃止リストの作成
 - TLS認証鍵の作成
- ブリッジの設定
- OpenVPNサーバーの設定



必要なパッケージの導入

- bridge-utilsパッケージをインストール
 - ブリッジモードで設定する場合に必要
- RPMforgeを使用可能にする
 - 手順は<https://rpmrepo.org/RPMforge/Using>を参照
- openvpnパッケージをインストール
 - `# yum install openvpn`
 - lzo2パッケージも一緒にインストールされる



easy-rsaの導入

- makeの実行でインストール可能
 - インストール先として/etc/openvpn/easy-rsaを指定
 - # cd /usr/share/doc/openvpn-*/easy-rsa/2.0/
 - # make install DESTDIR=/etc/openvpn/easy-rsa
- インストールの確認
 - # cd /etc/openvpn/easy-rsa/



CAの作成

- /etc/openvpn/easy-rsa/varsの修正
 - export KEY_COUNTRY="JP"
 - export KEY_PROVINCE="Tokyo"
 - export KEY_CITY="Chiyodaku"
 - export KEY_ORG="LPI-Japan"
 - export KEY_EMAIL=info@lpi.or.jp
- CAの作成
 - # source vars
 - # ./clean-all
 - # ./build-ca
- CAの証明書のコピー
 - # cp keys/ca.crt /etc/openvpn/



サーバー証明書を作成

- サーバー証明書の作成
 - # ./build-key-server server
 - チャレンジパスワードの設定は不要
 - [y/n]が聞かれたら、yを入力(2回)
- サーバー証明書のコピー
 - # cp keys/server.crt /etc/openssl/
 - # cp keys/server.key /etc/openssl/
 - # chmod 600 /etc/openssl/server.key



DHパラメータの作成

- DHパラメータの作成
 - # ./build-dh
- DHパラメータのコピー
 - # cp keys/dh1024.pem /etc/openvpn/



証明書廃止リストの作成

- openssl.cnfの修正
 - #[pkcs11_section]
 - #engine_id = pkcs11
 - #dynamic_path = /usr/lib/engines/engine_pkcs11.so
 - #MODULE_PATH = \$ENV::PKCS11_MODULE_PATH
 - #PIN = \$ENV::PKCS11_PIN
 - #init = 0
- ダミーのクライアント証明書の作成と破棄
 - # ./build-key dummy
 - # ./revoke-full dummy
- CRLのコピー
 - # cp keys/crl.pem /etc/openvpn/



静的暗号鍵の作成

- SSL/TLSのセキュリティ強化のための静的暗号鍵(事前共有鍵)を作成
 - 「HMACファイアウォール」とも呼ばれる
- サーバーとクライアントが同じ鍵を持っていないと接続が行えない
 - 詳細についてはマニュアルの--tls-authの記述を参照
- openvpnコマンドに--genkeyオプションをつけて実行
 - # openvpn --genkey --secret /etc/openvpn/ta.key



ブリッジの設定

- ブリッジ設定スクリプトのコピー
 - # cp /usr/share/doc/openvpn-2.0.9/sample-scripts/bridge-st* /etc/openvpn/
 - chmod +x /etc/openvpn/bridge-st*
- bridge-startスクリプトパラメータの編集
 - eth_ip="10.0.0.10"
 - eth_netmask="255.0.0.0"
 - eth_broadcast="10.255.255.255"



OpenVPNサーバーの設定

- 設定ファイルのコピー
 - `cp /usr/share/doc/openvpn-2.0.9/sample-config-files/server.conf /etc/openvpn/`
- 設定ファイルの修正
 - TCPを使用
 - tapデバイスとブリッジモードを使用
 - 証明書関係ファイルは/etc/openvpnディレクトリに配置



OpenVPNサーバーの設定詳細

```
# udpからtcpに変更  
proto tcp  
;proto udp
```

```
#devはtap0とする。  
dev tap0  
;dev tun
```

```
#証明書関係ファイルの指定  
#デフォルトでは/etc/openvpn/を参照  
ca ca.crt  
cert server.crt  
key server.key
```

```
#DHパラメータの指定  
dh dh1024.pem
```

```
#サーバーのアドレス設定はコメントアウト  
;server 10.8.0.0 255.255.255.0
```

```
#その代わりに、サーバーブリッジのコメントアウトを外し、修正  
server-bridge 10.0.0.10 255.0.0.0 10.0.0.50 10.0.0.100
```

```
#コメントアウトを外す(1箇所のみ)
```

```
#/etc/openvpn/ccdのクライアント別設定ファイルを参照する  
client-config-dir ccd
```

```
#コメントアウトを外す
```

```
client-to-client  
duplicate-cn  
tls-auth ta.key 0 ;サーバー側は0を設定。
```

```
#デーモンをnobody権限で実行する
```

```
user nobody  
group nobody
```

```
#ログファイル等は必要に応じて設定
```

```
status /var/log/openvpn-status.log  
log /var/log/openvpn.log  
log-append /var/log/openvpn.log
```

```
#CRLの有効化設定を追加
```

```
crl-verify crl.pem
```



OpenVPNサーバーの起動

- ブリッジの作成
 - # cd /etc/openvpn
 - # ./bridge-start
- ブリッジの確認
 - # brctl show
 - eth0とtap0がbr0に接続されていることを確認
 - # ifconfig
- OpenVPNサーバーの起動
 - # service openvpn start



OpenVPNクライアントの設定

- OpenVPNクライアントのインストール
- クライアント証明書を作成
- 各種証明書関連ファイルのコピー
- クライアント設定ファイルを作成



OpenVPNクライアントのインストール

- OpenVPN GUI for Windows
 - <http://openvpn.se/>
- Tunnelblick
 - Mac OS X用OpenVPNクライアント
 - <http://code.google.com/p/tunnelblick/>



クライアント証明書を作成

- CAの準備
 - # cd /etc/openvpn/easy-rsa
 - # source vars
- クライアント証明書の作成(client1用)
 - # ./build-key-pass client1
 - パスフレーズを2回入力する
 - [y/n]が聞かれたら、yを入力(2回)
 - keysディレクトリにclient1.crtとclient1.keyが作成される



各種証明書関連ファイルのコピー

- 認証に必要なとなる証明書関連ファイルをクライアントにコピー
 - 保存先は"C:¥Program Files¥OpenVPN¥config"
- CA証明書
 - ca.crt
- 静的暗号鍵
 - ta.key
- クライアント証明書
 - client1.crt
 - client1.key



クライアント設定ファイルの例

```
pull
tls-client
dev tap
proto tcp-client
remote 接続先アドレス 1194
resolv-retry infinite
nobind
persist-key
persist-tun
ca ca.crt
cert client1.crt
key client1.key
ns-cert-type server
tls-auth ta.key 1
comp-lzo
verb 3
```

- ca CAの証明書
- cert クライアント証明書
- key クライアント秘密鍵
- tls-auth 静的暗号鍵 1
 - クライアントには1を設定



その他の設定の意味

- proto
 - 使用するプロトコルを指定。UDP、またはTCPが選択できる。サーバーに合わせる。
- ns-cert-type server
 - サーバー証明書作成時に "nsCertType=server" と設定されていないサーバーと接続しない
 - build-key-serverスクリプトでは設定される



VPN接続の確認

- Windowsクライアントのトレイアイコンを右クリックし、「Connect」を選択
- 接続時のログは「View Log」で確認可能
- サーバー・クライアント間でPING確認

