VPN connectivity environment

VPN **接続環境**

Version 1.0

保証免責
本書は記載事項またはそれに関わる事項について、明示的あるいは黙示的な保証はいたしておりません。
したがいまして、これらを原因として発生した損失や損害についての責任を負いません。

表記について

本書では以下の書体を使用しています。

- イタリック文字

  本文中でのコマンド、ファイル名、変数など可変なパラメータ値を表します。

- 等幅文字

  ファイルの内容やコマンドの入出力例に使います。入力の場合にはボールドで表します。

```
$ cd /usr/src/sys/i386/conf
$ ls
GENERIC         Makefile      OLDCARD       SMP
GENERIC.hints   NOTES         PAE           gethints.awk
$
```

- 省略文字

  ファイルの内容やコマンドの入出力例を省略する場合に'...'を使います。

```
$ vi /etc/rc.conf
  ...
sshd_enable="YES"
named_enable="YES"
  ...
$
```

- プロンプト

  一般または、管理権限を持った実行環境をそれぞれ、'$'(ドル)、'#'(シャープ)のプロンプトで表します。

```
$ su
Password: root's passwd
#
```

目次

1. はじめに

1.1. 本書について

本書では、安価な VPN ソリューションの 1 つとして PPP over SSH を用いた拠点間の VPN 接続環境を構築した手順をまとめたものです。

1.2. 対象とする読者

Unix の一般的コマンド、および管理コマンドを使用することができ、vi による編集が可能であることを想定しています。また、実装される各ソフトウェアについて熟知してればなおよいでしょう。本書では、ソフトウェア上の設定に関して、*parameter = value* といった実際の設定情報についてのみ記述します。繰り返しになりますが、これらの設定情報についての詳細は関連マニュアルを参照するべきでしょう。

以下に挙げるドキュメントを参照しておくことを推奨します。

文献

文献	著者	リンク
FreeBSD Man Pages	The FreeBSD Project.	http://www.freebsd.org/cgi/man.cgi
PPP over SSH with NAT	Tomoki Sekiyama	http://hydro.energy.kyoto-u.ac.jp/~sekiyama/PPPoSSHwithNAT/pppossh.html
ppp over ssh	HIROSE Yuuji	http://www.gentei.org/~yuuji/rec/pc/memo/2002/02/01/
Why TCP over TCP Is Bad Idea なぜ TCP over TCP は悪いアイデアか	Olaf Titz 前田 修吾	http://sites.inka.de/sites/bigred/devel/tcp-tcp.html http://shugo.net/docs/tcp-tcp.html

1.3. 想定するネットワーク構成

以下の構成内容となります。後述するネットワーク構成図も併せて参照してください。

① VPN ゲートウェイ間をトンネリング

拠点ごとに VPN ゲートウェイを構築します。VPN ゲートウェイ間は SSH を確立し、その通信路上に PPP をトンネリングさせます。HOME Network は発呼側、Labs Network は着呼側となります。

② VPN ゲートウェイ上での NAPT

拠点間の PPP トンネリングを確立しただけでは、VPN server 以外のノードや他のネットワークに到達できません。このため、VPN server 上で NAPT を行ないます。

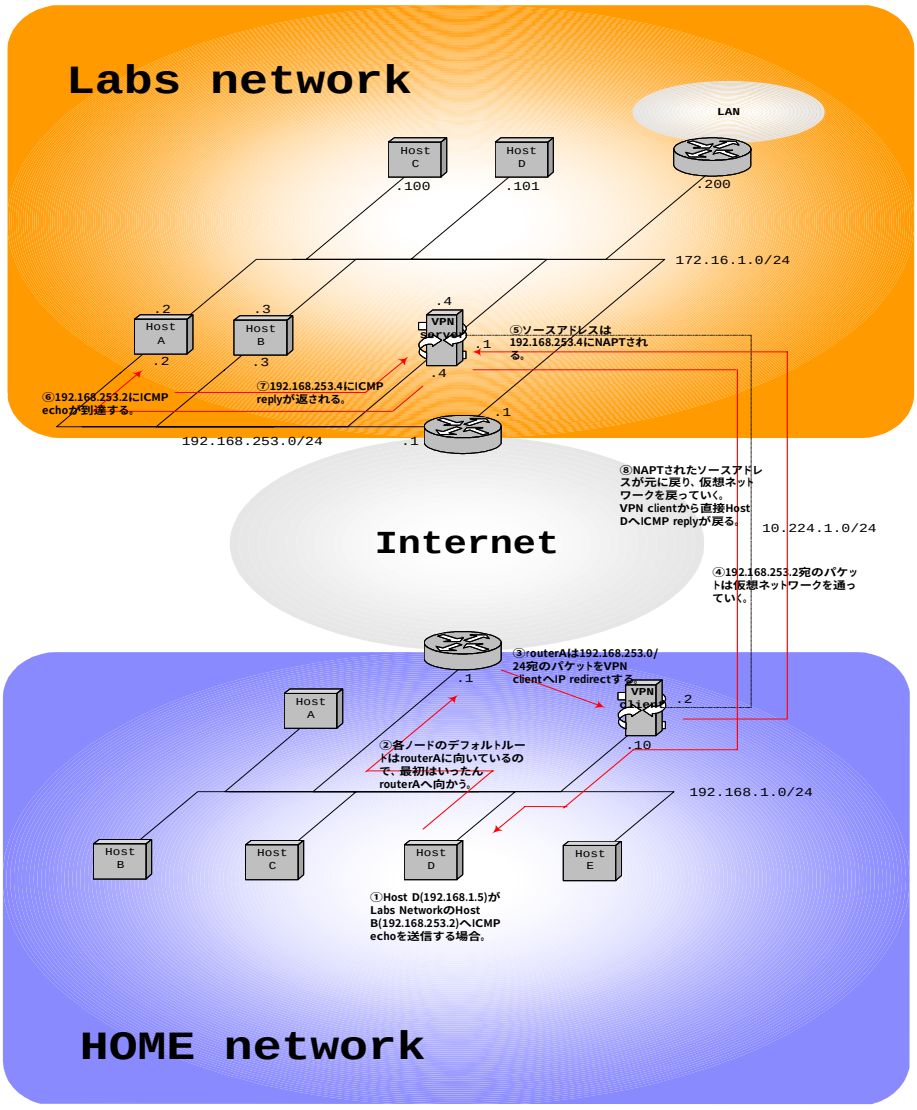
③ VPN ゲートウェイ上でのパケットフィルタリング

今日の拠点間通信において自由な相互通信が許されていることはまれだと考えられます。例えば、HOME Network から Labs Network へ、またはその逆には適切なフィルタリングを必要とするでしょう。

④ HOME Network の経路制御

HOME Network から Labs Network にアクセスする場合、HOME Network のルータは Labs Network の経路情報について知っている必要があります。HOME Network のルータは Labs Network へ向かうパケットに対して VPN client へルーティングするように設定します。

想定するネットワーク構成



1.4. 構築対象ノード

以下のノードとなります。ネットワーク構成図も併せて参照してください。

対象ノード

拠点	ノード名	I/F 名とアドレス	備考
Labs Network	routerA	192.168.1.1/24	ルータ。
Labs Network	VPN server	em0: 192.168.1.10 tun0: 10.224.1.2	VPN 着呼側。
HOME Network	VPN client	em0: 192.168.253.4 em1: 172.16.1.4 tun0: 10.224.1.1	VPN 発呼側。

2. VPN サービス構築 - サーバ編

2.1. サービス概要

PPP サーバとして、PPP クライアントからの VPN 接続を受け付けます。
PPP サーバでは、PPP over SSH による SSL 公開鍵認証を利用した PPP クライアントの真正性確認と、Internet 上でも SSL 暗号化による安全な通信路を確保します。

以下は Labs Network の VPN server に対する設定例を記述しています。

2.2. 準備

既にインストールされている ssh と ppp を使用します。

最初に、非特権ユーザでも動作するよう専用アカウント pppossh を作成しておきます。グループ権限には、network を含めるようにします。これにより、非特権ユーザでも ppp を動作させることができます。

```
$ grep pppossh /etc/passwd
pppossh:*:101:101:PPPoSSH account:/home/pppossh:/bin/sh
$ grep vpn /etc/group
vpn:*:101:
$ grep pppossh /etc/group
network:*:69:pppossh
```

特定の PPP クライアントからのみアクセスを許可できるように SSL 公開鍵認証によるアクセス制御を行います。PPP クライアントのホスト上で鍵ペアを生成し、パスフレーズは空にしておきます。

クライアントで作成した SSL 公開鍵を~pppossh/.ssh ディレクトリにコピーしておきます。この時点で、pppossh@*VPN server* で ssh によるログインが可能であることを確認するべきです。

```
$ su - pppossh
$ ls -l .ssh
  ...
-rw-r--r--  1 pppossh  vpn  856 Mar 30 16:24 authorized_keys
```

2.3. ppp 設定

最初に、非特権ユーザでも ppp の動作をカスタマイズできるようにします。デフォルトでは、/etc/ppp/ppp.conf ファイルを参照しますので、このファイルを以下のように変更します。

ppp.conf ファイル

パラメータ	用途･説明
!include ~/.ppp.conf	ユーザのホームディレクトリにある.ppp.conf ファイルを参照するように指定する。
default: allow user ppposh	デフォルトのエントリ。 ppp の実行を許可するユーザを列挙する。

~ppposh/.ppp.conf ファイルに以下の設定を行います。このファイルは、クライアントからの PPP over SSH 接続時に、ダイレクトモードで PPP 接続する際に参照されます。

.ppp.conf ファイル

パラメータ	用途
default:	デフォルトのエントリ。
ppposh_approve_4_HOMENetwork:	HOME Network 用エントリ。
allow mode direct	ダイレクトモードの接続を許可する。
set timeout 0	アイドルタイマ機能を無効にする。
set log All -Async -Debug -Physical -Radius -Sync -TCP/IP -HDLC -Timer	ログの出力レベルを指定する。 `-'の付いたレベルは出力しない。
set escape 0xff	リンク時にエスケープされる文字を指定。
disable ipv6cp	IPv6 制御プロトコルのケーパビリティを試行しない。
add 192.168.1.0/24 HISADDR	HOME Network への戻りのネットワーク経路を追加する。 この経路情報は拠点毎に依存します。

3. VPN サービス構築 - クライアント編

3.1. サービス概要

PPP クライアントから、PPP サーバへ VPN 接続を行います。
PPP クライアントは拠点ネットワーク上のゲートウェイとして動作します。この VPN サービスは拠点間 VPN 接続になります。

以下は HOME Network の VPN client に対する設定例を記述しています。

3.2. 準備

既にインストールされている ssh と ppp を使用します。

最初に、PPP サーバと同様に非特権ユーザで動作するよう専用アカウント pppossh を作成しておきます。PPP サーバは、SSL 公開鍵認証によるアクセス制御を行いますので、pppossh ユーザの鍵ペアを生成します。その際、パスフレーズは空にしておきます。

```
$ su - pppossh
$ ssh-keygen -t dsa
  ...
```

作成した SSL 公開鍵を PPP サーバの~pppossh/.ssh へコピーしてください。

3.3. ppp 設定

最初に、非特権ユーザでも ppp の動作をカスタマイズできるようにします。デフォルトでは、/etc/ppp/ppp.conf ファイルを参照しますので、このファイルを以下のように変更します。

ppp.conf ファイル

パラメータ	用途･説明
!include ~/.ppp.conf	ユーザのホームディレクトリにある.ppp.conf ファイルを参照するように指定する。
default: allow user pppossh	デフォルトのエントリ。 ppp の実行を許可するユーザを列挙する。

~ppposshl/.ppp.conf ファイルに以下の設定を行います。このファイルは、PPP クライアントから PPP サーバへ PPP 接続する際に参照されます。

.ppp.conf ファイル

パラメータ	用途
default:	デフォルトのエントリ。
pppossh_request:	
set timeout 0	アイドルタイマ機能を無効にする。
set log All -Async -Debug -Physical -Radius -Sync -TCP/IP -HDLC -Timer	ログの出力レベルを指定する。 `-'の付いたレベルは出力しない。
set ifaddr 10.233.1.2 10.233.1.1	IPCP ネゴシエーション時に割当てる仮想的なアドレス。 ローカル側とリモート側に指定する。
set device "!ssh pppossh@*VPN server* exec /usr/sbin/ppp -direct -unit0 pppossh_approve_4_"	PPP サーバへ ssh 接続後、ppp をダイレクトモードで起動する(PPP over SSH)。各拠点に整合したエントリを指定する。 ダイレクトモードでは、PPP サーバは標準入力で接続を受け入れるようになる。 Labs Network では、サーバ側のトンネリング I/F は tun0 を使用します。
add 192.168.253.0/24 HISADDR add 172.16.1.0/24 HISADDR	Labs Network への行きのネットワーク経路を追加する。 この経路情報は拠点ごとに依存します。
set server ~/tmp/tun.0 "" 0177	ローカルドメインソケットを指定する。 pppctl(8)などで利用することができる。
disable ipv6cp	IPv6 制御プロトコルのケーパビリティを試行しない。
ident "HOME Network"	PPP サーバに対して自己証明する。 これは認証用ではなく、ロギング用に設定。

3.4. 自動起動化

VPN client がリブートした場合でも、自動的に拠点間で VPN 接続を行うよう、諸設定を行う必要があります。最初に、起動スクリプトを用意します。起動スクリプトには実行権限を与えてください。

```
$ cat /usr/local/etc/rc.d/pppossh.sh
#!/bin/sh
  ...
# PROVIDE: pppossh
# REQUIRE: network1
# KEYWORD: FreeBSD shutdown

. /etc/rc.subr

PPPOSSH_HOME=/home/pppossh

name="pppossh"
rcvar=`set_rcvar`
command=""
procname="/usr/sbin/ppp"
command_args=""
pidfile="/var/run/tun0.pid"
required_files="$PPPOSSH_HOME/.ppp.conf $PPPOSSH_HOME/.ppposshrc"
start_precmd="pppossh_prestart"
stop_cmd="pppossh_stop"

pppossh_prestart() {
  su pppossh -c ". $PPPOSSH_HOME/.ppposshrc"
}

pppossh_stop() {
```

```
  if [ -e "$pidfile" ]; then
    su pppossh -c "kill -HUP `cat $pidfile`"
    echo "Stopping pppossh."
  fi
}

load_rc_config $name
run_rc_command "$1"
  ...
```

次に、/etc/rc.conf の以下の部分を追加します。

```
$ cat /etc/rc.conf
  ...
pppossh_enable="YES"
```

最後に、トンネリング用インタフェースへ、IP 転送が行えるように、以下の設定を追加します。
これは、rc.conf へ gateway_enable="YES"を追加する方法と等価です。

```
$ cat /etc/sysctl.conf
  ...
net.inet.ip.forwarding=1
```

3.5. HOME Network 用の静的経路を追加

これは、routeA で設定します。
Labs Network へ直接アクセス可能とするためには、HOME Network のデフォルトゲートウェイに、Labs Network への経路を追加して、IP が VPN client へ転送されるように設定する必要があります。routeA では、以下のように設定します。

```
$ cat /etc/rc.local
  ...
route add -net 192.168.253.0 -netmask 255.255.255.0 192.168.253.10
route add -net 172.16.1.0 -netmask 255.255.255.0 192.168.253.10
```

routeA が FreeBSD などのサーバ OS ではなかった場合でも、上述の静的経路を追加することに変わりはありません。
一度、デフォルトゲートウェイから VPN client へ IP 転送が行われると、送信元へ ICMP リダイレクトが送られ、送信元ホストの経路表に Labs Network への経路が追加されるでしょう。

4. パケットフィルタリング/NAPT サービス構築

4.1. サービス概要

トンネリングされた仮想ネットワークや HOME Network から入る、または Labs Network から出る不必要なパケットのフィルタリングや、HOME Network 上の各 PC が Labs Network の任意ノードへ接続できるよう NAPT を行います。また、必要なパケットのログを取得します。

本章での記述はあくまでも例ですので、適宜環境に応じて変更する必要があります。

4.2. 準備

ipf、ipnat、はローダブルモジュールとしてカーネルに組込みます。また、ipf のロギング用に ipmon を動作させます。動作制御は添付資料 **A. rc.conf 設定パラメータ**を参照してください。

4.3. ipf 設定

デフォルトポリシとしては「パケットを通過させない」としています。必要に応じてポートを開け、基本的にはステートフルでルールセットを動的に生成させ、戻りのパケットに関する静的ルールを記述しません。/etc/ipf.rules ファイルを以下のように設定します。

ipf.rules ファイル

パラメータ	用途
暗黙のグループ 0	
block in log quick from any to any with ipopts frag	IP オプションが付加された、または断片化した全ての IP パケットは拒否する。
block in log quick proto tcp from any to any with short	全ての短い TCP パケットは拒否する。
pass in quick on lo0 all pass out quick on lo0 all	ループバックに対する全てのパケットは許可する。
pass in on em0 all head 100 pass out on em0 all head 150	192.168.253.0/24 ネットワーク側 I/F(em0)の incoming/outgoing に対するルールを定義する。
未定義。	
pass in on em1 all head 200 pass out on em1 all head 250	172.16.1.0/24 ネットワーク側 I/F(em1)の incoming/outgoing に対するルールを定義する。
未定義。em1I/F は未使用。	
block in on tun0 all head 300 block out on tun0 all head 350	仮想ネットワーク側 I/F(tun0) の incoming/outgoing に対するルールを定義する。
pass in log quick proto tcp from any to any port = 21 flags S keep state keep frags group 300 pass in log quick proto tcp from any to any port 9900 >< 9990 flags S keep state keep frags group 300	内向の FTP セッションを許可する。 パッシブモードによる 9900～9990 ポート範囲によるデータ転送を許可する。

pass in log quick proto tcp from any to 172.16.1.100 port = 1433 flags S keep state keep frags group 300	内向の SQL Server インスタンスへの接続を許可する。
pass in log quick proto tcp from any to any port = 22 flags S keep state keep frags group 300	内向の SSH セッションを許可する。
pass in log quick proto tcp from any to any port = 23 flags S keep state keep frags group 300	内向の TELNET セッションを許可する。
pass in log quick proto tcp from 192.168.1.0/24 to 192.168.253.2 port = 53 flags S keep state keep frags group 300	内向きの DNS notify セッションを許可する。
pass in log quick proto udp from 192.168.1.8 to 192.168.253.2 port = 53 keep state group 300	内向きの DNS update パケットを許可する。
pass in log quick proto tcp from any to any port = 80 flags S keep state keep frags group 300	内向きの HTTP セッションを許可する。
pass in log quick proto tcp from any to any port = 3389 flags S keep state keep frags group 300	内向のターミナルサービスクライアントセッションを許可する。
pass in log quick proto icmp all group 300	内向の ICMP パケットを許可する。
block in log from any to any group 300	上記以外の内向の通信は拒否する。
pass out quick proto icmp all group 350	外向の ICMP パケットを許可する。
pass out log quick proto tcp from 192.168.253.2 to 192.168.1.6 port = 53 flags S keep state keep frags group 350	外向きの DNS ゾーン転送を許可する。
block out log from any to any group 350	上記以外の外向の通信は拒否する。

ipf のルールを変更した場合には、リブートするか、または以下のようにコマンドを実行します。念のため、NAT テーブルもクリアした方がよいでしょう。

```
# /sbin/ipf -Fa -Z -f /etc/ipf.rules
# /sbin/ipnat -CF -f /etc/ipnat.rules
```

4.4. ipnat 設定

デフォルトポリシとしては VPN server に接続されているすべてのネットワークセグメントに対して NAPT することにしています。

/etc/ipnat.rules ファイルを以下のように設定します。

ipnat.rules ファイル

パラメータ	用途
map em0 192.168.1.0/24 -> 192.168.253.4/32 portmap tcp/udp auto map em0 192.168.1.0/24 -> 192.168.253.4/32	HOME Network から来た 192.168.253.0/24 宛のパケットを NAPT する。
map em1 192.168.1.0/24 -> 172.16.1.4/32 portmap tcp/udp auto map em1 192.168.1.0/24 -> 172.16.1.4/32	HOME Network から来た 172.16.1.0/24 宛のパケットを NAPT する。
rdr em0 192.168.253.3/32 port 53 -> 192.168.1.6 port 53	192.168.253.2 から 192.168.1.6 宛へのゾーン転送を許可することを意味する。 つまり、192.168.253.2 から tcp/53 を 192.168.253.4 へ投げると、VPN server は静的 NAT を行なう。

5. シリアルコンソールサービスの構築

5.1. サービス概要

VPN 接続とは直接関係ありませんが、ネットワーク的な障害(故障、操作ミス等を含む)が発生した場合でも、簡易的なシリアルコンソールサーバとして利用することが可能です。以下は、USB - シリアル変換ケーブルを接続した場合に検証した例です。

5.2. 準備

最初に、非特権ユーザでも動作するようにユーザアカウントのグループ権限には、uucp、dialer を含めるようにします。これにより、非特権ユーザでも cu を動作させることができます。

```
$ grep fs /etc/passwd
isidore:*:1000:1000:maintainer:/home/isidore:/bin/tcsh
$ grep uucp /etc/group
uucp:*:66:isidore
$ grep dialer /etc/group
dialer:*:68:isidore
```

次に、必要なカーネルモジュールをロードします。リブート時にも自動的にモジュールがロードされるように、/boot/loader.conf ファイルに以下の設定を追加します。

```
# cat /boot/loader.conf
uftdi_load="YES"
uplcom_load="YES"
          ...
```

手動でロードする場合には、以下のコマンドを実行します。/boot/kernel/ディレクトリ以下に、uftdi.ko、uplcom.ko モジュールがコンパイルされているか確認してください。

```
# kldload uftdi
# kldload uplcom
```

5.3. 接続方法

モジュールをロードすると、/dev/ucom0 デバイスが生成されます。このアクセスラインに接続します。ログインプロンプトが表示されない場合は、1～2 回改行を入力します。

```
$ cu -l /dev/ucom0
Connected
  ...
~^D
```

接続を終了する場合には、~(チルダ)を入力して Ctrl+D を押下します。

A. rc.conf 設定パラメータ

rc.conf ファイル

パラメータ	用途･説明
hostname="talisker"	
ifconfig_em0="inet 172.30.254.3　netmask 255.255.255.0"	
defaultrouter="172.30.254.1"	
gateway_enable="YES"	IP 転送を有効
keymap="jp.106"	コンソール接続時のキーマップ
sshd_enable="YES"	
sendmail_enable="NONE"	
usbd_enable="YES"	
ipfilter_enable="YES"	
ipfilter_rules="/etc/ipf.rules"	
ipfilter_flags=""	
ipmon_enable="YES"	
ipmon_flags="-P /var/run/ipmon.pid -D /var/log/ipf.log"	
ipnat_enable="YES"	
ipnat_rules="/etc/ipnat.rules"	

/etc/defaults/rc.conf には全ての設定パラメータがあり、そのパラメータは、/etc/rc.conf によって上書きされます。/etc/defaults/rc.conf は直接変更せず、/etc/rc.conf または /etc/rc.conf.local に固有の設定を行うべきでしょう。

B. ログの監査

本サーバは、各拠点から PPP 接続を確立したのち、障害対応等の踏み台として利用されます。その際に、5W1H に基づいたポリシとして、ログの取得を可能な限り行います。

本章では、主要なログとその監査内容について記述しています。

B.1. PPP 接続時の監査

本サーバは、PPP over SSH で VPN を実装しているため、最初に ssh 接続の成否が /var/log/auth.log へ出力されます。

```
# cat /var/log/auth.log
  ... sshd[69196]: Accepted publickey for ppposssh from 211.1.83.235 port
4636 ssh2
  ... sshd[70955]: Illegal user anonymous from 210.86.143.194
          ...
```

本サーバへ PPP 接続を確立した各拠点のログは、/var/log/ppp.log へ出力されます。
弊社で管理対象としている仮想ネットワークの IP アドレスや、indent 文字列等が正規クライアントのものであるか確認できます。

```
$ cat /var/log/ppp.log
  ... ppp[69203]: Phase: Using interface: tun0
  ... ppp[69203]: Phase: deflink: Created in closed state
  ... ppp[69203]: tun0: Command: pppossh_approve_4_HOMENetwork: set escape
0xff
  ... ppp[69203]: tun0: Command: pppossh_approve_4_HOMENetwork: disable
ipv6cp
          ...
  ... ppp[69203]: tun0: LCP:  MAGICNUM 804044b9
  ... ppp[69203]: tun0: LCP:  TEXT HOME Network
  ... ppp[69203]: tun0: CCP:  DEFLATE[4] win 15
  ... ppp[69203]: tun0: IPCP: IPADDR[6] 10.224.1.2
          ...
  ... ppp[69203]: tun0: IPCP: IPADDR[6] 10.224.1.1
  ... ppp[69203]: tun0: IPCP: deflink: LayerUp.
  ... ppp[69203]: tun0: IPCP: myaddr 10.224.1.1 hisaddr = 10.224.1.2
```

B.2. シリアルコンソール接続時の監査

cu コマンドはデフォルトの/var/log/aculog ファイルに接続情報を出力します。

```
# cat /var/log/auth.log
          ...
fs (Mon Apr 18 22:42:50 2005) <cu9600, , /dev/ucom0> call completed
fs (Mon Apr 18 22:44:34 2005) <cu9600, , /dev/ucom0> call terminated
          ...
```

/dev/ucom0、/dev/ucom1 ... とそれぞれのアクセスラインがどのノードにシリアル接続されているかを管理する必要があります。

B.3. 踏み台利用時の監査

本サーバは、ipf ポリシに基づいて、パケットに関するログを/var/log/ipf.log へ出力されます。

telnet 接続した場合の例として、HOME Network のクライアント(192.168.1.215)から Labs Network 上のルータ(192.158.253.1)への通信情報を以下に示します。

```
$ cat /var/log/ipf.log
          ...
HOME Network から仮想ネットワーク上の tun0 から SYN パケットが通過した。
  ... tun0 @300:2 p 192.168.1.215,3936 -> 192.168.253.1,23 PR tcp len 20
48 -S K-S K-F IN

Labs Network(192.168.253.0/24)上の em0 へ SYN パケットが NAPT され、通過した。
  ... em0 @300:2 p 192.168.253.4,55360 -> 192.168.253.1,23 PR tcp len 20
48 -S K-S K-F OUT

192.168.253.1 から SYN+ACK が戻ってきた。
  ... em0 @300:2 p 192.168.253.1,23 -> 192.168.1.215,3936 PR tcp len 20 44
-AS K-S K-F IN

仮想ネットワーク上の tun0 へ SYN+ACK パケットが戻る。
  ... tun0 @300:2 p 192.168.253.1,23 -> 192.168.1.215,3936 PR tcp len 20
44 -AS K-S K-F OUT

仮想ネットワーク上の tun0 から ACK パケットが通過した。
  ... tun0 @300:2 p 192.168.1.215,3936 -> 192.168.253.1,23 PR tcp len 20
40 -A K-S K-F IN

Labs Network(192.168.253.0/24)上の em0 から ACK パケットが NAPT され、通過した。
  ... em0 @300:2 p 192.168.253.4,55360 -> 192.168.253.1,23 PR tcp len 20
40 -A K-S K-F OUT

192.168.253.1 から ACK+PSH が戻ってきた。
  ... em0 @300:2 p 192.168.253.1,23 -> 192.168.1.215,3936 PR tcp len 20 55
-AP K-S K-F IN

以降、両端で telnet セッションが確立され、通信が行われる。
  ... tun0 @300:2 p 192.168.253.1,23 -> 192.168.1.215,3936 PR tcp len 20
55 -AP K-S K-F OUT
  ... tun0 @300:2 p 192.168.1.215,3936 -> 192.168.253.4,23 PR tcp len 20
55 -AP K-S K-F IN
  ... em0 @300:2 p 192.168.253.4,55360 -> 192.168.253.1,23 PR tcp len 20
55 -AP K-S K-F OUT
          ...
```

クライアントの IP アドレスまでは把握できますが、人物、通信の開始･終了時間、および通信内容は特定することは困難か不可能です。

VPN connectivity environment

改版履歴

Version 1.0　　2005/04/13　　新規作成。

製作

Isidore

本書は 2005 年 5 月現在の情報を元に作成されております。本書に記載されております内容は、許可なく変更されることがあります。