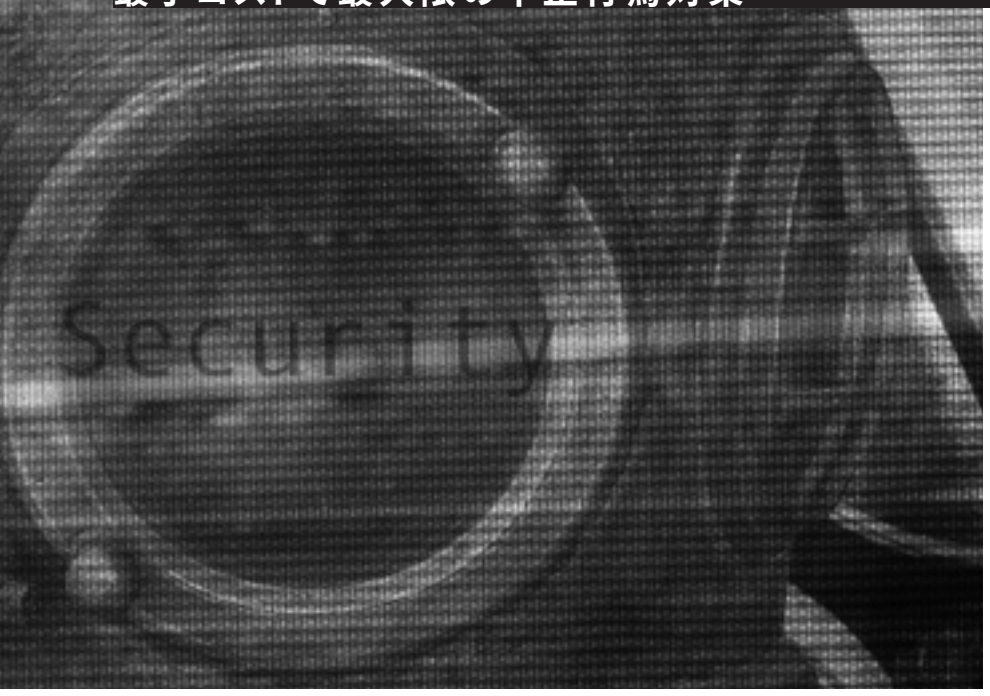
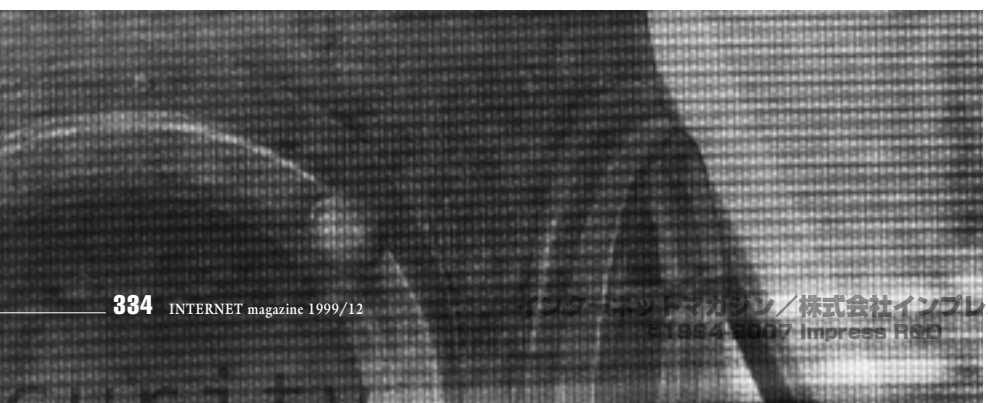


実践 Linux セキュリティー講座

今回は暗号技術を使って通信経路を保護するリモートシェルSSH（Secure SHell）に関して説明します。これまで紹介してきたツールとは違い、SSHはLinuxの標準ディストリビューションのパッケージには含まれていません。そのため、SSHを使うには別途、開発環境を持ったLinuxマシンを用意して自分自身の手でコンパイルし、サーバーマシンにインストールする必要があります。今回はインストール方法も紹介します。

第11回 リモートシェルSSHの使い方

ソフトウェアコンサルタント　すずきひろのぶ



SSHとはどんなものか？

rloginやrshはリモートホストへログインしたり、あるいはリモートでプログラムを実行したりするのに使われるコマンド類です。Linuxユーザーにとっては身近なコマンドで、日頃からよく使われていると思います。ポピュラーである一方、これらのコマンドが認証などに対して甘い面を持っていることはあまり認識されていません。

また、rloginやtelnetのようにリモートログインをするツールでは、どうしてもネットワーク越しにパスワードを入力する必要がありますが、パスワードが平文の形でネットワークを流れるのはとても危険なことです。

このような問題を解決するのがSSHです。

小規模SOHOで自分（あるいは非常に信用がおける利用者）以外に利用者はいないし、外部からのアクセスはtcp_wrapperなどでrloginやtelnetを保護しているので大丈夫だと言われる方も多いかと思います。それはそれで一理あるでしょう。しかし、今すぐ使わないとしても、セキュリティーの周辺知識としてSSHを知っているのといないのでは大きな違いがあると思います。

SSHの種類

入手できるSSHは大きく分けて以下の3つになります。

- SSHバージョン1
- SSHバージョン2
- LSH

SSHバージョン1はssh-1.2.27が（この原稿を書いている時点では）最新版です。このバージョン1系列は新たな開発をストップしたバージョンで、もっとも安定しています。ときどき、特定OS固有のセキュリティーホールが発見されたときに修正が施され、リビジョン番号が上がるぐらいです。

SSHバージョン2は開発中のバージョンでssh-2.0.13が（この原稿を書いている時点で

は）最新版です。今後はこちらのバージョンが中心となって改良されていきます。

これらのソースファイルは次のURLから入手できます。

ftp://ftp.cs.hut.fi/pub/ssh/

また国内のringサーバープロジェクト Jump01 でもミラーコピーされているので、そちらからも入手可能です。

SSHのディレクトリー

ftp://ring.etl.go.jp/pub/net/ssh/

ssh-1.2.27もssh-2.0.13も、個人的な使用もしくは学術的な目的以外では利用できないというライセンス条項が入っています。キチンとライセンスの注意書きをチェックしてください。

残りの1つはGPLライセンス下での実装であるLSHです。原稿を書いている時点での最新版はlsh-0.1.12です。こちらのほうは開発が始まって間もないので、非常に短いスパンで改良が続けられています。ただ、まだ登場して時間がたっていないだけに、十分にバグが取れていないおそれがあります。認証などに新しいメカニズムを採用しているので、SSHに慣れている人でも戸惑うかもしれません。また、利用ドキュメントも充実しているとは言えません。このため、LSHを試すのはもう少し様子を見てからでも遅くはないでしょう。最新版は Jump02 から入手できます。

Jump01 ring.etl.go.jp

Jump02 www.lysator.liu.se/~nisse/archive/

SSH1をコンパイルする

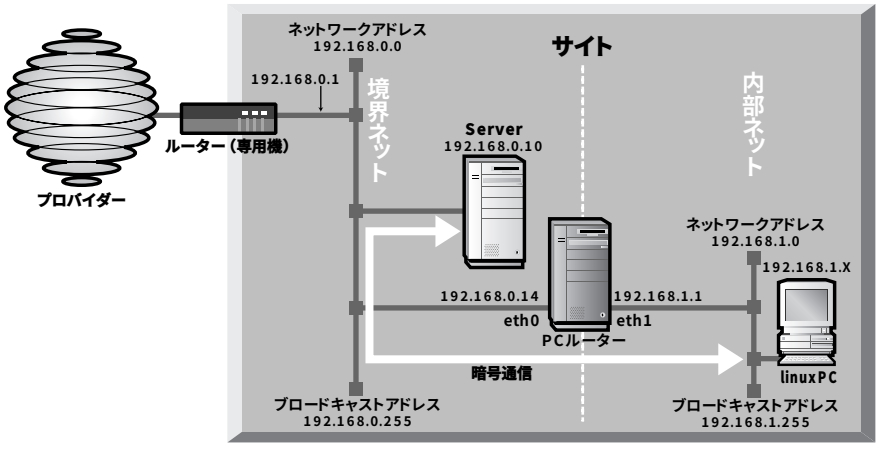
今回はもっともポピュラーなSSHバージョン1（以下、SSH1）を利用します。日本語のドキュメントも豊富です。下記のURL Jump03 が参考になるでしょう。

開発環境を用意しているLinuxマシン（以下、開発環境マシン）に、ftpサーバーからssh-1.2.27.tar.gzをダウンロードしてきたというところから話を始めます（注意：本連載での前提としてセキュリティー上の問題からサーバーマシンには開発環境がインストールされていません）。

今回のインストールでは、専用のディレクトリー/usr/local/ssh1を使います。そして、まずは開発環境マシンにSSH1をインストールし、それをコピーする形でサーバーにインストールします。なお、開発環境マシン（マシン名：linuxpc）は接続の際のクライアントとしてテストに使用します。

Jump03 www.vacia.is.tohoku.ac.jp/~s-yamane/FAQ/ssh

図❶　サイトのネットワーク構成図



開発環境マシン上へのインストール

まずは開発環境マシンでのコンパイルとインストールの手順を説明します。ssh-1.2.27.tar.gzファイルを展開した後、コンフィグレーションを実行させ、コンパイルを行って、開発環境マシン上にインストールします（リスト1、Step 1〜Step 5）。

次にインストールしたSSH1をターゲットであるサーバーにコピーします。手順は、まず開発環境マシンの/usr/local/ssh1をtarでアーカイブしてサーバー（マシン名：server）へftpで転送します。次に/etc/以下にある2つのSSH1用のコンフィグレーションをftpで転送します（リスト1、Step 6〜Step 8）。

コピーが終わったところでサーバーマシンにrootでログインし、必要なファイルを必要な場所へ展開、コピーします（リスト1、Step 9〜Step 11）。

SSH1の鍵を作る

次にSSH1で使用するサーバーの鍵ペアのファイルを作ります。開発環境マシン上では、make installのとき/etc/ssh_host_keyと/etc/ssh_host_key.pubの中にRSAの秘密鍵と公開鍵が自動的に作られます。これらは後ほどマシンの自己証明などに使います。

一方、サーバー上ではインストール作業をマニュアルで行っているので、この鍵ペアを作成する作業も自分自身の手で行わなければなりません（リスト1、Step 12）。

sshdサーバーの動作確認

続いて、サーバー上で/usr/local/ssh1/sbin/sshdを実行します。以降、sshdがデーモンとして動作します（リスト1、Step 13）。

クライアントから接続できるか？

開発環境マシンに戻ってサーバーにコマンドsshでログインできるかどうかをチェックしてみます（リスト1、Step 14）。

最初のログインのときに「Host key not found from the list of known hosts.」（既知ホストのリストからはホストの鍵がみつからない）と警告が出た後、SSHは「Are you sure you want to continue connecting (yes/no)?」（接続を続行するか yes/no）と尋ねてきます。yesと答えると、リストにマシン“ server ”が追加されます。

マシン“ server ”のsshdは先ほど作成したホスト固有の（秘密鍵とペアになっている）公開鍵を持っています。SSHはサーバーに接続するときに、サーバー側から認証のための署名付きデータを受け取ります。クライアントのSSHはサーバーの公開鍵を使って、接続先が正しいかどうかチェックします。

SSHを使えば「なりすまし」を防げる

コンピュータのIPネットワーク接続はIPアドレスを区別することによって相手に接続していますが、これを偽るのは非常に簡単です。送受信するIPパケットを途中で横取りするなどといった高度な技術を使った方法など必要ありません。ネットワーク接続しているマシンのネットワークケーブルを抜き、別の偽マシンにつなぎ替えるだけで十分です。もちろんSOHOでは目の届く所にマシンがあるのでこのような心配はさほどありません。しかし、企業や大学のように何十台、何百台という数でマシンが稼働していて、接続先サーバーがいったいどこに置かれているのかなど皆目見当もつかないような場合には、ネットワークケーブルを付け替えるといった単純な方法で「サーバーのなりすまし」が可能です。

公開鍵による署名を使って相手を確認すれば、このような「なりすまし」を防げます。

リスト❶　SSHをサーバーにインストールする手順

開発環境マシンでのコンパイルとインストール

Step 1: ssh-1.2.27.tar.gzをtarで展開する。

```
% tar zxf ssh-1.2.27.tar.gz
```

Step 2: カレントディレクトリーをssh-1.2.27に移動する。

```
% cd ssh-1.2.27
```

Step 3: ./configureを実行する（2〜5分ほどかかります）。

```
% ./configure --prefix=/usr/local/ssh1
creating cache ./config.cache
....
creating mpz/tests/Makefile
```

Step 4: makeを実行する（数分から十数分ほどかかります）。

```
% make
gcc -pipe -c -I.  -I./gmp-2.0.2-ssh-2 -I./zlib-1.0.4 ....
......
gcc -pipe  -o ssh-askpass ssh-askpass.o ... -L/usr/local/lib  -lutil
```

Step 5: 開発環境マシン上にインストールする。

```
% su
Password: <----------------------------------rootのパスワードを入力
# make install
umask 022; if test '!' -d /usr/local/ssh1; then \
  mkdir /usr/local/ssh1; fi; \
....
 /usr/local/ssh1/man/man8/`echo $p | sed 's,x,x,'`.8; fi;\
done
```

インストールしたSSHのサーバーへのコピー

Step 6: /usr/local/ssh1をtarコマンドでアーカイブする。

```
# tar zcvfP /tmp/ssh1.tar.gz /usr/local/ssh1
/usr/local/ssh1/
....
/usr/local/ssh1/man/man8/sshd.8
```

Step 7: 一般ユーザーに戻りサーバーマシンへftpする。

（デフォルトではROOTでのftpアクセスを許可していないため）

```
% cd /tmp
% ftp server
Name (server:hironobu):
331 Password required for hironobu.
....
ftp> cd /tmp
250 CWD command successful.
ftp> put ssh1.tar.gz
....
858139 bytes sent in 2.77 secs (3e+02 Kbytes/sec)
ftp> quit
```

Step 8: 同様の方法で/etc/sshd_configと/etc/ssh_configをftpする。

```
% cd /etc
% ftp server
ftp> cd /tmp
...
ftp> put sshd_config
...
ftp> put ssh_config
...
```

開発環境マシンからコピーしたファイルをサーバー上で展開、コピー

Step 9: サーバーマシンにROOTでログインする。

Step 10: tarを展開する。

```
# cd /tmp
# tar zxvfPp ssh1.tar.gz
/usr/local/ssh1/
...
/usr/local/ssh1/man/man8/sshd.8
#
```

Step 11: /tmp/ssh_configと/tmp/sshd_configを/etcへコピーする。

```
# cp /tmp/ssh_config /etc
# cp /tmp/sshd_config /etc
```

サーバーのための鍵のペアの作成

Step 12:

```
# cd /usr/local/ssh1/sbin
# ./ssh-keygen -b 1024 -f /etc/ssh_host_key  -N ''
Initializing random number generator...
Your identification has been saved in /etc/ssh_host_key.
Your public key is:
1024 35 157170088744161609633107125982884676703671170930404949 00
...
99050539663604754781 6183 root@server.h2np.net
Your public key has been saved in /etc/ssh_host_key.pub
```

サーバー上でsshdを実行

Step 13:

```
# /usr/local/ssh1/sbin/sshd
```

開発環境マシンからサーバーへsshを使ってログイン

Step 14:

```
% /usr/local/ssh1/bin/ssh server
Host key not found from the list of known hosts.
Are you sure you want to continue connecting (yes/no)? yes <-yesと入力
Host 'server' added to the list of known hosts.
Creating random seed file ~/.ssh/random_seed.  This may take a while.
hironobu@server's password:  <-------------------------パスワード入力
Last login: Mon Oct  4 17:22:48 1999 from linuxpc
No mail.
%
```

そして、公開鍵を使用してサーバーが正しいものであるかどうかを一度確認できれば、そのサーバー側の公開鍵が更新されない（破棄されない）限り、この認証は有効になります。

サーバーの公開鍵はデフォルトでは~/.ssh/known_hostsに格納されます。同時に同じディレクトリーにrandom_seedという名前の乱数用初期値ファイルも作成されます。

ここまでで、基本的なインストールは成功です。rloginやtelnetに比べ、より安全なSSHを使ってサーバーにログインできるようになりました。

次回はSSHの続きを

サーバーにSSHをインストールできたところで誌面が尽きてしまいましたので、来月はSSHの使い方の続きを紹介します。

予定では、ブート時に自動的にsshdを立ち上げるスクリプトの書き方、ユーザー自身の公開鍵を作って認証に使う方法、SSHを使ってパスワードなしでサーバーアクセスを行う方法、さらには、SSHエージェントの使い方などを紹介します。

その後、誌面が許せば、インターネット側からのVPN（Virtual Private Network）のようにアクセスする方法や、ファイアウォールを経由してのIPフォワーディングに関しても説明するつもりです。

［インターネットマガジン　バックナンバーアーカイブ］ ご利用上の注意

このPDFファイルは、株式会社インプレスR&D（株式会社インプレスから分割）が1994年～2006年まで発行した月刊誌『インターネットマガジン』の誌面をPDF化し、「インターネットマガジン　バックナンバーアーカイブ」として以下のウェブサイト「All-in-One INTERNET magazine 2.0」で公開しているものです。

http://i.impressRD.jp/bn

このファイルをご利用いただくにあたり、下記の注意事項を必ずお読みください。

- 記載されている内容（技術解説、URL、団体・企業名、商品名、価格、プレゼント募集、アンケートなど）は発行当時のものです。
- 収録されている内容は著作権法上の保護を受けています。著作権はそれぞれの記事の著作者（執筆者、写真の撮影者、イラストの作成者、編集部など）が保持しています。
- 著作者から許諾が得られなかった著作物は収録されていない場合があります。
- このファイルやその内容を改変したり、商用を目的として再利用することはできません。あくまで個人や企業の非商用利用での閲覧、複製、送信に限られます。
- 収録されている内容を何らかの媒体に引用としてご利用する際は、出典として媒体名および月号、該当ページ番号、発行元（株式会社インプレスR&D）、コピーライトなどの情報をご明記ください。
- オリジナルの雑誌の発行時点では、株式会社インプレス R&D（当時は株式会社インプレス）と著作権者は内容が正確なものであるように最大限に努めましたが、すべての情報が完全に正確であることは保証できません。このファイルの内容に起因する直接的および間接的な損害に対して、一切の責任を負いません。お客様個人の責任においてご利用ください。

このファイルに関するお問い合わせ先
株式会社インプレスR&D
All-in-One INTERNET magazine 編集部
im-info@impress.co.jp