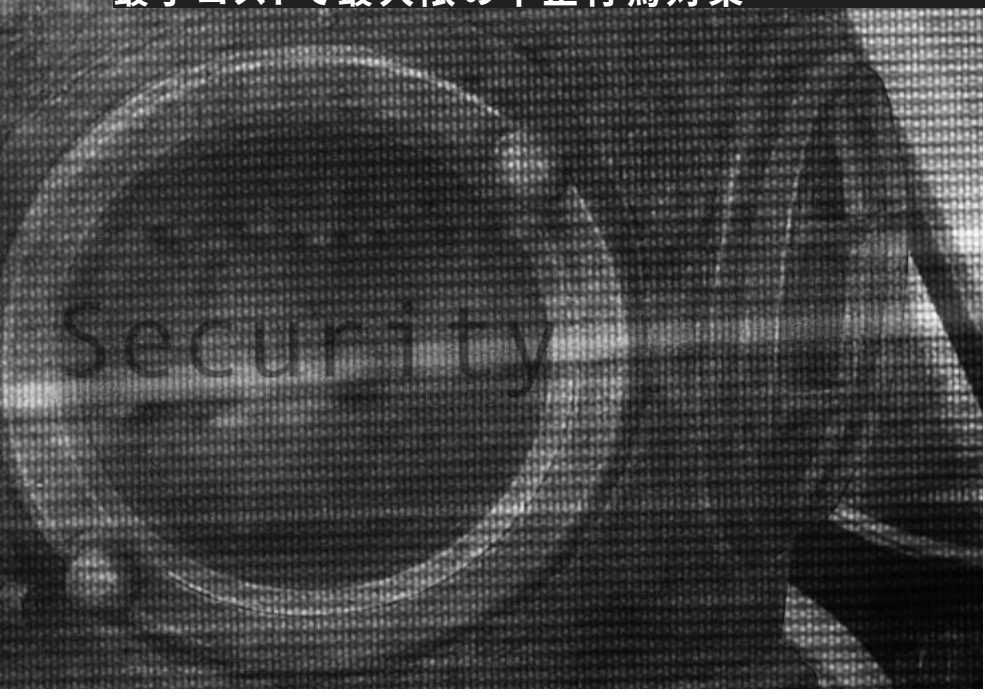


実践 Linux セキュリティー講座

前回に引き続き、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」を紹介します。前回は、SSHをインストールして、手作業で動かすまでを説明しました。今回はブート時に自動的にsshdを立ち上げる方法から始めたいと思います。その後にエージェントを使う方法やSSHをVPN(Virtual Private Network)のようにして使う方法などを説明します。

第12回 SSHの高度な使い方

ソフトウェアコンサルタント　すずきひろのぶ

sshdの自動立ち上げ

ブート時に自動的にsshd（SSHのサーバー）を立ち上げるには、リスト❶のような小さなスクリプトを書き、実行ファイル/etc/rc.d/init.d/sshとしてセーブします。

自分の鍵のペアを作る

先月はサーバーの鍵のペアについて説明しましたが、ユーザーも同じように自分自身の鍵のペアを作成し、認証に使用できます。まずは作成してみましょう（リスト❷）。

これで公開鍵が~/.ssh/identity.pubにセットされ、同時にペアである秘密鍵が~/.ssh/identityにセットされます。

内容を誰にも漏らさないために秘密鍵~/.ssh/identityは暗号化されており、先ほど入力したパスフレーズは秘密鍵を利用するために使われています。

パスフレーズはパスワードと同じ意味で使われ方も同じですが、最近は入力できる文字数が多いので「ワード」（語）という呼び方ではなく「フレーズ」（語句）という表現をする場合が多くなっています。

あとは作成した~/.ssh/identity.pubを相手マシン環境に設定すれば、通常のパスワード認証ではなく公開鍵の認証を使ってアクセスすることが可能になります。

ただし相手マシンでのidentity.pubが、ほかのマシンでファイル共有されていないことが前提です。ほかのマシン上でファイル内容が改竄されてしまう危険を避ける必要があるからです。

パスワードなしでサーバーにアクセス

「パスワードなし」と言っても、正確にはユーザー自身の公開鍵を使って認証を行い、サーバーにアクセスすることを意味しています。また、サーバーにアクセスするためのパスワードは必要ありませんが、公開鍵とペアになっ

ている秘密鍵を利用するために、一度だけパスフレーズが必要となります。

さて事前の用意です。先ほど作成した公開鍵をサーバー上の自分のログイン環境に登録します。この登録がすめば、登録された公開鍵によって接続時に自動的に認証が行われます（リスト❸、Step 1）。

これで公開鍵がサーバーのログイン環境に登録されました。以降、自動的に公開鍵の認証による接続方式になります。次からSSHでログインするときは、サーバーへのログインパスワードではなく、先ほど秘密鍵を利用するために設定したパスフレーズを使うことになります（リスト❸、Step 2）。

最大64文字のパスフレーズを利用できますので、通常のパスワード（8文字）よりも安心です。もちろん、容易に推測できないパスフレーズを使って初めて意味があります。

エージェントを使って より簡単に

今度は、パスフレーズをいちいち入力するのではなく、あらかじめ設定しておく方法です。

クライアント側からサーバーに接続していない状態でSSHエージェントを立ち上げ、その環境下で秘密鍵をエージェントに登録します。これ以降、エージェントが自動的にコマンドsshに対して秘密鍵を与えるので、いちいちパスフレーズを入力する必要がなくなります（リスト❹、Step 1〜2）

このエージェント環境にいるあいだは、パスフレーズを必要とせずに、すぐにサーバーにログインできます（リスト❹、Step 3）。

パスフレーズ入力が面倒になり簡単なものにしてしまうような利用者側の問題を解決するよい方法でしょう。

VPNのように使う

ここまではSSHをリモートシェルとして使ってきましたが、SSHはIPパケットをフォワード（転送）する機能を持っています。つま

リスト❶　sshdの自動立ち上げ用スクリプト

```
#!/bin/sh
# sshd

SERVER=/usr/local/sbin/sshd
[ -f $SERVER ] || exit 0

case "$1" in
  start)
          echo -n "Starting sshd: "
          touch /var/lock/subsys/sshd
          daemon $SERVER
          echo


          ;;
  stop)
          echo -n "Shutting down sshd: "
          killproc sshd
          rm -f /var/lock/subsys/sshd
          echo sshd
          ;;
  status)
          status sshd
          ;;
  restart)
          $0 stop
          $0 start
          ;;
  *)
          echo "Usage: sshd {start|stop|restart|status}"
          exit 1
esac

exit 0
```

リスト❷　自分の鍵のペアを作る手順

・ssh-keygenを実行して鍵のペアを作成する

```
% /usr/local/ssh1/bin/ssh-keygen
Initializing random number generator...
 ....
Enter file in which to save the key (/home/hironobu/.ssh/identity):  ← そのまま改行
Enter passphrase:  ← パスフレーズを入力
Enter the same passphrase again:  ← 再度パスフレーズを入力
Your identification has been saved in /home/hironobu/.ssh/identity.
Your public key is:
1024 37 175950342905967179908932391459....
.........................8607 hironobu@linuxpc.h2np.net
Your public key has been saved in /home/hironobu/.ssh/identity.pub
```

リスト❸　公開鍵を使ってサーバーにログインする手順

Step 1: サーバーの~/.ssh/authorized_keysに登録する

```
% /usr/local/ssh1/bin/ssh server 'cat >> ~/.ssh/authorized_keys' \
< ~/.ssh/identity.pub
hironobu@server's password:
```

← Serverのログインパスワード

Step 2: 公開鍵認証を使ってサーバーにログインする

```
% /usr/local/ssh1/bin/ssh  server
Enter passphrase for RSA key 'hironobu@linuxpc.h2np.net':
Last login: Mon Oct  4 20:58:02 1999 from linuxpc
No mail.
```

← パスフレーズ

リスト❹　エージェントを使ってサーバーにログインする手順

Step 1: /usr/local/ssh1/bin/ssh-agentを使い新しいシェル環境に入る

```
% /usr/local/ssh1/bin/ssh-agent bash
%
```

← エージェント環境で使うシェル

← ここはすでにエージェント環境に入っている

Step 2: /usr/local/ssh1/bin/ssh-addを使い秘密鍵をエージェントに登録する

```
% /usr/local/ssh1/bin/ssh-add
Need passphrase for /Users/hironobu/.ssh/identity \
(hironobu@linupc.h2np.net).
Enter passphrase:
Identity added: /Users/hironobu/.ssh/identity......
```

← パスフレーズを入力

Step 3:

```
% /usr/local/ssh1/bin/ssh  server
Last login: Mon Oct  4 21:33:02 1999 from linuxpc
No mail.
%
```

← serverへ既にログインしている

り、SSHを使って一種のVPN（Virtual Private Network）が実現できるのです。

ログイン＋αで利用できるので非常に便利です。大抵のことはできてしまいますが、上手に使うにはネットワークの仕組みをよく知っている必要があるかもしれません。

このような機能はSOHO環境のセキュリティーを高めるという目標を超えたレベルかもしれません。しかし、一応、知識として知っておいても決して損にはなりません。また、このようなバックグラウンドの知識を持つことによって、ネットワークセキュリティーに関してより深い理解が得られると思います。

X11パケットのフォワーディング

X11フォワーディングとは、SSHを使ってX11のプロトコルをトンネルさせ、遠隔のマシン同士で安全にX11を使う機能です。リモートマシンで実行されるXアプリケーションと手元のXサーバー間で通信されるXプロトコル（ここではXで使われるIPパケット）が暗号化されます。これにより、リモートログインの場合のメリットと同様、内容の盗聴、通信の乗っ取りなど、データが見えることによって発生する数々の攻撃を回避できます。

一番簡単な設定方法は、Xサーバーを立ち上げ、操作している状態で、シェルウィンドウからリモートマシンにSSHでログインしてしまうことです。この状態でXに必要なことは自動的にセットアップされます。次に、リモートマシン上でXのアプリケーションを実行すると自動的にXサーバー上に現れます。

たとえば、毎回リモートにログインせずに一度SSHでログインした後は、Xのアプリケーション（xtermやEmacsなど）を立ち上げて複数の作業を同時に行うようなことが可能になります。

ポートフォワーディング

ポートフォワーディングも、先に説明した

X11プロトコルのフォワーディングと原理は同じで、TCPのセッション（通信）をSSHを使って暗号化します。ただし、こちらの場合は任意のポート番号に対するフォワードのみを行います。sshdをファイアーウォールマシンの上で動作させると一種のプロキシーの役目を果たします。

右の例（図❶、リスト❺）は、インターネット上の任意の場所にあるクライアントから中継マシン（server.h2np.net）まで、3456ポートをSSHで保護して通信をトンネル化し、中継マシンのsshd（SSHのサーバー）を経由して目的のマシンの23ポート（TELNET）に接続しています。

ここでは分かりやすいようにTELNETを利用していますが、ほかのポートでも同じことで、通信を行うほとんどのコマンドに有効です。

（注意）ただし、マシンserverからlinuxpcへはTELNETでアクセスできることを前提にしています。また、本連載のポリシーでは境界ネットワークからリモートシェルなどを使って内部ネットワークにアクセスするのは基本的に禁止ということになっています。

ポートフォワーディングの応用

ポートフォワーディングの応用として、WWWでもIPパケットのフォワーディングが可能です。内部ネットワーク側のマシン「linuxpc」でウェブサーバーが実行されているとしましょう。サイト内であれば、どこからでもウェブサーバーにアクセスできる条件であっても、内部のIPアドレスはローカルアドレスであるため外部であるインターネット側からは直接はアクセスできません。その場合、ファイアーウォール外のクライアントからSSHを経由して、内部のウェブサーバーにアクセスすることも可能です。たとえば出張に出ていて、社内イントラネットの社内向けウェブサーバーにアクセスする場合など、この方法が使えます（リスト❻）。

次回はウェブサーバーについて

本連載も段々と佳境に入ってきました。次回はウェブサーバーについて説明します。

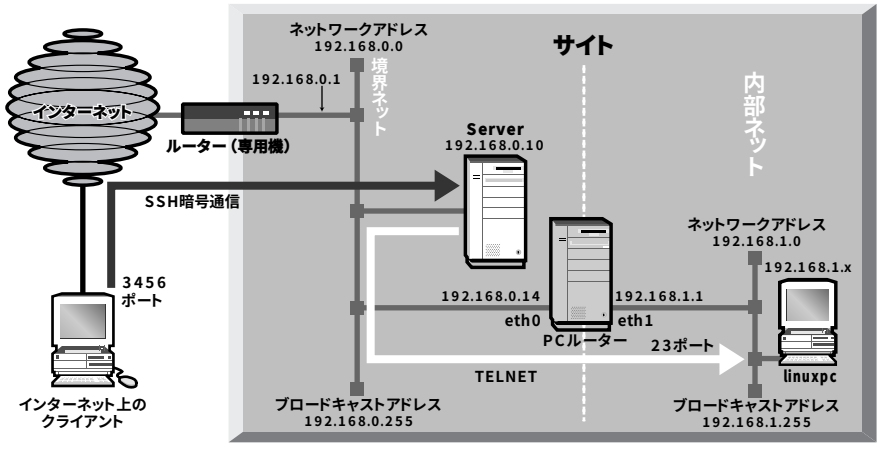
リスト❺　インターネット上のクライアントから内部ネットのlinuxpcにtelnetする

```
% ssh -L 3456:linuxpc:23 server.h2np.net
                    これで通常のSSHでのログインと同じ要領で一度server.h2np.netへログインする

・インターネット上のクライアントからlinuxpcへ接続する

% telnet localhost 3456
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

Welcome Linux (linuxpc)

login:
                    実際はインターネット上のクライアントからの接続だが、server.h2np.netが
                    中継しており、linuxpcではserver.h2np.netからのtelnet接続に見えてい
```

リスト❻　インターネット上のクライアントから内部ネットのウェブサーバーにアクセスする

```
% ssh -l username  -L 3456:linuxpc:80 server.h2np.net
 ....

・ファイアウォール外のクライアントマシン上で以下のURLをアクセスする
http://localhost:3456
```

図❶　サイトのネットワーク構成図



RPM形式ですでに用意されているApacheをそのまま利用する形になります。

残念ながら、連載ではすべてのウェブサーバーのセキュリティー技術をカバーすることはできません。そこでWebサーバーのセキュリティーに関する話題をポイントを押さえた形で述べていきたいと思います。

［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