SSH の使い方 (2)

– OpenSSH への移行について –

赤坂 浩一 *　　平野 彰雄 *　　沢田 篤史 *

1　はじめに

本センターでは、暗号技術を用いた安全な通信方式である **SSH**(Secure SHell) を UNIX 系システム (sakura,spp,vpp) で導入してきました。

これまで導入していた SSH は、バージョン 1 の SSH(SSH Protocol Version 1) でしたが、既にセキュリティ上の問題が指摘されており、より強度の高い暗号方式を用いるバージョン 2 の SSH(SSH Protocol Version 2) をサポートする OpenSSH の導入を検討し、平成 13 年 4 月より UNIX 系システムでサービスを開始しています。

SSH は、フィンランドのタトゥ・ウロネン氏によって開発され、当初はライセンスがあまり厳しくないものでしたが、後に設立された SSH Communications Security 社でライセンスを管理されるようになってからは、少しライセンスが厳しくなりました。なお、大学の教育機関 (アカデミック) や個人が非営利目的で使用することは許されているようです。

また、SSH プロトコルを実現するソフトウェアとして OpenBSD グループによって開発された OpenSSH があり、こちらは目的に制限なく利用することができ、今回、本センターではこの OpenSSH を導入しました。

OpenSSH は、バージョン 1 の SSH とバージョン 2 の SSH の二種類をサポートしていますので、これまで通りの接続も可能ですが、先に述べましたようにセキュリティ上の問題が指摘されていますので、今後はバージョン 2 の SSH での利用を推奨していきます。

本稿では、SSH Communications Security 社のバージョン 2 の SSH を **SSH2** と呼ぶことにし、OpenSSH のバージョン 2 の SSH を **OpenSSH2** と呼ぶことにします。

それでは、バージョン 2 の SSH を利用した本センターの UNIX 系システムの使い方を簡単に説明します。今回は、UNIX 系 OS からの利用方法を紹介し、Windows パソコンからの利用方法は次回に紹介します。また、SSH2 と OpenSSH2 とでは、利用する鍵ファイルのフォーマットが若干異なっているために相互に利用ができない問題があります。相互に利用するためには鍵ファイルのフォーマット変換を行なう必要がありますので、こちらについても簡単に紹介します。

なお、ここでは SSH の仕組み等の説明は行ないませんので、前回の解説記事である本センター広報「SSH の使い方 – より安全にネットワークを利用するために –」 (Vol.32 No.5) を参照してください。

2　OpenSSH2 での利用方法

ここでは、お使いのワークステーション (WS) に OpenSSH をインストールし、OpenSSH2 で利用する方法を説明します。

2.1　ソースコードの入手

OpenSSH は、最近の OpenBSD や FreeBSD・Linux のオペレーティングシステム (OS) に組み込まれていますので、そのままでも利用可能ですが、最新の OpenSSH を利用したい場合は、自分でインストールする必要があります。また、Solaris など他の OS をお使いの方は自分でインストールする必要がありますので、以下の OpenSSH のホームページから、ソースコードを入手してください。

`http://www.openssh.com/`

* あかさか ひろかず , ひらの あきお , さわだ あつし (京都大学大型計算機センター)

また、OpenSSH を利用するためには、この他に Zlib と OpenSSL の二つが必要になりますので、お使いの WS にインストールされていない場合は、あらかじめインストールしておく必要があります。

それぞれ、以下のホームページより入手してください。

- Zlib：`http://www.freesoftware.com/pub/infozip/zlib/`
- OpenSSL：`http://www.openssl.org/`

本稿執筆中のそれぞれの最新バージョンは、OpenSSH が `openssh-2.5.1p2.tar.gz` で、Zlib が `zlib.tar.gz` のバージョン 1.1.3、OpenSSL が `openssl-0.9.6.tar.gz` となっています。

2.2 インストール

OpenSSH をインストールする前に、先に Zlib と OpenSSL をインストールします。

それぞれ、ファイル一式がアーカイブされていますので、`tar` や `gzip` コマンドを使って展開します。

ディレクトリが作成され、ファイルが展開されていますので、そのディレクトリに移動し、`INSTALL` や `README` などのファイルからインストール方法を知ることができ、`configure` や `config` スクリプトを実行すると `Makefile` を自動的に生成してくれます。あとは、`make` コマンド実行すれば、コンパイルできます。

手順としては、次のようになります。

```
% gzip -dc アーカイブファイル | tar -xvf -
% cd 作成されたディレクトリ
% ./configure または ./config
%  make
```

コンパイルが正常に終了したら、root になってインストールします。

```
#  make install
```

必ず OpenSSH よりも先に Zlib と OpenSSL をインストールするようにしてください。

また、既にバージョン 1 の SSH をインストールされている場合は、先にアンインストールしておくのが良いかもしれません。特にお使いの WS で SSH のサーバを起動させている場合は、ホストの鍵ファイルや設定ファイルのインストール先が変更されているので注意が必要です。

OpenSSH のインストールが終わると次のようなプログラムおよびユーティリティが利用できるようになります。

クライアントプログラム	`ssh,slogin,scp,sftp`
サーバプログラム	`sshd,sftp-server`
その他ユーティリティ	`ssh-keygen,ssh-agent,ssh-add,ssh-keyscan`

特に指定しない限り、`sshd` は `/usr/local/sbin`、`sftp-server` は `/usr/local/libexec`、それ以外は `/usr/local/bin` ディレクトリにインストールされています。また、`/usr/local/etc` ディレクトリに設定ファイルやホストの鍵ファイルがインストールされています。

2.3 設定ファイルのカスタマイズ

OpenSSH は、バージョン 1 の SSH とバージョン 2 の SSH の両方をサポートしていますが、クライアントプログラムはバージョン 1 の SSH を優先するようになっていますので、より強度の高い暗号方式を用いるバージョン 2 の SSH を優先するように設定ファイルを変更します。

クライアントの設定ファイルは、上記のディレクトリ配下に `ssh_config` ファイルで置かれ、デフォルトの設定が定義されています。

この `ssh_config` ファイルの中に次のような行がありますので、`#` のコメントを外し、`1,2` を `2,1` に書き換えます。

```
#   Protocol 1,2
        ↓
    Protocol 2,1
```

これで、バージョン 2 の SSH を優先するようになります。`2,1` と記述することで、先にバージョン 2 で試し、それが失敗した場合にバージョン 1 での接続を試みます。

サーバの設定ファイル (`sshd_config`) では、デフォルトの設定では、X11 のポートフォワーディングが無効となっていますので、お使いの WS で有効としたい場合は、`sshd_config` ファイルの中の次の行で、`no` を `yes` に書き換えてください。

```
X11Forwarding no
      ↓
X11Forwarding yes
```

なお、クライアントもデフォルトでは無効となっていますので、 ssh_config ファイルの中に次のような行がありますので、 # のコメントを外し、 no を yes に書き換えます。

```
#   ForwardX11 no
      ↓
    ForwardX11 yes
```

なお、クライアントの設定は、ユーザのホームディレクトリに~/.ssh/ssh_config ファイルを用意して設定することができ、こちらの内容が優先されます。

設定ファイルについて詳しく知りたい方は、ssh(1) および sshd(8) のマニュアルを参照してください。

2.4 OpenSSH2 の認証方法

OpenSSH では、バージョン 1 では RSA アルゴリズムによる認証を行ないますが、バージョン 2 では DSA アルゴリズムによる認証を行ないます。

OpenSSH2 でパスワードを入力する代わりにパスフレーズを入力して接続する場合は、 DSA アルゴリズムで認証用の鍵ファイルを生成する必要があります。

認証用の鍵ファイルを生成する ssh-keygen コマンドはデフォルトでは、バージョン 1 で用いる RSA アルゴリズムで鍵ファイルを生成しますので、コマンドに次のようなオプション (-t dsa) を指定して実行してください。

```
%  ssh-keygen -t dsa
```

実行すると鍵ファイルの名前を問い合わせますが、そのままで良いでしょう。次に、設定するパスフレーズを二度入力すると、 DSA アルゴリズムの鍵ファイルが生成されます。秘密鍵は、~/.ssh/id_dsa に、公開鍵は、~/.ssh/id_dsa.pub に作成されます。

ここで作成された公開鍵を接続先となるホスト計算機 (sakura,spp,vpp) にコピーします。

バージョン 1 の RSA 認証用鍵ファイルは、ホスト計算機の~/.ssh/authorized_keys に書き込んでいましたが、バージョン 2 の DSA 認証用鍵ファイルは、~/.ssh/authorized_keys2 に書き込まなければなりません。

例では、 sakura の~/.ssh/authorized_keys2 に scp コマンドを使ってコピーしています。

なお、紙面の関係上 “ → ” で折り曲げています。

```
%  scp ~/.ssh/id_dsa.pub →
 userid@sakura:~/.ssh/authorized_keys2
```

複数の計算機から接続する場合は、それぞれの公開鍵を適当な名前で送り、~/.ssh/authorized_keys2 ファイルに追記します。例えば、他の計算機の公開鍵を other-host.pub と言う名前で、 sakura に送った場合は、次のように cat コマンドを使って追記します。

```
sakura%  cat other-host.pub >>  →
                ~/.ssh/authorized_keys2
```

これで、パスワードの代わりにパスフレーズを入力して接続できるようになります。

2.5 sftp の使い方

クライアントコマンドの ssh,slogin,scp は、基本的にバージョン 1 と同じように利用することができます。ここでは、バージョン 2 で新たにサポートされた sftp について紹介します。

通常、 ftp によるファイル転送には、コントロールポートとデータポートの二つのポートを利用しています。

ユーザ名とパスワードが流れるコントロールポートは、 SSH のポートフォワーディング機能を利用して安全な通信経路で通信を行なうことができますが、実際にファイルの中身が流れるデータポートは接続の度にポート番号が変わるので、 SSH のポートフォワーディング機能が利用できません。

そこで、この問題が解決するために、 sftp がサポートされました。使い方は、基本的に ftp コマンドと同じで、ファイルの転送には get や put コマンドを使いますが、バイナリー (binary) やアスキー (ascii) と言った転送モードはありません。また、リモート計算機のユーザ名を指定したい場合は、 sftp コマンド起動時に指定する必要があります。

```
%  sftp remote-host

%  sftp user@remote-host
```

3 SSH2 からの利用方法

ここでは、お使いのワークステーション (WS) に SSH Communications Security 社が提供するバージョン 2 の SSH をインストールし、SSH2 で利用する方法を説明します。

3.1 ライセンスに関して

SSH2 は大学などや個人が非営利目的で使用することは許可されていますが、企業や政府機関での使用にはライセンスの購入が必要です。

なお、京都大学では KUINS(京都大学学術情報ネットワーク機構) が、京都大学における非営利目的に限定した SSH2 のサイトライセンスを取得しています。

3.2 ソースコードの入手

以下の SSH2 のホームページから入手することができます。

`http://www.ssh.com/`

京都大学に所属されている利用者の方は、KUINS のホームページから入手することができますので、こちらを利用ください。

`http://www.kuins.kyoto-u.ac.jp/news/33/ssh.html`

本稿執筆中の最新バージョンは、SSH2(Secure Shell Ver.2.4.0) です。

3.3 インストール

SSH2 は必要なファイル一式がアーカイブされていますので、`tar` や `gzip` コマンドを使い展開し、その後、`configure` スクリプトを実行し `Makefile` を生成して、`make` コマンド実行すれば、コンパイルできます。

手順としては、次のようになります。

```
% gzip -dc ssh-2.4.0.tar.gz | tar -xvf -
% cd ssh-2.4.0
% ./configure
%  make
```

コンパイルが正常に終了したら、root になってインストールします。

```
#  make install
```

SSH2 はバージョン 1 の SSH を認識してくれるので、既にバージョン 1 の SSH をインストールしていても気にせずインストールして大丈夫です。

特に指定しない限り、サーバは `/usr/local/sbin` クライアントは `/usr/local/bin` ディレクトリにインストールされます。また、`/etc/ssh2` ディレクトリに設定ファイルやホストの鍵ファイルがインストールされています。

お使いの WS に合わせて設定ファイルは、必要に応じてカスタマイズしてください。

3.4 SSH2 での使い方

SSH2 も基本的にバージョン 1 の SSH と大差なく使えます。既にバージョン 1 の SSH がインストールされている環境に SSH2 をインストールした場合は、通常、バージョン 2 の SSH が優先され、接続先の計算機がバージョン 1 の SSH しか導入されていない場合は、自動的にバージョン 1 の SSH で接続を試みます。これは、お使いの WS がサーバとなるケースでも同じです。

バージョン 1 の SSH と違っている点は、認証用の鍵ファイルの置き場所と指定方法です。

SSH2 では、`~/.ssh2` ディレクトリ配下に鍵ファイルを置くことになっています。`ssh-keygen` コマンドを実行すると、DSA アルゴリズムで鍵ファイルが生成され、`~/.ssh2/id_dsa_1024_a`(秘密鍵) `~/.ssh2/id_dsa_1024_a.pub`(公開鍵) が作成されます。

また、他ホストの公開鍵の指定方法は、バージョン 1 の SSH の場合、`authorized_keys` ファイルに公開鍵の中身を書き込んでいましたが、SSH2 では、公開鍵ファイルを`~/.ssh2` に適当な名前で置き、`authorization` と言う名前のファイルを作成し、その中に次のような形式で他ホストの公開鍵を指定します。

```
Key    他ホストの公開鍵ファイル名
                 :
Key    他ホストの公開鍵ファイル名
```

複数のホストの公開鍵を指定する場合は、改行して同じような形式で追記していきます。

また、先にも述べましたが、OpenSSH2 と SSH2 では鍵ファイルのフォーマットが若干異なっている

ので、 sakura,spp,vpp を SSH2 で利用する場合はフォーマット変換が必要となります。

4 OpenSSH2 と SSH2 の相互運用

OpenSSH2 と SSH2 とでは、同じ DSA アルゴリズムで認証用の鍵ファイルを生成しますが、フォーマットが若干異なっていますので、このままでは、相互に認証用鍵ファイルを利用した接続ができません。ここでは、 SSH2 のクライアントから OpenSSH2 のサーバに接続する場合と OpenSSH2 のクライアントから SSH2 のサーバに接続するための鍵ファイルのフォーマット変換について紹介します。

なお、認証用の鍵ファイルを利用せず、通常のパスワードによる接続を行なう場合は、ここでの作業を省くことができます。

4.1 SSH2 から OpenSSH2 に接続する場合

SSH2 のクライアントから OpenSSH2 のサーバに接続する場合ですが、本センターの UNIX 系システム (sakura,spp,vpp) では、 OpenSSH を導入しましたので、お使いの WS に SSH2 をインストールされた場合は、こちらの手順で鍵ファイルのフォーマット変換を行なってください。

フォーマット変換は、 OpenSSH2 の `ssh-keygen` コマンドで行ないますので、まず、SSH2 のクライアントで生成した公開鍵ファイル (`id_dsa_1024_a.pub`) を OpenSSH2 のサーバに適当な名前でコピーします。

```
%  scp ~/.ssh2/id_dsa_1024_a.pub →
       userid@sakura:~/s2clnt.pub
```

次に、 OpenSSH2 のサーバ (sakura) にログインし、コピーした公開鍵ファイル (`s2clnt.pub`) を `ssh-keygen` コマンドにオプション (`-X`) を指定して OpenSSH2 形式にフォーマット変換します。

この `ssh-keygen -X` を実行すると標準出力に OpenSSH2 形式の公開鍵として出力されますので、これを`~/.ssh/authorized_keys2` ファイルに追記します。

```
sakura%  ssh-keygen -X -f s2clnt.pub →
           >> ~/.ssh/authorized_keys2
```

フォーマット変換が終われば、 `s2clnt.pub` ファイルは不要ですので削除しておいてください。

4.2 OpenSSH2 から SSH2 に接続する場合

OpenSSH2 のクライアントから SSH2 のサーバに接続する場合ですが、 sakura などの OpenSSH2 からお使いの WS に SSH2 のサーバをインストールされた場合は、こちらの手順で鍵ファイルのフォーマット変換を行なってください。

ここでも OpenSSH2 の `ssh-keygen` コマンドでフォーマット変換を行ないます。 OpenSSH2 で生成した秘密鍵ファイル (`id_dsa`) を `ssh-keygen` コマンドにオプション (`-x`) を指定して SSH2 形式のフォーマット変換します。

```
sakura%  ssh-keygen -x   →
   -f ~/.ssh/id_dsa  >  o2clnt.pub
```

フォーマット変換時に、 OpenSSH2 で生成したときの入力したパスフレーズの問合わせがあるので、正しいパスフレーズを入力すると SSH2 形式の公開鍵ファイルが `o2clnt.pub` に出力されます。

この公開鍵ファイルをお使いの WS の`~/.ssh2` ディレクトリ配下にコピーし、`~/.ssh2/authorization` ファイルの中で指定します。この例では、お使いの WS にログインしてから行なっています。

```
%  scp userid@sakura:~/o2clnt.pub →
          ~/.ssh2/o2clnt.pub
%  echo "Key o2clnt.pub" >> →
                  ~/.ssh2/authorization
```

なお、 sakura 上の SSH2 形式に変換した公開鍵ファイルは、上記の作業が完了すれば不要ですので削除しておいてください。

5 おわりに

今回、本センターに導入した OpenSSH を UNIX 系 OS から利用する方法について紹介しましたが、次回は、 Windows パソコンから利用する方法について紹介したいと思っています。

なお、最新情報は下記の URL で提供を考えていますので、本稿と合わせてご覧ください。

`http://www.kudpc.kyoto-u.ac.jp/Services/SSH/`

この記事に関して、ご意見・ご質問などございましたら、プログラム相談室までご連絡ください。