SSH : Secure Shell (2)

～ 小荷物を(packet) 秘かに港(port) で 横流し(forwarding) ～

伊東栄典* 池田大輔**

前回の記事 [1] で，SSH の概要，導入方法，利用方法について説明しました．SSH を用いる事で telnet 通信を，盗聴を防止できる暗号化通信に置き換える事が可能です．しかし，telnet 以外にも FTP・電子メール等の様々な通信があります．これらの通信内容も保護しなければ，安全性が十分だとは言えません．SSH による認証と暗号化を用いて他の通信を保護するために，ポート転送 (Port Forwarding) という手法があります．本稿では SSH によるポート転送について説明します．

1 ポート

インターネットに接続された計算機同士が通信を行なう場合，通信相手の特定に IP アドレスを用います．これは電話で会話をする際，通信相手の特定に電話番号を用いるのに似ています．現在の IP アドレスは 32bit の整数値で，133.5.9.1 といった表記を行ないます．通信相手となる計算機の特定には，IP アドレスの他に kyu-cc.cc.kyushu-u.ac.jp というような計算機名を用いることも可能です．このような計算機名は人間にとって憶えやすいように付けられている名前です．計算機内部の通信では，DNS(Domain Name System) などを用いて名前を IP アドレスに変換して通信が行なわれます．

IP アドレスと電話番号は類似していると述べましたが，インターネットの通信と電話には大きな違いがあります．それはインターネット上の計算機間通信の場合，複数の通信を同時に行なう事ができるという事です．電話の場合，ある人が A さんと話している最中，別の B さんと通話することはできません．つまり一時に一つの通話しか行なえません (ISDN では事情が異なります)．これに対してインターネットの通信では，マルチタスク OS の計算機を用いれば，telnet で遠隔利用しつつ，WWW ページデータを入手しつつ，FTP でファイルを転送し，かつ chat プログラムで文字会話をする，といった事を同時に行なう事ができます．

インターネットで同時に複数の通信を行なう場合，計算機が受けとった通信データがどのプログラムのデータなのかをどのように区別しているのでしょうか．IP で使われている TCP [1]および UDP [2]を用いた通信では，同時に行なわれる複数の通信を区別するために，ポート番号を用いています．ポート番号は 16bit の整数値で表現されます．つまりポート番号は 0 から 65535 ($= 2^{16} - 1$) の間の値を取ることができます．

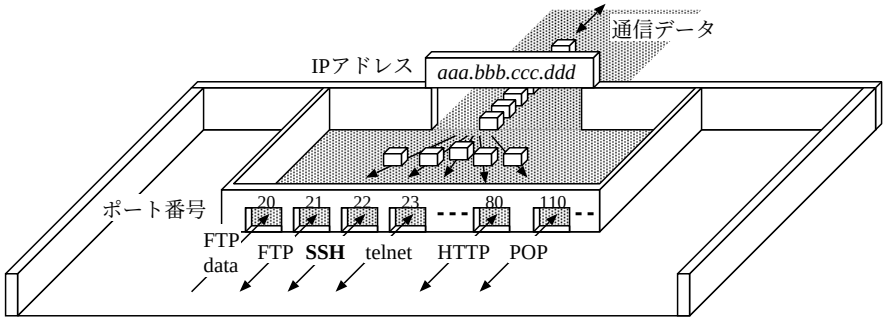

図 1: IP アドレスとポート

ポートは約 65000 個存在しますが，すべてのポートを自由に利用できるわけではありません．図 1 に示してい

*九州大学大型計算機センター
E-mail:itou@cc.kyushu-u.ac.jp, http://www.cc.kyushu-u.ac.jp/RD/itou/
**E-mail:daisuke@cc.kyushu-u.ac.jp,
[1]Transfer Control Protocol. データの到達を保証するプロトコル.
[2]User Datagram Protocol. データの到達を保証しないプロトコル.

るように，小さな番号のポートは，20,21 番ポートは FTP，23 番は telnet，25 番は SMTP(電子メールの転送)，80 番は HTTP(WWW データ通信) といった風に用途が決められています．これらの用途が知られているポートは「well-known ポート」と呼ばれます [2]．UNIX 系の OS では “/etc/services” というファイルに，あるポート番号がどのような通信サービスを行なっているのかが記述されています．

telnet の場合，サーバ側は 23 番ポートで通信を待ち受けます．多くの場合，telnet クライアントは自動的に OS が割り当てる大きめのポート番号を使います．UNIX 系 OS の場合，1024 番以上の番号を用いている場合が多いようです．SSH サーバは 22 番ポートで待ち受けています．SSH のクライアントとなる計算機は，遠隔地にあるサーバ計算機の 22 番ポートへ接続します．二者の間で暗号鍵交換等の交渉が行なわれた後，通信が確立するのです．現在，インターネット上で最も利用されている通信サービスである WWW は HTTP という通信方式を用いています．WWW サーバが待ち受けているポートは 80 番です．Internet Explorer や Netscape といった WWW クライアント (ブラウザ) は WWW サーバとなる計算機の 80 番ポートへ接続しているのです．

20 番ポートと 21 番ポートは FTP のために用いられます．普通は，telnet や HTTP のように，ある一つの通信では一つのポートだけを用いるのですが，FTP は 2 つのポートを用います．21 番ポートは FTP サーバ側の待ち受け口ポート番号です．もう一方の 20 番ポートは，FTP サーバがクライアントへデータを送り出すためのデータ送信専用ポート番号です．

ポートの接続状況は netstat コマンドで知る事ができます．ここでは Windows95/98 の MS-DOS プロンプト上で利用できる netstat コマンド例に示します．図 2 の下線部は，前回の記事で説明した TTSSH を用いて kyu-cc.cc.kyushu-u.ac.jp に SSH の接続している様子を示しています．手元の計算機 (firebird.cc.kyushu-u.ac.jp) の 748 番ポートと，遠隔地の計算機である kyu-cc.cc.kyushu-u.ac.jp の 22 番ポートが接続されている事がわかります．

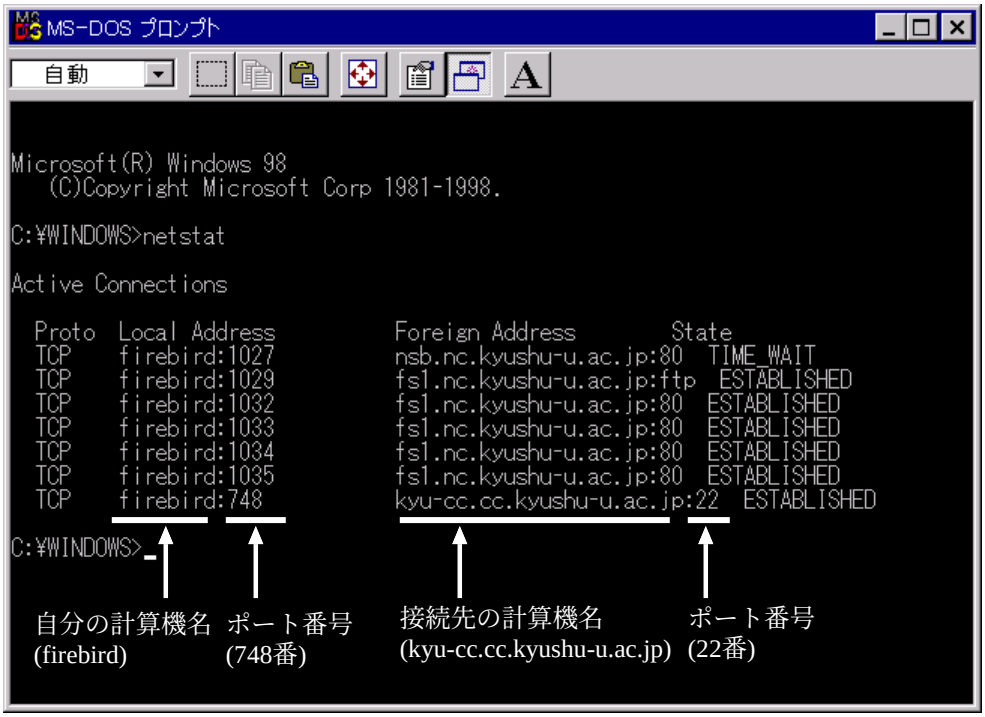

図 2: ポート番号の利用状況 (netstat コマンド)

2　ポート転送による安全な通信

前回の記事 [1] で解説したように，SSH を用いる事で telnet や rlogin コマンドを用いずに計算機を遠隔利用する事が可能です．SSH の暗号化機能により，利用開始時における利用者認証の情報を外部に知られる可能性が低くなります．しかしながら，telnet 以外にもセキュリティを考慮していない通信が存在します．電子メールの

POP(Post Office Protcol) やファイル転送用の FTP(File Transfer Protocol) などはセキュリティを考慮していません．これらの通信も暗号化しなければ，安全性の確保は片手落となります．

POP は携帯型計算機や家庭での電子メールの利用に使われている単純な電子メール受信プロトコルです．POP ではメール受信する際にユーザ名とパスワードで利用者認証を行ないます．その際，認証情報であるユーザ名とパスワード，それから電子メールの本文は，利用者の計算機とサーバとの間を平文で送信されます．ファイル転送を行なう FTP でも，利用者認証情報であるパスワードは平文で送信されます．これらのパスワードは簡単に盗聴される危険性があります．

POP や FTP と同様に，様々な通信の全てに認証情報や通信内容を暗号化し安全に情報を交換する方法が提供されているわけではありません．また，提供されていたとしても，サーバ側の設定の変更や別のサーバが必要な場合は，一般ユーザでは自由に導入できません．そのような場合のために，暗号化しない安全でない通信を，SSH を用いて暗号化した安全な通信路を介して通信を行なう「ポート転送」という方法があります．この記事では POP を例に用い，SSH によるポート転送について，その概念と利用方法を説明します．

2.1 POP3 : Post Office Protocol version 3

POP について簡単に説明します．POP(Post Office Protocol) は電子メールを受信するためのプロトコルです．POP には version1 と version2 も存在しますが，多くの場合，3 番目の version である POP version3 が利用されています．POP の verion3 なので POP3 と表記されます．POP3 の詳細については RFC[3, 4, 5] 等の文献を参照して下さい．Netscape messenger, MS-Outlook, ポストペットなどのほとんどの電子メール用ソフトウェアが POP3 に対応しています．クライアントからの要求を受け付ける POP3 サーバソフトとしては，UNIX 系 OS 上で動作する qpopper などが在ります．以下，POP3 の事を単に POP と表記します．

九州大学大型計算機センターでは，汎用計算機 kyu-cc.cc.kyushu-u.ac.jp とライブラリサーバ wisdom.cc.kyushu-u.ac.jp の 2 台の計算機で POP サーバを提供しています．この二つの計算機宛に届けられたメールは，POP クライアントを用いる事でメールの受信を行なう事ができます．

図 3 に POP における通信の様子を示します．POP は良く使われているサービスであるため，110 番という well-known ポートを用いる事が決められています．POP サーバは 110 番ポートで通信を待ち受けており．クライアントからの接続要求が来たら，サーバはユーザ名とパスワードの対による認証を行ないつつ，メールの送信等の操作を行ないます．POP クライアントプログラムを起動してサーバに接続する場合，クライアント側では，あるポート番号 *iiii* を用いて通信を行ないます．ポート番号 *iiii* は，クライアント側計算機の OS が適切に自動割当てしてくれます．

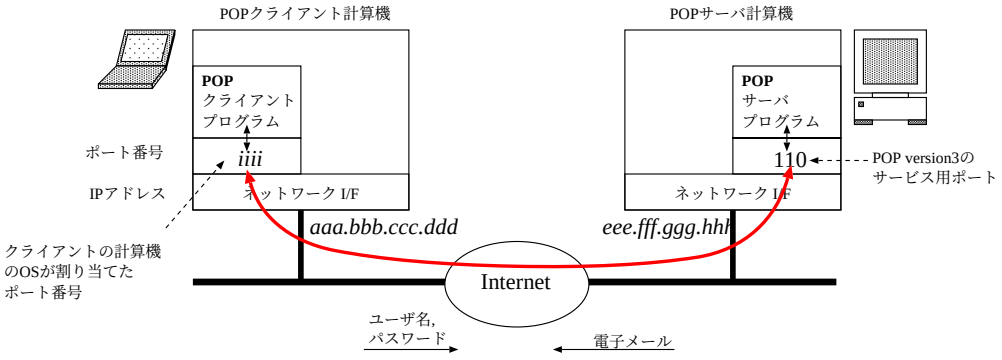

図 3: POP の概念図

POP では利用者の認証にユーザ名とパスワードを用いています．この利用者情報は平文で送信されます．このため，悪意を持ったネットワークに少し詳しい人が存在すれば，簡単にパスワードを盗聴する事ができます．盗聴したパスワードを利用すれば，その利用者になりすましてサーバ計算機に接続できます．POP は自宅や出張先などでも使う事が多いと思います．そのような場合，信用のおけないネットワークを経由して，自分のメール

サーバに接続する場合も多いでしょう．どこかでパスワードが盗聴されているかもしれないという不安を感じないでしょうか?

2.2 SSH のポート転送

SSH を用いると，遠隔地間にある 2 つの計算機間に安全な通信路を確立する事ができます．これは遠隔地間の通信を暗号化して行なうために，悪意がある者が途中経路で通信を盗聴していたとしても，内容を理解することが不可能である，という事実に因ります．この SSH の通信路を用いて，POP や FTP などの安全でない通信を行なうのが，ポート転送です．

ポート転送には，ローカル転送とリモート転送の二種類があります．ローカル転送は，手元の計算機のポートに出入りするデータを，接続先の計算機に転送するものです．リモート転送は，接続先の計算機のあるポートに出入りするデータを接続元の計算機へ転送し，手元の計算機上で提供されているサービスを遠隔地の計算機から利用する場合に利用します．サービスを受ける計算機が異なるだけで，どちらもほぼ同様に使うことができます．ローカル転送の方が良く利用されると思われますので，以後のポート転送は特に断らない限りローカル転送の事とします．

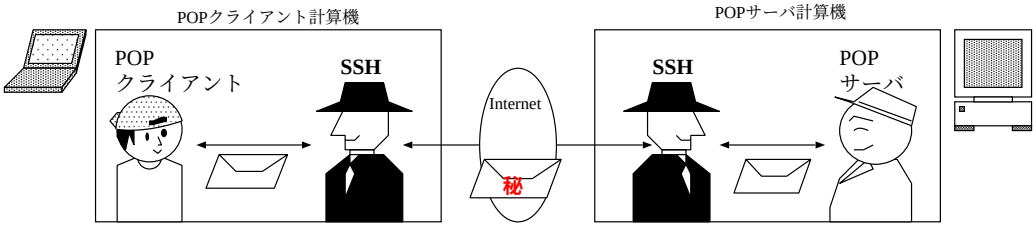

図 4: SSH と POP

POP を例にして SSH によるポート転送を説明します．図 4 にその概念図を示します．先に示した図 3 のように POP クライアントとサーバが直接会話するのではなく，別の SSH という暗号語を喋る通訳とか代理人を介して情報をやりとりする，という風に考えるとわかりやすいかもしれません．POP クライアントから見ると，SSH は POP サーバの代理人に見え，POP サーバからは SSH が POP クライアントのように見えます．

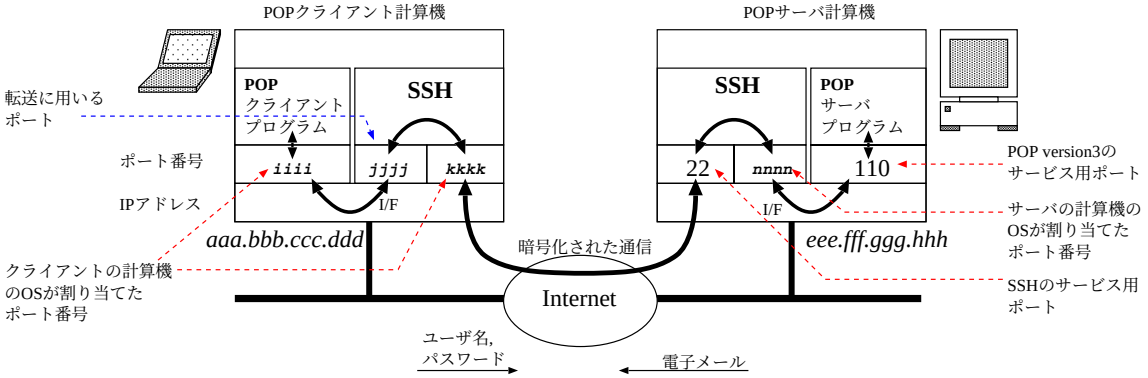

図 5: ポート転送を用いた POP

図 5 に，より詳細な関係図を示します．POP クライアント側の計算機では，POP クライアントを起動する前に，ユーザは転送に利用するポート番号 jjjj を指定して予め SSH を起動しておきます．その後 POP クライアントを「自分自身の計算機の jjjj 番ポートに対して接続する」ように起動します．このためには POP クライアントプログラムが，接続先のポート番号を設定可能でなければなりません．

クライアント側の jjjj 番ポートは，SSH がポート転送を行なうものとして予め起動されていますので，POP クライアントからの接続を受けたクライアント側 SSH は，サーバ側の SSH へデータを暗号化して転送します．この時クライアント側の SSH は kkkk 番ポートから，POP サーバ側の計算機の SSH(22 番ポート) へデータを送ります．kkkk 番ポートは，OS が SSH に自動的に割り当てた番号です．

サーバ側の SSH は，クライアント側計算機からの通信を解読して POP サーバに送ります．その際，内部に nnnn 番ポートを作成し，そのポートを通じて POP サーバと通信します．nnnn 番ポートも，OS を介して SSH に自動的に割り当てられる番号です．POP サーバは，サーバ側内部にある SSH を POP クライアントと見なして処理を行ないます．メールの送信のような POP クライアントへの応答は，サーバ側の SSH へ送ります．

POP クライアントは POP サーバと通信するように自分自身の計算機内部にある SSH と通信します．逆に POP サーバは POP クライアントと通信するように，サーバ側計算機内部にある SSH と通信するのです．このようにプログラムの通信情報を，SSH が代理人となって暗号化通信を行なうのがポート転送です．

3　UNIX 上でのポート転送

この節では UNIX 系の OS 上で，実際にポート転送をする方法について説明します．Windows95/98 におけるポート転送については 4 節を参照して下さい．UNIX 系 OS への SSH インストール方法については，前回の記事 [1] を参照して下さい．遠隔地の計算機に届けられている電子メールを POP で受信することと，FTP によるファイルの転送をを例に説明します．

3.1　ポート転送コマンド

SSH によるポート転送を用いるには，応用プログラムの実行前にあらかじめ SSH を起動しておく必要があります．ポート転送のための SSH の起動方法は，接続元の計算機のポート番号と，接続先の計算機である host とそのポート番号 hostport を指定して，以下のように ssh コマンドを実行します．

ssh -L*port*:*host*:*hostport*　*host*

“-L” はローカル転送のためのオプションです．“*port*” の所には転送に用いるポート番号を記述します．“*host*” にはサービスを利用したい計算機名を，“*hostport*” にはそのサービスが使用するポート番号を指定します．例えば，POP 場合は *hostport* を 110 に，FTP の場合は *hostport* を 21 に指定します．サービスとポート番号の対応については，UNIX 系の OS では “/etc/services” ファイルに記述されています．

このコマンドを実行しますと，*host* で指定した計算機へ SSH を用いてログインする場合と同様の認証が行なわれます．そのためポート転送を利用するためには計算機 host にログインできなければいけません．認証作業が終了すると *host* へログインされ，安全な通信路が確立されたことになります．このポート転送のための通信路は *host* にログインしている間にのみ存在するので，**転送したいクライアントの実行を終了するまで，ログアウトせずにいる必要があります**．

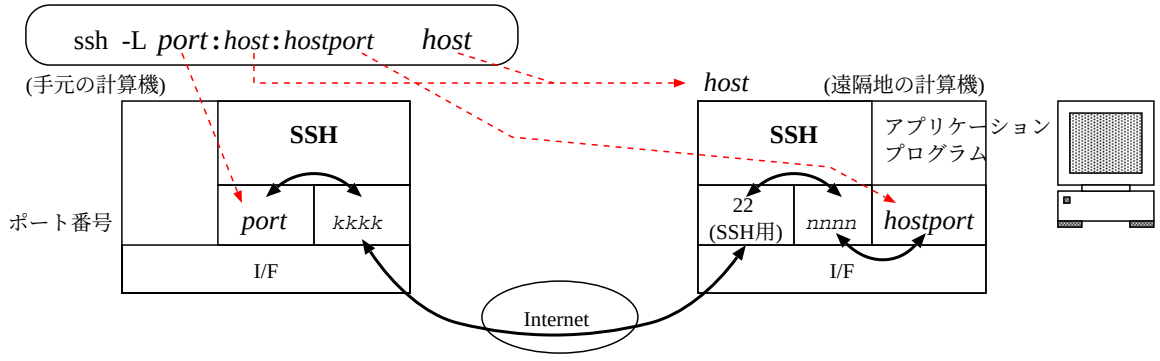

図 6: SSH のポート転送コマンド

3.2 ポート転送例 (POP)

九州大学大型計算機センターのライブラリサーバ (wisdom.cc.kyushu-u.ac.jp) に届いているメールを，自分が利用している計算機 (myhost) へ POP を用いて取得する，という場合の実行例を示します．ここで九州大学大型計算機センターでのユーザ名 (利用課題番号) は “a70099a” とします．myhost 側でポート転送に用いる一時的なポート番号は 40000 とします．40000 という大きな番号にしているのは，UNIX 系 OS の場合小さな番号 (1024 以下) のポートは OS が管理しているので一般ユーザ権限では利用できないからです[3]．ポート転送のためのコマンドは，以下のようになります．

```
myhost % ssh  -l a70099a  -L40000:wisdom.cc.kyushu-u.ac.jp:110  wisdom.cc.kyushu-u.ac.jp ↵
```

“myhost %” の部分はコマンド入力プロンプトです．“-l a70099a” という部分が増えているのは，接続先計算機でのユーザ名の指定です．

上記のコマンド入力後，パスフレーズの入力あるいは wisdom のパスワード入力といった認証作業が行われます．wisdom へログインすれば，ポート転送の準備終了です．POP 対応した電子メール用ソフトで，接続先を自分自身 (localhost) の 40000 番ポートに指定して接続すれば，SSH のポート転送を用いた POP 接続ができます．

```
 % telnet localhost 40000 ↵
Trying 127.0.0.1...
Connected to wisdom.cc.kyushu-u.ac.jp.
Escape character is '^]'.
+OK YAT server ready (version 3.12p.2 at wisdom)
quit ↵
```

図 7: telnet による転送の確認

図 7 は，telnet コマンドによって localhost の 40000 番ポートに接続する様子を示しています．1 行目は自分自身 (127.0.0.1) に接続しようとしています．しかし次の行では wisdom へ接続し，POP サーバプログラムである YAT server からのからの応答があったことが分かります．この時のポート番号の様子を図 8 に示します．

myhost(接続元) の netstat (OS:FreeBSD3.2)

```
tcp    0      0  myhost.1009            wisdom.ssh             ESTABLISHED
tcp    0      0  localhost.40000        localhost.1403         ESTABLISHED
tcp    0      0  localhost.1403         localhost.40000        ESTABLISHED
tcp    0      0  localhost.40000        *.*                    LISTEN
```

wisdom(接続先) 側の netstat (OS:Solaris2.4)

```
wisdom.22      myhost.mydomain.ac.jp.1009 17520      0  8760      0 ESTABLISHED
wisdom.56713   wisdom.pop            8192      0  8192      0 ESTABLISHED
wisdom.pop     wisdom.56713          8192      0  8192      0 ESTABLISHED
```

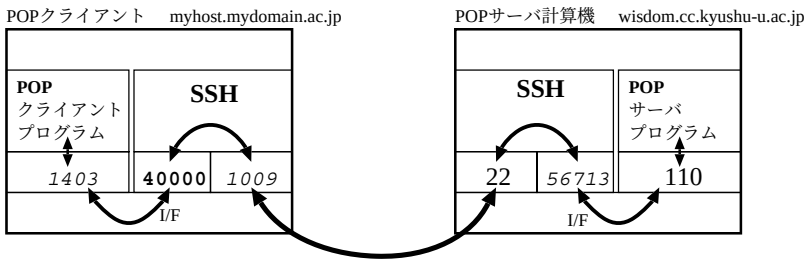

図 8: netstat によるポート転送の様子

[3]Windows95/98 ではこの制限はありません．

Mew の設定方法

UNIX 上でメールの受信を行なうソフトウェアは多数あるため，全てについて説明することはできません．ここでは，メール用ソフトウェアである Mew で、POP によるメールの受信を行なう場合について説明します．Mew については，Mew 公式 WWW ページ (`http://www.mew.org/`) や広報記事 [6] などを参照して下さい．

Mew のバージョン 1.9.*ではメールの入手などにコマンド群 IM を用いており，IM の中の imget コマンドによりメールの入手を行ないます．IM のコマンドの一つである imget で，SSH のポート転送を使って POP によりメールを入手するには次のように入力します．

```
myws % imget –src=pop/POP@localhost/40000 ↵
```

Mew の中から，POP で利用する場合には IM の設定ファイルである$HOME/.im/Config ファイルに次の記述を加えます．

```
Imget.Src=pop
PopAccount=POP@localhost/40000
```

4 Windows95/98 上でのポート転送

前回の記事 [1] で，Windows95/98 から SSH を利用するためのソフトウェアとして，TTSSH[7] と Tera Term[8] を解説しました．TTSSH は遠隔ログイン以外に，ポート転送にも用いる事ができます．TTSSH と Tera Term のインストールについては，前回の記事を参照して下さい．本節では，TTSSH を用いたポート転送について説明します．

UNIX でのポート転送方法解説と同様に，この節でも POP を例に取り上げ，遠隔地の計算機に届けられている電子メールを POP で受信することを例に説明します．POP サーバは九州大学大型計算機センターのライブラリサーバ wisdom.cc.kyushu-u.ac.jp，大型計算機センターのユーザ名 (利用課題番号) は “a70099a” とします．

4.1 TTSSH の準備

SSH のポート転送を用いて wisdom からメールを取得するという例で，ポート転送の方法を説明します．最初に TTSSH をポート転送に使うための設定をします．TTSSH を起動すると計算機への接続画面が出ますが，[Cancel] ボタンを押して非接続状態で TTSSH を起動します．

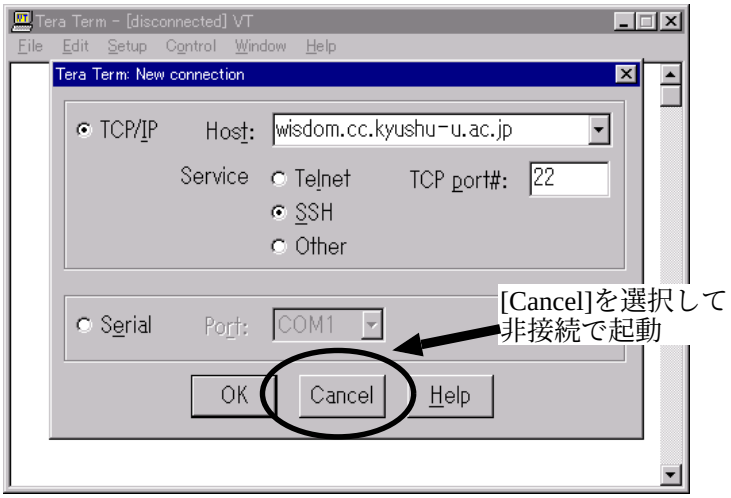

図 9: TTSSH の起動 (ポート転送の準備)

ポート転送をするためには，[Setup] メニューの [SSH Forwarding] を選択します．すると図 10 のような「SSH Forwarding」ウィンドウが表示されます．

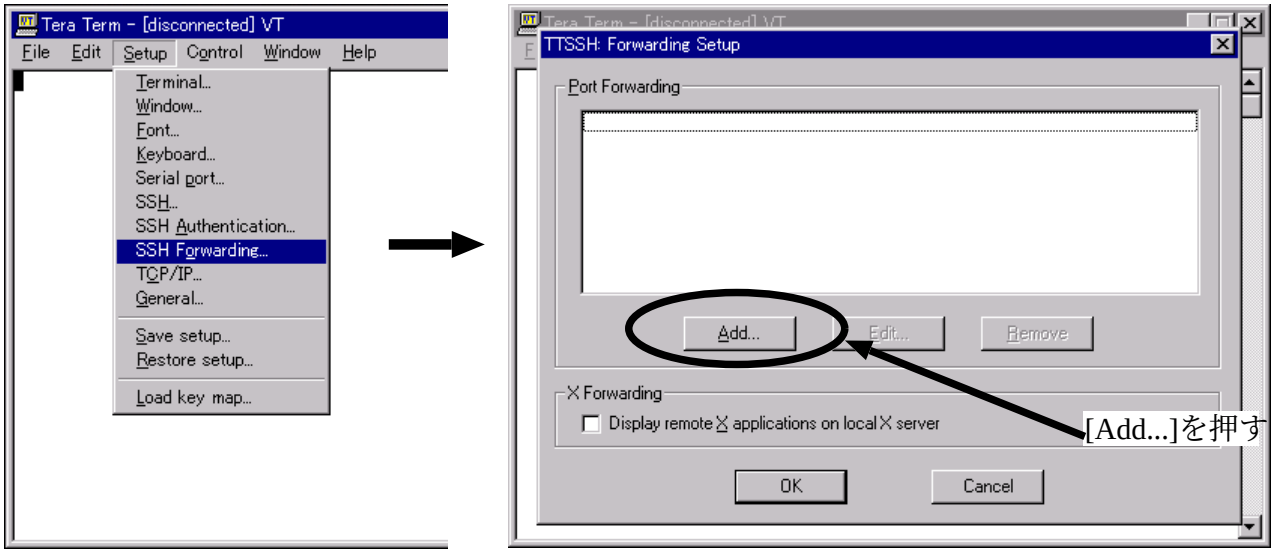

図 10: ポート転送設定ウィンドウ 1

次に「SSH Forwarding」ウィンドウで [Add] ボタンを押して，転送するホストやポート番号の追加を行ないます．すると図 11 のウィンドウが表示されます．

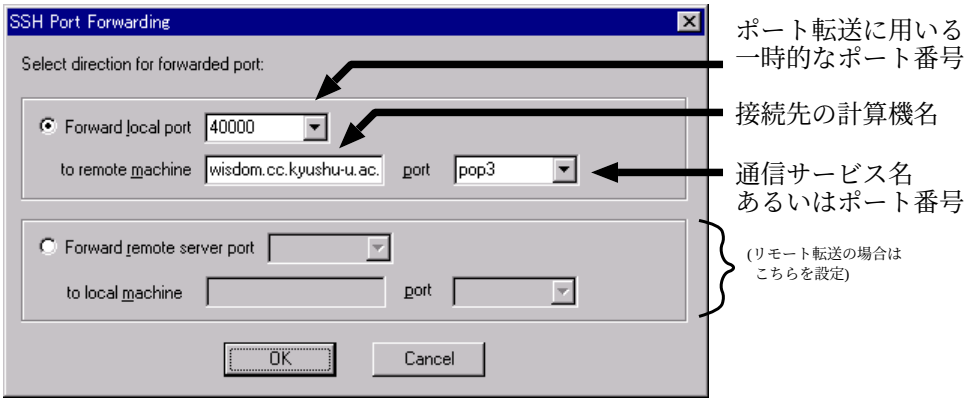

図 11: ポート転送設定ウィンドウ 2

[Forward local port] と [port] の欄で [pop3] を選択して，[to remote machine] の欄に `wisdom` を指定してください (図 11)．ポート番号は直接数字で入力しても構いません．数字指定の場合は，POP は 110，FTP は 21 にと指定しましょう．

設定が終了したら，[ok] ボタンを押し，[Setup] メニューの [Save setup...] を選択して，この設定をファイルに保存します．ここでは保存する設定ファイルの名前を “POP.ini” とします[4]．デフォルトの設定ファイルである “Teraterm.ini” で保存した場合は，ポート転送ではない SSH の利用のときにも，転送用の通信路を確保してしまうので注意してください．

[4]拡張子は指定しなくても自動で “ini” になります．

4.2 ポート転送用ショートカットの作成

TTSSH の設定ファイルとして “POP.ini” を利用することで，SSH によるポート転送が実現できます．しかし設定ファイルを毎回指定するのは面倒です．そこで，TTSSH へのショートカットを作成し，起動時に自動的に “POP.ini” ファイルを利用するように設定しましょう．

まず，TTSSH のショートカットをデスクトップ上に作成します．`ttssh.exe` ファイルをコピーし，デスクトップ上で [ショートカットの貼り付け] を行なえば，ショートカットアイコンが作成されます．作成されてたショートカットアイコンに，ここでは「SSH ポート転送 (POP)」という名前を付けました．次に作成したショートカットのプロパティを変更します．図 12 に示しているように，TTSSH のアイコンを右クリックし，[プロパティ] を選択して，プロパティウィンドウを表示させて下さい．

図 12: ポート転送用ショートカットの設定

この中の [リンク先] 欄を次のように変更します．

```
"C:¥Program Files¥TTERMPRO¥ttssh.exe" /F=POP.ini /ssh wisdom.cc.kyushu-u.ac.jp:22
```

最初の `C:¥Program Files¥TTERMPRO` の部分は，Tera term および TTSSH をインストールしているフォルダです．別のフォルダにインストールしている方は当該フォルダ名を指定して下さい．“`/F`” は起動時の設定ファイルを指定するオプションです．先に保存しておいた “POP.ini” を指定しましょう．最後の “22” は SSH サーバが利用するポート番号です．

4.3 ポート転送による POP 利用例 (Netscape Messenger)

作成したショートカットから TTSSH を起動します．すると相手計算機への SSH による接続が開始します．SSH による認証の後，ログインが終了しますと，SSH によるポート転送の準備が終った事になります．**POP クライアント利用中はログアウトしないで下さい**．ログイン期間中は SSH による転送が有効ですけれども，ログアウトしてしまうと SSH の転送も終了してしまうため，POP クライアントは相手計算機の応答がないものと見做してしまい，メールの取得ができません．ログアウトしてしまった場合は，もう一度起動して再度ログインして下さい．

次にポート転送したいアプリケーションを起動し，POP サーバの設定で，サーバ名を自分自身を表す localhost (または 127.0.0.1) に，ポート番号を “40000” にして，メール転送プロトコルを POP3 として指定します．ここでは「Netscape Communicator Versionl4.6 英語版」の設定例を説明します．

まず [Edit] メニューの [Preferences] 項目を選び,「Preferences」ウィンドウを開きます．図 13 が「Preferences」ウィンドウです．次に,

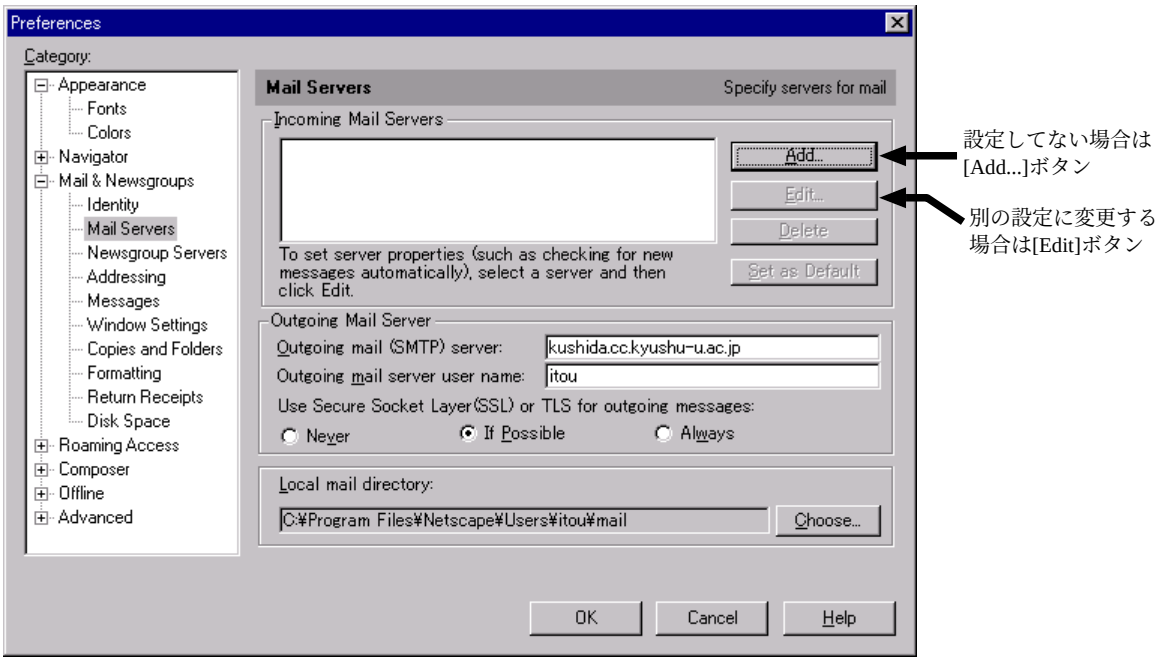

図 13: Netscape の設定例 (1)

図 13 のように，[Mail & Newsgroups] 項の [Mail Server] 項を表示させ，[Add] または [Edit] ボタンを押して図 14 のように,「Mail Server Properties」ウィンドウを表示させます.

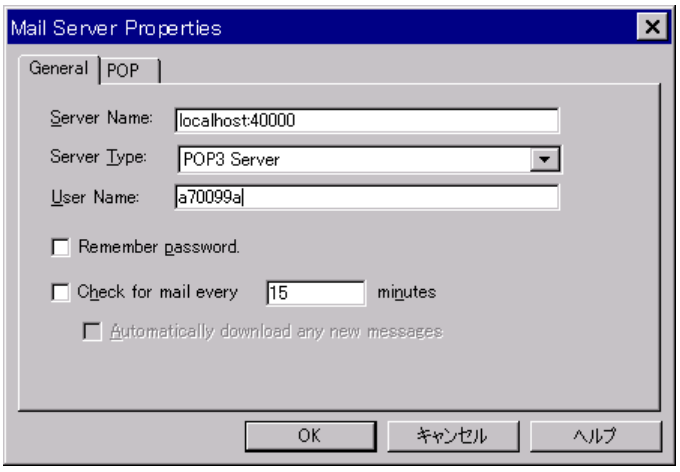

図 14: Netscape の設定例 (2)

サーバの種類は [POP3] を選びます．[General] タグを押し，サーバ名を “localhost:40000” にします．これは接続先として，計算機は自分自身 (localhost) の 40000 番ポートを指定する，という意味です．最後に，本当の接続先でのユーザ名を入力します．このように設定しておけば，POP でメールを取得する場合の通信を暗号化する事ができます.

図 15 にポート転送中の netstat コマンド実行結果を示します．firebird と名前を付けた Windows 計算機から，wisdom.cc.kyushu-u.ac.jp へ，ポート転送を用いて POP の通信を行なっているときの結果です．図によると，firebird の 1133 番ポートとと localhost:40000 の間に通信が確立している事がわかります．これはクライアントが自分自身の 40000 番ポートと POP による通信が行なっている事を示しています．また，firebird の 873 番ポー

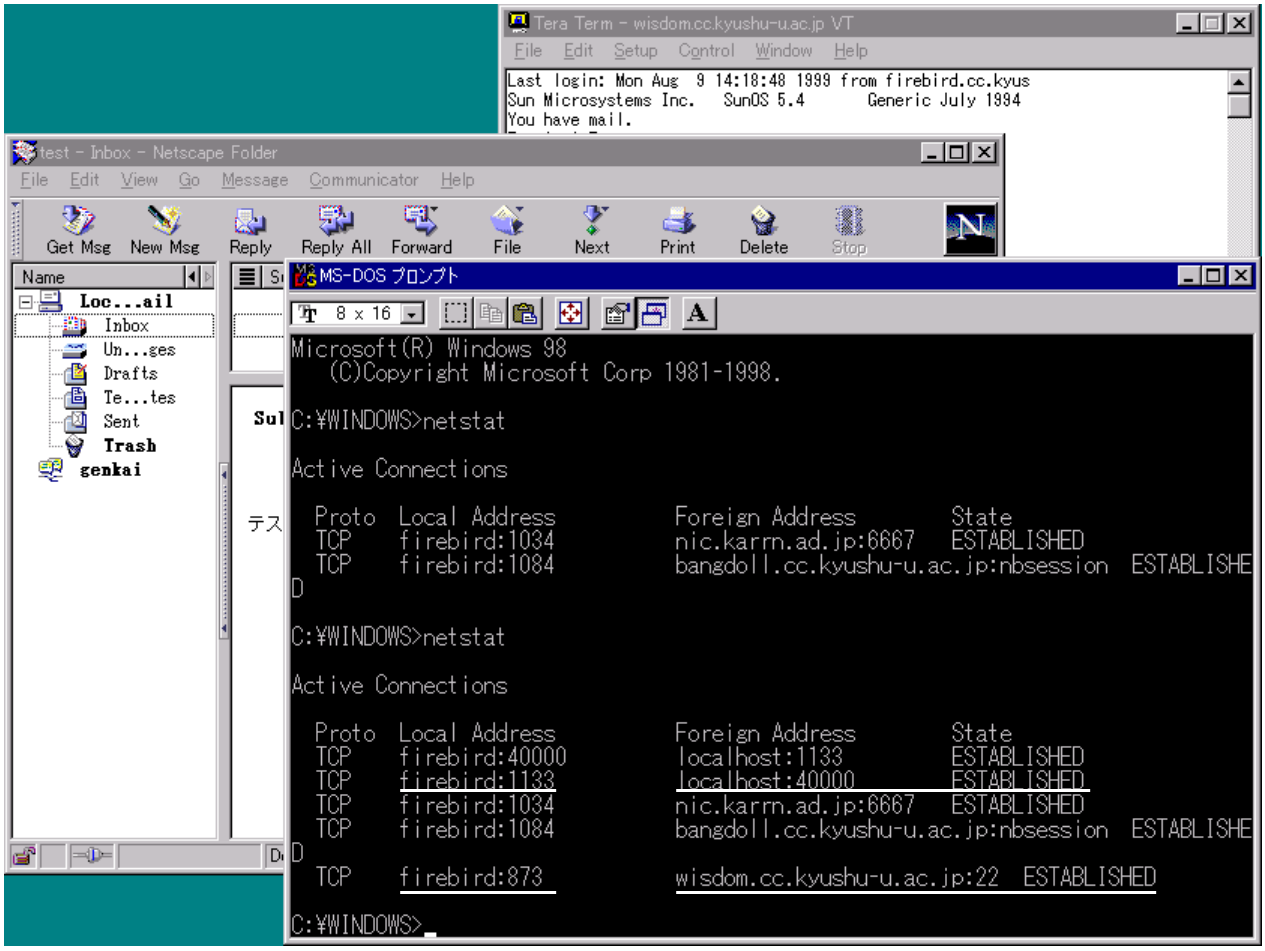

図 15: ポート転送時の状態 (netstat コマンド)

トと wisdom の 22 番ポート (SSH のポート) の間に通信が確立しています．この最後の通信が SSH による暗号化通信で，ポート転送による POP の通信を行なっている部分です．

5 おわりに

今回は SSH には SSH のポート転送機能について解説しました．これは前回の記事 [1] で解説した SSH を用いて，telnet や rsh 以外の通信を暗号化通信で隠してしまうものです．ポート転送機能は，小荷物の宛先や内容を見張られないように，別の SSH 用の出入口に横流しして，情報の小荷物 (パケット) を安全にやりとりするという風にも考えられるでしょう．あるいは，弱い POP プログラムが，堅固な通信エージェントである SSH さんに情報のやりとりをお願いする，といった風に考えても良いでしょうか．

本文中に例として取り上げた POP は，電子メールの受信に使うことのできる有名なサービスです．この POP は，利用者の認証情報であるパスワードを平文で送信していまうという安全性の問題を持っています．平文で送信してしまうため，POP を使っている通信を見張っていれば，簡単にパスワードを盗聴する事が可能です．しかし SSH のポート転送機能を用いれば，POP のような不安全な通信も SSH の暗号化通信路を経由するため，安心してサービスを用いる事ができます．

この記事は九州大学大型計算機センターの利用者の皆様が，SSH のポート転送を用いる場合を対象としています．SSH は大型計算機センター以外にも，様々な状況で用いる事が可能です．現在，電子メールは非常に重要かつ一般的な通信基盤になっています．研究室の計算機にも SSH を導入して，ポート転送を用いた安全に電子メールの受信を行なうように設定されてはいかがでしょうか?

最後にセキュリティについての一般的な事柄を述べます．計算機や通信の安全性について，JPCERT (コンピュータ緊急対応センター) という機関が WWW ページ (http://www.jpcert.or.jp/) や電子メールを用いて，危険性の警告，安全性についての報告，対策方法の連絡をしています．これらの WWW ページを定期的に参照し，対応策取るように心掛けると良いでしょう．

参考文献

[1] 伊東栄典：“SSH:Secure Shell ～おでかけ前に鍵かけて～”, 九州大学大型計算機センター広報, Vol.32, No.2, pp.76–89, 1999.

[2] W. Richard Stevens：“TCP/IP Illustrated, Volume 1. -The Protocols-”, *Addison Wesley Longman, Inc., (1994).*

[3] J. Myers and M. Rose：“Post Office Protocol - Version 3”, RFC1939, May 1996. (`http://ftp.kyushu-u.ac.jp/pub/rfc/rfc1939.txt` , 1999 年 8 月 1 日現在.)

[4] R. Nelson：“Some Observations on Implementations of the Post Office Protocol (POP3)”, RFC1957, June 1996. (`http://ftp.kyushu-u.ac.jp/pub/rfc/rfc1957.txt`, 1999 年 8 月 1 日現在.)

[5] R. Gellens, C. Newman, L. Lundblade：“POP3 Extension Mechanism”, RFC2449, November 1998. (`http://ftp.kyushu-u.ac.jp/pub/rfc/rfc2449.txt` , 1999 年 8 月 1 日現在.)

[6] 伊東栄典, 笠原義晃：“電子メール用ソフトウェア Mew の使い方”, 九州大学大型計算機センター広報, Vol.31, No.3, pp.143–155, 1998.

[7] “TTSSH ホームページ”, `http://www.zip.com.au/ roca/ttssh.html`, 1999 年 8 月 1 日現在.

[8] “Tera Term (Pro) ホームページ”, “`http://hp.vector.co.jp/authors/VA002416/` ,” 1999 年 8 月 1 日現在.