フリーソフトウェアを用いた通信のセキュア化

ヒゲタ醤油株式会社

■ 執筆者Ｐｒｏｆｉｌｅ ■

猿　田　智　勇

1996年　ヒゲタ醤油（株）入社
　　研究開発部
1998年　財務部システムグループに異動
　　PCネットワークシステム管理者

■ 論文要旨 ■

ネットワークの重要性が増し，情報インフラとして重要な位置を占めるにつれて，通信のセキュリティは重要な問題となっている．

しかしネットワークにおいて，セキュリティは最も高コストのものの一つであり，非常に高価格の機器・ソフトウェアが用いられている．

今回は，当社で使用している環境を元に，フリーソフトウェアを主に用いて，リモートログオン，電子メールの送受信，ファイルの送受信を安全に行う方法について模索した．

■ 論文目次

１．はじめに ・・・《 3》

２．通信の危険性 ・・・《 3》

３．概要 ・・《 4》
３．１　使用機器および構成 ・・・《 4》
３．２　通信内容 ・・《 4》

４．運用 ・・《 4》
４．１　リモートログオン ・・《 4》
４．１．１　認証の保安化 ・・・《 5》
４．１．２　通信内容の暗号化 ・・《 5》
４．２　電子メール送受信 ・・・《 5》
４．２．１　認証の保安化 ・・・《 5》
４．２．２　通信内容の暗号化 ・・《 6》
４．３　ファイルの転送 ・・《 7》
４．３．１　認証の保安化 ・・・《 7》
４．３．２　通信内容の暗号化 ・・《 7》

５．総括 ・・《 9》

■ 図表一覧 ■

図１　TeraTermを用いたリモートログオン ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・《 4》
図２　POP3/SMTPのPort Forwarding概略図 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・《 6》
図３　ttsshの, SSH Forwarding設定 ・・《 7》
図４　S/MIMEのしくみ ・・《 7》
図５　FTP/HTTPのPort Forwarding概略図 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・《 8》
図６　Internet Explorer5.01sp2のプロキシサーバ設定画面 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・《 8》
図７　ブラウズされたFTPサーバ ・・《 9》

1. はじめに

現在，インターネットへの常時接続回線の整備が急速に進んでいる．それに伴い，社会全体のインターネットへの依存は高まっており，重要なインフラとなりつつある．

また，自由であるという特性上，無数の使用者によってさまざまな情報がやりとりされており，その中には，違法なもの，悪意を持ったものも含まれている．

JPCERT/CC(コンピュータ緊急対応センター)に寄せられる被害報告も，2000年1月を境に急増しており [1]，警察庁によるアンケート調査[2]によると，過去1年間に不正アクセスなどの被害を受けていたサイトは，全体の19.8%にものぼっている．

外部からの攻撃が成立した場合，システムの不正使用，データの盗用，サービスの停止といった問題が生じ，その被害は，巨大サイトの場合数億ドルにのぼることもある．

以上，セキュリティに対する対応は急務である．しかし，システム構築の中で最も難しく，コストと経験を要するものの一つがセキュリティであり，またコストをかけたからといって高いセキュリティが得られるとは限らない．

かと言って，セキュリティに常にコストがかかるとは限らない．低コストで運用する必要がある場合には，サービスを可能な限り限定すれば，相対的に守る範囲も小さくなる．

今回は，WindowsクライアントからUNIXサーバへアクセスを行う場合を想定して，通信をセキュア化し，公開するサービスを少なく保ちながら，各種の一般的なサービスを行うことについて検討した．

なお，本文で用いられているソフトウェアは，WindowsおよびInternet Explorerを除いて，特に明記ない限り，すべて"無料"で"ソースコードが公開"されている”フリーソフトウェア"である．

２．通信の危険性

ネットワーク上の通信は，街中で会話をしているのと同様であり，同一ネットワークに他の使用者が存在する場合，通信内容は常に傍受されている可能性がある．

そのため，特に対策を行なわないと，正規ユーザのパスワードが悪意の者によって傍受されてしまい，そこからサーバやLANへ侵入されたり，データを盗まれたりということが考えられる．

被害の規模は攻撃の種類によって差があるが，顧客データや人事データが流出した場合，クレジットカード番号や，顧客や社員のプライバシー情報が流出する可能性がある．

さらに，秘密保持契約を結んでいたデータが流出した場合，被害は1社にとどまらない．

また，侵入者が，侵入先のサイトから別のサイトへと攻撃を仕掛けるかもしれない．この場合，攻撃を受けたサイトから見ると侵入先サイトからの攻撃に見えるため，同様に被害者であるにもかかわらず，攻撃を受けたサイトから訴訟を受ける危険性もある．

そういった場合，企業イメージの失墜は避けられない．

本報告では，上のような危険を未然に防止するための対応について検討を行っている．

３．概要

３．１　使用機器および構成

クライアントとして，Windows NT4.0 + ServicePack6a を用いる．また，ウェブブラウザとして Internet Explorer 5.01sp2を用いる．

サーバは，フリーソフトウェアである FreeBSD4.3STABLE[3]を用いて構築している．

３．２　通信内容

TCP/IPを用い，リモートログオン(4.1)，電子メールの送受信(4.2)，FTPによるファイルの転送(4.3)を行う．

４．運用

今回は，対策を「認証の保安化」「通信内容の暗号化」の二通りに分類して概説する．

認証の保安化を行うことで，一般的に，手軽にセキュリティを向上させることができる．

通信内容は平文のため，盗聴される危険性がある．

→なりすまし，侵入，不正使用を防ぐことができる．

通信内容の暗号化処理を行うことで，盗聴された場合でもデータの流出を防ぐことができる．

→なりすまし，データの漏洩，不正使用を防ぐことができる．

４．１　リモートログオン

Windowsクライアントからターミナルソフト(TeraTerm[4])を用いて，UNIXサーバにログオンする．

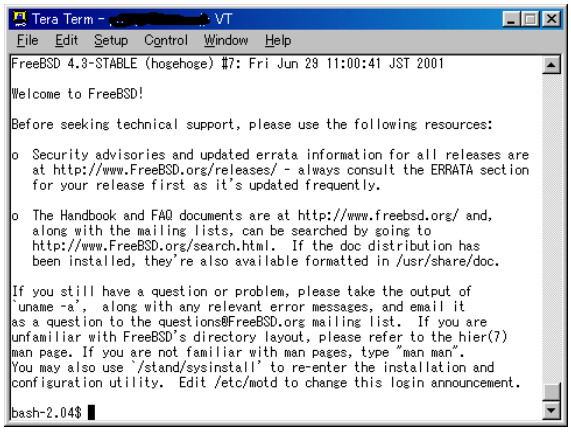

図１　TeraTermを用いたリモートログオン

４．１．１　認証の保安化

OTP(ワンタイムパスワード) を用いる方法.

OTPは，一度だけ有効なパスワードを使って認証を行う手段である．そのため，パスワードが盗聴されたとしても，盗聴した者はそのパスワードを使うことができない.

FreeBSDの場合，s/key [5]等が存在する.

４．１．２　通信内容の暗号化

通信内容全体を暗号化した場合，盗聴された場合でも内容を秘匿することができる.

SSH(Security SHell) プロトコルは，公開鍵[6]方式で通信を暗号化するプロトコルであり，広く使われている．SSHプロトコルを実装したフリーソフトウェアとして，OpenSSH[7] 等がリリースされている.

a.　クライアント側の設定

ttssh[8]は，TeraTerm用のSSH機能拡張プラグインである.

TeraTermにこのソフトを追加することで，TeraTerm上でSSHプロトコルを使用することができる.

ttsshには，暗号鍵を作成するツールが含まれていない．そのため，UNIXにてあらかじめ作成した鍵ペアを，フロッピーディスク等を用いる等，“安全な方法で”クライアントにインストールする.

b.　サーバ側の設定

FreeBSD 4.3Rには，最初からOpenSSHが入っているので，環境に応じた設定を行う.

設定例として， /etc/ssh/sshd_config を以下の通りとする.

```
RSAAuthentication yes      # RSA認証を用いる
PasswordAuthentication no  # パスワード認証を用いない
PermitEmptyPasswords no    # 空のパスフレーズを使わない
PermitRootLogin no # 管理者でのログオンを禁止
(他の部分はdefault)
```

サーバに，「公開鍵(identity.pub)」を，authorized_keys として登録すると，SSHプロトコルを通した通信が可能となる.

詳細な使用法は，各ソフトウェアのマニュアルを参照のこと.

４．２　電子メール送受信

４．２．１　認証の保安化

a.　APOPの使用

APOPは，通常のPOP3プロトコル(電子メールを受信するプロトコル)の認証を暗号化したものである.

最近では，Outlook Express，NetscapeMessanger以外，ほとんどの電子メールソフトがサポートしている.

b.　IMAP4+CLAM-MD5 の使用

IMAP4 は，電子メールを受信するためのプロトコルであるが，POP3より多機能なものである．その認証を，CLAM-MD5 という方法で安全にしているのがIMAP4+CLAM-MD5 と呼ばれるプロトコルであるが，現時点においてWindows用の電子メールソフトで対応しているものは少ない．

４．２．２　通信内容の暗号化

a.　SSHの port forwarding機能(**図２**)を使用する

ttsshの，ssh forwarding... 設定から，サーバのPOP3/SMTP(電子メールを送信するプロトコル) ポートを，クライアント機の別ポートにforwardする設定を行う．(**図３**)

次に，電子メールソフトのPOP3/SMTPサーバ設定に，localhost(クライアント自ホスト)の，forward元ポートを指定すると，SSHによって暗号化されたトンネルを通ってサーバに届けられる．

今回は例として，

POP3(標準では110番ポート)を，localhostの8110番ポート(POP3 over SSH)，

SMTP(標準では25番ポート)を，localhostの8025番ポート(SMTP over SSH)

からトンネリングさせることとした．(**図2，3**)

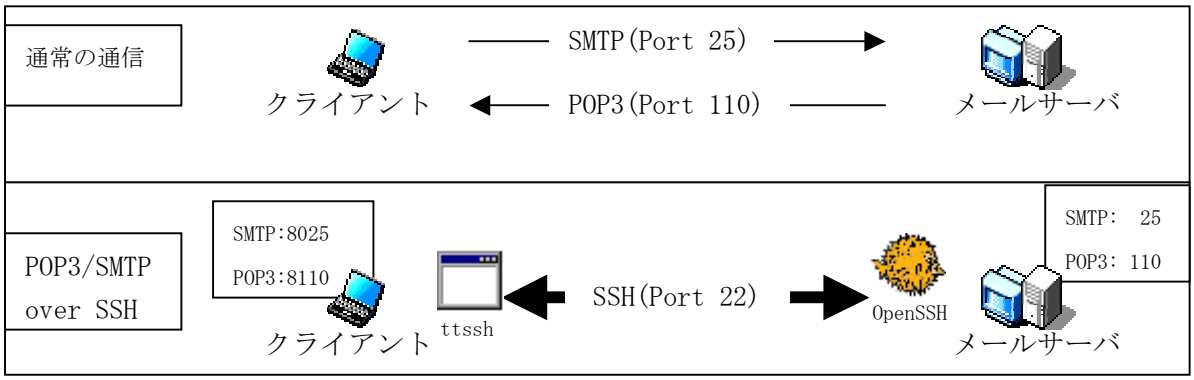

図２　POP3/SMTP の Port Forwarding概略図

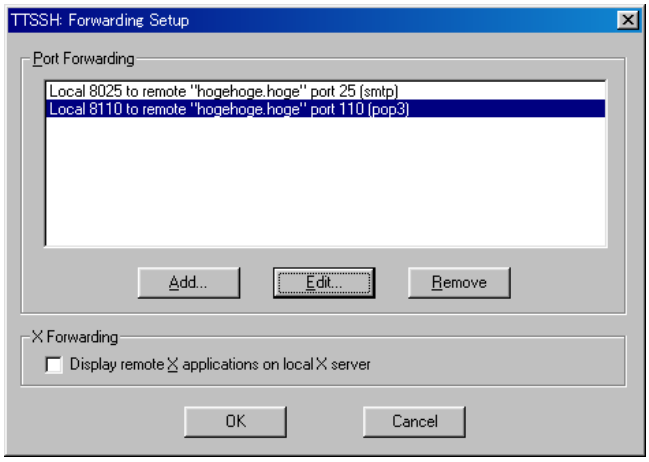

図３　ttsshの，SSH Forwarding設定

b.　PGP[9]，S/MIME[10]を使用する

PGP，S/MIMEともに，公開鍵暗号を使用して，電子署名・メールの暗号化を行うソフトウェアである．

S/MIMEの場合，認証局に公開鍵を登録することで，公開鍵を安全に配布することができる．

認証局への登録は一般的に有償であるが，OpenSSL[11]を用いて，自分で認証局を作成することも可能である．

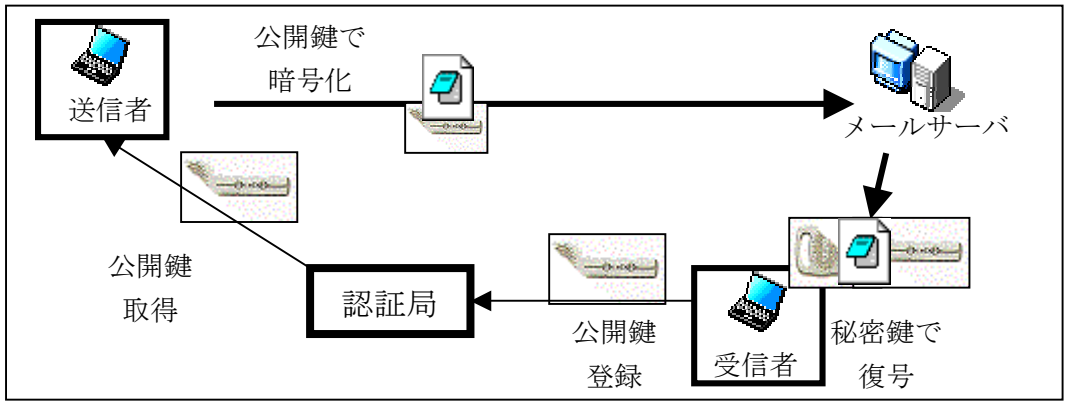

図４　S/MIMEのしくみ

PGPの場合もほぼ同様であるが，認証局が存在しないので，公開鍵は何らかの信頼できる手段でやりとりする必要がある．

４．３　ファイルの転送

４．３．１　認証の保安化

SSHの port forwarding機能を使用する．

4.2.2に示した事例と同様に設定することで，ttsshのport forwarding 機能を使用して，localhost 8020番ポートから，サーバの20番ポート(FTPの制御ポート)をSSHでトンネリン

グさせることで，パスワード送信を暗号化することができる．

その状態で，適切なFTPクライアントソフト(FTP explorer[12]等)を用いることで，パスワード認証を保安化した状態でのファイル送受信ができる．

※ FTP explorerは，個人使用，教育目的で使用する場合には無料だが，それ以外の目的に使用する場合は$30のシェアウェアとなる．

４．３．２　通信内容の暗号化

サーバ側にプロキシサーバを置き，その間の通信をSSHでトンネリングする．

(1) 4.2.2に示した事例と同様，ttsshのport forwarding 機能を使用して，localhost 8080番ポートから，サーバの8080番ポートをトンネリングさせる．

(2) FreeBSDサーバに，squid[13]ないしdelegate[14] 等の，プロキシサーバ(代理サーバ)を導入し，8080番ポートに設定しておく．（**図５**）

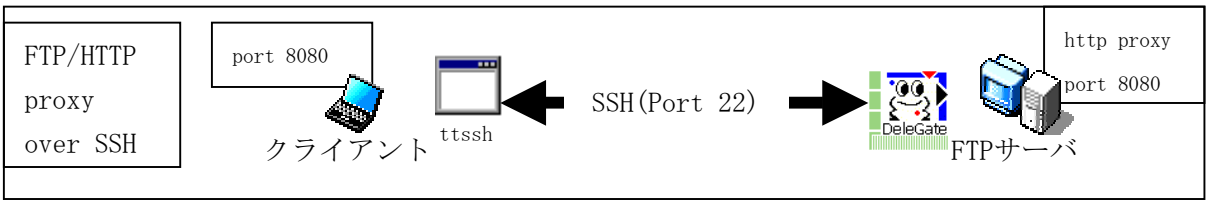

図５　FTP/HTTPの Port Forwarding概略図

(3) InternetExplorerの，プロキシサーバ設定において，localhost 8080番ポートを設定する．（**図6**）

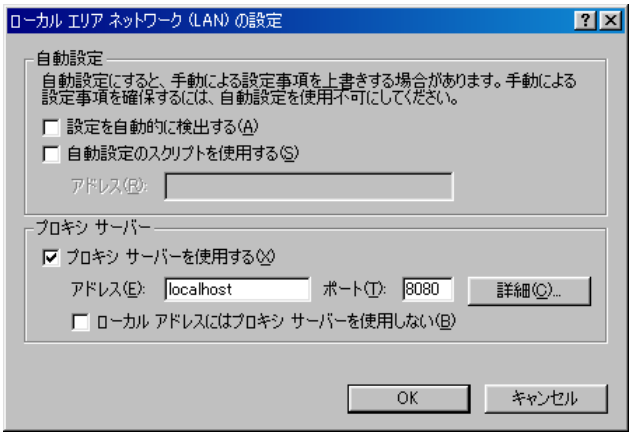

図６　InternetExplorer5.01sp2の，プロキシサーバ設定画面

この状態で，FTPサーバをブラウジングすると，通信内容すべてがSSHにトンネリングされ，安全なファイルのやり取りが可能である．（**図７**）

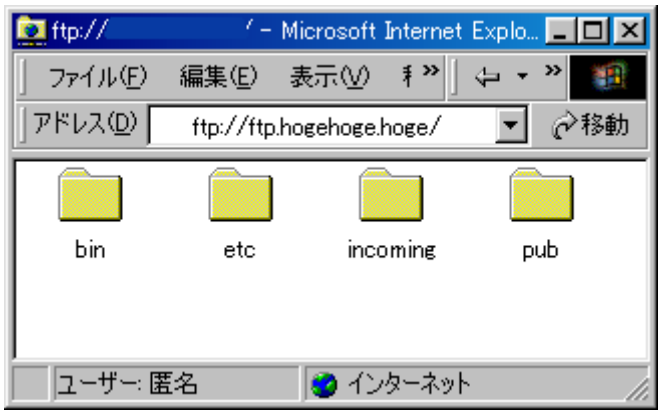

図７　ブラウズされたFTPサーバ

５．総括

現在，ネットワーク犯罪による実被害を受ける危険性が急速に高まっている．また，ネットワークへの依存度が高まるにつれて，その被害規模も拡大しつつある．

現在，セキュリティへの対応は最優先事項の一つと言っても過言ではないであろう．

一般的に，セキュリティを高めると利便性は低下する．

セキュリティ×利便性＝技術力×運用力

という計算が成り立ち，セキュリティを高めるために利便性が低下することが多い．

こういった現状の中でセキュリティを高めるためには，

a.　システム管理者の技術力と運用力

b.　管理業務者の理解，安全と利便性のトレードオフを行う政治的な判断

c.　ユーザの協力

が必要であり，どれか一つ欠けてもセキュリティの向上は困難である．

そのためには，管理業務者へのプレゼンテーション，ユーザ教育等，ネットワーク技術とは直接の関連のない，社会的な対応が非常に重要となる．

今回は，リモートログオン，電子メールの送受信，ファイルのやりとりについて，高セキュリティ条件下でのサービスの提供方法について検討を行ったが，さらに他の各種通信を暗号化することは可能であり，それを行うことで，セキュリティを保ちながらサービスの質を保持することが可能である．そして，サービスを維持するためには，さらに上記 a, b, c のすべての向上が必要である．

そして，ソフトウェアには必ずバグがあり，高セキュリティのソフトウェアにも必ずセキュリティホールが発見される事から，サービスは提供して終わりではなく，継続してメンテナンスされることが重要であると言える．

今回取り上げることはできなかったが，本論文に書かれていない，多数の実装・プロトコルがある．

例として，Windows上で使用できる，SSHを実装したソフトウェアには，Winscp.exe,

ssh.exe，port forwarderといったものがあり，それぞれ使い勝手が異なり，物によって機能が特化されている．

また，SSH以外にもRSA認証を実装した暗号化ソフトウェアとして，Zebedeeが知られている．

そういった最新の情報を常に追いかけ，複数の選択肢を持っておくことは重要である．

以上より，完璧なセキュリティというのは不可能であるということを常に意識し，またセキュリティに対する対応を持続させることが，もっとも安全にサービスを提供することと言って良いだろう．

脚注

[1] JPCERT/CC(コンピュータ緊急対応センター)
< http://www.jpcert.or.jp/stat/reports.html >

[2] 不正アクセス対策に関するアンケート報告書（平成13年6月14日）
< http://www.npa.go.jp/police_j.htm >

[3] FreeBSD < http://www.JP.FreeBSD.ORG/ >
4.3BSDを起源とした，フリーなUNIXライクOSのひとつ.

[4] TeraTerm < http://hp.vector.co.jp/authors/VA002416/ >
寺西 高氏作成のターミナルソフト.

[5] FreeBSD 日本語マニュアル
<http://www.jp.freebsd.org/cgi/mroff.cgi?subdir=man&lc=1&cmd=&man=skey&dir=jpman-4.3.0%2Fman&sect=0>

[6] 公開鍵暗号方式は，暗号化と復号化に異なる鍵を用い，暗号化の鍵を公開しておき，復号化の鍵だけを秘密にしておけば，誰でも公開された鍵で暗号化できるが，それを復号化できるのは秘密鍵を持っている者に限られる，という方式.

[7] OpenSSH < http://www.openssh.org/ >
SSHプロトコルを実装したフリーウェア.

[8] ttssh < http://www.zip.com.au/~roca/ttssh.html >
Robert O'Callahan氏作成の，TeraTerm用SSHツール.

[9] PGP < http://www.pgpi.org/ >
Pretty Good Privacy. 公開鍵暗号方式により，メールの電子署名および暗号化を行う.

[10] 代表的な認証局の一つ ベリサイン社
< https://digitalid.verisign.co.jp/client/index.html >

[11] OpenSSL <http://www.openssl.org/>
OpenSSHと同様，SSLのフリーな実装.

[12] FTP explorer < http://www.ftpx.com/ >

[13] squid < http://www.squid-cache.org/ >
代表的なプロクシサーバソフトのひとつ.

[14] delegate < http://www.delegate.org/delegate/ >
代表的なプロクシサーバのひとつ. 多彩な機能がある.