様々な回線とシステムでの インターネット接続認証の統合

広瀬　雄二

東北公益文科大学公益学部

概　　要

ある計算機をインターネット回線に繋ぐためにはなんらかの形式で「利用権」を得る必要がある。最も一般的な利用形態であるイーサネットによる有線回線では、物理的回線に直接接続するだけでインターネット回線に自由に接続できる権利を与えるのが一般的である。しかし、組織外の人間が自由に立ち入ることのできる場所では情報漏洩を防ぐために、本人認証を行なう必要がある。ただしこれには特殊な機能を有した機器が必要となるため現実的には認証なしで利用している事例が多い。

一方、無線 LAN のように地理的に離れた場所からもアクセスできるものでは、盗聴等も容易にされる可能性があるので、WEP などの共有秘密鍵機構を用いて部外者には利用できないようにすることが既に一般的となっている。

これまで、インターネット回線接続のための認証手法は、「有線回線」、「無線回線」、「リモートアクセス(ダイヤルアップ等)」といった物理回線によって別々のものを用いるのが一般的であった。さらに、利用する認証手法によっては接続クライアントのシステム(OS)を選ぶものもあった。

本研究では、回線の種別、接続クライアントの OS を問わずに全く同一の手法で利用できる認証方式を提案しその有効性について考察を行なう。

Abstract

To obtain connectivity to the Internet of personal computers, they must be authenticated by certain means. As to wired connection, Ethernet for example, since authentication equipments for Ethernet are not popular to end users yet, it is general that administrator allow any user who plugs cable into HUB to access to their LAN.

As to wireless LAN on the other hand, authentication method, such as WEP, is commonly used because we recognize 'tapping' air waves is much easier than that for wired connection.

Thus we have to choose suitable authentication method for physical line - wired LAN, wireless LAN or remote access(dial-up connection). End users have to setup a number of authentication client software according to physical connections they use. In this paper, we propose the generic authentication method which is independent of any physical line nor of OS.

1. はじめに

パーソナルコンピュータ (以下 PC) であっても、ネットワークに接続して利用するのが自然になり、とくに大きな投資をすることなくインターネットに接続できる環境がそろっている。中でも無線 LAN による接続形態は空間的制約を軽減できるため、末端利用者にとって快適な選択肢となっている。無線接続は有線通信に比べて速度的に劣ることから、これまでは補助的な位置付けと認識されていたが、802.11g[1] などが普及価格帯に下りてきたことから、実用的な接続としての選択肢に上がってきた。その反面、物理的な接続にしばられないという自由度の高さから生ずる問題もある。

無線 LAN は通信基地局の近傍にいるだけでアクセスできる。これは組織に関係ない者が持ち込んだ PC にも接続権利が発生することになる。部外者の利用を抑止するために無線 LAN 接続にはいくつかの認証機構があり、WEP や 802.11X はその代表的なものである。

WEP は 802.11b の普及初期から使われている共有秘密鍵を用いた通信暗号化のためのプロトコルである。設定が手軽である反面、鍵を盗まれたり見破られたりする危険性が高く、もはや安全とはいえないことが指摘されている。

802.11X は公開鍵基盤を用いてサーバ側とクライアント双方を認証してか

[1]IEEE 802.11g の通信スループットの理論値は54Mbps 。これより古くから普及している802.11b は理論値11Mbps 。

ら通信を許可する方式である。通信のための秘密鍵は動的に更新して行くことで安全性を高めている。サーバに対して認証依頼をするクライアント側モジュールを、「サプリカント」と呼ぶが、現状ではもっともシェアの高いMS-Windows用のものしか用意されていない場合が多く、FreeBSD, Linux, Mac OS X など多様性をもつネットワークには導入できないのが現状である。初期設定もサーバ側とクライアントの両方に鍵を登録しなければならないので、多くのクライアントPCをかかえる場所では管理コストが無視できない。

また、802.11X認証のために行なう設定は、802.11X認証のためにしか使えない。言い換えると、VPNなど外部ネットワークからのLAN接続には全く別の認証準備設定をしなければならない。

2. 汎用的で統合的な通信路認証機構の提案

前節で述べた、既存の通信回線認証機構の問題点を考慮して、本稿では以下の利点を備えた通信路認証機構を提案する。

- 認証クライアントのシステムの種類を問わない
  (Windowsだけでなく Unix、Macからも使える)
- 認証を要求する通信路の種類を問わない
  (無線LANでも、外部インターネットからのVPN接続でも利用可能)
- 長期的将来に渡って利用し続けることができる

これらの観点から判断し、認証機構としてPPP over SSHを利用する方法を提案する。

2.1 PPP over SSH

PPP over SSH は、2台のホスト間でppp接続を行ないそれをVPN[2]として

[2]Virtual Private Network、物理的に異なるセグメントに設置されたホストどうしを、既存のインターネットを通じて接続して同一のセグメントに配置する論理的なネットワーク。

利用する手法である。PPPは通常プロバイダへのダイヤルアップ通信確立のために使用されるのが一般的であるが、ダイヤル発呼の代わりに一般プロセスを使用することもでき、その部分にSSH[3]によるリモートホストコマンド起動を使うのがPPP over SSHである。現在普及しているOSでは、PPPもSSHもそれを実現するためのソフトウェアが標準で装備されているかもしくは容易に導入可能な状態になっている。

2.2 OSごとの実現方法

代表的なOSでPPP over SSHの接続に必要なものを表1に示す。

表中に記載された実装のうち、Windows用のPPP over SSH client for WindowsだけはOS非標準のものであるが、これは`http://www.kmc.gr.jp/proj/vpn/`により自由に入手し導入可能なパッケージである。

3. PPP over SSHを利用した無線LAN認証

一般に、無線LANではステーションとそれに接続する機器で同じサブネット内のIPアドレスを割り当てて通信を行なう。多くの場合、無線LAN機器に、実際に外部インターネットにアクセス可能なIPアドレスを割り当てて使う(図1)。この場合、割り当てたIPアドレスでの通信を許すことは外部インター

表1：PPP over SSHに利用する実装

OS	SSH実装	PPP実装
FreeBSD	OpenSSH	FreeBSD user-ppp
NetBSD	OpenSSH	pppd
Linux一般	OpenSSH	pppd
Mac OS X	OpenSSH	pppd
Windows 2000 Windows XP	PPP over SSH client for Windows	

[3]Secure SHell,暗号化された安全なリモートログインのためのツール群

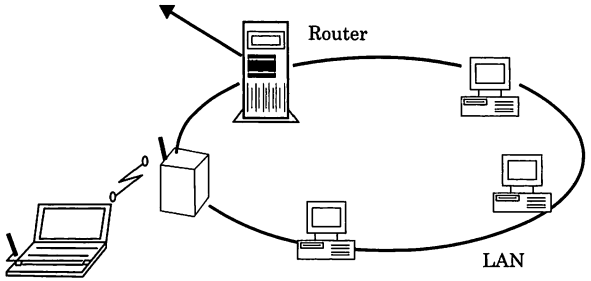

図1：無線APをLAN内に配置

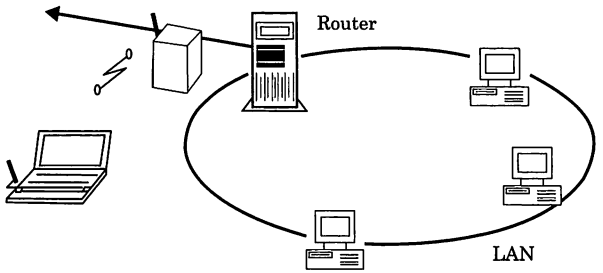

図2：無線APをLAN外のセグメントに配置

ネットへの全ての通信を許すことになるため、部外者の利用を禁止したい場合はIPアドレス割り当ての段階で認証を課すことになる。しかしこの場合、無線LANでしか有効でない認証機構を使用せざるを得ないことになる。

ここで発想を転換し、無線LANに物理的に割り当てられたIPアドレスは、接続してきたクライアントにも自由に割り当てるようにする代わりに、「外部インターネットと同じくらい信用できないもの」と見なすことにし、そのままではLANにアクセスすることができないようなLAN設計とする（図2）。その上で、LAN内に設置されたサーバマシンとVPN接続を求め、VPN接続を確立できたもののみLANに接続を許すような形にする。つまり、出先からLAN

にアクセスする際にVPN接続を提供するのと同じ認証を無線LANに適用する。こうすることで、回線の種類にも、OSにも依存しない無線LAN認証が可能となる。

3.1 PPP over SSHの設定

PPP over SSHは、以下の手順で構築する。

・VPN接続クライアント側

1．サーバのVPNユーザとしてログインするときに使用するSSH鍵を作成する

2．1で作成した鍵を利用して認証サーバにログインするためのスクリプトを作成する

3．2のスクリプトをデバイスとしてPPP接続を行なう設定を作成する(PPP実装に応じて異なる)

・サーバ側

1．無線LANの物理IPアドレスはLANに通さないようにパケットフィルタリングする

2．VPN 接続ユーザを作成(vpn とする)

3．クライアントのSSH接続公開鍵をvpnユーザの~/.ssh/authorized_keysに追加

4．クライアントに与えるIPアドレスを決定しクライアントからのPPP接続依頼についてそれを割り当てる

VPN接続クライアントは表1にある全てのシステムで利用可能であるが、サーバ側に関しては標準状態でデーモン起動が可能なシステムに限られるためWindows 以外のものに限定する。

リスト1　ipfwの例

```
ipfw add allow tcp from 192.168.11.0/24 to me 22
ipfw add reset tcp from 192.168.11.0/24 to any
ipfw add deny all from 192.168.11.0/24 to any
```

4. 設定例

ここでは、VPN認証サーバとしてFreeBSDを、そのクライアントとしてMac OS Xを使用した例について記述する。

4.1 VPN認証サーバの設定

表2に示したホストをVPN認証サーバとして用いた。その手順の概略を示す。なお、本稿の実験では、無線LANセグメントと基幹LANに付与したサブネットを表3に示したものとした。

1. 無線LANの物理IPアドレスはLANに通さないようにパケットフィルタリングするシステム標準装備のパケットフィルタリングツールipfwを利用して無線LANセグメントアドレスからのパケットはSSHポートのみに限定する(リスト1)。
2. VPN接続ユーザを作成(vpnとする)
   スーパーユーザにて

```
pw useradd vpn -d /etc/ppp/vpn
```

   として、/etc/ppp/vpnをホームディレクトリとするユーザvpnを作成する。ホームディレクトリとSSH用の個人管理ディレクトリも作成する。

表3：実験で用いたサブネットアドレス

無線LANセグメント	192.168.11.0/24
基幹LANセグメント	10.0.0.0/24

表2：VPN認証サーバの概要

CPU	AMD Duron 900MHz
OS	FreeBSD 5.2.1
PPP実装	FreeBSD user-ppp
SSH実装	OpenSSH_3.6.1p1

リスト２　鍵に対してコマンドを限定するエントリ例

```
command="/usr/sbin/ppp -direct vpn-1" ssh-dssAAAAB3NzaC1kc3MAAACBALmMCZhAnVFh...
```

```
mkdir -p /etc/ppp/vpn/.ssh
chown -R vpn /etc/ppp/vpn
chmod 700 /etc/ppp/vpn/.ssh
```

3．クライアントのSSH接続公開鍵をvpnユーザの~/.ssh/authorized_keysに追加した公開鍵によるログインに対しては、決められたコマンドのみを起動するようにする。ここでは

```
ppp -direct vpn-1
```

というコマンドを許可するものとする。リスト２のようにして１行の鍵を追加する。

4．クライアントに与えるIPアドレスを決定しクライアントからのPPP接続依頼についてそれを割り当てる

/etc/ppp/ppp.confのvpn-1エントリを作成し、割り当てたIPアドレスをそこに記述する。サーバのLAN内IPアドレスを10.0.0.1、PPPのエンドポイントとなるIPアドレスのうちサーバ側のものを10.0.0.254、クライアントに割り当てるものを10.0.0.200とすると、リスト3のようなエントリとする。

リスト３　ppp.conf 内のvpn-１エントリ

```
vpn-1:
  allow user vpn
  set ifaddr 10.0.0.254 10.0.0.200
  set timeout 0
  set log phase chat connect lcp ipcp command
  set escape 0xff
  allow mode direct
  enable proxy
```

表4：VPN認証クライアントの概要

CPU	PowerPC G4 1.33MHz
OS	MacOS 10.3 (Darwin 7.5.0)
PPP実装	pppd
SSH実装	OpenSSH_3.6.1p1+CAN 3.6.1p1

リスト4　ssh起動スクリプトの例

```
#!/bin/sh
ssh -i /etc/ppp/keys/vpn vpn@10.0.0.1 dummy
```

4.2　VPN認証クライアントの設定

表4に示したVPNクライアントでの手順の概略を以下に示す。

1．サーバのVPNユーザとしてログインするときに使用するSSH鍵を作成するPPPを起動できるユーザ権限でssh-keygenコマンドを用い、認証用の鍵を作成する。生成された鍵ファイルが/etc/ppp/keys/vpnであるとする。これに対応する公開鍵ファイル(vpn.pub) はサーバ側に送って登録しておく。

2．1で作成した鍵を利用して認証サーバにログインするためのスクリプトを作成するリスト4を/etc/ppp/start-vpnとして保存する。

3．2のスクリプトをデバイスとしてPPP接続を行なう設定を作成する(PPP実装に応じて異なる)

Mac OS XのPPP実装であるpppdの場合は、コマンドラインで起動スクリプトを指定する。

```
pppd pty /etc/ppp/start-vpn
```

5. 評 価

4.2 の設定により無線LANを経由したVPN接続が可能となる。サーバ上でパケットフィルタリングを施していることにより、単に無線LANの電波を捕捉して同一セグメントとなるIPアドレスを付けた場合にはアクセスポイントより先には一切アクセスできない。これにより盗聴や無断使用等の危険性を排除できる。

ただし、PPP over SSHによるオーバーヘッドがあるため、表5に示したような通信速度低下が見られる[4]。スループットで3倍近くの差が認められるものの、巨大なファイルの転送を行なわない限り実用上支障を感じない速度が得られているといえよう。

定性的に見て最も効果が大きいのは、クライアント側で設定した項目がそのまま外部からのVPN接続に利用できる点である。これまでは、無線LANへの接続とは全く独立したVPN専用ソフトウェアを導入する必要があったが、本稿で提案した手法を用いることで同一の方法でLANへのアクセスが確保できる。また、既存のVPN専用ソフトウェアは対応OSが限られるため「Windowsのみ」という状況がまま見られた。この問題も、PPP over SSHを利用することで、OSを選ばないVPN接続が得られる。

表5 ：scpを利用した通信速度の比較

PPP over SSH	6.6Mbps
WEP (104bits)	17.1Mbps

[4]アクセスポイントとクライアント間でWEPを使用した場合で通信速度を比較した。測定はscpによりサーバ上にある10MBのファイルをクライアントにコピーすることを10回行なった。

6. 結　論

無線LANの物理アドレスを「信頼できない」ものとして、LANへのアクセスをPPP over SSHで確立したVPN経由のものに限定することで、権利のない者の利用を未然に防げることが確認できた。また、個人が利用する代表的なOSにおいては特別なソフトウェアを導入することなく汎用的なものを利用するだけで認証できることから、システムを選ばないという有効性も確認できた。さらには無線LANだけでなく、外部からの接続でも共通の設定でLANへの接続が行なえるので利用者の手間も削減できることが確認できた。

ただし、PPP over SSHを構築するための設定作業が容易なものとは言いがたいという問題点がある。その作業を自動化するシステムが求められるであろう。