Various IPSec convertibility over Layer2 Tunnel in MiM

HAYASHI yu-ichi
Core and Information Technology Center

1 はじめに

MiM[1]において End-to-End のセキュリティを確保することは重要である。現在 IP レベルでのセキュリティを提供する技術として IPSec があげられるが、ここでは IPSec を X.509 を利用した場合のみに言及し、FreeBSD、Linux、Windows2000 における IPSec with X.509 の実相状況を確認し、同時に相互接続性も確認する。またここで IPSec を X.509 を用いた通信のみについて言及するのは ID-Protection[2]の観点からみた安全性において X.509 がもっとも優れているからである。

2 各インプリメンテーションの対応

2．1 FreeBSD(KAME)[3]

先日まで FreeBSD では X.509 はサポートされていないと思っていたがどうやらサポートされているようである[3]。以前マニュアルや Web ページを参照して設定した際には動かなかったことを記憶している。そういった訳で今回は X.509 の接続実験には残念ながら参加はないわけであるが、それもかわいそうなので Pre Shared Secret を利用した接続状況についてここでレポートする。

また以下の接続状況は、暗号化アルゴリズムは IKE のネゴシエーションの際双方で決定してもらうことにしたが、ESP 機密性は 3DES、ESP 整合性は HMAC-MD5 を利用していたようだ。また Phase2 の Quick モードでは原則として PFS を利用することを設定の際加えた。

また上記の IKE のネゴシエーションで利用したポリシーは Pre Shared Secret である。

No	Initiator	Responder	Phase1	接続性	IPSec-mode
1	Win2k	FreeBSD	Main	○	Tunnel
2	FreeBSD	Win2K	Main	○	Tunnel
3	Win2k	FreeBSD	Aggressive		Tunnel
4	FreeBSD	Win2K	Aggressive		Tunnel
5	Win2k	FreeBSD	Main	○	Transport
6	FreeBSD	Win2K	Main	○	Transport
7	Win2k	FreeBSD	Aggressive		Transport
8	FreeBSD	Win2K	Aggressive		Transport
9	Win2k	Linux	Main	○	Tunnel
10	Linux	Win2K	Main	○	Tunnel
11	Win2k	Linux	Aggressive		Tunnel
12	Linux	Win2K	Aggressive		Tunnel
13	Win2k	Linux	Main		Transport
14	Linux	Win2K	Main		Transport
15	Win2k	Linux	Aggressive		Transport
16	Linux	Win2K	Aggressive		Transport
17	FreeBSD	Linux	Main	○	Tunnel
18	Linux	FreeBSD	Main	○	Tunnel
19	FreeBSD	Linux	Aggressive		Tunnel
20	Linux	FreeBSD	Aggressive		Tunnel
21	FreeBSD	Linux	Main	○	Transport
22	Linux	FreeBSD	Main	○	Transport
23	FreeBSD	Linux	Aggressive		Transport
24	Linux	FreeBSD	Aggressive		Transport

表: Pre Shared Secret を利用した IPSec の相互接続性

上記の接続性の部分で空欄になっている場所は未実験であることをしめしている。実験を行い接続が確立されたものに関しては是非ご報告いただきたい。

また各 OS の実相は、Windows2000：ローカルセキュリティーポリシー、Linux：FreeS/WAN、FreeBSD：KAME を利用した。

2．2 Linux(FreeS/WAN)[4]

Linux における IPSec のインプリメンテーション FreeS/WAN は X.509 をもちいた IPSec を正式サポートしているわけではない。現在はパッチを当てて対応するということになっている[5]。ここでは簡単に X.509 の設定方法を述べるとともに MiM における Window2000 との相互接続性について述べる。

まず X.509 のパッチをサイトからダウンロードし、カーネルを再構築する。ここまでは Linux を利用したことがある人なら簡単に行うことができる。あくまで正式サポートではないので FreeS/WAN が正式にサポートしてもらえる日と夢見て今回はパッチを利用するものとする。また際当然あると思って設定を進めた際なかったディレクトリがあるのでここではそれを記述することにする。FreeS/WAN では rootCA の Certificate を格納する”/etc/ipsec.d/cacerts” と crl を格納する ”/etc/ipsec.d/crls” が存在しないため X.509 のパッチを当てたあとに作成しておくことを推奨する。 また後述になってしまったが “/etc/ipsec.d”もはじめから存在していないので、これも各自作成してほしい。あくまでこのディレクトリは IKE が参照するの

で IKE を動かす権限でパーミッションを設定することを推奨する。FreeS/WAN の設定ファイルに関しては付属のドキュメントや Web 上に主種多様な設定例があるのでここでは割愛する。Windows2000 との X.509 を利用しての相互接続性については Windows2000 の項を参照されたし。

2．3　Windows2000（IPSec Policy）

一度でも Windows で IPSec を利用したユーザならその GUI の設定のしにくさ、そして Debug のしにくさにうんざりすることだろう。ましてや日頃から CUI での設定をしているユーザにとっては決して設定を行いたくないたぐいのものであることは間違いない。そこで今回は Windows の管理ツールの中にあるローカルセキュリティーポリシーによる GUI の設定は利用せずマイクロソフトが提供する Ipsecpol(Internet Protocol Security Policies Tool)[6]を利用して CUI による設定を行うものとする。なおさらにこの Ipsecpol を FreeS/WAN 風に利用できる Wrapper Tool[7] がありこれを利用することで Windows2000 での IPSec を FreeS/WAN と同じ設定で利用することができる。

上記を用いて今回は MiM の中で利用する場合を想定し、Phase1 Main-mode, SPD, SA は IKE 同士のネゴシエーションで設定し、Phase2 Quick-mode では PFS を有効にした。また、今回は MiM の構成上 Tunnel-mode を利用して実験を行った。実験構成図[8]では途中に利用する IPSec 以外の Tunnel プロトコルは L2TP と VTUN[9] の２つを利用した。

No	Initiator	Responder	Tunnnel	接続性
1	Win2K	Linux	VTUN	×
2	Linux	Win2K	VTUN	×
3	Win2K	Win2K	VTUN	○
4	Linux	Linux	VTUN	
5	Linux	Win2K	L2TP	
6	Win2K	Linux	L2TP	
5	Win2K	Win2K	L2TP	○
6	Linux	Linux	L2TP	

表: IPSec over L2TP or VTUN の相互接続性

上記の IPSec over L2TP or VTUN による over head は次のようになる。

VTUN	IPSec	ESP	IP	Data	IPSec	IPSec
Header	Tunnel Header	Header	Header	ペイロード	ESP トレーラ	認証トレーラ

図: VTUN による over head

L2TP Header	IPSec Tunnel Header	ESP Header	IP Header	Data ペイロード	IPSec ESP トレーラ	IPSec 認証トレーラ

図: L2TP による over head

VTUN での相互接続性が確立できなかった点だが、Windows2000 側から Linux 側へは isakmp のネゴシエーションの為のパケットが流れるが、Linux 側からのパケットはいっさい流れなかった。（現在原因解明中）L2TP と通した Linux 同士の接続実験もまだ行っていない。

3　まとめ

MiM のアーキテクチャーでは現在利用できる X.509 を用いた通信は Windows2000 での IPSec の通信のみである。L2TP を越えた通信に障害がでる IKE は実装上問題があるのか、IPSec のプロトコルと L2TP や VTUN のプロトコルの間に問題があるのかもしれない、これらを注意深く実験し、接続性を確かめていくことが今後の課題である。

参考文献

[1] A.Kara “Protecting the Security of End-to-End Communications in NAT-divided Internet Environment” 2002年　電子情報通信学会総合大会

[2] Y.Hayashi “ID-Protection for IPSec Phase1” CITEC Archival report

[3] KAME Project : http://www.kame.net

[4] Linux FreeS/WAN: http://www.freeswan.org/

[5] X.509 patch: http://www.strongsec.com/freeswan/

[6]Ipsecpol:http://agent.microsoft.com/windows2000/techinfo/reskit/tools/existing/ipsecpol-o.asp

[7] win2k ipsec tool: http://vpn.ebootis.de/

[9] VTUN:http://vtun.sourceforge.net/

[8]実験構成図