

IPsec interoperability tests with respect to deployment as “Last Mile” security solution

Petr Holub

2002-02-28

1 Abstract

This report summarizes interoperability tests between KAME based FreeBSD, Windows 2000 and FreeS/WAN Linux IPsec implementation. Furthermore I am developing concept of securing wireless last mile solution which is by its nature usually most susceptible to security vulnerabilities like eavesdropping.

2 Introduction

I have targeted my effort at IPsec[2] interoperability testing because this technology can be deployed in last mile connectivity as security solution. Most likely this is convenient for wireless networks 802.11b that are quite insecure even when using WEP encryption[7][8].

I have picked up the topic studied by other CESNET group[10] two years ago. Since that work addressed Linux IPsec implementation and its interoperability issues I haven't included these issues in my report. I have done similar tests for KAME based implementation and I have developed several models for last mile security solution based on results of these tests.

IPsec can be run in two basic modes:

- transport
- tunnel

Transport mode can be used for end-to-end security while tunnel mode means secure tunneling over insecure part of the network.

IPsec can use keys as follows:

- manually set keys
- automatically negotiated keys using Internet Key Exchange (IKE) protocol
 - using pre-shared key (shared secret)
 - X.509 certificates sent either by client or stored in DNS (Secure DNS)

3 Possible scenarios

3.1 End-to-end security

End-to-end scenario uses transport mode for securing the whole path between two communicating machines. For my purposes this is not very useful solution.

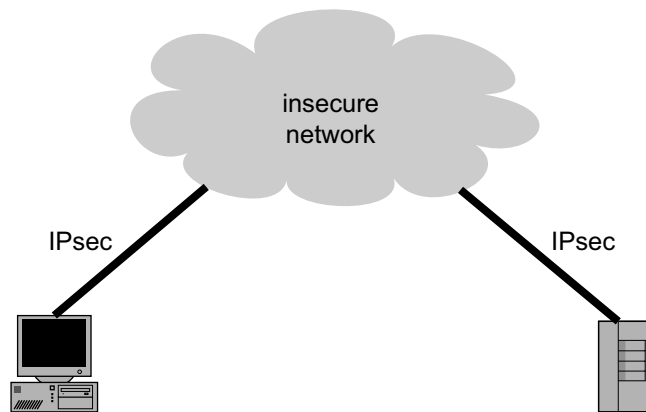


Figure 1: End-to-end security scenario

3.2 Road-Warrior scenario for client security

Let's have a client that's connected to wireless network. Wireless network can be protected by WEP encryption but this doesn't provide reasonable protection against an attack led from one computer already participating in such a network. In this scenario I want to have all the traffic from client over insecure part of network (in my case the wireless part) encrypted using IPsec. After passing gateway traffic gets decrypted. On it way back traffic gets encrypted on the gateway and travels back to the client. Furthermore each client has it own unique pre-shared key or certificate and clients don't process any unencrypted traffic.

This scenario can be created either using gateway located behind the access-point (picture 2) or creating IPsec capable access-point (picture 3) e.g. using wireless card inserted in computer capable of forwarding packets between interfaces (e.g. using router software).

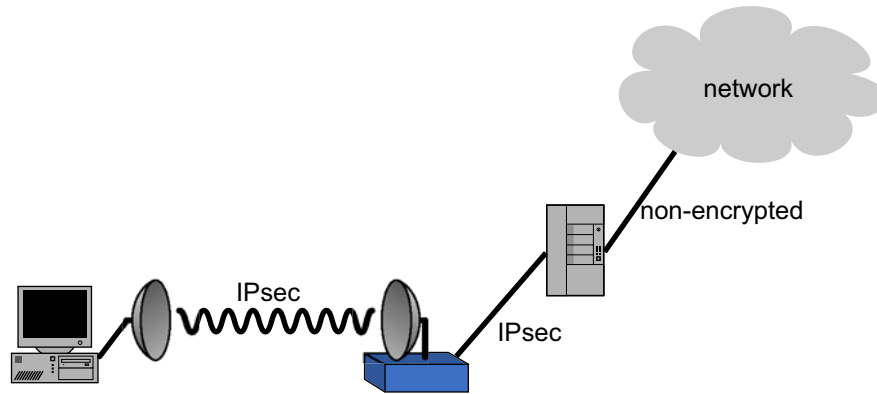


Figure 2: Road-Warrior scenario for client security scenario (1)

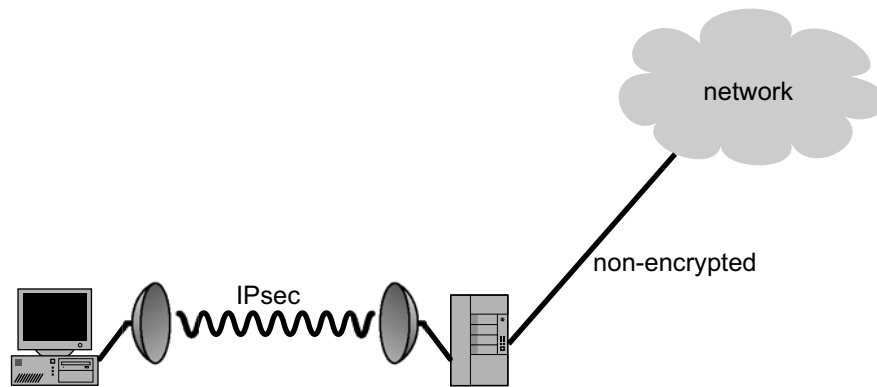


Figure 3: Road-Warrior scenario for client security scenario (2)

5 Tests and measurements

5.1 End-to-end security

5.1.1 Setup

IPsec configuration (*/etc/ipsec.conf*):

```
spdadd 192.168.3.38 192.168.3.44 any -P out ipsec
esp/transport//use;
spdadd 192.168.3.44 192.168.3.38 any -P in ipsec
esp/transport//use;
spdadd 192.168.3.38 192.168.3.20 any -P out ipsec
esp/transport//use;
spdadd 192.168.3.20 192.168.38 any -P in ipsec
esp/transport//use;
```

This configuration can be loaded using following command: *setkey -f /etc/ipsec.conf*
Similar configurations are used on remaining two machines.

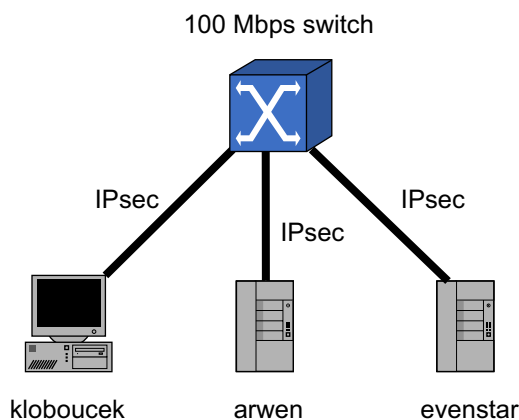


Figure 5: End-to-end security testbed setup

Racoon configuration (*/usr/local/etc/racoon/racoon.conf*):

```
path include "/usr/local/etc/racoon" ;
path pre_shared_key "/usr/local/etc/racoon/psk.txt" ;
path certificate "/usr/local/etc/cert" ;

padding
{
    maximum_length 20;      # maximum padding length.
```

```

        randomize off;           # enable randomize length.
        strict_check off;       # enable strict check.
        exclusive_tail off;     # extract last one octet.
    }

listen
{
    #isakmp ::1 [7000];
    #isakmp 202.249.11.124 [500];
    #admin [7002];             # administrative's port by kmpstat.
    #strict_address;          # required all addresses must be bound.
}

timer
{
    # These value can be changed per remote node.
    counter 5;                # maximum trying count to send.
    interval 20 sec;          # maximum interval to resend.
    persend 1;                # the number of packets per a send.

    # timer for waiting to complete each phase.
    phase1 30 sec;
    phase2 15 sec;
}

remote anonymous
{
    exchange_mode aggressive,main;
    doi ipsec_doi;
    situation identity_only;

    #my_identifier address;
    #my_identifier user_fqdn "sakane@kame.net";
    #peers_identifier user_fqdn "sakane@kame.net";
    #certificate_type x509 "mycert" "mypriv";

    nonce_size 16;
    lifetime time 10 min;     # sec,min,hour
    initial_contact on;
    support_mip6 on;
    proposal_check obey;     # obey, strict or claim

    proposal {
        encryption_algorithm 3des;
    }
}

```

```

        hash_algorithm md5;
        authentication_method pre_shared_key ;
        dh_group 2 ;
    }
}

sainfo anonymous
{
    pfs_group 1;
    lifetime time 600 sec;
    encryption_algorithm 3des,des,cast128,blowfish ;
    authentication_algorithm hmac_sha1,hmac_md5;
    compression_algorithm deflate ;
}

```

The same racoon configuration will be used in subsequent tests and setups. The only difference is change in file containing pre-shared key.

5.1.2 Performance

Table 1 summarizes results of performance measurements of KAME IPsec stack on differently powerful machines. Measurements have been done using *netperf* tool which runs tests between pair of machines. In this pair one machine acts as traffic generator and the other one as replying server.

	arwen	evenstar
intial delay:	0.2-0.6 ms	0.2 ms
<i>netperf</i>		
TCP_STREAM	3,514.35 kbps	21,382.09 kbps (16,864.51 kbps)
TCP_STREAM (n-e)	57,358.72 kbps	94124.03 kbps
UDP_STREAM (to kouboucek)	3,886.98 kbps	33,092.48 kbps
UDP_STREAM (from kloboucek)	24,587.70 kbps	24,474.49 kbps (19,456.85 kbps)
UDP_STREAM (to kloboucek, n-e)	59,639.57 kBps	95,636.25 kbps
UDP_STREAM (from kloboucek, n-e)	95,742.14 kbps	95,726.72 kbps

Table 1: IPsec performance overview on FreeBSD.

Measurement parameters:

- neperf version: 2.1.3
- Confidence level: 99%

- Confidence interval: 2,5%
- Command: `./netperf -i 2,3 -f k -H hostname -t TCP_STREAM`

TCP_STREAM test measures real TCP data throughput between two machines. UDP_STREAM tests rather upper limit of the machine ability to push data on the network. Therefore this test can give very different result when running on each of the pair of testing machine while TCP_STREAM test should give identical results. Notebook called kloboucek was tested in two modes:

parameter	kloboucek	evenstar	arwen
processor	Pentium III 900 MHz (700 MHz)	Athlon 1200 MHz	Pentium MMX 233 MHz
RAM	256 MB	512 MB	128 MB
NIC	RealTek 8139	SMC EtherPower II	SMC 9332DST 21140

Table 2: Hardware configuration of testing machines.

- when running on AC – processor frequency was 900 MHz
- when running on battery – processor frequency was 700 MHz

Since values for these two modes differ only when using IPsec, values for 700 MHz operation are given in brackets.

5.2 Road-Warrior scenario for client security

5.2.1 Setup

This setup is most easily done dedicating one subnet for wireless connected clients. Gateway `evenstar` located behind the access-point can work as router as well. Let us summarize the setup:

- wireless network address range 192.168.1.0/24
- gateway address 192.168.1.1
- client address 192.168.1.3

IPsec configuration on client:

```
spdadd 192.168.1.3 0.0.0.0/0 any -P out ipsec
esp/tunnel/192.168.1.3-192.168.1.1/require;
spdadd 0.0.0.0/0 192.168.1.3 any -P in ipsec
esp/tunnel/192.168.1.1-192.168.1.3/require;
```

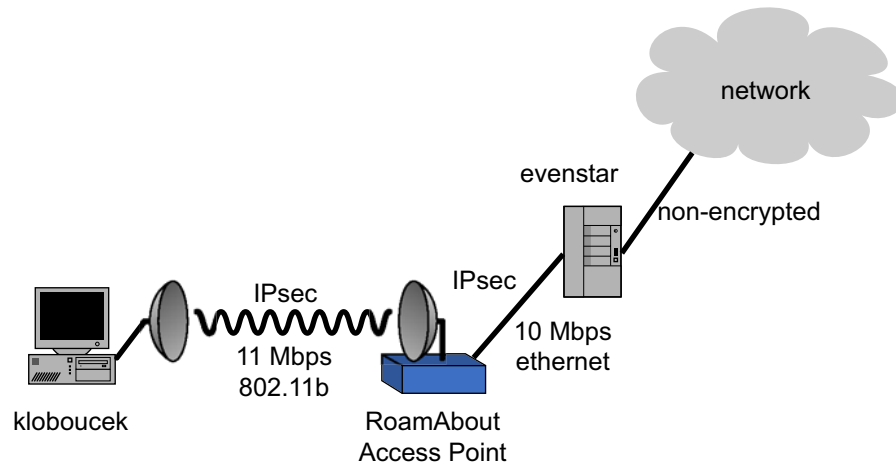


Figure 6: Road-Warrior scenario for client security testbed setup

IPsec configuration on gateway:

```
spdadd 0.0.0.0/0 192.168.1.0/24 any -P out ipsec
esp/tunnel/192.168.1.1-192.168.1.3/require;
spdadd 192.168.1.0/24 0.0.0.0/0 any -P in ipsec
esp/tunnel/192.168.1.3-192.168.1.1/require;
```

5.2.2 Performance

According to table 1 on any reasonably new machine the performance is rather limited by 802.11b maximum connection speed of 11 Mbps.

5.3 Road-Warrior scenario for securing of traffic out of internal network

5.3.1 Setup

Gateway configuration can look in similar way as shown in picture². Namely:

- Internal network connected behind the wireless connection is 192.168.1.0/24
- We want to have encrypted only the outbound traffic going through the wireless connection
- Gateway A from internal network has address 192.168.1.3 (this gateway acts as one endpoint of IPsec tunnel)

²*obr:testbed03*

- Access-point has address 192.168.1.2 (if we want to talk to access-point we don't want to use IPsec since access-point doesn't understand IPsec)
- Gateway B located behind the access-point from the internal network view has address 192.168.1.1 (this gateway acts as the other endpoint of IPsec tunnel; it can perform NAT and other services for masquerading and routing of internal network)

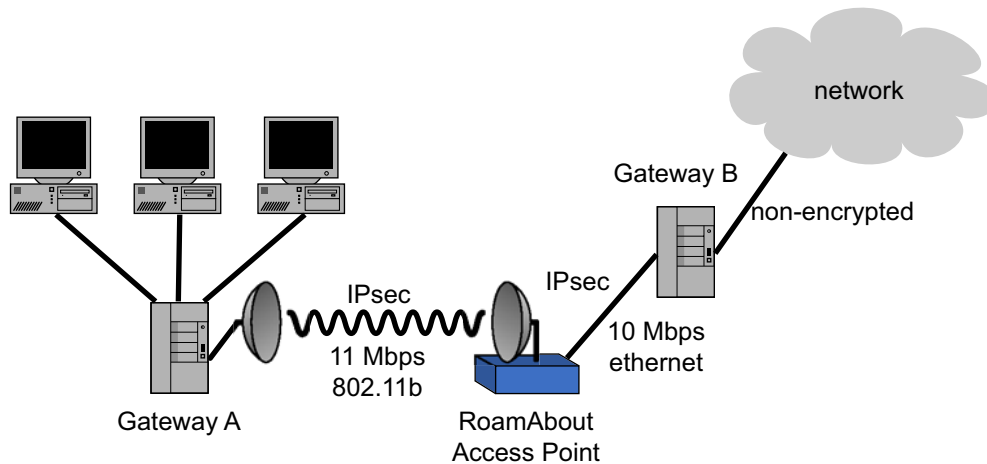


Figure 7: Road-Warrior scenario for client security testbed setup

Configuration for gateway A:

```

spdadd 192.168.1.3 192.168.1.1 any -P out ipsec
esp/tunnel/192.168.1.3-192.168.1.1/require;
spdadd 192.168.1.1 192.168.1.3 any -P in ipsec
esp/tunnel/192.168.1.1-192.168.1.3/require;
spdadd 192.168.1.3 192.168.1.0/24 any -P out none;
spdadd 192.168.1.0/24 192.168.1.3 any -P in none;
spdadd 192.168.1.3 0.0.0.0/0 any -P out ipsec
esp/tunnel/192.168.1.3-192.168.1.1/require;
spdadd 0.0.0.0/0 192.168.1.3 any -P in ipsec
esp/tunnel/192.168.1.1-192.168.1.3/require;

```

Configuration for gateway B:

```

spdadd 0.0.0.0/0 192.168.1.3 any -P out ipsec
esp/tunnel/192.168.1.1-192.168.1.3/require;
spdadd 192.168.1.3 0.0.0.0/0 any -P in ipsec
esp/tunnel/192.168.1.3-192.168.1.1/require;

```

6 Interoperability issues

6.1 Windows 2000

6.1.1 Native Windows 2000 IPsec

First I have tested Windows 2000 native implementation of IPsec. This implementation features both transport and tunnel mode. When configuring tunnel mode I would recommend to follow guidelines described in Microsoft Knowledge Base article[6].

I have tested both transport and tunnel mode successfully. The only deficiency I have found is that it is not allowed to specify 0.0.0.0/0 as outbound traffic destination and inbound traffic source while using tunnel mode, i.e. this implementation does not fit into Road-Warrior for client security scenario³.

6.1.2 PGPnet

I have tested PGPnet that comes with free version of PGP 7.0.3 [12]. This free version of PGPnet doesn't fit into Road-Warrior scenario for it runs only in transport mode. According to help information included with the free version commercial version could solve this⁴.

PGPnet also suffers from severe problems with changing network interfaces which is quite usual situation in mobile computing. Generally I would recommend using native Windows 2000 implementation.

7 Conclusions

I have suggested and tested several IPsec based security models for last mile connectivity. It has been shown that for 802.11b networks the performance is limited by bandwidth and not by IPsec processing demands.

References

- [1] IP Security Protocol,
<http://www.ietf.org/ids.by.wg/ipsec.html>,
<http://www.ietf.org/html.charters/ipsec-charter.html>

³It is required to specify network source/destination and its mask at least according to A/B/C network specification, i.e. netmask for network 124.0.0.1 has to be at least 255.0.0.0.

⁴According to documentation which seems to be common for both free and commercial versions there is some feature called gateway that should be able to run in tunnel mode.

- [2] Kame Project,
<http://www.kame.net/>
- [3] FreeS-WAN Project,
<http://www.freeswan.org/>
- [4] FreeBSD IPsec mini-HOWTO,
<http://www.x-itec.de/projects/tuts/ipsec-howto.txt>
- [5] Step-by-Step Guide to Internet Protocol Security (IPSec),
<http://www.microsoft.com/windows2000/techinfo/planning/security/ipsecsteps.asp>
- [6] How to Configure IPSec Tunneling in Windows 2000,
<http://support.microsoft.com/support/kb/articles/q252/7/35.asp>
- [7] Stein J., *Replacing WEP With IPsec*,
http://rt.fm/jcs/ipsec_wep.html
- [8] Security of the WEP algorithm,
<http://www.isaac.cs.berkeley.edu/isaac/wep-faq.html>
- [9] DeGraw-Bertsch M., *IPsec Tunneling Between FreeBSD Hosts*,
OnLamp, O'Reilly Network, 12/10/2001.
<http://www.onlamp.com/pub/a/bsd/2001/12/10/ipsec.html>
- [10] Rudolf V., Cizek J., *IPSec pro IPv4 v Linuxu*,
technical report 7/2000, CESNET (Czech only)
<http://www.cesnet.cz/doc/techzpravy/2000-7/>
- [11] VPN through IPSec protocol Howto,
<http://www.toxiclinux.org/vpn.html>
- [12] Pretty Good Privacy – PGP,
<http://www.pgpi.org/>