基于 FreeS/WAN IPSec 配置手册

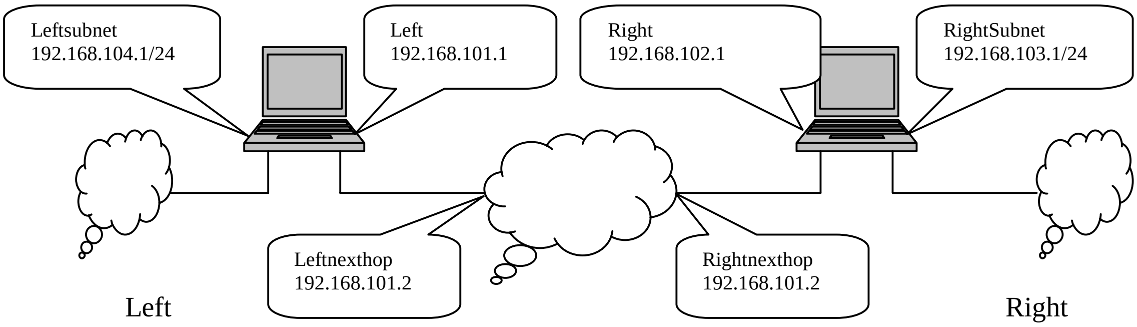
胡建斌

目 录

一　拓扑结构

Linux 端：Linux inter-102-59 2.4.7-10custom ＋ Freeswan 1.97

Windows 端：Windows 2k Profession + SP2 + ipsecpol + ipsec 工具



二 各种配置情况

1. 采用预共享密钥的 Gateway－Gateway 连接方式

1) /etc/ipsec.conf

```
# /etc/ipsec.conf - FreeS/WAN IPsec configuration file
# More elaborate and more varied sample configurations can be found
# in FreeS/WAN's doc/examples file, and in the HTML documentation.

# basic configuration
config setup
        # THIS SETTING MUST BE CORRECT or almost nothing will work;
        # %defaultroute is okay for most simple cases.
        interfaces=%defaultroute
        # Debug-logging controls:    "none" for (almost) none, "all" for lots.
        klipsdebug=none
        plutodebug=none
        # Use auto= parameters in conn descriptions to control startup actions.
        plutoload=%search
        plutostart=%search
        # Close down old connection when new one using same ID shows up.
        uniqueids=yes

# defaults for subsequent connection descriptions
# (these defaults will soon go away)
conn %default
        keyingtries=0
        disablearrivalcheck=no
        #authby=rsasig
        #leftrsasigkey=%dnsondemand
        #rightrsasigkey=%dnsondemand

# connection description for opportunistic encryption
# (requires KEY record in your DNS reverse map; see doc/opportunism.howto)
conn me-to-anyone
        left=%defaultroute
        right=%opportunistic
        keylife=1h
        rekey=no
        # for initiator only OE, uncomment and uncomment this
        # after putting your key in your forward map
        #leftid=@myhostname.example.com
        # uncomment this next line to enable it
        #auto=route

# sample VPN connection
conn sample3
        # Left security gateway, subnet behind it, next hop toward right.
        left=192.168.101.1
        #leftsubnet=192.168.104.1/24
        #leftnexthop=10.22.33.44
        # Right security gateway, subnet behind it, next hop toward left.
        right=192.168.102.1
        #rightsubnet=192.168.0.0/24
        #rightnexthop=10.101.102.103
        # To authorize this connection, but not actually start it, at startup,
        # uncomment this.
        auto=start
        keyingtries=0
        spi=0x200
        esp=3des-md5-96
        espenckey=0x01234567_89abcdef_02468ace_13579bdf_12345678_9abcdef0
        espauthkey=0x12345678_9abcdef0_2468ace0_13579bdf
```

2) /etc/ipsec.secrets

```
# This file holds shared secrets or RSA private keys for inter-Pluto
# authentication.   See ipsec_pluto(8) manpage, and HTML documentation.

# RSA private key for this host, authenticating it to any other host
# which knows the public part.   Suitable public keys, for ipsec.conf, DNS,
# or configuration of other implementations, can be extracted conveniently
# with "ipsec showhostkey".

192.168.101.1  192.168.102.1:  PSK   "jxj52SjRmUu3nVW521Wu135R5k44uU5lR2V3kujT24U1lVu
mWSkT52Tu11WVnm1Vu25lV52k4"
```

3) Windows 端 ipsec.conf 文件

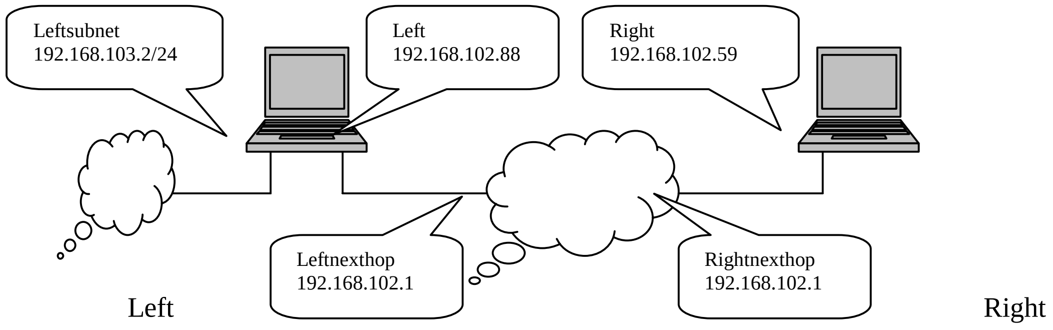
```
# /etc/ipsec.conf - FreeS/WAN IPsec configuration file
# More elaborate and more varied sample configurations can be found
# in FreeS/WAN's doc/examples file, and in the HTML documentation.

# basic configuration
config setup
        # THIS SETTING MUST BE CORRECT or almost nothing will work;
        # %defaultroute is okay for most simple cases.
        interfaces=%defaultroute
        # Debug-logging controls:   "none" for (almost) none, "all" for lots.
        klipsdebug=none
        plutodebug=none
        # Use auto= parameters in conn descriptions to control startup actions.
        plutoload=%search
        plutostart=%search
        # Close down old connection when new one using same ID shows up.
        uniqueids=yes

# defaults for subsequent connection descriptions
# (these defaults will soon go away)
conn %default
        keyingtries=0
        disablearrivalcheck=no
        authby=rsasig

# sample VPN connection
conn sample3
        # Left security gateway, subnet behind it, next hop toward right.
        left=192.168.102.59
        #leftsubnet=192.168.102.251
        #leftnexthop=10.22.33.44
        # Right security gateway, subnet behind it, next hop toward left.
        right=192.168.102.251
        #rightsubnet=192.168.0.0/24
        #rightnexthop=10.101.102.103
        # To authorize this connection, but not actually start it, at startup,
        # uncomment this.
        auto=start
        keyingtries=0
        spi=0x200
        esp=3des-md5-96
        espenckey=0x01234567_89abcdef_02468ace_13579bdf_12345678_9abcdef0
        espauthkey=0x12345678_9abcdef0_2468ace0_13579bdf
        presharedkey="jxj52SjRmUu3nVW521Wu135R5k44uU5lR2V3kujT24U1lVumWSkT52Tu11WVnm1Vu25lV52k4"
        network=lan；这是只有 windows 上才有的语句，必须加上！!!
        auto=start
        pfs=yes
```

2. 采用预共享密钥的 Net-Gate 连接方式



1) Gateway 主机上的网络路由配置

```
[Left Gateway]# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref     Use Iface
192.168.102.0   *               255.255.255.0   U     0      0         0 eth0
192.168.103.0   *               255.255.255.0   U     0      0         0 eth1
default         192.168.102.1   0.0.0.0         UG    0      0         0 eth0
```

```
[Right Gateway]# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref     Use Iface
192.168.102.0   *               255.255.255.0   U     0      0         0 eth0
127.0.0.0       *               255.0.0.0       U     0      0         0 lo
default         192.168.102.1   0.0.0.0         UG    0      0         0 eth0
```

2) Left/Right Gateway 配置

■ /etc/ipsec.conf

```
# /etc/ipsec.conf - FreeS/WAN IPsec configuration file
# More elaborate and more varied sample configurations can be found
# in FreeS/WAN's doc/examples file, and in the HTML documentation.

# basic configuration
config setup
        # THIS SETTING MUST BE CORRECT or almost nothing will work;
        # %defaultroute is okay for most simple cases.
        interfaces="ipsec0=eth0"
        # Debug-logging controls:   "none" for (almost) none, "all" for lots.
        klipsdebug=none
        plutodebug=none
        # Use auto= parameters in conn descriptions to control startup actions.
        plutoload=%search
```

```
            plutostart=%search
            # Close down old connection when new one using same ID shows up.
            uniqueids=yes
# defaults for subsequent connection descriptions
# (these defaults will soon go away)
conn %default
            keyingtries=0
            #disablearrivalcheck=no
            #authby=rsasig
            #leftrsasigkey=%dnsondemand
            #rightrsasigkey=%dnsondemand

# defaults for subsequent connection descriptions
conn %default
       # How persistent to be in (re)keying negotiations (0 means very).
       keyingtries=0
       # Parameters for manual-keying testing (DON'T USE OPERATIONALLY).
       spi=0x200
       esp=3des-md5-96
       espenckey=0x01234567_89abcdef_02468ace_13579bdf_12345678_9abcdef0
       espauthkey=0x12345678_9abcdef0_2468ace0_13579bdf

# sample VPN connection
conn sample3
            # Left security gateway, subnet behind it, next hop toward right.
            left=192.168.102.88
            leftsubnet=192.168.103.0/24
            #leftfirewall=yes
            #leftnexthop=192.168.102.1
            # Right security gateway, subnet behind it, next hop toward left.
            right=192.168.102.59
            #rightsubnet=0/0
            #rightnexthop=10.101.102.103
            # To authorize this connection, but not actually start it, at startup,
            # uncomment this.
            auto=start
            #auto=route
            keyingtries=0
            spi=0x200
            esp=3des-md5-96
            espenckey=0x01234567_89abcdef_02468ace_13579bdf_12345678_9abcdef0
            espauthkey=0x12345678_9abcdef0_2468ace0_13579bdf
```

- ### ipsec.secrets 文件

```
192.168.102.88 192.168.102.59 : PSK "jxj52SjRmUu3nVW521Wu135R5k44uU5lR2V3kujT24U
1lVumWSkT52Tu11WVnm1Vu25lV52k4"
```

3) Left /Right Gateway 启动 IPSEC

```
＃ipsec setup restart
＃ipsec manual –up sample3（可不执行）
```

4) 启动 IPSec 后的路由及网络状况

[Left Gateway]# route

```
Kernel IP routing table
Destination     Gateway         Genmask          Flags  Metric  Ref     Use Iface
192.168.102.59  192.168.102.59  255.255.255.255  UGH    0       0       0 ipsec0
192.168.102.0   *               255.255.255.0    U      0       0       0 eth0
192.168.102.0   *               255.255.255.0    U      0       0       0 ipsec0
192.168.103.0   *               255.255.255.0    U      0       0       0 eth1
default         192.168.102.1   0.0.0.0          UG     0       0       0 eth0
```

[Right Gateway]# route

```
Kernel IP routing table
Destination     Gateway         Genmask          Flags Metric  Ref     Use Iface
192.168.102.0   *               255.255.255.0    U     0       0       0 eth0
192.168.102.0   *               255.255.255.0    U     0       0       0 ipsec0
192.168.103.0   192.168.102.88  255.255.255.0    UG    0       0       0 ipsec0
127.0.0.0       *               255.0.0.0        U     0       0       0 lo
default         192.168.102.1   0.0.0.0          UG    0       0       0 eth0
```

5) 说明

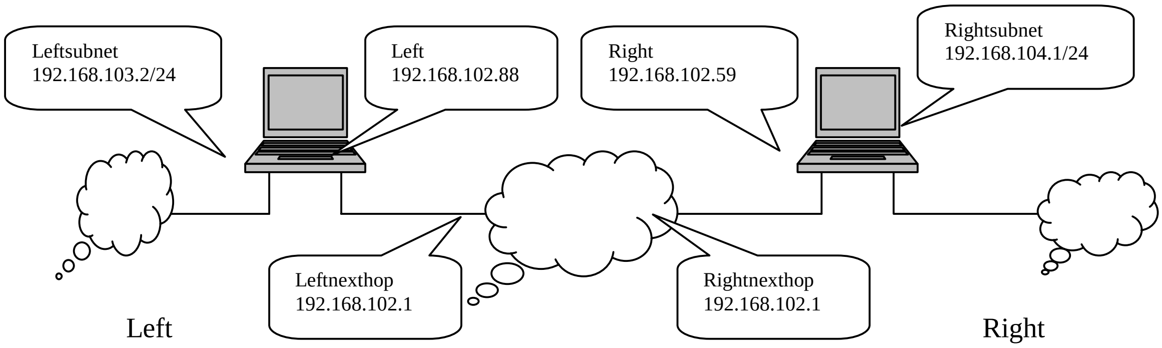
- left gateway 上有 eth0: 192.168.102.59   eth1:192.168.103.2，其中 eht1 作为 192.168.103.0/24 网段的默认路由来设置，但是没有必要在该主机上做 MASQ 或 NAT，因为 IPSec 能处理这样的问题；
- 在 gateway 上最好将 ip_forward 打开: echo 1 >/proc/sys/net/ipv4 ip_forward
- left gateway(192.168.102.59)和 right gateway(192.168.102.88)之间直接用 ping 是不能互通的;
- right gateway 可 ping 通 192.168.103.2 以及 192.168.103.0/24 网段内的所有主机互通；
- 如果想在 left gateway 上设置成 192.158.103.0/24 网段上的机器都通过 MASQ 后通过默认的网管访问另外的地方，则可以如下设置：

```
# /sbin/ipchains -A forward -j ACCEPT -i eth1 -s 192.168.103.0/24 -d 192.168.103.0/24
# /sbin/ipchains -A forward -j MASQ -i eth0 -s 192.168.103.0/24
```

- 通过如下命令可以查看两个网关之间的通信情况：

```
# tcpdump –i ipsec0(或 eth0)   host 192.168.102.59 and 192.168.102.88
```

3. 采用预共享密钥的 Net-Net 连接方式



1) /etc/ipsec.conf

修改如下语句即可：

```
 left=192.168.102.88
leftsubnet=192.168.103.0/24
right=192.168.102.59
rightsubnet=192.168.104.0/24
```

2) 启动 IPSec 后的路由及网络状况

[Left Gateway]# route

```
Kernel IP routing table
Destination     Gateway          Genmask          Flags Metric Ref    Use   Iface
192.168.102.0    *                255.255.255.0    U     0       0     0    eth0
192.168.102.0    *                255.255.255.0    U     0       0     0    ipsec0
192.168.103.0    *                255.255.255.0    U     0       0     0    eth1
192.168.104.0    192.168.102.59  255.255.255.0    UG    0       0     0    ipsec0
default          192.168.102.1   0.0.0.0          UG    0       0     0    eth0
```

[Right Gateway]# route

```
Kernel IP routing table
Destination     Gateway          Genmask          Flags Metric Ref    Use Iface
192.168.102.0    *                255.255.255.0     U     0       0   0    eth0
192.168.102.0    *                255.255.255.0     U     0       0   0    ipsec0
192.168.103.0    192.168.102.88  255.255.255.0    UG     0       0   0    ipsec0
192.168.104.0    *                255.255.255.0     U     0       0   0    eth1
127.0.0.0        *                255.0.0.0         U     0       0   0    lo
default          inter-102-1.jad  0.0.0.0          UG     0       0   0   eth0
```

3) 说明

- 两个 Gateway 之间相互无法 ping 通，也无法 ping 通各个内部子网 IP;
- 一定注意路由的设置：
- 当网关之间进行协商时一定要指定通过 Ipsec0
- 当访问外部非网关机器时一定要通过 eth0 而不是 Ipsec0

- 注意路由表里的路由项有先后的顺序关系
- 必要时如果不能进行协商的话可以适当删除 eth0 项：

```
route del –net 192.168.102.0 netmask 255.255.255.0 dev eth0
```

适当的时候再进行添加。

- 很多时候都是路由的问题，切记！

- 注意防火墙的设置，有时候可能是防火墙过滤了数据包导致失败

4. 采用预共享密钥的混合模式连接方式

修改 /etc/ipsec.conf 文件：

```
conn left-Gateway—right-Gateway
        left=192.168.102.88
        right=192.168.102.59
        auto=start
        keyingtries=0
        spi=0x200
        esp=3des-md5-96
        espenckey=0x01234567_89abcdef_02468ace_13579bdf_12345678_9abcdef0
        espauthkey=0x12345678_9abcdef0_2468ace0_13579bdf
conn left-net—right-Gateway
         left=
         leftsubnet=
         right=
         ……..
conn left-Gateway—right-net
         left=
         right=
         rightsubnet=
         ……..
conn left-net—right-Gateway
         left=
         leftsub=
         right=
         rightsubnet=
         ……..
```

这样会在两个网关之间建立多个连接，可包含以上的各种情况。

5. Raw RSA Authentication Configuration

1）生成密钥文件

```
# ipsec rsasigkey --verbose 1024 > keyfile
getting 64 random bytes from /dev/random...
```

```
looking for a prime starting there (can take a while)...
found it after 87 tries.
getting 64 random bytes from /dev/random...
looking for a prime starting there (can take a while)...
found it after 550 tries.
computing modulus...
computing lcm(p-1, q-1)...
computing d...
computing exp1, exp1, coeff...
output...
```

```
#cat keyfile
 #  RSA 1024 bits      black      Tue Sep 24 14:08:49 2002
            # for signatures only, UNSAFE FOR ENCRYPTION

#pubkey=0sAQNzVs3ajlqvuiztMRtd0GgLG6cvkPCfjAaTVgAHZ+i+SGzfzg79uD6TM3SV+
n8L2LnVRK7+xSlUn3h1hz+df9FU9EENY2MF12X6Wb/bm82BVgbMm05LnA9G30qSQr7
UcDp0Ozu54KKekNhYrGXCWIjY8xUQNeYBzK2HE/ed/CNHUQ==
#IN KEY 0x4200 4 1
AQNzVs3ajlqvuiztMRtd0GgLG6cvkPCfjAaTVgAHZ+i+SGzfzg79uD6TM3SV+n8L2LnVRK7+
xSlUn3h1hz+df9FU9EENY2MF12X6Wb/bm82BVgbMm05LnA9G30qSQr7UcDp0Ozu54KKek
NhYrGXCWIjY8xUQNeYBzK2HE/ed/CNHUQ= =
            # (0x4200 = auth-only host-level, 4 = IPSec, 1 = RSA)
            Modulus:
0x7356cdda8e5aafba2ced311b5dd0680b1ba72f90f09f8c069356000767e8be486cdfce0efdb83e9
3337495fa7f0bd8b9d544aefec529549f7875873f9d7fd154f4410d636305d765fa59bfdb9bcd8156
06cc9b4e4b9c0f46df4a9242bed4703a743b3bb9e0a29e90d858ac65c25888d8f3151035e601ccad
8713f79dfc234751
            PublicExponent: 0x03
            # everything after this point is secret
            PrivateExponent:
0x04ce4893c5ee71fd17348cb6793e0455cbd1a1fb5f5bfb2af0ce40004eff07edaf33fdeb4a9257f0
cccf863fc54b2907be3831f548370e314faf904d513aa8b8d093d4bfde2b85d3d9261c306343884b
ac849109984bf7bf0542a89eeadb3a1bca3ceb9955a6758c701bb8be31934947e6365f90ce2bc525
df5858133e38f46b
            Prime1:
0xe60a29183893abc4006d5d0538552c0c7e355e7607f3ac0dc7e75b226be816e13c6b863c74fd0a
abce3dc8460938a742b60b4fc173be3c3494ed343beb224961
            Prime2:
0x805af24e565d9fc59e59be4d15238e315c29a3f1fc8929509923683a4e5ee6be42199f1d640a8eb
a8781924b135102d8a7f0cae11e2914ec01de81943baa13f1
            Exponent1:
0x995c1b657b0d1d2d559e3e037ae372b2fece3ef95aa272b3da9a3cc19d4564962847aed2f8a8b1
c7ded3dad95b7b1a2c795cdfd64d297d786348cd7d476c30eb
```

```
        Exponent2:
0x5591f6dee43e6a83bee67ede0e17b420e81bc2a153061b8b10c2457c343f447ed6bbbf68ed5c5f
2705010c320ce0ac906ff5dc96141b6348013f010d7d1c0d4b
        Coefficient:
0x45f812ee6fa6198b808b2c24eea4c5def1241bd5432c289da8ae889202172e5063f1078202cd62
57f43e8bd0f71cc58d49b61a851486e189b5a6e155a096ac04
```

2) /etc/ipsec.conf

```
# /etc/ipsec.conf - FreeS/WAN IPsec configuration file

config setup
        # THIS SETTING MUST BE CORRECT or almost nothing will work;
        # %defaultroute is okay for most simple cases.
        interfaces=%defaultroute
        # Debug-logging controls:    "none" for (almost) none, "all" for lots.
        klipsdebug=none
        plutodebug=none
        # Use auto= parameters in conn descriptions to control startup actions.
        plutoload=%search
        plutostart=%search
        # Close down old connection when new one using same ID shows up.
        uniqueids=yes

conn tim
        # Left security gateway, subnet behind it, next hop toward right.
        left=10.101.85.113
        leftsubnet=192.168.0.0/24
        #leftnexthop=10.101.83.1
        # Right security gateway, subnet behind it, next hop toward left.
        right=10.101.85.115
        rightsubnet=192.168.1.0/24
        #rightnexthop=10.101.24.1
        # To authorize this connection, but not actually start it, at startup,
        # uncomment this.
        #auto=add
        keyingtries=0
        authby=rsasig
        auto=start

leftrsasigkey=0sAQN+MvfkzsJZLOayzpBm4dqoOUl3vhnuJesNpvXsXps+Mp51vIomAb2TtDg7
YWKG7LqKeJtFlzwFCo9TAFUnHygO65HmoPvqjMhfFZktzxnlqN0ezf7zbJECgL66HWIg3PlD
F5ppd9AhKnH7UIfwGmlaA9QOWikabSpXhGLwqlJJVQ==
```

//取本机上 keyfile 文件中 publickey= 的红色部分

```
rightrsasigkey=0sAQOEYhcHK1hSW5DQ52NDz43uJzgmkVrcDmzLPZo93RqrqtQ++HEeYadT
32SseydRLaqQv0907cwQbUkUokWaSGmwn3f0gckyb2V4TNJvHytkCsdrabVABSN+/UNrh0Cn
m4hQFP3j/6YoAxU/JTGJyRwgfXfiZq1naJdNPcJwiUrqMw==
```

注：

“authby=rsasig” tells FreeS/WAN that this connection is going to use raw RSA keys to authenticate. The keys in this file (ipsec.conf) are the public keys, which means they are safe for distribution (you don’t care if your enemies have your public key). You get your public key from your recently generated keyfile; look for the only line that starts “pubkey=.......” (where ..... is your key, it will be long!).

3) `/etc/ipsec.secrets`

//取本机上 keyfile 文件的蓝色部分

```
10.101.85.113 10.101.85.115: RSA {
        Modulus:
0x7e32f7e4cec2592ce6b2ce9066e1daa8394977be19ee25eb0da6f5ec5e9b3e329e75bc8a2601bd93b4383b616286ecba8a789b45973c050a
8f530055271f280eeb91e6a0fbea8cc85f15992dcf19e5a8dd1ecdfef36c910280beba1d6220dcf943179a6977d0212a71fb5087f01a695a03d4
0e5a291a6d2a578462f0aa524955
        PublicExponent: 0x03
        # everything after this point is secret
        PrivateExponent:
0x15087ea62275b9877bc877c2bbd04f1c098c3e9faefd0651d79bd3a76519dfb31a68f4c1b1004a4348b409e5906bd21f171419e0ee8a00d7
17e32ab8dbda86ace9dd535b96a3a48914c05c835b08c6b94236ff18e0172e1e101edaa12aa0921fbf3412e4f1fa4a7237fda400bbbfe965b3ed
54d3b5b1e8fc4054cd64a752774b
        Prime1:
0xe06f150d436ef617dfa50c8736b1ba9bea98d25b259435c67838fb6fb633ff1f7f816953946156b1d20b0461650bc6e05391dfdcf6ff7d10c3
0b78de02d64e25
        Prime2:
0x8ff2dd6e30a5bb7a02ee6192763382b5653c010e8d4d4687a7cc9ee6ac29711b485dbfb837910bcb4ffe7422248f2a1778b23186efef7a301
27c19b6bb8d2f71
        Exponent1:
0x959f635e2cf4a40fea6e085a24767c67f1bb36e76e62ce845025fcf52422aa14ffab9b8d0d9639cbe15cad96435d2f4037b6953df9ffa8b5d75
cfb3eac8edec3
        Exponent2:
0x5ff73e4975c3d25157499661a422572398d2ab5f08de2f051a8869ef1d70f612303e7fd0250b5d3235544d6c185f7164fb217659f54a51756
1a811247d08ca4b
        Coefficient:
0x6bdfd50d78ca75b87109c0f2b3015baa322c0f5542fe5a6cc1937b91d1983f49c56e75269c91c9e9a648c0f4c7df7e6b107ddb803de47d931
3a7dfb15c01b5fc
        }
```

4) 说明

- 生成密钥文件最好在每个计算机上自己做，因为涉及到随机数的问题，在另外机器上生成的文件可能在本机器上不能运行

  ```
  ipsec rsasigkey --verbose 1024 > keyfile
  ```

- left 和 Right 主机上的 ipsec.conf 文件是一致的，而 ipsec.secrets 则不相同；
- keyfile: 公、私钥对
- left/right rsasigkey:定义的是公钥
- /etc/ipsec.secrets:定义的是本地公钥对应的私钥
- 目前基于 Windows 的 client 端不支持这种 Raw RSA 方式的连接

三 基于X.509 证书的连接

1. 证书的生成和发布

1) 生成 CA 根证书

# /usr/lib/ssl/misc/CA –newca

输入文件名：直接回车即可

输入口令： 720801

Subject: CA Root

……

生成：

CA certificate: **/usr/lib/ssl/misc/demoCA/cacert.pem**

The private key of the CA : **/usr/lib/ssl/misc/demoCA/private/cakey.pem**

# cp /usr/lib/ssl/misc/demoCA/cacert.pem /etc/ipsec.d/cacerts

2) Creating the FreeSWAN Certificate

# /usr/lib/ssl/misc/CA –newreq

输入口令：700629

Subject: FreeS/WAN Administrator hjbin@infosec.pku.edu.cn

……

# /usr/lib/ssl/misc/CA.sh –sign ；用 CA Root 的公钥签署证书

输入 CA Root 私钥的口令 720801

# mv newreq.pem /etc/ipsec.d/private/freeswan-priv.pem

# mv newcert.pem /etc/ipsec.d/freeswan-cert.pem

生成：

私钥：freeswan-priv.pem

签署的证书： freeswan-cert.pem

To let FreeSWAN read the X.509 Certificate, it has to be in DER format. The key should be in **/etc/x509cert.der**

To be able to export it to DER format we use the following command:

# openssl x509 -in /etc/ipsec.d/freeswan-cert.pem -outform DER -out /etc/x509cert.der

Make sure the **/etc/ipsec.secrets** file looks like this in your favorite text editor.

# vi /etc/ipsec.secrets

: RSA freeswan-priv.pem "700629"

So far the FreeSWAN Certificate.

3) Creating the Roadwarrior Certificate

# /usr/lib/ssl/misc/CA –newreq

输入口令：700000

Subject: Hubei hsj hsj@263.net

……

＃ /usr/lib/ssl/misc/CA.sh –sign ；用 CA Root 的公钥签署证书

输入 CA Root 私钥的口令 720801

＃ mv newreq.pem /etc/ipsec.d/private/client-priv.pem

＃ mv newcert.pem /etc/ipsec.d/client-cert.pem

生成：

私钥：client-priv.pem

签署的证书： client-cert.pem

4) Certificate Revocation List

To create the CA's revocation list:

Make sure the **/etc/ipsec.d/crls** directory exists when executing the following command.

# openssl ca -gencrl -out /etc/ipsec.d/crls/crl.pem

This creates an empty revocation list with a validity that is listed in openssl.cnf

If you want to revoke a certificate you can do this as follows:

# openssl ca -revoke certificate.pem

Then the revocation list has to be regenerated using the following command:

# openssl ca -gencrl -crldays xx -out /etc/ipsec.d/crls/crl.pem

Where xx is the number of days.
If for some reason, you want to view the contents of the crl then it can be listed with the following command:

# openssl crl -in /etc/ipsec.d/crls/crl.pem -noout -text

5) 准备客户端证书的发布

In order to import the created certificates into **PGPNet** we need to convert them to a readable format that PGPNet understands and supports.

First we need to export the public key to **.p12** format. This format is also supported in Internet Explorer and Netscape. If for some reason you also want it in IE or netscape use this.：

＃ openssl pkcs12 –export -in /etc/ipsec.d/client-cert.pem -inkey /etc/ipsec.d/private/client-priv.pem -certfile /usr/lib/ssl/misc/demoCA/cacert.pem -out

/tmp/client.p12

Enter PEM pass phrase:<ROADWARRIOR_PASSWORD>
Enter Export Password:<EXPORT_PASSWORD>
Verifying password - Enter Export Password:

The **freeswan-cert.pem** created by openssl **can't**be imported into PGPNet straight away. This is because PGPkeys does not accept certificates in DER format. It has to be in base64 format to import them into PGPkeys.
The following command will convert it from DER format to base64 format.

```
# openssl x509 -in /etc/ipsec.d/freeswan-cert.pem -out /tmp/freeswan-cert.pem
```

6) 查看根证书的 subject

```
# openssl x509 -in demoCA/cacert.pem -noout -subject
```

并记载如下：

C=CN, ST=Beijing, L=Haidian, O=pku, OU=infosec lab, CN=CA Root, Email=CARoot@infosec.pku.edu.cn

2. Left 主机配置(Linux FreeS/WAN)

(**IP: 192.168.102.59, Linux， Root CA 及各种证书都在该机器上生成**)

- **/etc/ipsec.conf**

```
config setup
        interfaces="ipsec0=eth0"
        klipsdebug=none
        plutodebug=none
        plutoload=%search
        plutostart=%search
        uniqueids=yes
conn %default
        keyingtries=0
        authby=rsasig
conn sample3
        left=192.168.102.59
        leftcert=freeswan-cert.pem
        leftrsasigkey=%cert
        right=192.168.102.251
        rightcert=client-cert.pem
        rightrsasigkey=%cert
        auto=start
        keyingtries=0
        pfs=yes
        compress=yes
        type=transport
```

说明：

当 Right 主机为任意主机时，可以删除 rightcert 选项，并设置 right=%any

- **/etc/ipsec.secrets**

```
192.168.102.59 192.168.102.251 : RSA freeswan-priv.pem "700629"
```

3. Right 主机配置(Windows)

(**IP: 192.168.102.251, Windows 2000**)

- Client-cert.p12 导入到个人证书列表中，可看到包含一个私钥；
- Cacert.pem 导入到根证书列表中，可看到可信任的证书链；
- Ipsec.conf 配置如下：

```
config setup
        klipsdebug=none
        plutodebug=none
        plutoload=%search
        plutostart=%search
        uniqueids=yes
conn %default
        keyingtries=0
        authby=rsasig
         network=lan
conn sample3
        left=192.168.102.59
        right=192.168.102.251
        rightca="C=CN,ST=Beijing,L=Haidian,O=pku,OU=infoseclab,CN=CA Root, Email=CARoot@infosec.pku.edu.cn"   //此处来自于 Root CA 证书的 Subject
        auto=start
        keyingtries=0
        pfs=yes
       type=transport
```

小结：

- **Windows Client 端只要下载根证书和本地证书，然后制定 rightca 即可，不必制定本地证书的 ID 等，由系统自动识别，但也可以指定本地证书；**
- **在 Linux 服务器端只要指定本地证书 leftcert 即可，但也可以指定根证书；**

4. Right 主机配置（Linux FreeS/WAN）

- **复制 Left 主机上的下列配置文件到本地主机：**

```
# cp   client-cert.pem   /etc/ipsec.d
# cp   client-priv.pem   /etc/ipsec.d/private
# cp   freeswan-cert.pem   /etc/ipsec.d     //可选
# cp cacert.der   /etc/ipsec.d/cacerts/.
# cp crl.pem /etc/ipsec.d/crls
```

- 在本地机上：

```
#  openssl x509   -in client-cert.pem -outform der   -out /etc/x509cert.de
```

- **/etc/ipsec.conf**

```
config setup
        interfaces="ipsec0=eth0"
        klipsdebug=none
        plutodebug=none
        plutoload=%search
        plutostart=%search
        uniqueids=yes
conn %default
        keyingtries=0
        authby=rsasig
conn sample3
        left=192.168.102.59
        right=192.168.102.251
        rightcert=client-cert.pem
        auto=start
        keyingtries=0
        pfs=yes
        compress=yes
        type=transport
```

- **/etc/ipsec.secrets**

```
192.168.102.59 192.168.102.251 : RSA client-priv.pem "710000"
```

四 配置文件及说明

1. Ipsec.conf 文件格式说明

type name; 每个 section 以此开头

{*parameter*=*value*}; 包含多个参数/值对

also=Other section name; 另外 section 的配置将被加入到本 section 中

type %default; 所有与此 *type* 相同类型的 section 都取本 section 定义的 *parameter value*

type := {config|conn};

parameter:= {type|

2. CONN section 通用字段定义

1. **type**

   tunnel (the default): 表示 host-to-host, host-to-subnet, or subnet-to-subnet tunnel;

   transport, signifying host-to-host transport mode;

   passthrough (supported only for manual keying), signifying that no IPsec processing should be done at all;

2. **left** **(required)** **t**he IP address of the left participant's public-network interface.

   %defaultroute, and **interfaces=%defaultroute** is used in the **config setup** section, **left** will be filled in automatically with the local address of the default-route interface (as determined at IPsec startup time); this also overrides any value supplied for **leftnexthop**. (Either **left** or **right** may be **%defaultroute**, but not both.)

   %any signifies an address to be filled in (by automatic keying) during negotiation;

   %opportunistic signifies that both left and leftnexthop are to be filled in (by automatic keying) from DNS data for left's client.

3. **leftsubnet**
4. **leftnexthop**
5. **leftupdown**

   what ``updown" script to run to adjust routing and/or firewalling when the status of the connection changes (default **ipsec _updown**). May include positional parameters separated by white space (although this requires enclosing the whole string in quotes); including shell metacharacters is unwise. See *ipsec_pluto*(8) for details. Relevant only locally, other end need not agree on it.

6. **Leftfirewall = {yes|no}**

3. CONN section for AUTOMATIC KEYING 字段定义

1. **Keyexchange=IKE**
2. **auto** what operation, if any, should be done automatically at IPsec startup

   add (signifying an **ipsec auto --add**),

   route (signifying that plus an **ipsec auto --route**)

   start (signifying that plus an **ipsec auto --up**)

   manual (signifying an **ipsec manual**)

   ignore (default) (signifying no automatic startup operation).

   but in general, for an intended-to-be-permanent connection, both ends should use **auto=start** to ensure that any reboot causes immediate renegotiation).

3. **auth**

   whether authentication should be done as part of ESP encryption, or separately using the AH protocol

   esp (the default)

   ah.

4. **authby**

   how the two security gateways should authenticate each other; acceptable values are **secret** for shared secrets and **rsasig** for RSA digital signatures (the default). Digital signatures are superior in every way to shared secrets.

5. **leftid**

   how the left participant should be identified for authentication; defaults to **left**. Can be an IP address (in any *ipsec_ttoaddr*(3) syntax) or a fully-qualified domain name preceded by **@** (which is used as a literal string and not resolved).

6. **leftrsasigkey** the left participant's public key for RSA signature authentication.

   %none means the same as not specifying a value (useful to override a default).

   %dnsondemand (the default) means the key is to be fetched from DNS at the time it is needed.

   %dnsonload means the key is to be fetched from DNS at the time the connection description is read from *ipsec.conf*; currently this will be treated as **%none** if **right=%any** or **right=%opportunistic**.

   %dns is currently treated as %dnsonload but will change to %dnsondemand in the future. The identity used for the left participant must be a specific host, not %any or another magic value.

   Caution: if two connection descriptions specify different public keys for the same **leftid**, confusion and madness will ensue.

7. **leftrsasigkey2**

   if present, a second public key. Either key can authenticate the signature, allowing for key rollover.

8. **pfs**

   whether Perfect Forward Secrecy of keys is desired on the connection's keying channel (with PFS, penetration of the key-exchange protocol does not compromise keys negotiated earlier); acceptable values are **yes** (the default) and **no**.

9. **keylife**

   how long a particular instance of a connection (a set of encryption/authentication keys for user packets) should last, from successful negotiation to expiry; acceptable values are an integer optionally followed by **s** (a time in seconds) or a decimal number followed by **m**, **h**, or **d** (a time in minutes, hours, or days respectively) (default **8.0h**, maximum **24h**). Normally, the connection is renegotiated (via the keying channel) before it expires. The two ends need not exactly agree on **keylife**, although if they do not, there will be some clutter of superseded connections on the end which thinks the lifetime is longer.

10. **rekey**

    whether a connection should be renegotiated when it is about to expire; acceptable values are **yes** (the default) and **no**. The two ends need not agree, but while a value of **no** prevents Pluto from requesting renegotiation, it does not prevent responding to renegotiation requested from the other end, so **no** will be largely ineffective unless both ends agree on it.

11. **rekeymargin**

    how long before connection expiry or keying-channel expiry should attempts to negotiate a replacement begin; acceptable values as for **keylife** (default **9m**). Relevant only locally, other end need not agree on it.

12. **rekeyfuzz**

    maximum percentage by which **rekeymargin** should be randomly increased to randomize rekeying intervals (important for hosts with many connections); acceptable values are an integer, which may exceed 100, followed by a `%' (default set by *ipsec_pluto*(8), currently **100%**). The value of **rekeymargin**, after this random increase, must not exceed **keylife**. The value **0%** will suppress time randomization. Relevant only locally, other end need not agree on it.

13. **keyingtries**

    how many attempts (an integer or **%forever**) should be made to negotiate a connection, or a replacement for one, before giving up (default **%forever**). The value **%forever** means ``never give up'' (obsolete: this can be written **0**). Relevant only locally, other end need not agree on it.

14. **ikelifetime**

    how long the keying channel of a connection (buzzphrase: ``ISAKMP SA'') should last before being renegotiated; acceptable values as for **keylife** (default set by *ipsec_pluto*(8), currently **1h**, maximum **8h**). The two-ends-disagree case is similar to that of **keylife**.

15. **compress**

    whether IPComp compression of content is desired on the connection (link-level compression does not work on encrypted data, so to be effective, compression must be done *before* encryption); acceptable values are **yes** and **no** (the default). The two ends need not agree. A value of **no** is absolute: IPsec will neither propose nor accept compression. A value of **yes** causes IPsec to propose both compressed and uncompressed, and prefer compressed.

16. **disablearrivalcheck**

    whether KLIPS's normal tunnel-exit check (that a packet emerging from a tunnel has plausible addresses in its header) should be disabled; acceptable values are **yes** and **no** (the default). Tunnel-exit checks improve security and do not break any normal configuration. Relevant only locally, other end need not agree on it.

4. CONN section for MANUAL KEYING 字段定义

1. **spi**

   (this or **spibase** required for manual keying) the SPI number to be used for the connection (see *ipsec_manual*(8)); must be of the form **0x***hex*, where *hex* is one or more hexadecimal digits (note, it will generally be necessary to make *spi* at least **0x100** to be acceptable to KLIPS, and use of SPIs in the range **0x100-0xfff** is recommended)

2. **spibase**

   (this or **spi** required for manual keying) the base number for the SPIs to be used for the connection (see *ipsec_manual*(8)); must be of the form **0x***hex***0**, where *hex* is one or more hexadecimal digits (note, it will generally be necessary to make *spibase* at least **0x100** for the resulting SPIs to be acceptable to KLIPS, and use of numbers in the range **0x100-0xff0** is recommended)

3. **esp**

   ESP encryption/authentication algorithm to be used for the connection, e.g. **3des-md5-96** (must be suitable as a value of *ipsec_spi*(8)'s **--esp** option); default is not to use ESP

4. **espenckey**

   ESP encryption key (must be suitable as a value of *ipsec_spi*(8)'s **--enckey** option) (may be specified separately for each direction using **leftespenckey** (leftward SA) and **rightespenckey** parameters)

5. **espauthkey**

   ESP authentication key (must be suitable as a value of *ipsec_spi*(8)'s **--authkey** option) (may be specified separately for each direction using **leftespauthkey** (leftward SA) and **rightespauthkey** parameters)

6. **espreplay_window**

   ESP replay-window setting, an integer from **0** (the *ipsec_manual* default, which turns off replay protection) to **64**; relevant only if ESP authentication is being used

7. **leftespspi**

   SPI to be used for the leftward ESP SA, overriding automatic assignment using **spi** or **spibase**; typically a hexadecimal number beginning with **0x**

8. **ah**

   AH authentication algorithm to be used for the connection, e.g. **hmac-md5-96** (must be suitable as a value of *ipsec_spi*(8)'s **--ah** option); default is not to use AH

9. **ahkey**

   (required if **ah** is present) AH authentication key (must be suitable as a value of *ipsec_spi*(8)'s **--authkey** option) (may be specified separately for each direction using **leftahkey** (leftward SA) and **rightahkey** parameters)

10. **ahreplay_window**

    AH replay-window setting, an integer from **0** (the *ipsec_manual* default, which turns off replay protection) to **64**

11. **leftahspi**

    SPI to be used for the leftward AH SA, overriding automatic assignment using **spi** or **spibase**; typically a hexadecimal number beginning with **0x**

5. CONFIG Section

At present, the only **config** section known to the IPsec software is the one named **setup**, which contains information used when the software is being started (see *ipsec_setup*(8)). Here's an example:

```
config setup
 interfaces="ipsec0=eth1 ipsec1=ppp0"
 klipsdebug=none
 plutodebug=all
 manualstart=
```

Parameters are optional unless marked ``(required)''. The currently-accepted *parameter* names in a config setup section are:

1. **interfaces**

   virtual and physical interfaces for IPsec to use: a single *virtual*=*physical* pair, a (quoted!) list of pairs separated by white space, **%none**, or **%defaultroute** (the default) which means to find the interface *d* that the default route points to, and then act as if the value was ``**ipsec0=***d*''. (Also, in the **%defaultroute** case, information about the default route and its interface is noted for use by *ipsec_manual*(8) and *ipsec_auto*(8).)

2. **forwardcontrol**

   whether *setup* should turn IP forwarding on (if it's not already on) as IPsec is started, and turn it off again (if it was off) as IPsec is stopped; acceptable values are **yes** and (the default) **no**. For this to have full effect, forwarding must be disabled before the hardware interfaces are brought up (e.g., **net.ipv4.ip_forward = 0** in Red Hat 6.x */etc/sysctl.conf*), because IPsec doesn't get control early enough to do that.

3. **rp_filter**

   whether and how *setup* should turn adjust the reverse path filtering mechanism for the phyiscal devices to be used. Values are **%unchanged** (to leave it alone) or **0**, **1**, **2** (values to set it to). */proc/sys/net/ipv4/conf/PHYS/rp_filter* is badly documented; it must be **0** in many cases for ipsec to function. The default value for the parameter is **0**.

4. **syslog**

   the *syslog*(2) ``facility'' name and priority to use for startup/shutdown log messages, default **daemon.error**.

5. **klipsdebug**

   how much KLIPS debugging output should be logged. An empty value, or the magic value **none**, means no debugging output (the default). The magic value **all** means full output. Otherwise only the specified types of output (a quoted list, names separated by white space) are enabled; for details on available debugging types, see *ipsec_klipsdebug*(8).

6. **plutodebug**

   how much Pluto debugging output should be logged. An empty value, or the magic value **none**, means no debugging output (the default). The magic value **all** means full output. Otherwise only the specified types of output (a quoted list, names without the **--debug-** prefix, separated by white space) are enabled; for details on available debugging types, see *ipsec_pluto*(8).

7. **plutoopts**

   additional options to pass to pluto upon startup. See *ipsec_pluto*(8).

8. **plutostderrlog**

   do not use syslog, but rather log to stderr, and direct stderr to the argument file.

9. **dumpdir**

   in what directory should things started by *setup* (notably the Pluto daemon) be allowed to dump core? The empty value (the default) means they are not allowed to.

10. **manualstart**

    which manually-keyed connections to set up at startup (empty, a name, or a quoted list of names separated by white space); see *ipsec_manual*(8). Default is none.

11. **pluto**

    whether to start Pluto or not; Values are **yes** (the default) or **no** (useful only in special circumstances).

12. **plutowait**

    should Pluto wait for each negotiation attempt that is part of startup to finish before proceeding with the next? Values are **yes** or **no** (the default).

13. **prepluto**

    shell command to run before starting Pluto (e.g., to decrypt an encrypted copy of the *ipsec.secrets* file). It's run in a very simple way; complexities like I/O redirection are best hidden within a script. Any output is redirected for logging, so running interactive commands is difficult unless they use */dev/tty* or equivalent for their interaction. Default is none.

14. **postpluto**

    shell command to run after starting Pluto (e.g., to remove a decrypted copy of the *ipsec.secrets* file). It's run in a very simple way; complexities like I/O redirection are best hidden within a script. Any output is redirected for logging, so running interactive commands is difficult unless they use */dev/tty* or equivalent for their interaction. Default is none.

15. **fragicmp**

    whether a tunnel's need to fragment a packet should be reported back with an ICMP message, in an attempt to make the sender lower his PMTU estimate; acceptable values are **yes** (the default) and **no**.

16. **packetdefault**

    what should be done with a packet which reaches KLIPS (via a route into a virtual interface) but does not match any eroute; acceptable values are **pass** (*insecure unless you really know what you're doing!!!*), **drop** (the default), and **reject** (currently same as **drop**, but eventually it will send an ICMP notification back to the sender).

17. **hidetos**

    whether a tunnel packet's TOS field should be set to **0** rather than copied from the user packet inside; acceptable values are **yes** (the default) and **no**.

18. **uniqueids**

    whether a particular participant ID should be kept unique, with any new (automatically keyed) connection using an ID from a different IP address deemed to replace all old ones using that ID; acceptable values are **yes** (the default) and **no**. Participant IDs normally *are* unique, so a new (automatically-keyed) connection using the same ID is almost invariably intended to replace an old one.

19. **overridemtu**

    value that the MTU of the ipsec*n* interface(s) should be set to, overriding IPsec's (large) default. This parameter is needed only in special situations.

6. IPSEC.SECRETS

```
# sample /etc/ipsec.secrets file for 10.1.0.1
10.1.0.1 10.2.0.1: PSK "secret shared by two hosts"

# an entry may be split across lines,
# but indentation matters
www.xs4all.nl @www.kremvax.ru
    10.6.0.1 10.7.0.1 1.8.0.1: PSK "secret shared by 5"

# an RSA private key.
# note that the lines are too wide for a
# man page, so ... has been substituted for
# the truncated part
@my.com: rsa {
         Modulus: 0syXpo/6waam+ZhSs8Lt6jnBzu3C4grtt...
          PublicExponent: 0sAw==
          PrivateExponent: 0shlGbVR1m8Z+7rhzSyenCaBN...
          Prime1: 0s8njV7WTxzVzRz7AP+0OraDxmEAt1BL5l...
          Prime2: 0s1LgR7/oUMo9BvfU8yRFNos1s211KX5K0...
          Exponent1: 0soaXj85ihM5M2inVf/NfHmtLutVz4r...
          Exponent2: 0sjdAL9VFizF+BKU4ohguJFzOd55OG6...
          Coefficient: 0sK1LWwgnNrNFGZsS/2GuMBg9nYVZ...
    }
```