OPPORTUNISTIC ENCRYPTION USING IPSEC

Linux Security Summit, Toronto August 2016

Presented by Paul Wouters,
RHEL Security

THE LIBRESWAN PROJECT

An Internet Key Exchange ("IKE") daemon for IPsec

- Enterprise IPsec based VPN solution
- Make encryption the default mode of communication

- Certifications (FIPS, Common Criteria, USGv6, etc.)
- Contributing to IETF Standards for IKE and IPsec

IPsec PRIMER

IKE + IPsec = VPN

IKE (USERLAND)

ISAKMP, IKE SA, PHASE 1
UDP PORT 500 AND 4500

- Command Channel
- Peer authentication
- Connection parameter negotiation
- IPsec symmetric key generation
- Communicates to kernel

- IKE itself is encrypted
- IKE does not encrypt the IP traffic

IPsec (KERNEL)

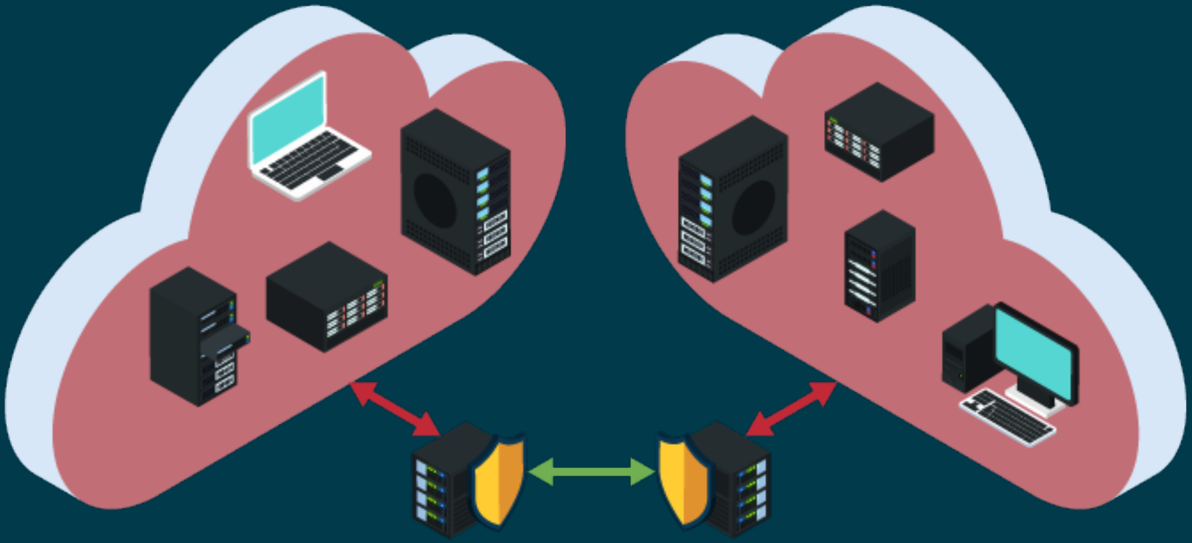
IPsec SA, CHILD SA, PHASE 2
PROTOCOL 50 AND 51

- Data Channel
  - Encapsulated Security Payload (ESP) IP packet encryption
  - Authenticated Header (AH)
  - ESPinUDP (for NAT)

- Tunnel Mode (IP in IP)
- Transport Mode

TYPICAL SITE TO SITE VPN

Individual networks are unencryped, only the interconnect is encrypted



TYPICAL REMOTE ACCESS VPN

End device to site network access point encrypted – LAN still unencrypted

HISTORY: THE FreeS/WAN PROJECT (1996-2003)

“My project for 1996 was to secure 5% of the Internet traffic against passive wiretapping”

– John Gilmore

- Opportunistic Encryption
- Enable encryption between any two nodes without pre-configuration

HISTORY:
THE FreeS/WAN PROJECT (1996-2003)

- **S/WAN stands for "Secure Wide Area Network" (trademarked by RSA Inc.)**
  - The term "Virtual Private Network" (VPN) became popular instead
- **Predates OpenSSL**
  - Used some SSLeay code (with special permission of Eric Andrew Young)
- **Predates the Internet key Exchange ("IKEv1" – RFC 2409) published in 1998**
- **Predates CryptoAPI (kerneli.org started around 2002)**
- **Predates United States export laws for cryptography**
  - 1995 – 1999: Bernstien v. United States on "crypto is free speech"
  - 1996: Allow export of 56-bit crypto (RC4) with key recovery backdoor
  - 1999: Allow export of 56-bit crypto (DES, RC4) without backdoor, and 1024-bit RSA
- **Predates DNSSEC and most of the CA industry**

HISTORY:
THE FreeS/WAN PROJECT (1996-2003)

Where FreeS/WAN succeeded

- Supported Linux 2.0 and up
- Became the gold standard of IKE and IPsec
- Strong deployment in the Enterprise

HISTORY:
THE FreeS/WAN PROJECT (1996-2003)

Where FreeS/WAN failed

- IKEv1 protocol specification took 3-years
- Developers could not be United States citizens (Oh Canada...)
- Packet triggered events relied on IP address
  - And no real access to the reverse DNS in-addr.arpa
- The Internet became reliant on NAT
  - And no universal deployment of IPv6 to obsolete it
- Required mutual authentication is problematic (SSL got it right)
- Unauthenticated encryption rejected as unsafe, confusing for enduser
- Secure DNS needed for key distribution took 15-years
  - Root zone finally signed in 2010
  - FreeS/WAN kicked out DNSSEC KEY/SIG records, back to TXT
- 2001: Echelon spying network exposed – no one cares

HISTORY:
THE OPENSWAN PROJECT (2003-2011)

Ex-employees and volunteers forked FreeS/WAN

- John Gilmore gives up on OE
- Americans can submit code

- Enterprise deployments just work
- Support native IPsec (XFRM/NETKEY)
- Use optional NSS crypto library
- Hardware acceleration support (OCF and native)
- Initial rough IKEv2 implementation

- Opportunistic Encryption mothballed
- 2012: A lawsuit requires that the project renames itself

HISTORY:
THE LIBRESWAN PROJECT (2011-ONGOING)

The Great Overhaul

- Only use NSS crypto library
- IKEv2
- Crypto suites update
- Modern network timers
- Event loops
- Cleanup codebase to support FIPS, CAVP, Common Criteria
- Cloud support

- Revisit Opportunistic Encryption

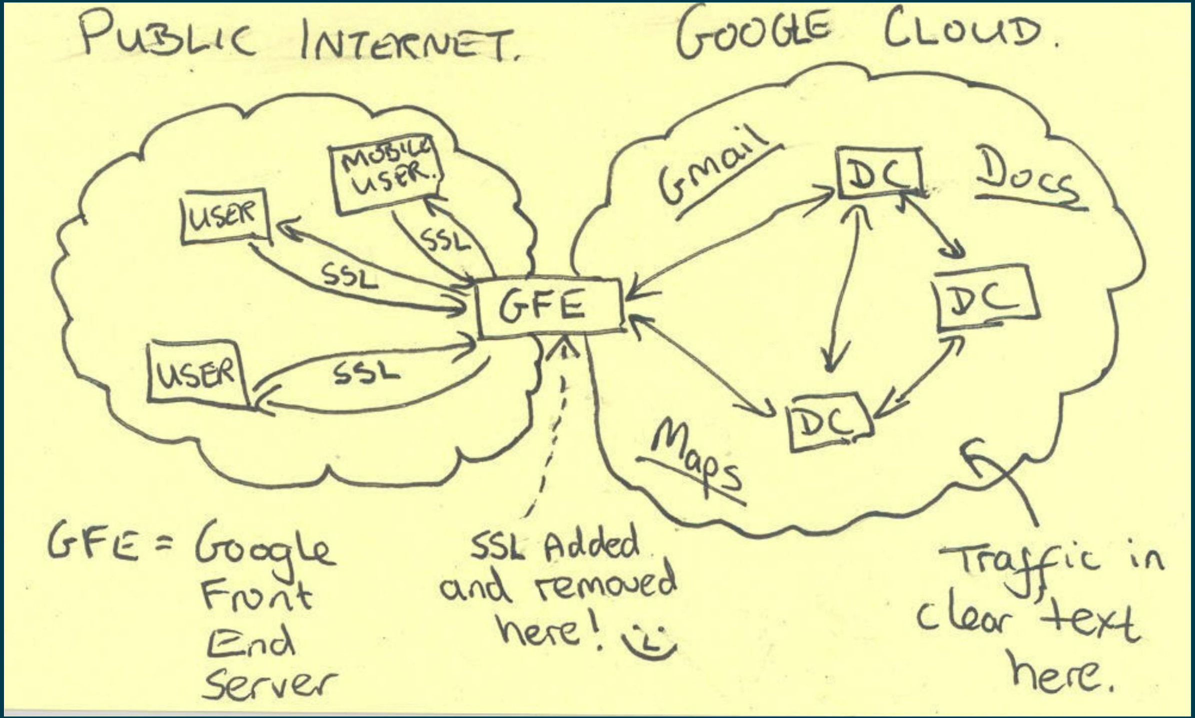
HISTORY: REVISITING OPPORTUNISTIC ENCRYPTION

- IKEv2 allows asymmetric AUTH like SSL/TLS
- IKEv2 allows assigning IP addresses natively
- Linux conntrack vastly improved
- Linux XFRM/NETKEY vastly improved (TCP packet caching)
- DNSSEC expected to go on the end node
- Unbound DNS server with DNSSEC-trigger
- Allows DNS based triggers for Opportunistic Encryption

- DNSSEC triggers to replace the reverse DNS in-addr.arpa for ID AUTH
- Linux conntrack and IKEv2 addresspool to resolve NAT problem

- If people only realized they want ambiguous encryption.........

HISTORY:
EDWARD SNOWDEN (2013)



HISTORY: THE CRYPTO RUSH

- Encryption is more important now
- Enterprises must encrypt:
  - Cloud instances
  - Data centres
  - MPLS, fibre
  - Transit cables
- Puppet / ansible does not scale for mesh encryption
- IPsec mesh encryption needed configuration modification on all nodes!

- Opportunistic encryption is cloud encryption
- Opportunistic encryption is internet encryption

HISTORY:
IETF RESPONSE

Internet Engineering Task Force steps up encryption

- RFC 7258 “Pervasive Monitoring Is an Attack” (May 2014)

  “Pervasive monitoring is a technical attack that should be mitigated in the design of IETF protocols, where possible.”

- RFC 7435 “Opportunistic Security” (Dec 2014)

  “Protocol designs based on Opportunistic Security use encryption even when authentication is not available, and use authentication when possible, thereby removing barriers to the widespread use of encryption on the Internet.”

OPPORTUNISTIC IPsec AT IETF

- RFC 7619 “NULL Authentication for IKEv2” (Aug 2015)
  - IKEv2 (2008) already allowed asymmetrical authentication
  - Allow Anonymous client to Authenticated Server
  - Allow Anonymous to Anonymous

- draft-antony-ipsecme-oppo-nat (Mar 2015)
  - NAT-Traversal support for Opportunistic IPsec

LINUX IPsec IMPLEMENTATION

XFRM/NETKEY interaction with userland

1. IPsec in the kernel has policies (SPD) and states (SAD)
   - Packets matching policies without a linked state cause ACQUIREs
     - If TCP, store packet. Else drop packet
   - Packets matching policies with a linked state causes encryption/decryption
2. Userland (libreswan) opens netlink socket to the kernel
   - Request to receive ACQUIREs and inserts “trap” policies
3. Userland (libreswan) processes ACQUIREs
   - Perform IKE negotiation with remote peer
   - Send IPsec policy and encryption/authentication keys to the kernel
4. Kernel processes netlink messages
   - Install received crypto keys in state, link crypto state to policy
   - If TCP triggered, send out cached packet

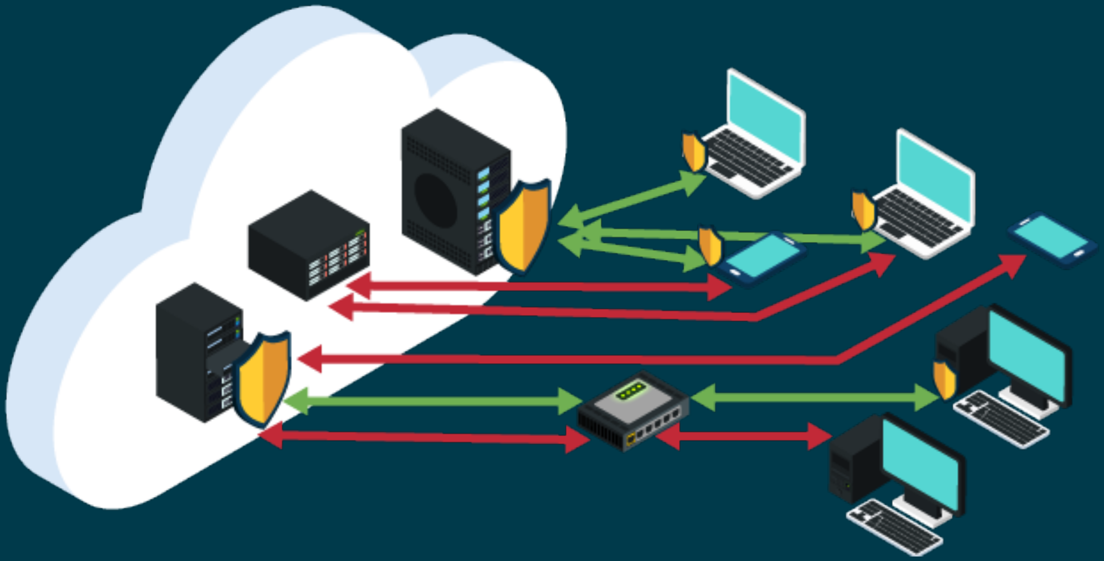
LINUX XFRM / NETKEY KERNEL STATE

```
src 10.3.230.191/32 dst 10.0.0.0/8
        dir out priority 666 ptype main
        tmpl src 76.10.157.68 dst 209.132.183.55
                proto esp reqid 16413 mode tunnel
src 0.0.0.0/0 dst 10.3.230.191/32
        dir fwd priority 666 ptype main
        tmpl src 209.132.183.55 dst 76.10.157.68
                proto esp reqid 16413 mode tunnel
src 0.0.0.0/0 dst 10.3.230.191/32
        dir in priority 666 ptype main
        tmpl src 209.132.183.55 dst 76.10.157.68
                proto esp reqid 16413 mode tunnel

src 209.132.183.55 dst 76.10.157.68
        proto esp spi 0x605ad2be reqid 16413 mode tunnel
        auth-trunc hmac(sha1) 0x4b7e46cdee9c27588a1a75f6846073cea 96
        enc cbc(aes) 0x11ddc9080945111087e81f9ebda5aacb7612c78af1895
src 76.10.157.68 dst 209.132.183.55
        proto esp spi 0x8ca00de3 reqid 16413 mode tunnel
        auth-trunc hmac(sha1) 0x1119585d334a88e023134a100eca6b09f 96
        enc cbc(aes) 0x310b852b9cbaf2cace7979c1aeb5df4b32eb418c5c300
```

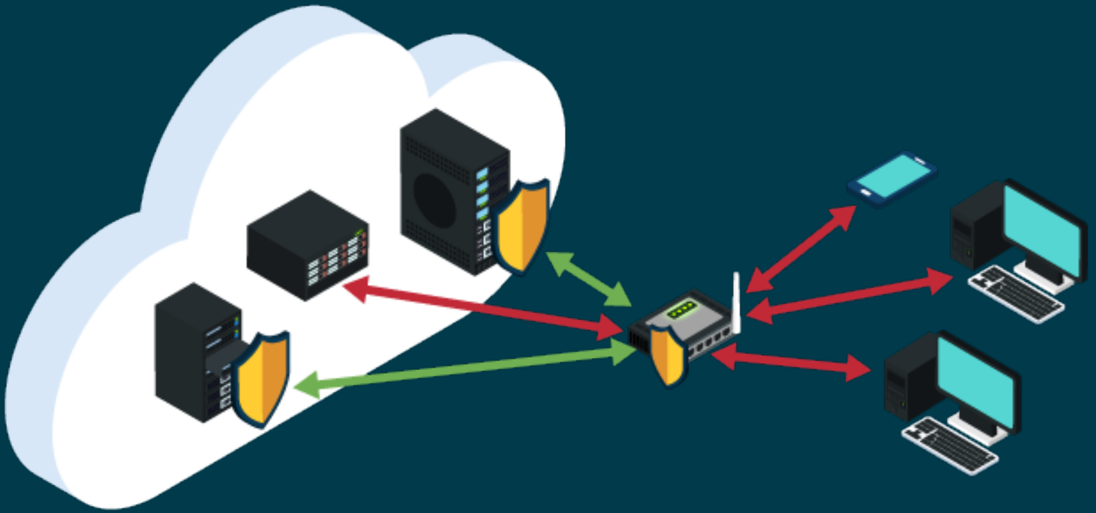
OPPORTUNISTIC IPSEC DEPLOYMENT

End-to-end encryption using IPsec



OPPORTUNISTIC IPSEC GATEWAY

Use a Linux gateway to protect devices not able to run opportunistic



DRAFT-ANTONY-IPSECME-OPPO-NAT

Eliminating the IP address conflicts caused by NAT

```
193.110.15.131  Remote Opportunistic IPsec server
192.168.2.45      Opportunistic Client pre-NAT IP address
100.64.0.2      IP address from IPsec server address pool
# ip xfrm pol
src 100.64.0.2/32 dst 193.110.157.131/32
           dir out priority 2080 ptype main
           tmpl src 192.1.2.45 dst 193.110.157.131
                      proto esp reqid 16389 mode tunnel
src 193.110.157.131/32 dst 100.64.0.2/32
           dir fwd priority 2080 ptype main
           tmpl src 193.110.157.131 dst 192.1.2.45
                      proto esp reqid 16389 mode tunnel
src 193.110.157.131/32 dst 100.64.0.2/32
           dir in priority 2080 ptype main
           tmpl src 193.110.157.131 dst 192.1.2.45
                      proto esp reqid 16389 mode tunnel
src 192.168.2.45/32 dst 193.110.157.131/32
           dir out priority 2080 ptype main
           tmpl src 192.1.2.45 dst 193.110.157.131
                      proto esp reqid 16389 mode tunnel
```

DRAFT-ANTONY-IPSECME-OPPO-NAT

Eliminating the IP address conflicts caused by NAT

```
193.110.15.131   Remote Opportunistic IPsec server
192.168.2.45     Opportunistic Client pre-NAT IP address
100.64.0.1       Client IP address assigned y Opportunistic Ipsec server

# iptables -t nat -L -n

Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination
DNAT       all  --  193.110.157.131      100.64.0.1 \
                      policy match dir in pol ipsec to:192.168.2.45

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
SNAT       all  --  0.0.0.0/0            193.110.157.131 \
                      policy match dir out pol ipsec to:100.64.0.1


Basically: NAT within the IPsec subsystem
```

LIBRESWAN – GROUP POLICIES

Group files in /etc/ipsec.d/policies/* list network CIDRs to match

```
/etc/ipsec.d/policies/block               Drop all packets
/etc/ipsec.d/policies/clear               Only allow cleartext
/etc/ipsec.d/policies/clear-or-private   Default clear, allow crypto
/etc/ipsec.d/policies/private             Mandate crypto, hard fail
/etc/ipsec.d/policies/private-or-clear   Attempt crypto, allow clear


# cat /etc/ipsec.d/policies/private-or-clear
193.110.157.0/24
193.111.228.0/24
# cat /etc/ipsec.d/policies/private
10.0.0.0/8
192.168.0.0/16
```

ENTERPRISE CLOUD MESH ENCRYPTION

Configuration for mandated mutual certificate based authentication

```
For example add 10.0.0.0/8 to /etc/ipsec.d/policies/private

# install localcertificate: ipsec import node1.example.com.p12
# /etc/ipsec.d/YourCloud.conf

conn private
        left=%defaultroute
        leftid=%fromcert
        # our certificate
        leftcert=node1.example.com
        right=%opportunisticgroup
        rightid=%fromcert
        # their certificate transmitted via IKE
        rightca=%same
        ikev2=insist
        authby=rsasig
        failureshunt=drop
        negotiationshunt=hold
        auto=ondemand
```

OPTIONAL OPPORTUNISTIC IPSEC

Configuration for optional anonymous IPsec

```
For example add 0.0.0.0/0 to /etc/ipsec.d/policies/private-or-clear

conn private-or-clear
        left=%defaultroute
        leftid=%null
        rightid=%null
        right=%opportunisticgroup
        authby=null
        ikev2=insist
        failureshunt=passthrough
        negotiationshunt=passthrough
        # to not leak during IKE negotiation, use
        # negotiationshunt=hold
        auto=ondemand
        # clear-or-private uses auto=add
```

LETSENCRYPT CERTIFICATES

Preparing libreswan to use LetsEncrypt certificates

```
mkdir letsencrypt ; cd letsencrypt
wget https://letsencrypt.org/certs/lets-encrypt-x4-cross-signed.pem
wget https://letsencrypt.org/certs/lets-encrypt-x3-cross-signed.pem
wget https://letsencrypt.org/certs/isrgrootx1.pem
# https://www.identrust.com/certificates/trustid/root-download-x3.html
wget https://nohats.ca/LE/identrust-x3.pem
yum install libreswan
ipsec initnss
certutil -A -i lets-encrypt-x3-cross-signed.pem -n lets-encrypt-x3 \ -
   -t CT,, -d sql:/etc/ipsec.d
certutil -A -i lets-encrypt-x4-cross-signed.pem -n lets-encrypt-x4 \
   -t CT,, -d sql:/etc/ipsec.d
certutil -A -i isrgrootx1.pem -n isrgrootx1 -t CT,, -d \
   sql:/etc/ipsec.d
certutil -A -i identrust-x3.pem -n identrust-x3 -t CT,, -d \
   sql:/etc/ipsec.d
```

LETSENCRYPT CLIENT FOR IPSEC

Anonymous client to authenticated server

```
cd /etc/ipsec.d
wget https://nohats.ca/LE/oe-letsencrypt-client.conf
echo "193.110.157.131/32" >> /etc/ipsec.d/policies/private-or-clear
(if adventurous, echo ”0.0.0.0/0" to private-or-clear)

ping letsencrypt.libreswan.org
PING letsencrypt.libreswan.org (193.110.157.131) 56(84) bytes of data.
64 bytes from letsencrypt.libreswan.org (193.110.157.131): icmp_seq=2
ttl=64 time=96.2 ms
64 bytes from letsencrypt.libreswan.org (193.110.157.131): icmp_seq=3
ttl=64 time=96
.7 ms

ipsec whack --trafficstatus
006 #2: "private-or-clear#193.110.157.131/32"[1] 100.64.0.2/32=== ...
193.110.157.131, type=ESP, add_time=1471926595, inBytes=252,
outBytes=252, id='CN=letsencrypt.libreswan.org', lease=100.64.0.2/32
```

LETSENCRYPT SERVER FOR IPSEC

/etc/ipsec.d/oe-letsencrypt-client.conf

```
conn private-or-clear
        left=%defaultroute
        leftid=%null
        leftauth=null
        leftmodecfgclient=yes
        leftcat=yes
        #
        rightid=%fromcert
        rightrsasigkey=%cert
        rightauth=rsasig
        right=%opportunisticgroup
        rightmodecfgclient=yes
        #
        narrowing=yes
        negotiationshunt=hold
        failureshunt=passthrough
        ikev2=insist
        auto=ondemand
```

LETSENCRYPT SERVER FOR IPSEC

Authenticated server for anonymous clients

```
# Install LetsEncrypt CA certs as documented on earlier slide

yum install letsencrypt
letsencrypt certonly -d yourserver.example.com

cd /etc/letsencrypt/live/yourserver.example.com
openssl pkcs12 -export -in cert.pem -inkey privkey.pem \
    -out yourserver.example.com.p12 –name yourserver.example.com \
    -CAfile chain.pem -certfile chain.pem -caname lets-encrypt-x3
ipsec import letsencrypt.libreswan.org.p12
cd /etc/ipsec.d
wget https://nohats.ca/LE/oe-letsencrypt-server.conf
echo "0.0.0.0/0" >> /etc/ipsec.d/policies/clear-or-private
ipsec restart
(for opportunistic server plus client, add 0.0.0.0 to private-or-clear)
```

LETSENCRYPT SERVER FOR IPSEC

/etc/ipsec.d/oe-letsencrypt-server.conf

```
conn clear-or-private
        leftid=%fromcert
        leftrsasigkey=%cert
        # your LetsEncrypt certificate
        leftcert=yourserver.example.com
        leftauth=rsasig
        left=%defaultroute
        leftaddresspool=100.64.0.1-100.64.255.254
        leftmodecfgclient=yes
        rightid=%null
        rightauth=null
        right=%opportunisticgroup
        negotiationshunt=passthrough
        failureshunt=passthrough
        ikev2=insist
        sendca=issuer
        auto=add
```

FEATURES PLANNED VERY SOON [TM]

- Unbound DNS server python module
  - Provide DNS based triggers
  - Give libreswan IKE daemon : DNS request, answer, and IP address
  - Lookup IPSECKEY before attempting Opportunistic IPsec
- Integration with Linux Virtual Tunnel Interface (VTI) using ipsec devices
- Hardening packet triggers against rerouting attack (coffee shop attacks)
- GSSAPI / Kerberos based trigger for Opportunistic IPsec
- Native kernel support for IPsec-NAT
- New libreswan library for
  - DBUS API for add, remove, status
  - Fake getsockopt() similar to tcpcrypt ?
  - Fake setsockopt() to require authenticated encryption
  - Get new socket options into kernel :-)

PRE-RELEASE SOFTWARE AVAILABLE

Source code and configuration files at https://nohats.ca/LE/

- RPMS and DEBs available
- readme.txt documents the necessary commands

- Feedback: swan-dev@lists.libreswan.org

- Opportunistic IPsec developers:
  - Antony Antony
  - Hugh Redelmeier
  - Paul Wouters