BlackHat®

USA 2003

IPsec: Opportunistic Encryption using DNSSEC

By Paul Wouters
<paul@xtdnet.nl>

Overview presentation
PART ONE: DNSSEC

- Theory of DNSSEC
- Using bind with DNSSEC
- Securing ".nl" with SECREG
- Deploying DNSSEC on large scale

Overview presentation
PART TWO: IPsec

- IPsec basics
- What is Opportunistic Encryption
- Protecting the net with OE
- Installing and Configuring OE
- Securing the wireless: WaveSEC
- Testing OE

Zone view of DNS

- From To Question Answer
- Clientip Resolver A www.freeswan.nl 193.110.157.9
- <file> - NS . NS A.ROOT-SERVERS.NET.
- A 198.14.0.4 (GLUE)
- Resolver 198.14.0.4 NS nl. NS NS.DOMAIN-REGISTRY.NL
- A 193.176.144.2 (GLUE)
- Resolver 193.176.144.2 NS freeswan.nl. NS ns.xtdnet.nl.
- A 193.110.157.2 (GLUE)
- Resolver 193.110.157.2 A www.freeswan.nl. A 193.110.157.9 (AUTHORATIVE)

Organisational view of DNS

- The Root Registry (InterNic/VeriSign/DoD/IANA/IAB/ICANN)
- ROOT-SERVERS.NET (IETF)
- CC:TLD-SERVERS.NET (IETF/RIPE/RIPE NCC)
- The NL Registry DOMAIN-REGISTRY.NL (SIDN)
- (AMS-IX,KPN,NIKHEF,SURFNET,RIPE NCC,NIC.SE,NIC.FR,UUNET)
- Registrar of 157.110.193.IN-ADDR.ARPA. (XTDNET,RIPE,BBC)
- ISP of 157.110.193.IN-ADDR.ARPA. (XTDNET,BBC,EASYNET,INTERNATION)
- Registrant (FreeS/WAN)

- 23 Organisations (+everyone with access to routing or BGP tables)

IPsec: Opportunistic Encryption using DNSSEC

Vulnerabilities of DNS:
1) Integrity of data
2) Authenticity of data

- Client <-> DHCP Server
- Client <-> Resolver
- Resolver itself (rootfile)
- Resolver's communication to the net
- Various glue records and their update mechanism
- Various nameserver <-> nameserver communication
- Various network <-> network communication

Protect DNS with digital signatures

- Secure client <-> resolver communication
  - Secure LAN/DHCP?
  - DNSSEC aware Resolver on Client(!)
- Secure communication nameservers
  - Zone transfers (AXFR)
  - dynamic updates
- Secure data storage integrity
  - Zonefiles
  - Caches

Secure nameserver communication

- TSIG: Preshared Secret Key to protect AXFR
  - Strictly speaking not necessary with secure zones
  - Secure the IP layer
    - IPsec tunnel between master and slaves
    - Transfer zones from master to slave using SSH/SCP/SFTP
- SIG0: Public key cryptography
  - See above
  - Useful for dynamic updates

DNSSEC: The new record types

- The KEY record: The public key used
  unfortunately for DNSSEC only since RFC3445
- The SIG record: The signatures created by that key
- The NXT record: For denial of existance
- The DS record: For building the chain of trust

- New flag: the Authenticated Data (AD) flag
  Not protected by a signature!

DNSSEC: The KEY record

```
RIlabel  TTL  IN  KEY  <flags> <protocol> <algorithm> <key material>
```

```
reeswan.nl.  3600  IN  KEY  256  3  5 (
                AQPRv8TN8ayfxrtRo1dveOMVSSpT4PGEZvfGjaERldQZ
                izYKgVBj/l84DjVktGUbkJ3pBiLBAzZ+5nbGkWn+Lz5Z

HMlQnjWde/mKKDlZnwQ13vU+HPt3cszNy9CdBmn6l8=
                ) ; key id = 56954
```

ags: authentication, confidentiality
rotocol: DNSSEC = 3, IPsec = 4 [only protocol 3 is allowed since RFC3445]

DNSSEC: The SIG record

```
RRlabel  TTL  IN  SIG  <type covered> <algorithm> <nr. of labels>
                                       <original ttl> <sig expiration>
<sig inception>
                                       <key tag> <signer name> <key
material>
```

```
reeswan.nl.        3600 IN NS ns.xtdnet.nl.
reeswan.nl.        3600 IN NS ns1.xtdnet.nl.
reeswan.nl.        3600 IN SIG NS 5 2 3600 20030506165654 (
                    20030406165654 56954 freeswan.nl.
                  bTKJvyrwmP+nsFoE8oelC4gFqoyJxkawNIExMVupI+ie
```

DNSSEC: The NXT record

RRlabel TTL IN NXT <alphanumeric next RRlabel>
<list of existing RRsets>

reeswan.nl. 3600 IN NXT activeOE.freeswan.nl. NS SOA MX SIG KEY
NXT

Denial of existance: We now know there is no RRset abc.freeswan.nl.

DNS: Example zone

```
$TTL            3600
freeswan.nl.            IN      SOA     ns.xtdnet.nl. hostmaster.freeswan.nl. (
                             2003040619 ; Serial
                                28800   ; Refresh
                                7200    ; Retry
                                604800  ; Expire
                                3600)   ; Minimum

freeswan.nl.            IN      NS      ns.xtdnet.nl.
freeswan.nl.            IN      NS      ns1.xtdnet.nl.
www.freeswan.nl.        IN      A       193.110.157.9
freeswan.nl.            IN      MX      50      www.freeswan.nl.
                                                        1,1           All
```

IPsec: Opportunistic Encryption using DNSSEC

```
freeswan.nl.            3600    IN SOA  ns.xtdnet.nl. hostmaster.freeswan.nl. (
                                        2003040620 ; serial
                                        28800      ; refresh (8 hours)
                                        7200       ; retry (2 hours)
                                        604800     ; expire (1 week)
                                        3600       ; minimum (1 hour)
                                        )
                        3600    SIG     SOA 5 2 3600 20030510221048 (
                                        20030410221048 56954 freeswan.nl.
                                        [...] HVEb/ksXy/vNrY3JKVoo )
                        3600    NS      ns.xtdnet.nl.
                        3600    NS      ns1.xtdnet.nl.
                        3600    SIG     NS 5 2 3600 20030510221048 (
                                        20030410221048 56954 freeswan.nl.
                                        [...] ePuHDsOldZwDXllcha+j )
                        3600    MX      50 www.freeswan.nl.
                        3600    SIG     MX 5 2 3600 20030510221048 (
                                        20030410221048 56954 freeswan.nl.
                                        [...] a1MzpYlwxO9AWoyNIf/g )
                        3600    KEY     256 3 5 (
                                        [...] FSc6wWff2MSuG9qZ6o1H
                                        ) ; key id = 49601
                        3600    KEY     256 3 5 (
                                        [...] 13vU+HPt3cszNy9CdBmn618=
                                        ) ; key id = 56954
                        3600    SIG     KEY 5 2 3600 20030510221048 (
                                        20030410221048 49601 freeswan.nl.
                                        [...] Lzktbwf51HlqD+sQ3w== )
                        3600    SIG     KEY 5 2 3600 20030510221048 (
                                        20030410221048 56954 freeswan.nl.
                                        [...] G/z+MdVYVsafdmXgh9+O )
                        3600    NXT     www.freeswan.nl. NS SOA MX SIG KEY NXT
                        3600    SIG     NXT 5 2 3600 20030510221048 (
                                        20030410221048 56954 freeswan.nl.
                                        [...] cNkca6K/mc9M32VD6gRp )
www.freeswan.nl.        3600    IN A    193.110.157.9
                        3600    SIG     A 5 3 3600 20030510221048 (
                                        20030410221048 56954 freeswan.nl.
                                        fVXogRSsfmm5SStZ4rok )
                        3600    NXT     freeswan.nl. A SIG NXT
                        3600    SIG     NXT 5 3 3600 20030510221048 (
                                        20030410221048 56954 freeswan.nl.
                                        [...] bXLB4JYJxOfMWXehBYS7 )
```

The Delegation problem

- The Parent should securely delegate authority of the child zone
  - Parent cannot give a “non-authoritative” answer
- Parent cannot not sign child zone data
  - It has no private key of child
- Parent should not sign child zone data
  - It is not authoritative for child zone
- Parent will need to serve NS (and perhaps glue) records of child zone
- Answer needs to be secure

DNSSEC: The DS record

```
RRlabel  TTL IN DS <key tag> <algorithm> <20 bit SHA-1 Digest>

reeswan.nl.      345600 IN NS ns.xtdnet.nl.
reeswan.nl.      345600 IN NS ns1.xtdnet.nl.
reeswan.nl.      345600 IN DS 49601 5 1 (
                     C7D3B76F7DEE10E6A73B7D0F6EDAF55FFF60CA78 )
reeswan.nl.      345600 IN SIG DS 1 2 345600 20030416070311 (
                         20030409070311 6869 nl.

W2pmK7IGF1W7SDJxyyTep707lDRQ36IEkmyEhezJO72U
                 3g1YeWTI4r5lSAOkGW/+u74FRuQgMFzYzRisCZKYCiBm
                 rNiatRg+TTf9+yzJcqg9A2CuygNBi8I7aVloYxsM+qri
                 9J1CJQuxAzbKLPAppQw4UP1VOiB4NvHWG2jwFNw= )
```

These are all the freeswan.nl records at the parent

Delegation fixed: chain of trust

- In an ideal world: Only one trusted key is needed
  - The root (“.”) key
- In the real world: Secure entry points
- Your world: Make your own trusted key(s)

```
trusted-keys {
“nl.” 256 3 1
“AQOtBQXOH5L/wmOt01PuxXAfSk1bw/dneW
PoCyl4yi8tLCjz+DkAs0mz AAvd9XUNp
YDaf5KTciSs9254oeiE0s0FuYbxS4nm7
veZSPCgWoHULFNJ tKPNeb4EEblNkAsE
GagwQJoIrjlAYKx4CEn3hPwElUlVko23
I5tSSPPs sxrVnQ==”;
};
```

Problem: Time is not on our side

- Small keys can be attacked using brute force
- Large keys are strong, but CPU expensive
- Keys can become useless
  - Key can be stolen, lost or compromised
  - Key can be based on impure random
  - Key can leak information when in use (DSA)
- Keys will need to be replaced at some point
  - Parent needs to be (securely) informed to update DS record
  - We want to minimize parent <-> child interaction
  - Cache, TTL, Signature expiration: Both keys are needed at the same time

Two keys: ZSK and KSK

- One Zone Signing Key (ZSK, 768bit, one month)
  - 768bit
  - Validity of one month
  - Signs all RRsets in the zone (including KEY records)
  - Can be changed without parent notification
- One Key Signing Key (KSK, 2048bit, one year)
  - Parent's DS record points to this key
  - 2048bit
  - Validity of one year
  - Only signs the key records
  - Must inform parent when this key changes

Scheduled ZSK Rollover

	normal	prepare	rollover
parent:	DS(KSK)	DS(KSK)	DS(KSK)
child:	KSK	KSK	KSK
	ZSK1	ZSK1, ZSK2	ZSK2
	KSK(KSK,ZSK1)	KSK(KSK,ZSK1,ZSK2)	KSK(KSK,ZSK2)
	ZSK1(zone)	ZSK1(zone)	ZSK2(zone)

Scheduled KSK Rollover

	normal	prepare	rollover
parent:	DS(KSK1)	DS(KSK1)	DS(KSK2)
child:	KSK1	KSK1,KSK2	KSK2
	ZSK	ZSK	ZSK
	KSK1(KSK1,ZSK)	KSK1(KSK1,KSK2,ZSK)	KSK2(KSK2,ZSK)
	ZSK(zone)	ZSK(zone)	ZSK(zone)

Unscheduled Rollover

PANIC!!!

- Have emergency procedure ready!
- Have spare KSK in zone for emergency rollover?
- Contact everyone who has your key as trusted key!
- Contact children!
- Short TTL's and short SIG lifetime help contain disaster
- Emergency out of bound contact needed with parent

Setup bind

- Only use latest snapshot on signer machine
  - As of writing: bind-9.3.0s20021217 [CHECK]
  - ./configure –with-openssl
  - Threads broken in latest snapshot, use –disable-threads
- Do not use "host" or "nslookup"
  - For "host" like output, use "dig +multiline"
  - For dnssec, use "dig +dnssec"
  - To ask for data without checks, use "dig +cdflag"
- You can use stable bind8/9 to serve secure zones
  - Note: bind8 does NOT serve data with expired SIG record. This data will disappear on bind8. Bind9 serves data with expired SIG records

Bind: Create secure zonefile

~> dnssec-keygen -a RSASHA1 -b 2048 -n ZONE freeswan.nl
Kfreeswan.nl.+005+49601
~> dnssec-keygen -a RSASHA1 -b 768 -n ZONE freeswan.nl
Kfreeswan.nl.+005+56954
(creates .key and .private files)

~> cat *key >> /var/named/freeswan.nl
(increase serial number in zone)

~> dnssec-signzone -o freeswan.nl -k Kfreeswan.nl.+005+49601.key
/var/named/freeswan.nl Kfreeswan.nl.+005+56954.key

(upload to master and change named.conf to load zone “freeswan.nl.signed” instead of “freeswan.nl”)

Test: dig +multiline +dnssec -t key freeswan.nl @ns.xtdnet.nl

IPsec: Opportunistic Encryption using DNSSEC

Secure .nl: http://secreg.nlnetlabs.nl/

File Edit View Go Bookmarks Tools Help

http://secreg.nlnetlabs.nl/cgi-bi

Linux

SECREG: .NL

make secure | block | key rollover | change secc | whois notify

tools | zone listing | forms | procedures | about

Make a .nl domain secure

Zone key(s) from fnl.nl

Please select the KEY you want to use as your zone KEY

```
fnl.nl. 3600    IN      KEY     256  3  5 (
                        AQPAUnF5M/w9du2kB1ASmFsEQmuUPeIQ/tVf
                        0pZwT8WwOkx9kH8hm0/Of3YNU1FZ+gtIkxz7
                        qlkpqvYq6mYZuvLXBgFFU4IqwAUi2C4kdzyg
                        JRt3/roqZh5rbTBcCOoWMeHSIr6uL1NlRXFG
                        /petl7jRXFgXQKfeAzQzIUEUi4gFdw==
                        ) ; Key ID = 25541
```

```
fnl.nl. 3600    IN      KEY     256  3  5 (
                        AQO5rfR72fQkQFVn1kvYwhMPMmztyumSSCwU
                        PEA6dPN4amz8uo9BIwlbXEJKBzWCzAwYctBr
                        C6fzRxVItJ9spmlUcD/W2vMmOp9oIPJxOVRp
                        sW5qVuqdPdRna2n4oPPZ9PBvNjinLNFoBJnt
                        GLZLeZ3HrQ3d5nryrHO1EEabaD4Esw==
                        ) ; Key ID = 16217
```

Proceed | Clear

IPsec: Opportunistic Encryption using DNSSEC

Secure .nl: http://secreg.nlnetlabs.nl/

File Edit View Go Bookmarks Tools Help

http://secreg.nlnetlabs.nl/cgi-bi

Linux

SECREG:.NL

make secure | block | key rollover | change secc | whois notify

tools | zone listing | forms | procedures | about

Key rollover

Domain name: ct.nl

Authentication

Sign this record: 3600 IN TXT "random txt m8OfNOLtmpXs2aQ"

use sign.pl

Paste output here (extra whitespace allowed)

```
ct.nl.   3600    IN      SIG     TXT  5  2  3600  20030315125047 (
                         20030213125047 35861  ct.nl

KHR718ehTtidLkTUbnf+NPPJymSo2jvrVTkP0vjKySFTe

v6tRoCPkvQ2pZxnn7aR3h0CrHAWnva7WCUJ4GzdPJD2fx
                         M0zYkH722evhWnayy2TTgGQ81cpVnbag/95/0R )
)
```

Proceed | Clear

Securely Resolving .nl domains

Use bakbeest.sidn.nl or alpha.nlnetlabs.nl

```
; <<>> DiG 9.3.0s20021217 <<>> +dnssec +multiline -t a www.fnl.nl @bakbeest.sidn.nl
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 16017
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;www.fnl.nl.            IN A

;; ANSWER SECTION:
www.fnl.nl.             2998 IN A 193.110.157.9
www.fnl.nl.             2998 IN SIG A 5 3 3600 20030318164405 (
                                20030216164405 25541 fnl.nl.
                                gkNbAoyANPDh+3A71C7T1x2kApSax9gni/NJ2REbmUmO
                                qeZ6rTmePn+45qc6HdIvRitoUKiuyEHfBq5M9ilqoZ76
                                ASICiXBqVJcvXvayKUDZJd7v6YN6SwcOi1RJMNazIBiG
                                wCWUDmCr4keujIelSQFL5ZB3feVba7sv9KM1ueM= )

;; Query time: 50 msec
;; SERVER: 193.176.144.170#53(bakbeest.sidn.nl)
;; WHEN: Wed Feb 19 00:55:42 2003
;; MSG SIZE  rcvd: 221
```

DNSSEC experiment for com/net/org domains

See: www.dnssec.verisignlabs.com

- No personal experience
- Still needs pre-DS dnssec-makekeyset tool which is no longer in the bind snapshot (java signer available)
- Trusted keys need to be pulled from zonedata
- Experiment is purely technical, no policy issues addressed.

Mass Deployment

- Net::DNS and Net:DNS::SEC (in CPAN)
  - Has bug for large TXT records (opportunistic encryption records)
- DNSSEC-Maint and DNSSEC-Maint-Zone (RIPE NCC)
  - Supports notion of KSK and ZSK and 2 step rollovers
  - Very easy to use and maintain keys, zones and rollovers (!)
    (we currently maintain about 150 dnssec zones)
    - Maintkeydb create RSASHA1 zonesigning 768 freeswan.nl
    - Dnssigner -o freeswan.nl /var/named/freeswan.nl
    - maintkeydb rollover freeswan.nl zonesigning yes
    - Maintkeydb rollover freeswan.nl zonesigning yes [check]
- You can use stable bind8/9 to serve secure zones
  - Note: bind8 does NOT serve data with expired SIG record. Records will disappear when SIG expired on bind8. Bind9 serves data with

Changes in organisation

- Location of DNS Zonefile now on secure signer machine
- New task: maintain secure zones
  - Don't let the SIG records expire!!
- No more direct edits of zonefile
  - Or extra step if generating from database (and how secure is the database machine?)

Applications using DNSSEC

- FreeS/WAN: IPsec Opportunistic Encryption
  - supports dnssec since version 2.01
  - http://untappable.xtdnet.nl/ does DNSSEC aware OE
- OpenSSH: host keys in DNS
  - Only patch for old version currently available
- ISC dhclient: secure dynamic updates
- NXT-walk software (tsk tsk)
- Browser plugins???

DNSSEC References

- Bleeding edge: http://www.ripe.net/disi/
- Documentation:
  - http://www.xtdnet.nl/paul/blackhat/ (updates of these slides)
  - http://www.xtdnet.nl/paul/dnssec/
  - http://www.ripe.net/training/dnssec/
  - http://www.dnssec.net/
- Software
  - ftp://ftp.isc.org/isc/bind9/snapshots/
  - http://www.miek.nl/projects/resolver/resolver.html
- Secure Registery experiments
  - http://secreg.nlnetlabs.nl/

RFC 3445: KEY record limitation

RRlabel TTL IN KEY <flags> <protocol> <algorithm> <key material>

KEY record may only be of protocol number 3 (DNSSEC). All other applications need to use the APPKEY record. IETF feared too many application keys in APEX of zone.

Result: Breaks all FreeS/WAN Opportunistic Encryption machines on the Internet!

IETF basicly broke the first and only DNSSEC aware application !!!

FreeS/WAN will likely ignore RFC3445, but will reverse the order of lookups. Where it used to first try KEY record, then TXT record, as of version 2.01 it will first try TXT record, then fallback on nonrfc compliant KEY record).

Part two: Opportunistic Encryption

- Opportunistic Encryption: IPsec for the masses

http://www.freeswan.org/freeswan_snaps/CURRENT-SNAP/doc/quickstart.html

Overview presentation
PART TWO: IPsec

- IPsec basics
- What is Opportunistic Encryption
- Protecting the net with OE
- Installing and Configuring OE
- Securing the wireless: WaveSEC
- Testing OE

IPsec in a nutshell

- Phase 1: Diffie-Hellman Key Exchange
  - Ensures privacy
  - Vulnerable to Man in the middle attack
- Phase 2: Identity exchange and verification
  - Exchange ID's
  - Both parties independantly check ID.
  - Both parties agree on encryption method, eg PreShared Secret (PSK) or RSA key based. PSK or RSA key of other party needs to be known beforehand!
  - Both parties agree on a stream cipher for the encryption, eg AES,3DES
  - Both parties agree to pass along certain packets, eg 10.0.1.0/24
- ID's originally were just text based, eg "Paul"

Opportunistic Encryption

- Goal: IPsec connection without prior arrangement or exchange of information with parties you never knew before, allowing mass deployment of computers on the internet to talk securely and privately. Force eavesdroppers from passive to active attacks.
- Support the notion of one OE security gateway protecting a whole subnet.
- Information needs to come from "trusted third party" to prevent MITM attack in Phase 1.
- The trick: Use the IP address as a pointer to external information.
  - Obviously we cannot use PSK, since everyone would be able to fetch the secret. We need to use a public key system, such as RSA or X.509

DNS to advertise OE-capability and RSA key

- Put special TXT record in the reverse, eg:

```
; RSA 2192 bits   bofh.xtdnet.nl   Thu Oct 17 12:32:33 2002
17.157.110.193.in-addr.arpa. IN TXT "X-IPsec-Server(10)=193.110.157.17"
" AQOkF1Ggd4iFfI2nQxJYbN9HGDhhIAKIXCoAPX+z+fNI9j7rxxR9QhThIZZeOx
+X9WB4hIa8/8xAnELmc RhkD8CxfznE4tCQ/Ws+9ibXUdD8Wee3JusSMrmLCu
IScNUQuBtRe+l+nn16dzvw3/PGB67gid+AvGvJJJnxiFjibd/4ayVebJRj 6Bu/FRex
pXr3jEgg0TJwxu9y1xBR7i0tRYCdSQPKNClNrgmX7YZTp4bu6gizhil63/sR6"
```

- ISC DHCP server and client support this with DNS Dynamic Updates.
- If two parties both have their RSA key in the reverse DNS, they can fetch each other's key and setup a secure connection.
- FreeS/WAN upto 2.00 also supported putting the RSA key in a KEY record instead of TXT record, but again: IETF killed this with RFC3445. FS 2.01 only falls back on the KEY record.

Most do not control their reverse DNS: iOE

- If Phase 1 ID is of the format "@FQDN" (eg @vaio.xtdnet.nl) then do not use reverse dns of IP address, but look for a TXT record in the forward zone eg:

; RSA 2192 bits vaio.xtdnet.nl Thu Oct 17 12:32:33 2002
vaio.xtdnet.nl. IN TXT "X-IPsec-Server(10)=127.0.0.1""AQOkF1Ggd4iFfI2nQx
JYbN9HGDhhIAKIXCoAPX+z+fNI9j7rxxR9QhThIZZeOx+X9WB4hIa8/8xAnELmc
RhkD8CxfznE4tCQ/Ws+9ibXUdD8Wee3JusSMrmLCuIScNUQuBtRe+l+nn16dzvw
3/PGB67gid+AvGvJJJnxiFjibd/4ayVebJRj6Bu/FRexpXr3jEgg0TJwxu9y1xBR7i0tR
YCdSQPKNClNrgmX7YZTp4bu6gizhil63/sR6"

- For now, at least the answering party needs reverse TXT, otherwise man in the middle attack possible.
  - We call this: initiator-only OE (iOE)
- If two parties both have their RSA key in the reverse DNS, they can fetch each other's key and setup a secure connection.
- Note: if remote initiates, one clear packet will go over the wire

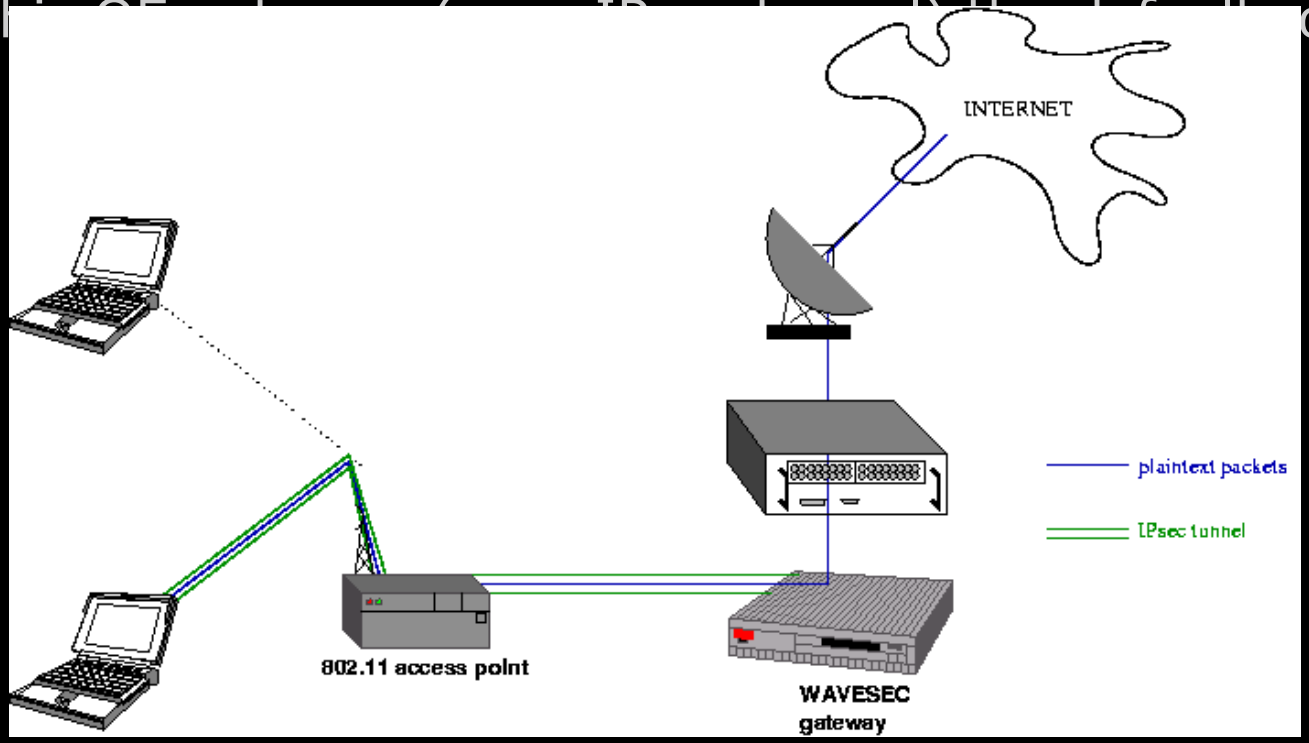
OE on the responding side

- Determining OE takes time (one or more DNS lookups).
- Doing DNS lookups on very busy webservers takes resources.
- Most OE attempts on responder will fail anyway, since otherwise client would have initiated to us to begin with.

- Passive OE: Only respond to incoming requests, do not start outgoing requests. Ideal for busy webservers.
  (www.freeswan.org does passive OE)
- Note: two passive OE servers will talk in the clear!!

OE to protect 802.11: WaveSEC

- Use DHCP and dynamic updates to automaticly find OE gateway and put our key in the reverse. Then we can setup OE to OEGW
- Make th[illegible]oute



OE getting ugly: NAT-traversal

- The reverse DNS becomes meaningless, we must switch to iOE or we will fail to connect to OE-capable hosts.
  (FreeS/WAN tries to detect private space IP)
- 'normal' NAT-traversal can then happen.
- If NAT gateway is OE-capable, there is no need for OE on the inside machines (They can even be Windows,IP-phones,etc)

OE getting ugly: OE & NAT to protect subnet

- Typical setup: ADSL with one IP, no control of reverse
- Run SNAT on internal interface
  (not on external which is more common)
- Run iOE on public interface (eg ppp0, eth1)
  (perhaps slighly lower MTU if going through more tunnels, such as PPTP)
- Remember: first inbound packet will be in the clear!

OE getting ugly: SNAT to protect subnet

- If using SNAT, we can protect another subnet without becoming critical infrastructure
  (SNAT is needed because otherwise response packets from server to OE-client will flow back in the clear, and dropped at the client)

DIA IMAGE

Configuring OE: requirements

- PF_KEY sockets with OE extensions
  - KLIPS, the FreeS/WAN kernel code (Linux 2.0,2.2,2.4,2.5)
  - Linux 2.5 native IPsec stack (2.5.69+)
- OE support in the IKE daemon
  - Pluto, the FreeS/WAN userland keying daemon
  - ipsec-tools (for native 2.5) doesn't support OE yet.
- RSA public key in the DNS
  - For full OE: TXT record in the reverse
  - For iOE: TXT record in any forward
  - For compatibility with FreeS/WAN 1.9x-2.01: KEY records
- No filters for UDP-500 (IKE), UDP-4500(NAT-T IKE), PROTO50(ESP) and PROTO51(AH)

OE: How it works: The routing hack

- FreeS/WAN uses a routing hack to trick packets to enter the ipsec device (pre netfilter/netlink legacy)

  Example with OE disabled:
  # **ip ro li**
  193.110.157.0/24 dev eth0 proto kernel scope link src 193.110.157.17
  127.0.0.0/8 dev lo scope link
  default via 193.110.157.254 dev eth0

  When OE is enabled:
  # **ip ro li**
  193.110.157.0/24 dev eth0 proto kernel scope link src 193.110.157.17
  193.110.157.0/24 dev ipsec0 proto kernel scope link src 193.110.157.17
  127.0.0.0/8 dev lo scope link
  0.0.0.0/1 via 193.110.157.254 dev ipsec0
  128.0.0.0/1 via 193.110.157.254 dev ipsec0
  default via 193.110.157.254 dev eth0

- Note: To do OE within the same subnet, repeat this trick for

OE: How it works: The routing hack

- Old style route command output (you all use **ip** command right?)

```
# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref Use
Iface
193.110.157.0   0.0.0.0         255.255.255.0   U     0      0        0
eth0
193.110.157.0   0.0.0.0         255.255.255.0   U     0      0        0
ipsec0
127.0.0.0       0.0.0.0         255.0.0.0       U     0      0        0
lo
0.0.0.0         193.110.157.254 128.0.0.0       UG    0      0
0 ipsec0
128.0.0.0       193.110.157.254 128.0.0.0       UG    0      0        0
ipsec0□0.0.0.0          193.110.157.254         0.0.0.0         UG
0       0       0 eth0
```

OE: Look at the internal state: eroute

- To see which connections are plaintext or crypted, cat /proc/net/eroutes:

```
5  193.110.157.17/32  -> 0.0.0.0/0                    => %trap
3  193.110.157.17/32  -> 131.174.124.204/32 => %pass
2  193.110.157.17/32  -> 193.110.157.10/32  => tun0x1006@193.110.157.77
6  193.110.157.17/32  -> 193.110.157.2/32           => %pass
0  193.110.157.17/32  -> 193.110.157.250/32 => %hold
12 193.110.157.17/32  -> 193.110.157.77/32  => tun0x1002@193.110.157.77
2  193.110.157.17/32  -> 193.110.157.9/32           =>
tun0x1004@193.110.157.77
```

- %trap : catch these packets and process
- %pass : Other side doesn't support IPsec/OE, send out in clear
- %hold : Other side advertises OE, but failed to setup a tunnel with, attack or misconfiguration, do not send anything

OE: FreeS/WAN 2.x Policies

- FreeS/WAN 2.x allows you to set more elaborate policies, see

/etc/ipsec.d/policies/block (never talk to these CIDR's)
/etc/ipsec.d/policies/clear (no OE, eg if they block IPsec)
/etc/ipsec.d/policies/clear-or-private (Passive OE CIDR's)
/etc/ipsec.d/policies/private (encrypt to them or dont talk)
/etc/ipsec.d/policies/private-or-clear (try OE, else allow clear)

Configuring OE: Installing FreeS/WAN

- Most distributions already have FreeS/WAN support
  - Debian, SuSe, Mandrake, Connectiva, etc. have binaries.
  - RedHat binaries on ftp://ftp.xs4all.nl/pub/crypto/freeswan/binaries
- Install from source

Installing FreeS/WAN kernel plus userland

- Download latest FreeS/WAN (or superfreeswan if you also want NAT-T or X.509 support)
- Install kernel-source and gmp with header files (gmp-devel)
- Build a standard kernel in /usr/src/linux
- untar and go into the freeswan directory
- issue a '**make oldmod**' (this builds and installs freeswan)
- **cd /usr/src/linux ; make modules_install** (if building ipsec support as module)
- If you added NAT-T support, you must use '**make oldgo**' and also install the new kernel itself, since the ESPinUDP patch changes some fundamental network structures.

Installing FreeS/WAN userland on native 2.5

- Download latest FreeS/WAN (do not use superfreeswan, since 2.5 kernel doesn't yet support NAT-T or X.509)
- Install kernel-source and gmp with header files (gmp-devel)
- untar and go into the freeswan directory
- issue a '**make programs**' (this builds and installs freeswan)

Configuring OE: Putting keys in the DNS

- If you have a file /etc/ipsec.secrets, your keys have already been generated by starting FreeS/WAN. If not, issue:
  # **ipsec newhostkey --output /etc/ipsec.secrets**
- To display a proper reverse dns record for your host, issue:
  # **ipsec showhostkey --txt 1.2.3.4**
  where 1.2.3.4 is the IP address of your OE gateway, or your own IP address if there is no OE gateway
- To display a proper forward dns record for your host, issue:
  # **ipsec showhostkey --txt 127.0.0.1**
- **To displau an old style KEY record for the forward, issue:**
  # ipsec showhostkey
- **Alternatives:**
  # ipsec mailkey --me my@address.tld --forward hostname.domain.tld

Configuring OE: FreeS/WAN 1.9x

- Add or enable the following connection(s) in /etc/ipsec.conf

```
conn client-to-anyone              # for our client subnet
   leftsubnet=10.0.2.0/24          # any single client in our subnet
     also=me-to-anyone             # rest is same as for SG

conn me-to-anyone                          # for ourself
   left=%defaultroute              # outside interface
   right=%opportunistic  # use DNS records to find keys
   authby=rsasig                   # almost always the right choice
   keyingtries=2     # don't be persistent -- peer might
disappear
   auto=route       # enable at ipsec startup
   #leftid=@vaio.xtdnet.nl         # only needed for iOE
```

Configuring OE: FreeS/WAN 2.x

- OE is enabled per default for 2.x in /etc/ipsec.conf.
- You can disable it by creating a connection called **OEself:**

```
conn OEself
   auto=ignore
```

- **iOE also needs specific OEself definition:**

```
conn OEself
   left=%defaultroute
   leftrsasigkey=%dnsondemand
   leftid=@vaio.xtdnet.nl
   right=%opportunistic
   rightrsasigkey=%dnsondemand
   auto=route
```

Configuring WaveSEC: DHCPD

- ISC 3.0.1rc9 or higher
- http://www.wavesec.org/patches/dhcp-3.0.1rc9-oe-key.patch
  (might be integrated by now)
- in /etc/dhcpd.conf

```
option oe-key     code 159 = string;
option oe-gateway code 160 = ip-address;
on commit {
  if (not static and ((config-option server.ddns-updates = null) or
(config-option server.ddns-updates != 0))) { if exists oe-key {
        set ddns-rev-name = concat (binary-to-ascii (10, 8, ".", reverse (1, leased-address)),
".", pick (config-option server.ddns-rev-domainname, "in-addr.arpa."));
        set full-oe-key = option oe-key;
        switch (ns-update (delete (IN, 25, ddns-rev-name, null), add (IN, 25,
      ddns-rev-name, full-oe-key, lease-time / 2)))  {
default:
          unset ddns-rev-name;
          break;
        case NOERROR:
          on release or expiry {
            switch (ns-update (delete (IN, 25, ddns-rev-name, null))) {
```

Configuring WaveSEC: Bind

- **See: www.wavesec.org/dyndns.phtml**
- **And: http://ops.ietf.org/dns/dynupd/secure-ddns-howto.html**

Configuring WaveSEC: FreeS/WAN

- See http://www.wavesec.org/
- Add both interfaces to the IPsec machinery in ipsec.conf:
  interfaces="ipsec0=eth0 ipsec1=wlan0"
- To enable your WAVEsec clients to connect, create a connection description for each IP address that will be served. A future version of FreeS/WAN may reduce this to a single definition, but this is not yet implemented. For each, you will need a conn like:

```
conn host66-to-world
        left=192.139.46.254   # IP of WAVEsec gateway
        leftsubnet=0.0.0.0/0
        right=192.139.46.66   # IP of potential client.
        keylife=1h            # IP may be reused after 1 hour idle
        rekey=no
        auto=add
```

Configuring WaveSEC: FreeS/WAN server

- **Exempt DHCP and DNS packets from IPsec**

```
#!/bin/sh
iptables -A PREROUTING -t mangle -p udp -s 0.0.0.0/0 -d 192.139.46.64/29 --sport 53 -j MARK --set-mark 1
iptables -A PREROUTING -t mangle -p udp -s 0.0.0.0/0 -d 192.139.46.64/29 --sport 67:68 -j MARK --set-mark 1
iptables -A PREROUTING -t mangle -p icmp -s 0.0.0.0/0 -d 192.139.46.64/29 -j MARK --set-mark 1
iptables -A OUTPUT -t mangle -p udp -s 0.0.0.0/0 -d 192.139.46.64/29 --sport 67:68 -j MARK --set-mark 1
iptables -A OUTPUT -t mangle -p udp -s 0.0.0.0/0 -d 192.139.46.64/29 --sport 53 -j MARK --set-mark 1
iptables -A OUTPUT -t mangle -p icmp -s 0.0.0.0/0 -d 192.139.46.64/29 -j MARK --set-mark 1

ip rule add fwmark 1 table dhcpd
ip route add 192.139.46.64/29 dev wlan0 table dhcpd
```

Configuring WaveSEC: FreeS/WAN Client

- **Install and configure FreeS/WAN**
- **Patch for dhclient: www.wavesec.org/download/dhcp-3.0.1rc9.freeswan.tar.gz**
- **Patch for RedHat ifup script www.wavesec.org/download/sbin-ifup.dhclient.rhl7.3.patch**
- **Run: ipsec showhostkey –-dhclient > /etc/dhclient.conf**

Configuring WaveSEC: FreeBSD/KAME Client

- Though it requires some manual settings it should work with wavesec.
- See http://www.wavesec.org/kame.phtml
- Also see: http://www.wavesec.org/KAME-WAVEsec.txt
- The same probably applies to Linux 2.5 native IPsec with ipsec-tools

Configuring WaveSEC: Other Clients

- We will try to get something running using X.509 certificates for Windows machines. If we succeed, we will publish the information on www.wavesec.org
- OpenBSD should be able to work with WaveSEC
- SSH IPsec Express should work
- SSH Sentinel should work

Testing OE

- # ipsec verify

```
Checking your system to see if IPsec got installed and started correctly
Version check and ipsec on-path
[OK]
Checking for KLIPS support in kernel                    [OK]
Checking for RSA private key (/etc/ipsec.secrets)          [OK]
Checking that pluto is running                          [OK]
DNS checks.
Looking for forward key for bofh.xtdnet.nl              [NO KEY]
Does the machine have at least one non-private address  [OK]
Two or more interfaces found, checking IP forwarding    [OK]
Checking NAT and MASQUERADING                           [N/A]
```

- Note: verify doesn't catch all problems.

Testing OE

- **http://untappable.xtdnet.nl/**
  Once you are connected, telnet to the gopher port, you will see something like:

```
Trying 193.110.157.74...
Connected to untappable.xtdnet.nl (193.110.157.74).
Escape character is '^]'.
Results of query on 193.110.157.74 -> 192.139.46.38 with seq 2
Received reply of 33124 bytes.
Strength:   32
Bandwidth:  32
authdetail: 0
esp_detail: 3
comp_detail: 0
credentials: 1
      DNSSEC identity: 192.139.46.38 (SIG KEY 1 6 7200
20030613025618 20030514025618 28815 46.139.192.in-addr.arpa.
Lv8xM9ihADevmp8Zf7X7
```

Testing OE: other sites

- http://oetest.freeswan.org/
- http://oetest.freeswan.nl/
- http://www.xtdnet.nl/paul/
- ipsec verify –host mail.xtdnet.nl

- irc.sandelman.ca (only allows OE connections)

Known caveats

- When OE is started, the first connections will fail because it will trigger lots of other OE's to the various dns servers, and all will fail until %pass routes to the dns servers are created.
- Starting FreeS/WAN before the internet connection doesn't work properly yet. Restart FreeS/WAN after inserting a pcmcia card.
- People often make mistakes pasting in the OE dns records.
- Do not change the IP address in an OE record. If a digit gets added, it can break the record.
- 2.00 and 2.01 cannot talk OE to 2.02
- KEY versus TXT issues as a result of RFC3445.
- NAT'ing IPsec packets to hell
- MTU/fragmentation problems with tunnels within tunnels.
  Seems to mostly happen to people who do ADSL/PPTP/IPsec

Debugging OE

- Check your logfiles (/var/log/secure, /var/log/messages)
- Check your logfiles AGAIN (check startup errors)
- Run "ipsec verify"
- Disable any firewall rules to test if you are dropping packets
- Use "ikeping" to test if your ISP filters udp 500. Turn off FreeS/WAN before trying ikeping.
- Post the output of "ipsec barf" on a website and post a URL to the mailinglist with an explanation of what you are trying to do.
- Try the people at FreeNode (formerly OpenProjects) on the irc channel #freeswan

DO NOT ENABLE KLIPSDEBUG or PLUTODEBUG !!!

Try out OE at the conference!

- Grab me during the conference if you need help.
- We should be running WaveSEC
- We recommend setting up DNSSEC on your local laptop, trusted keys for the reverse will published on: http://www.xtdnet.nl/paul/conference
- If your connection is NAT'ed, talk to us to get a real IP.

- **Enjoy your privacy :)**