



# The Future of IPsec on Linux

Ken Bantoft <ken@xelerance.com>  
Xelerance Corporation / Openswan Team  
<http://www.xelerance.com/talks/linuxtag2004/>

# Topics

- History
- Kernel Stacks
- Userland Tools
- Current Issues
  - SPD/Routing
  - Doing NAT
  - Tunneling Through NAT
- New Developments
  - netfilter+ipsec patches
- Future Developments
  - IKEv2, Async Hardware Acceleration, more complicated policies
  - OE, IPsec on demand

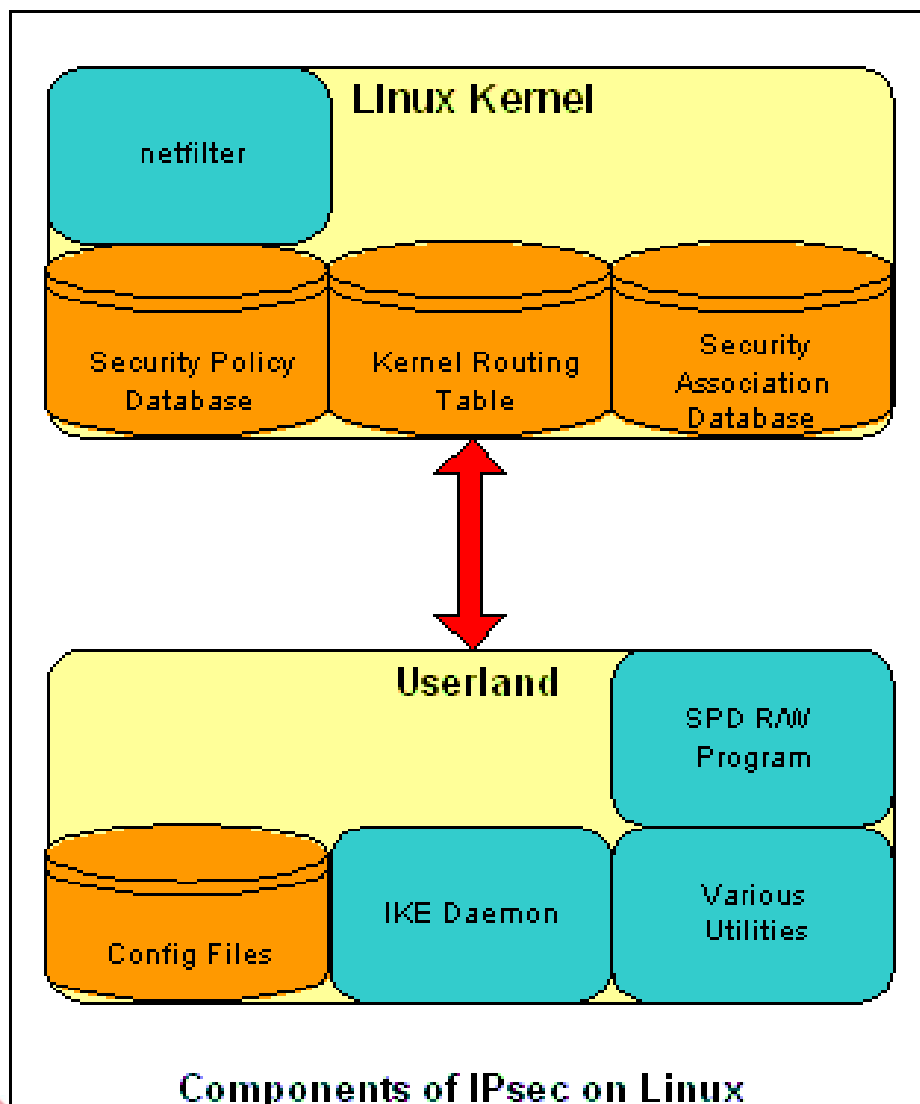
# IPsec History

- Lots of politics
- FreeS/WAN started in 1997 for kernel 2.0
  - Supported 2.0, 2.2, 2.4 and 2.6 Kernels
  - Officially over April 2004
  - Numerous unintegrated patches
  - “Anti US” management team refused US patches
- Forked by USAGI and others for academic and commercial reasons

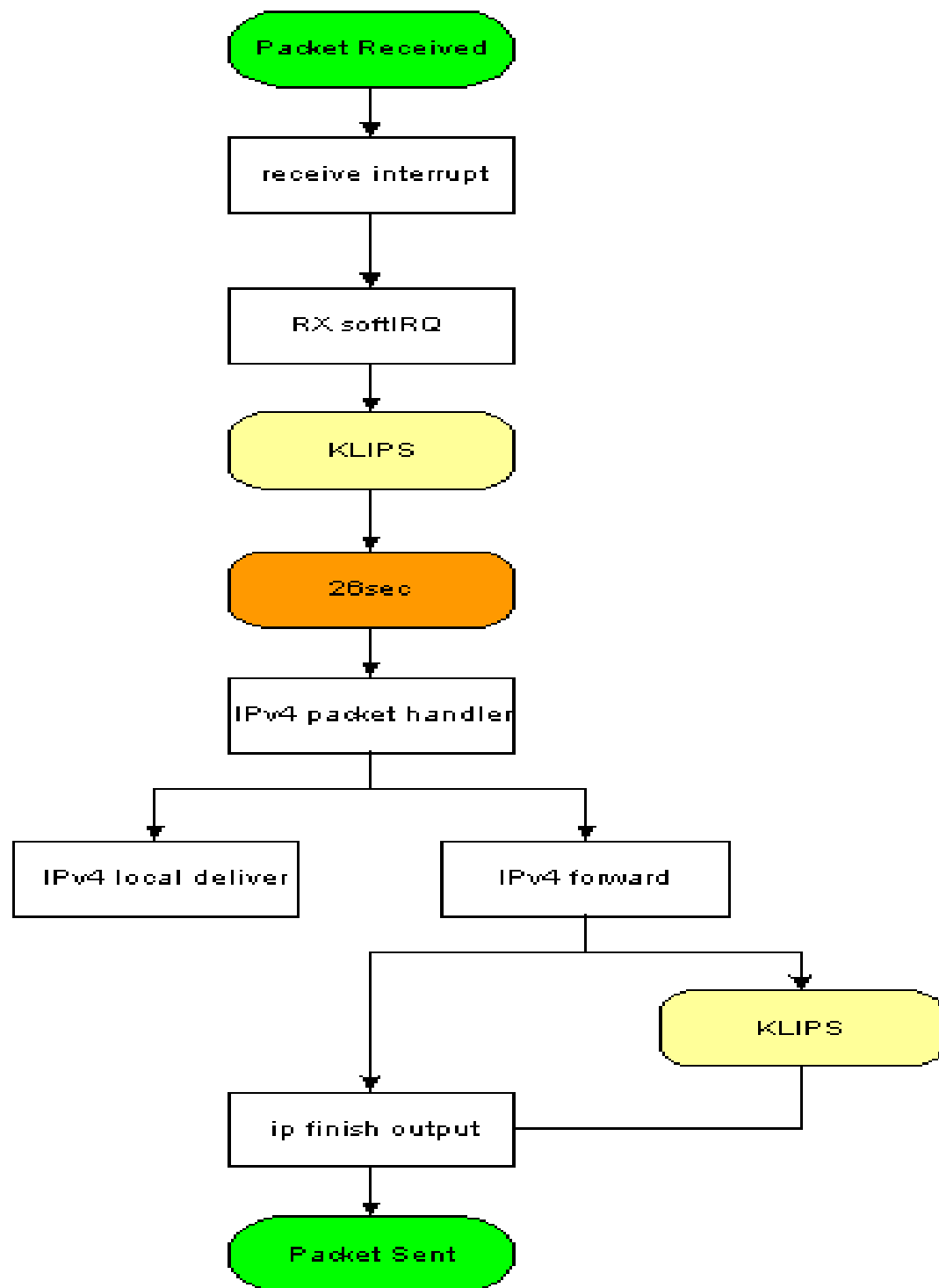
# IPsec History Con't

- During mid-2.5 Kernels, a “new” IPsec stack was integrated
  - pieces from USAGI stack (IPv6) and others
  - 'afkey' implementation
- setkey + racoon ported from \*BSD to Linux
- Openswan started June 2003 by former FreeS/WAN developers and other members of the community.

# Kernel Stacks



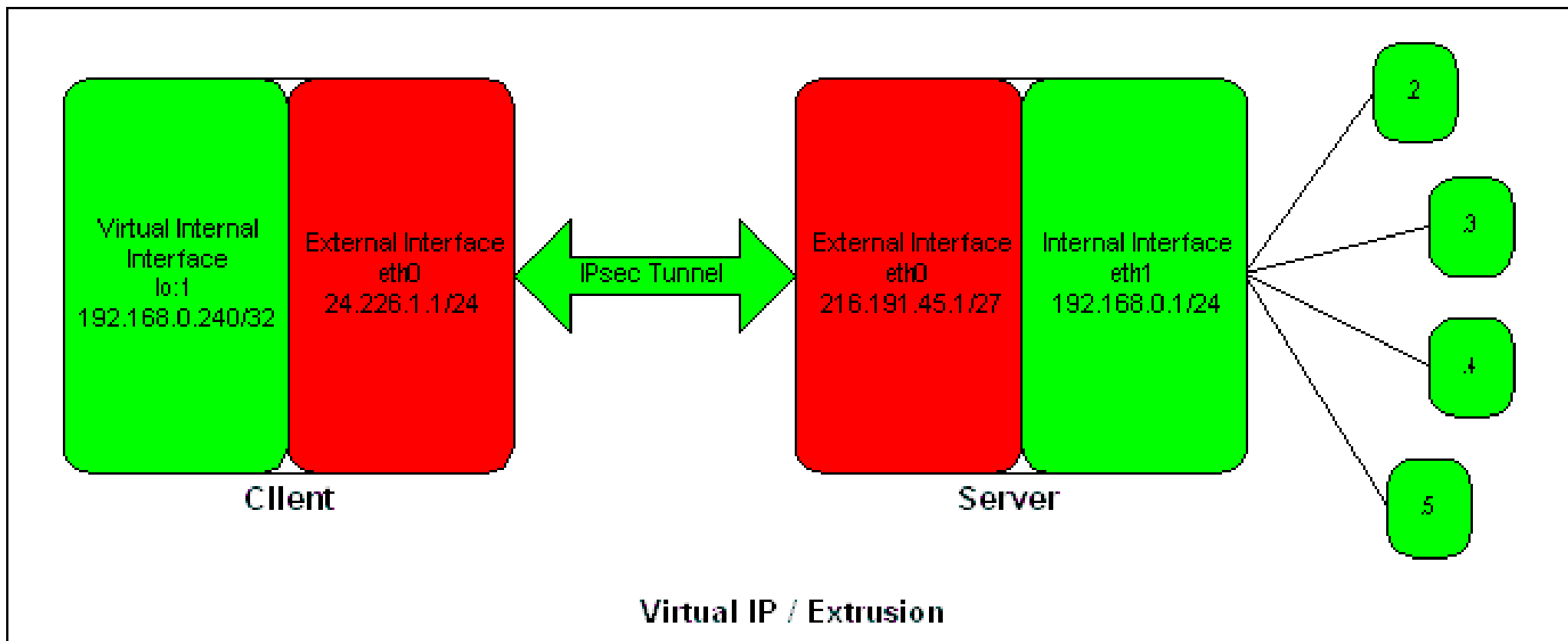
- **KLIPS**
  - ipsec# device
  - 'bump in the stack'
  - 'routable' device
- **26sec**
  - no more fun with network tricks
  - no ipsec0 device, which seems to confuse users



IPv4 Packet Flow in The Linux Kernel

- KLIPS had 2 hooks into the packet processing code
- 26sec currently only has one
  - no way to 'see' packet before/after encryption
  - tcpdump cannot see the encrypted packets – troubleshooting is hard.

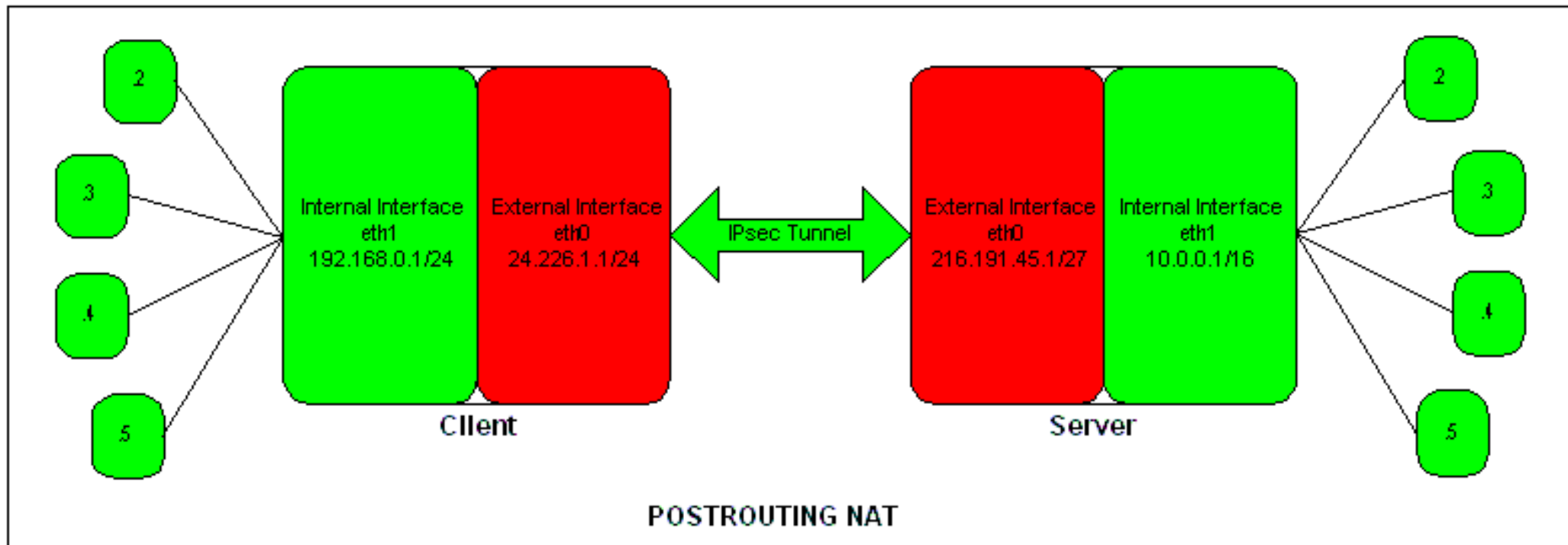
# Current Issues with 26sec



# Current Issues Con't

- The 26sec SPD does not currently allow you to 'exempt' a portion of an SPD entry from the tunnel. Configuring a setup like this, even if you assign the IP extrusion to the lo:1 interface, won't work. While the SA is established, you aren't even be able to ping locally!
- Thus you can't assign local LAN IP addresses to roadwarriors, unless you use a separate subnet, and don't include a tunnel for it (this also means you won't be able to have RW to RW communication)

# Current Issues Con't



- With the 26sec code, because the SPD lookups are before the kernel routing table, the packets are encapsulated before they hit any of the netfilter PRE/POSTROUTING hooks. This means SNAT/DNAT won't work, so this configuration will no longer work.

# NAT Traversal

- Currently 2 (or maybe 3 now) methods of doing the ESP in UDP encapsulation
- 2.4 Kernel method, by Arkoon Networks
- 2.6 Kernel method
  - Was included in 2.4 backport of 26sec
- 2.6 Kernel method by SuSE, recently merged
  - Will probably be included in next 2.4 backport
  - Hopefully the new 'standard' way of doing this

# New Developments

- netfilter+ipsec patches
  - Set of 4-6 patches, currently in netfilter pom-ng
  - Maintained by Patrick McHardy
  - Lets you do firewalling and NAT like KLIPS did, but by sending the packets back through the forward chain after crypt/decrypt

# Future Developments

- General
  - Asynchronous hardware via CryptoAPI
  - 26 Embrace and Extend
- Openswan Specific:
  - IKEv2
  - KLIPS v3/MAST – a per John Denkers spec
  - Integrated CA functionality, via simpleCA
  - CP Hybrid IKE support
  - Online live test capability

# Asynchronous Capabilities

- Offload crypto to hardware accelerators
- Let dedicated kernel thread do work rather than network bottom half
  - Something for the extra two CPUs on a Quad Xeon/AMD64 to do, after binding one CPU to each network interface
  - Provides for limiting CPU bandwidth used for crypto operations (useful on web or database servers...)
  - Higher bandwidth, but potentially higher jitter/latency.
- Continue to support current model for low latency applications.

# Extend 26sec

- The 26sec code came from KAME. KAME is a nice stack, but is too strict to RFC2401, and limits many common uses and interop with not-quite-compliant vendors
- Very good low-level crypto routines, and CryptoAPI support
- Stackable dst is a good idea, but SPD is in wrong place
  - Why is there yet-another-firewall?

# KLIPS on 2.6 Kernel

- Mostly a question of getting the Makefiles right. Minimal code changes needed. Nate Carlson submitted patches yesterday for all of these, so they will be integrated shortly
  - Provides support for ipsecX devices – good for legacy apps, and vendors with existing investments
  - Currently easier to firewall with
  - Later on ipsecX gets renamed to mastX
  - Support for 2.0 waning. 2.2 still available. 2.4 will be supported for three or four years.
  - KLIPS code has evolved a lot since 2000 - take a look at it.
- Novell.** The key to evolution is ease of testing.

# IKEv2

- Currently in IETF draft phase
  - EAP (Extended authentication protocols) support, as in 802.1x, FreeRadius, etc... no more XAUTH!
  - Load Balancing
  - Clarifies the use of X.509 for authentication
- MOBIKE extensions
  - Support multiple (dynamic) interfaces, for wifi/gprs roaming – IP address changes should be transparent

# Opportunistic Encryption

- Near RFC status
- Obvious solution to TCP RST attacks.
- We want to make it easy to ship it with every distro, even for installation mode
- Needs to co-exist with VPNs
  - This was a problem with the old FreeS/WAN goals

# Trivial VPNs

- WAY, WAY, WAY too hard to setup. Blame Microsoft and Cisco.
- X.509 CA's provide for distributed identity – but who cares who they are.
  - WHAT CAN THEY DO?
  - This is role for Attribute Authorities
  - SPKI, KeyNote, and pki4ipsec.
  - Look for experiments in the \*SWAN projects.
- SSH style “leap-of-faith” possible, where you can trust the other end.

# IPsec on Demand

- Imagine two laptops in a busy airport lounge
- Geek1 says to geek2, let me send you this file.
- Need security for this transaction. 802.11, Bluetooth, etc. have proven themselves unable to do this seriously.
- IPsec can and should be easy to setup, even on ad-hoc basis.

# Testing and more testing

- Testing is the major thing that distinguishes Openswan from other projects
- Currently run 126 tests nightly (in UML) on all trees:
  - Patching, Compiling, Installing, Running
  - Initial tests for most features now completed
- We spend a lot of our time on this
- Automated nightly testing makes it easy to add new features, and find bugs.
- Still have about 30% of KLIPS to cover, and 50% of pluto to cover.

# Q & A