Overview

- Man ipsec-tools
- Past
- Present
- Future
Ipsec-tools ???
IPsec in one single slide...

• IETF normalized security protocol
  – RFC2401 – 240x (“IKEv1”)
  – RFC4301 – 430x (“IKEv2”)
  – Lots of other RFCs and [expired] drafts....

• Security for IP
  – Peer's authentication
  – Confidentiality
  – Integrity

• At IP layer
  – “end to end”: Transport mode
  – “net to net”: Tunnel mode between gateways
    • Used to connect RFC1918 networks over internet
Okay, IPsec in two slides...

- Designed for gates with multiple tunnels
- Internet Key Exchange: dynamic negotiation
  - Authenticates peer (X509 or preshared secrets)
  - Generates session keys with lifetime
- Kernel/Userland Interface
  PFKeyV2 (RFC 2367)
  - Socket interface
  - Various more or less standardized extensions
- Mandatory for IPv6
Ipsec-tools (very quick) overview

• Userland tools for IPSec
  – Library for PKFey interface
    • Messages between userland and kernel
  – IKE daemon (negociates keys[,,...] with peers)
  – Command line tool to manipulate IPSec stack

• Runs on various OS
  – [Free|Net]BSD, MacOSX
  – Linux 2.6+
  – ????
Your attention please: IPsec and security issues...
IPsec and security...

• IPsec protects traffic on the way. NO GUARANTEE about “what is the traffic”
• IKE's aggressive mode is weak, don't use it !
• PresharedKey's authentication is secure if your shared secret is a secret...
• Implementations may have some bugs
• DES is obsolete
• ESP is vulnerable to bit flipping
  – Use authentication for ESP !
Past....
A long time ago, in a far far galaxy...

- 1983: 4.2 BSD includes a TCP/IP stack :-)
- 1995/12: RFC 1883 for IPv6
- 1998: Launch of the KAME project
  - Main goal: provide an IPv6 stack for BSDs
- 1999: racoon in KAME's CVS
KAME's racoon issues

- Important features missing
  - Functionnal Roadwarrior mode
  - NAT-T
  - ModeCFG / XAuth / Hybrid
- Performance issues
- Security issues
- Quite no more reaction from the team
IPsec-tools's history

- 2003/02/26: Initial CVS revision
  - Goal: temporary fork to add Linux support
- 2004/09/13: Compiles again on NetBSD
- 2005/04/21: KAME drops racoon's support
  - Ipsec-tools is the “official” racoon for everyone
- 2006/09/15: Moved to NetBSD's CVS
  - Technical issues with Sourceforge
  - Some facilities: nightly builds, Coverity, ...
  - But still no SVN ;-)
- 2008/10/19: New homepage, bugtracker, etc...
Present
For the public...

- pkg_install ipsec-tools
  (apt-get install ipsec-tools if needed....)
- Sources hosted at cvs.netbsd.org
  – cvs -d anoncvs.netbsd.org co ipsec-tools
- New homepage/tracker:
  http://trac.ipsec-tools.net
  – Bugtracker
  – Wiki (with doc soon ?)
  – Old page at Sourceforge.net still up but obsolete
- Mailing lists still at lists.sourceforge.net
  – ipsec-tools-core@lists.sourceforge.net also quite obsolete
The actual team

• Some “core developers”!
  – manu@NetBSD.org (NetBSD)
  – vanhu@FreeBSD.org (FreeBSD)
  – mgrooms@shrew.net (FreeBSD, tests on *)
  – timo.teras@iki.fi (Linux 2.6)

• “More or less members”
  – guillaume@free-4ever.net (admin)
  – julien.vanherzeele@netasq.com (NETASQ qualif)

• Many contributors on ipsec-tools-devel
  – Not only developers!

• People are welcome!
Great List of Cool Features

• NAT-Traversal (RFCs 3947 / 3948)
  – Native support for Linux 2.6+ / NetBSD
  – Patch report in progress for FreeBSD
  – Multiple peers behind the same IP ([Net|Free]BSD)
  – PFKey extension not clean
  – NAT-OA support in progress

• Dead Peer Detection (RFC 3706)

• ModeConfig / XAuth / Hybrid (expired drafts)

• Configuration Reload

• Privilege Separation

• Clean roadwarrior support
Not “features” but also cool...

• Lots of bugs fixed through years :-)  
  – Some were security issues
• Improved performances
  – Scheduler (HEAD)
  – “fastquit” (still disabled by default)
  – Logging mechanism
• Some code cleanups
• “Obey” checkmode no more in sample confs
• Autotools mechanism
  – Hey, we're talking about cool things !!!
 cvs diff -u -r past:present
How to use Configuration Reload...

• First, edit your racoon.conf :-)
• For dynamic peers (generate_policy):
  – Just kill -HUP <racoon's PID>
• For peers with static SPD
  – REQIDs have to be the same
  – PH1IDs have to be the same
    ➔ Do NOT use spdflush !
  ➔ Use spddelete to remove obsolete entries, then spdadd to add new ones
Dealing with roadwarriors: past

- On the “client side”, we just don't care...
- A long long time ago:
  It was just not possible on server side
- Some years ago: ModeConfig + ph1_[up|down].sh
- “generate_policy on” works since a few years
- Need to create “anonymous” sainfo entries
  - No control at all about traffic endpoints
Corresponding racoon.conf

Remote anonymous{
    ....
    generate_policy on;
    ....
}

sainfo anonymous {
    ....
}

Various issues...

• SPD entries generated from traffic endpoints
  – Provided by peer, and just accepted
  – Peer can force local traffic to go through his tunnel
  – Peer can generate dummy SPD entries to generate a local DOS

• No link between remote and sainfo section
  – Any “gateway” peer may match anonymous sainfo
  – A roadwarrior may also match another sainfo!
Dealing with roadwarriors now

• `generate_policy [on|unique];`
  – Unique needed to establish more than one phase2 with the same peer

• Ph1id
  – A strong link between a “remote” and a “sainfo” section

• “semi anonymous” sainfos (0.7)
  – You still can't predict peer's IP
  – You know what is your local network!
Remote anonymous{
    ....
    generate_policy unique;
    ph1id 42;
    ....
}

sainfo 192.168.0.0/24 any anonymous {
    # 192.168.0.0/24 is Gate's network
    ph1id 42;
    ....
}
Dealing with roadwarriors soon (HEAD)

- generate_policy [on|unique];
- Ph1id
- “clientaddr” sainfos (HEAD)
  - Mode Config: address given by Mode Config
    - Already in HEAD
  - No Mode Config:
    - Peer's IP (tunnel endpoint)
    - If NAT-T, can also be an IP which match one of peer's hashes
    - Not yet implemented
    - What about “virtual adapters” on client side?
Remote anonymous{
    ....
    generate_policy unique;
    ph1id 42;
    ....
}

sainfo 192.168.0.0/24 any clientaddr {
    # 192.168.0.0/24 is Gate's network
    ph1id 42;
    ....
}
A few words about DPD

- DPD just checks IsakmpSA with peer
  - IsakmpSA can be ok but IPsec SAs are broken
  - IsakmpSA can be broken but IPsec SAs are ok
- DPD just flushes everything related to peer
  - Kernel/Peer will ask for new negotiations “later”
- A bad DPD configuration can be worst than no DPD!
A few words about XAuth

• Please remember that XAuth's security relies on Phase1 security
  – No “group password”
  – No aggressive mode

• Hybrid authentication is your friend
  – Gate's authentication in phase1 with an X509 certificate
  – “Client” will authenticate with Xauth
  – Don't worry, nothing will be allowed after phase1 except XAuth
Future
NAT-T evolutions: PFKey cleanup

- Userland: in progress
- FreeBSD: in progress (in perforce)
- NetBSD: will need to be synced from FreeBSD
- Linux 2.6+
  - may already be ok,
  - won't get worst!
- What about old FreeBSD/NetBSD kernels?
  - Support for “legacy_natt_pfkey”?
  - How to detect it?
NAT-T evolutions: Drafts and RFC...

- Supported versions selection actually done at configure time
  - Some need for a peer by peer setup?
- RFC widely supported now
- Drafts 00/01 don't jump to UDP 4500, and some ugly configured firewalls only accept UDP 500...
- Remove support for drafts 05+?
IPSec and lot of SPD/SA entries (1)

• “Lot of” means something like 1 000++
  – Some of our customers want that (and more)
  – Looks like more people are asking it now!

• IPSec-tools problems: fast negotiations....
  – Will need some optimizations
  – Threaded raccoon? It may be faster to rewrite it!
  – Actually, it can work.... with long lifetimes!
  – Of course, good hardware required!
IPSec and lot of SPD/SA entries (2)

- Main problem: Pfkey interface
  - One PFKey request to dump SPD/SAD
  - One message by answer
  - The buffer of PFKey's socket will fill quickly

- Solutions?
  - Socket_buff_size++ (seems to works on Linux)
  - Kernel thread dedicated to PFKey (userland will have a chance to read while kernel writes)
  - Specific extension “[SPD]DUMP_FROM_X”
    - Kernel must know how to answer “buffer full”
    - Non aware userland tools will just fail.... as now
    - SPD/SAD changes at the same time will be announced by other PFKey messages
IPSec and lot of SPD/SA entries (3)

• Performance issues with huge SPD/SADB
  – Huge list, we have to find one entry...

• Solutions for SAs
  – Put used SAs at the beginning of the list
  – Use an SA cache?

• Solutions for SPD?
  – Common solutions for routing tables won't work
  – Order **is** important
  – FreeBSD6/FAST_IPSEC: spdcache (see graph)
  – We'll have to do “something”
SPDcache (soon in perforce.freebsd.org)

- Basic idea: remind recent SPD evaluations
  - Hash table, key computed from packet profile
  - One cache per entry: just kicks out previous cached value when hash collisions
  - The whole cache is invalidated each time the SPD changes
  - 2 specific results: “no cache” and “cache says No IPsec”
- No need to garbage
- Updates are done only when accessing the cache
SPDcache benches: setup

- 1000 Tunnels => 2000 SPD entries
- From 0 to ~ 400 used (randomly chosen)
- Try to generate highest throughput
- Not the best case for SPDcache
- Benchmarked:
  - FreeBSD4
  - FreeBSD6 (FastIPsec)
  - FreeBSD6 (FastIPsec) + SPDCache
SPDcache benches: results

- FreeBSD6 + SPDCache
- FreeBSD6
- FreeBSD4
Missing features...

- High Availability
- Backup tunnel
- Multithreaded
- IKEv2?
- Automated non-regression test suite
  - Project may start soon...
Up to date slides at
http://people.freebsd.org/~vanhu
http://www.netbsd.org/~vanhu

Questions ?