EuroBSDCon 2007
NETASQ and BSD: a success story

Yvan VANHULLEBUS
vanhu@netasq.com
vanhu@NetBSD.org
September 2007

Overview

- Whois netasq.com && finger vanhu
- NETASQ's history (very quickly)
- NETASQ products technical overview
  - What do we provide ?
  - What are R&D constraints ?
- Building/maintaining an appliance firmware HowTo
- Working with Opensource software / OS
  - Using OpenSource ?
  - Why do we contribute ? What ? How ?
  - Social engineering: working with Opensource community

What's **not** in this talk

- No source code dumps
- No advanced technical things
  - Ok, perhaps one or two.... quickly...
  - Feel free to talk to us after the slides
- No discounts on products !
  - No sales at all :-)
- No trolls
  - xBSD Vs yBSD
  - Emacs Vs other editors (Vs other OS :-)

vanhu@darkstar ~$ whois netasq.com

```
[Verisign terms of use, etc.....]
```

- ~~Firewalls~~ ~~IPS~~ UTM appliances manufacturer
- HQ based near Lille, in France
- Software (“firmware”) developped by R&D
- Hardware design
  - Externalized production
- About 12 M€ in 2006
- Most of the sales done in EU

vanhu@darkstar ~$ finger vanhu

```
Login: vanhu          Name: VANHULLEBUS Yvan
Directory: /home/vanhu  Shell: /bin/bash
```

- NETASQ R&D
  - VPN project manager
  - Perl / Shell guru for NETASQ :-)
  - LDAP, UNIX, kernel, etc....
- IPSec-tools maintainer
  - NetBSD developer
- FreeBSD contributor (IPSec stack, etc...)
- Google: vanhu+CV+feeling lucky...

NETASQ's history
(very quickly)

1998: Netasq is born !

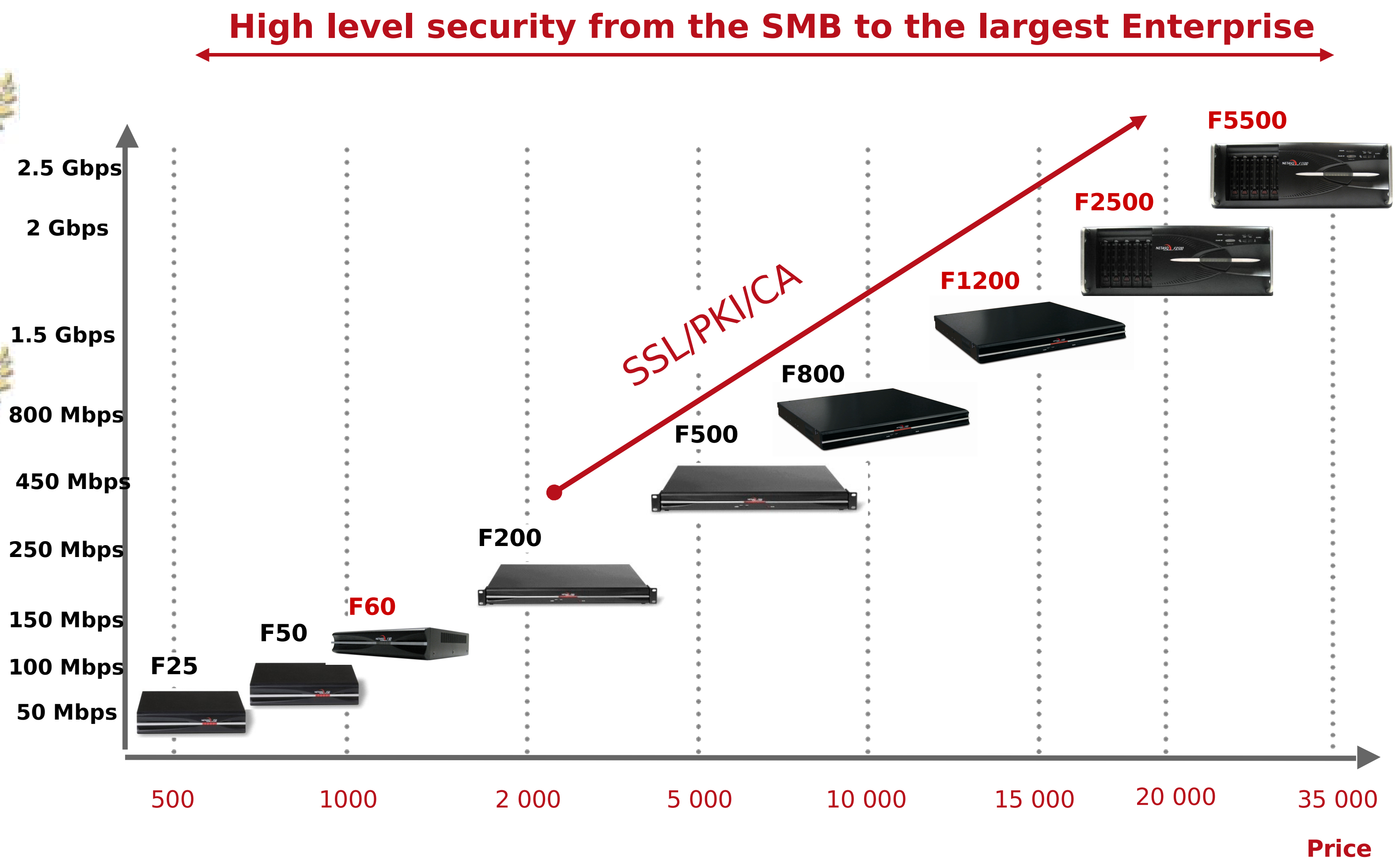
- Employees: ~5
  - R&D: 2
- First product: F10
  - 3x10 Mbs RJ45+BNC
  - 32Mb RAM
  - FreeBSD 2.2.7
  - Firmware V1.0
  - Stateless packet filtering (using ipf)
  - GUI

NETASQ today (~ 10 years later)

- Still alive !
- Employees: ~ 60
  - R&D: ~ 20
- Appliances for all (SOHO -> huge networks)
  - Same security level for all
- Firmware v7.0
  - FreeBSD 4.11
    - FreeBSD 6.x for the next major release in 2008
  - Lots of features
- ~ 15 000 units sold in 2006
  - Available in +30 countries

NETASQ products today

Appliances for all



F25 hardware

- ~ 500Mhz CPU
- 2 x 100Mbs ethernet
- 128Mb RAM
  - 64 Mb two years ago
- 128Mb flash
  - Also 64 Mb two years ago
  - Not so much disk space....
  - Quite slow
  - Must limit write access to flash !

F5500 hardware

- 2 x ~3.5Ghz CPU
- 1Gb RAM
- 140Gb RAID1 SCSI (hotplug)
- Up to 24 Gigabit Ethernet interfaces
- ~ 35Kg :-)

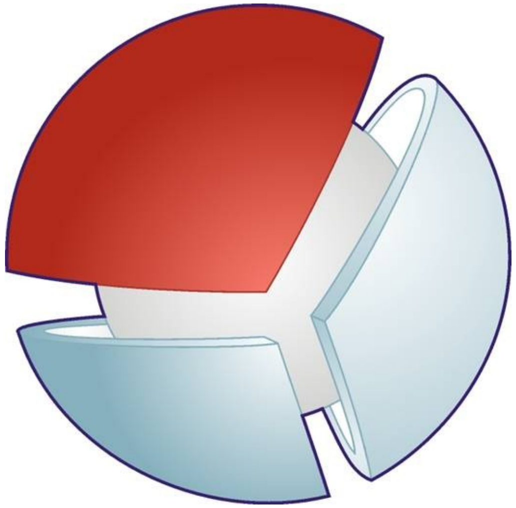
NETASQ Offers *Unified Security*

NETWORK SECURITY
IPS . FIREWALL . ANTIVIRUS . ANTISPAM . ANTISPYWARE . QoS . CONTENT FILTERING

Risk Management

Antivirus

IPsec VPN



Antiphishin g

SSL VPN

Antispyware

Web Filtering

Antispam

Application Firewall

ASQ for Real-time
Intrusion Prevention

What's provided....

- Firewall+Intrusion Prevention System (ASQ)
  - Advanced plugins for some protocols
  - HTTP filtering
- NAT
- VPN (IPSec, SSL)
- LDAP (including server if needed)
  - PKI for F200+
- Antivirus / Antispam
- SNMP, NTP, DNS, DHCP features
- [some other internal stuff]
- Probably some other things I forgot....

What's running behind...

- “NSBSD” (NETASQ Secure BSD)
  - FreeBSD
  - ~ 700k of kernel patches+some userland patches
- ~ 10Mb of NETASQ sources (binaries + libs)
  - Mostly in C
- ~30 “contribs” + ~ 650K patches
  - Ipsec-tools
  - OpenLDAP
  - Isc-dhcp
  - Ntpd
  - P7zip
  - ....

Almost everything is provided on all products, from F25 to F5500....

Some #ifdef NETASQ_MODEL in the sources, mainly for memory usage

R&D constraints....

- Security (of course.....)
  - Security of the code itself
  - Secure our customer's networks !
- Memory constraints, for F25 / F50
- MB/s for all products
- Write to / as less as possible
- Compatibility with older products

  (No, we don't support F10 anymore :-)
- New features !

Okay, let's log on it:

```
ssh admin@firewall
```

Ssh admin@firewall : WARNING !!!!!

```
F100XD01217020070 1> echo $SHELL
/bin/csh
F100XD01217020070 1> echo $EDITOR
joe
F100XD01217020070 1> ls -l vi # On some older versions
vi -> joe
F100XD01217020070 1> emacs
emacs: Command not found.
```

- Forget ifconfig/route/etc... : they do NOT configure ASQ !!!
- Lots of bins/libs are NOT here

NETASQ

Ssh admin@firewall

```
F100XD0121702007O1> mount
/dev/ad0s1a on / (ufs, local, noatime, synchronous)
[ad0 or md0] on /tmp
/dev/md0c on /var (ufs, local)
/dev/ad0s1f on /log (ufs, local, noatime)
```

- / is synchronous
  - Not so much writes
  - More reliable than softupdates
- /var is on RAM
  - Lots of generated /etc/* are links to /var/tmp/*
- Logs for F200+

Ssh admin@firewall

```
F100XD012170200701> ls -l | grep -i interesting
```

- /COPYRIGHT :-)
- /kernel[.gz]
- /usr/Firewall
  - User's configuration / datas
  - NETASQ binaries/libraries
  - Firewall specific informations
- /var
  - Generated configurations for contribs
  - Generated hosts, networks, etc...

Ssh admin@firewall

```
F100XD01217020070l> du -h Update/update.tgz
9,2M    Update/update.tgz
F100XD01217020070l> df -m
Filesystem   1M-blocks  Used Avail Capacity  Mounted on
/dev/ad0s1a        256    38   218     15%    /
```

- Used size can be much more
  - URL groups: ~ 35Mb
  - AntiVirus: up to ~ 15Mb
  - Appliance's configuration: up to 5-10Mb ?
- Size of / is model dependant
  - 1 Gb for High end products
  - 128 Mb for low end products
- Size of update.tgz is almost the same for all

A few words about upgrading...

- We can't just dd of=/dev/ad0s1a
  - Configuration and Firewall specific informations on the partition
- It's safer to upgrade files during a boot
- Customers often unplug the device during the update if it takes “too much time”
  - We need to do it as fast as possible
- BOOT.tgz / NSBSD.tgz / FW.tgz
  - Each one is written to disk / extracted only if needed
  - BOOT.tgz extracted before the reboot
  - Others extracted by custom /sbin/init

Building and maintaining a firmware HowTo

What do we need ?

- A good editor (Out of topic: no time for trolls)
- A repository for our work
- A Programming-Howto (Out of topic)
- An easy way to manage contributions
- An unified way to build our sources
  - Binaries
  - Libraries
  - Default configuration, scripts, etc...
- An easy way to manage a patched kernel
- Get a minimal FreeBSD system (userland)
- One command to rule them all.........

A repository for our work

- CVS used for some years
  - cvs annotate was useful :-)
  - Commit per file can be problematic
  - checkout.sh needed to get the complete sources
- SVN used now
  - r1: 2006-02-22 21:42:55 +0100
  - One commit by feature (or by fix :-)
  - Easy to import CVS tree
  - Easy to use for CVS users
  - “Externals” obsoleted checkout.sh
  - svn blame is great :-)

NETASQ

An easy way to manage contributions

- Fetch / build / clean contribs
  - We can “install” what we need by simple cp
- Updating contribs must be easy
- Some contribs are patched
  - Patches must be stored “somewhere”
  - Patches must be used by build process
- Having our copy of patches sources is NOT a good solution !
  - Updating the contrib won't be easy !
- FreeBSD's ports system is perfect for such stuff !

An unified way to build our sources

- Binaries / libraries
  - Build, with shared options and specific options
  - Install in a specific location
  - Compile again only when needed
- That's Makefile's job !
  - Lots of make commands
  - Lots of syntaxs, Makefile styles, etc....
  - bsd.*.mk are good: our Makefiles are small
- Cmake is needed for crossplatform works

NETASQ

An easy way to manage patched kernel

- Build / clean kernel
  - We'll have to handle various kernel config files
- Updating kernel sources must be easy
- Kernel is patched
  - Patches must be stored “somewhere”
  - Patches must be used by build process
- Having our copy of kernel sources is NOT a good solution !
  - Updating kernel sources will really not be easy !
- FreeBSD's ports system is perfect for such stuff !

Get a minimal FreeBSD system

- Only a few userland files are patched
  - We can apply those patches on the build host
- We know the list of needed binaries
  - We can get them from the build host
- We have to check needed libraries
  - Dangerous to forget a new important lib !
  - Interesting to remove everything except what's really needed
- We generated a FreeBSD.tgz for those files
  - File name + MD5 known for each tagged revision
  - FreeBSD.tgz files are archived for years
  - ~ 3.8Mb for FreeBSD.tgz actually

One command to rule them all....

- [make clean &&] ./build.sh
- Generates all shell env (CFLAGS, ...)
- Knows firmware revision, model, etc...
- Extracts/builds everything
  - Knows what needs to be done first
- Checks FreeBSD.tgz MD5 sum
- Calls all clean/check/etc... scripts
- Generates a tarball for the whole firmware
  - Generates dynamic informations used by update process

Working with OpenSource: Using OpenSource code ?

Using Opensource for security project ?

- Lots of reports confirms “Opensource is secure”
- Lots of reports confirms “Closed sources projects are more secure”
- Some Opensource programs have an heavy vuln. History.....
- Some closed sources programs too.....
- It's always easy to get some numbers who will tell “what you want”
- So ?????????????

An OpenSource program...

```
/* This program is licensed under
 * any random OSI approved license...
 */

#include <string.h>

int main(int argc, char **argv){
   char buffer[256];
   strcpy(buffer, argv[1]);
   return 0;
}
```

A closed sources program...

```
/* Hey ! The EULA of this program does
 * NOT allows you to read this source !!!
 */

#include <string.h>

int main(int argc, char **argv){
  char buffer[256];
  strcpy(buffer, argv[1]);
  return 0;
}
```

Conclusion about security:

- Closing sources for security is
  - Quicker
  - Easier
  - More seductive
  - But not more ~~powerful~~ secure
- Security of the code does NOT depend on licence
- Security of the code just depends on developers

Using an Opensource project ?

- Does it provide the needed features ?
  - Or would it be easy to add them ?
- What is the project's licence ?
  - We sometime cannot use “GPL style” licensed programs
- Is the code stable enough ?
  - And is it secure enough ?
- How much would it cost us to rewrite it from scratch ?
- How much would it cost to use a 3rd party program ?

man 7 “how much”

- “How much”
  - Time
  - Money
- How much will it cost to have the functionnality we need
- How much will it cost to maintain it
- How much will it cost to extend it
- 3rd parties: royalties ?

Working with OpenSource: Using FreeBSD...

Why did we make the good choice

- Of course: the BSD licence !
- Of course: robust and efficient network stack
  - Polling works for years
  - Netgraph / MPD
- At the beginning: IPF
  - We replaced it easily by ASQ when it was ready to also do the filtering job
- Lots of ports
- FreeBSD is also usable as a workstation....
- When one BSD is supported by 3rd parties, it's FreeBSD !

Drawbacks.....

- People knows “Linux”, not ???BSD
  - Our shareholders, CEO, etc... are “people”....
- Drivers are often not available
  - Hardware RAID
  - Actually, the soundcard of my workstation :-)
- Some 3rd parties only support Linux

Drawbacks: From FreeBSD 4.x to 6.x

- FreeBSD.tgz: +25%
- Network performances: ~ -25% on first tests
  - Polling
- We also had to migrate ASQ from spl*() to mutexes
- Some kernel crashes in first versions
  - FreeBSD bugs (fixed by FreeBSD most of the time)
  - Some NETASQ patches who needed some changes
- Some savecore / kgdb problems
- Problem with GEOM (remounting / ro)
  - “Geom is in the kernel”...

Don't worry: FreeBSD 6.x is great !

- Maintained version :-)
  - Security fixes
  - Hardware support
  - Ports
- Better SMP support
  - Will be even better in FreeBSD 7.x
- Lots of other features... which are sometime useless for such an appliance...

Race condition: supported versions

- Normal support for FreeBSD versions: 1 year
  - Told yesterday by security officier :-)
- We release a major version ~ each year
  - Some few time to upgrade from FreeBSD X.Y to X.Z
- Upgrading from X.Y to Z.T takes lot of time
  - Will have taken more than 2 years for us to migrate from 4.11 to 6.2
  - Okay, we had other stuff to do during that time
- We will release our major version based on FreeBSD 6.x (3 ?) when it will come to EOL
- No real solution actually...

Working with OpenSource: (some) past and future contributions

Why do we contribute ?

- Because it's fair
  - Our shareholders don't care about that....
- Because we won't have to update our patches when we'll upgrade
- Because we can have some feedback from the community
  - Bugs reports, improved versions, etc...
- To become a member of the community

Various kinds of contributions

- Maintaining projects
- PRs, patches, features, etc...
- Being there at each EuroBSDCon ! :-)
- Documentation ?
- Benchs and feedbacks !
- Talk about BSDs

Not contributing some things ?

- Because it's internal stuff
  - NETASQ's logging system
  - NETASQ's configuration system
  - ASQ !
- Because it's an ugly hack
  - We avoid such really ugly hacks as much as possible, of course !
  - Sometimes, a patch is perfect for our specific usage, but not for general purpose
- Because we don't have time !

NETASQ and ipsec-tools

- Racoon used in NETASQ appliances
  - racoon20011215a really lacks features, stability, ...
- We needed to do lots of patches
  - Internal patches (logs, etc...)
  - Features, fixes, etc...: mostly reported to KAME
- Ipsec-tools fork was far more reactive
  - DPD and other patches reported quickly
  - Manu@NetBSD.org was already in the place :-)
  - Commit bit since late 2004
  - Hosted at NetBSD since ~ 1 year

Past contributions to ipsec-tools

- Bugfixing / cleanups / optimizations !
  - We have **lots** of customers running racoon...
  - We do some non-regression tests for each version
- DPD (RFC 3706) support
- Configuration reload
  - No needs to kill racoon, Only flush what is obsolete
- Functional road-warrior mode
- Some works on NAT-T
- Finds netipsec/ipsec.h for FreeBSD 7 ;-)
- Contributor's patches audit/report
- Release engineering, support, etc...

Ipsec-tools: being a member of the team

- Gets security reports before everyone !
  - And also get the patches beore everyone !
  - More easy to synchronize both ipsec-tools and NETASQ releases
  - Of course, we also need to synchronize with others
- More easy to report my work for NETASQ
  - Don't have to maintain lots of patches in lots of repositories copies
- Needs some time to do that work !
- Direct contributions to NetBSD ?
  - According to Manu@NetBSD.org, I already contribute to NetBSD :-)

Some past contributions: FreeBSD

- FreeBSD's port of ipsec-tools
- Some patches to IPSec stack
  - OpenBSD's enc0 port (not completely reported)
  - IPIP decapsulation problem in FAST_IPSEC stack
  - Fixes, etc... reported to KAME or FreeBSD
- Some other kernel bug reports
  - VIA padlock: i386/114331 (+patch)
  - Broadcast forward problems: kern/103950 (+patch)
  - Em driver hardlock: kern/66634
- A few other reports/patches for userland and ports

Expected future contributions

- More works on IPSec, of course :-)
  - NAT-T (wait a few more slides for details...)
  - High number of SPD/SA entries (next slide)
  - Feedbacks on FAST_IPSEC
- Feedbacks on network performances
  - We have the needed hardware for benchs !
  - Pollng (also in a few slides)
- Some pr for ports ?
- Other things ? I hope so !
  - Perhaps some old patches we forgot :-)

NETASQ

IPSec and lot of SPD/SA entries (1)

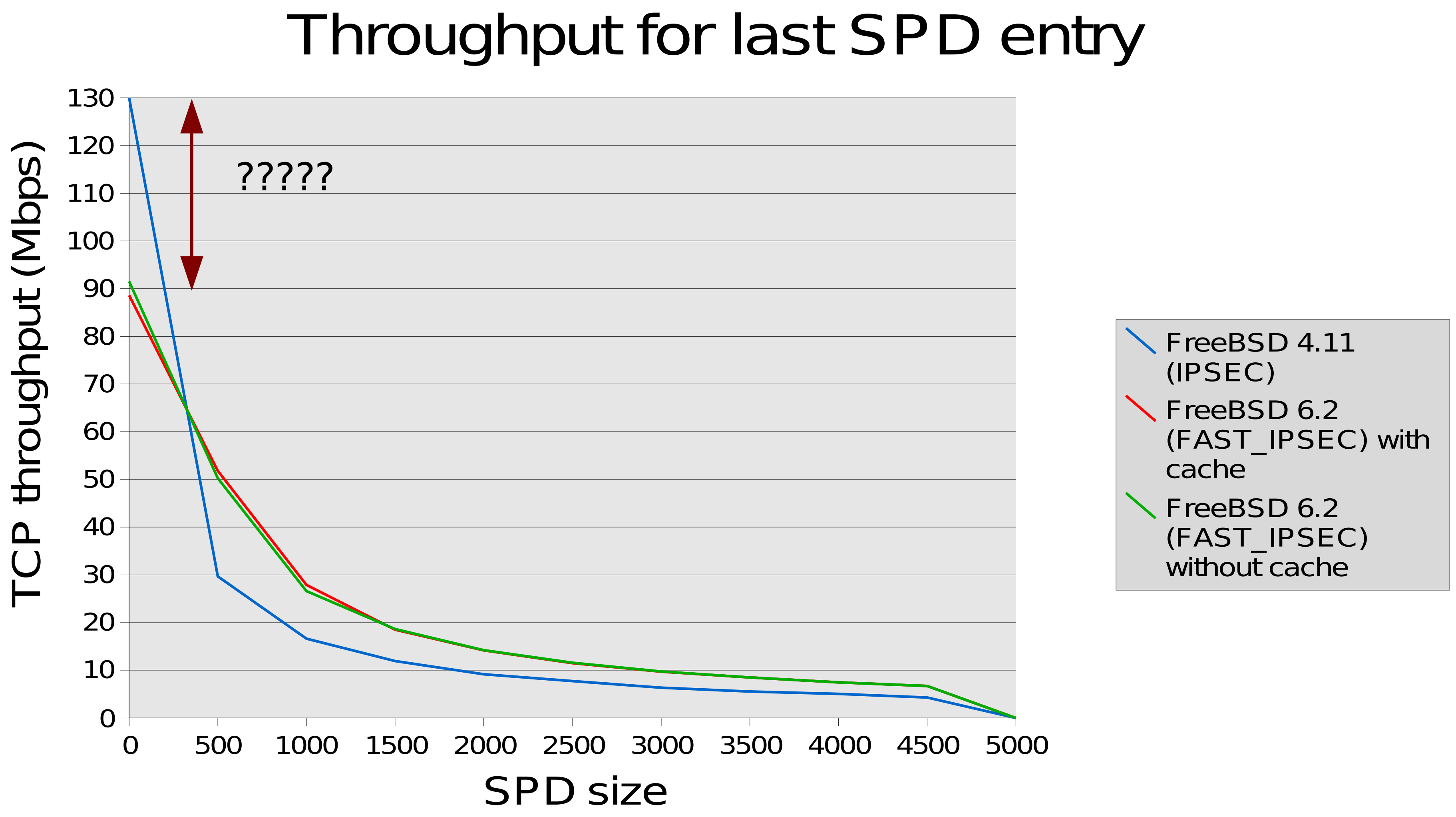
- “Lot of” means something like 1 000++
  - Some of our customers want that (and much more)
- First problem: Pfkey interface
  - One PFKey request to dump SPD/SAD
  - One message by answer
  - The buffer of PFKey's socket will fill quickly
  - Also old problems with sbspace() macro (fixed in 6.x ?)
  - We fixed that using a single buffer for all answers
- IPSec-tools problems: fast negociations....
  - Will need some optimizations
  - Threaded racoon ? It may be faster to rewrite it !
  - Actually, it can work.... with long lifetimes !

NETASQ

IPSec and lot of SPD/SA entries (2)

- 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”

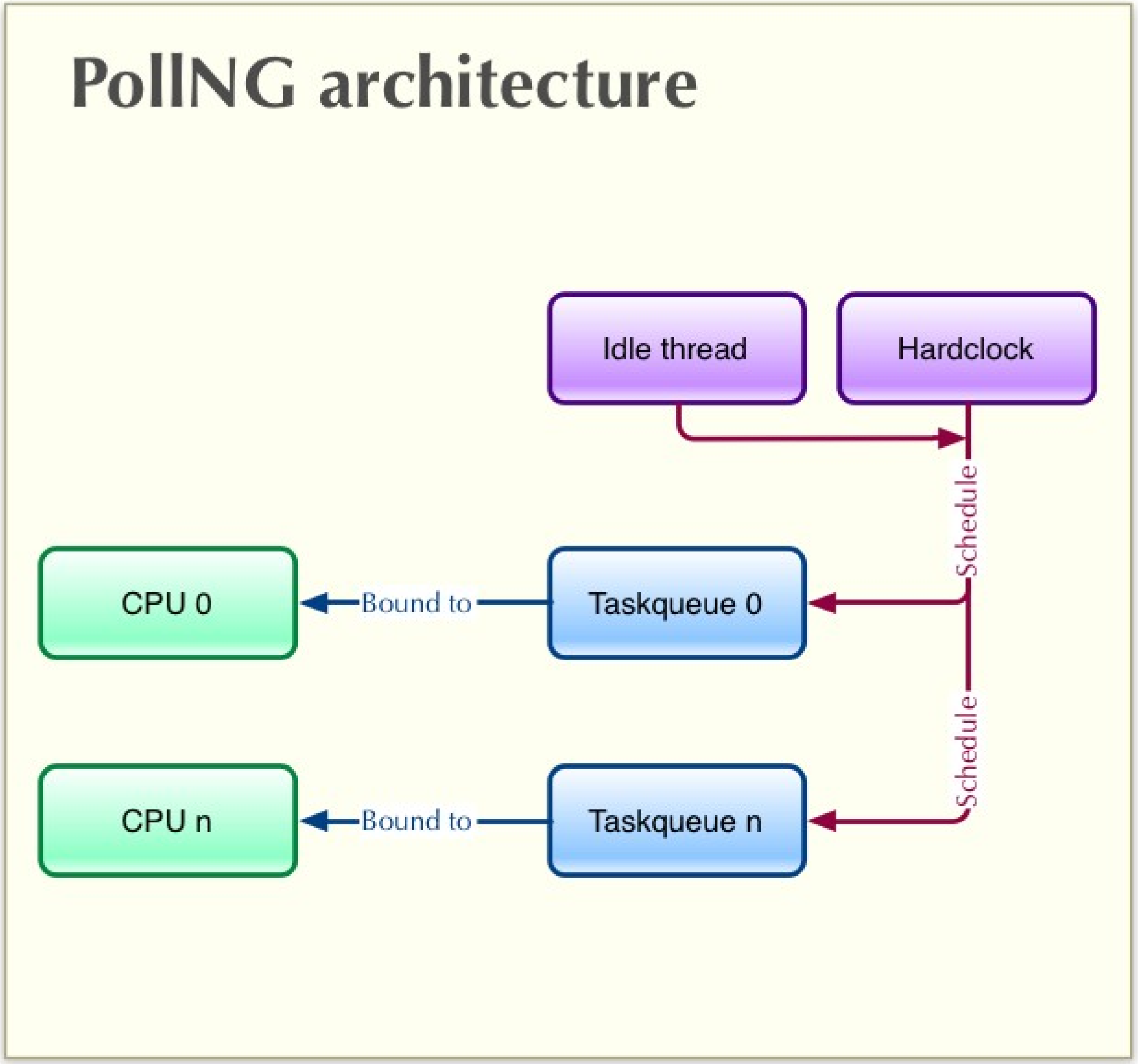
Benchmark with huge SPD

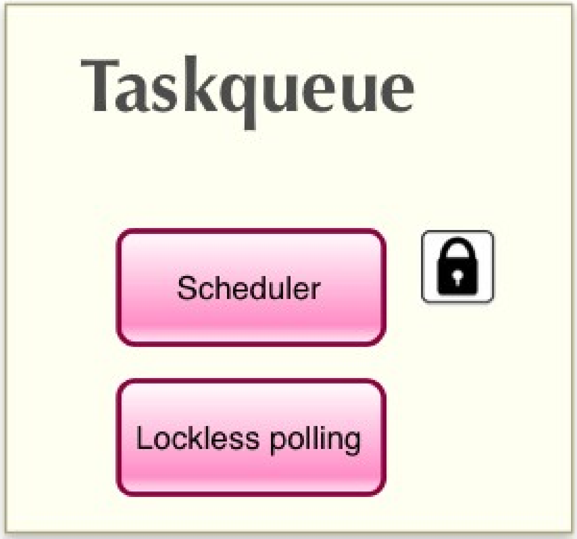


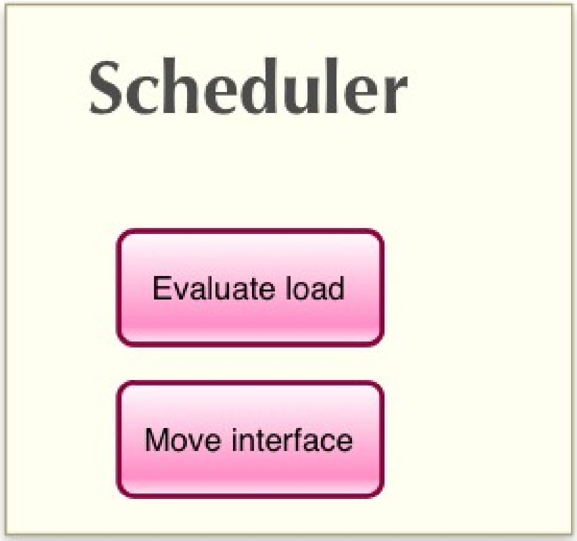
In progress: Pollng

- Performances loss on FreeBSD6 with polling
- Mail sent last week on freebsd-net@
  - http://www.netasq.com/opensource/pollng-rev1-freebsd.tgz
- Each NIC is attached to a specific CPU
  - More scalable
  - Attaching all NICs to a single CPU can also be interesting
- Each polling queue is processed by a specific thread
  - More easy to monitor
- Also interesting with a single CPU
- Differences between FreeBSD 6.2 and 7

Pollng


PollNG architecture
Idle thread
Hardclock
Schedule
Schedule
CPU 0
Bound to
Taskqueue 0
CPU n
Bound to
Taskqueue n


Taskqueue
Scheduler
Lockless polling


Scheduler
Evaluate load
Move interface

Still pending: NAT-T kernel support...

- IPSec NAT-T encapsulation: RFC 3948
  - RFC3947 support already in ipsec-tools
- First patch submitted years ago for 4.11
  - Patent problems
  - Looking at the patch again, it was quite ugly :-)
- Patch maintained, and provided for 6.x/HEAD
  - I know, Bz's patch have been reported late
  - HEAD patch updated (FAST_IPSEC->IPSEC), but still not validated
- May be reported in FreeBSD 8 ???
- Everyone else have native NAT-T support for years, even if it's not always perfect !

Working with OpenSource: Some social engineering....

Contributor's mind when it takes time ?

- “What are they doing ?”
- “Do they really want us to contribute ?”
- “Will I really continue maintaining patches ?”
- “What are they doing ??????”
- “I'll send them a new mail for that..... tomorrow”
- Of course, it's not so simple....

Commiter's mind when getting requests ?

- It's always more interesting to spend time on owns work
- “What's that patch ????”
- “Will it break anything ?”
- “Why doesn't he answers my questions ?”
- “I'll have a look at it tomorrow”......
- Of course, that's not so simple.....

Contributor's constraints

- Available time
- Motivation
- Skills
- Sometime, contributor's memory

Commiter's constraints

- Available time
- Motivation
- Skills ?
- Sometime, commiter's memory

Extra context for employees

- We can spend work time on some things
- We can spend no work time on other things
- “Things” can move from one to the other categories

So ?

So ?

- The real problem is quite always related to free time
  - Rumors says some people also have a social life !!!
- We can't patch days to have 48 hours
- Sometime, we just have delayed ACKs
- We must find solutions to improve communications, and faster things
  - Keep an easy to use track of problems (PRs ?)
  - Find an easy way to tell “I don't have time”
  - Grow up community ?
  - Commit but disable by default ?
- Avoid moving to a “Linux style” development

Conclusions

- It's possible to make business with BSDs !
- It's possible to make a security device from a BSD
  - Hopefully !
- It's possible to do business AND contribute
- Some things can be improved, on all sides
  - We will all take benefits of such improvements

Questions