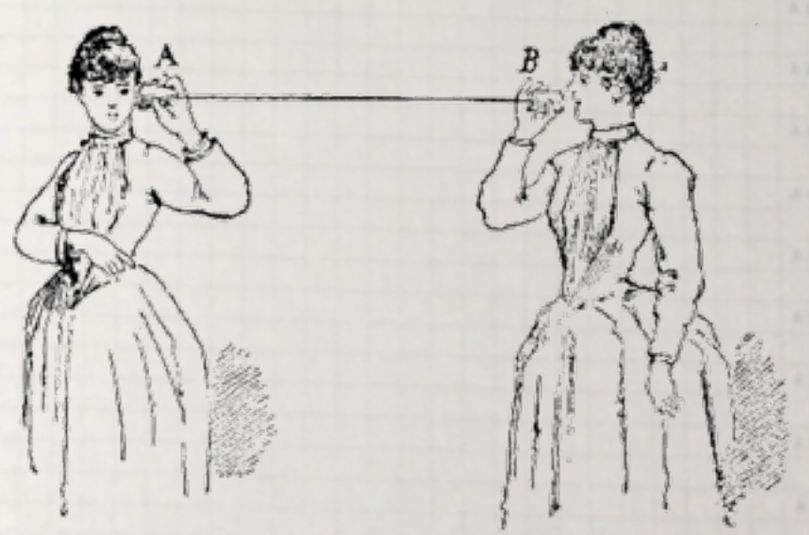
OpenIKED, AsiaBSDCon 2013

Reyk Flöter (reyk@openbsd.org)
ライクフローター

Agenda

1. Background & History
2. Design & Implementation
3. Portable "OpenIKED"
4. The GUI



Why another VPN protocol?

- We have lots of existing VPN protocols:
  - IPsec IKEv1 with isakmpd(8)
    - L2TP, PPTP and more with npppd(8)
    - OpenSSH (SSH-VPN with tun(4))
    - OpenVPN in ports...
- And many vendor-specific SSL-VPNs
  - Microsoft's SSTP: PPP over HTTPS
  - Cisco AnyConnect, Juniper, Citrix, ...

Why another VPN protocol?

- Different VPN types for different use cases
  - SSL-VPN: lots of overhead but passes web proxies; different protocols
  - IPsec: does a better job on IP but IKEv1 is less flexible with NAT and mobility
  - OpenVPN: for religious people
  - BGP MPLS VPN: large virtual networks but without privacy (it should be "VN")
- We need a standardized, widely adopted, secure, flexible and low-overhead protocol

IKEv1 and ISAKMP/Oakley

- IKE? ISAKMP? Oakley? DOI?
  - Internet Key Exchange; RFC 2409
  - on top of ISAKMP/Oakley; RFC 2408
  - on top of the Internet DOI; RFC 2407

  + many additional RFCs
- Widely adopted and (mostly) interoperable
  - Cisco, racoon, strongswan, Windows, ...
- Long history with strong security research
  - Known weaknesses, do's and dont's

isakmpd(8)

- Written 1998 by Niklas Hallqvist and Niels Provos for Ericsson
- Supports the full ISAKMP and DOI layering
  - but IKE is the only protocol on top of it
- Uses an .ini-style configuration (yay '98)
  - And the KeyNote policy language
- Does not support some of the extensions
  - No XAUTH (user/password), No IKECFG
- Doesn't work very well with road warriors

ipsec.conf(5) and ipsecctl(8)

- Workaround isakmpd to make it useable
  - The daemon is ok, but the usability...
- ipsec.conf is a nice config grammar that will be loaded into isakmpd.fifo by ipsecctl
  - Benefit: you don't need to touch the .ini and the KeyNote policy anymore
  - Problems: Two steps to run isakmpd

```
# isakmpd -K &&
  ipsecctl -f ipsec.conf
```

    - Doesn't do reloads - kill & restart

Internet Key Exchange version 2 (IKEv2)

- They learned a lesson and simplified IKE
  - No ISAKMP+DOI layers anymore
  - The IKEv2 payload is now like ESP
- One 4-way handshake, optional cookies
- Improved network robustness and mobility
  - Even PSK works with road warriors now
- New concept of traffic selectors (flows)
  - IKEv1 embedded the flows in the ID
- RFC 4306 by Microsoft, updated by RFC 5996

Internet Key Exchange version 2 (IKEv2)

- A quick reference:
    - Traffic Selectors: One or more flows per IKEv2 SA (from x.x.x.x to y.y.y.y)
    - IKESA: formerly known as Phase 1
    - CHILDSA: Phase 2 for IPsec ESP/AH
    - Initiator: the client
    - Responder: the server
    - PRF: pseudo-random function
    - EAP: Extensible Authentication Protocol

iked(8)

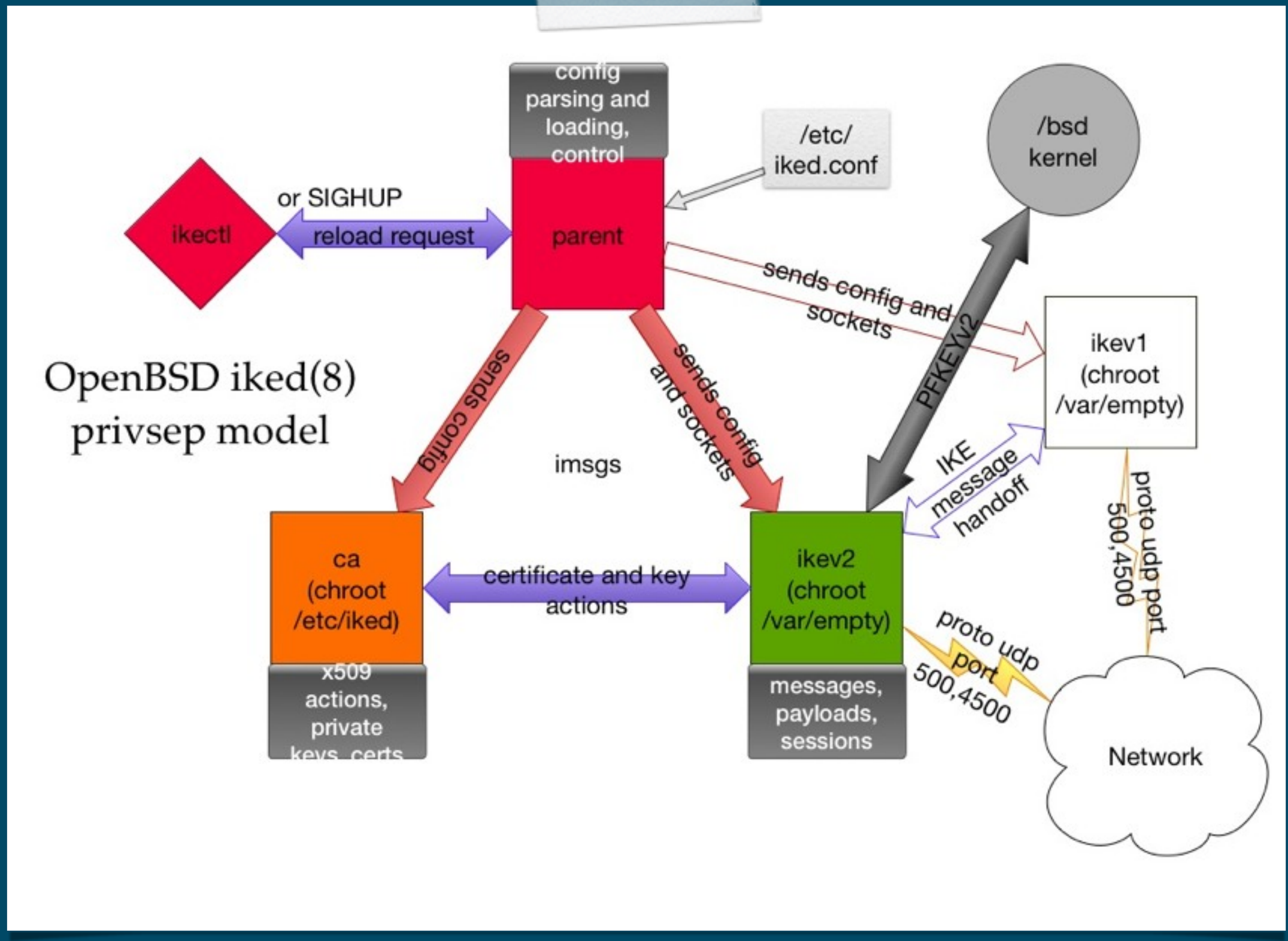
- A new implementation of IKEv2; RFC 5996
- Based on the following design decisions:
  - A privsep'd daemon for OpenBSD
  - An integrated ipsec.conf-style config
  - Stateful config reloads
  - Control with ikectl(8) **not** isakmpd.fifo
  - Scalable with gw2gw and roadwarriors
  - Provide better X.509 CA useability
  - Use OpenSSL instead of custom crypto

ikectl(8)

- Control, reload and monitor iked
- „ikectl ca“ to manage a simple X.509 CA
    - Simple configuration of certificates

```
# ikectl ca test create
# ikectl ca test install
# ikectl ca test cert 10.1.1.1 create
# ikectl ca test cert 10.1.1.1 install
# ikectl ca test cert 10.1.1.2 create
# ikectl ca test cert 10.1.1.2 export
```

Design & implementation of iked(8)



Design & implementation of iked(8)

ikectl

iked parent

Network

ikev1

ca

kernel

ikev2

All kinds of strong crypto

- Modern ciphers for IKESA and CHILDSAs
  - eg. Auth & PRF with SHA2
  - More AES modes (CBC, CTR, GCM)
  - More Diffie-Hellman modes
    - 26 groups, up to modp8192, ecp521
    - Elliptic curve groups are fast and secure
- Supports authenticated encryption
  - AES-GCM support added by Mike Belopuhov (mikeb@openbsd.org)

Interoperable VPN with iked(8)

- Interoperability, so far with:
  - Windows 7/8: really easy to set up!
  - Linux Strongswan: *narf*
- Not tested:
  - Cisco IOS & AnyConnect 3
  - Not-so-OpenSolaris
  - BlackBerry 10 (any BB Z10 donation?)

MOBIKE and other future work

- Finish the basic IKEv2 support
  - Cleanup, fixes, serious reviews
- Additional authentication methods
  - RSA public key authentication
  - ECDSA support
- MOBIKE improves mobility support by allowing peers to reuse SAs with a changed IP address
  - We want to support RFC 4555 / 4621
- Add RADIUS support for other EAP types.

Configuration examples

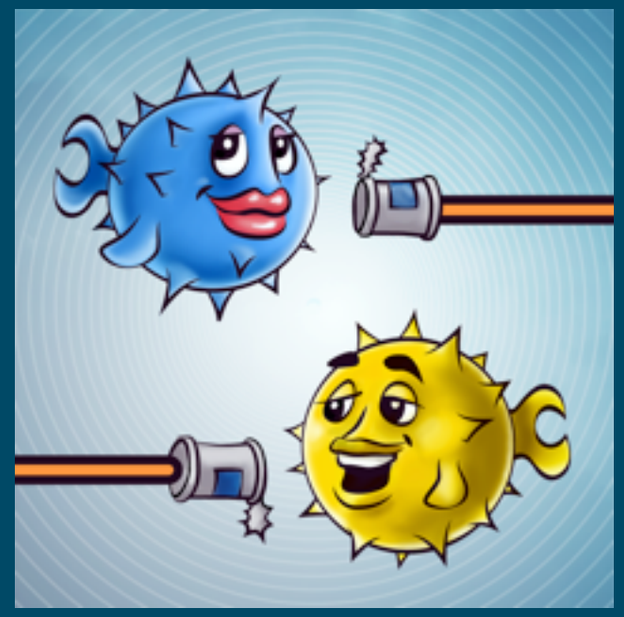
```
# ikectl ca …
# mg /etc/iked.conf && iked [-dvv]
```

```
# Simple gateway to gateway configuration (remote iked)
ikev2 esp from 10.0.5.0/30 to 10.0.5.4/30 peer 192.168.1.2

# A bit more complicated: Accepting a Windows client
user "user1" "password123"
ikev2 "win7" passive esp \
        from 10.1.0.0/24 to 10.2.0.0/24 \
        local any peer any \
        eap "mschap-v2" \
        config address 10.2.0.1 \
        config name-server 10.1.0.2 \
        tag "$name-$id"
```

The portable *Open*IKED project

- In need for a simple IKEv2 client,
  I ported iked(8) to OS X
- This was the first step to create
  a „portable“ version of iked(8)
- The „OpenIKED“ subproject was born
  - The „tin can“ artwork was contributed by
    Markus Hall from Sweden
  - The webpage is http://www.openiked.org/
  - Partially hosted at GitHub,
    outside of OpenBSD



*Open*IKED project goals

- Lean: Provide a small and monolithic architecture that supports the main standards and most important features of IKEv2.
- Clean: Write readable and clean code following strict coding style(9) guidelines.
- Secure: Implement secure code with strict validity checking, bounded buffer operations, and privilege separation to mitigate the security risks of possible bugs. Use strong cryptography with sane but secure defaults.
- Interoperable: Provide good interoperability with other IKEv2 implementations, support non-standard extensions if it is required to interoperate with other major implementations.
- Configurable: Make the configuration easy and nice with sane defaults, minimalistic configuration files and good documentation in the manual pages. Avoid the headaches of past and other IKE implementations.

OpenBSD‘s portability approach

- Based on OpenSSH‘s development process
  - ➡ Damien Miller, Secure Portability, http://www.openbsd.org/papers/portability.pdf, October 2005.
- The „core“ development happens in OpenBSD
  - The code is hosted in OpenBSD‘s CVS
  - No compatibility glue in OpenBSD‘s version
  - Improves maintainability and security
- The „portable“ version adds compatibility glue
  - Hosted outside of OpenBSD‘s CVS
    - OpenIKED is currently on GitHub
  - Includes `#ifdef`‘s and `openbsd-compat`

Portable requirements of *Open*IKED

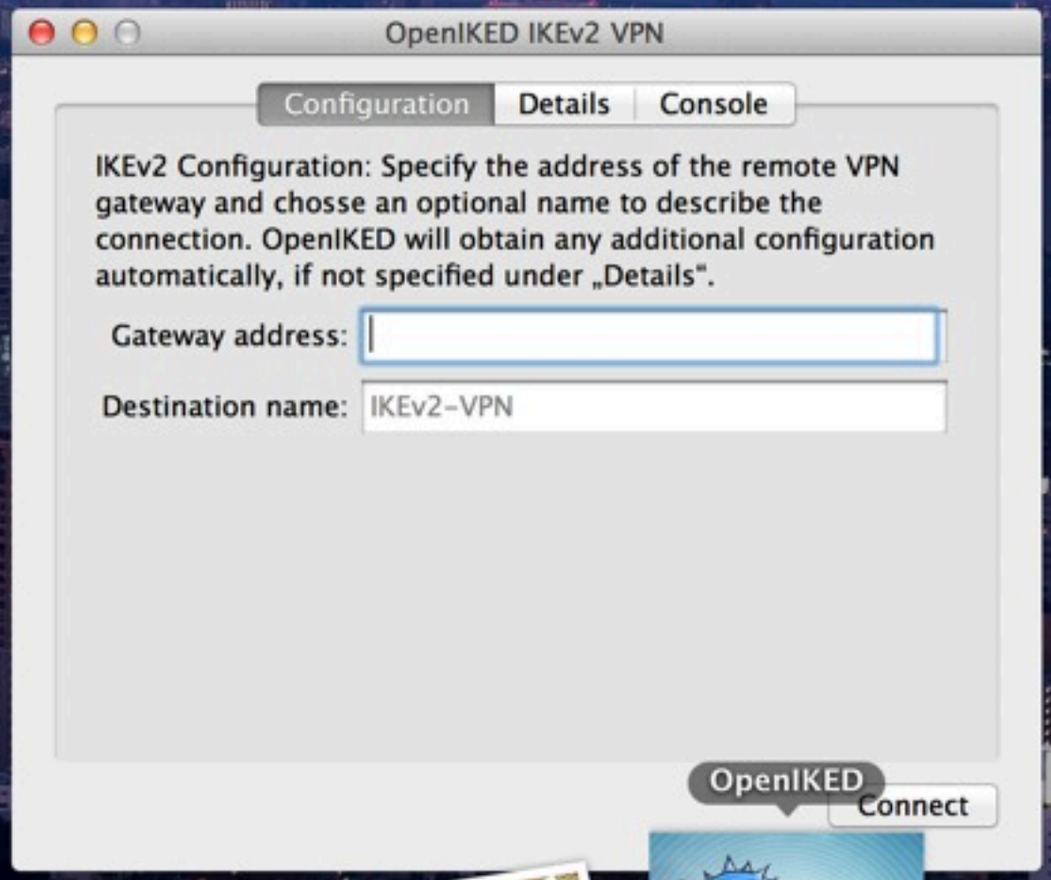
- An UNIX/BSD/POSIX-like operating system
- Strong crypto
- Libraries:
  - OpenSSL 1.0 or newer
  - libevent 1 (version 2 is not tested)
- Kernel APIs:
  - IPv6 support
  - IPsec KAME-compatible stack with PFKEYv2
    - Or OpenBSD‘s PFKEYv2 variant

Supported Operating Systems

- The „core“ version: OpenBSD
- The initial port: Apple OS X / Darwin (10.8)
- GNU/Linux (Ubuntu 12.10 Linux 3.5.0)
    - The PFKEYv2 interface is deprecated but still available.
- FreeBSD 9.0, NetBSD 6.0
    - As of 2013, IPsec is still disabled in GENERIC and you have to compile a custom kernel
- I gave up on DragonFlyBSD … anyone?

The GUI

- I started writing a little GUI for Apple OS X
  - Goal: provide something as simple as Windows‘ recent VPN wizard
  - Optimal configuration:
    1. Install certificates
    2. Enter gateway
    3. Go!



Conclusions

- (non-HTTPS) VPN is still important
  - Even more in the „age of cloud computing“
- IPsec is not dead
  - But IKEv1/ISAKMP will certainly disappear
- Mobility demands IKEv2‘s features
  - See Windows, BlackBerry, … more to follow
- OpenIKED is a fairly new implementation
  - Not the only one … but FREE, lean, clean, secure, interoperable and configurable (^_^)

Danke!

...thanks for supporting the OpenBSD project!