The IPSec Architecture in OpenBSD

Niels Provos

provos@citi.umich.edu

CITI - Center for Information Technology Integration

University of Michigan

Overview

1. Introduction
2. History
3. Kernel Details
4. IKE implementation
5. Policy using Keynote
6. Hardware Crypto Acceleration
7. VLAN with Bridging and IPSec
8. Conclusion

Introduction

- OpenBSD is a free UNIX-like operating system with a strong emphasis on security.
- We realized early that many security problems can be solved only by using cryptography.
- Cryptography is integrated seamlessly into the system and is an essential component:
  - IPSec, IPv6, key management daemons,
  - Cryptographic Hardware acceleration
  - Kerberos, AFS, OpenSSH, etc...

History

- OpenBSD 2.1 released with IPSec support in July 1997:
  - Based on BSD/OS code by John Ioannidis
  - Ported to OpenBSD by Angelos Keromytis
  - no automatic keying, only manual setup
- OpenBSD 2.2 released in December 1997:
  - Photuris key management daemon
  - new ESP and AH support according to RFC 2406 + 2402

History

- OpenBSD 2.3 released in May 1998:
  - more encryption and authentication transforms, e.g. Blowfish,
  - `PF_ENCAP` notifications for key management daemon.
- OpenBSD 2.5 released in May 1999:
  - `isakmpd` as IKE daemon,
  - `PFKEYv2` as generic key management API.

History

- OpenBSD 2.6 released in December 1999:
  - support kernel acquire messages.
- OpenBSD 2.7 released in June 2000:
  - support IPSec tunnels for bridging,
  - hardware acceleration for IPSec,
  - ACLs for ingress flows,
  - IPv6 IPSec support.

Kernel Details

- incoming IPSec packets are decapsulated and processed in `ipsec_common_input()`, and
- passed pack into the IP input queue.
- Outgoing packets are IPSec processed and encapsulated in `ipsp_process_packet()`.
- SPD is stored in the BSD routing radix tree:
  - SA selection can be based on protocol and port numbers.
  - bypass flows can be stored too.

Kernel Details

- Prevent source address spoofing by specifying permitted source addresses for specific SAs.
- Cryptographic processing of packets is done asynchronously, so that network stack processing does not block:
  - just by itself caused a performance improvement.
- Hard- and soft-expirations based on SA life time, processed bytes, etc...

Kernel Details

- Flows specify IPSec policy:
  - acquire/require flows send `PFKEY ACQUIRE` messages to registered key management daemons,
  - permit/bypass either make use of existing IPSec SAs or bypass IPSec completely.
- Multiple key management daemons can run at the same time, e.g. photuris and isakmpd.

IKE Implementation

- `isakmpd`, new implementation started in 1998 since our IKE daemon had to be
  - scalable, needs to support thousands of SAs,
  - flexible, the standard was still evolving,
- Protocol complexity mandates modular design.
- `isakmpd` is an event driven UNIX daemon using `select(2)`.

IKE implementation

- `isakmpd` supports:
  - Main Mode, Aggressive Mode,
  - Diffie-Hellman groups 1 - 5, including elliptic curve cryptography,
  - Authentication via pre-shared pass phrases and RSA signatures with X509 certificates.
- Main abstraction is the exchange:
  - Provides the context for negotiations as script-driven state machine.
  - Enables high-level syntax checking of messages.

IKE implementation

- Very flexible configuration that allows good interoperability.
- Portable with system dependency layer. Runs on OpenBSD, Linux, NetBSD.
- About 36 KLOC code size (10% platform dependent, 5% regression tests).

Policy using Keynote

- Uses the Keynote trust management system for policy decisions.
- Keynote determines who can negotiate what kind of SAs.
- Allows very fine grained access controls:

```
KeyNote-Version: 2
Authorizer: "POLICY"
Licensees: "passphrase:blubberdubber"
Conditions: app_domain == "IPsec policy" &&
            pfs == "yes" && esp_present == "yes" &&
            esp_enc_alg != "null" -> "true";
```

  (Requires the use of PFS and ESP, where the ESP algorithm is not `null`.)

- Works with X509 certificates, pass phrases and Keynote credentials.

Hardware Crypto Acceleration

- Cryptographic framework manages available hardware accelerators:
  - Asynchronous interface,
  - Callbacks on completion of cryptographic operation.
- Software cryptographic engine as fallback.
- Currently supports:
  - HiFn 7751 (e.g. Powercrypt),
  - Broadcom BCM5805 (ubsec).

Hardware Crypto Acceleration

Engine	Transform	Speed
software (P3/550)	3DES	11 MBit/s
software (P3/550)	Blowfish	34 MBit/s
HiFn	3DES	70 MBit/s
BCM5805	3DES	122 MBit/s

- Broadcom card has no problems dealing with 100 MBit ethernet.

Hardware Crypto Acceleration

- Support for IRE cards coming soon, 3DES > 200 MBit/s
- Currently the first crypto accelerator card that supports an algorithm will be used.
- In the future:
  - We want to round robin the cards in setups with more than one crypto accelerator.
  - Also use hardware acceleration to speedup Diffie-Hellman and RSA operations in `isakmpd`.

VLAN with Bridging and IPSec

- In OpenBSD, a bridge device creates a logical link between a set of network interfaces.
- The bridge copies frames read from one interface to the other interfaces associated with the bridge device.
- Supports filtering on Layer-2 and Layer-3 $\Rightarrow$ transparent firewall.

VLAN with Bridging and IPSec

- Supports extended LAN over the Internet.
  - Ease of administration, no routing required.
- Uses IPSec as encapsulation mechanism:
  - Associate encapsulation `enc` interface with an SA.
  - `enc` interface may become a bridge member.
  - Ethernet in IP over IPSec.

Conclusion

- OpenBSD has state-of-the-art IPSec architecture:
  - IPSec is part of every OpenBSD installation,
  - Very fast with hardware acceleration,
  - Bridge and IPSec allow easy to administrate VPNs.
- This work is mostly due to Niklas Hallqvist, Angelos Keromytis, Håkan Olsson, Jason Wright.
- OpenBSD code and information can be found at `http://www.openbsd.org/`.
- Further reading: `http://www.openbsd.org/crypto.html`.

