Implementing Internet Key Exchange

Niklas Hallqvist
Applitron Datasystem AB
niklas@openbsd.org

Angelos D. Keromytis
Distributed Systems Lab, University of Pennsylvania
angelos@openbsd.org

Talk outline

- Motivation
- Brief IPsec & IKE tutorial
- isakmpd implementation
- Metrics & performance
- Security considerations

Motivations

- Spring 1998, Ericsson Radio Systems wanted IKE
- Should run on multiple platforms
- Evolving standards, flexibility important
- Scalability, must handle tens of thousands of SAs
- Existing software not adequate

Brief IPsec tutorial

- ESP - Encapsulating Security Payload
  - Confidentiality, integrity, authentication, replay protection
- SA - Security Association
  - Parameters necessary for secure communication
- SAD & SPD - SA & Security policy databases
  - Inbound

    IPsec -> SAD -> decrypt/verify -> IP
  - Outbound

    IP -> SPD -> SAD -> encrypt/authenticate -> IPsec

IKE

- Establishes and maintains SAs for IPsec

- Complex standards (drafts at the time)
  - ISAKMP, IPsec DOI, IKE (ISAKMP/Oakley)

- Phase 1
  - Main/Aggressive mode, over UDP port 500, authenticates the peers and negotiates a protected tunnel for phase 2 negotiations.
  - Uses Diffie-Hellman to create key-material.

- Phase 2
  - Quick mode, inside protected tunnel of phase 1, negotiates SAs for the IPsec protocols.
  - Optional Diffie-Hellman provides PFS (Perfect Forward Secrecy)

isakmpd implementation

- Protocol complexity mandated modular design

- Typical event-driven Unix daemon (select(2))
  - Message arrivals, application events, timeouts, signals, user events

- Messages
  - Validation, payload management

- Application
  - Key management, (PF_KEY/PF_ENCAP/Netlink)
  - Security Policy Database management

- User interface
  - Asynchronous
  - Debugging, SA management, dynamic configuration

isakmpd implementation

- Policy
  - Who can establish what kind of SAs?
  - Trust-management system (KeyNote)
  - Work in progress in IPSP/IETF
  - Retroactively fitted-in, uses existing framework for verifying SAs
  - Interoperates with X.509, passphrases, etc.
- Exchanges, an abstraction of its own? If so, how?
  - The context of negotiations, script-driven state machine
  - Provides high-level syntax checking of messages
  - ISAKMP RFC defines many generic exchange attributes
  - Threads? Coroutines?
- Configuration
  - Static? Dynamic?
  - Simple? Flexible?

- Portability
  - System dependency layer
  - Standards are often not
  - Build environment issues, compile-time removable features
- Logging & Debugging
  - IKE is a complex protocol doing many big-number math computations with lots of intermediate results, and it uses cryptography.
  - Too much of the good?
  - Memory abuse? ElectricFence & Boehm GC are good.
  - Demonstration ahead...
- RSA problem
  - Shared libraries provides options
  - Static linking still wanted
  - The problem goes away Sep 21st

Metrics & performance

- Code size
  - 36 KLOCs, 10% platform-dependent, 5% regression-tests
  - KeyNote library 5KLOCs, glue code 1.2KLOCs
    - Mostly data conversion
- Static memory footprint (i386)
  - Fully featured: 950KB
  - Stripped down: 300KB
- Performance
  - DH accounts for almost all (on the order of a second)
  - All else just disappears in comparison
  - Hardware supported assymetric cryptography might change this

Security considerations

- Complexity
  - Single largest problem
  - Modularity helps a bit
  - Protocol generality hurts
  - PROTOCOLS SHOULD NOT BE AS COMPLEX
- Aggressive mode attacks
  - Denial of service attacks by forcing CPU contention
  - Timeouts which removes unfinished exchanges
  - Careful resource tracking

Availability

- Availability
  - ftp://ftp.gsnig.org/pub/isakmpd/
  - isakmpd@gsnig.org (gsnig.net is gone)
  - http://www.openbsd.org/