"IPsec "Opportunistic Encryption"

(where opportunistic here means authenticated)

Paul Wouters <pwouters@redhat.com>

IETF89

Design requirements

- Support standard unmodified applications
- Publish public keys in DNS, protect with DNSSEC
- Trigger on DNS lookups
- Avoid creating "DNS lies"
- Use IKEv2 for IPsec negotiation
- Perform server authentication on client
- Allow optional client authentication by server
- Do not require kernel IPsec modifications
- Applications should be able to get encryption status

The OE mechanism

1. User wants to browse to nohats.ca
2. Application send DNS query for A/AAAA to 127.0.0.1
3. Local DNS server receives query:
    1. If in cache, return A/AAAA record
    2. If not in cache, resolve A/AAAA but also IPSECKEY
    3. Wait for both A/AAAA and IPSECKEY (or proof of non-existence)
    4. If IPSECKEY found, tell IKE daemon to setup IPsec SA's
    5. return A/AAAA record to the application
4. Application sends traffic which is now encrypted using IPsec

Prototype Implementation

- Uses libreswan-3.9 and unbound with python plug-in
- Overhead around 1 second, mostly due to python/execs
- Not yet using draft-smyslov-ipsecme-ikev2-null-auth)
  (uses PSK "test" for client, RSA for server authentication)
- hard fail - if IPSECKEY, then MUST NOT send plaintext
- Available at nohats.ca and libreswan.org
- Supports ipv4 and ipv6
- Uses PSK "test" for client auth
- Will switch to "Integrity Algorithm Transform" value 1024

Outstanding

- IPsec NAT-T handling (implementation or protocol issue?)
- What to do with (multiple) A/AAAA entries?
- What to do when some but not all tunnels fail (like v6)
- Should we limit IKEv2 options (MODP, algo, keysizes) ?
- What to do with DNS TTL versus IKE/IPsec lifetimes?
- Support for mutual authenticated endpoints (on todo list)
- Support for IPsec "gateway servers"? (very difficult problem)
- Should we support un-authenticated IKE (mutual auth-none) ?
- Should we support leap-of-faith (NO!)

A new socket option?

- A socket option for getsockopt() to determine if traffic would be encrypted?
  - Encrypted? or Encrypted and Authenticated?
  - Vulnerable to race condition
- A socket option for setsocktopt() to close/block without encryption

Non-DNS triggered traffic?

- Publishh IPSECKEY in in-addr.arpa
- Could even use the "gateway" option
- Requires packet capture by kernel, signal to IKE daemon
- Requires "negative caching" in IKE daemon or kernel
- Reverse DNS is dying - especially with IPv6

Enterprise "OE" with private DNS zone

Enterprise PKIX deployments are problemetic. Why not use OE?

- Client connects to enterprise network - LAN, WPA2, VPN
- Client issues authenticated NSUPDATE for A and AAAA
- Client issues authenticated NSUPDATE for IPSECKEY
- (and issues NSUPDATE for IPSECKEY in reverse)
- Client to Client encryption via OE IPsec

No more X.509 certificates, expire accidents or human interaction