Running IKE Phase 2 over Artificial Kerberos IKE SA

Tero Kivinen
`<kivinen@ssh.fi>`
SSH Communications Security

IETF-49, San Diego

Draft-ietf-kink-ike-over-kkmp-00.txt

- Running normal Phase 2 over artificial IKE SA
- Artificial IKE SA created directly from kerberos session key
- Can support everything you can do in IKE phase 2 (new group mode, quick mode, delete notification, error notifications)

What is needed for IKE SA

- CKY-I / CKY-R
  - Cookies used to identify the IKE SA
- SKEYID / SKEYID_{e,a,d}
  - Keying material used for various purposes
- Base IV
  - Base IV used to calculate IV for Phase 2 negotiations
- IKE SA algorithms
  - encryption, hash, message authentication algorithms

CKY-I and CKY-R

- Just random numbers
- Identifies the IKE SA
- Should remain constant as long as the kerberos ticket is valid
  - **`CKY-I = SHA-1(kerberos_session_key | 42)[0..15]`**
  - **`CKY-R = SHA-1(kerberos_session_key | 42)[0..15]`**

SKEYID generation

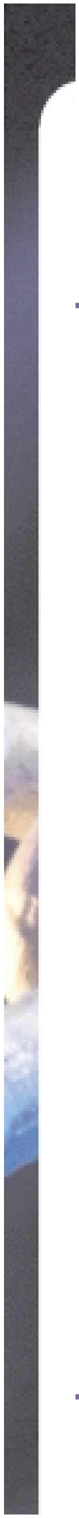
- Generated from the kerberos session key instead of Diffie-Hellman shared secret
- Do not include cookies, as they are generated from the same material
  - `SKEYID = kerberos_session_key`
  - `SKEYID_d = prf(SKEYID, kerberos_session_key | 0)`
  - `SKEYID_a = prf(SKEYID, SKEYID_d | kerberos_session_key | 1)`
  - `SKEYID_e = prf(SKEYID, SKEYID_a | kerberos_session_key | 2)`

Base IV generation

- "Last phase 1 CBC output block"
- Used to generate IV used in the beginning of the new negotiation
- Just random string
- Both ends need to know it
  - BASE-IV = KRB_AP_REQ
  - IV = SHA-1(KRB_AP_REQ | MESSAGE-ID)[0..7]
- Might want to use kerberos session key instead

IKE SA algorithms

- For simplicity use fixed algorithms
  - 3DES, SHA-1, HMAC-SHA-1
- Only used to encrypt IKE SA traffic (i.e less than 1 kB per negotiation)
- Select safe algorithms
- We might also define it so that we always use the same algorithms used to protect KRB_AP_REQ (etype)
- Hash algorithm would still remain fixed



Transmitting KRB_AP_* messages inside IKE

- We define new payload type for adding kerberos packets to IKE packet
- Must be first payload
- That payload is always sent without encryption, encryption starts after it
- It is still calculated to the authentication hash using revised hash calculation
- KRB_AP_REQ is added always
- KRB_AP_REP is optional

One Round Trip Quick Mode

- Normally we do not use PFS
- This means responder can install inbound IPsec SA when it sees first QM packet
- Initiator can install IPsec SA when it sees responders first QM reply packet
- Responder can install outbound IPsec SA when he sees first authenticated packet to IPsec SA or when he sees third QM packet

Example QM Negotiation

Host A **Host B**

HDR, KRB_AP_REQ, *HASH(1), SA, Ni, ... ->

Install Inbound IPsec SA

<- HDR, [KRB_AP_REQ], *HASH(2), SA, Nr, ...

Install IPsec SA, and start using it

HDR*, HASH(3) ->

Install Outbound IPsec SA

Summary

- Reuses phase 2 code from IKE
- Simple to implement
- Only one round trip per IPsec SA
- Will "automatically" benefit from later IKE development