SSHv2 and the GSS-API

Nicolas Williams

Nicolas.Williams@sun.com

Sun Microsystems, Inc.

SSHv2 & SSO

- SSHv2 lacks SSO
  - Pubkey userauth comes close, but no infra
    - Could do x509 userauth, but not specified
  - Requires known host public keys for key exchange
- But SSHv2 is extensible so:
  - New kex methods can be defined
  - New userauth methods can be defined

GSS-API

- Generic Security Service
  - A generic wrapper for Kerberos, PKI, and other forms of authentication and session crypto
- Kerberos is quite popular now for key distribution, authentication and SSO
- GSS-API is screaming for an application like remote secure shell access

A Match Made in Heaven

- SSHv2 + GSS-API ==
  - No host keys
  - SSO
  - No need for pubkey or ssh-agent
  - Leverage Kerberos and PKI infrastructures

SSHv2 + GSS Experience

- Once you have a Kerberos infra and deploy SSHv2 + GSS you stop bothering with pubkey
  - Kerberos credential mgmt is easy and can be transparent to most users
  - Kerberos authorization mgmt is easy too
  - SSHv2 pubkey is harder to manage
- If you already have SSH host keys might as well keep them, otherwise forget 'em

Issues

- Implementation availability
  - OpenSSH w/ Simon Wilkinson's patches came first
  - Draft defines GSS key exchange and userauth
    - Implementors SHOULD give priority to GSS key exchange (see below)
- SSHv2 cryptosystem weakness means frequent re-keys?
  - GSS key-ex is faster than traditional SSHv2 keyex
  - New crypto profiles for SSHv2 (counter mode?)

Issues (cont.)

- Error handling
  - Get it right or users get misleading error messages or silent disconnects
    - Make sure you send GSS error tokens (yes, there is such a thing!)
  - SSHv2 keyex failures are fatal
    - can't be re-tried in same SSHv2 connection
    - So disconnect and let user try again with right GSS target name, valid initiator creds, etc...
      - or w/o GSS

Protocol Concepts

- GSS keyex
  - GSS context establishment
    - Mutual auth, integrity required
    - Can forward credentials
  - DH key exchange
  - Version strings, KEXINIT packets, optional server pubkey, DH pubkeys, shared key bound to GSS ctx
    - MIC of hash of above exchanged

Protocol Concepts

- Re-keying
  - Forward fresh creds (big plus)
  - Server can force re-key
  - Client can force re-key
  - Expired creds fail re-key
    - Should server force re-key when GSS ctx expires?
- “ External-keyex” userauth
  - Authentication taken from GSS keyex
  - No host pubkeys needed
- GSS userauth
  - Independent of keyex
  - Host pubkeys still needed

Questions

- Q&A