SPKM BOF – IETF 67
SSiLKey

Mike Eisler

email2mre-ietf AT

yahoo.com

Background and Motivation

- NFSv4 (and other upper layer protocols) need a certificate-based authentication system
- TLS (or SSL) is ubiquitous, why doesn't NFS use TLS?
  - NFS uses ONC RPC, and ONC RPC does not fit cleanly with TLS
- TLS/SSL enabled http servers are abundant, can we leverage that?
  - Not proposing using https as the transport for NFS

SSiLKey: Session Keys via SSL (aka TLS)

- A GSS initiator creates a SSiLKey context token to <svc_name>@<host_name>
  - Issues an https call to

`https://<host_name>/SSiLKey/InitialContext/service_name?target=<svc_name>@<host_name>`

  - The https server response is encrypted via TLS, and contains:
    - Authenticator, in GSS InitialContext form (i.e. the ASN.1 OID for SSiLKey), consisting of
      - OID of a symmetric encryption algorithm
      - OID of a one way hash algorithm
      - Cipher text (wrapped via encryption and HMAC of one way hash algorithms) using service key consisting of
        - » session key between initiator and target
        - » initiator's certificate (if any)
        - » sequence number (for replay protection)
    - the session key (same as that in Authenticator)
- NFS via RPCSEC_GSS, sends Authenticator NFS server

SSiLKey: Target Processing of Context Token

- NFS server, via RPCSEC_GSS, acquires its credentials (a secret symmetric key accessible to the NFS server and the https server)
- calls GSS_Accept_Sec_Context() which unwraps cipher text in token
  - Checks sequence number for replay
  - If the initiator did not send its certificate, then initiator is anonymous
    - Initiator will use LIPKEY to authenticate itself
    - LIPKEY will layer itself over SSiLKey as it does over SPKM-3
  - If a certificate was sent, the target maps it an operating environment cred (e.g. UNIX uid, gid, gid list)
    - My opinion: implementers should add a gss_ctx_to_uid() call to their GSS libraries if they haven't already
    - If there is a best practice for mapping certs to uids, use it, otherwise, define one and move on

Parts Needed

- Client side: TLS code, http client, SSiLKey GSS initiator
- Server side:
  - https server
  - SSiLKey CGI scripts or binaries to process SSiLKey request for InitialContext
  - Tools for creating random service keys, and service key tabs to store them
  - SSiLKey GSS target