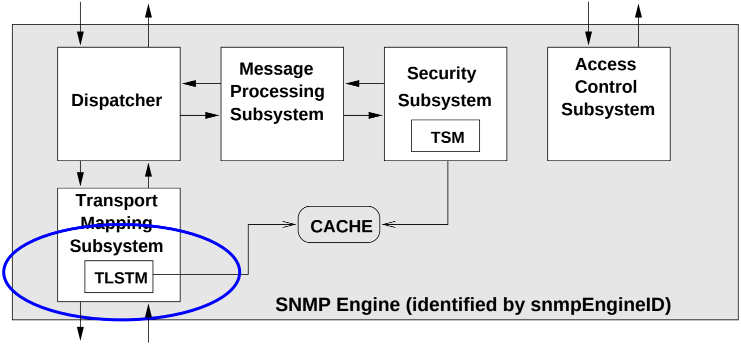
Transport Layer Security (TLS) Transport Model for the Simple Network Management Protocol (SNMP) draft-marinov-isms-tlstm-00

Vladislav Marinov and Jürgen Schönwälder

Jacobs University Bremen
Bremen, Germany

68. IETF March 2007

TLS Transport Model


Dispatcher
Message Processing Subsystem
Security Subsystem
TSM
Access Control Subsystem
Transport Mapping Subsystem
TLSTM
CACHE
SNMP Engine (identified by snmpEngineID)

- The TLS transport model fits into the transport subsystem defined in draft-ietf-isms-tmsm [1]

Session Establishment Overhead (San Diego)

Protocol	Time	Data	Packets
v2c/UDP	1.03 ms	232 bytes	2
v2c/TCP	1.13 ms	824 bytes	10
v3/USM/UDP	1.97 ms	668 bytes	4
v3/USM/TCP	2.03 ms	1312 bytes	12
v2c/SSH	**16.17 ms**	**4388 bytes**	**32**
v2c/TLS	**15.03 ms**	**3930 bytes**	**16**

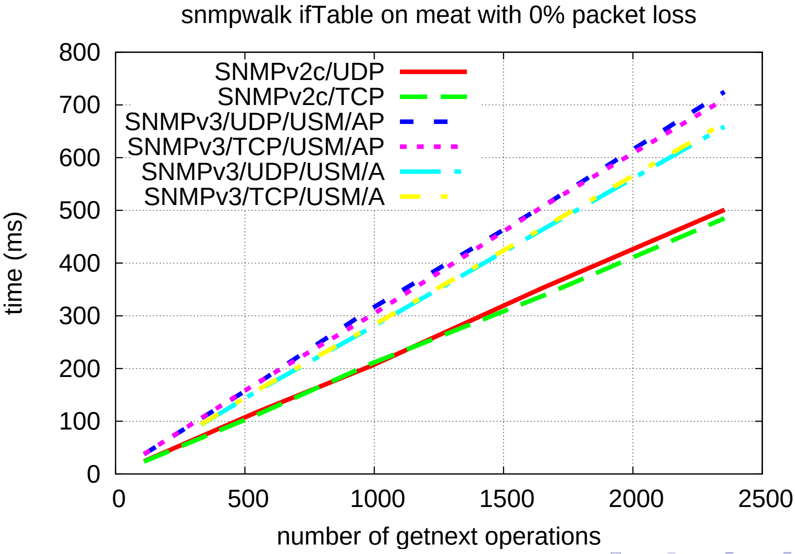
- Prototype implemented using the NET-SNMP open source implementation and the OpenSSL library
- Overhead of TLS session establishment was measured using response time of `snmpget` operation
- **Significant overhead for session establishment!**

Session Establishment Overhead (Prague)

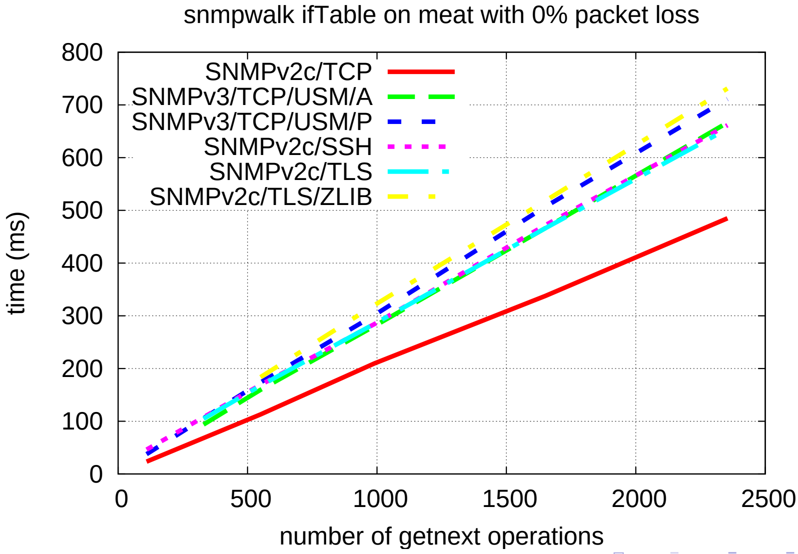
Protocol	Time	Data	Packets
v2c/UDP	1.03 ms	232 bytes	2
v2c/TCP	1.13 ms	824 bytes	10
v3/USM/UDP	1.97 ms	668 bytes	4
v3/USM/TCP	2.03 ms	1312 bytes	12
v2c/SSH	**16.17 ms**	**4388 bytes**	**32**
v2c/TLS	**15.03 ms**	**3930 bytes**	**16**
v2c/TLS/session resumed	**2.00 ms**	**1689 bytes**	**15**

- The session resumption feature of TLS reduces the SNMP session establishment overhead to the range of the USM over UDP and TCP domains
- Are there any known proposals to add something like session resumption to SSH?

Overhead for Long Sessions



Overhead for Long Sessions



Overhead for Long Sessions

- Performance for long sessions measured by using `snmpwalk` on `ifTable`
- All measurements use AES-128 as the encryption mechanism (where encryption is used)
- Performance of both TLS and SSH (without compression) better than USM over TCP or UDP
- Almost no difference between TLS and SSH
- TLS with compression is close to USM over TCP or UDP

Overview of TLS

- TLS Record Protocol [2]: runs on top of some reliable transport (e.g., TCP). Provides data privacy and message integrity. It is used for encapsulation of higher level protocols.
- TLS Handshake Protocol [2]: runs on top of the TLS Record Protocol. It allows the server and client to authenticate each other and to negotiate an encryption algorithm and cryptographic keys before application data is transmitted
- X.509 certificates [3]: Data structures that binds public keys to subjects. X.509 certificates can be signed by certificate authorities (CAs)

Initialization and Authentication

- Agents and Managers are initialized with:
  - X.509 certificates which contain their public keys
  - A list of trusted CAs
  - Certificate revocation list
- Agents and Managers exchange certificates at the beginning of TLS session establishment
- An Agent/Manager verifies that the received peer certificate has been signed by a trusted CA and has not been revoked
- If verification is successful, the common name/alternative name extension field of the X.509 certificate is the authenticated transport dependent security name

Elements of Procedure

- Similar to SSHTM described in draft-ietf-isms-secshell [4]
- If a session is negotiated as "is resumable", retain session and transport parameters (in the LCD) after a session is torn down
- Processing of incoming messages:
    - If a session exists in the LCD but is marked as closed, then try to resume it instead of creating a new one
    - Otherwise, cache session and transport parameters and create an entry in the LCD
- Processing of outgoing messages:
    - Look up session using the `tmStateReference`
    - If session exists and is alive then use it for sending the message (same as SSHTM)
    - If a session exists in the LCD but is marked as closed, then try to resume it instead of creating a new one

DTLS

- Stay tuned!

References

D. Harrington and J. Schoenwaelder.
Internet draft (work in progress) <draft-ietf-isms-tmsm-07.txt> transport subsystem for the simple network management protocol (snmp), March 2007.

T. Dierks and C. Allen.
RFC 2246: The TLS protocol version 1, January 1999.
Status: PROPOSED STANDARD.

R. Housley, W. Ford, W. Polk, and D. Solo.
RFC 3280: Internet X.509 public key infrastructure certificate and CRL profile, April 2002.
Status: PROPOSED STANDARD.

D. Harrington and J. Salowey.
Internet draft (work in progress) <draft-ietf-isms-secshell-05.txt> secure shell transport model for snmp, October 2006.