USEC and Scotty

Implementation details and first experiences.

Jürgen Schönwälder

Department of Computer Science
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

E-mail: schoenw@cs.utwente.nl

What is Scotty?

- Scotty is a network management extension for the Tool Command Language.
- APIs for ICMP, DNS, NTP, HTTP, selected ONC RPCs and SNMPv1/v2c/v2u.
- Freely available (http://www.cs.tu-bs.de/ibr/projects/nm/scotty/).
- How people use the Scotty Tcl extension:
  - Management scripting language to build site-specific management solutions.
  - SNMP prototyping (agents and management applications).
  - Implementation of manager and agent (regression) test tools.
  - Experimental platform for distributed network management.

Why did we add USEC to our SNMP engine?

- Authentication strongly needed for our experiments with programmable agents.
- USEC provides a **simple** solution for our security requirements.

Our USEC implementation history:

- Implementation of the first USEC specification in August '95.
- Experiences reported to the SNMPv2 working group.
- Updates to conform to the RFC 1910 USEC specification in March '96.

USEC changes to the Tcl API:

- Three new SNMP session configuration options:
  - The `-user` option defines the USEC user name.
  - The `-password` option defines the password of a USEC user.
  - The `-context` option may be used to set the context (usually empty).
- Changes to session configuration dialogs to support these new options.
- Nothing else!

USEC changes to the SNMP engine:

- Decoding and encoding of the parameters field.
  - Fast and easy to implement (we do not use ASN.1/BER compilers).
- MD5 digests and localized authentication keys.
  - Very easy. Just take the sources from RFC 1321 and RFC 1910.
- Generating report PDUs.
  - Straight forward and easy to implement.
- Logic to implement the "Elements of Procedure" described in RFC 1910.
  - 11 condensed pages that take some time to implement. A tool which performs "security attacks" would be valuable to test USEC implementations.

USEC changes to the SNMP engine:

- Caches to keep session related state information in the SNMP engine.
  - Tcl SNMP session handles are usually short-lived objects (created just to perform a simple task like walking a table).
  - SNMPv1 and SNMPv2c sessions are completely described by their configuration options – easy and fast to create.
  - USEC session require to "learn" the agentID in order to compute the localized authentication key and to synchronize to the agent time.
  - First cache associates agentID, agentBoots and agentTime values with a transport address.
  - Second cache keeps knowledge about computed authentication keys.

USEC changes to the SNMP engine:

- Fixed length encodings allow to patch the parameters section.
  - The protocol engine keeps BER encoded packets to avoid to call the BER encoder for each retransmission.
  - USEC requires to update the parameters section to

    a) keep the packets inside of the time window and to

    b) adapt requests to newly learned agentIDs.
  - Fixed size encodings allow to patch these fields just before retransmitting an already BER encoded packet.

First experiences with USEC implementations:

- Very easy to configure and use.
- Some simple performance measurements (don't take the numbers too serious):

Test 1 : get sysUpTime.0				(average over 100000 tests)
Manager	Agent	SNMPv2c	SNMPv2u	SNMPv2u MD5
Scotty (Sparc 10)	IBM (RS 6000)	5.8 ms	5.8 ms	7.8 ms
Scotty (Sparc 10)	CMU (Sparc ELC)	3.6 ms	4.0 ms	5.9 ms

Test 2 : getnext [mib successor system]				(average over 100000 tests)
Manager	Agent	SNMPv2c	SNMPv2u	SNMPv2u MD5
Scotty (Sparc 10)	IBM (RS 6000)	8.5 ms	8.8 ms	10.6 ms
Scotty (Sparc 10)	CMU (Sparc ELC)	5.9 ms	6.8 ms	9.3 ms

Where can I get the Scotty Tcl extension?

- RFC 1910 USEC support will be in Scotty version 2.1.0 (and later).
- Requires Tcl7.5/Tk4.1 (version 2.1.0 will not work with Tcl7.4/Tk4.0).
- Will be available shortly after Tcl7.5/Tk4.1 leaves the beta state.
- Watch for an announcement on comp.lang.tcl.announce.
- Subscribe to the tkined mailing list (tkined-request@ibr.cs.tu-bs.de).
- Check the ftp server ftp.ibr.cs.tu-bs.de in /pub/local/tkined.