SNMPv3

TUTORIAL T4 - PART 2
PRESENTED AT NOMS’2004
SEOUL, KOREA
19 APRIL 2004

AIKO PRAS
UNIVERSITY OF TWENTE
THE NETHERLANDS

pras@cs.utwente.nl
http://wwwhome.cs.utwente.nl/~pras
SNMPv3

OVERVIEW:

DESIGN DECISIONS

ARCHITECTURE

SNMP MESSAGE STRUCTURE

SECURE COMMUNICATION
  • USER SECURITY MODEL (USM)

ACCESS CONTROL
  • VIEW BASED ACCESS CONTROL MODEL (VACM)

RFCs
DESIGN DECISIONS

ADDRESS THE NEED FOR SECURITY SET SUPPORT

DEFINE AN ARCHITECTURE THAT ALLOWS FOR LONGEVITY OF SNMP

ALLOW THAT DIFFERENT PORTIONS OF THE ARCHITECTURE MOVE AT DIFFERENT SPEEDS TOWARDS STANDARD STATUS

ALLOW FOR FUTURE EXTENSIONS

KEEP SNMP AS SIMPLE AS POSSIBLE

ALLOW FOR MINIMAL IMPLEMENTATIONS

SUPPORT ALSO THE MORE COMPLEX FEATURES, WHICH ARE REQUIRED IN LARGE NETWORKS

RE-USE EXISTING SPECIFICATIONS, WHENEVER POSSIBLE
CONCEPTS: snmpEngineID
CONCEPTS: snmpEngineID

SYNTAX DEFINED VIA TEXTUAL CONVENTION

OCTET STRING (5..32)

THE VALUE OF snmpEngineID MAY BE DETERMINED BY:
  • HUMAN OPERATOR
  • AUTOMATIC ALGORITHM

AUTOMATIC ALGORITHM USES:
  • PRIVATE ENTERPRISE NUMBER
  • IPv4 ADDRESS / IPv6 ADDRESS / MAC ADDRESS

TEXTUAL CONVENTION DEFINED IN SNMP FRAMEWORK MIB
### THE TERM EngineID IS FREQUENTLY USED

<table>
<thead>
<tr>
<th>Concept</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SnmpEngineID</td>
<td>The textual convention.</td>
</tr>
<tr>
<td>snmpEngineID</td>
<td>The identifier of an SNMP engine.</td>
</tr>
<tr>
<td>securityEngineID</td>
<td>Parameter of primitives in the architecture. The authoritative SNMP entity</td>
</tr>
<tr>
<td></td>
<td>(which is the receiver of a confirmed PDU, the sender of a trap).</td>
</tr>
<tr>
<td>contextEngineID</td>
<td>Parameter in messages. Identifies the engine associated with the data.</td>
</tr>
<tr>
<td>msgAuthoritativeEngineID</td>
<td>Parameter in messages. USM security parameter.</td>
</tr>
<tr>
<td>usmUserEngineID</td>
<td>An object in the snmpUsmMIB.</td>
</tr>
<tr>
<td></td>
<td>In a simple agent, this is the agent’s own snmpEngineID. It may also be the</td>
</tr>
<tr>
<td></td>
<td>snmpEngineID of a remote SNMP engine with which this user can communi-</td>
</tr>
<tr>
<td></td>
<td>cate.</td>
</tr>
<tr>
<td>usmStatsUnknownEngineID</td>
<td>An object in the snmpUsmMIB.</td>
</tr>
<tr>
<td>snmpCommunityContextEngineID</td>
<td>An object in the communityMIB.</td>
</tr>
<tr>
<td>entLogicalContextEngineID</td>
<td>An object in the entityMIB.</td>
</tr>
<tr>
<td>snmpProxyContextEngineID</td>
<td>An object in the proxyMIB.</td>
</tr>
</tbody>
</table>
CONCEPTS: Context

The context can be reached from this engine, thus:
contextEngineID=1

The context can be reached from this engine, thus:
contextName=card1
contextName=card2
MODULES OF THE SNMPv3 ARCHITECTURE

DISPATCHER AND MESSAGE PROCESSING MODULE
• SNMPv3 MESSAGE STRUCTURE
  • snmpMPDMIB
    • RFC 3412

APPLICATIONS
• snmpTargetMIB
• snmpNotificationMIB
• snmpProxyMIB
  • RFC 3413

SECURITY SUBSYSTEM
• USER BASED SECURITY MODEL
  • snmpUsmMIB
    • RFC 3414

ACCESS CONTROL SUBSYSTEM
• VIEW BASED ACCESS CONTROL MODEL
  • snmpVacmMIB
    • RFC 3415
SNMPv3 MESSAGE STRUCTURE

- **msgVersion**
- **msgID**
- **msgMaxSize**
- **msgFlags**
- **msgSecurityModel**

**msgSecurityParameters**

- **contextEngineID**
- **contextName**

**PDU**

**USED BY**

- MESSAGE PROCESSING SUBSYSTEM
- SNMPv3 PROCESSING MODULE
- SECURITY SUBSYSTEM
- ACCESS CONTROL SUBSYSTEM AND APPLICATIONS
SNMPv3 PROCESSING MODULE PARAMETERS

- **msgVersion**: 0..2147483647
- **msgID**: 0..2147483647
- **msgMaxSize**: 484..2147483647
- **msgFlags**: authFlag, privFlag, reportableFlag
- **msgSecurityModel**: USM
- **msgSecurityParameters**: SNMPv1, SNMPv2c
- **contextEngineID**
- **contextName**
- **PDU**
SECURE COMMUNICATION VERSUS ACCESS CONTROL

MANAGER

APPLICATION PROCESSES

AGENT

MIB

TRANSPORT SERVICE

GET / GET-NEXT / GETBULK
SET / TRAP / INFORM

ACCESS CONTROL

SECURE COMMUNICATION
## USM: SECURITY THREATS

<table>
<thead>
<tr>
<th>THREAT</th>
<th>ADDRESSED?</th>
<th>MECHANISM</th>
</tr>
</thead>
<tbody>
<tr>
<td>REPLAY</td>
<td>YES</td>
<td>TIME STAMP</td>
</tr>
<tr>
<td>MASQUERADE</td>
<td>YES</td>
<td>MD5 / SHA-1</td>
</tr>
<tr>
<td>INTEGRITY</td>
<td>YES</td>
<td>(MD5 / SHA-1)</td>
</tr>
<tr>
<td>DISCLOSURE</td>
<td>YES</td>
<td>DES</td>
</tr>
<tr>
<td>DENIAL OF SERVICE</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>TRAFFIC ANALYSIS</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>USM MESSAGE STRUCTURE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>msgVersion</td>
<td></td>
<td></td>
</tr>
<tr>
<td>msgID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>msgMaxSize</td>
<td></td>
<td></td>
</tr>
<tr>
<td>msgFlags</td>
<td></td>
<td></td>
</tr>
<tr>
<td>msgSecurityModel</td>
<td></td>
<td></td>
</tr>
<tr>
<td>msgAuthoritativeEngineID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>msgAuthoritativeEngineBoots</td>
<td></td>
<td></td>
</tr>
<tr>
<td>msgAuthoritativeEngineTime</td>
<td></td>
<td></td>
</tr>
<tr>
<td>msgUserName</td>
<td></td>
<td></td>
</tr>
<tr>
<td>msgAuthenticationParameters</td>
<td></td>
<td></td>
</tr>
<tr>
<td>msgPrivacyParameters</td>
<td></td>
<td></td>
</tr>
<tr>
<td>contextEngineID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>contextName</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PDU

- **REPLAY**
- **MASQUERADE/INTEGRITY/DISCLOSURE**
- **MASQUERADE/INTEGRITY**
- **DISCLOSURE**
IDEA BEHIND REPLAY PROTECTION

Nonauthoritative Engine

LOCAL NOTION OF REMOTE CLOCK

ID | BOOTS | TIME | DATA

Authoritative Engine

ALLOWED LIFETIME

LOCAL CLOCK

ID | BOOTS | TIME | DATA

+ -> >?
IDEA BEHIND DATA INTEGRITY AND AUTHENTICATION

ADD THE MESSAGE AUTHENTICATION CODE (MAC) TO THE DATA AND SEND THE RESULT
IDEA BEHIND AUTHENTICATION

Diagram showing the process of authentication:

1. USER provides KEY and DATA to the system.
2. DATA is processed through the HASH FUNCTION to produce MAC.
3. MAC is appended to DATA to form a MAC-DATA package.
4. The system compares the received MAC-DATA with the expected MAC-DATA.
5. If the MACs match, access is granted; otherwise, access is denied.
IDEA BEHIND THE DATA CONFIDENTIALITY (DES)

DES-KEY → DATA → DES ALGORITHM → ENCRYPTED DATA
IDEA BEHIND ENCRYPTION

DES-KEY  DATA

DES ALGORITHM

ENCRYPTED DATA

USER  ENCRYPTED DATA

DES-KEY  DATA

DES ALGORITHM

ENCRYPTED DATA

USER  ENCRYPTED DATA
VIEW BASED ACCESS CONTROL MODEL

ACCESS CONTROL TABLE

MIB VIEWS
# ACCESS CONTROL TABLES

<table>
<thead>
<tr>
<th>MIB VIEW</th>
<th>ALLOWED OPERATIONS</th>
<th>ALLOWED MANAGERS</th>
<th>REQUIRED LEVEL OF SECURITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Table</td>
<td>SET</td>
<td>John</td>
<td>Authentication Encryption</td>
</tr>
<tr>
<td>Interface Table</td>
<td>GET / GETNEXT</td>
<td>John, Paul</td>
<td>Authentication</td>
</tr>
<tr>
<td>Systems Group</td>
<td>GET / GETNEXT</td>
<td>George</td>
<td>None</td>
</tr>
</tbody>
</table>

...
MIB VIEWS
SNMPv3 RFCs

**SNMP ENTITY**

RFC 3411

**SNMP APPLICATIONS**

RFC 3413

**SNMP ENGINE**

RFC 3412

DISPATCHER

RFC 3412

MESSAGE PROCESSING SUBSYSTEM

USM: RFC 3414

SECURITY SUBSYSTEM

VACM: RFC 3415

ACCESS CONTROL SUBSYSTEM