Implementation and Performance Analysis of SNMP on a TLS/TCP Base.

Du, Shayman and Rozenblitz

Sarwar S Raza
WPI – CS 577
Outline

- Overview of SNMP
- Introduction
- TLS
- Implementation of SNMP/TLS/TCP
- Performance comparisons
- Conclusions
Network Management

- A **Network Management System** is composed of:
  - Network elements or nodes containing a processing entity called an agent, responsible for performing the management functions requested by the management station.
  - Management applications which monitor, configure and control managed elements.
  - A management protocol used to communicate management information between the management stations and the agents in the network elements.
  - Management information.
Network Management

Network Management Station

- Application
- Application
- Manager

Device

- Agent
- Configuration Data
- Status Parameters
- Statistics

- Read or Change Configuration
- Read or Change Status
- Read Performance or Error Stats

- Respond to Request
- Report Problems
SNMP: A brief history
Simple Network Management Protocol

- SNMP is the prevailing standard for management of TCP/IP networks. SNMP is layered on top of UDP, the User Datagram Protocol.
- An SNMP management station monitors and controls a managed node by issuing requests directed to the agent residing in the managed node. The agent interprets the request and performs the function accordingly.
- All SNMP transactions take place using PDUs (Protocol Data Units).
Simple Network Management Protocol

- IETF RFCs 1155, 1156, and 1157 define the Simple Network Management Protocol (SNMP). The Internet community developed SNMP to allow diverse network objects to participate in a global network management architecture. Network managing systems can poll network entities implementing SNMP for information relevant to a particular network management implementation. Network management systems learn of problems by receiving traps or change notices from network devices implementing SNMP.
Choice of underlying protocol

- UDP has been chosen and recommended for SNMP transport protocol. This is fine because at the beginning, SNMP was targeted at managing Internet nodes and the predominant Internet protocol suite TCP/IP. The choice of TCP/IP suite continues to make sense because IP became the protocol for commercial backbone networks. And users can count on a TCP/IP implementation available on any type of host and router.
Choice of underlying protocol

• TCP and UDP provide transport services. But UDP was preferred. This is due to TCP characteristics, it is a relatively complicated protocol and consumes more memory and CPU resources, whereas a UDP stack is easy to build and run. Device vendors need only have built simple version of IP and UDP on their devices. Thus the software requirements are kept simple enough, and, can, in most cases, be stored in ROM. UDP is well suited to the brief request / response message used in network management communication.
SNMP PDU types

- There are 4 types of request PDUs:
  - GET  get a specific name or instance
  - GET-NEXT  get the next-object that follows the given name/instance
  - SET  set variables(s)
  - TRAP  report a trap event that has occurred.
SNMP Message format

<table>
<thead>
<tr>
<th>Version</th>
<th>Community</th>
<th>PDU</th>
</tr>
</thead>
</table>

# PDU Format

<table>
<thead>
<tr>
<th>PDU type</th>
<th>Request ID</th>
<th>Error status</th>
<th>Error index</th>
<th>Object 1, value 1</th>
<th>Object 2, value 2</th>
<th>…</th>
</tr>
</thead>
</table>
The GET and GETNEXT PDUs

• The GET and GET-NEXT PDUs consists of pairs of variables to get, and a value of null for those variables.

• The generate a RESPONSE PDU, where the null value is replaced with the data that was requested, or an error code is sent back.

• The variable/value pairs are called variable bindings. Multiple varbinds may be enclosed in a single PDU.
The SET PDU

• The SET PDU consists of pairs of variables to set, and the value that the network operator wants to assign to those variables.

• It generates a RESPONSE PDU from the agent, that contains success/error status and the value of the variable that was just SET.

• Though returning back the same value is redundant, it allows for the SET-RESPONSE PDU to be virtually identical to the GET/GET-NEXT RESPONSE PDU.
The TRAP PDU

• An agent can *asynchronously* send an unsolicited TRAP to the management application to signal an extraordinary event.
Management Information Base

• Defines the management information that is exchanged between the managed node and the management application.

• A unit of managed information, referred to previously as a variable, is called a *Managed Object*.

• A MIB is a collection of Managed Objects.

• MIBs are specified in ASN.1, a somewhat primitive data declaration language.
The complete picture
Example of use

- You want to bring a switch port from a state of *Down* to a state of *Up*.
- The Switch port state is identified by a variable ‘SwPortState’ where:
  \[ \text{Down} = 0 \]
  \[ \text{Up} = 1. \]
Example

• We can first look at the current status of the variable using an SNMP get-request.

• GET: variable = SwPortState, value = null

• GET Response: variable = SwPortState, value = 0, error = no error
Example

• To change the value of the variable, use an SNMP set request.

• SET: variable = SwPortState, value = 1

• SET Response: variable = SwPortState, value = 1, error = no error.
Example

- After the set has been completed, the AGENT will run handlers on the device that will change the port state from down(0) to up(1).
Security in SNMP

- SNMP v1 – very limited security
  - Security in SNMP is commonly referred to as *trivial authentication*.
  - You must know the device’s IP address in order to talk to it.
  - Your must also know the *community string*, a “password” that is sent in *clear text* as part of the SNMP message.
Security improvements – SNMP V3

• SNMPv3 provides encryption and authentication as part of the core protocol. Specifically, SNMPv3 with USM (User based security model) recognizes three levels of security:

1. Without authentication and without privacy (noAuthNoPriv)
2. With authentication but without privacy (authNoPriv)
3. With authentication and privacy (authPriv)
The authors’ premise…

- When large amounts of data need to be transferred, they must be transported using small-sized SNMP over UDP messages which result in excessive latency.

- Transporting SNMP over TCP reduces the latency by removing the limitation on message size and by allowing several segments of data to be in transit at the same time (due to TCP window mechanism).
The authors’ premise...

- TCP has the additional advantage of taking care of retransmission. This GREATLY simplifies management applications since retransmission need not be implemented at the application level, but can be relegated to the transport protocol.
Protocol level Security options

• SNMP over UDP $\rightarrow$ IPSec at layer 3
• SNMP over TCP $\rightarrow$ TLS
TLS Introduction

• TLS = Transport Layer Security
• A protocol that provides communication security over the Internet.
• Is aimed at preventing eavesdropping, tampering and message forgery between client server communications.
• Based on SSLv3, and is an IETF standard.
TLS Layers

- The protocol is composed of two layers:
  1. The TLS Record Protocol
TLS Record Protocol

• Provides connection security with two basic properties:

1. Connection privacy. Symmetric cryptography is used for data encryption (e.g. DES, 3DES, RC4). Encryption can be turned off.

2. The connection is securely reliable. Message transport includes a keyed cryptographic message authentication check (MAC).
TLS Handshake Protocol

- Allows the server and client to authenticate each other and to negotiate an encryption algorithm and cryptographic keys before the application protocol transmits or receives its first byte of data.
TLS Handshake protocol

- Allows peers to authenticate their identities, using asymmetric, or public key cryptography.
- Man in the middle attacks can be thwarted – the negotiation of a shared secret is secure.
- The negotiation is reliable. It is not possible to modify the traffic being communicated without one or both parties being alerted.
Implementation of SNMP/TLS/TCP

1. Setup TCP connection
2. Setup TLS over TCP
3. Begin client server communication
4. TLS Change Cipher Spec Protocol causes the pending cipher state to be copied into current cipher state
5. TLS Alert protocol is used to convey TLS related alerts to peer
TLS Handshake and Record Protocols

--- Begin TLS Handshake Protocol ---

<table>
<thead>
<tr>
<th>Client</th>
<th>Server</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TLS_client_hello</strong></td>
<td><strong>TLS_get_client_hello</strong></td>
</tr>
<tr>
<td><strong>TLS_get_server_hello</strong></td>
<td><strong>TLS_send_server_hello</strong></td>
</tr>
<tr>
<td><strong>TLS_get_server_certificate</strong></td>
<td><strong>TLS_send_server_certificate</strong></td>
</tr>
<tr>
<td><strong>TLS_get_key_exchange</strong></td>
<td><strong>TLS_send_server_key_exchange</strong></td>
</tr>
<tr>
<td><strong>TLS_get_certificate_request</strong></td>
<td><strong>TLS_send_certificate_request</strong></td>
</tr>
<tr>
<td><strong>TLS_get_server_done</strong></td>
<td><strong>TLS_send_server_done</strong></td>
</tr>
<tr>
<td><strong>TLS_send_client_certificate</strong></td>
<td><strong>TLS_get_client_certificate</strong></td>
</tr>
<tr>
<td><strong>TLS_send_client_key_exchange</strong></td>
<td><strong>TLS_get_client_key_exchange</strong></td>
</tr>
<tr>
<td><strong>TLS_send_client_verify</strong></td>
<td><strong>TLS_get_cert_verify</strong></td>
</tr>
<tr>
<td><strong>TLS_change_cipher_spec</strong></td>
<td><strong>TLS_change_cipher_spec</strong></td>
</tr>
<tr>
<td><strong>Finished</strong></td>
<td><strong>Finished</strong></td>
</tr>
</tbody>
</table>

--- End TLS handshake protocol ---

--- Begin TLS Record Protocol ---

TLS Read OR TLS Write


--- End TLS Record Protocol ---

--- End TLS Session ---
Performance Tests and Results

• Measurement environment:
  Network: Ethernet 10Mbit
  Hardware: One Sun Sparc 10 workstation (manager), 128 MB RAM; One Sun Sparc 5 workstation (agent), 128 MB RAM.
  Software: Solaris 2.6, UCD-SNMP agent SNMP/TLS/TCP implementation
Overhead of TLS Security

Measuring overhead of TLS security

4 scenarios:

1. No security, no compression, no MAC, no encryption
2. Integrity protection only: no compression, has MAC, no encryption
3. Privacy protection only: has compression, no MAC, has encryption
4. Integrity and Privacy Protection: has compression, has MAC, has encryption.
Overhead of TLS Security

- Tests were performed for short sessions (single message, single SNMP variable queried) and for longer sessions, where GET-NEXT was used to walk across an object group in a MIB (34 objects grouped under System).
Overhead of TLS Security

![Graph showing time in milliseconds for different security scenarios (No security, Integrity protection only, Privacy protection only, Integrity and Privacy Protection) comparing Snmpwalk and Snmpget.]
Analysis

• For the short session, Integrity protection takes 4.01% of the session time, privacy protection about 4.52%. Total security takes 8.53% of the session time.

• For the long session, integrity protection takes 7.28%, privacy protection 14.66% and total security 21.94% of the session time.
Analysis

• Longer latency times for long session can be explained as follows:
  The setup times for SNMP, TCP and TLS are incurred only once per session. However, MAC and encryption overheads are incurred for each message in the session.

• NOTE: actual latency per message actually decreases for longer messages.
Overhead of TLS/TCP Session Setup

- SNMP/UDP does not incur a setup penalty analogous to the setup of TLS/TCP, where the TLS handshake protocol is used for the client and server to authenticate each other.

- For long sessions, however, this ONE TIME setup cost gets amortized over a large number of messages and only amounts to a low amount of overhead per message.
Test setup

• Since SNMPv1/UPD has no security, for testing purposes, it was first only to SNMPv1/TLS/TCP with peer authentication, but no integrity or privacy protection.
• TLS with full security also used in separate comparison.
• TLS setup time found to be constant: 300 ms
Analysis

Over of TLS/TCP Session Setup

- Ratio: Secure TLS time to UDP time
- Ratio: TLS time to UDP time

Number of messages per session vs. Ratio
Analysis

- When the number of messages in one session is small, the TLS message time is about 1.4~1.6 times the UDP message time. As the number of messages increases, the ratio declines to approximately 1.2 for sessions containing at least 500 messages.

- For cases where number of messages $> 500$, the ratio of the per message time for TCS and UDP becomes essentially constant.
Light vs. Heavy traffic

- Light traffic – one SNMP transaction every 5 minutes
- Heavy traffic – one SNMP transaction per second.
- NOTE: for TLS/TCP, each transaction requires new TLS session.

<table>
<thead>
<tr>
<th>Time</th>
<th>TLS/TCP</th>
<th>TLS Setup</th>
<th>UDP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Light traffic (/5mins.)</td>
<td>648</td>
<td>296</td>
<td>123</td>
</tr>
<tr>
<td>Heavy traffic (/sec.)</td>
<td>424</td>
<td>273</td>
<td>91</td>
</tr>
</tbody>
</table>
Packet by Packet Timing analysis

<table>
<thead>
<tr>
<th>Message</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>TLS</td>
<td>3.6</td>
<td>3.5</td>
<td>3.4</td>
<td>3.5</td>
<td>3.4</td>
<td>3.5</td>
<td>3.7</td>
<td>3.7</td>
<td>3.5</td>
<td>3.5</td>
<td>3.5</td>
</tr>
<tr>
<td>UDP</td>
<td>3.0</td>
<td>3.3</td>
<td>3.1</td>
<td>3.0</td>
<td>3.0</td>
<td>3.1</td>
<td>3.2</td>
<td>3.2</td>
<td>3.1</td>
<td>3.1</td>
<td>3.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Message</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
<th>21</th>
<th>22</th>
</tr>
</thead>
<tbody>
<tr>
<td>TLS</td>
<td>3.6</td>
<td>3.6</td>
<td>3.6</td>
<td>3.9</td>
<td>3.6</td>
<td>3.7</td>
<td>3.6</td>
<td>3.6</td>
<td>3.8</td>
<td>3.7</td>
<td>3.7</td>
</tr>
<tr>
<td>UDP</td>
<td>3.2</td>
<td>3.2</td>
<td>3.3</td>
<td>3.4</td>
<td>3.9</td>
<td>3.4</td>
<td>3.2</td>
<td>3.2</td>
<td>3.3</td>
<td>3.3</td>
<td>3.3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Message</th>
<th>23</th>
<th>24</th>
<th>25</th>
<th>26</th>
<th>27</th>
<th>28</th>
<th>29</th>
<th>30</th>
<th>31</th>
<th>32</th>
<th>33</th>
<th>34</th>
</tr>
</thead>
<tbody>
<tr>
<td>TLS</td>
<td>3.7</td>
<td>3.7</td>
<td>3.9</td>
<td>3.6</td>
<td>3.6</td>
<td>3.6</td>
<td>3.6</td>
<td>3.9</td>
<td>3.6</td>
<td>3.6</td>
<td>3.6</td>
<td>3.7</td>
</tr>
<tr>
<td>UDP</td>
<td>3.2</td>
<td>3.3</td>
<td>3.3</td>
<td>3.2</td>
<td>3.2</td>
<td>3.2</td>
<td>3.2</td>
<td>3.3</td>
<td>3.2</td>
<td>3.2</td>
<td>3.2</td>
<td>3.4</td>
</tr>
</tbody>
</table>
Analysis

• Individual message exchange times are larger in TLS/TCP than under UDP. This is because of the TLS/TCP session has to implement the TLS record protocol when sending and receiving data. Thus for each message, the following takes place:
  1. Fragment data/reassemble data
  2. Compress/decompress
  3. Calculate client/server MAC
  4. Encrypt/Decrypt
  5. Append/Remove TLS Record header.
Comparison of SNMP/TLS/TCP w/ SNMPv3/UDP and SNMPv3/TCP

- MD5 is used as the authentication protocol
- DES is used as the encryption algorithm
- All SNMPv3/TLS/TCP and SNMPv1/TLS/TCP are without USM
- SNMPv3/TCP and SNMPv3/UDP have USM.
## Snmpget – short sessions

<table>
<thead>
<tr>
<th>SNMP-v1 security feature</th>
<th>a</th>
<th>b</th>
<th>b - a</th>
<th>d</th>
<th>d - b</th>
<th>d - a</th>
</tr>
</thead>
<tbody>
<tr>
<td>Or corresponding</td>
<td>NoAuth</td>
<td>Auth</td>
<td>Auth</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SNMP-v3 security level</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Snmpget-v1/UDP</td>
<td>472</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Snmpget-v1/TCP</td>
<td>523</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Snmpget-v1/TLS/TCP</td>
<td>774</td>
<td>805</td>
<td>31</td>
<td>840</td>
<td>35</td>
<td>66</td>
</tr>
<tr>
<td>Snmpget-v3/TLS/TCP (no USM)</td>
<td>976</td>
<td>989</td>
<td>13</td>
<td>1,124</td>
<td>136</td>
<td>148</td>
</tr>
<tr>
<td>Snmpget-v3/UDP (USM)</td>
<td>665</td>
<td>1,632</td>
<td>967</td>
<td>2,735</td>
<td>1,103</td>
<td>2,070</td>
</tr>
<tr>
<td>Snmpget-v3/TCP (USM)</td>
<td>881</td>
<td>1,990</td>
<td>1,109</td>
<td>3,634</td>
<td>1,644</td>
<td>2,753</td>
</tr>
<tr>
<td>Ratio : v3-UDP / v1-TLS-TCP</td>
<td>85.9%</td>
<td>203%</td>
<td>326%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ratio : v3-UDP / v3-TLS-TCP</td>
<td>68.1%</td>
<td>165%</td>
<td>217%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ratio : v3-TCP / v1-TLS-TCP</td>
<td>114%</td>
<td>247%</td>
<td>133%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ratio : v3-TCP / v3-TLS-TCP</td>
<td>90.3%</td>
<td>201%</td>
<td>108%</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Snmpwalk – long sessions

<table>
<thead>
<tr>
<th>SNMP-v1 security feature</th>
<th>a</th>
<th>b</th>
<th>b - a</th>
<th>d</th>
<th>d - b</th>
<th>d - a</th>
</tr>
</thead>
<tbody>
<tr>
<td>Or corresponding NoAuth</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SNMP-v3 security level</td>
<td>NoPriv</td>
<td>Auth</td>
<td></td>
<td>Auth</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Snmpwalk-v1/UDP</td>
<td>678</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Snmpwalk-v1/TCP</td>
<td>762</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Snmpwalk-v1 TLS/TCP</td>
<td>1,044</td>
<td>1,120</td>
<td>76</td>
<td>1,273</td>
<td>153</td>
<td>229</td>
</tr>
<tr>
<td>Snmpwalk-v3/TLS/TCP (no USM)</td>
<td>1,063</td>
<td>1,135</td>
<td>72</td>
<td>1,323</td>
<td>188</td>
<td>260</td>
</tr>
<tr>
<td>Snmpwalk-v3/UDP (USM)</td>
<td>648</td>
<td>1,848</td>
<td>1,200</td>
<td>2,976</td>
<td>1,128</td>
<td>2,328</td>
</tr>
<tr>
<td>Snmpwalk-v3/TCP (USM)</td>
<td>947</td>
<td>2,025</td>
<td>1,078</td>
<td>3,305</td>
<td>1,280</td>
<td>2,358</td>
</tr>
<tr>
<td>Ratio v3-UDP / v1-TLS-TCP</td>
<td>62.1%</td>
<td>165%</td>
<td></td>
<td>234%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ratio v3-UDP / v3-TLS-TCP</td>
<td>60.9%</td>
<td>163%</td>
<td></td>
<td>225%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ratio v3-TCP / v1-TLS-TCP</td>
<td>90.7%</td>
<td>181%</td>
<td></td>
<td>260%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ratio v3-TCP / v3-TLS-TCP</td>
<td>89.1%</td>
<td>178%</td>
<td></td>
<td>249%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Analysis

• With minimal security, SNMPv3/UDP always faster than SNMPv3/TCP
• Addition of security makes a HUGE difference.
• In summary, SNMP/TLS/TCP without USM far more efficient than using SNMPv3 (with USM) over TCP and UDP, with same security features.
Conclusions

• Authors assert that session setup overhead and per message security overhead not excessive (A subjective claim). They see SNMP/TLS/TCP as a viable choice for secure, reliable network management.

• Comparisons with SNMPv3: Authors assert that it is not at all clear to them to what degree the advantages of are ‘structural’, and to what degree they are affected by relative levels of code optimization in their implementation.

• Further experience with different software implementation is required to verify generality of results.