U. Blumenthal
Intel Corporation
L. Dondeti
Nortel Networks
R. Presuhn, Ed.
R. Presuhn Consulting
E. Rescorla
RTFM, Inc.
February 13, 2005

Comparison of Proposals for Integrated Security Models for SNMP (Simple Network Management Protocol)
draft-ietf-isms-proposal-comparison-00

Status of this Memo

Copyright Notice

Abstract

Although the Simple Network Management Protocol (SNMPv3) is secure, operators and administrators have found that deploying it can be problematic in large distributions, due to a lack of integration between its User-Based Security Model (USM) and other authentication and user management infrastructure. This memo contains an evaluation of three proposals for an integrated security model for SNMP, and, based on these proposals, suggests how the ISMS (Integrated Security Model for SNMP) working group might move forward.

Table of Contents

1. Introduction

SNMPv3 became a full Internet Standard in late 2002, adding security to the previous versions of the protocol. Although the enhanced protocol is secure, operators and administrators have found that deploying it can be problematic in large distributions, due to a lack of integration between USM and other authentication and user management infrastructure. This memo contains an evaluation of three proposals for an integrated security model for SNMP, and, based on these proposals, suggests how the ISMS working group might move forward. Readers should be reminded that, in accordance with [RFC2418] section 6.5, the working group is free to adopt, reject, ignore or modify these recommendations in whatever way it sees fit.

1.1 Terms and Abbreviations

The following list defines the terms and abbreviations used here, and, where appropriate, provides references to further information on the relevant concepts.

```
AAA:        Authentication, Authorization and Accounting [RFC2903]
CLI:        Command Line Interface
Diameter:   (not an acronym) [RFC3588]
DTLS:       Datagram Transport Layer Security [I-D.rescorla-dtls]
EUSM:       External User Security Model
            [I-D.kaushik-snmp-external-usm]
IKE:        Internet Key Exchange [RFC2409]
IPsec:      IP Security [RFC2401]
Kerberos:   a third-party authentication protocol defined in
            [RFC1510]
LDAP:       Lightweight Directory Access Protocol [RFC2251]
MAC:        Message Authentication Code
PDU:        Protocol Data Unit
RADIUS:     Remote Authentication Dial In User Service [RFC2865]
Session:    Not quite a connection, nor an association, and not to
            be confused with a CLI session, we use the term
            session here to refer to a sequence of exchanges
            between two SNMP engines making on behalf of a single
            user and secured by the same key, as well as any PDU
            exchanges needed to establish or tear down the
            session.
SBSM:       Session Based Security Model
            [I-D.hardaker-snmp-session-sm]
SNMP:       Simple Network Management Protocol [RFC3410]
SSH:        Secure Shell protocol [I-D.ietf-secsh-architecture]
```

TACACS+: Terminal Access Controller Access Control System (plus) [RFC1492]
TCP: Transmission Control Protocol [RFC0793]
TLS: Transport Layer Security protocol [RFC2246]
TLSM: Transport Layer Security Model [I-D.schoenw-snmp-tlsm]
UDP: User Datagram Protocol [RFC0768]
USM: User-Based Security Model [RFC3414]
VACM: View-Based Access Control Model [RFC3415]

2. Overview

Version 3 of the Simple Network Management Protocol (SNMPv3) was elevated to Internet Standard in late 2002 and added security to the previous versions of the protocol. Although the enhanced protocol is secure, operators and administrators have found that deploying it could be problematic in large distributions. There have been two major sources of difficulty. First, most networking devices already contain local accounts or the ability to negotiate with authentication servers, such as RADIUS servers. However, SNMPv3 does not make use of these authentication mechanisms, but instead adds its own SNMPv3-specific authentication system, which needs to be maintained across all networking devices. Secondly, the distribution and maintenance of View-based Access Control Model (VACM) rules is also difficult in large-scale environments.

2.1 Background

In principle, SNMP has modular security, with communications security being provided via different "security models". In practice, SNMP security is done via the "User-based Security Model" [RFC3414], which is essentially per-message encryption and authentication inside of SNMP. USM keying is based on per-user shared keys, used between SNMP engines that need to communicate securely on behalf of management applications. (The traditional terms "manager" and "agent" are not particularly helpful, since both kinds of applications may make use of a given SNMP protocol engine at the same time.) Key localization techniques are used to minimize the impact of the compromise of a shared key.

There are two major sources of discontent with USM:

1. Key management with manual keying is extremely difficult in any system. SNMP is no different. (major reason)
2. USM anti-replay protection is limited by design (minor reason). A message may be replayed within a 150 second window. Note that for set operations where ordering and non-duplication can be important, this can be largely mitigated by the use of TestAndIncr [RFC2579] objects.

A number of alternative designs have been proposed to improve SNMP security. We discuss three here:

EUSM: External User Security Model [I-D.kaushik-snmp-external-usm]
SBSM: Session Based Security Model [I-D.hardaker-snmp-session-sm]
TLSM: Transport Layer Security Model [I-D.schoenw-snmp-tlsm]

Evaluating these documents is difficult because they vary along two primary axes:

1. Architectural -- each of these designs is very different in terms of how it integrates with the SNMP architecture.
2. Features -- each design provides some support for automatic key management, but with a fair amount of variety in the kinds of credentials supported.

To a first order, these concerns are orthogonal. The intent of this document is to evaluate the merits of various architectural approaches, without regard to the specific implementation details of the authentication mechanisms proposed in these drafts. In order to do this, we will confine the following discussion to idealized architectural sketches of the approach used by each protocol. We begin by describing the existing User-based Security Model, both for reference and to simplify comparison of the other schemes.

2.2 Goals

The ISMS (Integrated Security Model for SNMP) working group was chartered to identify a solution for the first of the two above-mentioned problems: creating a security model for SNMPv3 that will meet the security and operational needs of network administrators. The goals were to maximize usability in operational environments to achieve high deployment success and at the same time minimize implementation and deployment costs to minimize the time until deployment would be possible. The ability to make use of existing and commonly deployed security infrastructure was a requirement, as was consideration of the following as potential existing authentication infrastructures to make use of within the new security model, with at least one being mandatory:

* Local accounts
* SSH identities
* RADIUS
* TACACS+
* X.509 Certificates
* Kerberos
* LDAP

* Diameter

The working group's charter constrains the solution. It must not modify the other aspects of SNMPv3 protocol as defined in STD 62. It should also be compliant with the security model architectural block of SNMPv3, as outlined in [RFC3411]. Finally, it should also not change any other protocols.

In addition to goals and requirements given in the working group charter, discussion on the mailing list and in working group meetings helped identify additional points to consider in evaluating proposals:

* Must be at least as secure as USM [RFC3414].
* Must not preclude the use of USM [RFC3414], particularly if network instability could cause problems for the proposed solution
* Must be able to work with VACM.
* The protocol itself should support multiple security infrastructures, but an implementation may support some subset of these.
* Must not break basic device discovery. (Retaining USM support would satisfy this goal.)

In the documents and on the mailing list, some additional potential goals and requirements have been mentioned, but did not seem to enjoy widespread support. This does not mean that the evaluation team contests that they may be desirable; it simply means that these were given secondary importance in our evaluation. These include:

* support for anonymous secured access (to ensure integrity of results)
* implementation impact
* deployment impact
* time to market
* stronger reorder / replay protection

2.3 Operational Scenarios

How well proposals satisfy the goals described above can be evaluated by looking at specific operational scenarios or use cases. Ones mentioned on the mailing list include:

* deploying a new device, such as a router
* adding access for a new user
* revoking access for a user
* changing a user's secrets

In looking at each of these cases, one must consider what steps are needed with classic USM, which steps under the proposal under evaluation would be needed anyway (in order to support, for example, a CLI), how much of the classic USM work could be avoided under the proposal, and what additional configuration is needed to support the proposal.

2.4 Proposals Considered

This section provides a brief review of USM, and then gives an overview of each of the proposals.

2.4.1 User Security Model

The User-based Security Model [RFC3414] is a simple shared secret scheme. The operator configures secrets in the SNMP engines used by management applications. (Management applications include things traditionally considered "managers" and "agents".) These secrets may be "localized keys", computed from a passphrase known only to that user. These independent authentication and encryption keys are used to secure communication. Traffic may be authenticated, or authenticated and encrypted. The protocol also supports unauthenticated unencrypted exchanges, but they are not of interest here.

```
+------------------------+                +------------------------+
|        Manager         |                |        Managed         |
|        Computer        |                |        Device          |
| +--------------------+ |                | +--------------------+ |
| |    SNMP Engine     | |                | |     SNMP Engine    | |
| |                    | |                | |                    | |
| |                    | | <------------> | |                    | |
| |          +-------+ | |                | | +-------+          | |
| |          |  USM  | | |                | | |  USM  |          | |
| |          +-------+ | |                | | +-------+          | |
| +--------------------+ |                | +--------------------+ |
+------------------------+                +------------------------+
```

Figure 1

With USM security, the USM is part of the SNMP implementation. Messages to be protected are passed through the USM for transformation (encryption and/or authentication). When protected messages are received, they are passed through the USM for processing (decryption and/or authentication). The SNMP engine must keep track of whether a message had been encrypted and authenticated in order to make access control decisions.

As previously noted, this scheme has two disadvantages. First, even though all the localized keys for a given user in an administrative domain may be generated automatically from a single passphrase, device and user churn can still make it difficult to configure. Secondly, the particular message protection scheme, by design, does not provide replay protection inside a 150 second window. Being able to perform management operations while the underlying protocols were experiencing packet loss, duplication, or re-ordering was considered more important than protecting against replay attacks within a 150 second window, particularly since such attacks can be blocked by using a TestAndIncr [RFC2579] object to protect set requests which could do damage if replayed within the 150 second window.

2.4.2 External User Security Model

The External User Security Model [I-D.kaushik-snmp-external-usm] replaces USM's key management but leaves the USM transport alone. EUSM assumes that an external key management process will be co-resident with SNMP engines, and will install the keys, as with IKE/IPsec.

```
+------------------------+                   +----------------------+
|        Manager         |                   |        Managed       |
|        Computer        |                   |        Device        |
|                        |                   |                      |
|            +---------+ |                   | +--------+           |
|            | Key     | |  Key establish    | | Key    |           |
|            | Mgmt    | |<----------------->| | Mgmt   |           |
|            +---------+ |                   | +--------+           |
|                 ^      |                   |     ^                |
| +---------------|----+ |                   | +---|--------------+ |
| |   SNMP Engine |    | |                   | |   | SNMP Engine  | |
| |               |    | |  Message traffic  | |   |              | |
| |               v    | | <---------------> | |   v              | |
| |          +-------+ | |                   | | +-------+        | |
| |          |  USM  | | |                   | | |  USM  |        | |
| |          +-------+ | |                   | | +-------+        | |
| +--------------------+ |                   | +------------------+ |
+------------------------+                   +----------------------+
```

Figure 2

In EUSM, the SNMP engines establish keys using the external key management system. Those keys are then passed to the USM, and create (or at least key) users (or pseudo-users). From that point on, ordinary USM message protection is used.

One advertised advantage of EUSM is tight integration with

pre-existing network AAA systems such as RADIUS and DIAMETER. The general picture is shown below: the key management processes coordinate with the AAA server to perform the key agreement exchange.

```
                                +---------+
                                |   AAA   |
                      --------> | Server  |<--------|
                      |         +---------+         |
                      |                             |
                      |                             |
   +------------------|-----+                   +---|-----------------+
   |        Manager   |     |                   |   |    Managed      |
   |        Computer  |     |                   |   |    Device       |
   |                  v     |                   |   v                 |
   |            +---------+ |                   | +--------+          |
   |            | Key     | |  Key establish    | | Key    |          |
   |            | Mgmt    | |<----------------->| | Mgmt   |          |
   |            +---------+ |                   | +--------+          |
   |                 ^      |                   |     ^               |
   | +---------------|----+ |                   | +---|-------------+ |
   | |   SNMP Engine |    | |                   | |   | SNMP Engine | |
   | |               |    | |  Message traffic  | |   |             | |
   | |               v    | | <---------------> | |   v             | |
   | |          +-------+ | |                   | | +-------+       | |
   | |          |  USM  | | |                   | | |  USM  |       | |
   | |          +-------+ | |                   | | +-------+       | |
   | +--------------------+ |                   | +-----------------+ |
   +------------------------+                   +---------------------+
```

Figure 3

It's important to recognize that this is not an architectural advantage. Whether an AAA server is involved or not is completely orthogonal to the external key management design. One could use this general design with or without AAA integration. In order to reduce diagram complexity, we will not show AAA integration for the other approaches, although it is in general as easy to add *architecturally* to one approach as to the others.

The primary advantage of this scheme is that it has very little impact on the SNMP implementation. Because the implementation already has the ability to accept externally specified (manual) keys, it is straightforward to modify the design to accept externally specified keys which are generated via an automatic key management process. This is part of the reason why a similar design was chosen for IKE/IPsec. EUSM also has the desirable feature of integrating very well with RADIUS, desirable due to its large market presence. An additional advantage, as well as a limitation of this approach, is

the extent to which it reuses USM, since at leaves USM's anti-replay mechanisms intact.

This approach has several drawbacks. The first is that it inherits the communications security limitations of USM with respect to replay attack. The second is that it because the integration of the external key management module with the SNMP implementation is relatively loose, it can make policy setting confusing. In particular, it could be difficult to coordinate users between various processes. The coordination of permissions could likewise be a problem, but VACM coordination is out of scope of this effort, and none of the other proposals address this issue, which has been a substantial problem with IKE/IPsec. An additional consideration is that it's not clear how this could be used with Kerberos, at least as Kerberos is normally used.

A secondary problem is that it can be difficult for the key management system to give the USM and the rest of the SNMP implementation information about users. Looking ahead to the integration of access control configuration, the ability to reuse the existing key management interfaces starts to be inadequate. However, none of proposals do anything to address this, since almost all the user-specific information other than keys belongs to VACM. This proposal's way of addressing policy integration by providing the user/group mapping for VACM seems to be a good tradeoff, giving dynamic authentication and integration with existing access control without excessive overhead.

2.4.3 Session-Based Security Model

The Session-Based Security Model [I-D.hardaker-snmp-session-sm] addresses the first disadvantage by replacing the USM entirely. The new security model (SBSM) is an integrated session establishment and messaging protocol. When two network entities wish to communicate initially, SBSM performs a key agreement handshake to establish a session. From then on, messages are protected inside of SNMP. That is, the ciphertext and/or MAC data is encapsulated inside of an SNMP message, just as with USM and EUSM.

```
+----------------------+          +----------------------+
|       Manager        |          |       Managed        |
|       Computer       |          |       Device         |
| +------------------+ |          | +------------------+ |
| |   SNMP Engine    | |          | |    SNMP Engine   | |
| |                  | |          | |                  | |
| |                  | | <------> | |                  | |
| | +-----+ +------+ | |          | | +------+ +-----+ | |
| | | USM | | SBSM | | |          | | | SBSM | | USM | | |
| | +-----+ +------+ | |          | | +------+ +-----+ | |
| +------------------+ |          | +------------------+ |
+----------------------+          +----------------------+
```

Figure 4

This design allows the tightest coupling between the security system and the rest of the SNMP implementation. Authentication information, such as user names and session lifetime constraints, can be easily passed between them. This approach has the advantage of providing an extensible architecture and making it possible to integrate almost any authentication or privacy mechanism.

The major disadvantage of this design is that because it is so tightly coupled to the SNMP implementation, it may require a fair amount of reinvention. Indeed, the current SBSM design is a completely new security protocol. In principle, one could use an existing protocol such as IKE or TLS, but encapsulate all the message traffic in SNMP in the same way as the current SBSM, but it's not clear that that design would be superior to TLSM (below). In any case, going with SBSM would require a careful evaluation of the security protocol.

An open issue here is how to tie whatever identity the this security model has for a user with the security name needed by VACM to do the user-to-group mapping. If these need to be pre-configured in VACM, the value is diminished.

2.4.4 Transport-Layer Security Model

The Transport Layer Security Model [I-D.schoenw-snmp-tlsm] simply rehosts all SNMP communications over a new secure transport. The obvious choices here are TLS (for TCP) and DTLS (for UDP). However, from an architectural perspective, any self-contained generic channel security mechanism (such as IPsec or SSH) would also be fine. This produces a picture like the one below.

```
+-------------------------+               +-------------------------+
|        Manager          |               |        Managed          |
|        Computer         |               |        Device           |
| +---------------------+ |               | +---------------------+ |
| |    SNMP Engine      | |               | |     SNMP Engine     | |
| |                     | |               | |                     | |
| |                     | | <- - - - - -> | |                     | |
| | +------+ +-------+  | |               | | +-------+ +------+  | |
| | | TLSM | |  USM  |  | |               | | |  USM  | | TLSM |  | |
| | +------+ +-------+  | |               | | +-------+ +------+  | |
| +----^----------------+ |               | +-----^---------------+ |
|      |                  |               |       |                 |
| +----v----------------+ |               | +-----v---------------+ |
| |  Security layer     | | <-----------> | |   Security layer    | |
| +---------------------+ |               | +---------------------+ |
+-------------------------+               +-------------------------+
```

Figure 5

With the TLSM architecture, there is a new security model, the TLSM, but it's effectively a shim. All the heavy lifting is done by the generic security layer/protocol. The TLSM's only job is to provide the user name and security level to the security layer and to collect the per-message security properties. The downside to this is that the question of how the security layer's user names get mapped to security names for VACM is left open. This can be addressed either by preconfiguration, which defeats the purpose of the proposal, or by borrowing something like the EUSM proposal's mechanism for emulating the security name to group mapping for VACM.

From the network perspective, SNMP traffic secured with TLSM looks like every other kind of traffic secured with the underlying security layer. The only indication that it is SNMP is (potentially) the choice of port number.

TLSM allows reuse of standard communications security tools (e.g., TLS) while allowing a reasonable amount of coupling between the security layer and the SNMP implementation. However, it will likely be more work to integrate than EUSM and allow a less rich coupling than SBSM.

3. Recommendations

The evaluation team was unable to identify a clear winner among the three proposals. Each of the documents contained some of the elements needed for a complete solution. In part, this was due to the preliminary nature of the internet drafts that were available, and the different teams had fleshed out different elements of their

proposals in differing degrees of detail.

3.1 Architectural View

The evaluation team recommends the inclusion of an architectural perspective, like that provided by the TLSM proposal, though, of course, adapted to the technical specifics of the agreed solution. In particular, it's important to spell out where the various bits come from that are needed for the message processing model and security model abstract service interfaces, as well as what goes into message wrappers and any inter-layer redundancies. When authentication or encryption services are provided by other protocols outside SNMP proper, whether existing APIs provide these bits is an important consideration in gauging what the implementation effort would be.

The SNMPv3 architecture goes to some length to avoid talking about managers and agents, and is instead described in terms of SNMP engines and applications. This was motivated by practical considerations and implementation experience, not just architectural purity. The evaluation team recommends that this working group maintain this perspective.

3.2 Supported Security Infrastructures

The evaluation team recommends that the working group adopt an approach that can accommodate multiple security infrastructures concurrently. The EUSM proposal was clearest in this regard and went into the greatest detail, though clearly still greater detail will be needed for an interoperable specification. Note however that EUSM specifies integration with a specific authentication architecture, viz., AAA; we suggest the incorporation of a generic authentication architecture, with AAA as the case study. Either the EUSM specification or an independent specification can then describe how Kerberos (or any other authentication architecture) might be integrated. This is justified by the extreme diversity of security infrastructures currently in use, and the lack of compelling arguments justifying the selection of one to the exclusion of all others.

The evaluation team also recommends, in the interest of interoperability, that the working group select a single mandatory-to-implement mechanism. The evaluation team recommends RADIUS [RFC2865] for this purpose, due to its widespread use.

3.3 Integration with VACM

The SNMP architecture in general recognizes that how a user is

identified in a particular security model may need to be mapped to a protocol-independent identifier, allowing integration of different authentication schemes, for example. USM uses the usmUserSecurityName object to accomplish this. The evaluation team recommends that ways to support such mappings be investigated, since none of the proposals directly addresses this issue.

Dynamic authentication of users is not operationally sufficient, given how VACM works. Requiring security administrators to pre-configure the vacmSecurityToGroupTable [RFC3415] for dynamically authenticated users would defeat the whole purpose of doing dynamic authentication. Consequently, the evaluation team recommends the inclusion of something similar to the EUSM proposal's mechanism for conveying user-to-group mappings from the AAA-server-equivalent. This should not be confused with full-scale configuration of VACM, which is out of scope for this working group.

3.4 Sessions

All the proposals introduce something like a session. This allows the cost of authentication to be amortized over potentially many transactions.

3.4.1 Session Keys

The SBSM proposal's mechanism for session key establishment is attractive in explicitly addressing the perfect forward secrecy goal, at least for encryption keys. The same functionality could, however, be obtained using the TLSM approach as well.

Perfect forward secrecy guarantees that compromise of long term secret keys does not result in disclosure of past session keys. While this is a useful property, it comes at a fairly substantial computational cost, and in some cases additional message exchanges. There is no clear consensus in the evaluation team about this requirement.

3.4.2 Number of Security Levels per Session

The discussion on the mailing list and in face-to-face meetings led the evaluation team to recommend that a session should have a single user and security level associated with it. If an exchange between communicating engines would require a different security level or would be on behalf of a different user, then another session would be needed. An immediate consequence of this is that implementations should be able to maintain some reasonable number of concurrent sessions.

3.4.3 User Caching

The discussion of user cache lifetimes revealed that different types of interactions had different requirements. For example, ongoing polling was different from configuration requests. Consequently, the evaluation team recommends that cache lifetimes not be hard-wired. Lifetime could be communicated with the authentication results from the authentication server, with a configurable default in the managed device for those cases where the authentication server does not communicate a user cache entry lifetime.

3.5 Need for Initial Shared Secrets

Since RADIUS requires a shared secret to be established between the RADIUS client and server, it has the same out-of-box problem as USM, where one needs to establish the keys for the security administrator, who can then create the other users and VACM configurations. None of the proposals address this problem.

3.6 Reuse of Existing Security Protocols

The evaluation team considers designing and developing a new key management protocol for SNMPv3 an unnecessarily complex process and generally recommends reuse of existing security protocols where possible and appropriate. In other words, if an existing protocol is sufficient for the task at hand, the WG's energies are better spent elsewhere in the design of the overall solution for SNMPv3 security.

3.7 Conclusion

We conclude that neither of the three proposals matches all recommendations. On the other hand, each of them has one or more desirable properties that others might draw on to improve their original designs. It is quite tempting to conclude that the protocols be "merged" to create a single ISMS protocol. However, that would be ambiguous and would delay the process further.

The evaluation team concludes that the EUSM architecture would be the right direction for the ISMS WG. However, a number of aspects of the EUSM design need moderate to substantial revision. In the following, we first describe the components of the design that we consider most attractive, and then list the components that need to be revised and suggest components from other proposals as examples as appropriate:

- o EUSM keeps the current USM model intact.
- o EUSM integrates well with the AAA architecture. The evaluation team has the following recommendations to improve this interaction.

  * First, as noted earlier, where possible the design should not distinguish between agent and manager, following the SNMPv3 architecture.
  * Next, consider the possibility of using the AAA architecture or any external authentication infrastructure to establish shared secrets a la 802.11i architecture. Specifically, one of the problems with USM is that between n SNMP engines, there might O(n^2) pre-shared keys. The use of a centralized architecture will help reduce these keys to O(n), with each engine only authenticating to a common entity external to the SNMP world. As necessary, any two engines engaging in secure communication can establish a common key between them, and generate session keys as required by running Bellare-Rogaway's "entity authentication and key distribution" protocol [EAKD]. This will help reduce the number of interactions between the SNMP engines and the AAA server.
  * The evaluation team recommends that the eventual EUSM architecture be generic enough to support Kerberos and other authentication architectures.
- EUSM is also quite inline with the consensus in the evaluation team that, as much as possible the ISMS protocol should be reusing existing protocols.
  * We recommend the EUSM specification clearly identity the work in progress protocols that they use, so that the ISMS WG is aware of the dependencies on, say PANA, PEAP to name a few.
  * The other consideration is that the overall architecture once complete should be evaluated thoroughly from a security point of view. It is very easy to put together two independently secure protocols and open an avenue for a MiTM attack as shown by the compound binding attack.
  * We have several specific concerns (listed below) about some protocol choices in EUSM, which should be evaluated and justified with analysis and/or WG consensus as applicable.
    + There is a reference to CBC-DES in Section 3.5.1. Perhaps it is a typo.
    + EAP-GTC is suggested as the inner EAP method for client authentication. Q: Shouldn't the inner EAP method need to be a key generating method for compound binding? Furthermore WLAN EAP method recommendations draft specifically excludes GTC as the inner EAP method (see [I-D.walker-ieee802-req]).
    + There should be a discussion on PEAP vs. other tunneled EAP methods, e.g., EAP-TLS, TTLS etc.
- As noted in Section 3.1, the evaluation team suggests that EUSM be revised to integrate well with RFC 3411 architecture as done by the TLSM specification (Please see Section 3.3 - 3.5 and Section 4 in the TLSM specification [I-D.schoenw-snmp-tlsm]).

4. Acknowledgments

Working group co-chairs Ken Hornstein and Juergen Quittek facilitated the meetings of the evaluation team, goading the team to stay focused and on schedule.

The working group charter provided text on requirements and goals.

5. Security Considerations

This document compares three different proposals that fix the problems associated with USM for SNMPv3 security. They are all architecturally distinct from each other, and have different security properties and potential security issues. The reader is referred to the security considerations section within the drafts describing the three proposals.

6. IANA Considerations

This document requires no actions by IANA.

All of the proposals evaluated herein would require IANA action if adopted by the working group, but in no case was this seen to present a significant obstacle. For example, all the proposals would require the allocation of a new value for SnmpSecurityModel [RFC3411].

7. Informative References

[EAKD] Bellare, M. and P. Rogaway, "Entity Authentication and Key Distribution", Advances in Cryptology, Crypto '93, Lecture Notes in Computer Science 773, 1994.

[I-D.hardaker-snmp-session-sm]
Perkins, D. and W. Hardaker, "A Session-Based Security Model (SBSM) for version 3 of the Simple Network Management Protocol (SNMPv3)", Internet-Draft draft-hardaker-snmp-session-sm-03, October 2004.

[I-D.ietf-secsh-architecture]
Lonvick, C., "SSH Protocol Architecture", Internet-Draft draft-ietf-secsh-architecture-20, December 2004.

[I-D.kaushik-snmp-external-usm]
Narayan, K., "External User Security Model (EUSM) for version 3 of the Simple Network Management Protocol

(SNMPv3)",

Internet-Draft draft-kaushik-snmp-external-usm-01,
February 2005.

[I-D.rescorla-dtls]
Rescorla, E., "Datagram Transport Layer Security",
Internet-Draft draft-rescorla-dtls-02, December 2004.

[I-D.schoenw-snmp-tlsm]
Harrington, D. and J. Schoenwaelder, "Transport Layer
Security Model (TLSM) for the Simple Network Management
Protocol version 3 (SNMPv3)",
Internet-Draft draft-schoenw-snmp-tlsm-01, November 2004.

[I-D.walker-ieee802-req]
Stanley, D., Walker, J. and B. Aboba, "EAP Method
Requirements for Wireless LANs",
Internet-Draft draft-walker-ieee802-req-04, August 2004.

[RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768,
August 1980.

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7,
RFC 793, September 1981.

[RFC1492] Finseth, C., "An Access Control Protocol, Sometimes Called
TACACS", RFC 1492, July 1993.

[RFC1510] Kohl, J. and B. Neuman, "The Kerberos Network
Authentication Service (V5)", RFC 1510, September 1993.

[RFC2246] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0",
RFC 2246, January 1999.

[RFC2251] Wahl, M., Howes, T. and S. Kille, "Lightweight Directory
Access Protocol (v3)", RFC 2251, December 1997.

[RFC2401] Kent, S. and R. Atkinson, "Security Architecture for the
Internet Protocol", RFC 2401, November 1998.

[RFC2409] Harkins, D. and D. Carrel, "The Internet Key Exchange
(IKE)", RFC 2409, November 1998.

[RFC2418] Bradner, S., "IETF Working Group Guidelines and
Procedures", BCP 25, RFC 2418, September 1998.

[RFC2579] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J.,
McCloghrie, K., Rose, M. and S. Waldbusser, "Textual
Conventions for SMIv2", STD 58, RFC 2579, April 1999.

[RFC2865] Rigney, C., Willens, S., Rubens, A. and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, June 2000.

[RFC2903] de Laat, C., Gross, G., Gommans, L., Vollbrecht, J. and D. Spence, "Generic AAA Architecture", RFC 2903, August 2000.

[RFC3410] Case, J., Mundy, R., Partain, D. and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.

[RFC3411] Harrington, D., Presuhn, R. and B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks", STD 62, RFC 3411, December 2002.

[RFC3412] Case, J., Harrington, D., Presuhn, R. and B. Wijnen, "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3412, December 2002.

[RFC3413] Levi, D., Meyer, P. and B. Stewart, "Simple Network Management Protocol (SNMP) Applications", STD 62, RFC 3413, December 2002.

[RFC3414] Blumenthal, U. and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", STD 62, RFC 3414, December 2002.

[RFC3415] Wijnen, B., Presuhn, R. and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3415, December 2002.

[RFC3416] Presuhn, R., "Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3416, December 2002.

[RFC3417] Presuhn, R., "Transport Mappings for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3417, December 2002.

[RFC3418] Presuhn, R., "Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3418, December 2002.

[RFC3588] Calhoun, P., Loughney, J., Guttman, E., Zorn, G. and J.

Arkko, "Diameter Base Protocol", RFC 3588, September 2003.

Authors' Addresses

Uri Blumenthal
Intel Corporation
USA

Phone: +1 973-967-6446
Email: uri.blumenthal@intel.com

Lakshminath Dondeti
Nortel Networks
600 Technology Park Drive
Billerica, Massachusetts 01821
USA

Phone: +1 978-288-6406
Email: ldondeti@nortel.com

Randy Presuhn (editor)
R. Presuhn Consulting
San Jose, California 95120
USA

Phone: +1 408-268-2075
Email: randy_presuhn@mindspring.com

Eric Rescorla
RTFM, Inc.
2064 Edgewood Drive
Palo Alto, California 94303
USA

Phone:
Email: ekr@rtfm.com

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.