External User Security Model (EUSM) for version 3 of
the Simple Network Management Protocol (SNMPv3)
draft-kaushik-snmp-external-usm-00.txt

Status of this Memo

By submitting this Internet-Draft, I certify that any applicable
patent or other IPR claims of which I am aware have been disclosed,
and any of which I become aware will be disclosed, in accordance with
RFC 3668.

Internet-Drafts are working documents of the Internet Engineering
Task Force (IETF), its areas, and its working groups. Note that
other groups may also distribute working documents as Internet-
Drafts.

Internet-Drafts are draft documents valid for a maximum of six months
and may be updated, replaced, or obsoleted by other documents at any
time. It is inappropriate to use Internet-Drafts as reference
material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at
http://www.ietf.org/ietf/1id-abstracts.txt The list of Internet-Draft
Shadow Directories can be accessed at

Copyright (C) The Internet Society 2004.
Abstract

SNMPv3 provides a framework for user identity based authentication, privacy and granular access control. SNMPv3 aids secure manageability and overcomes one of major drawbacks in previous versions of the SNMP standard. There has been a significant lack of uptake for deployment of SNMPv3, and a number of organizations are still using SNMPv1/SNMPv2c. This is because SNMPv3 does not integrate well with administrative security schemes defined for existing management interfaces like the device command line interfaces. We believe this is because the SNMPv3 standard does not address the issue of management and distribution of the keying material for SNMP.

This specification defines a framework for SNMPv3 authentication and key distribution via an external AAA server. This kind of external authentication for SNMPv3 allows for common identity management for SNMP and CLI. AAA servers have been commonly used in 802.1x environments for distributing keys to Wireless Access Point (APs), this specification leverages work done in that area.

This specification defines a new security model for SNMPv3, the External User Security Model (EUSM). This EUSM model differs from USM in only one respect: it uses AAA protocols to obtain keying material for the user from the AAA server for achieving the security goals defined in [RFC3414].
Table of Contents

1. Introduction .................................................. 4
   1.2. Requirements Language. ................................. 4
2. Overview ...................................................... 5
   2.1. Design Considerations ................................ 7
3. Elements of the Solution ................................. 8
   3.1. EUSM Processing ......................................... 9
      3.1.1 EUSM Request Processing ............................ 9
      3.1.2 EUSM Trap Processing ............................... 9
      3.1.3 EUSM Inform Processing ............................ 10
   3.2. Modifications to SNMPv3 View Based Access Control (VACM) 11
   3.3. Security Context Setup ................................ 12
   3.4. PEAP Protocol Exchange ................................ 12
      3.4.1 PEAP Inner Method Recommendations ............... 14
   3.5. Key Distribution to SNMPv3 agents .................... 15
      3.5.1. Key Distribution using RADIUS ................. 15
   3.6. Key Caching ............................................. 17
      3.6.1. Handling Cache Synchronization Issues .......... 17
   3.7 Security Context Identification ....................... 19
   3.8 AAA Server Failover ..................................... 19
4. Security Considerations .................................... 19
5. IPv6 Considerations ......................................... 20
6. Acknowledgements ........................................... 20
7. References .................................................. 20
   7.1. Normative References ................................. 20
   Authors’ Addresses ......................................... 21
   Intellectual Property and Copyright Statement .......... 22
1. Introduction

This specification defines a new security model for SNMPv3, the External User Security Model (EUSM). The EUSM primarily addresses the same threats as defined in the SNMPv3 USM Specification [RFC3414], this model intends to achieve the same goals as USM, this model intends to remove the constraint, i.e. "The security protocol nor its underlying security mechanisms should depend upon the ready availability of other network services". This EUSM model will use AAA protocols to obtain keying material for the user from the AAA server for achieving the security goals defined in [RFC3414]. EUSM is exactly similar to USM in all aspects, with only one key difference, that security information is retrieved dynamically from the AAA Server, as opposed to having it permanently stored in the Local Configuration Datastore.

EUSM defines a framework for SNMPv3 authentication and key distribution via an external AAA server. External authentication for SNMPv3 allows for common identity management for SNMP and CLI. AAA servers have been commonly used in 802.1x environments for distributing keys to Wireless Access Point (APs), this specification leverages work done in that area.

This memo allows SNMPv3 authentication and privacy keys to be served from an external AAA server to the agents based on the user identity. Current schemes require SNMPv3 keys to be configured on a per agent basis. Using the AAA server provides the advantage of allowing a common identity to be used with/shared across all management protocols since the other network management interfaces like device CLI and XML access are capable of authentication with the same AAA server. Such sharing of common identities removes the need for their separate maintenance, and thereby reduces administrative overhead.

1.1. Specification of Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].
2. Overview

EUSM describes the setup of a security context between the SNMPv3 Manager and the AAA server prior to any communication between the SNMPv3 manager and the agents. The security context refers to a pair of data structures that contain shared state information, which is required in order that per-message security services may be provided to SNMPv3 packets. Examples of state that are shared between SNMPv3 Manager and the AAA server as part of a security context are cryptographic keys, and message sequence numbers. The security context specified here is different from the SNMP context described in [RFC3411].

As part of the establishment of a security context, the SNMPv3 Manager and the AAA server are mutually authenticated, and on successful authentication, will result in the generation of the master session key. Keys derived from the master session key will be used to protect the communication channel between the SNMPv3 manager and SNMPv3 agents.

The Extensible Authentication Protocol (EAP) will be used to setup the security context. Not all EAP mechanisms may be suitable for this exchange. The suitable mechanisms include PEAP, and EAP-TLS, with the former being the recommended mechanism. The EAP exchange and derivation of master session keys would happen as defined in [PEAP]. The use of PEAP ensures support for a wide variety of client authentication mechanisms including token-based and client-side certificate based authentication.

The SNMPv3 agent receives keying material from the AAA server via a AAA protocol, (e.g. RADIUS) using a request/response mechanism and it caches these keys for a short time interval; this time interval is supplied by the AAA Server in the response. The rest of this document assumes RADIUS is the AAA protocol, but any other AAA Protocol with similar characteristics could also be used. The AAA Server derives the per-agent "session" keys from the master session key by the process of SNMPv3 key localization. Each SNMPv3 manager will have to generate a unique security context with the AAA server for every user logged on to the management station. The message exchanges are depicted in figure below.

EUSM supports two schemes for the EAP exchange, the first scheme described in the figure below requires an EAP exchange between the SNMPv3 Manager and AAA Server using the RADIUS protocol to carry the EAP messages. This scheme requires the SNMPv3 Manager to have explicit knowledge of the AAA server.
The second scheme described in the figure below is similar to EAP exchanges used in 802.1x environments. The SNMPv3 Manager similar to a 802.1x supplicant has no knowledge of the AAA server and EAP exchange between the SNMPv3 Manager and the AAA server uses the SNMPv3 Agent to relay the EAP messages. The SNMPv3 agent acts as a 802.1x authenticator. The EAP exchange between the SNMPv3 Manager and the SNMPv3 Agent may use a protocol similar to the Protocol for Carrying Authentication for Network Access [PANA].
Both schemes described above only differ in the mechanism to setup the Security Context using the EAP exchange. All other aspects of this memo apply identically irrespective of the scheme used to setup security context. The rest of this memo uses the first scheme for illustration of EUSM, all the flows and processing will identically apply to second scheme as well.

2.1. Design Considerations

The design for EUSM are based on the following considerations:

1) The requirement of a scheme for SNMPv3 to integrate SNMPv3 authentication with an external AAA server to unify the approach for administrative security model for SNMPv3 and CLI.
2) The requirement for SNMPv3 to use strong authentication and key exchange and eliminate the need to use long term secrets to protect SNMPv3 packets.
3) The requirement for minimum number of changes, preferably none to the SNMPv3 packet format given the current status of the SNMPv3 standard.
4) The scheme MUST extend the capability of the AAA server to provide authentication, privacy and integrity protection for SNMPv3 agents.

5) The scheme MUST provide support for a variety of client authentication mechanisms including passwords, tokens and certificates.

6) The scheme MUST to optimize the key management scheme to scale to large numbers of agents.

7) The scheme MUST distribute keys based on valid security context.

8) The scheme MUST ensure that a separate AAA request is not generated for every SNMP request.

9) This scheme MUST be generic and should apply to existing and future AAA protocols.

SNMPv3 USM requires the users and user keys to be configured on every agent. Although the process can be automated via a SNMPv3 user management application, it does require a configuration management application that can configure ten of thousands of network elements in real time. Further, the USM model creates a parallel universe of SNMPv3 users which are completely different from the users that are used by the other management interfaces like CLI, TCL engine and XML based programmatic interfaces (XMLCONF).

3. Elements of the Solution

This memo specifies the use of EAP to establish a security context between a SNMP manager and an EAP-Server. Each SNMPv3 manager MUST setup a security context using EAP prior to any communication with the SNMPv3 agents. The master session key contained within the context will be the EAP MSK defined in [RFC3748]. The EAP-Server uses the master session key to derive a key unique to an agent using the key localization process described in [RFC3414]. The SNMPv3 Manager MUST also use the same key localization algorithm to derive session keys for agents from the master session key. These per-agent keys derived from the master session key will be referred to as "session keys" in the remainder of this memo. Session keys will be passed on from the AAA server to the SNMPv3 agents using a AAA protocol protected by a security context existing between the AAA server and the SNMP agent. This memo specifies session key delivery with RADIUS, but this can be adapted to other AAA protocols.

PEAP is the recommended method to setup the security context; it is possible to use alternate EAP mechanisms that are capable of establishing security contexts like EAP-TLS. The PEAP exchange mutually authenticate the SNMPv3 manager and the AAA server and result in the generation of a master session key (MSK).
3.1. EUSM Processing

The EUSM model will use AAA protocols to obtain keying material for the user from the AAA server for achieving the security goals defined in [RFC3414]. The EUSM will not modify the SNMPv3 message format defined in [RFC3414], there would only be a small change in the agent processing. In case of packets received with the security model set to EUSM. SNMPv3 agents will not use the USM Local Configuration Database (LCD); they would use the EUSM cache. The EUSM cache would be populated via the AAA request/response as defined in section 3.5.

3.1.1 EUSM Request Processing

SNMPv3 requests such as Get/Set/GetNext/GetBulk are initiated by the SNMPv3 Manager and the SNMPv3 agents respond to these requests. The SNMPv3 agent is authoritative for all these requests and the message is protected in USM using the user’s localized key at the SNMPv3 agent.

EUSM agent session keys will be used to protect all the requests. The SNMPv3 Manager MUST setup the security context before sending out any SNMPv3 request. The SNMPv3 Manager will derive the session key for the SNMPv3 agent before sending out the request, it will then use the session key to protect the SNMPv3 packet.

The SNMPv3 agent on reception of the SNMPv3 request will lookup it’s local cache of session keys to obtain any cached keys that might apply to the SNMPv3 request. In case there aren’t any cached keys, the SNMPv3 agent will send a AAA request to the AAA server to obtain it’s session key for the relevant security context. The AAA server authorizes the agent and returns the session keys. The agent proceeds with the processing of the packet on reception of the relevant session keys.

3.1.2 EUSM Trap Processing

The SNMPv3 agent is authoritative for the SNMPv3 Trap message, and the message is protected in USM using the user’s localized key at the SNMPv3 agent. The SNMPv3 Trap Messages will be protected by the EUSM session keys for the SNMPv3 agent.
The SNMPv3 agent will use the AAA exchange to obtain the SNMPv3 agent’s session key, this will be similar to the exchange during the processing of SNMPv3 Get/Set request at the SNMPv3 agent. The SNMPv3 agent will first look up its cache for its session keys, since the session keys might be cached on the agent as a result of a previous SNMPv3 Get/Set request. The AAA exchange will not be initiated in case the SNMPv3 agent obtains the session keys from the cache.

The SNMPv3 Trap Receiver will generate the session keys for the sending agent before processing the SNMPv3 Trap Message. The SNMPv3 Trap Receiver MUST setup a security context on initialization, this is a pre-requisite for SNMPv3 agents to use EUSM session keys.

Implementation Note: EUSM recommends that SNMPv3 agents that frequently and/or urgently send SNMP notifications as Traps should consider re-retrieving the localized keys just prior to the expiry of their cache-time.

3.1.3 EUSM Inform Processing

The SNMPv3 Manager is authoritative for the SNMPv3 Inform message, and the message is protected using the user’s localized key at the SNMPv3 Manager. The SNMPv3 Inform messages will be protected by EUSM session keys for the SNMPv3 Manager.

All SNMPv3 exchanges except for the SNMPv3 Inform require the SNMPv3 agent to use its own session keys for protecting SNMP messages. The SNMPv3 Inform would require the SNMPv3 agent to use the SNMPv3 Manager’s session keys. This will require the EUSM flows to be reversed, prior to any SNMPv3 Informs are sent, the SNMPv3 agent MUST setup the security context and generate the master session key, this is the role typically played by the SNMPv3 Manager. The SNMPv3 Manager for Inform processing will acquire its session keys from the AAA server using the AAA exchange, similar to the SNMPv3 agent. The figure below illustrates the EUSM Inform Exchanges.
3.2 Modifications to SNMPv3 View Based Access Control (VACM) Processing

This specification would require a minor modification to the SNMPv3 VACM processing as defined in [RFC3415]. The current VACM specification has the vacmSecurityToGroupTable, this table maps a combination of securityModel and securityName into a groupName which is used to define an access control policy for a group of principals. The EUSM model will have not a Local Configuration Database and securityNames would be defined only at the AAA server, hence there would be no entries in the vacmSecurityToGroupTable for the EUSM model.

The AAA server MUST return the groupName for the user on successful authentication and the agent MUST not look up the vacmSecurityToGroupTable to retrieve the groupName for a given security Name. It is possible for the AAA server to return a groupName that has not been defined on the agent; in such cases, the vacmAccessTable would not have an entry for this group. The SNMPv3 agent MUST return with a noAccessEntry error for the request.
All other aspects of the SNMPv3 VACM would be applicable to this specification. Although the vacmSecurityToGroupTable would be empty for EUSM users, it would not effect other VACM tables that depend on the groupName. The vacmAccessTable has the groupName as of the indexes but the groupName is not dependent on instances in vacmSecurityToGroupTable, quoting the statement in [RFC3415], "Entries in the vacmAccessTable can use an instance value for object vacmGroupName even if no entry in table vacmAccessSecurityToGroupTable has a corresponding value for object vacmGroupName."

3.3. Security Context Setup

This specification uses an authenticated key exchange to establish keys between the AAA server and the SNMP manager. The key exchange mechanism is encapsulated using EAP messages; the use of EAP is based on a variety of design considerations including the support for multiple client authentication mechanisms and the ability to work with current and future AAA protocols.

Any EAP Mechanism can be used as long as it meets the following requirements:

1) Derive key material between EAP-Peer and EAP-Server
2) Export key material from the method (Extended Master Session Key) [RFC2748]
3) Establishes a security association identifier that is exported from the mechanism

The SNMPv3 keys are derived from Extended Master Session Key and are distributed to the agents as described in [KEYWRAP]. Not all EAP mechanisms may be suitable for this exchange. The suitable mechanisms include PEAP, EAP-TLS with PEAP being the recommended mechanism.

3.4. PEAP Protocol Exchange

This proposal relies on PEAP for session key derivation, though PEAP is the recommended method in this draft, alternate methods like EAP-TLS may also be used in place of PEAP. The details of the PEAP protocol have been specified in [PEAP], figure 2 illustrates a typical exchange between the SNMPv3 Manager and the AAA server to setup a security context using PEAP.
SNMP Manager               AAA Server
-------------------        -------------
RADIUS Access-Request/   
EAP-Message/Start ->      

RADIUS Access-Challenge/ 
<- EAP-Request/ 
EAP-Type=PEAP 
(PEAP Start, S bit set)

RADIUS Access-Request/ 
EAP-Response/ 
EAP-Type=PEAP 
(TLS client_hello)->

RADIUS Access-Challenge/ 
<- EAP-Request/ 
EAP-Type=PEAP 
(TLS server_hello, 
 TLS certificate, 
[TLS server_key_exchange,] 
[TLS certificate_request,] 
TLS server_hello_done)

RADIUS Access-Request/ 
EAP-Response/ 
EAP-Type=PEAP 
([TLS certificate,] 
 TLS client_key_exchange, 
[TLS certificate_verify,] 
TLS change_cipher_spec, 
TLS finished) ->

RADIUS Access-Challenge/ 
<- EAP-Request/ 
EAP-Type=PEAP 
(TLS change_cipher_spec, 
TLS finished)

TLS channel established  
(messages sent within the TLS channel)

RADIUS Access-Request/ 
EAP-Response/ 
EAP-Type=PEAP ->

RADIUS Access-Challenge/ 
<- EAP-Request/ 
Identity
RADIUS Access-Request/
EAP-Response/
Identity (MyID) ->

RADIUS Access-Challenge/
<- EAP-Request/
EAP-Type=EAP-GTC

RADIUS Access-Request/
EAP-Response/
EAP-Type=EAP-GTC
User-Name,
Password ->

RADIUS Access-Challenge/
<- EAP-Success

TLS channel torn down
(messages sent in cleartext)

3.4.1. PEAP Inner Method Recommendations

PEAP provides tunneled authentication, i.e. TLS is used to establish an "outer" protective tunnel, which identifies the server to the client using certificates. The outer tunnel is then used in the second step, when an authentication protocol (EAP method) is run through the TLS tunnel. PEAP supports all existing EAP methods to be used as the inner authentication protocol, this proposal does not require all EAP methods to be implemented.

Identity repositories for network managers and administrators would be in directory servers (including Active Directory) and token stores. This would mean that an inner EAP method that provides a Password Authentication Protocol (PAP) like authentication method would be widely used for this proposal. The EAP-GTC method does provide such a mechanism and since there is a TLS tunnel for protection, we can safely used EAP-GTC method for reusable passwords in addition to authentication via token passwords. EAP-TLS would be used to support client side certificate based authentication, i.e. environments that have deployed a PKI infrastructure.
3.5. Key Distribution to SNMPv3 agents

This memo defines the use of AAA transactions to pass on the keys derived from the master session key to the SNMPv3 agents. The proposal currently supports the use of RADIUS for this purpose. The master session key is localized to per agent session keys via the process of key localization as defined in [RFC3414].

The SNMPv3 agent as part of the SNMPv3 packet processing would obtain the keys from the AAA server, the SNMP agent would pass on the manager IP address and username to the AAA server to retrieve keys for the specific security context. The username and manager IP address are used to identify the security contexts and keying material, in the future the use of a security context identifier (SCID) is desirable. The AAA server MUST make sure that it is distributing the keys to the agent that is authorized to see those keys. This is ensured since the AAA server is administratively configured with the SNMPv3 Engine IDs for all the SNMPv3 agents that would be requesting keys.

The AAA server MUST respond back with key material, i.e. symmetric key and IV. The SNMPv3 agent MUST cache the SNMPv3 keys, the lifetime of the key material MUST be specified in the response from the AAA server.

The AAA request response would be secure by security mechanisms relevant to the AAA protocol, it MAY involve encryption of the keying material using pre-configured secrets.

3.5.1. Key Distribution using RADIUS

The RADIUS key deliverance mechanism described in [KEYWRAP] MUST be used to download keying material from Radius servers to SNMPv3 agents. The RADIUS exchange between the RADIUS server and SNMPv3 agent is illustrated in the Figure below.
Figure 3: EUSM with RADIUS key deliverance

The RADIUS Key attribute defined in the [KEYWRAP] is used without modification. The SNMPv3 agent while processing the SNMPv3 packet SHOULD request the RADIUS server for SNMPv3 keys relevant to that particular request, the username and SNMP Manager IP address would be utilized as security context identifier to determine the SNMPv3 keys relevant to the particular request. The SNMPv3 agent MUST construct a Key_Request and MUST include the Key attribute with the App ID field set to the value indicating that SNMPv3 is the application requesting the keys. The SNMPv3 agent MUST also include the username from the SNMPv3 packet from the Manager in the User-Name attribute and the IP Address of the SNMPv3 Manager must be included in the Calling-Station-ID attribute. The request must also contain a message authentication attribute defined in [KEYWRAP]. The RADIUS server on processing the Key_Request MUST reference the App ID field in the Key attribute to determine that the request represents an SNMPv3 key request from an SNMPv3 agent. The RADIUS Server MUST proceed to process the User-Name and Calling-Station-ID attributes and MUST obtain the appropriate master session key based on the username and Manager IP Address. The RADIUS server MUST then create localized keys using the SNMP engine ID for the agent and the Master Session Key, the SNMPv3 engine ID MUST be configured for every agent on the AAA server.
The Radius Server MUST then construct an Key_Accept and include the Key attribute, the Key attribute will carry the localized key, the Initialization Vector (IV), the Key Lifetime, Key ID, the KEK ID and App ID. The App ID included in the key attribute should be the same as the App ID received in the request. The Key Lifetime represents the duration of time beyond which the SNMPv3 agent MUST NOT cache the keys retrieved from the RADIUS server. The RADIUS Server MUST also include the SNMP-Protection-Type and SNMP-Group-Name attributes, the SNMP-Protection-Type attribute specifies the SNMP encryption and authentication protocols to be used. HMAC-MD5-96 and HMAC-SHA-96 are the supported authentication protocols and CFB-AES and CBC-DES are the supported privacy protocols. The SNMP-Group-Name attribute specifies the VACM group for the user. The SNMPv3 authentication and privacy keys are placed in two separate Key attributes.

3.6. Key Caching

The key cache-time for the session keys cached on the SNMPv3 agents should typically be in the range of 90-180 seconds, this duration represents the typical time window for a network management job. This agent session key cache-time can be configured on the AAA server. The caching interval for the master session key would in the order of eight hours, this time interval can be configured on all authoritative agents that acquire the master session key, it can also be configured on the AAA server. The actual cache interval to be used is determined during the EAP key negotiation.

3.6.1. Handling Cache Synchronization Issues

The SNMPv3 agents MUST NOT cache the localized session keys for any longer than the duration of the session key cache time. In case the master session key changes during that time window, it would result in authentication failures for at least one request on all SNMPv3 agents with valid session keys derived from that master session key. The authentication failure would be the only way to detect such synchronization issues which is not appropriate.

This memo defines a method to handle such cache synchronization issues, this is done by maintaining a residual timer on the master session key. Every time the AAA server responds to SNMPv3 agent requests for session keys, it MUST save a residual timer on the master session key, the value of the timer will equal the session key cache time. There MUST be only one residual timer for a master session key and it MUST updated every time a localized key is distributed to agents.
The AAA server during security context setup MUST check on any valid residual timer on the previous master session key, it SHOULD proceed with the key exchange but MUST return the residual timer value along with the new master session key. The SNMP Manager MUST NOT use the new master session key until after expiry of the residual timer. This will prevent any synchronization issues between the per-agent session keys and master session keys.

Implementation Note

The residual timer will result in a small black out period for the SNMP Manager during which time the manager will not communicate with the agent. This really is not a significant problem since the only time the residual timer would be used is a case wherein the SNMP manager creates a new security context for a user within 90-180 secs of terminating an older security context for the same user. In such cases, network management operations performed by that user on the manager would be delayed for a few seconds. Most of the real time collection in network management applications is not tied to active user sessions and in such cases there will never be a residual timer set, a new security context for the network management system user would be created only on expiration of the older security context and the residual timer would never be set in such cases.

3.7. Security Context Identification

All references to "context" in this document refer to a security context which has no relationship to a SNMPv3 context; that is, a security context is independent of, and applies to all relevant SNMPv3 contexts.

This memo uses the IP Address of the SNMP Manager and the username to identify the security context. Alternate methods of security context identification like the use a Security Context ID are currently not possible, since that would involve changes to the SNMPv3 protocol. This method of security identification SHOULD be replaced with an appropriate Security Context ID in future revisions whenever possible. The use of IP Address of the SNMP Manager doesn’t result in any loss of functionality and this proposal is fully capable of supporting SNMP Managers that use dynamic IP Address assignment.
The design considerations for use of Manager IP Address in this proposal are.

1) AAA protocol servers like Radius and TACACS+ servers use IP Address based association to authenticate requests from clients using the shared secret, the shared secrets are configured per client or one for several clients and the IP Address is used by the AAA server to determine the appropriate shared secret. This would mean that the AAA server would have an IP Address based association with the SNMPv3 manager.

2) This proposal does not use the IP Address for any long term association, the IP Address of the SNMPv3 Manager is used by the AAA server to index the appropriate master secret. The master secret is typically valid for the duration of 8 hours and the association with a particular SNMPv3 manager IP address would last only for that duration. Moreover in event of a reboot of the SNMPv3 manager and reassignment of it’s IP Address, the SNMPv3 manager would create new master secrets for the new sessions initiated after the reboot. This makes this proposal capable of working with dynamic IP addresses.

3.8. AAA Server failover

Network elements currently allow for definition of multiple AAA servers to handle failover. In case, all AAA servers are unreachable, the SNMPv3 EUSM will not work. SNMPv3 Managers can still communicate with the device via USM, this allows SNMPv3 authentication to occur local to the device in case the network interface providing the transport to the AAA server goes down. This behavior is similar to other management interfaces like CLI that would be accessible via a local user database in case the AAA server is unreachable.

4. Security Considerations

PEAPv1 is susceptible to a man in the middle attack as described in [CompoundBinding]. The attack uses lack of mutual authentication during security context establishment, since the PEAP client is authenticated via the inner EAP method after the security context has been established. PEAPv2 [PEAP] mitigates the problem by creating a cryptographic binding between the inner and the outer protocols. This memo recommends the use of PEAPv2 to enable the use of the security enhancements.
The master session key used to derive localized keys for SNMPv3 agent SHOULD be completely random and has no co-relation to the user's password, the user’s password is never used to seed any key derivation. The process of localization defined in [RFC3414] derives per agent session keys from the master session key using a one way hash (MD5), this effectively ensures that the compromise of the per agent session key does not compromise the master session key.

The session keys are distributed to the agents from the AAA server by using RADIUS exchanges as defined in this memo. The security for this exchange would be based on the security provided by the underlying AAA protocol. The RADIUS Key Attribute is protected using the scheme defined in [KEYWRAP]. The AAA servers MUST make sure that agents are authorized to receive the keys they are requesting.

The usage of IPSec to protect these RADIUS and TACACS+ exchanges is recommended for better security.

5. IPv6 Considerations

Every mention of "IP Address" in this document is intended to provide equivalent support for IPv6 and IPv4. For example, in MIB syntax, the usage would be of InetAddress, not IpAddress.

6. Acknowledgements


7. References

7.1 Normative References


Authors’ Addresses

Kaushik Narayan
Cisco Systems, Inc.
125 Rio Robles
San Jose, CA. 95134 USA

Phone: +1-408-526-8168
EMail: kaushik@cisco.com

Keith McCloghrie
Cisco Systems, Inc.
170 East Tasman Drive
San Jose, CA. 95134-1706 USA

Phone: +1-408-526-5260
EMail: kzm@cisco.com
Joseph Salowey  
Cisco Systems  
2901 Third Avenue  
SEA1/6/  
Seattle, WA 98121 USA  
Phone: +1-206-256-3380  
EMail: jsalowey@cisco.com

Chris Elliot  
Cisco Systems, Inc.  
7025-1 Kit Creek Rd  
PO Box 14987  
Research Triangle Park, NC 27709-4987 USA  
Phone: +1-919-392-2146  
EMail: chelliot@cisco.com

Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF’s procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.
Full Copyright Statement

Copyright (C) The Internet Society (2004). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assignees.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgment

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