Kerberos Security Model
for SNMPv3

Rajaram Pejaver
Yiu Lee
Wes Hardaker
Ken Hornstein

November 2010
Introduction: Kerberos Security Model for SNMPv3

Why we need a new security model

Use cases driving this proposal

New requirements for security model

Proposed security model

Elements of Procedure

Next steps
About the authors

- Rajaram Pejaver, Comcast Cable
- Yiu L. Lee, Comcast Cable
- Wes Hardaker, SPARTA, Inc.
- Ken Hornstein, US Naval Research Laboratory

Previous Submission: draft-hornstein-snmpv3-ksm-00

Use cases driving this proposal

KSM for SNMPv3

- Untrusted Managed Devices
  - Examples: Modems, Set Top Boxes, Home Routers.
  - They can tampered with because they are physically located in customer’s homes.
    - It may be possible for an attacker to replace and spoof one of these devices.
  - Any globally sensitive data sent to them may be compromised.
    - Example: SNMP administrator’s SSH’s username and password.

- Low end Managed Devices
  - Examples: Modems, Set Top Boxes, Home Routers.
  - They may not have the math processing capabilities to do PK operations quickly.
  - They may not be able to maintain session state due to memory limitations.

- Large numbers of Managed Devices
  - Examples: there are millions such devices deployed in North America.
  - Devices will be periodically queried to retrieve device health & traffic load values.
  - Automated Managers will poll multiple devices per second.
  - Human administrators will access multiple devices while troubleshooting.

IETF 79: Beijing, China, November 10, 2010
USM and RADIUS Models

- USM has its own local table of users.

- RFC5592 + RFC5608 requires:
  - Use SSH to establish a secure session between Network Management Application to the SNMP Engine/RADIUS Client.
  - SSH may outsource the validation of a user’s password via a local RADIUS client to a RADIUS server.
  - Upon successful authentication, SNMP stack may receive the `groupName`.
  - This model requires the Network Management Application and SNMP Engine to form a SSH session.
Kerberos Security Model Supports

- Centralized Security Administration.
  - For authentication of Kerberos users (device administrators)
    - Authentication is handled without interaction with the managed device
  - For authorization of SNMP users (device administrators)
    - Addressed the same way as the I-D.ietf-isms-radius-vacm draft

- Strong Authentication (using two factor mechanisms.)
  - Enterprises typically require this for accessing sensitive Managed Devices.
  - Hardware security tokens sometimes require additional interactions with the user.
    - Not explicitly addressed by RFC5608, but could be extended.

- Convenience
  - Each subsequent device does not require user re-authentication.

- Efficiency
  - Does not require Managed Devices to save state between SNMP requests.
  - Does not require Managed Devices to perform excessive computations.
  - Minimizes the setup overhead before sending request.

IETF 79: Beijing, China, November 10, 2010
Proposed security model: KSM

Architectural placement of KSM

- This model is a peer to USM in the SNMP architecture.
- It uses VACM, and does not require any modifications to it.
- It uses VACM just like I-D.ietf-isms-radius-vacm does.
- It does not use or rely on any transport models.

Dependencies

- This model requires a Kerberos KDC server.
- It uses an Authorization Database for centralized authorization mappings.
  - Specifically, it maps securityName ➔ groupName.
  - For example:
    - Jack ➔ ConfigurationMgr; Jill ➔ Auditor; Joe ➔ Assistant;
    - Jack gets write access; Jill gets read access; Joe gets nothing.
  - The groupName may also be thought of as a role, permissions, …
  - This value of groupName must be recognized by the Command Responder.
- The KDC and Authorization Database will not be discussed here.
SNMPv3 Headers:

- `securityModel` must contain a new value indicating KSM.
- `securityParameters` must contain `ksmSecurityParameters`.
  - `ksmSecurityParameters` must contain Kerberos AP_REQ or AP_REP.
- `securityLevel` must contain `noAuthNoPriv`, `authNoPriv`, or `authPriv`.

Kerberos interaction and the authorization "in the cloud" are not discussed here.
ksmSecurityParameters

ksmSecurityParameters ::= SEQUENCE {
  ksmChecksumType INTEGER(0..2147483647),
  ksmChecksum OCTET STRING,
  ksmKerberosMsg OCTET STRING
}

- Message is encrypted when the securityLevel is authPriv
  ✓ scopedPDU is encrypted, resulting in a KRB_PRIV message.
- All messages are Integrity protected, except for noAuthNoPriv messages.
  ✓ The entire message, including the SNMPv3 header, is protected.
  ✓ Kerberos ‘checksums’ are actually keyed hashes, described in RFC 3961.

KSM notes

- Timeliness & replay detection are addressed by KRB_PRIV methods.
- securityNames for users and devices must be Kerberos Principal names.
  ✓ Example: joe@example.com
- Each request and response must carry a Kerberos message (AP_REQ/P).
Elements of Procedure

KSM for SNMPv3

- Procedure for Outgoing Requests
  - Command Generator contacts the KDC server to retrieve the Kerberos ticket. The ticket contains the *groupName* and *securityName*.
  - Command Generator hashes the SNMP’s PDU and creates the *ksmChecksum*.
  - Command Generator creates the *ksmSecurityParameters* and sends the request to the Command Responder.

- Procedure for Incoming Requests
  - Command Responder extracts the kerberos ticket, decrypts the PDU and extracts the *groupName* and *securityName* from the ticket.
  - Command Responder creates an entry in *vacmSecurityToGroupTable*:
    - *vacmSecurityModel* is KSM
    - *vacmSecurityName* is the extracted principle and realm (*joe@example.com*)
    - *vacmGroupName* is the extracted value
    - *vacmSecurityToGroupStorageType* is “volatile”
    - *vacmSecurityToGroupStatus* is “active”
Next steps

Status
- draft-pejaver-isms-kerberos-01 was published.
  - It needs more work.
  - Issues are open for discussion.

Demo of sample implementation.

Adopt KSM as a ISMS Working Group item

Discussion