

SNMP Evolution: Track the evolution from SNMPv1 to SNMPv3 to exploit the rich security features.

Overview

SNMP (Simple Network Management Protocol) was introduced to meet the growing need for a standard for managing Internet Protocol (IP) devices. SNMP provides its users with a simple set of operations that allows these devices to be managed remotely.

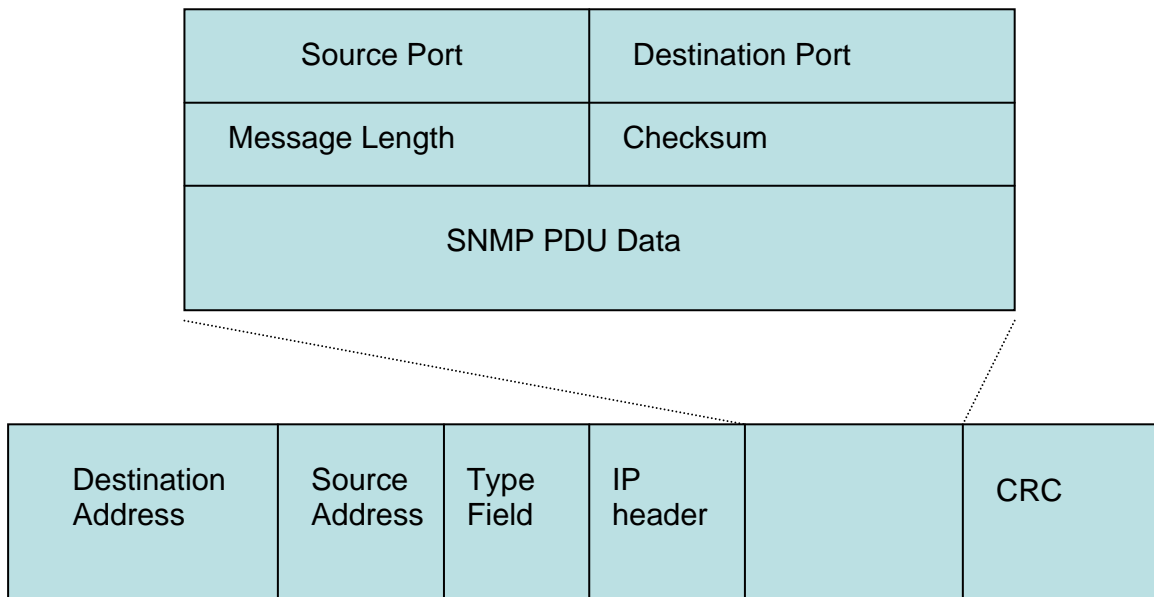
The core of SNMP is a simple set of operations that gives administrators the ability to monitor or change the state of some SNMP-based device. SNMP is usually associated with managing routers, printers, modem racks, power supplies and more. Any device running software that allows the retrieval of SNMP information can be managed. This includes not only physical devices but also software, such as web servers and databases. Network monitoring can also be done with the help of SNMP.

SNMP belongs in the TCP/IP (Transmission Control Protocol/Internet Protocol) application suite of protocols in the application layer and in most implementations, works over UDP (User Datagram Protocol). UDP is a connectionless service and offers no reliability and it is the application provider's responsibility to build reliability in the application layer. It is almost easy to guess why SNMP works over UDP: By their nature, management functions are used to manage and oversee network entities and these functions are not a priority for reliable operations when compared to the operations of the managed entities themselves and hence, loss of packets may be acceptable.

SNMP has evolved from SNMPv1 to which provided little or no security to SNMPv3 which provides extensive authentication and security features for all the three of the managing, managed and PDU (Protocol Data Unit) entities. History has that the other in between SNMP versions except for SMPv2c have been obsolete since they found little or no implementations. By understanding the evolution of SNMP through the major versions, both expert and novice SNMP users can exploit the rich features it offers today.

The following illustration shows the structure of an IP Datagram comprising of SNMP PDU (Protocol Data Unit), how it fits within the IP Packet.

PICTURE1: SNMP PDU With in UDP Datagram



Evolution

Protocol operations via SNMPv1 and SNMPv2c message wrappers support only trivial authentication based on plain-text community strings aptly called the “poor man’s password” and, as a result, are fundamentally insecure. When the SNMPv3 specifications for security and administration, which include strong security, reached full Standard status, the full Standard SNMPv1, and the experimental SNMPv2c specifications described in RFC (Request For Comment) 1901, were declared Historic due to their weaknesses with respect to security and to send a clear message that the third version of the Internet Standard Management Framework is the framework of choice.

In SNMPv1, a pairing of a SNMP community with a SNMP community profile is called SNMP access policy. An access policy represents a specific community profile afforded by the SNMP agent of a specified SNMP community to other members of that community. All administrative relationships among SNMP application entities are architecturally defined in terms of SNMP access policies. For example, the often used community profiles for an agent are {Public, Read Only}, {Private, Read/Write}. In this example community “Public” has “Read only” access and community “Private” has “Read/Write” access. Entities using community name “Private” can execute both get and set methods on the agent. Otherwise, there was no authentication support to ensure that the request originated from this entity or that the entity belonged to community “Private”.

The SNMPv1 framework is described in RFC 1157. The SNMP PDUs, which are equivalent to an abstract SNMP command, supported with this version are (1) GetRequest (2) GetNextRequest (3) SetRequest (4) GetResponse and (5) Trap.

The SNMPv2 Management Framework is described in RFC 1901 and coexistence and transition issues relating to SNMPv1 and SNMPv2 are discussed in RFC 1908. SNMPv2 provides several advantages over SNMPv1, including expanded data types (e.g., 64 bit counter), improved efficiency and performance (GetBulkRequest PDU), confirmed event notification (inform operator), richer error handling (errors and exceptions), improved sets, especially row creation and deletion, fine tuning of the data definition language.

However, the SNMPv2 Framework, as described in these documents, is incomplete in that it does not meet the original design goals of the SNMPv2 project. The unmet goals included provision of security and administration delivering so-called "commercial grade" security with (1) authentication: origin identification, message integrity, and some aspects of replay protection (2) privacy: confidentiality (3) authorization and access control and (4) suitable remote configuration and administration capabilities for these features.

A simple network sniffer can see the SNMPv1 and SNMPv2c PDU encapsulated within the UDP datagram which in turn is encapsulated within IP data packet. The text-based community string will be visible to any such sniffer, intentional or malicious.

It is easy to guess that applications that work over SNMPv1 and SNMPv2c are thus vulnerable to the following threats: modification of information, masquerade, message stream modification, and disclosure. SNMPv3 features for Authentication and Security promise to protect the entities from these threats.

The SNMPv3 management framework, addresses these significant deficiencies.

In the world of SNMP there were two basic kinds of entities: managers and agents. A manager is a server running some kind of software system that can handle management tasks for a network. Managers are often referred to as Network Management Stations (NMS). The agent is a piece of software that runs on the network device which is being managed.

The SNMP framework before the evolution of SNMPv3 consisted of (1) several managed nodes, each with an SNMP entity which provides remote access to management instrumentation, traditionally called an agent (2) at least one SNMP entity with management applications (typically called a manager) (3) a management protocol used to convey management information between the SNMP entities, and (4) management information. Managed elements are devices such as hosts, routers, terminal servers, etc., which are monitored and controlled via access to their management information.

The framework was architected with a protocol-independent data definition language and Management Information Base (MIB) along with a MIB-independent protocol. This

separation was designed to allow the SNMP-based protocol to be replaced without requiring the management information to be redefined or reinstrumented.

SNMP design is based on a modular architecture with evolutionary capabilities with emphasis on layering. As a result, SNMPv3 can be thought of as SNMPv2 with additional security and administration capabilities. Therefore, while most remains the same in SNMPv3, authentication and security features have been added to SNMPv3 by *enhancing* the underlying SNMP framework.

Following the enhanced Architectural guidelines, SNMPv3 supports USM (User-based Security Model) for Authentication. The USM utilizes MD5 (Message Digest Algorithm) [1] and the SHA (Secure Hash Algorithm) [2] as keyed hashing algorithms for digest computation to provide data integrity. The USM uses the Data Encryption Standard (DES) [3] in the Cipher Block Chaining mode (CBC) if disclosure protection (data security) is desired.

SNMPv3 supports View-based Access Control (VACM) for managing access related policies which facilitates creating and enforcing, on the managed entity side, to restrict access to various levels within a single context, for example a section of the MIB, for the same user. This is superior to the earlier community based policy in which there was no facility for context based access control. A manager was either able to “read” or “read and write” the entire set of MIBs supported by the agent.

| RFC | Purpose | Obsoletes |
|------------------------------------|---|------------------------------------|
| 1157 | SNMPv1 | 1098 |
| 1905 | SNMPv2 | 1448 |
| 1908 | Coexistence between Version 1 and 2 | |
| 3410, 3411, 3412, 3413, 3414, 3415 | Discussions on SNMPv3 Overview, Architecture and Sub-systems | 2570, 2571, 2572, 2573, 2574, 2575 |
| 3584 | Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework | 2576 |

The above tables summarizes the essential RFCs that define SNMPv1, SNMPv2c and SNMPv3.

New Architecture

The specifications of the Internet Standard Management Framework are based on a modular architecture. SNMPv3 framework is more than just a protocol for moving data. It consists of (1) a data definition language (2) definitions of management information

(the Management Information Base, or MIB) (3) a protocol definition, and (4) security and administration.

The architecture has been designed to meet the needs of implementations of:

- minimal SNMP entities with command responder and/or notification originator applications (traditionally called SNMP agents),
- SNMP entities with proxy forwarder applications (traditionally called SNMP proxy agents),
- command line driven SNMP entities with command generator and/or notification receiver applications (traditionally called SNMP command line managers),
- SNMP entities with command generator and/or notification receiver, plus command responder and/or notification originator applications (traditionally called SNMP mid-level managers or dual-role entities),
- SNMP entities with command generator and/or notification receiver and possibly other types of applications for managing a potentially very large number of managed nodes (traditionally called (network) management stations).

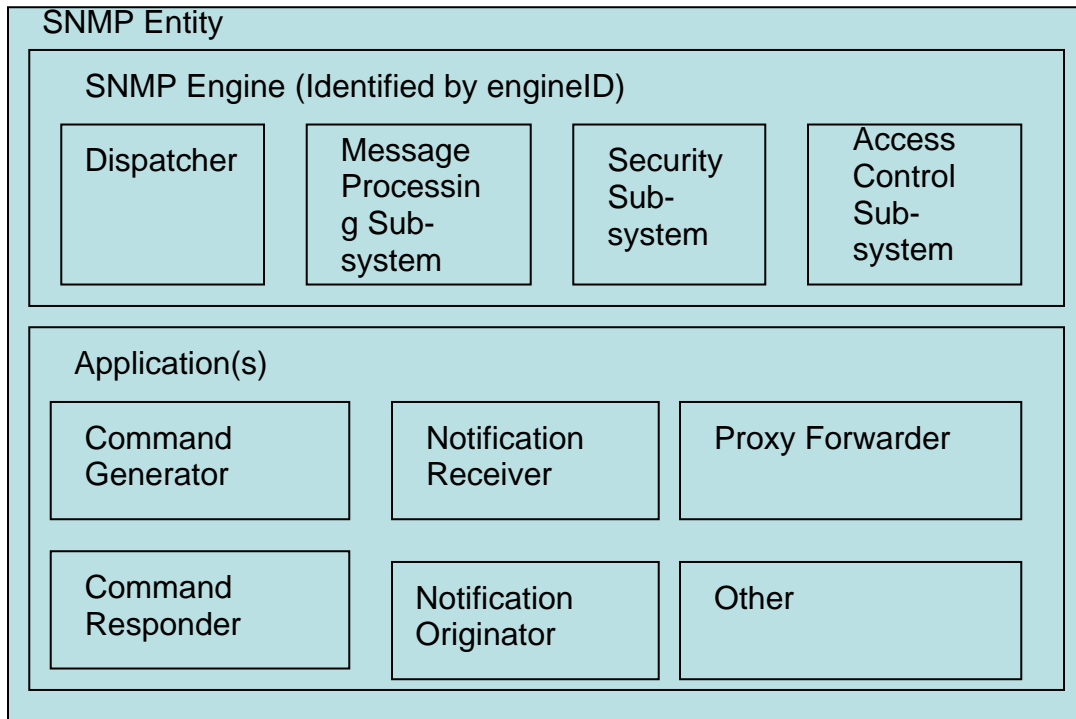
This architecture was driven by the following goals:

- Use existing materials as much as possible. It is heavily based on previous work, informally known as SNMPv2u and SNMPv2*, based in turn on SNMPv2p.
- Address the need for secure SET support, which is considered the most important deficiency in SNMPv1 and SNMPv2c.
- Make it possible to move portions of the architecture forward in the standards track, even if consensus has not been reached on all pieces.
- Define an architecture that allows for longevity of the SNMP Frameworks that have been and will be defined.
- Keep SNMP as simple as possible.
- Make it relatively inexpensive to deploy a minimal conforming implementation.

- Make it possible to upgrade portions of SNMP as new approaches become available, without disrupting an entire SNMP framework.
- Make it possible to support features required in large networks, but make the expense of supporting a feature directly related to the support of the feature.

The following illustrates an SNMP entity within the enhanced SNMPv3 Architecture.

PICTURE 2: SNMP Entity in SNMPv3 Architecture



An SNMP engine provides services for sending and receiving messages, authenticating and encrypting messages, and controlling access to managed objects. There is a one-to-one association between an SNMP engine and the SNMP entity which contains it. Within an administrative domain, an `snmpEngineID` is the unique and unambiguous identifier of an SNMP engine. Since there is a one-to-one association between SNMP engines and SNMP entities, it also uniquely and unambiguously identifies the SNMP entity within that administrative domain. Note that it is possible for SNMP entities in different administrative domains to have the same value for `snmpEngineID`.

The engine contains (1) a Dispatcher (2) a Message Processing Subsystem (3) a Security Subsystem, and (4) an Access Control Subsystem. There is only one Dispatcher in an

SNMP engine. It allows for concurrent support of multiple versions of SNMP messages in the SNMP engine. The Message Processing Subsystem is responsible for preparing messages for sending, and extracting data from received messages. The Message Processing Subsystem can potentially contains multiple Message Processing Models, for example one for processing SNMPv1 messages and another for SNMPv2c and yet another for SNMPv3. The Security Subsystem provides security services such as the authentication and privacy of messages and potentially contains multiple Security Models. The Access Control Subsystem provides authorization services by means of one or more of Access Control Models.

Applications include command generators, which monitor and manipulate management data, command responders, which provide access to management data, notification originators, which initiate asynchronous messages, notification receivers, which process asynchronous messages, and proxy forwarders, which forward messages between entities. These applications make use of the services provided by the SNMP engine. An SNMP entity containing one or more command generator and/or notification receiver applications (along with their associated SNMP engine) has traditionally been called an SNMP manager.

An SNMP entity containing one or more command responder and/or notification originator applications (along with their associated SNMP engine) has traditionally been called an SNMP agent. An SNMP context, or just "context" for short, is a collection of management information accessible by an SNMP entity. An item of management information may exist in more than one context. An SNMP entity potentially has access to many contexts. The combination of a contextEngineID and a contextName unambiguously identifies a context within an administrative domain; note that there may be multiple unique combinations of contextEngineID and contextName that unambiguously identify the same context.

Refer to RFC 3411 [4] for the detailed information on SNMPv3 Architecture.

Authentication and Security

USM comprises of Authentication and Security features. Authentication protects against masquerade. Security protects against message stream modification, and disclosure. The two primary and two secondary threats are defended against by the USM. The primary threats are: modification of information and masquerade; the secondary threats are: message stream modification and disclosure.

The USM utilizes MD5 and the Secure Hash Algorithm as keyed hashing algorithms for digest computation to provide data integrity (1) to directly protect against data modification attacks (2) to indirectly provide data origin authentication, and (3) to defend against masquerade attacks.

The USM uses loosely synchronized monotonically increasing time indicators to defend against certain message stream modification attacks. Automatic clock synchronization mechanisms based on the protocol are specified without dependence on third-party time sources and concomitant security considerations.

The USM uses the Data Encryption Standard (DES) in the cipher block chaining mode (CBC) if disclosure protection is desired. Support for DES in the USM is optional, primarily because export and usage restrictions in many countries make it difficult to export and use products which include cryptographic technology.

SNMPv3 architecture recognizes three levels of security (1) without authentication and without privacy (noAuthNoPriv) (2) with authentication but without privacy (authNoPriv) (3) with authentication and with privacy (authPriv).

For more discussions on USM refer to RFC 3414 [5].

Access Control

In SNMPv3, administration policies are enforced per View-based Access Control (VACM). The VACM can simultaneously be associated in a single engine implementation with multiple Message Processing Models and multiple Security Models.

It is architecturally possible to have multiple, different, Access Control Models active and present simultaneously in a single engine implementation, but this is expected to be **_very_** rare in practice and **_far_** less common than simultaneous support for multiple Message Processing Models and/or multiple Security Models.

The security rendered by a combination of USM and VACM can be best explained using an example. The complex example discussed here is from tri-lingual open source implementation of SNMPv3 called as net-snmp (<http://www.net-snmp.org>) which implements VACM for Access Control, implements USM via MD5/SHA and CBC-DES for security. This example **purposely ignores the implementation details and semantics** in order **to explain how policies, authentication and security can be enforced**.

Contents of configuration file used by the net-snmp Agent to enforce policies, authentication and security, placed in the \USR\SHARE\SNMP folder of the system where net-ANMP Agent is installed:

```
#  
# First, map the community name (COMMUNITY) into a security name  
# (local and mynetwork, depending on where the request is coming
```

```

# from):
#      sec.name      source      community
com2sec test          192.168.1.2  public
#
# Second, map the security names into group names:
#
#      sec.model      sec.name
group MyRWGroup      v1      test
group MyRWGroup      v2c     test
group MyRWGroup      usm     test
group MyROGroup      v1      test
group MyROGroup      v2c     test
group MyROGroup      usm     test
#
# Third, create a view for us to let the groups have rights to:
#      incl/excl  subtree      mask
viewall included .1              80
#
# Finally, grant the 2 groups access to the 1 view with different
# write permissions:
#      context      sec.model  sec.level      match read write notif
access MyROGroup "" any          noauth         exact all none none
access MyROGroup "" v2c         authNoPriv     exact all none none
access MyRWGroup "" usm         authPriv       exact all all  none

```

Contents of configuration file placed in the \USR\PERSIST\SNMP folder that contains the user information for net-snmp Agent:

```

#
# net-snmp persistent data file.
#
engineBoots 3
oldEngineID 0x800007e580d24600009b4eca41
rwuser test
createUser test SHA password DES 1234567890abcdef

```

In the above example, user “test” is a designated “read/write” user requiring authentication per USM model utilizing SHA by the net-SNMP Agent.. User “test” desires security utilizing DES key “1234567890abcdef”.

While in the “MyROGroup” context can exercise using “noauth” security level, access all read (Get, GetNext, GetBulk) methods and he has no access to write (Set) methods. While in the “MyRWGroup” context he can exercise using “authPriv” security level, access to all read (Get, GetNext, GetBulk) methods and write (Set) methods. In either contexts, he does not have permissions to solicit notifications (Traps).

For more information on VACM refer to RFC 3415 [6].

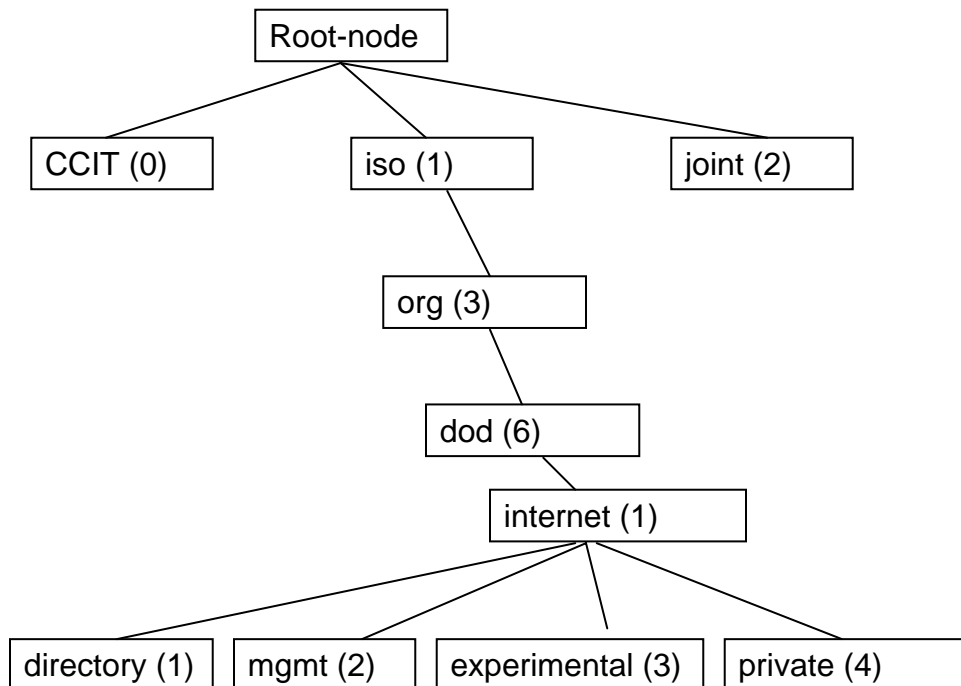
Management Information Base (MIB)

MIBs are defined SMI (Structure of Management Information) which is a DDL. During the SNMPv1 days, MIBs were defined using SMIv1. With the evolution on SNMPv2c, SMI evolved to SMIv2 supporting such data types as “Counter64”.

Surprisingly, the evolution of MIBs remains the same, except for the evolution of the DDL, SMI from SMIv1 to SMIv2. There are a large and growing number of standards-track MIB modules. As of today, there are more than 100 standards-track MIB modules with a total number of defined objects exceeding 10,000. In addition, there are an even larger and growing number of enterprise-specific MIB modules defined unilaterally by various vendors, research groups, consortia, and the like resulting in an unknown and virtually uncountable number of defined objects.

An object within a MIB is identified using the “dotted” notation using SMI. For example, the value of MIB2 (RFC 1155) system.sysDescr object is represented in dotted notation as "1.3.6.1.2.1.1.1" whose value, say for example, for host CRT, could be "CISCO Router 5500". Looking at the following SMI Object tree, we can deduce the name of the object as “iso.org.dod.internet.mgmt.1.1.1” The later part “1.1.1” can be further deduced using MIB2 (RFC 1158) definition for this object to “system.sysDescr.1” which makes the named object “iso.org.dod.internet.mgmt.system.sysDescr.1”.

PICTURE 3: SMI Object Tree



Tri-lingual Entities

It is not only imperative that NMS entities support tri-lingual SNMP (SNMPv1, SNMPv2C and SNMPv3) to interact with old devices whose vendors have not upgraded their firmware and manage them, it is may also be necessary for today's devices to support tri-lingual SNMP in the context of old NMS software who have not upgraded to SNMPv3 and discovery tools utilizing SNMP.

In general, management information defined in any MIB module, regardless of the version of the DDL (Data Definition Language) used, can be used with any version of the protocol. For example, MIB modules defined in terms of the SNMPv1 SMIv1 are compatible with the SNMPv3 Management Framework and can be conveyed by the protocols specified therein. Furthermore, MIB modules defined in terms of the SNMPv2 SMI (SMIv2) are compatible with SNMPv1 protocol operations and can be conveyed by it. However, there is one noteworthy exception: the *Counter64* datatype which can be defined in a MIB module defined in SMIv2 format but which cannot be conveyed by a SNMPv1 protocol engine. It can be conveyed by a SNMPv2 or an SNMPv3 engine, but cannot be conveyed by an engine which exclusively supports SNMPv1.

Please refer to RFC 3584 [7] which form the basis for discussions on tri-lingual entities for further information on co-existence issues.

Conclusion

SNMPv1 and SNMPv2c provide little to no security leaving the entities in communication and data vulnerable. SNMPv3 demonstrates successful elimination of these issues. By tracking the evolution of SNMP through the major version, the need for the security features and their purpose become evident in SNMPv3.

As a crossover strategy moving to SNMPv3 or tri-lingual entities, it is important that users deploying tri-lingual systems with insecure prior protocol versions such as SNMPv1 and SNMPv2c exercise sufficient due diligence to insure that configurations limit access via SNMPv1 and SNMPv2c appropriately, in keeping with the organization's security policy.

Just as they should carefully limit access granted via SNMPv3 with a security level of no authentication and no privacy which is roughly equivalent from a security point of view. For example, it is probably unwise to allow SNMPv1 or SNMPv2c a greater level of access than is provided to unauthenticated SNMPv3 users, e.g., it does not make sense to guard the front door with armed guards and unrestricted access through an open back door.

RFC and Article References:

[1] Rivest, R., "Message Digest Algorithm MD5", RFC 1321, April 1992.

[2] Secure Hash Algorithm. NIST FIPS 180-1, (April, 1995)

<http://csrc.nist.gov/fips/fip180-1.txt> (ASCII)

<http://csrc.nist.gov/fips/fip180-1.ps> (Postscript)

[3] Data Encryption Standard, National Institute of Standards and Technology. Federal Information Processing Standard (FIPS) Publication 46-1. Supersedes FIPS Publication 46, (January, 1977; reaffirmed January, 1988).

[4] D. Harrington and R. Presuhn and B. Wijnen: An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks, Enterasys Networks, BMC Software, Lucent Technologies, [RFC 3411](#), December 2002..

[5] RFC 3414, "User-Based Security Model (USM) for Version 3 of the Simple Network Management Protocol (SNMPv3)"

[6] RFC 3415, "View-based Access Control Model (VCAM) for the Simple Network Management Protocol (SNMP)"

[7] RFC 3584, "SNMPv3 Coexistence and Transition", describes coexistence between the SNMPv3 Management Framework, the SNMPv2 Management Framework, and the original SNMPv1 Management Framework.

Book References:

SNMP, SNMPv2, SNMPv3 and RMON 1 and 2, Third Edition, William Stallings ISBN 0-201-48534-6

Essential SNMP, Help for System and Network Administrators, First Edition, Douglas R. Mauro and Kevin J. Schmidt, ISBN 0-596-00020-0

AUTHOR BIO:

Uma Sundaram holds a B.S., in Computer Science and Engineering. Since 1992, she has worked for diverse type of employers in variety of networking-related projects, such as developing an SNMP based auto-discovery, semi-auto-configuration tool for managing multi-function networked office equipments, Windows NT based SNMP agent for Xerox ODPG, and enhancing SNMP based network monitoring and management for Ipswitch, Inc. Currently she operates a consulting company offering a variety of services based on SNMP, including foundation component products backed by SNMP based NMS techniques. She can be reached at: uma@rigconsulting.com.

This electronic document Copyright® is owned by Uma Sundaram. Reproduction rights are hereby granted only to parties to whom this article is submitted or parties who request in writing for reprint permission.