Security in SNMPv3 versus SNMPv1 or v2c

Ubizen AETHIS
Member of the Ubizen group

Introduction

Simple Network Management Protocol (SNMP) is an interoperable standards-based protocol that allows for external monitoring of the Content Engine through an SNMP agent.

An SNMP-managed network consists of three primary components: managed devices, agents, and management systems. A managed device is a network node that contains an SNMP agent and resides on a managed network. Managed devices collect and store management information and use SNMP to make this information available to management systems that use SNMP. Managed devices include routers, access servers, switches, bridges, hubs, computer hosts, and printers.

An agent is a software module that has local knowledge of management information and translates that information into a form compatible with SNMP: the Management Information Base (MIB). The agent can send traps, or notification of certain events, to the manager.

A manager is a software module that listens to the SNMP notifications sent by SNMP agents. The manager can also send requests to an agent to collect remote information from the Management Information Base (MIB).

The communication between the agent and the manager uses the SNMP protocol, which is an application of the ASN.1 BER (Abstract Syntax Notation 1 with Basic Encoding Rules), typically over UDP (for IP networks).

Version 1 (SNMPv1, described in RFC 1157) is the initial implementation of SNMP.

Version 2 (SNMPv2c, described in RFC 1902) is the second release of SNMP. It provides additions to data types, counter size, and protocol operations.

Version 3 (SNMPv3, described in RFC 2271 through RFC 2275) is the most recent version of SNMP. It became a full IETF standard, making SNMPv1 and v2c historical.

Threats of Network Security and their Relation to Network Management

SNMP version 1, or SNMPv1, has enjoyed unparalleled success as an interoperable management solution. However, it had multiple shortcomings, the most notable of which was its lack of strong security

This section describes the different kind of threats. Within the explanation of each threat it's also explained what kind of effects the threats have on the network management in general. That is, how the threats show up in the security of the network management.

- ✓ **Masquerading** means that an attacker succeeds to act in someone else's role and perform some tasks on behalf of the victim. In the security of network management at the moment, this is perhaps the most critical threat. One way to masquerade is to use **spoofing**. If a malicious attacker succeeds to act as an authorized manager, the doors are open for him to manage the network with the rights of the authorized manager: The attacker can do anything that is permitted to the manager that is the victim.

- ✓ **Modification of Information.** The threat of the modification of information means that some third party can intercept the transmission of the message and maliciously modify the in-transit message. Then the modified message is passed to the original receiver. Now the receiver of the message thinks that the message was sent by the trusted source while the contents of the message are changed. In network management, an authorized network manager can generate a valid management PDU. If an attacker succeeds to intercept the transmission, the whole PDU can be changed while keeping the authentication information unchanged. Of course, this is possible only if the PDU is not signed, nor encrypted.
- ✓ **Message Stream Modification** means that the stream of messages is modified somehow. This means that the messages could be reordered, or the messages could be recorded and replayed. The network management design originally aimed to connectionless management protocols. And since the most of the management protocols were designed to operate on connectionless transport services the message stream modification is a severe threat in network management. An attacker could for example record the valid management message that orders the router to shut down. Then, in the future, the attacker could use the captured message to perform the router shutdown whenever he wanted to do so.
- ✓ **Disclosure.** The threat of disclosure means that confidential information is leaked to the people who shouldn't see it. In network security in general, sniffing the traffic that is not encrypted is one way to do it. Also, in network management, some managment PDUs can carry some crucial information about the network and managed nodes itself. So, if an attacker spies the management traffic in a network segment, he could get some important information. That information could be used as the basis for other attacks, such as masquerading. A way to fight the threat of disclosure is to encrypt the messages.
- ✓ **Denial of Service** (DoS) means that some network service will become blocked somehow. Attacker could for example try to open TCP connections to a host continuously and that way block all the other connection requests. In network management this could mean that an attacker succeeds in blocking the flow of management protocol messages between the manager and the agent. In the network management, the DoS can also be a consequence if the other threats take place. For example, if an attacker succeeds to masquerade and act as the network manager, he can possibly give the shutdown command to a specific router. And this is, in fact, a denial-of-service type of threat taking place.
- ✓ **Traffic pattern analysis** is a threat where the information contents of the messages are ignored. Instead, the crucial information of the system is extracted from the usual patterns of the traffic flow. Both of these last two threats are hard to prevent.

(In-)security of SNMPv1

The basic SNMP has very primitive security functions. The only mechanism to authenticate a manager is by so called *community name*. The *community name* is used in defining management groups with differing access rights. That is, the community name is used to define which managers are allowed to submit get or set requests. The same *community name* mapping is used to define access policies for different managers. That is, some names may be restricted to operate only on some the areas of MIB while the others may have greater rights.

The SNMP community is locally defined at a node and the same name may be used at multiple nodes. When a manager wants to perform some kind of management task (get or set) it always has to present the *community name* that matches its need for access rights. In order to manage a selection of nodes the manager has to maintain a list of all the relevant community names.

As one might think, **this can't be considered a secure way to authenticate the user**. Anybody who knows a community name with powerful rights can act as a manager for a possibly large selection of nodes. And, in addition, compromising a community name compromises the security of the management in the network. The second problem with the security of the SNMP is the fact that there is no privacy. That is, there is no possiblity to encrypt the management message. When all the traffic flows through unsecured public network, nobody can tell if someone is spying the traffic. This leaves a second huge hole in the security of SNMP: anybody could listen to unencrypted UDP based SNMP traffic and catch the community name at a router, for example. This means that **eavesdropping** and **masquerading** are the most obvious threats to take place.

The weak authentication of the SNMP is bad enough by itself. Combining this with the **lack of privacy** make things even worse. The ease to sniff plain-text UDP traffic makes the weak authentication of the SNMP even weaker. And, that in turn makes the threat of sniffing even greater threat it normally would be. So, in sense, the two basic problems with the SNMP worse each other.

Due to the total insecurity of the SNMP, it is mostly used only for monitoring the agents. Actually, in most implementations, the SNMP "set" function is disabled just because it is ridiculously easy for an attacker to maliciously manage someone else's devices.

(In-)security of SNMP version 2

The SNMPv2 standardization wasn't successful. The standardization process resulted in three mutually incompatible standards (SNMPv2 party-based, SNMPv2u and SNMPv2*). Originally, the specification and designing of the SNMPv2 was initiated to enhance SNMP functionalities and the security was given some priority. A security scheme called "Party-Based Security" was introduced. Because the original SNMPv2 proposal was never really taken into any broader use, the Party-Based Security Model isn't introduced here. The standardization process of SNMPv2 was stuck with two competing proposals: SNMPv2u, SNMPv2* which both had a user-based security model. Unfortunately, a compromise was made and a proposal named SNMPv2c was standardized.

Why the SNMPv2 has a weak security? The answer is easy: The fruit of the standardization, the SNMPv2c, has no change to basic SNMP in terms of security - it relies completely on the familiar community strings.

Security of SNMPv3: a new message processing module

Using SNMPv3, users can securely collect management information from their SNMP agents without fear that the data has been tampered with. Also, confidential information, such as SNMP set packets that change a device's configuration, can be encrypted to prevent their contents from being exposed on the wire. Also, the group-based administrative model allows different users to access the same SNMP agent with varying access privileges.

Basically, the effective PDU, that is either SNMPv1-PDU or SNMPv2-PDU, is encapsulated in an SNMPv3 packet. This encapsulation provides security related functions on the level of message processing.

In SNMPv3, each entity – manager and agent – contains a single SNMPv3 engine to perform the message processing. When an application wants to send SNMP PDUs to the node in the network the following happens: The engine first accepts the SNMP datagram to be sent from SNMP application level, performs the appropriate security functions, encapsulates the PDU into an SNMPv3 message and then transmits the message to the network. When the engine receives an SNMPv3 message from the network, it performs the necessary decryption and authentication functions before passing the PDU to the SNMP applications.

User-Based Security Model (USM)

USM is the security model that implements the actual security services for authentication and privacy. Two different secret keys are needed, one for privacy (encryption key or privacy key, *privKey*) and the other for authentication (authentication key, *authKey*). These keys are not stored in the MIB of the node. Therefore they are not directly accessible through SNMP get- or set-functions.

Authentication and Integrity

For authentication of sender and checking the integrity of messages the USM supports two different authentication protocols, both of which are based on a widely used HMAC. HMAC-MD5-96 is a protocol where the secure hash function is MD5. The HMAC-SHA-96, on the other hand, uses SHA-1 for hashing. Inputs for both of the hash functions are the message to be sent and secret authentiction key of the user (*authKey*). Both hash functions produce an output, which is in both cases truncated to a message authentication code (MAC) of 12 octets. The calculated and truncated MAC is then appended to the message to be sent.

Upon reception the recipient does the following. The received message and the *authKey* are used as inputs for HMAC to calculate the MAC as was done when the message was sent. Now, if the calculated MAC is not the same as carried with the received message, the message is ignored. If, on the other hand, the MAC that was just calculated is the same MAC the received messages contained, the recipient can be sure about two things:

1. Integrity: The message couldn't be changed during the transmission. An attacker would have to know the secret *authKey* to change the message without being noticed.
2. Authenticity: To calculate the correct MAC the sender has to know the secret key. And, if the secret key is only known by the sender and the recipient, one can be sure that the message was sent by the authentic party.

Timeliness Verification of Messages

The security function of authentication doesn't prevent message delay or message replay attacks since actually there is nothing unusual in replayed messages. To make the SNMPv3 secure against this kind of flow manipulating attacks the USM has a timeliness mechanism. Actually, SNMPv3 demands that the messages must be received within reasonable time window.

The timeliness mechanism is based on two counters associated with each single SNMP engine: the snmpEngineBoots and snmpEngineTime. When an SNMP engine is installed, both of the two values are set to zero. After the SNMP engine has been started, snmpEngineTime is incremented once per second. Using a complex synchronization mechanism, an SNMP engine maintains an estimate of the values of time for each of the remote engines with which it communicates. These estimated values are placed in each outgoing message. The receiving management node's SNMP-engine then determines whether or not the incoming message is in the acceptable time window of 150 seconds. If the message doesn't fit the time window, it is simply ignored.

Privacy Through Encryption

For privacy, the USM uses Data Encryption Standard for ciphering messages. More precisely the CBC-mode of DES is used. The secret key needed for encryption is gained by taking the first eigth octets of the privacy key (*privKey*) assosiated with the user. The initial vector (IV) needed for the DES encryption algorithm is same as the last eight octets of privacy key. The encryption of the messages is optional. Like authentication key, the encryption key has to be set locally at the managed node.

Using secret key (symmetric) cryptography presents the SNMPv3 system additional challenges. The management of keys becomes a usability and security issue. If all the managed nodes have different secret keys the manager has to possess the same number of secret keys that there are managed nodes. The setup and management of keys quickly becomes a burden. If the management station is hacked and the secret keys compromised, all the nodes have to be reconfigured by hand. On the other hand one might like to use just one and the same secret key at all the managed nodes. The problem with this is that the compromising the only secret key compromises the security of the entire management system – that is, all the managed nodes. In USM, however, this problem is solved by utilizing a technique called Key Localization. The key creation, update and management are described in RFC-2274. The idea is to generate a unique key (called localized key) for each user-SNMP-engine pair by using user's password and snmpEngineID, which is the id of the target SNMP engine.

View-Based Access Control

In general, there exist various ways to manage the access control, that is, determing the access rights of a remote user to alter or view the local MIB. This means that the access control is a security function that is performed at the PDU level. The access control mechanism intended to be used with SNMPv3 is called View-Based Access Control (VACM), specified in RFC-2575.

The VACM is specified to determine the access rigths per group basis. This is different from the USM which specifies the authentication of users individially. In VACM each user has to be included in some group and the different groups can then be granted different security levels. This means that there might be, for example, a group of root managers, which have the ultimate control to alter all the parameters in all the managed nodes.

Additionally, there could be a group of minor, observing managers who would be granted only read permissions to certain parts of each local MIB.

The access rights are stored in different tables at the node and each of the tables is consulted to determine the access rights of the requesting manager. The procedure is based on the following concepts:

- **"Who"** is the subject of the operation and is defined by securityModel and securityName. This subject then belongs to one group at this SNMPv3 node.
- The contextName specifies **"where"** the desired management object can be found.
- **"How"** is the combination of securityModel and securityLevel and it defines how the incoming request was protected.
- The **viewType** specifies the type of access request **("why")**. The options are read, write, or notify access request.
- The object of the SNMP operation **("what")** is defined by the **variableName**.

The final access decision is made by comparing the variableName to the retrieved MIB view. If the variableName is found in the MIB view the access is granted.

Credits

Configuring Network Management: Copyright © 1992--2002 Cisco Systems, Inc.

The Security of Network Management: Toni Paila, Department of Computer Science, Helsinki University of Technology, 1999.

SNMP Research Contact information
(European Competence Center)

Ubizen AETHIS
Rue du Bosquet, 7 - B-1348 Louvain-la-Neuve - Belgium

Tel.: + 32 10 456 130 - Fax: + 32 10 456 155
Email: SNMP@aethis.ubizen.com - Website: http://www.aethis.com