Securing SNMP: A Look at Net-SNMP (SNMPv3)

Michael Stump

Securing SNMP: A Look at Net-SNMP (SNMPv3)

Michael Stump

Practical Exam for GSEC Certification
Version 1.4b, Option 1

Abstract

This paper addresses the many improvements, enhancements, and additions that comprise net-snmp, as well as the benefits of using SNMP to monitor network devices and computers. A discussion on the benefits of systems monitoring is included for relevance. Shortcomings of previous versions of SNMP are explained, and solutions to these faults are described in terms of net-snmp's capabilities.

The bulk of this practical focuses on the specific additions to SNMP that make net-snmp the ideal candidate for systems monitoring. The User-based Security Model is explained with regard to SNMP, and encryption is topically dealt with for completeness. Throughout the paper, various topics within network security and operation are discussed to emphasize the improvements with net-snmp. Overall, SNMPv3 provides the best of both worlds: ready access to system monitoring information, and sophisticated security.

Abstract....2
Introduction....4
What's Different about SNMPv3?....5
How does SNMPv3 Implement Encryption?....8
Getting Started with snmpconf....10
A Very Brief History of SNMP....10
The Simple Times....10
Where to Find MIBs on the Internet....10
Utilities for Gathering SNMP Data....11
Conclusion....11
Where to Learn More About NET-SNMP....11
Glossary of Terms....12
Resources and References....13

Introduction

Good systems administrators know the value of monitoring their systems. Early detection of problems is often critical to reduce, and perhaps altogether avoid lengthy and costly outages and unscheduled downtimes. In addition, many security incidents can be detected during the footprinting stage of the attack, thereby minimizing the effects of an intrusion. Of the many tools and services available to systems administrators to manage their systems, Simple Network Management Protocol (SNMP) is among the best. Its authentication method, encryption implementation, and ability to expose vast amounts of data to administrators are invaluable. Furthermore, when coupled with third-party utilities such as MRTG (Multi Router Traffic Grapher), SNMP can give administrators great insight into their systems.

SNMP is called “simple” because of the simple nature of the architecture. You need only have a host running SNMP and a workstation, normally referred to as the Network Monitoring System (NMS). The SNMP agent runs on the host, and requests are sent from the NMS. In a larger environment, the same structure works: one NMS managing data from numerous SNMP agents.

However, SNMP offers very little in terms of security. Version 1 of SNMP relies on a community string to protect data exchanged between two computers. This community string is passed in cleartext, which effectively voids the security measure. The data is also sent in the clear, so any intruder running tcpdump or a similar packet-sniffing application would be able to gather all the data obtained from the get request. For example, here is what tcpdump captured for an snmpget request for uptime (sysUpTime) from a server running SNMPv1:

```
13:55:47.406508 server.snmp > workstation.32769:
C=secretpassword GetResponse(32)  system.sysUpTime.0=1262054
```

Read left to right, the log entry shown above lists the following: time of entry, source host and port (shown as snmp, which is UDP 161), the destination host and port, SNMP community name, SNMP command, and the data requested (in this case, time in seconds). As you can see, the community name (*secretpassword*, as identified in the C= section of the log entry) is passed over the network in plaintext. The intruder would be able to easily grab this community name and begin to inventory all devices running SNMP. In a small network, it is possible that the administrator has used this same password elsewhere, possibly for a domain administrator account. The intruder would, in this case, have access to a root-level account. More maliciously, an attacker could use this captured data to set values on SNMP agents. With growing emphasis on network security, it is clear that version 1 cannot stand up to serious scrutiny.

Just over one year ago, the CERT® Coordination Center released a critical advisory related to many implementations of SNMPv1 (view the advisory here: http://www.cert.org/advisories/CA-2002-03.html). The implementations covered

in this advisory are vulnerable to denial of service attacks, and in worst cases root-level access to the agent. The vulnerabilities disclosed in this advisory re-emphasized the demand for secure network monitoring.

Many systems administrators would be surprised to know how much information about their systems is available via SNMP. Aside from system information, such as contacts for the server, a description of the hardware, the physical location of the server, and the server uptime (which, incidentally, is not a measure of how long the server has been up, but a measure of how long the SNMP agent has been running), SNMP can also expose detailed information about the software packages running on the host. For example, a snmpwalk on a Windows 2000 Server running SNMP (the SNMP service bundled with Windows is version 1) will return which services are running, what software has been installed, when it was installed, and where it is installed. It can also enumerate local users, shares, and disk volumes. With SNMPv1, there is no reasonable way to restrict the data that is available. Either a user has access to the MIB or not.

SNMP version 1 provides two security groups: public and private. The default configuration is to grant read-only access to the public group and read-write access to the private group. Obviously, this set up does not allow for accounting beyond these two groups. Since auditing is essential to thorough monitoring, the dual-group structure of this version leaves something to be desired. And again, the community name for the read-write group is passed in plaintext over the wire, so from a security viewpoint, there *is* no security.

SNMPv2 aimed to resolve these security issues. However, due to disagreement on the security model (View-based Access Control Module, or VACM) that was implemented in this version, the four key developers split up and SNMPv2 failed to become widely adopted. In fact, there are four distinct versions of SNMPv2: SNMPv2c, SNMPv2u, SNMPv2*, and SNMP-NG (NG is for Next Generation) (Wijnen). SNMPv2u was the first version to make use of the USM, but the level of complexity involved proved to be too great, and very few products adopted this implementation of SNMP (Miller, 238). SNMP-NG uses MD5 to verify the integrity of the communication and its source (Scambray, et al., 429). Two years after the split up, a working group was formed to begin drafting the RFC for SNMPv3 (Pras).

RFC1446 (Galvin and McCloghrie) describes the security protocols used by some of the variants of SNMPv2. In an attempt to provide authentication and encryption, the Digest Authentication Protocol (using MD5) and the Symmetric Privacy Protocol (using DES) were introduced. RFC3414 (Blumenthal and Wijnen) refines these protocols in respect to SNMPv3.

What's Different about SNMPv3?

SNMPv3 addresses these security concerns by adding three very important features: users, authentication, and encryption.

SNMPv3 uses the USM (User-based Security Module) for controlling access to information available via SNMP. In order to retrieve data from the SNMP agent, a username is required. This username is referred to as a securityName in the USM. Administrators may set up as many usernames as necessary. More importantly, administrators can grant these users access to specific parts of the available MIBs. For example, user SysAdmin may be granted access to the MIB-II portion of the tree, while user NTAdmin may be granted access to Microsoft portion. Here is what an snmpget request looks like when you provide a username and specify an authentication method:

```
localhost# snmpget –v 3 –u sysadmin –l authNoPriv –a SHA –A passphrase
```

The authentication is handled by specifying the type of authentication you are using (SNMPv3 supports SHA and MD5 to hash your password) and supplying a passphrase, or authKey. (Note that SHA is only available if you have built the package with OpenSSL installed on your machine). In the above example, the user has specified SHA for their authentication protocol and has supplied an accompanying passphrase. The combination of username, authentication method, and passphrase authenticates the user. SNMPv3 uses authentication to verify that the user sending the snmp command is authorized to access the information. Note that the –a switch is always paired with the –A switch. The same pattern is true for the encryption arguments –x and –X, which are discussed later in this paper.

User passwords are stored as machine-readable hashes in the snmp.conf file, which by default is in /var/net-snmp. Here's what a user's profile looks like in snmp.conf:

```
usmUser 1 3 0x800007e580eeb71668afbf223e 0x6c6f6e6700 0x6c6f6e6700 NULL
.1.3.6.1.6.3.10.1.1.2 0xd24700d82d2bc4e73cb85d3cfeb9ba13
.1.3.6.1.6.3.10.1.2.2 0xd24700d82d2bc4e73cb85d3cfeb9ba13 ""
```

Where is the passphrase, you ask? It has been replaced by a key that is generated from the password specified in the net-snmp-config command seen below. According to the man page for snmpd.conf, "This key is a localized key, so that if it is stolen it can not be used to access other agents" (Hardaker, et al.).

To create a user who has read-write access, you can use the following command:

```
Localhost# net-snmp-config –create-snmpv3-user –a "password" username
```

where "password" is the user's password and username is the user's login. (Note: the SNMP daemon must be stopped prior to creating users). After you enter this command, you will be prompted to provide the passphrases for authentication and encryption. Unfortunately, you are not prompted if your passphrase is too short; passphrases must be at least eight characters in length.

If you do assign a passphrase that is fewer than eight characters, you will only be notified of the error when you attempt to do an snmpget or snmpwalk with the user. Here's the command and error you will receive:

```
localhost# snmpget –v 3 –u user –l authNoPriv –a MD5 –A 2short –x DES –
X 2short2 localhost sysUpTime.0
Error: passphrase chosen is below the length requirements of the USM
(min-8).
Snmpget:  (The supplied password length is too short.)
Error generating a key (Ku) from the supplied authentication pass
phrase.
```

At this point it may seem that the authentication process in SNMPv3 is more trouble than it is worth because of the command line overhead. However, most of these switches can be coded into each user's snmp.conf file as default settings. Each command line option has a corresponding token that can be hardcoded into snmp.conf. For example, instead of passing the –u username argument to each snmp command, you can set defSecurityName to your username to avoid entering your username with each request. You can also set the default version of SNMP to use in your get/set requests (remember: only v3 can handle the user/passphrase portions of the requests, so you will want to set the default version to 3 unless you have reason to regularly get and set data from previous versions.) You can also enter default authProtocol and privProtocol values. Here's what your snmp.conf would look like if you added default values for all arguments:

```
defaultport
defversion  3
defsecurityname  user
defcontext
defsecuritylevel  authPriv
defauthtype  MD5
defauthpassphrase  passphrase
defprivtype  DES
defprivpassphrase  passphrase
```

(Note: the comments and instructions in the snmp.conf file have been removed to conserve space.)

With the values in this snmp.conf file set, this command:

```
Localhost# snmpget –v 3 –u user –l authPriv –a MD5 –A passphrase –x DES
–X passphrase target sysUpTime.0
```

Can be reduced to:

```
Localhost# snmpget target SysUpTime.0
```

Adding default values to your snmp.conf file will dramatically reduce the options you need to pass to the SNMP agent.

It is worth pointing out that the snmp.conf file will allow you to enter passphrases as default settings. This is a bad idea, as it potentially allows an intruder to grab all of your passphrases by viewing the contents of this file. To mitigate this risk, administrators are advised to grant read access only to the user. Granted, if an intruder has the ability to obtain your snmp.conf file you assuredly have larger problems. However, this potential risk is pointed out for completeness.

Another potential vulnerability is the SNMP user's shell history file. Since the passphrases used for authentication and encryption are not hashed at the command line, the user's shell history file will show the passphrases in plaintext. Since the history file is by default only available to root and the user, this risk is not critical.

Much attention has been directed to default settings and configurations in terms of security. Administrators contend that vendors need to ship their products with the security turned on rather than off. The idea is that administrators should only be required to enable what they need, as opposed to disabling everything that they do not need. For example, if SNMP shipped with a default configuration that granted read-only access to a community named "public", and the administrator did not change this, then anyone who has access to the agent could successfully retrieve information by using the default community name. This practice of setting the default values for the public and private community names has been well documented. In fact, in Hacking Exposed: Network Security Secrets and Solutions, four major network device manufacturers are listed along with their well-known community names. For example, Cisco uses public and private for their community strings (Scambray et al, 433). Clearly, this represents a substantial security risk.

To mitigate this risk, SNMPv3 contains a default snmp.conf file that will only grant users access to the system portion of the MIB tree. In order to take advantage of the capabilities of the package, the administrator is forced to read the entire configuration file and customize it before it will work. The effect is two-fold: the system administrator is forced to learn about the program he is managing, and SNMP requests will only return the information that the administrator has enabled.

How does SNMPv3 Implement Encryption?

Encryption is handled by using DES. DES may seem an unlikely choice, as it has been all but replaced by 3DES and, more recently, AES. However, with these newer encryption methods, the process of encrypting and decrypting data becomes more resource intensive. SNMPv3 uses DES to add a reasonable amount of security without bogging down the application and server with CPU-intensive cryptography. It is worth mentioning, and should be obvious, that

encryption cannot be enabled if authentication is not enabled. You need to authenticate the transmission before you can encrypt the data.

Earlier we saw what an snmpwalk request looks like in SNMPv1; the community name was passed over the wire in plaintext. Now that we have set up SNMPv3 with encryption, here is what tcpdump captures for an snmpget request:

```
18:16:13 workstation.54098 > server.snmp:  F=r U= E=  C= GetRequest(4)
[|snmp] (DF)
18:16:13 server.snmp > workstation.54098:  F= U= [|snmp][|snmp] (DF)
18:16:13 workstation.54098 > server.snmp:  F=ar U=ms [|snmp][|snmp]
(DF)
18:16:13 server.snmp > workstation.54098:  F=a U=ms [|snmp][|snmp] (DF)
```

Not much to see here, aside from the source and destination hosts. Let's try tcpdump with the –vvv argument, or very, very verbose logging:

```
18:50:14.015768 workstation.54349 > server.snmp:  |30|3e|02|01{ SNMPv3
|30|11{ |02|04|02|03|04|01F=r |02|01} { USM |04|10|30|0e|04|00|02|01B=0
|02|01T=0 |04|00U= |04|00|04|00} { ScopedPDU |30|14|04|00E=  |04|00C=
|a0|0e{ GetRequest(4) |02|04R=31030 [|snmp]} } }  (DF) (ttl 253, id
1459, len 92)

18:50:14.017344 server.snmp > workstation.54349:  |30|6a|02|01{ SNMPv3
|30|11{ |02|04|02|03|04|01F= |02|01} { USM |04|1e|30|1c|04|0d|02|01B=62
|02|02T=12503 |04|00U= |04|00[|snmp]} { ScopedPDU [|snmp]} }  (DF) (ttl
64, id 0, len 136)
```

We can see that a workstation has sent a request to a server on SNMP's default port (UDP 161). Aside from that, there is not much else to see. You can determine the version of SNMP in use (SNMPv3), and the USM section stands out, but the rest is hexadecimal data and packet information. In other words, no information is sent over the wire in plaintext, so intercepting and deciphering the data is not as easy.

SNMPv3 has the ability to log the source addresses of incoming requests. This allows the administrator to monitor who is pulling information from the SNMP agent. This feature is not new; it is, however, highly recommended to aid any type of incident response activity. For example, you may notice that many set requests have been made by an unauthorized workstation by looking at which IP addresses have been interacting with your SNMP daemon. It certainly never hurts to log as much information as possible, assuming you have the disk space.

To this end, I recommend running your SNMP daemon with the following arguments: -a –A –s and –d. Here is that each argument does:

- -a Tells the daemon to log the source of all incoming SNMP requests.
- -A Instructs the daemon to append information to the logfile, as opposed to truncating it.
- -s Forces snmpd to log to syslog (default logfile is snmpd.log).

-d Tells the daemon to dump all incoming/outgoing packets in hexadecimal format.

This level of logging gives you a lot of data to look over, but will aid you if you need to investigate a potential security incident.

Getting Started with snmpconf

To simplify the process of configuring your snmp.conf file, the developers of Net-SNMP have written a Perl script that walks you through the process. This script is named snmpconf and is included in the standard distribution of Net-SNMP. You need only supply a small amount of information to the script in order to have a working SNMP agent. However, to get the most out of the application, you will need to go through each menu in this script. The script can help you configure the three core configuration files: snmpd.conf, snmptrapd.conf, and snmp.conf. These files control the behavior of the SNMP daemon, the SNMP trap daemon, and other Net-SNMP applications, respectively.

A Very Brief History of SNMP

SNMP is the successful result of a joint UC-Davis and Carnegie-Mellon University project, which started in the early 1980's. It was also partially derived from the Simple Gateway Monitoring Protocol, or SGMP (Miller, 201). Several packages are still available, including UCD-SNMP and CMU-SNMP. Of course, Net-SNMP is the most active development project to date. The most recent release, version 5.0.8, was released on March 19th, 2003 (Hardaker et al.).

The Simple Times

This newsletter began as a means to share information related to implementing SNMP in various network environments. Although the newsletter has no set publishing schedule (the last edition is dated December 2002), it remains an excellent source of information on the history of SNMP.

Where to Find MIBs on the Internet

Before you can start using SNMP to monitor your network devices and applications, you will need to locate the MIB specific to your need. Many network applications have their own MIBs (RIM's BlackBerry Enterprise Server, for example, ships with a blackberry.mib that can be used to retrieve information on server usage, incoming/outgoing messages, etc.). Likewise, many network devices (for example, the Cisco Aironet 350 has built-in support for the Cisco Aironet MIB, which exposes configuration and usage data for the wireless device) have vendor-provided MIBs to allow administrators to collect and analyze performance data.

To locate a MIB for your device, visit the manufacturer's website. Most vendors will provide MIBs under the support section of their site. It is also probable that the MIB will be included with your installation CD. And of course, a quick search on Google will most likely turn up what you are looking for.

Utilities for Gathering SNMP Data

Phillipe Simonet's Getif is an excellent Win32 program for walking MIBs. Getif 2.2 will allow the user to view data from any SNMPv1 agent; it cannot handle the USM or encryption methods in version 3. Still, Getif remains a great tool for verifying that your SNMP agent is correctly responding to SNMP commands. Of course, the snmpwalk and snmpget utilities are great for testing, too, and they are included in the Net-SNMP package.

Tcpdump was used to capture the network traffic used in the examples. If you have not used it before, tcpdump is a packet capture utility that is included in most distributions of Unix and Linux. There is also a Win32 version called WinDump. Using tcpdump is a topic that warrants its own paper, so if you are interested in using this program please read the man pages, or visit www.tcpdump.org.

Once you have your agents up and running, you can use a program called MRTG to represent this data in graphical format. MRTG is a web-based application that polls SNMP agents at set intervals and graphs the results. For example, you could use MRTG and SNMP to monitor the network traffic inbound and outbound on any one of your agents. MRTG also maintains a history of up to one year, so you can view trends over time. For more information on using MRTG in this way, please visit the MRTG homepage at http://people.ee.ethz.ch/~oetiker/webtools/mrtg.

Conclusion

SNMP is a great way to monitor your network devices. Previous versions of SNMP provided an insecure way to access this data. Although attempts were made to enhance the security in the second version of SNMP, the enhancements proved to be more complex that the developers thought, and the protocol was not adopted as a result (Miller, 238). SNMPv3 addresses the security shortcomings with the addition of a user-based system for access control, a means to properly authenticate users, and a method for encrypting SNMP traffic between agent and host.

Where to Learn More About NET-SNMP

The bulk of the content of this paper is based on the wealth of information available at the NET-SNMP Project Home Page (Hardaker, et alia), which is listed below in the bibliography section. If you are interested in experimenting with NET-SNMP, I highly recommend visiting the site. The man pages are also a great source of information. There is also a SourceForge site located at http://sourceforge.net/projects/net-snmp. You are encouraged to read the FAQs thoroughly before you start posting questions to the email discussion list. However, if you need to ask a question, simply join by visiting this webpage: http://lists.sourceforge.net/lists/listinfo/net-snmp-users. Questions are answered by one of many developers, or by experienced users.

Glossary of Terms

Authentication – The method of verifying the identity of users or computers. Authentication ensures that users are who they say they are.

DES – Digital Encryption Standard. DES gave way to 3DES (Triple DES), which has become a widely accepted and used encryption scheme. 3DES has withstood decades of attempts to crack its algorithm. Daniel D. Houser has written an excellent history of DES, and this document is available in the SANS Reading Room at: http://www.sans.org/rr/encryption/DES.php.

Encryption – The process of converting plaintext into ciphertext, thereby making the text unreadable to unintended recipients. In regard to SNMP, data can be encrypted to prevent electronic eavesdropping of sensitive data. SNMPv3 uses DES to encrypt data.

Footprinting – The process of scouting a network for potential entry points and vulnerabilities. Footprinting is usually the first stage in a planned attempt to penetrate a network. It includes activities such as port scanning, DNS lookups, and war dialing. Chapter 1 of Hacking Exposed: Network Security Secrets & Solutions thoroughly discusses footprinting (see references for book information).

MIB – Management Information Base. A MIB is a collection of objects which describe an SNMP managable entity (Cikoski).
.

User-based Security Model – The default model for providing used-based security to SNMPv3. In short, users are defined and assigned various attributes that dictate the access level of that user. For a detailed description, please view the RFC at ftp://ftp.ietf.org/rfc/rfc2574.txt.

View-based Access Control Model – This is the default access control module for SNMP versions 1 and 2. It has been replaced by the USM for SNMPv3. For detailed information, view the VACM RFC at ftp://ftp.ietf.org/rfc/rfc2575.txt.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Resources and References

Blumenthal, Uri and Bert Wijnen. User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3). ftp://ftp.rfc-editor.org/in-notes/rfc3414.txt. December 2002. Online.

CERT/CC. CERT® Advisory CA-2002-03 Multiple Vulnerabilities in Many Implementations of the Simple Network Management Protocol (SNMP). February 12, 2002. http://www.cert.org/advisories/CA-2002-03.html. Online.

Cikoski, Tom. SNMP FAQ Index. 9 January 2003. http://www.faqs.org/faqs/snmp-faq/. Online.

Galvin, James and Keith McCloghrie. Security Protocols for version 2 of the Simple Network Management Protocol (SNMPv2). April 2003. http://www.ietf.cnri.reston.va.us/rfc/rfc1446.txt. Online.

Hardaker, Wes et alia. The NET-SNMP Project Home Page. 06 January 2003. http://www.net-snmp.org. Online.

Miller, Mark A. Managing Internetworks with SNMP. New York City, M&T Books/Henry Holt and Company, Inc. 1997. 208-239.

Oetiker, Tobias. MRTG: The Multi Router Traffic Grapher. 28 November 2002. http://people.ee.ethz.ch/~oetiker/webtools/mrtg.

Pras, Aiko. SNMPv2: Overview. April 28th, 1995. http://www.simpleweb.org/nm/research/results/presentations/pras/bellcore.pdf. April 2, 2003. Online.

Scambray, Joel, Stuart McClure and George Kurtz. Hacking Exposed: Network Security Secrets & Solutions. Second Edition. Berkeley, Osborne/McGraw-Hill, 2000.

Various. The Simple Times. December 2002. http://www.simple-times.org. Online.