



Multiple Vulnerabilities in SNMP

Guofei Jiang, Institute for Security Technology Studies (ISTS), Dartmouth College

For more than a decade, many network administrators have relied on SNMP, the Simple Network Management Protocol, to monitor and manage network devices. Now in its third release, SNMP has become the de facto standard for network management since its development in 1987. However, a recent report from the computer security watchdog CERT Coordination Center¹ indicates that vulnerabilities in many SNMP implementations have left the products of more than 100 vendors vulnerable to attack. Successful exploitation of these vulnerabilities could lead to unauthorized privileged access, denial of service attacks, or other undesirable behaviors.

How SNMP WORKS

SNMP is popular for network management because of its simplicity. Commonly used to manage network devices such as routers, switches, hubs, printers, workstations, and servers, SNMP employs only three general types of SNMP operations. *Get* requests retrieve management data from the device, *set* requests modify the remote device's configuration, and *trap* messages let a device send asynchronous notification and signal condition changes.

A network management system usually consists of two primary elements: a network management station (NMS) and SNMP agents. The NMS is the console through which an administrator performs management functions. Agents are

the entities that interface to the actual device being managed. The primary communication protocol in SNMP is UDP, the User Datagram Protocol. While SNMP agents listen on UDP port 161 to receive requests from the NMS, the NMS listens on UDP port 162 to receive asynchronous traps. Figure 1 shows a simplified SNMP architecture, where the dynamic port is assigned by the operating system. SNMP supports trivial authentication by using a community name, which serves as a password for either retrieving or modifying management data.

WHERE THE VULNERABILITIES ARE

Researchers at Finland's Oulu University have developed tests that reveal numerous vulnerabilities in various SNMPv1 implementations. Though they only tested SNMPv1, these vulner-

abilities likely exist in SNMPv2c and SNMPv3 as well.

The *Protos* research project² undertaken by the Oulu University Secure Programming Group (OUSPG) develops security test suites for a variety of protocol implementations. These test suites are generally used to analyze a protocol and produce messages that probe various design limits within the implementation. As a part of *Protos*, the OUSPG developed a SNMPv1 test suite to test weaknesses in several SNMPv1 implementations. The test packets can contain overly-long or malformed object identifiers and other combinations of exceptional values in various fields. The *Protos* test suite for SNMPv1 contains approximately 53,000 individual test cases.³

By applying the *Protos* SNMPv1 test suites to a variety of popular SNMPv1-enabled products, the OUSPG revealed the following SNMP vulnerabilities:¹

- *Trap handling.* Multiple vulnerabilities were found in how numerous NMSs decode and process SNMP trap messages.
- *Request handling.* The testing also revealed weaknesses in the way many SNMP agents decode and process SNMP request messages.

These vulnerabilities resulted from insufficient checking of SNMP messages as they were received and processed by an

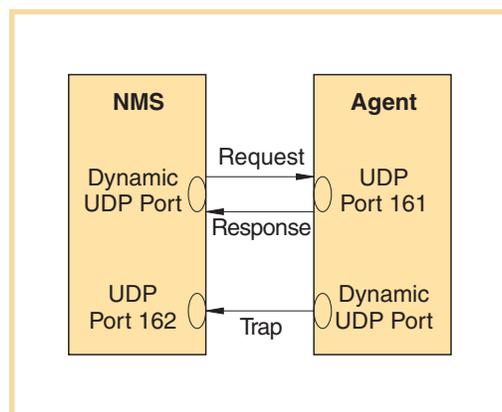


Figure 1. A simplified SNMP architecture

affected system. For products of different vendors, these vulnerabilities can lead to denial of service attacks, format string vulnerability, and buffer overflows.

INSECURE SETTINGS

Some of the vulnerabilities found by Protos do not require the SNMP message to use the correct community name, which make these vulnerabilities easily exploitable. Also, since UDP is a connectionless communication protocol, SNMP agents and trap-aware NMSs accept incoming requests and traps without any prior session setup. Most SNMP-enabled products ship with default community strings of “public” for read-only access and “private” for read-write access. The community name string is embedded within an SNMP message and transported across the network in plain text. Even if it is well configured, the community name string is still vulnerable to packet sniffing. Hackers can guess or sniff the community name if they have to use the community name to exploit these vulnerabilities.

SPOOFING

Network access control is also insufficient to block attacks on these vulnerabilities, because UDP source addresses can be easily spoofed. An attacker can send packets with the spoofed source address of an authorized NMS to crash the destination device. Moreover, some SNMP implementations by default accept SNMP packets sent to the network broadcast address. Attackers can easily send broadcast packets to compromise the whole network even if they do not know the target device’s address and the SNMP community name.³

ASSESSING THE THREAT

Although CERT has not yet observed significant activities or prevalent tools that exploit the weaknesses in SNMP products, the risk of system compromise is extremely high. These vulnerabilities can cause denial of service conditions or service interruptions and in some cases can allow an attacker to gain access to an affected device. Specific effects vary from product to product. Several vendors, such as Microsoft and Cisco, have published security advisories to address

SNMP vulnerabilities and provide fixes for their products.

Since in most cases SNMP services are not enabled by default, home users are not directly threatened by these vulnerabilities. However, because SNMPv1 is widely used in critical network infrastructure devices such as routers and switches, the exploitation of these vulnerabilities can lead to large-scale network instability and outage. Especially if attackers combine these vulnerabilities with the security flaws in Internet routing protocols such as the Border Gateway Protocol, the compromise of one main router can cause the whole Internet to become unstable. If a large number of devices, such as Cisco routers, have the same buffer overflow vulnerabilities in SNMP, hackers also could write a worm like Code Red to exploit

the buffer overflow, which could lead to another round of worm outbreaks.

SOLUTIONS

Most vendors of SNMP-enabled devices have released recommendations for removing the vulnerabilities from their products. Based on CERT’s advisory, here we list some general solutions that can protect your network.

SNMPv1 SCANNERS

Several organizations have released tools that scan networks for devices running SNMP. SNMPing, developed by SANS, is a Windows-based tool that seeks SNMP daemons on port 161 or a user-specified port. SNScan, a similar Windows-based utility developed by Foundstone, quickly and accurately identifies SNMP-enabled devices on a network. Both SNMPing and SNScan are free to download. To get a copy of SNMPing, send an e-mail to snmptool@sans.org; for SNScan, see www.foundstone.com/knowledge/free_tools.html.

VENDOR PATCHES

Once you’ve located the SNMP-enabled devices on your network, you can check with the vendors of these devices to find

out if they have developed patches. Vendor-provided patches improve the handling of malformed SNMP messages in various ways, such as by adding stronger checking to test the validity of incoming SNMP messages.

DISABLING THE SNMP SERVICE

If you do not require SNMP service for your network, CERT recommends disabling or removing this service. However, OUSPG’s testing showed that some affected products were susceptible to denial of service attacks or other unstable behavior even with SNMP disabled.

INGRESS FILTERING

Firewalls and routers can be set up to perform ingress filtering at a network border. Ingress filtering of UDP ports 161 and 162 can prevent attacks from

If you do not require SNMP service for your network, CERT recommends disabling or removing this service.

external networks onto vulnerable devices within the local network. Other ports that handle SNMP-related services—including TCP and UDP ports 161, 162, 199, 391, 750, and 1993—can require ingress filtering as well. An advisory notice from the US Department of Energy’s Computer Incident Advisory Capability provides more information about these ports.⁴

EGRESS FILTERING

Devices that provide public services do not normally initiate outbound traffic to the Internet. To control traffic leaving your network, implement egress filtering. Filtering outgoing traffic from UDP ports 161 and 162 at your network border can prevent your system from being used as a launching pad for attack.

CHANGE DEFAULT COMMUNITY STRINGS

As already mentioned, most SNMP-enabled products have the default community strings “public” for read-only access and “private” for read-write access. These community strings should be changed from the default settings. The new community name will still be vulnerable to the packet sniffing, however.

UPDATE SIGNATURES FROM VENDORS

Up-to-date IDS signatures could provide another solution. Signatures that directly address the flaws found by Protos are now available from many intrusion detection system vendors. For example, the open source network intrusion detection community Snort (www.snort.org/) has created several rules specific to the malformed packets created with the Protos suite. Cisco has updated the signature for its Secure Intrusion Detection System, available for anonymous download at <ftp://ftp-eng.cisco.com/csids-sig-updates/S17/>. And Internet Security Systems (www.iss.net/) has released a generic signature for its RealSecure and BlackICE products.

With the simplicity of the popular Simple Network Management Protocol comes an inherent vulnerability to attack. Because SNMP is so widely deployed, networks far and wide could be exploited with disastrous consequences. CERT, researchers, and vendors have provided some solutions that can help minimize the attack potential from these vulnerabilities.

For more on this topic, see "Protocol-Related Problem Threatens Internet Security," April *Computer* p. 20. 

REFERENCES

1. "CERT Advisory CA-2002-03: Multiple Vulnerabilities in Many Implementations of the Simple Network Management Protocol (SNMP)," 12 Feb. 2002, www.cert.org/advisories/CA-2002-03.html (current 11 March 2002).
2. "PROTOS: Security Testing of Protocol Implementations," 19 July 2001, www.ee.oulu.fi/research/ouspg/protos (current 11 March 2002).
3. "PROTOS Test-Suite: c06-snmpv1," 12 Feb. 2002 www.ee.oulu.fi/research/ouspg/protos/testing/c06/snmpv1 (current 11 March 2002).
4. "M-042: Multiple Vulnerabilities in Multiple Implementations of SNMP," 12 Feb. 2002, www.ciac.org/ciac/bulletins/m-042.shtml (current 11 March 2002).

Guofei Jiang is a senior research engineer with the Institute for Security Technology Studies at Dartmouth College. Contact him at gfj@Dartmouth.EDU.

It is tempting to believe that our main focus ought to be on building more secure and reliable systems in the first place. However, it is important to recognize that most engineering designs are based on assumptions, models, and paradigms that do not scale well or age gracefully. For example, civil engineering scholar Henry Petrosky has observed that catastrophic bridge failures occur at regular intervals (*To Engineer is Human*, Vintage Books, 1992).

Any given bridge design methodology is eventually pushed, by naiveté or error, beyond the validity of its modeling assumptions, literally to the breaking point—with failures occurring in the field, not on the drawing board. Similar cycles of paradigm failures seem to run in electric power grids, financial marketplaces, transportation systems, and the environment. The point is that good design is not enough; design assumes a context and that context will eventually morph into something we cannot predict.

Let me propose a new thinking based on the premise that our infrastructure systems will be compromised or fail outright eventually, regardless of our diligence in designing them. Whether due to malicious attacks or organic failures, our networks will always be vulnerable. If so, we should focus more effort on developing detection and control technologies that will intercept signs of failure modes early on, mitigate the effects, and respond aggressively with countermeasures, all on the time scale of the threats or failures themselves that are increasingly measured in milliseconds. This kind of thinking naturally raises issues in diversity (heterogeneity of platforms and protocols), privacy (early detection requires comprehensive monitoring), and liability (shutting down critical applications and services when compromised).

Not by coincidence, these same issues arise in modern health care, a system that has evolved over thousands of years. Health care is a tripod, whose three legs are basic biomedical research (done by academic and industry researchers), the delivery subsystems (consisting of physicians, clinics and hospitals), and the public health system (detecting epidemics and tracking trends in a population). Of these three components, the analog of the public health system for networked infrastructures is arguably the least developed. We have no effective counterparts to the Centers for Disease Control or state-based public health offices in the network technology domain, counterparts that can operate on the time-scale of the attacks themselves, not the time-scales of human analysts and software patches.

Regardless of our approaches, we must recognize that addressing the technical challenges of modern security and privacy will be a long march. There are no quick fixes, no silver bullets. Imagine, again by analogy with health care, that we increase funding for medical research tenfold over the next 10 years. Surely this would accelerate the discovery of new therapies for cancer, heart disease, and other illnesses, but few of us who survive that decade would expect to be immortal by the next.

New thinking that leads to long-term solutions in security and privacy won't be manifest in short-term hardware or software gizmos. No, the new thinking has to be a wholly different attitude about the role and importance of networked infrastructure in our lives. Only such thinking will lead to the long-term, sustainable institutions and investments in security and privacy that we deserve and that will ultimately make a difference. 



George Cybenko is the Dorothy and Walter Gramm Professor of Engineering at Dartmouth College, Hanover, NH. Contact him at gvc@dartmouth.edu.