

Purpose of Briefing

1. The purpose of this technical note is to make members of the UNIRAS constituency aware of what is known by the National Infrastructure Security Co-ordination Centre (NISCC) about vulnerabilities in the Simple Network Management Protocol (SNMP) and to provide details of suggested remedial action to minimise the risk of attack against your networks.

Current Situation

2. The situation at the time of writing is that vulnerabilities relating to implementations of a commonly used network management protocol, SNMP, are in the public domain and these are being exploited. All versions of SNMP are potentially vulnerable. The vulnerabilities can lead to denial of service of SNMP enabled equipment, including routers. However, a number of these vulnerabilities are buffer overflows, which means that arbitrary code could be executed on a remote system with the privileges of the SNMP agent, which is usually system administrator. This means that an attacker could remotely gain control of a system running SNMP. These vulnerabilities have been recognised and are being exploited. Both denial-of-service and buffer overflow exploit scripts have now been seen in the wild, coupled with scanning for the SNMP vulnerabilities.

3. These vulnerabilities were discovered by the Secure Programming Group of the University of Oulu in Finland (OUSPG) as part of their programme of testing implementations of commonly used communications protocols (see <http://www.ee.oulu.fi/research/ouspg/protos/testing/c06/snmpv1/index.html>).

Background on SNMP

4. In order to understand the nature of the vulnerabilities in implementations of SNMP discovered by the University of Oulu, some background on the SNMP protocol is required.

5. SNMP is a protocol for the management of networked devices by means of agents on those devices reporting to management console or master. In terms of packet structure, the application level of an SNMP packet consists of a binary encoded sequence comprising SNMP version, community string and data, known as a protocol data unit. SNMP messages are specified in a data representation language called Abstract Syntax Notation One (ASN.1) and are binary encoded using a subset of the ASN.1 Basic Encoding Rules (BER). SNMP Version 1 is specified in Request For Comments (RFC) 1157, which is available on-line at <http://www.ietf.org/rfc/rfc1157.txt>.

6. The protocol data unit can represent 5 types of action, which are as follows:

- Get request (to get the value of an object instance)
- GetNext request (to get the value of the next object instance)
- Get response (the response to a get request)
- Set request (to set the value of a variable)
- Trap (to report a particular event to the management console)

7. For all packet types other than a trap, the packet data unit of the SNMP message comprises a request identifier, error status, error index and a list of object identifiers and values. If the data field is a trap, the data field has a different format, namely enterprise (type of object causing the trap), agent IP address, generic trap identifier, specific trap identifier, time stamp and a list of object identifiers and values.

8. The BER encoding is split into

- Data type class (one of universal, application, context-specific or private) and data type identifier (eg integer or printable string, which have universal class)
- Length of data
- Value of data

9. These binary encoded values are represented as a sequence of bytes. Using the Basic Encoding Rules values can be comprised of further triples of data type identifiers, length of data, value of data. This nesting of the value field is used for encoding sequences and sets of values.

10. The vulnerabilities that the University of OULU Secure Programming Group discovered were found by passing specially constructed BER encoded SNMP packets to various SNMP applications. This approach is generally known as interface testing. The tests used to assess the SNMP application were divided by packet data unit type (ie whether the packet data unit is a Get request, a Get response, etc) and fell into the following categories:

- SNMP fields of incorrect type
- SNMP fields of incorrect
- Null values
- Values in fields larger than the limit of the data type
- Boundary values of data type (e.g. largest integer which does not represent a negative integer)
- Data containing special characters, such as those used in format strings vulnerabilities in the C programming language
- Completely incorrect data

11. The testing on SNMP was performed according to the University of Oulu's general interface methodology with tests categorised into the following types:

- Bit pattern exceptions
- BER encoding exceptions
- Format string exceptions
- Integer value exceptions
- Missing symbol exceptions
- Overflow exceptions

12. Each of these types was applied to the following SNMP packet field types:

- Community strings
- Data field values, especially object identifiers
- Data type identifiers
- Data lengths
- Data values for the corresponding data types

13. Examples of the kind of test performed included issuing a Get request for an object with a zero length object identifier or sending a request with a community string that contained C formatting information (e.g. "%n" to write to memory).

14. It should be observed that these interface tests are designed to test for potential buffer overflows and format string errors in both the parsing by an SNMP application of the BER encoding rules and in constituents of the SNMP packets.

15. The vulnerabilities that were exposed by the tests conducted by the University of OULU are documented in the CERT Co-ordination Centre advisory CA-2002-03 (see <http://www.cert.org/advisories/CA-2002-03.html>), and include:

- Buffer overflows in community strings
- Buffer overflows in SNMP trap packets
- Format string errors in community strings
- Denial of service in all request types

16. These vulnerabilities do not affect all products tested, but no product passed all tests. The OUSPG informed various product vendors whose products contain SNMP implementations. The response of the vendors is included in CERT Co-ordination Centre advisory CA-2002-03.

Remediation

17. The best form of remediation is to apply vendors' patches if available. Check with equipment vendors for upgrades to SNMP implementations, and apply these as soon as they are released. However, at the time of writing some vendors have not produced patches for all of their products. If a patch is not available then the following remedial action is recommended.

18. Disable SNMP where practicable. In any case always disable SNMP (and remove any SNMP-related binaries where possible) on all hosts that do not use SNMP for network management.

19. If you cannot operate without SNMP, use a private management network with a separate interface and with an access control list restricting access only to authorised management workstations on the management network.

20. Use strong passwords (eg 7 characters or more using a range of ASCII characters, not only letters) for all community strings and change them regularly. It is vital that community strings of PUBLIC (read-only access) and PRIVATE (read-write access) are changed.

21. Block SNMP (commonly TCP/UDP ports 161 and 162) at your network perimeter to external networks. If this is really not possible, use an encrypted virtual private network across untrusted networks (e.g. the Internet) to tunnel SNMP traffic.

22. Restrict SNMP connections on internal networks to legitimate network management hosts only (ie use device-based access control). As IP addresses can be forged, the firewall should check that internal addresses have not been received on an external interface.

23. Consider enabling SNMP traps to flag up attempted connections (a) with incorrect community strings or (b) from non-network management hosts and logging all SNMP traffic. These steps will provide a detection and auditing mechanism in the event of an attack.