

Windows 2000, SNMP and Security

Dirk Wisse 2001-04-25

Windows 2000, SNMP and Security

by *Jan van Oorschot, Jeroen Wortelboer and Dirk Wisse*

last updated April 25, 2001

SNMP - The Mission Statement

Not too long ago, when the Internet was still young and yet to be commercialized, security was mainly something for the other guy to be concerned about. For most people involved in the enterprise, the problem at hand was to get the thing up and running, and keep all the connected boxes happy. For every systems person involved, this meant managing dozens of routes, switches and computers, some of which where either too far away to get to the keyboard, or had no keyboard at all. Telneting to the box was a good solution to this sneaker-wear problem, but not all boxes had enough intelligence to provide a shell.

It was for this problematic domain that the Simple Network Management Protocol (SNMP) was invented. SNMP had to be simple enough to be implemented by even the stupidest box (yes indeed, even the toaster), yet powerful enough to enable remote management. Complex network management stations could be developed using only SNMP as communication facility. Stateless UDP/IP was used for all network traffic, and all management operations were expressed as the reading and setting of simple variables.

Supporting SNMP became a necessity for any box that could be connected to the Internet, and Microsoft was ever-attentive to such market demands. Soon, Windows 95 and Windows NT 4 became SNMP-capable, followed by their successors Windows 2000 and Windows XP.

Unfortunately, in trying to keep SNMP simple, the original architects forgot to include some basic security features, such as believable authentication and adequate encryption. To remedy these and some other shortcomings, SNMP V2 was invented; unfortunately V2 went overboard and became much too complex (will we techies ever learn?). This version of SNMP, also know as 'V2 Classic', is hardly used any more, and was replaced by SNMP V2c, which resembled V2 but without all that irritating authentication and encryption stuff built in. In turn, V2c has since been replaced by V3, which enhances SNMP even further, but which still lacks massive support.

In this article, we will examine SNMP in the context of Windows 2000, focusing mainly on the security aspects. SNMP can be beneficial for the overall level of security but can also be a risk - this discussion will examine both aspects. Only standard Windows 2000 features and tools will be covered in this discussion, except for some locally developed tools that illustrate the possible (mis)usage of information gathered through SNMP.

Windows 2000 and some Basic SNMP Operations

Providing a Windows 2000 box with SNMP support is as simple as adding an extra Windows component. As Administrator, just go to ?Start/Settings/Control Panel/Add/Remove Programs?, and select the button ?Add/Remove Windows Components?. In the familiar popup box that appears, select ?Management and Monitoring Tools? and let the stuff install itself. Voila, your Windows 2000 Professional or Server box accessible through SNMP, both locally and remotely (Yesss!), by everybody (Oops.)

Lets do some hands-on and try to get the number of network interfaces in this machine:

```
C:> snmputil get localhost public .1.3.6.1.2.1.2.1.0
      Variable = interfaces.ifNumber.0
      Value    = Integer32 1
```

There is much to discuss about this command and its output. The ?snmputil.exe? utility itself, which is a part of the Windows 2000 resource kit provides you with a basic, lowlevel SNMP utility to ?manage? your boxes, by displaying and modifying SNMP variables. In this example, the variable is called 1.3.6.1.2.1.2.1.0, and we ?get? its value, which turns out to be ?1?. This piece of text was developed on a disconnected W2K portable, so this number could be correct since only the ?MS TCP Loopback interface? is available.

The weird variable name (1.3.6. etc..) is called an object identifier or OID. An alternative to this is found in the output of SNMPUTIL. The ?interfaces.ifNumber.0? is the same OID, but is more easily readable for the user. (More about these variable names in the next section about Management Information Bases or MIBs.) The second and third arguments to SNMPUTIL designate the box to which the SNMP request will be sent (?localhost?), and community (authentication string or password) to use (?public?). And yes, this password will go clear-text over the line, in every SNMP UDP packet that is sent to the managed box. The ?public? community is the default when SNMP support is installed on a Windows 2000 box, and it allows the user to read all variables present. Since even the number of interfaces in a box is sensitive data, we have found our first security issue!

Lets try another feature of our snmputil program: ?getnext?:

```
C:\> snmputil getnext localhost public interfaces.ifNumber.0
Variable = interfaces.IFTable.ifEntry.ifIndex.1
Value    = Integer32 1
C:\>snmputil getnext localhost public interfaces.ifTable.ifEntry.ifIndex.1
Variable = interfaces.ifTable.ifEntry.ifDescr.1
Value    = String <0x4d><0x53><0x20><0x54><0x43><0x50><0x20><0x4c><0x6f>
            <0x6f><0x70><0x62><0x61><0x63><0x6b><0x20><0x69><0x6e>
            <0x74><0x65><0x72><0x66><0x61><0x63><0x65><0x00>
```

The ?getnext? operation doesn?t get the value of the given SNMP variable; rather, it gets the value (and

the OID) of the next variable. Since all SNMP variables in a box are lexicographically ordered, this ? powerful? getnext operation enables you to retrieve all variables from a box without knowing their exact names (OID?s) beforehand. (By the way, if you decode the String value, you?ll get the name of the one interface ?MS TCP Loopback interface? with a zero at the end). Tired of typing the ?getnext? command? Want to automate the process? This is already made possible by the snmputil utility by the ?walk? command, the following example of which will give the reader (almost) all variables present in his or her local computer:

```
C:\> snmputil walk localhost public .1.3
```

There is a last SNMP message that we have not discussed until now: the infamous SNMP trap. The main reason why we didn?t show a command line operation to illustrate a trap is that there is no such command. An SNMP-capable box sends SNMP trap messages whenever it feels like it, but primarily when it has something important to say. We will see later on that SNMP traps can be very beneficial when securing a Windows 2000 setup.

MIBs - A Simple View on a Box

As we have seen, variable names in SNMP are variable-length sequences of numbers called OIDs. Their namespace is extendable, so people can always define new variables without generating name clashes with existing SNMP boxes. Also, any set of these variables can be ordered lexicographically, enabling the querier to walk through a set of variables in a deterministic way. But that is just about all the good news you?ll find about OIDs: they are hard to read, write and remember. But worse, an OID gives no clue whatever about the meaning of that SNMP variable.

For this reason, people developing new groups of SNMP variables are expected to define these variables in a formal document called a ?Management Information Base? or MIB. Quite a few of these MIBs already exist, defining everything from the CPU temperature to the load of a network interface. But the ones that are relevant to this discussion are those that define the variables inside a W2K box. Have a look at the *.mib files in the \winnt\system32 directory, which are part of the Windows 2000 SNMP service.

Reading these MIB files will teach you a lot about all these variables. But before ?snmputil.exe? will display the human readable variable names, these MIB?s first have to be compiled using the ?mibcc? utility from the Windows 2000 Resource kit. The following example should give you enough to get started:

```
C:\> MIBCC -w3 ?o c:\winnt\system32\mib.bin msft.mib acs.mib dhcp.mib wins.mib
```

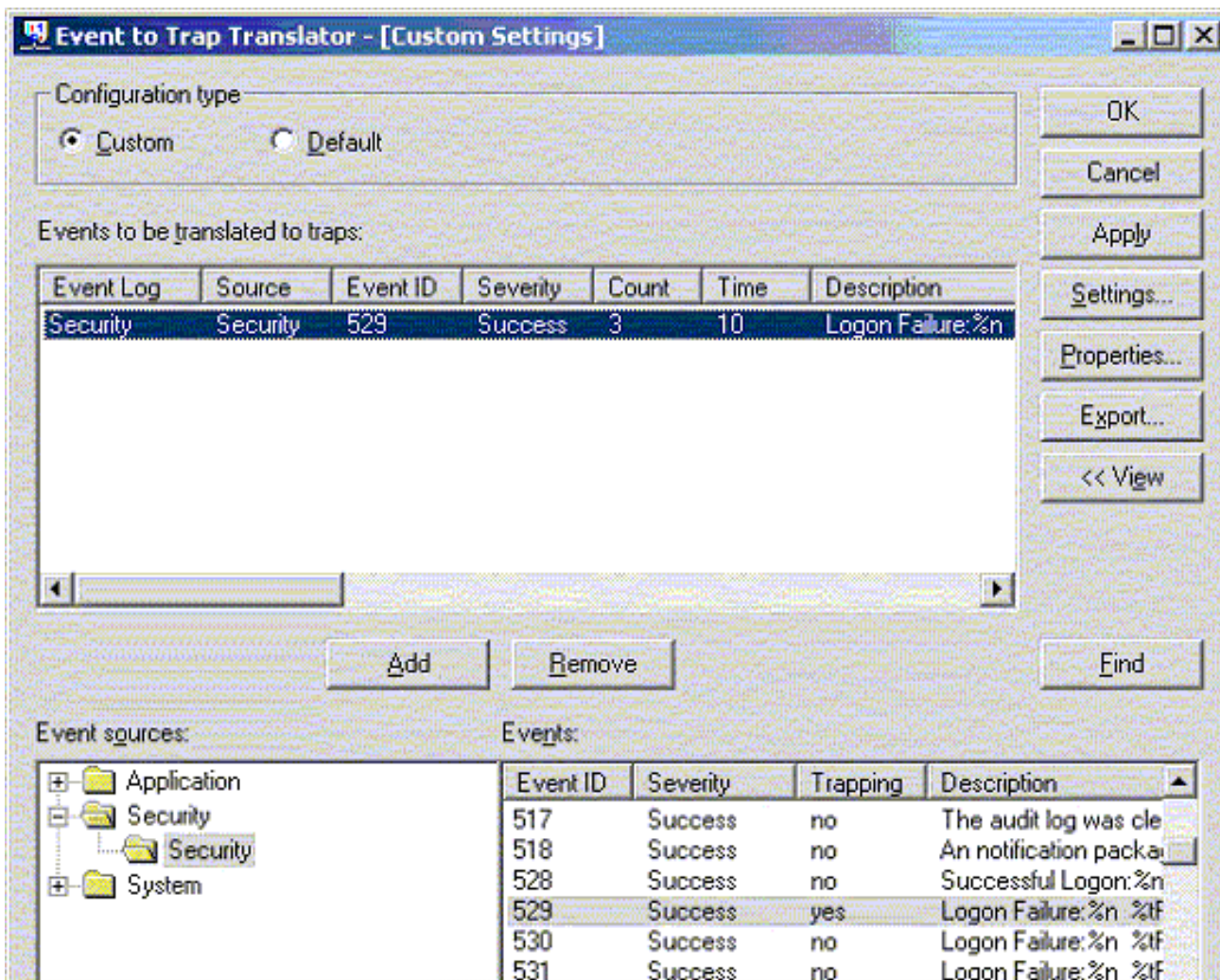
Note that the default ?mib.bin? that is shipped with Windows 2000 is generally sufficient for most purposes, so compiling your own is optional.

W2K SNMP - Enhancing Security

So Windows 2000 implements several SNMP MIBs and a number of tools are available to collect these variables, now what? We will skip the possibilities for using SNMP as a valuable system management tool, but will instead focus on security issues, starting with some positive aspects.

In the Windows 2000 environment, any event in the eventlog can be used to generate a SNMP trap containing that event. Microsoft calls this facility the "SNMP Event Translator". It is possible to configure a group of Windows 2000 boxes so that they send an SNMP trap when somebody tries unsuccessfully to login to a system or mount a share. By directing these traps to your own workstation and having some small application play a warning melody when a trap is received, you have the start of a multimedia security management station.

The tools you need to configure these traps are "eventwin.exe" and "eventcmd.exe", both of which are part of the standard Windows 2000 installation. The following screenshot shows the configuration of the mentioned login failure using "eventwin.exe":



Use "eventwin.exe" to generate traps on some major events, and start our old friend "snmputil.exe" on

your management workstation:

```
C:\> snmputil trap
snmputil: listening for traps...
Incoming Trap:
  generic      = 6
  specific     = 529
  enterprise   = .iso.org.8.83.101.99.117.114.105.116.121
  agent        = 127.0.0.1
  source IP    = 127.0.0.1
  community    = public
  variable     = .iso.org.1.0
  value        = String
  variable     = .iso.org.2.0
  value        = String SYSTEM
  variable     = .iso.org.3.0
  value        = String JVO
  variable     = .iso.org.4.0
  value        = String 16
  variable     = .iso.org.5.0
  value        = String 2
  variable     = .iso.org.dod.0
  value        = String nonexistinguser
  variable     = .iso.org.7.0
  value        = String ADS
  variable     = .iso.org.8.0
  value        = String 3
  variable     = .iso.org.9.0
  value        = String NtLmSsp
  variable     = .iso.org.10.0
  value        = String MICROSOFT_AUTHENTICATION_PACKAGE_V1_0
  variable     = .iso.org.11.0
  value        = String JVO
```

It is beyond the purpose and scope of this article to delve too deeply into the details of the trap we received; however, W2K experts will recognize that all parts of a Windows event are present. Wrapping the trap receiver in a Perl or VBA script would enable users to store the events in a database or send out an email to the security officer.

Remember that SNMP traps will only be generated for events that appear in the Windows 2000 eventlog. For stand-alone workstations, this is configured in the ?Local Security Settings? screens (Start/Administrative Tools/Local Security Policy). For boxes that are in a domain, you can use the Active Directory equivalent.

Using SNMP traps in this manner offers two big advantages. First, administrators don?t have to access each and every Windows 2000 box individually. Second, users don?t have to wade through megabytes of

eventlogs. This could make the difference between ignorance and reacting to security threats in an informed manner.

The `evntwin.exe` utility, and its command line counterpart `evntcmd.exe`, are easy to use and enable users to set thresholds on the number of events and on a time interval. Defining a good policy will make it hard for culprits to flood users with SNMP traps. At the same time it will warn users of suspect activity. Unfortunately, there seems to be a bug in the `SNMP Event Translator`. According to Microsoft's TechNet, "the SNMP Event Agent is not notified for every new event that is logged when events come into the log at a very fast rate. The SNMP agent is notified when a new event comes into the log and a trap is sent, but that is dependent on a new event being logged, which may introduce a long delay?". (<http://support.microsoft.com/support/kb/articles/Q284/2/55.ASP>).

Another way of using SNMP for additional security is to actively monitor some variables. Some possibilities for monitoring include:

- `IpForwarding (1.3.6.1.2.1.4.1.0)` - Is your box forwarding? This is not a good sign for most workstations.
- `IcmpInRedirects (1.3.6.1.2.1.5.7)` - Is somebody sending your box icmp redirect messages? This will only make sense in an environment in which you would not expect redirects.
- `TcpOutRsts (1.3.6.1.2.1.6.15)` - A counter indicating the number of RST's send by the box. This counter will increase rapidly when port-scanned.
- `UdpNoPorts (1.3.6.1.2.1.7.2)` - A counter indicating traffic to ports where no service was present. Also a possible port-scan signal.

If you happen to have a SNMP management station, setting thresholds on these variables will give you nice alerting information.

W2K SNMP, Threatening Security

Many sites enable SNMP support on their Windows 2000 servers and workstations but leave the default `READcommunity` unchanged (i.e. `public`). Even if they remember to change the default community, it is entirely too easy to sniff it from the network or to get it using a dictionary or brute force attack (most SNMP stacks allow you to try, try and try again). This may not seem worrisome but the Windows 2000 SNMP variables contain countless treasures for the sniffing cracker. The following list describes some of the tables that are available when one has `READ` access to the SNMP tree in a Windows 2000 box:

- `Interface Table` - This table identifies all boxes with multiple interfaces, plus useful details like their IP and MAC addresses. This will help crackers in their search for more important targets.
- `Route Table` and `ARP Table` - With access to these tables, a cracker can quickly build an accurate picture of a network and continue its search for vulnerabilities.

- TCP Table and UDP Table - These will show which TCP and UDP ports are actively used, and on which ports services are listening for new clients. You would not want the cracker to have to sweep through all ports - the IDS software might even get alerted.
- Device Table and Storage Table - Knowing what hardware is attached to a Windows 2000 machine gives crackers clues about what kind of machine it is dealing with. They might even pay the site a visit in the flesh to enlarge their stock of hardware.
- Process Table and Software Table - Knowing what software is installed and what software is running (DNS server, DHCP server) gives away details about how to attack the system. They even show which fixpacks have been installed (and which haven't!!!)
- User Table - Knowing what user names are valid on a machine makes it much easier to guess passwords and gain access to a system without actually having to break things. Especially tasty when eaten on a Domain Controller!
- Share Table - If the cracker knows what shares are exported and used by a Windows machine, it can just try ?using? them. You would be amazed how many shares on portable computers can be used by just about anyone.

W2K SNMP, Dos and Don'ts

Enabling SNMP on Windows 2000 boxes can be both good and bad for security. Often it is company policy to enable SNMP for system management purposes, even though the security officer might object. In any case, here are some hints and tips for using SNMP in the Windows 2000 environment:

- Don't Install if Not Used - Only install SNMP support if somebody is actually (and legally) going to use it. Otherwise remove this potential backdoor from your systems.
- Change Default Community and Restrict Access - Change the default community! Also, identify the workstations from which SNMP requests may be issued, and restrict SNMP access to these addresses. If possible restrict access to READ-ONLY. All these configurations can be done by changing the properties of the ?SNMP Service? (Start/Administrative Tools/Services).
- Authenticate/Encrypt using IPSEC - SNMP (V1) may not have adequate authentication and encryption facilities built in but this is where IPsec can come to the rescue. You can define Ipsec policies in your monitored systems and management stations so that all SNMP traffic is authenticated and/or encrypted. See the article titled "Securing SNMP Messages with IP Security" in the Win 2000 server resource kit for details.
- Collect Traps - If SNMP is enabled, start using those Windows 2000 eventlogs. Effective auditing of your boxes will actually raise the level of security.

Relevant Links

Scripts Used in This Article

Carotechnology

SNMP Resources

The SimpleWeb

[athena-2k.pl - A Windows 2000 SNMP auditing tool](#)

Jacob Shaw

Introduction to SNMP v3

Network Working Group

[JoeSNMP - SNMP for Java](#)

OpenNMS

[Privacy Statement](#)

Copyright 2006, SecurityFocus