Global Information Assurance Certification Paper

Guillaume Tamboise

HOW-TO securely use SNMP on a BGP/MPLS VPN network

GSEC Practical Assignment Version 1.4 (amended April 8, 2002), Option 1.

Abstract

Let us consider the case of an IP/MPLS Service Provider offering extra-net connectivity, along with access to services. The Service Provider manages its MPLS network and in some cases the Customer Edge (CE) routers. The network operations are made possible by its Operations and Business Support System (OSS/BSS) devices, hosted behind some of its own CE routers. Some value may be added by on-demand services hosted behind CE routers on managed servers. All these components can be managed using SNMP; we will see how to make the components interact safely.

Introduction

Simple Network Management Protocol (SNMP) is a widely spread tool to monitor large-scale networks. It potentially gives read and write access to configuration information of the device that runs a SNMP agent, if not full control of it when exploiting the latest buffer overflow [CERT107186, CERT854306]. The whole purpose of SNMP is to allow a centralized management, which requires network connectivity between the Network Management Stations (NMS) and the devices. The different players must interact, without opening transversal and non-desirable doors between them. The threat agents we will consider are SNMP packets: it can be queries, answers, or bogus packets. The threat against the service provider's assets contains not exclusively information warfare techniques:

- Reconfiguration of the provider's edge and core network
- Denial of Service on the provider's network or the provider's services
- Denial of Service on the provider's network or services in the name of another customer
- Tampering with the provider's services
- Tampering with the provider's services in the name of another customer
- Tampering with the provider's OSS tools

Quick Overview of SNMP

SNMP is the standard network management protocol running on IP. Its very first version, covered by RFC 1067, was published in 1988 [RFC1067].
[SNMPLS] gives an overview of SNMP:

> The strategy implicit in SNMP is that the monitoring of network states at any significant level of detail is accomplished primarily by polling for appropriate information for making the best possible management solution. A limited number of unsolicited messages (traps) guide the timing and focus of the polling. Limiting the number of unsolicited messages is consistent with the goal of simplicity and minimizing the amount of traffic generated by the network management function.
> In other words, SNMP is a set of rules that makes many hardware

devices, such as computers and routers, being able keep track of different statistics that measure important features, such as number of packets received on an interface. The different information SNMP retrieves is kept in each separate database, called Management Information Base (MIB). Other kinds of equipment have configuration information available through SNMP.

SNMP might be used for various purposes: fault management, device configuration, Quality of Service (QoS) monitoring to report on Service Level Agreement (SLA) compliance. The Network Management System is part of the OSS/BSS system.

The reporting abilities of the OSS/BSS servers are central to the Service Provider. A Denial of Service attack carried on a Service Level Agreement management server can have a disastrous impact on the Service Provider. They are definitely worth being monitored themselves; SNMP provides all the required functionalities for this monitoring.

SNMP can be used to monitor the network health, including sophisticated QoS measurements: response time of a web server from a Provider Edge router customer facing interface, edge-to-edge latency or jitter...

SNMP uses Management Information Base (MIB) that follows a hierarchical structure. Most vendors define their own MIBs to fully utilize the particular possibilities of their devices (where the SNMP agent is running). The strength of a Network Management System lies on its ability to import the vendor specific MIBs, easily extending its reporting functionalities.

A security test run on the current implementations of SNMP version 1 led to the discovery of a series of flaws detailed in [CERT107186, CERT854306]. They led to a massive upgrade of SNMP implementations, helping people to realize that it was time to upgrade to newer versions.

SNMPv3

The implementation of SNMPv3 is in the road map of many NMS as it is an IETF proposed standard. However, SNMPv2c is still the most commonly used. On the agent side, SNMPv3 is already widely spread, though the lack of wide implementation on the NMS side is a limiting factor.

Versions 1 and 2c are still largely deployed, and feature security vulnerabilities such as plain text authentication. Additionally, SNMP is build on top of UDP, making spoofing easier than if the attacker had to guess a TCP sequence number.

SNMPv3 uses a user model [RFC2574], which gives more granularity and “modern” authentication schemas. SNMPv3 is to SNMPv2 what ssh is to telnet: a necessary improvement that all network and system administrator should take into account.

Cisco’s website has a table that summarizes the authentication and encryption features of the various versions [CISCOv3]:

SNMP Security Models and Levels

Model	Level	Authentication	Encryption	What Happens

v1	noAuthNoPriv	Community String	No	Uses a community string match for authentication.
v2c	noAuthNoPriv	Community String	No	Uses a community string match for authentication.
v3	noAuthNoPriv	Username	No	Uses a username match for authentication.
v3	authNoPriv	MD5 or SHA	No	Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithms.
v3	authPriv	MD5 or SHA	DES	Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithms. Provides DES 56-bit encryption in addition to authentication based on the CBC-DES (DES-56) standard.

Since SNMPv3 gives the choice for authentication between MD5 and SHA, let us use SHA if possible. Indeed, MD5 has proven vulnerabilities to birthday attacks (the possibility to find another password that gives the same hash)[HAC]: "*Collision for MD5 (and similar hash functions) can […] be found in O(2^64) operations and without significant storage requirements*."
That feature reduces the effective length of the key from 128 bits to 64 bits.

The encryption is based on DES; we just have to live with this weak algorithm. Being able to correctly authenticate the different SNMP peers is far more critical than worrying about someone being able to decrypt the management traffic on the fly.

A neat security feature in the SNMPv3 protocol is the localization of passwords. In the Cisco IOS world, the localization if provided by the Identifier of the snmp engine [CISCOv3]:

> Changing the value of snmpEngineID has important side effects. A user's password (entered on the command line) is converted to an MD5 or SHA security digest. This digest is based on both the password and the local engine ID. The command line password is then destroyed, as required by RFC 2274. Because of this deletion, if the local value of engineID changes, the security digests of SNMPv3 users will be invalid, and the

users will have to be reconfigured.

Having access to a local configuration no longer gives a nearly full SNMP access; the attacker is then contained for a little while.

Now that we have seen the main device management tool we are going to use, let us describe BGP/MPLS VPNs as a tool to enforce a network security policy.

Network Security

BGP/MPLS VPNs

Multi Protocol Label Switching is a switching method designed to ensure a fast and easy forwarding of IP packets. Forwarding is assured by adding labels to the IP packets on the edge of the MPLS network: routing decision is taken only once, at the edge, and then faster label switching forwards packet across the core.

In the MPLS world, the IP addressing plan of our core network does not need to be known by our Customers. We are not dealing with security though obscurity, but one more way to slow down the attacker. The only thing the customer knows is the IP address of the interfaces on the PE routers it is facing. All IP addresses of our core MPLS network are routed using a completely separate protocol (usually OSPF) that the customer does not need to know about at all.

When the Customer sends a packet on a MPLS network, the first IP hop is the Customer facing interface on the ingress PE router, the second IP hop is the Customer facing interface on the egress PE router.

BPG/MPLS VPNs are defined by RFC 2547 [RFC2547]. To keep it short [JUNIPER200001]:

> [The VPN service model proposed in RFC 2547bis] provides a mechanism that simplifies WAN operations for a diverse set of customers that have limited IP routing expertise. RFC 2547bis is a way to efficiently scale the network while delivering revenue-generating, value-added services.

To make our point on using MPLS VPNs features to safely use SNMP, we first need to give an understanding of how MPLS VPNs are working. In order to do that, we will use some simplifications and a logical representation, which does not necessarily reflect the exact mechanism behind. The way MPLS itself is working is out of the scope of this practical.

We first need some rough definitions:

- Border Gateway Protocol (BGP) version 4 is the current de facto exterior routing protocol in the Internet. It uses a path vector protocol over TCP to carry routing information between Autonomous Systems (AS). BGP has been extended to support MPLS VPN, becoming MPBGP, multi-protocol BGP.
- A VRF is a VPN routing/forwarding instance or in other words a virtual routing table. The VRF is local to a router. Interfaces going out of the MPLS cloud must be associated with a VRF.
- A RT is a Route Target. This is the mean by which routes can be exchanged

between VRF. A RT can be seen as a set of routes, which is known by all the PEs on the MPLS network (via MPiBGP, multi-protocol internal BGP). To advertise the routes from a VRF, a router will export the routes from a VRF to one or several of these sets. The router can also import into a VRF the routes existing in one or several of these sets.

- The MPLS cloud is made of P (Provider) and PE (Provider Edge) routers. A Provider Edge router (PE) is a piece of equipment that has interfaces connected to the MPLS cloud and interfaces connected to customer equipments. PEs exchange RTs information between each other with MPiBGP. They also exchange routes between the VRF associated to a CE and the corresponding CE using eBGP (or RIPv2 or OSPF).
- A Customer Edge router (CE) is a router connected to the customer network on one side and to a PE on the other side. It is a regular router; in the scope of this document it does not have any knowledge of MPLS.

Isolating the Customers with a Management VPN

The management VPN enables the OSS devices of the Service Provider to have connectivity to all managed services (including the Customer Edge routers if relevant), without enabling connectivity between those devices.

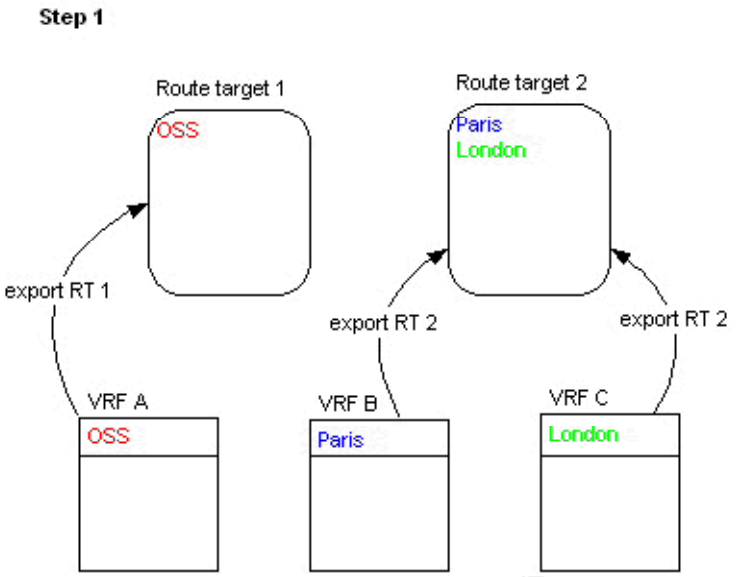
Using the next diagrams, we will see how the MPLS/VPN technology enforces the security policy “the OSS segment must have connectivity to the CE routers, but the CE routers must not have connectivity in the general case between each other”.

Let us consider three VRF. In the scope of this document, it is equivalent to considering three customer-facing interfaces on PE routers. VRF A leads to the OSS segment, VRF B to a customer in Paris and VRF C to another customer in London.

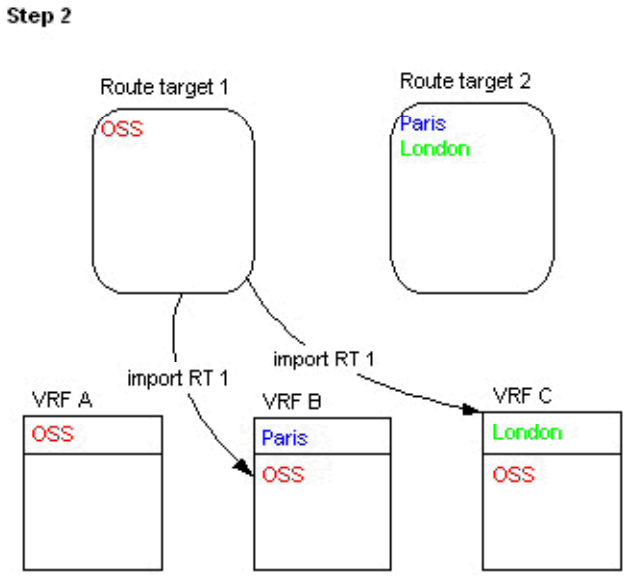
We also define two Route Targets, number one and two. The set of these two route targets can be called “management VPN”.

The purpose of RT 1 is to give customers access to the OSS segment; the purpose of RT 2 is to give the OSS network access to all customers’ networks.

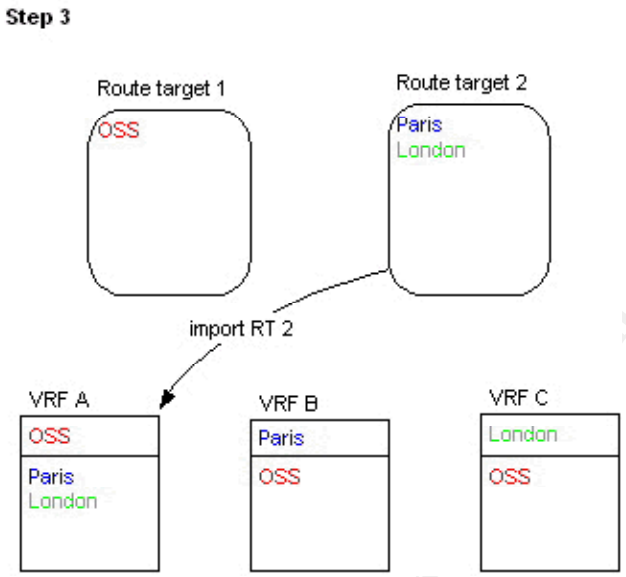
As a first step, VRF A exports its routes to Route Target 1. Route Target 1 will then contain the required routing information to reach the OSS segment. VRF B exports its routes to Route Target 2, so does VRF C. Route Target 2 will then contain the required routing information to reach both Paris and London.



As a second step, VRF B imports Route Target 1. So does VRF C. After this step, both VRF B and VRF C know the required routing information to reach the OSS segment. However, they do not have any routing information on each other.



Last, in step three, VRF A imports Route Target 2. The OSS segment has now the required information to reach both customers. However, both customers cannot reach each other.



This topology is known as a hub and spoke, with the OSS VRF as a hub. The spokes can speak with the hub, but not between each other.

It is important to underline that VRF B and C do not contain any information about the provider's core network. The customer is only provided information useful for its connectivity: the IP address of the interface he is facing. If he wants to get more, he will have to make illegal attempts that will be logged.

Protecting the Management VPN

Since the management VPN is implemented using BGP, we have all the BGP filtering functionalities at our disposal to implement it securely. We will not go in the details of the Route Descriptors, but they would be required to get this configuration to work.

On the Cisco PE router that gives the Customer access to the MPLS cloud, the steps to enforce our security policy would be:

1. Define the managed network, here we choose 192.0.2.128/25:
   ```
   ip prefix-list managed_network seq 5 permit 192.0.2.128/25
   ```
2. Prepare the export to the Management VPN of the routes leading to the managed devices. The `aa:cc` extended BGP community contains all routes to all managed devices, we need those routes to be added to this community (we called it route target 2 in the previous representation).
   ```
   route-map managed_network_map permit 5
   match ip address prefix-list managed_network
   set extcommunity rt aa:cc additive
   ```
3. Import and export the Route Targets. The `aa:dd` community contains the "normal" customer routes, the one that belong to its production VPN. The `aa:bb` community contains the routes to the NMS network, we called it route target 1 in the previous representation.
   ```
   ip vrf Customer_VRF
   export map managed_network_map
   ```

```
route-target export aa:dd
route-target import aa:dd
route-target import aa:bb
```

It is important to underline that when we apply a route-map, first the export is done (and the route target is set) and then the route-map is applied (adding some route targets in our case).

On the PE router that connects the NMS network to the MPLS network, the VRF of the NMS network's interface needs to export its routes to the route target aa:bb.

```
ip vrf NMS_VRF
route-target export aa:bb
```

We now have the basic knowledge to securely segregate the customers' traffic. This first step based on routing filters is definitely valuable, however it has to be accompanied with some sanity checks on the network access traffic itself.

Network Access

The situation is more common that we would expect in a "civilized" world. We should not be surprised that, because of poor security measures on our edges, one of our customers attempts, for example, a Denial of Service attack on another customer's database right before an important delivery date, because these two customers compete against each other. This kind of information warfare will definitely impact on our image as an ISP.

Protecting our customers from each other's turns into protecting ourselves as a service provider, from traffic originating from our customers.

We would like to be protected against the threat of impersonating, whether to get higher access or to launch a Denial of Service in the name of another customer. Ideally, we would like to make sure that all the traffic coming on a customer facing interface has a source IP belonging to this customer.

At least, we would like to make sure that the source IP does not belong to another customer - in case of two competitors willing to spoof each other's IP address. However, we will show that it can be rather difficult to discriminate between legitimate and not legitimate traffic. MPLS is not a magic bullet [JUNIPER200001]:

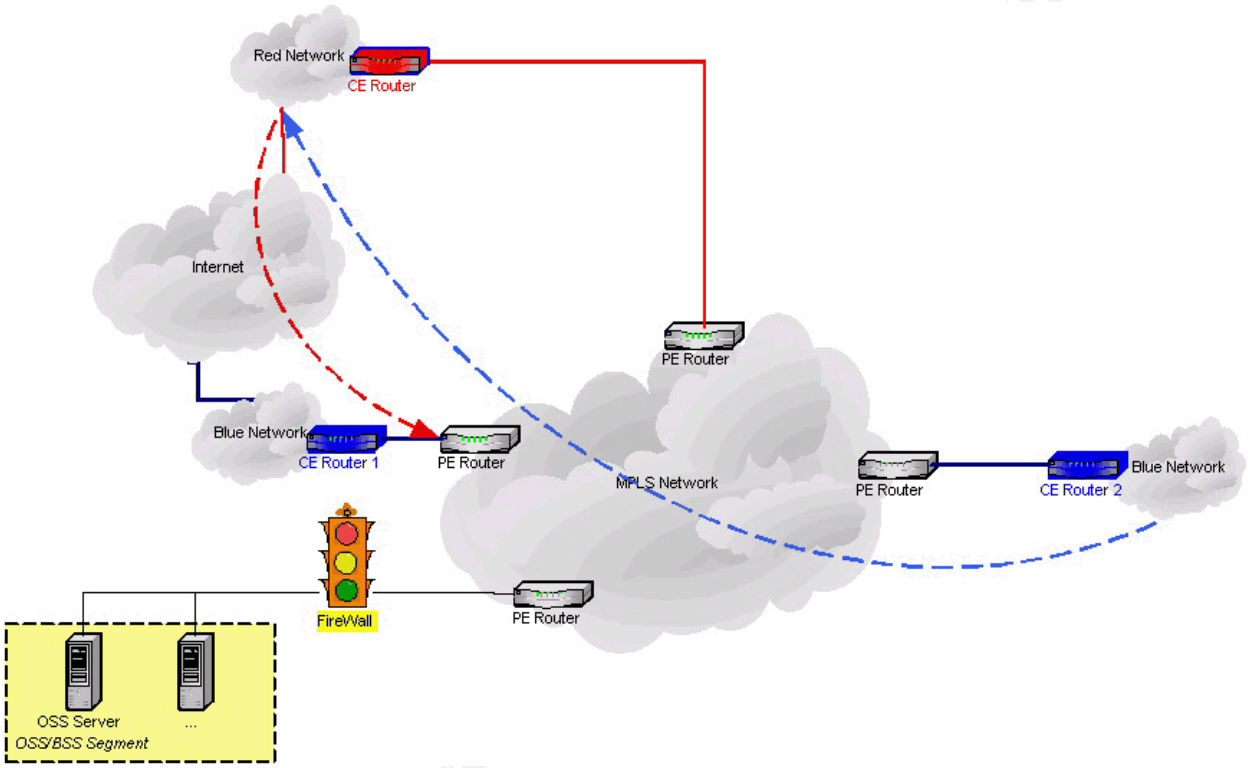
> Others in the Internet community believe that MPLS was designed to completely eliminate the need for conventional, longest-match IP routing. This never was an objective of the MPLS working group because its members understood that traditional Layer 3 routing would always be required in the Internet.
>
> Packet filtering at firewalls and ISP boundaries is a fundamental component of supporting security and enforcing administrative policy. Because packet filtering requires a detailed examination of packet headers, conventional Layer 3 forwarding is still required for these applications.

Let us imagine that one of our customers has an access point to the Internet. It

also has two separate sites; site one with an access point to the Internet, site two is a remote site and has no Internet connection. Site two uses the Internet access of site one to connect to the Internet. Let us call this customer the blue customer.

Some other customer of our MPLS cloud also uses our services, and another provider connects his network to the Internet. Let us call him the red customer.

The following diagram expresses this architecture:



Let us imagine that someone in the blue customer organization in site two wants to browse the web site of the red customer. He tries to open the connection and send a packet destined to the red network. The VRF table he has access to does not tell the packet that the MPLS cloud could give him direct access to the red customer, which is the whole point of MPLS VPNs (separating routing tables). The packet is forwarded to the default route to the Internet, using the main office.

When the packet comes back, its source IP address belongs to the red customer. Our PE router that connects the blue customer's main office sees a packet coming from the blue customer, but with a source IP of the red network. If we are not careful enough with our anti-spoofing access lists, we could well decide that this traffic is not legitimate, even though it is perfectly normal.

Let us imagine that we as a Service Provider manage the blue CE routers. To mitigate the risks while granting the best connectivity between the two blue offices, we know that only CE Router1 (respectively CE Router2) can send snmp answers to the OSS segment on the CE Router1-to-PE Router interface (respectively CE Router2-to-PE Router interface).

The access list to implement this policy would be, on the PE Router connected to CE Router1, if the PE Router is a Cisco router:

```
input access-list
permit udp CE_Router1 OSS_Segment snmp
deny udp any OSS_Segment snmp log
deny ip OSS_Segment any log
permit ip any any
```

The first line specifies that we are defining an input access-list (on the PE Router's interface facing the CE Router).

Then, we allow the UDP/SNMP traffic from CE Router1 to the OSS Segment.

All other snmp traffic going to the OSS segment is then dropped and logged. A correct configuration of `snmp-server host` can automatically send a trap to the NMS in case of violation.

On this interface, any traffic with a source address belonging to the OSS segment is a spoofing attempt, so the packets are dropped and logged.

Last, all the rest is permitted.

The strength of this access-list is that it is reproducible in a systematic way on all customer-facing interfaces and does not change when customers change.

This configuration remains valid even if the CE router is managed by the Service Provider and located in the Customer's premises. In this case, we cannot guarantee the physical security of the CE. The Service Provider is exposed to the threat that the Customer tampers with the CE and uses the CE's privileged accesses in a non-legitimate way.

We might consider protecting our MPLS network with a firewall between our PE router and some CE routers, especially those hosting our additional services or our management network. However, this solution might well not fit our network:

- Our firewall needs to understand the access technology that is used. This restriction would apply if we offer a VPN common to two customers, but restricted to some services.
- Our firewall needs to handle potentially high throughput. First-class firewall can reach throughputs of several gigabytes. A more reasonable firewall can handle a trunk carrying several hundreds of megabytes – it also has to support 802.1Q in this case.

Now that we have some view of our global architecture, we will see how to secure individual network elements.

Device Security

Securing the Core Network

Each snmp agent, no matter where, must have an access list specifying what NMS it is allowed to talk to. This includes both the core network routers and the OSS/BSS servers themselves if they are managed and/or monitored by SNMP.

The security checklist would contain:

- Use SNMPv3 if the NMS supports it. Use SHA instead of MD5 if possible.
- Restrict the access to the SNMP agent on the IP level. On a Cisco router, we would create an access list and add it at the end of the `snmp-server` statement.
- Use well designed password for community strings (or shared secrets in the case of SNMPv3). Use software like `makepasswd` [MKPWD] to create these shared secrets: nobody "human" needs to remember them, they can be as complicated as possible.
- Do not forget to set authentication (via community strings for SNMPv2) for traps.
- Regularly change these shared secrets. We can either remove the previous community string and then add the new one, or add the new one and then remove the previous one. We have a synchronization problem here: either we disrupt the monitoring during a short period (after we removed the previous community string and before we put the new one) or we offer to someone who hacked the old community string the opportunity to get the new one (after we added the new community string and before we add the new one). The "right" answer mainly comes from the efficiency of the integration between your OSS elements (provisioning and monitoring).
- Restrict the views each NMS has access to (a view is a subset of the MIB tree). Do not give a NMS more access that it needs. We will show later in the document how to achieve this.

Securing the OSS/BSS segment

There are quite a few reasons to be extremely careful regarding the security of the OSS/BSS segment. First, as explained in the introduction, the threats are significant. Then, vendors typically do no put security very high in their concerns as they tend to consider that these servers are not directly exposed, as a web server would be for example. Vendors tend to use dangerous services on their servers like RPC and unsafe proprietary protocols to communicate between the various NMS elements.

A basic protection the OSS devices must have is a statefull firewall between the OSS segment and the MPLS network. From the OSS segment perspective, we would typically allow outgoing SNMP queries (the statefull firewall would only allow the relevant answer) and ingoing SNMP traps. The firewall itself cannot do much against malicious snmp traps, except filtering on the source IP address.

Other traffics we might want to authorize would be outgoing telnet connections (or better, ssh connections for the routers that support it), and specific traffic between NMS elements if we split our OSS segment for fault tolerance or disaster recovery purposes.

Like always, the rest must be dropped, logged, and the logs must be sent to the NMS for analysis.

Restricting the Views

There is no particular reason to enable general access all SNMP objects; some of them could easily be used for DoS attacks. For example, setting the OID .1.3.6.1.4.1.9.2.9.9.0 to 2 will reboot a Cisco switch. OID restrictions should be designed just the way everything is to be designed in security: if a community does not need access to a view, it should not have it.

ucd-snmp is a free snmp agent. If we are using it, we can restrict views in three steps using the `snmpd.conf` configuration file. We will document the configuration steps for SNMPv2; SNMPv3 is similar:

- Map the community name (Community) into a security name:
  `com2sec mysecurename 192.0.2.0/24 Community`
  `com2sec` indicates that we are building a mapping from a community name to a security name.
  `mysecurename` is the name of the community
  Community is the secret share between the agent and the NMS.
  A good way to set it is using programs like `makepasswd` [MKPWD]:
  `makepasswd --minchar 16 --maxchar 16`
  to get a strong password with sixteen characters.
- Map the security names into group names:
  `group MyGroup v2c mysecurename`
  `(group MyGroup usm mysecurename)`
- Define a view:
  `view MyGroup included interfaces.ifTable.ifEntry.ifIndex.1 ff.a0`
  Knowing that
  `interfaces.ifTable.ifEntry.ifIndex.1 = .1.3.6.1.2.1.2.2.1.1.1`
  and `ff.a0 = 11111111.10100000`, the mask `ff.a0` fixes the row index but lets `MyGroup` browse the field of the row and gives `MyGroup` access to
  - `.1.3.6.1.2.1.2.2.1.1.1` (Interface Index for IfIndex 1, so 1),
  - `.1.3.6.1.2.1.2.2.1.2.1` (Interface Description for IfIndex 1),
  - `.1.3.6.1.2.1.2.2.1.10.1` (Interface Incoming Octets for IfIndex)
  - and so on...

Now that we have ideas on the MPLS/BGP architecture, the way SNMP fits in it and the impact it has on the configuration of individual network elements, let us see how to monitor the security of this whole. To stay in a network-oriented perspective, we will only consider network Intrusion Detection Systems based on SNMP.

IDS: knowing what is going on

The firewall and routers logs sent to a central location are the very first step to take to know what is going on. The very next one is to get a warning from the routers in case of Access-List violation or configuration changes, using syslog or SNMP traps when available. SNMP traps feature a higher level of security than syslog messages, especially when using SNMPv3. The next step towards a true Intrusion Detection

System (IDS) approach is to install servers at the right place to monitor the SNMP traffic.

The IDS probes can send traps back to the NMS if it considers that the traffic is suspect. Snort supports this functionality [SNORTSNMP]. The main advantages of Snort are that it is both highly configurable and light. However, it would be overwhelmed by a 100 MB link: it is more advisable to place it near services (web, mail, database...) or in the OSS segment. Typically, we would use a server with two Network Interface Cards. One card without any IP address would listen to the traffic. The other one connected directly either on the Management VPN or on a specialized VPN via a management VLAN would bring the data back to the management system.

The purpose of the management VLAN is to provide a separate logical network to bring the valuable information from the IDS probes back to the management servers, and to manage the IDS probes themselves. This separation of traffic can be provided by VLANs, even if this solution is to be considered carefully. As a first precaution, the switches at each side of a trunk should not have any interface leading to a non-trusted network. Typically, we would use them between a PE and our services, with a firewall located between the PE and the switch (the Firewall would have to support 802.1Q). In most cases, a physical separation is more secure.

The Snort output plugin that can be used to generate traps is `SnmpTrapGenerator`. It supports both SNMPv2c and better, SNMPv3. The line to add in `snort.conf` would be:

```
output trap_snmp: alert, 7, trap -v 2c -p 162 myTrapListener myCommunity
```

if we wanted to send on port 162, using SNMP version 2c, an alert registered as coming from node 7. Again, if our NMS supports SNMP version 3, let us use it instead of version 2c.

Data from snort or from any IDS get far more value when it can be correlated among various probes to get a global understanding of what is going on. Moreover, alerts are much more likely to be looked at if they are centralized on a single screen, instead of having to log individually on various probes to get the results.

Conclusion

SNMP is definitely a great success in network management: it is simple, yet powerful and has proven it. However, it has to be used in a carefully designed architecture. BGP/MPLS VPNs features powerful functionalities that integrate nicely with SNMP security requirements; the point is to use them.

The same way SNMP is simple, BGP/MPLS has simple concepts that inherently protect the core network. Keeping things simple also makes sure that more people are likely to understand what the security engineers are trying to accomplish. This way, it would be less likely that some inexperienced engineer opens by inadvertence a large hole in our security architecture because he does not understand it.

Cited References

[CERT107186] "Multiple vulnerabilities in SNMPv1 trap handling", CERT Vulnerability Note VU#107186 (12 Feb 2002)
http://www.kb.cert.org/vuls/id/107186 (06 August 2002)

[CERT854306] "Multiple vulnerabilities in SNMPv1 request handling", CERT Vulnerability Note VU#854306 (02/12/2002)
http://www.kb.cert.org/vuls/id/854306 (06 August 2002)

[CISCOv3] "SNMPv3", Cisco Systems Inc, 7 March 200
http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120t/120t3/nmp3.htm (19 July 2002)

[HAC] Menezes, A., van Oorschot , P. and Vanstone , S., Handbook of applied cryptography, 1996, Ch 9. P 370
http://www.cacr.math.uwaterloo.ca/hac/ (06 August 2002)

[JUNIPER200001] Semeria, Chuck "Multiprotocol Label Switching - Enhancing Routing in the New Public Network", White Paper, Juniper Networks, Inc (2000)
http://www.juniper.net/techcenter/techpapers/200001.pdf (06 August 2002)

[MKPWD] Ingram, Johnie, "makepasswd 1.10-1", debian/GNU Linux
http://packages.debian.org/stable/admin/makepasswd.html (06 August 2002)

[RFC1067] Case, J., Fedor, Schoffstall, M., Davin, J. "Request for Comments 1067 - A Simple Network Management Protocol", University of Tennessee at Knoxville, NYSERNet, Inc., Rensselaer Polytechnic Institute, Proteon, Inc. (August 1988)
ftp://ftp.isi.edu/in-notes/rfc1067.txt (06 August 2002)

[RFC2274] Blumenthal, U., Wijnen, B., "Request for Comments: 2274 - User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", IBM T. J. Watson Research (January 1998)
ftp://ftp.isi.edu/in-notes/rfc2274.txt (06 August 2002)

[RFC2547] E. Rosen, Y. Rekhter, "Request for Comments: 2547 - BGP/MPLS VPNs", Cisco Systems, Inc. (March 1999)
ftp://ftp.isi.edu/in-notes/rfc2547.txt (06 August 2002)

[RFC2574] U. Blumenthal, B. Wijnen "Request for Comments: 2574 - User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", IBM T. J. Watson Research (April 1999)
ftp://ftp.isi.edu/in-notes/rfc2574.txt (06 August 2002)

[SNMPLS] Hallstein Lohne Johannes Vea, "Mechanisms for OAM on MPLS in large IP backbone networks" May 2002
http://www.siving.hia.no/ikt02/ikt6400/g20/Thesis.pdf (06 August 2002)

[SNORTSNMP] Martin Roesch, "Snort Users Manual - Snort Release: 1.9.x"
http://www.snort.org/docs/writing_rules/chap2.html#tth_sEc2.5.11 (06 August 2002)

[UCDSNMP] Carnegie Mellon University and contributors, "snmpd.conf man page", (08 Feb 2002)
http://net-snmp.sourceforge.net/man/snmpd.conf.html (06 August 2002)