Global Information Assurance Certification Paper

Introduction

The arsenal of tools available for both protecting and penetrating our environments is daunting in both quantity and ability. Some of these tools are highly specialized in purpose while other are multipurpose and serve as building blocks for a larger toolkit. One such tool is the network sniffer. Network sniffing, in its most general form, consists of intercepting frames from the network and viewing their contents. The ability to do this has been widespread for some time and was used by everyone from network administrators (who were troubleshooting problems) to crackers (who were pulling passwords and files from the wire). If we gauge our timeframe by the release dates of some of the more recent sniffing tools, it was only recently that network sniffing was publicized to be possible on a switched network. This delay in information dissemination led to the, still prevalent, train of thought that migrating your environment to a switched network would prevent anyone from sniffing your network traffic (unless, of course, they could connect to the backplane/uplink port on your switch). Needless to say, through the efforts of the security community (or the anti-security community, depending on your point of view), tools have surfaced which allow for network sniffing on switched networks. To gain a better understanding of this, we need a brief explanation of how non-switched networks work and how they are sniffed. I'll then move on to show how switched networks work and show how they are sniffed. I'll finish with a discussion of how to protect your traffic in both non-switched or switched environments.

Sniffing Non-switched Networks

In the non-switched network environment, we carry the concept of a network segment. A segment[1] is a network architecture that resides behind a router, bridge, hub or switch in which every node is directly addressable from every other node. In the non-switched environment, frames are handled in a broadcast manner. That is, when one node transmits a frame, it is 'seen' by every node on the segment. Each node, in turn, will briefly exam the frame to see if it is addressed to them. If not, it is discarded. However, if they are the intended recipient, they accept the frame for processing. I liken this to a (now uncommon) party-line phone call where a single phone call can simultaneously ring multiple phones. Everyone answers and, after determining that the call is not for them, hangs up (at least, they hang up if they respect the privacy of others sharing the party-line). For the purposes of this paper, Node B will be designated as the host to be used as a sniffing agent. Node A and Node C will represent the 'innocents' who are merely trying to communicate with each other. This is shown in Figure 1.

By: Steven Sipes
Sept. 10, 2000

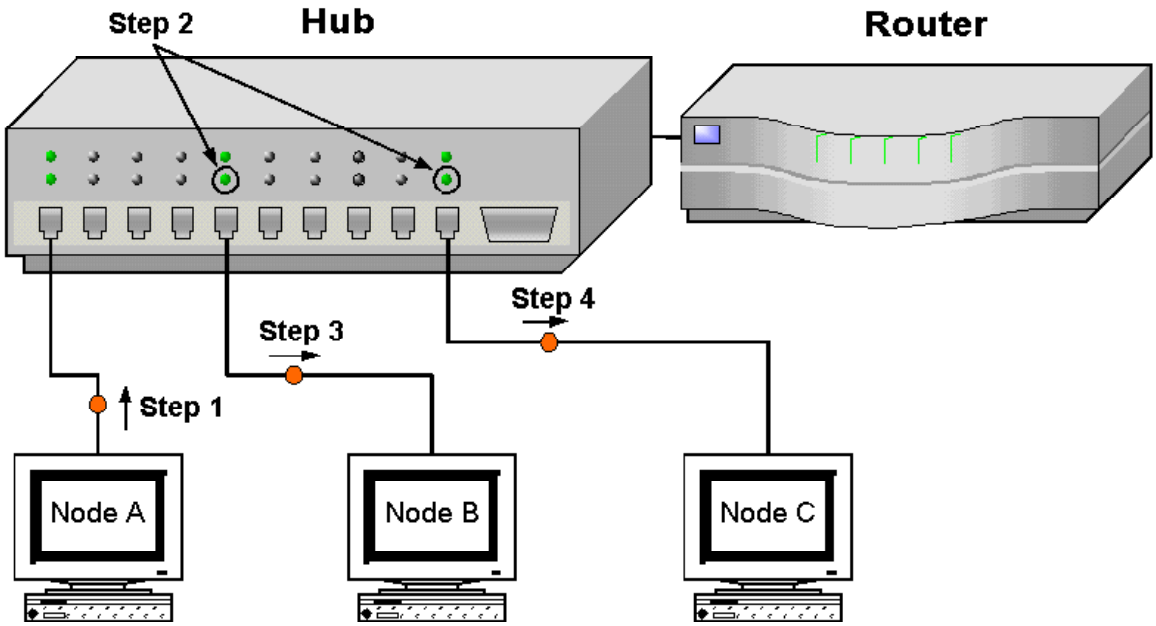

Figure 1

The normal flow of traffic in a non-switched network is as follows:

- **Step 1**: Node A transmits a frame to Node C.
- **Step 2**: The hub will broadcast this frame to each active port.
- **Step 3**: Node B will receive the frame and will examine the address in the frame. After determining that it is not the intended host, it will discard the frame.
- **Step 4**: Node C will also receive the frame and will examine the address. After determining that it is the intended host, it will process the frame further.

For completeness, it should be noted that steps 3 and 4 can be reversed in order, as the prediction as to which node will receive the frame first is beyond the scope of this document. For practical purposes, we can assume that they happen at the same time.

In order for a host to be used as a sniffing agent, the network interface must be set to 'promiscuous' mode. Setting this mode requires root or administrator access. After this mode is set, the network interface will no longer drop network frames which are addressed to other hosts. Rather, it will pass them up to the higher network layers with the expectation that some software at a higher layer will process them. In referencing Figure 1, the steps of this process would be as follows:

- **Step 1**: Node A transmits a frame to Node C.
- **Step 2**: The hub will broadcast this frame to each active port.
- **Step 3**: Node B will receive this frame and will accept it because the network interface has been set to 'promiscuous' mode. This allows a network interface to accept any frames, regardless of the MAC (Media Access Control) address in the frame. Even though the interface will save the frame, some higher level software is required to process the data.
- **Step 4**: Node C will also receive the frame and will process it as expected. It has no way of knowing that another host has also processed the frame.

Again, steps 3 and 4 can be transposed.

You can see that the non-switched environment lends itself quite nicely to sniffing. It requires little extra effort on the part of the sniffing agent since the hub broadcasts the frames to all active ports. Several sniffing utilities exist. Some of the publicly available tools used to sniff non-

switched networks include:

- ADMsniff - Sniffer for either Linux or SunOS[2]
- esniff - A platform independent sniffer[2]
- linsniffer - Linux specific ethernet sniffer[2]
- sniffer - Specializes in sniffing particular ports. Runs under Windows[2]
- sniffit - A network sniffer that run under LINUX, SunOS, Solaris, FreeBSD and IRIX.[3]
- snmpsniff - Specializes in sniffing SNMP date[2]
- solsniffer - Solaris specific ethernet sniffer[2]
- sunsniff - SunOS specific ethernet sniffer[2]
- websniff - Specializes in sniffing webserver login/auth information[2]

This list is by no means exhaustive.

Armed with this basic knowledge of how non-switched networks work and how they are sniffed, I'll now do a parallel comparison with switched networks.

Sniffing Switched Networks

In the switched network environment, we still carry the concept of a network segment, but the segment includes only the node and the switch. Data frames are handled in a direct manner. That is, frames from Node A to Node C are only sent across the circuits in the switch that are necessary to complete a connection between Node A and Node C. I liken this to a common person-to-person call in which Person A calls Person C. This is shown in Figure 2.

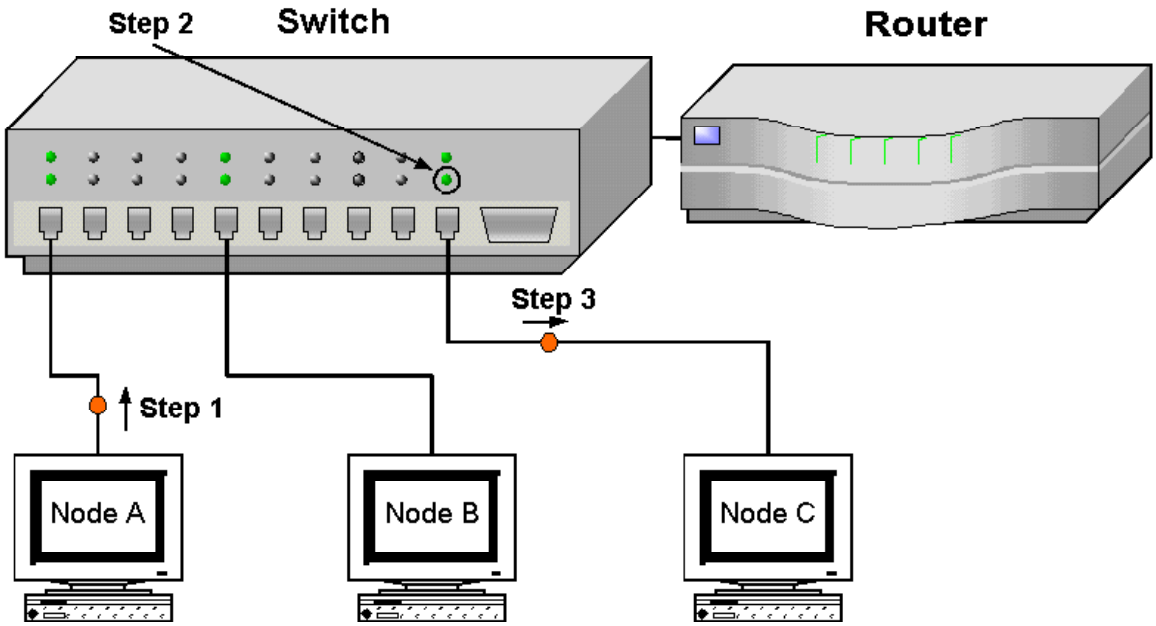

Figure 2

- **Step 1**: Node A transmits a frame to Node C.
- **Step 2**: The switch will examine this frame and determine what the intended host is. It will then set up a connection between Node A and Node C so that they have a 'private' connection.
- **Step 3**: Node C will receive the frame and will examine the address. After determining that it is the intended host, it will process the frame further.

Please note that the hosts still perform the examination of the destination address even though

the switch 'guarantees' that they are the intended host. While this may cause a very small amount of unnecessary overhead for the host, it is a necessary step because hosts may migrate from a switched to a non-switched environment (i.e. consider the case of a laptop).

This mode of operation carries some intrinsic benefits:

- Lower network traffic because we aren't broadcasting to each node which translates to a higher bandwidth through a reduction in the collision domain.
- Lower node processing overhead since the node only has to process frames that are meant for it.

However, there are some tradeoffs:

- Higher overhead processing on the switch since it must create, on the fly, virtual connections between machines.

As you can see, a switched network does not lend itself to sniffing as easily as a non-switched network does since it does not broadcast most frames. You'll also notice that under the 'intrinsic benefits' section, I did not list that the environment was more secure. The development of switched networks was driven by the need for more bandwidth, not for the need of more secure networks. Indeed, investigation reveals that several methods are available to sniff switched networks. Some of these methods are:

- ARP Spoofing[4]
- MAC Flooding[5]
- MAC Duplicating[4]

I'll briefly discuss these three options. Please note that there are more ways to sniff switched networks which are presented in the documentation of the mentioned tools. I simply chose to develop three of the cases as an introduction.

ARP Spoofing

One of the basic operations of the Ethernet protocol revolves around ARP (Address Resolution Protocol) requests and replies. In general, when Node A wants to communicate with Node C on the network, it sends an ARP request. Node C will send an ARP reply which will include the MAC address. Even in a switched environment, this initial ARP request is sent in a broadcast manner. It is possible for Node B to craft and send an unsolicited, fake ARP reply to Node A. This fake ARP reply will specify that Node B has the MAC address of Node C. Node A will unwittingly send the traffic to Node B since it professes to have the intended MAC address. Some available tools are specialized for sending fake ARP replies to classes of machines (i.e., NFS servers, HTTP servers, etc). One such tool is dsniff[5] and it works well to sniff for specific types of traffic. Other tools listen for the general ARP request and send the fake ARP reply at that time. The parasite[4] program falls into this category and it serves well to sniff the entire network. For this type of attack to work, we need the ability to forward on the frames we receive to their intended host. This is most commonly achieved through some type of IP forwarding, either at the kernel or application level.

MAC Flooding

Since switches are responsible for setting up the virtual circuits from one node to another, they must keep a translation table that tracks which addresses (specifically, which MAC addresses) are on which physical port. The amount of memory for this translation table is limited. This fact allows the switch to be exploited for sniffing purposes. On some switches, it is possible to bombard the switch with bogus MAC address data. The switch, not knowing how to handle the excess data, will 'fail open'. That is, it will revert to a hub and will broadcast all network frames to all ports. At this

point, one of the more generic network sniffers will work.

MAC Duplicating
It's not difficult to imagine that, since all frames on the network are routed based on their MAC address, that the ability to impersonate another host would work to our advantage. That's just what MAC duplicating does. You reconfigure Node B to have the same MAC address as the machine whose traffic you're trying to sniff. This is easy to do on a Linux box if you have access to the 'ifconfig' command. This differs from ARP Spoofing because, in ARP Spoofing, we are 'confusing' the host by poisoning it's ARP cache. In a MAC Duplicating attack, we actually confuse the switch itself into thinking two ports have the same MAC address. Since the data will be forwarded to both ports, no IP forwarding is necessary.

Protection

There are several methods to protect again these attacks. Some of these methods are applicable to both the non-switched and switched environments.

IP Filtering
By enabling IP filtering on your switch, you directly specify which traffic is allowed to flow to and from each port. This can be a monumental effort to put in place and manage, especially if your environment is dynamic.

Port Security
If your hub or switch has the ability to enable port security, this will help to protect you from both the MAC Flood and MAC Spoofing attacks. These feature effectively prevents the hub or switch from recognizing more than 1 MAC address on a physical port. This, like many security procedures, restricts the environment and amplifies the need for a management process as well as a auditing process.

Routing Security
Routing should only be performed by the designated routers. That is, no workstations should be allowed to run a routing protocol as they may be compromised. Also, it goes without saying that management of any of your network gear should be through a secure connection and not through telnet which passes the administrative login/password in cleartext.

Conclusion

I believe this examination provides another justification for 'defense-in-depth'. Networks are an infrastructure enabler for us to perform our daily functions. The general networks we use were never meant to be used as a security feature; although, they continue to be used in this manner. Providing a managed network infrastructure is a key component of any good defensive position. While compromising a single host may gain the attacker access to a few systems, the ability to sniff userids and passwords for several machines will effectively give away the keys to the kingdom. Network managers must be aware that they have 2 realistic options. They can either manage the network appropriately and be part of the team trying to protect the environment, or they can configure the environment so that it is 'hands-off' or 'self-maintaining' which, traditionally, translates into lax security.

References

1 "Webopedia" http://webopedia.internet.com/TERM/s/segment.html
2 "AntiCode Website" http://www.AntiOnline.com/cgi-bin/anticode/anticode.pl
3 "Insecure Website" http://www.insecure.org/
4 van Hauser. "Parasite 0.5" http://thc.inferno.tusculum.edu/files/thc/parasite-0.5.tar.gz

5 Song, Dug. "Dsniff" http://www.monkey.org/~dugsong/dsniff/

Mail Lists

Dsniff mailing list: Send an email to dsniff-request@monkey.org with 'subscribe' as the subject