Detection of Promiscuous Nodes Using ARP Packets

Version 1.0

31-Aug-01
Written by: Daiji Sanai <hyler@securityfriday.com>
Translated by: Kelvin King-Pang Tsang

http://www.securityfriday.com

Contents

Abstract 3
1. Introduction 3
2. The Principle of Sniffing 3
3. Basic Concepts of Promiscuous Node Detection 4
4. The Basics 4
1)Hardware Filter 4
Unicast 4
Broadcast 5
Multicast 5
All Multicast 5
Promiscuous 5
2)ARP Mechanism 5
5. Basics of Promiscuous Node Detection 6
6. Software Filter 7
1) Linux 7
2) Microsoft Windows 9
7. Promiscuous Detection 11
8. Promiscuous Node Detection 12
9. Exceptions 12
1) Old NIC's 12
2) 3Com NIC 13
3) Windows 2000 Packet Capture Driver 13

Abstract

On a local network, security is always taken into consideration. When plain text data is being sent onto the network, it can be easily stolen by any network user. Stealing data from the network is called sniffing. By sniffing the network, a user can gain access into confidential documents and cause intrusion into anyone's privacy. Many freely distributed software on the Internet provides this functionality. Despite the easiness of sniffing, there is no good way to detect such malicious act yet. This document explains the mechanism used by PromiScan, a piece of software that can effectively scan sniffers on the network. Sniffers work by receiving all packets being sent onto the network. To achieve this, all sniffers must set the Network Interface Card (NIC) of their PC's into a mode called "promiscuous mode". Then the NIC will blindly receive all packets and pass them to the system kernel. The Address Resolution Protocol (ARP) request packets are used to query hardware addresses from IP addresses. We make use of this kind of packet to verify whether the NIC's on the network are set to promiscuous mode. ARP request packets are used because it is available on all IPv4 based Ethernet. When a NIC is receiving all packets, packets that are not supposed to arrive to that PC are no longer blocked by the NIC. All packets are passed to the system kernel and, the system kernel may make mistake by responding to some packets that it is not supposed to respond. With the presence of the above mechanisms, we can compose fake ARP request packets and send them to every node on the network. These packets normally are blocked by the NIC's, but if some nodes respond to it, then some promiscuous NIC's exist. Those PC's with promiscuous NIC's are running sniffers. Thus sniffers can be successfully detected.

1. Introduction

In the local network, the act of sniffing has been a big thread. Malicious users can easily steal confidential documents and anyone's privacy by sniffing the network. Sniffing causes intrusion into privacy, but it can be done simply by downloading free sniffer software (sniffers) from the Internet and installing them into their personal computer. However, so far there is no good way to detect which PC's are sniffing the network. This documentation will discuss the use of Address Resolution Protocol (ARP) packets to effectively detect malicious users when they are sniffing the office's or the school's networks.

2. The Principle of Sniffing

The local network is usually composed of the Ethernet. On an Ethernet using IP protocol (IPv4), information is sent on the cable in plain text, unless an encryption program is used. When someone sends information onto the network, she expects someone on the other side of the network to receive that information. Unfortunately, the mechanism of Ethernet gives unauthorized people a chance to steal and look at the data. We know that an Ethernet based network works by sending messages to all nodes on the network, and it expects that only the intended node(s) will receive the messages. At the same time, the other nodes simply drop the messages. Whether to receive or drop the messages is controlled by the Network Interface Card (NIC). The NIC does not receive all the packets on the network although it is connected to the Ethernet; instead it filters out the

desired packets, which this specific computer should receive. For the rest of this document, we will call the filter of the NIC the Hardware Filter. Sniffing is done by setting the NIC of its own PC to a specific mode, such that the NIC will receive all data arriving to it, no matter whether it is the intended destination. This NIC mode is called the Promiscuous Mode.

3. Basic Concepts of Promiscuous Node Detection

Instead of sending out illegal packets, network sniffing is performed by receiving all packets. Since it does not interfere the network traffic at all, it is difficult to detect such behavior. Nonetheless, the state of the NIC in promiscuous mode is obviously different from that in normal mode. A packet that is supposed to be filtered by the hardware filter is now passed to the system kernel. As a result, whether to respond to the packet relies totally on the internal software.

Our way to detect promiscuous node can be demonstrated by an example from the real world. Imagine that there is a conference in a meeting room. Then sniffing the conference can be done by putting ones' ear against the wall of the meeting room. When she is sniffing, she wants to hold her breaths and quietly listen to all the conversations going on in the meeting room. However, if the name of the sniffer is called in the conference, "Miss XX?" the sniffer may occasionally make a mistake by responding to it, "Yes!" This analogy sounds a little ridiculous, but it can be applied to network sniffing. Since the sniffing node receives all the packets, including those that are not targeting to it, it may make mistakes such as responding to a packet, which originally is supposed to be filtered by the NIC. Therefore, our promiscuous node detection is performed by checking the responses of ARP packets, when ARP request packets are sent to all nodes on the network.

4. The Basics

1) Hardware Filter

First of all, let us begin with the differences between the NIC in promiscuous mode and in normal mode. All the NIC’s on the Ethernet are represented by a 6-byte hardware address. The manufacturer assigns this address such that each address is unique in the whole world. Theoretically, there are no two NIC's having the same hardware address. All communications on the Ethernet are based on this hardware address. The NIC, however, can set up different filters in order to receive different kinds of packets. The followings are a list of hardware filters:

Unicast
: Receive all packets having the same destination address as the hardware address of the NIC.

Broadcast

Receive all broadcast packets. Broadcast packets have destination address FF FF FF FF FF FF. The purpose of this mode is to receive the packets which are supposed to arrive at all nodes existing on the network.

Multicast

Receive all packets which are specifically configured to arrive at some multicast group addresses. Only packets from the hardware multicast addresses registered beforehand in the multicast list can be received by the NIC.

All Multicast

Receive all multicast packets. Since this mode may also correspond to other high-level protocols other than IPv4, All Multicast will receive all packets that have their group bit set.

Promiscuous

Receive all packets on the network without checking the destination address at all.

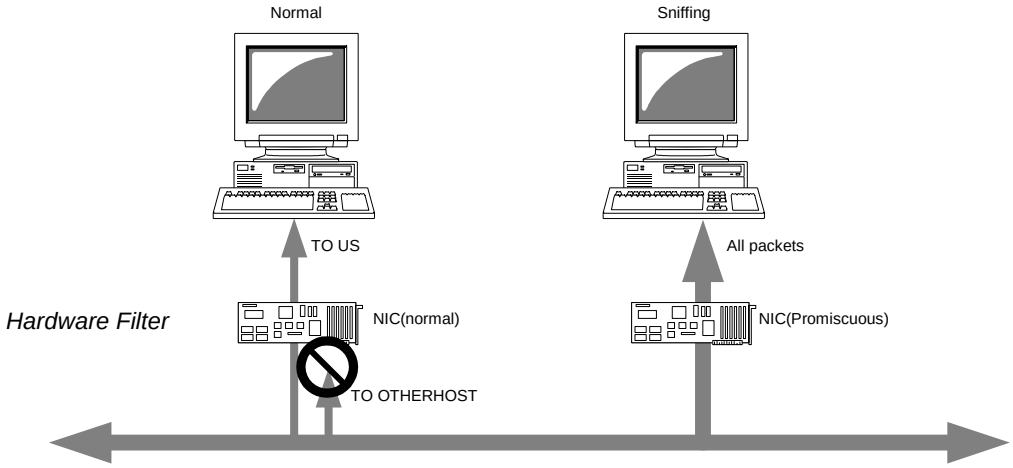

fig.1 Hardware Filter

fig.1 illustrates the operations of a hardware filter when it is in normal mode and when it is sniffing. Normally, PC's set their NIC hardware filter to unicast, broadcast and multicast address 1. They only receive packets that have its destination address set to the PC's own hardware address, broadcast address (FF FF FF FF FF FF), and multicast address 1(01 00 5E 00 00 01).

2) ARP Mechanism

On an Ethernet linked by IP addresses, packets are in fact sent and received based on hardware addresses. Packets cannot be sent by just using an IP address. Therefore, the Ethernet needs a mechanism that converts IP addresses into hardware addresses. At this time, Address Resolution Protocol(ARP) packets are used. ARP packets belong to the

link layer, which is the same layer as IP, so ARP packets does not affect the IP layer. Since IP addresses resolving is always available on an IP network, ARP packets become the suitable packets for testing the response of the nodes when detecting promiscuous nodes.

In the following example, we illustrate the operations of using an ARP packet to resolve an IP address:

> A PC (X) with IP address 192.168.1.1 and hardware address 00-00-00-00-00-01 wants to send messages to another PC (Y) with IP address 192.168.1.10. X will first compose an ARP request packet, which is used to query the hardware address corresponding to 192.168.1.10. The destination hardware address field of the ARP packet is set to a broadcast (FF-FF-FF-FF-FF-FF) such that all nodes in the local network will receive this packet. When each PC on the network receives this packet, it checks whether the IP address of the ARP packet is the same as its own. If they are different, this ARP packet is ignored. If they are the same, that PC will reply to the packet, along with its own hardware address and IP address. In this case, Y will send a reply packet to X, and X will cache this hardware/IP address pair. Since X successfully queried the hardware address of Y, X can begin sending the actual data.

5. Basics of Promiscuous Node Detection

As stated before, packets are filtered differently when the NIC is set to promiscuous mode and that to normal mode. When the NIC is set to promiscuous mode, packets that are supposed to be filtered by the NIC are now passed to the system kernel. By using this mechanism, we come up with a new way to detect promiscuous nodes: if we configure an ARP packet such that it does not have broadcast address as the destination address, send it to every node on the network and discover that some nodes respond to it, then those nodes are in promiscuous mode.

Here is a walk-through of the correct request/respond operations of ARP. First of all, an ARP packet is generated in order to resolve 192.168.1.10. Its destination address is set to the broadcast address such that all nodes on the network can receive it. Theoretically, only one node with exactly the same IP address will respond to it.

Then, what about the ARP packet destination is set to a different address other than the broadcast address? For instance, what will happen if we set the destination address to 00-00-00-00-00-01? When the NIC is in normal mode, this packet is considered to be "to other host" packet, so it is refused by the hardware filter of the NIC. However, when the NIC is in promiscuous mode, the NIC does not perform any filter operation. Then this packet is able to pass to the system kernel. The system kernel assumes that this ARP request packet arrives because it contains the same IP address as that PC, so it should respond to the packet. To our surprise, the kernel indeed will not respond to the packet(fig.2). This unexpected result shows that there exists some sort of filter in the software, because a packet is actually filtered again by the system kernel. For the time being, we will call this the Software Filter.

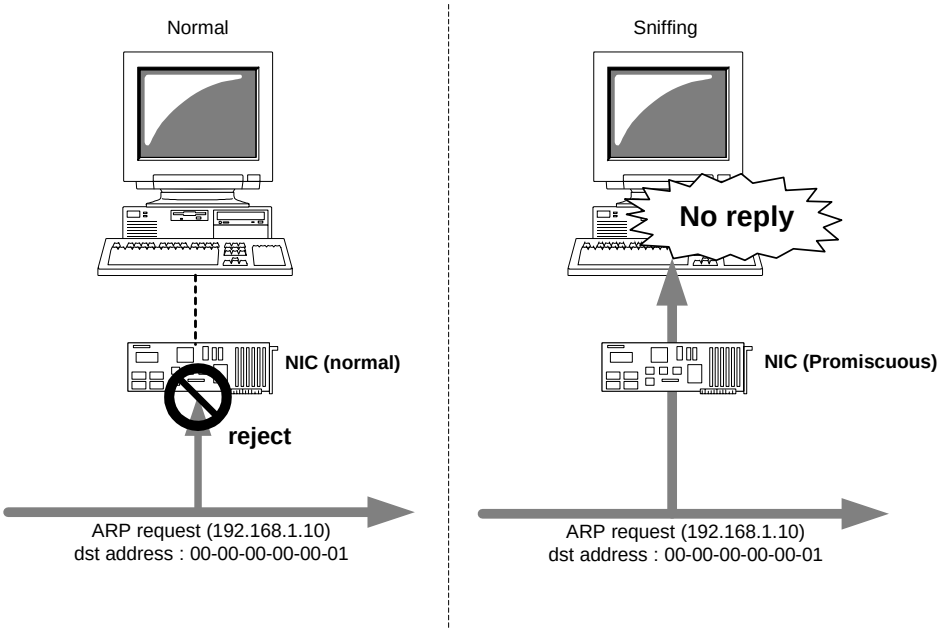

fig.2 Basics of Promiscuous Node Detection

To take one step further, the detection of promiscuous NIC can be achieved by observing the differences between the hardware filter and software filter. The hardware filter usually blocks invalid packets (those are not supposed to arrive to the system kernel). So if a packet can passes the hardware filter, it usually passes the software filter as well. Now, we want to compose packets that are supposed to be blocked by the hardware filter and, at the same time, are able to pass the software filter. By sending such a packet to a node, a NIC in normal mode will not respond. Instead if the NIC is in promiscuous mode, it will respond.

6. Software Filter

The software filter depends on the operating system kernel, so understanding how the software filter of the system kernel works is necessary. Since Linux is open-source, its software filter mechanism can be obtained. However, since the source code of Microsoft? Windows is unpublished, the software filter mechanism can only be guessed by experiments.

1) Linux

In the Ethernet module of Linux, packets are classified depending on the hardware address.

Broadcast packets:
FF FF FF FF FF FF

Multicast packets:
All packets having the group bit set, except the broadcast packets.

TO_US packets:
All packets having the same destination address as the hardware address of the NIC.

OTHERHOST packets:
All packets having a different destination address ad the hardware address of the NIC.

Here, we assume that all the packets with the group bit set are multicast packets. The Ethernet multicast packet corresponding to the IP network is in the form 01-00-5E-xx-xx-xx, and originally, the multicast packets cannot be classified by just verifying the group bit. However, this assumption is not a mistake. It is because 01-00-5E-xx-xx-xx is an IP-based multicast address, but the NIC hardware address is also used by other upper-level protocols.

Next, let us look at the ARP module used by Linux. The ARP module first rejects all the OTHERHOST packets. Then it will respond to the Broadcast, Multicast, and TO_US packets. Table1 illustrates the responses of the hardware and the software filters. We demonstrate by showing how the hardware and software filters operate when six kinds of hardware address are sent to the NIC:

	gr bit	normal mode			promiscuous mode		
		hw filter	sw filter	res.	hw filter	sw filter	res.
to_us	off	→	→	✓	→	→	✓
other host		reject	-	-	→	reject	-
broadcast	on	→	→	✓	→	→	✓
multicast (in the list)		→	→	✓	→	→	✓
multicast (not in the list)		reject	-	-	→	→	✓
group		reject	-	-	→	→	✓

Table 1 Software filter of Linux

TO_US packets:
When the NIC is in normal mode, all the TO_US packets can pass the hardware filter. And they can pass the software filter too, so the ARP module will respond to the packets regardless of whether the NIC is in promiscuous mode.

OTHERHOST packets:

When the NIC is in normal mode, it rejects the OTHERHOST packets. Even when the NIC is in promiscuous mode, the software filter rejects those packets. So there will be no response to ARP requests.

BOARDCAST packets:

In normal mode, BOARDCAST packets pass both hardware and software filters. So there will be a response regardless of the mode of the NIC.

MULTICAST packets:

In normal mode, packets with hardware address not registered in the multicast list are rejected. But if the NIC is in promiscuous mode, this kind of packets will pass the hardware filter even if the hardware address is not registered in the multicast list. And, due to the fact that the software filter does not reject multicast packets, a response will be obtained. In this case, since a different result will be obtained when a same packet is sent to a NIC in normal mode and that in promiscuous mode, this kind of packets will be used for promiscuous node detection.

Group bit packets:

These are the packets that are neither BROADCAST nor MULTICAST packets, but with the group bit set. In normal mode, the hardware filter rejects these kinds of packets but in promiscuous mode, these packets are passed. And since these kinds of packets are classified as multicast packets by the software filter, they are able to pass the software filter. These group bit packets can be used to detect promiscuous nodes.

2) Windows

Windows is not an open-source operating system, so we cannot analyse its software filter behaviour by examining its source code. Instead we perform experiments testing the software filter of Windows. The following seven kinds of hardware addresses are used:

FF-FF-FF-FF-FF-FF broadcast address:

All nodes should receive this kind of packet and respond because it is a broadcast address. A usual ARP request packet uses this address.

FF-FF-FF-FF-FF-FE fake broadcast address:

This address is a fake broadcast address missing the last 1 bit. This is to check whether the software filter examines all bits of the address and whether it will respond.

FF-FF-00-00-00-00 fake broadcast 16 bits:

This address is a fake broadcast address in which only the first 16 bits are the same as the broadcast address. This may be classified as a broadcast address and replied when the filter function only checks the first word of the broadcast address.

FF-00-00-00-00-00 fake broadcast 8 bits:

This address is a fake broadcast address in which only the first 8 bits are the same as the broadcast address. This may be classified as a broadcast address and replied when the filter function only checks the first byte of the broadcast address.

01-00-00-00-00-00 group bit address

This is an address with only the group bit set. This is to check whether this address is considered as a multicast address as Linux does.

01-00-5E-00-00-00 multicast address 0

Multicast address 0 is usually not used. So we use this as an example of a multicast address not registered in the multicast list of the NIC. The hardware filter should reject this packet. However, this packet may be misclassified to be a multicast address when the software filter does not completely check all bits. The system kernel thus may reply to such packet when the NIC is set to promiscuous mode.

01-00-5E-00-00-01 multicast address 1

Multicast address 1 is an address that all hosts in the local network should receive. In the other word, the hardware filter will pass this kind of packets by default. But it is possible that the NIC does not support multicast mode and does not respond. So this is to check whether the host supports multicast addresses.

Results:

The test results of the experiment using the 7 addresses are listed in table 2. The tests are performed against Windows 95, 98, ME, 2000 and Linux. As expected, all kernels respond to the broadcast address and multicast address 1 when the NIC is in normal mode.

However, when the NIC is set to promiscuous mode, the results are OS dependent. Windows 95, 98 and ME responds to the fake broadcast 31, 16, and 8 bits. So we may say that the software filter of Windows 9x series determines the broadcast address by checking only 1 bit.

In the case of Windows 2000, it responds to fake broadcast 31 and 16 bits. So we may conclude that the software filter of Windows 2000 determines the broadcast address by checking 8 bits.

In the case of Linux, it responds to all seven kinds of hardware address. In the other words, Linux responds to all seven kinds of hardware address when the NIC is set to promiscuous mode.

HW Address	Windows9x/ME		Windows2k/NT4		Linux2.2/2.4	
	normal	promisc	normal	promisc	normal	promisc
FF:FF:FF:FF:FF:FF	✓	✓	✓	✓	✓	✓
FF:FF:FF:FF:FF:FE	-	✓	-	✓	-	✓
FF:FF:00:00:00:00	-	✓	-	✓	-	✓
FF:00:00:00:00:00	-	✓	-	-	-	✓
01:00:00:00:00:00	-	-	-	-	-	✓
01:00:5E:00:00:00	-	-	-	-	-	✓
01:00:5E:00:00:01	✓	✓	✓	✓	✓	✓

Table 2 Result

7. Promiscuous Detection

The results so far prove that we can use ARP packets to determine a promiscuous node, whether the systems are running Windows or Linux. Thus, in a similar manner, this detection method can be applied to the local network. Here is the procedure:

1)We want to check whether the machine with IP address (A) is in promiscuous mode, we compose an ARP packet. An ARP packet has the following format:

Ethernet address of destination	FF FF FF FF FF FF
Ethernet address of sender	00 11 22 33 44 55
Protocol type (ARP = 0806)	08 06
Hardware address space (Ethernet = 01)	00 01
Protocol address space (IPv4 = 0800)	08 00
Byte length of hardware address	06
Byte length of protocol address	04
Opcode (ARP request = 01, ARP reply = 02)	00 01
Hardware address of sender of this packet	<Own NIC's Device Address>
Protocol address of sender of this packet	<Own PC's IP Address>
Hardware address of target of this packet	00 00 00 00 00 00
Protocol address of target	<IP Address (A)>

- Ethernet address of destination is the destination address of the Ethernet packet. In this case, it is the hardware address of the NIC of the target. The destination of ARP packet should be set to the broadcast address FF FF FF FF FF FF, because we want all hosts to receive this packet. And we want one host to reply if its IP address corresponds to the one the ARP packet is querying. But, as stated before, we want to

compose a packet that is supposed to be blocked by the hardware filter and is able to pass the software filter, we will use FF FF FF FF FF FE instead.

- Ethernet address of sender is the hardware address of the sender. For instance, 00 11 22 33 44 55 is a six-byte hardware address.
- Protocol type is 08 06 when this is an ARP packet.
- Hardware address space is 01 when the Ethernet is being used.
- Protocol address space is 08 00 when IPv4 protocol is being used.
- Byte length of hardware address is the length in byte of the hardware address. In this case, it is 06.
- Byte length of protocol address is the length in byte of an IPv4 address. In this case, it is 04.
- Opcode is 00 01 when it is an ARP request packet.
- Hardware address of sender of this packet is the PC's hardware address, for example, it can be set to 00 11 22 33 44 55.
- Protocol address of sender of this packet is the 4-byte IP address of the sender's PC.
- Hardware address of target of this packet is 00 00 00 00 00 00 because it is currently unknown. (The purpose of ARP request packet is to query this field)
- Protocol address of target is the 4-byte IP address of the node that is being checked whether it is in promiscuous mode.

2)After we compose this packet, we can send it onto the network.

3)Now, this packet is supposed to be blocked by the hardware filter of the target machine. However, if that machine is in promiscuous mode, this packet will pass the hardware filter and the software filter will respond. If we receive a respond, then that machine is in promiscuous mode.

8. Promiscuous Node Detection

To detect all the promiscuous nodes present on the local network, we apply the technique described in 7. to all nodes on the network sequentially. If there exists some machines that cannot be reached by ARP packets, then this method of promiscuous detection cannot be achieved.

9. Exceptions

Here we present some exceptions that promiscuous detection cannot be used:

1) Old NIC's

Some old NIC's do not support multicast list. For example, 3COM's EtherlinkIII does not support multicast list. Packets are possible to reach the software filter without being checked by the hardware filter. Due to the fact that the packets we are sending out have their group bit set, it is not possible for this kind of NIC's to distinguish between a promiscuous detection packet and a multicast packet. If such situation happens, they should be replaced by new NIC's.

2) 3Com NIC

When 3Com's 3c905 series NIC's are installed on Linux machines, they are set to all multicast mode by default, so it is not possible for us to distinguish multicast mode from promiscuous mode. The appearance of this exception is due to the fact that the driver provided by Linux does not support multicast list, and the NIC's become all multicast mode by default. Notice that 3c59x.o is chosen to be the driver of such NIC's by the Linux installer. If such case happens, edit the line in /etc/modules.conf (/etc/conf.modules) and change the driver to 3c905x.o, then this problem will be resolved.

3) Windows 2000 Packet Capture Driver

This exception arises when Windows 2000 packet capture driver is dynamically loaded. WinPcap2.1 (2.01 is different) and SMS are two dynamically loaded drivers. When they are installed on Windows 2000 system, they have special responses. The NIC will respond to the fake broadcast 16 bit even if it is not in promiscuous mode (Sniffers with those drivers will not operate accurately as well). In the other words, sniffers can be detected even if it is not being run. Microsoft may create this exception intentionally in order to detect promiscuous nodes.

Copyright© 2001,SecurityFriday.com, all right reserved.