



Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permitted without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Security Essentials Bootcamp Style (Security 401)"
at <http://www.giac.org/registration/gsec>

Sniffer Detection Tools and Countermeasures

This paper focuses on tools designed specifically for detecting network interface cards in promiscuous mode and on some countermeasures that lessen their effectiveness. To avoid both a false level of confidence in network integrity and unnecessary panic, system administrators should be familiar with the capability and limitations of these tools and countermeasures.

Detection at Local Host:

Unique tools for detection at the local host level exist for different operating systems.

For most versions of UNIX, use "ifconfig". "ifconfig" will tell the user whether the network interface card is in promiscuous mode or not. However, since it is usually trojanized during an unauthorized sniffer installation, its output may not be reliable. Other key utilities that a system administrator can use to detect the presence of a sniffer, such as "ls," "df," "du," "ps," "find," and "netstat," are typically trojanized during a compromise as well. A number of popular, publicly available trojanization tools have their configuration information in ASCII files under the "/dev/" directory. There should be no ASCII files in the "/dev/" directory, and a systems administrator should look for them because their presence is a sign of an intrusion during which a sniffer has been installed on the system. The configuration information contained within the ASCII file typically consists of processes and files, such as sniffers and their associated output files, to be hidden from the system administrator.

For most versions of UNIX, use lsof: [lsof](#) (LiSt Open Files), while not designed specifically for detecting sniffers, can find sniffers by finding large open files. Since intruders frequently send sniffer output to a file that grows quite large while the sniffer is running, lsof can be used to detect the presence of a sniffer by detecting the sniffer's output file. Since many sniffers write the string "TCP/IP" to their output files, a system administrator can pipe lsof output to "grep" to cut down on false positives.

For BSD, use cpm. cpm (check promiscuous mode) is a tool developed by CERT/CC in response to a large number of sites reporting compromised usernames and passwords as well as key binaries being trojanized in 1994. The associated advisory can be found at <http://www.cert.org/advisories/CA-1994-01.html>. cpm uses socket(2), ioctl(2) to read whether the network card (or cards if multihomed) have been set in promiscuous mode and then reports the results to the console. The program will list to the console only those devices found in promiscuous mode.

For SunOS 5.5 and 5.6 use ifstatus. ifstatus is available at <ftp://ftp.cerias.purdue.edu/pub/tools/unix/sysutils/ifstatus/>. This program reports to the console the flags of network interface cards, indicating which cards are in debug or promiscuous mode.

For NT there is no known tool. Regrettably, there is no publicly available tool known by the author that will test for promiscuous mode at the host level in Microsoft operating systems. Compounding this problem is the fact that the tools that are part of the Microsoft operating system may be trojanized; there is a publicly available "alpha" version of an NT "rootkit" (www.rootkit.com). There are very few remote Administrator-level access vulnerabilities, and this lack of vulnerabilities may be one reason there has not been more development to date on sniffers or sniffer detection tools for Microsoft.

Detection at Local Network Segment (non-switched):

Watch DNS Traffic. System administrators can also run their own sniffer, such as tcpdump, Windump, or snoop, and look for a large amount of DNS traffic from a host on the network. Typically, sniffers automatically perform DNS queries on IPs, since system administrators often give key servers names that denote host functionality. For example, the mail server at a site might be named mail.foo.com, and the DNS server might be ns.foo.com. Therefore, intruders are sometimes able to gain additional information about a network simply by having the sniffer perform DNS queries.

Use Antisniff. Antisniff, available at www.l0pht.com/antisniff, is the only commercially available production level tool for detecting sniffers at the network level. According to documentation by L0pht, Antisniff works by sending out crafted frames to elicit responses expected by systems in a promiscuous mode. Antisniff uses three different types of tests: Operating System specific tests, DNS tests, and System latency tests.

Antisniff Operating System Specific Tests: When in promiscuous mode, operating systems will respond to certain types of packets that they otherwise would not. Operating System specific tests by Antisniff send a packet with a non-existent Ether address with either a broadcast or unicast address and listens for a response. By sending an ICMP echo request inside the bogus Ethernet frame, the target system responds when in promiscuous mode and correctly ignores the packet when not. For testing for Linux systems in promiscuous mode, a unicast packet is sent using a non-existent Ether address. For BSD, a packet with a non-existent Ether address is sent to the multicast address. When in promiscuous mode, NT incorrectly checks only the first octet of the Ether address against the IP address, so it will respond if the packet is crafted with an Ether address of ff:00:00:00:00:00. False positives with Antisniff may exist for some drivers on NT systems; however, it may be that not all vendors model Microsoft's default NIC behavior.

Antisniff DNS Tests: For this test Antisniff sends information about an IP and listens for any DNS resolution requests from a host on the network segment for reasons mentioned above.

Antisniff Latency tests: According to Antisniff documentation, latency tests are the most thorough and most resource-intensive tests. Antisniff sends out unicast ICMP packets and times the response of the systems. Unlike a host in normal mode, a system in promiscuous mode does not filter packets at the network interface card level. An increase in network traffic will affect a promiscuous host much more than one in normal

mode, since the promiscuous system isn't relying on hardware level traffic filtering; therefore, systems running sniffers can be detected by their noticeably longer response time.

Countermeasures:

Patch software and remove any services not needed. Check vendor and computer security sites such as [CERT/CC](#), [Securityfocus](#), and [SANS](#), for news about the latest vulnerabilities, patch releases or other countermeasures, and for security configuration guides.

Check key binaries routinely. Since sniffer installation by an intruder is usually accompanied by trojanization of key binaries, system administrators should routinely check the system's integrity, using tools such as [tripwire](#) or anti-virus software, in accordance with the proscribed procedures from the vendor.

Use a switched network. A switched network is designed to negate packet collision at each host by having the local hub deliver only broadcast packets to all devices on the network, and to deliver packets destined for a particular host to that host only. Sniffers are not as effective on switched networks since unicast traffic received by the switch, tsuch as telnet, ftp, or smtp (mail), is directed only to the destination host. An intruder can force a switch to act as a dumb hub sending all traffic to all hosts on the network by using Address Resolution Protocol (ARP) spoofing and/or ARP overloading. Switches are designed to take ARP updates from hosts. By flooding a switch, which has a memory limit, with ARP packets, the network will be placed back in full broadcast mode and all hosts will get copies of packets sent from the switch thus enabling an intruder to obtain addressing information otherwise not available to him/her. The information can be used by the intruder to redirect traffic meant for another host on the network by updating the switch with a forged Ethernet address of the intended recipient. The sniffing host can even avoid suspicion by relaying redirected information back to the intended host. ARP spoofing tools, such as [dsniff](#) and [parasite](#), are publicly available to aid intruders.

Disable kernel loading. To hide the presence of a sniffer at the host level, an intruder can modify the system binaries or the kernel itself using a loadable kernel module. A system administrator can prevent kernel loading by building a static kernel that has had its module loading capability removed,. While an intruder with access to a compiler and source code can recompile the kernel, this is very unlikely because of the time and effort required to do it.

Use encryption. Avoid protocols that send information using clear text. Encrypted authentication, done through use of programs such as [secure shell and secure copy](#) and protocols such as IPv6, provide not only secure authentication but also session content confidentiality.

Use one-time passwords. Although they cannot protect against sniffers collecting certain types of information such as mail, use of one-time passwords can defeat sniffers

collecting usernames and passwords. Both hardware and software one-time password systems are available. See <http://www.cert.org/advisories/CA-1994-01.html> for more information.

Conclusions:

There is no one single defense available that will negate either the installation of or effectiveness of unauthorized sniffers. Tracking and applying vendor patches is not enough. System administrators should take all reasonable steps to make unauthorized sniffing difficult by addressing network design, monitoring the network, following security bulletins, and understanding tool use and limitations.

Sources:

CERT/CC. CERT/CC Advisory 1994-01 Ongoing Network Attacks
<http://www.cert.org/advisories/CA-1994-01.html> 2 Feb 95 (13 Oct 2000)

CERT/CC. Tools that Aid in Detecting Intrusions
<http://www.cert.org/security-improvement/implementations/i042.07.html> 12 Sep 2000
(9 Oct 2000)

Able, Vic. Isof <ftp://vic.cc.purdue.edu/pub/tools/unix/lsf/README> 22 Aug 2000 (13 Oct 2000)

Curry, David. ifstatus <ftp://ftp.cerias.purdue.edu/pub/tools/unix/sysutils/ifstatus/README.local>,
12 Oct 1995 (9 Oct 2000)

CERT/CC. cpm <ftp://ftp.cerias.purdue.edu/pub/tools/unix/sysutils/cpm/cpm.1.2.tar> 1 Feb 1996 (9 Oct 2000)

Rootkit, <http://www.rootkit.com/> (13 Oct 2000)

L0pht Heavy Industries. Antisniff Technical Documentation <http://www.l0pht.com/antisniff/tech-paper.html>, 19 Jul 1999 (9 Oct 2000)

Song, D. dsniiff <http://www.monkey.org/~dugsong/dsniiff/> (18 Oct 2000)

Van Hauser, parasite <http://thc.pimmel.com/thc/parasite-0.5.tar.gz> (18 Oct 2000)

Creed (pseud.). Knark Kernel Loadable Module <http://packetstorm.securify.com/mag/b4b0/b4b0-09.txt>
B4B0 Issue#9, 1999 (Oct 13 2000)

Spoon (pseud.). Linux Capabilities (LCAP) <http://pweb.netcom.com/~spoon/lcap/> 22 Dec 1999 (11 Oct 2000)

Test questions for Sniffer Detection and Counteracting

1. Antisniff uses forged DNS packets to find sniffers on a local network. (False)
2. Antisniff is designed to detect sniffers regardless of the operating system running the sniffer. (True)
3. Sniffers collect just usernames and passwords (False)
4. Kernel loading is enabled by default in Linux (True)
5. A switched network will defeat sniffing on that network (False).
6. Antisniff uses which types of tests to detect sniffers: (B)
 - A. Sending ICMP with forged MAC Addresses, sending DNS Query Request, and sending ICMP Echo Requests
 - B. Sending ICMP with forged MAC Addresses, sending bogus packets addressed to a particular IP and listening for DNS Query, and sending ICMP Echo Requests.
 - C. Sending ICMP with TTLs set to the same value of the MAC address of the target system, sending bogus packets addressed to a particular IP and listening for DNS Query, and sending ICMP Echo Requests.
 - D. Sending ICMP with forged MAC Addresses, sending bogus packets addressed to a particular IP and listening for DNS Query, and sending UDP with TTLs decrementing by one after each packet is sent
7. Some tools used to detect signs of a sniffer being installed are: (D)
 - A. LCAP, lsof, cpm
 - B. Lsof, switched network, cpm
 - C. Antisniff, knark, cpm
 - D. Cpm, antisniff, lsof
8. lsof will run on which operating system(s): (C)
 - A. Linux
 - B. NT
 - C. Most versions of unix operating systems
 - D. None of the above
9. cpm will run on which operating system(s): (C)
 - A. NT
 - B. SunOS
 - C. BSD
 - D. Novell
10. lsof is designed to: (C)
 - A. test for vulnerabilities
 - B. detect Buffer overflow attempts
 - C. List open files
 - D. None of the above

© SANS Institute 2003, Author retains full rights.

Upcoming Training

Click Here to
{Get CERTIFIED!}



SANS Prague 2017	Prague, Czech Republic	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Salt Lake City 2017	Salt Lake City, UT	Aug 14, 2017 - Aug 19, 2017	Live Event
Community SANS Omaha SEC401*	Omaha, NE	Aug 14, 2017 - Aug 19, 2017	Community SANS
SANS New York City 2017	New York City, NY	Aug 14, 2017 - Aug 19, 2017	Live Event
Community SANS Trenton SEC401	Trenton, NJ	Aug 21, 2017 - Aug 26, 2017	Community SANS
Virginia Beach 2017 - SEC401: Security Essentials Bootcamp Style	Virginia Beach, VA	Aug 21, 2017 - Aug 26, 2017	vLive
SANS Chicago 2017	Chicago, IL	Aug 21, 2017 - Aug 26, 2017	Live Event
SANS Virginia Beach 2017	Virginia Beach, VA	Aug 21, 2017 - Sep 01, 2017	Live Event
SANS Adelaide 2017	Adelaide, Australia	Aug 21, 2017 - Aug 26, 2017	Live Event
Community SANS Pasadena SEC401 @ NASA	Pasadena, CA	Aug 23, 2017 - Aug 30, 2017	Community SANS
Mentor Session - SEC401	Minneapolis, MN	Aug 29, 2017 - Oct 10, 2017	Mentor
SANS Tampa - Clearwater 2017	Clearwater, FL	Sep 05, 2017 - Sep 10, 2017	Live Event
SANS San Francisco Fall 2017	San Francisco, CA	Sep 05, 2017 - Sep 10, 2017	Live Event
Mentor Session - SEC401	Edmonton, AB	Sep 06, 2017 - Oct 18, 2017	Mentor
SANS Network Security 2017	Las Vegas, NV	Sep 10, 2017 - Sep 17, 2017	Live Event
Mentor Session - SEC401	Ventura, CA	Sep 11, 2017 - Oct 12, 2017	Mentor
Community SANS Albany SEC401	Albany, NY	Sep 11, 2017 - Sep 16, 2017	Community SANS
Community SANS Dallas SEC401	Dallas, TX	Sep 18, 2017 - Sep 23, 2017	Community SANS
Community SANS Columbia SEC401	Columbia, MD	Sep 18, 2017 - Sep 23, 2017	Community SANS
SANS Baltimore Fall 2017	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Copenhagen 2017	Copenhagen, Denmark	Sep 25, 2017 - Sep 30, 2017	Live Event
Community SANS Boise SEC401	Boise, ID	Sep 25, 2017 - Sep 30, 2017	Community SANS
Baltimore Fall 2017 - SEC401: Security Essentials Bootcamp Style	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	vLive
Community SANS New York SEC401	New York, NY	Sep 25, 2017 - Sep 30, 2017	Community SANS
Rocky Mountain Fall 2017	Denver, CO	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS London September 2017	London, United Kingdom	Sep 25, 2017 - Sep 30, 2017	Live Event
Community SANS Sacramento SEC401	Sacramento, CA	Oct 02, 2017 - Oct 07, 2017	Community SANS
SANS DFIR Prague 2017	Prague, Czech Republic	Oct 02, 2017 - Oct 08, 2017	Live Event
Community SANS Charleston SEC401	Charleston, SC	Oct 02, 2017 - Oct 07, 2017	Community SANS
Mentor Session - SEC401	Arlington, VA	Oct 04, 2017 - Nov 15, 2017	Mentor