Published on **O'Reilly Network** (http://www.oreillynet.com/)
http://www.oreillynet.com/pub/a/network/2002/04/26/nettap.html
See this if you're having trouble printing code examples

Network Forensics: Tapping the Internet

by Simson Garfinkel, author of Web Security, Privacy & Commerce, 2nd Edition
04/26/2002

During the Gulf War, computer hackers in Europe broke into a UNIX computer aboard a warship in the Persian Gulf. The hackers thought they were being tremendously clever -- and they were -- but they were also being watched.

Just before penetrating the PACFLEETCOM computer and reading the Navy's email, the hackers hopped through a computer at Los Alamos Laboratory. And unknown to the attackers, every packet in or out of Los Alamos over the Laboratory's Internet connection was recorded and preserved for later analysis on magnetic tape.

The incident in the Persian Gulf became a cause celebre in the years that followed. Tsutomu Shimomura bragged about the incident in his book *Takedown*. Many experts in the field of computer security used the story as proof, of sorts, that the U.S. military was asleep at the switch when it came to computer security.

One of the more dramatic outcomes of the incident was a videotape played at the annual meeting of the American Association for the Advancement of Science in February 1993 -- a video that showed each of the attacker's keystrokes, replete with mistakes, and the results, as he systematically penetrated the defenses of the ship's computer and scavenged the system.

I was one of the journalists in the audience watching the 1993 video. At first, I was incredulous -- how could a national laboratory possibly record every bit of data moving back and forth over the Internet connection? But then I did the math.

In 1991 the lab most likely had a T1 link, which transmits at most 1.544 million bits each second. Multiplying that by 60 seconds per minute, 60 minutes per hour, and 24 hours a day, comes to 133 gigabits, or a little less than 17 gigabytes. In 1993, that was a lot of data, but not an impossibly large amount. Certainly a big national lab like Los Alamos could hire somebody to load a new DAT tape every two hours, if that was what was required to archive all the information.

In the decade that followed the Gulf War, Moore's law had its way not only with processors, but with bandwidth and storage as well -- but each unequally. While the clock on the average workstation surged from 25 Mhz to 1.1 Ghz, and while the typical "big" hard drive jumped from a few hundred megabytes to 160 GB, bandwidth increased at a comparatively modest rate -- from 28.8 kbps to 384 kbps for many homes and small businesses. Even today, few businesses have more than a T1's worth of Internet bandwidth.

These trends are accelerating. For the foreseeable future, both the amount of information that we can store and our ability to process that information will far outpace the rate at which we can transmit information over large distances. As a result, where it once took the prowess of a national laboratory to systematically monitor all of the information sent over its external Internet connection, now this capability is available to all.

Today some organizations are following Los Alamos's precedent and routinely recording some or all of the traffic on their external Internet connections. Little of this information is actually analyzed. Instead, it is collected in expectation that it might be useful at some future point. After all, if you want to be able to review the information moving over your Internet connection at some point in the future, you must record it now -- fast as they are, today's processors still can't travel back through time.

Capturing everything moving over the network is simple in theory, but relatively complex in practice. I call this the "catch it as you can" approach. It's embodied in the open source programs tcpdump and windump, as well as in several commercial systems like NIKSUN's NetVCR and NetIntercept, which my company, Sandstorm Enterprises, recently brought to market.

Another approach to monitoring is to examine all of the traffic that moves over the network, but only record information deemed worthy of further analysis. The primary advantage of this approach is that computers can monitor far more information than they can archive -- memory is faster than disk. So instead of being forced to monitor the relatively small amount of network traffic at the boundary between the internal network and the external network, you can actively monitor a busy LAN or backbone.

A second advantage of this approach is privacy -- captured traffic almost invariably contains highly confidential, personal, and otherwise sensitive information: if this data is never written to a computer's disk, the chances of it being inappropriately disclosed are greatly reduced.

In some circumstances, it may not even be legal to record information unless there is a compelling reason or court order. Call this the "stop, look, and listen" approach. This approach, pioneered by Marcus Ranum in the early 1990s, is now the basis of Ranum's Network Flight Recorder (NFR) as well as Raytheon's SilentRunner, the open source snort intrusion detection system, NetWitness by Forensics Explorers, and even the FBI's "Carnivore" Internet wiretapping system (since renamed DCS 1000).

Recently, *Information Security* magazine coined the term Network Forensic Analysis Tool (NFAT) to describe this entire product category. (Ranum coined the term "Network Forensics" back in 1997.)

With the heightened interest in computer security these days, many organizations have started to purchase monitoring appliances or have set up their own monitoring systems, using either commercial or open source software. If you are charged with setting up such a project, or if you are just curious about the technical, ethical, and legal challenges these systems can cause, read on.

Build a Monitoring Workstation

In many ways, a system that you would use for monitoring a computer network looks a lot like any other high-end Windows or UNIX workstation. Most run on a standard Intel-based PC and capture packets with an Ethernet interface running in promiscuous mode.

"Catch it as you can" systems immediately write the packets to a disk file, buffering in memory as necessary, and perform analysis in batches. As a result, these systems need exceptionally large disks -- ideally RAID systems. "Stop, look and listen" systems analyze the packets in memory, perform rudimentary data analysis and reduction, and write selected results to disk or to a database over the network. Of course, no matter which capture methodology is employed, the disks eventually fill up, so all of these systems have rules for erasing old data to make room for new data.

How much attention you need to give the hardware you use for network monitoring depends to a large extent on the complexity of your network, the amount of data at the points you wish to monitor, and how good a job you want to do. If you are trying to capture packets as they travel over a 384kbps DSL link, a 66Mhz 486 computer will do just fine. If you are trying to make extended recordings of every packet that goes over a fully-loaded gigabit link, you will find it quite a challenge to build a suitable capture platform and disk farm.

To explore the differences between different operating systems and hardware platforms, Sandstorm Enterprises purchased two identically-configured Pentium III-based dual-processor systems with removable disk drives. One system was set up as a packet generator using a program that transmitted individually serialized Ethernet packets of varying sizes. The second system was set up with rudimentary capture software -- either tcpdump on the UNIX systems, or windump for Windows.

We then wrote an analysis package that examined the recorded dump files and calculated both the percentage of dropped packets and the longest run of dropped packets under varying network load. By holding the processor, bus, and Ethernet cards constant and loading different operating systems onto different hard disks, we were able to determine effects of different operating systems on overall capture efficiency. Once we found the best operating system, we were able to swap around Ethernet adapters and disable the second CPU to determine the effects of different hardware configurations.

The results of our testing were more reassuring than surprising. Over the six operating systems tested, FreeBSD had the best capture performance and Windows NT had the worst. Under FreeBSD, we found that Intel's EtherExpress cards had the best packet capture performance. Finally, we found that FreeBSD did a somewhat better job capturing packets when run with a single processor than when run with two processors, although if additional analysis work was being done at the same time on the same computer, having two processors was vastly preferable. The reason for this is that no process can dominate both processors at the same time, and thus one processor ends up doing packet capture, and the other processor ends up doing analysis.

Sandstorm used the results of this testing to choose the hardware configuration for its NetIntercept appliance, although the results are applicable to any organization setting up a monitoring system. Of course, for many installations the choice of hardware and software will largely be determined by available equipment, training, and the supported hardware or software of the monitoring software to be used.

For example, organizations with significant Linux experience will almost certainly prefer using Linux-based systems for their packet capture systems, rather than acquiring experience with FreeBSD. And unless you are on a heavily loaded 100BaseT network, the overall packet capture differences between FreeBSD and Linux are probably irrelevant.

(Note: some vendors have developed specialty hardware for directly tapping T1, OC3, ATM, and other kinds of wide-area network connections. If this sort of surveillance work is what you need to do, such equipment can be tremendously useful. In many applications, however, it's possible to get around the need to tap these physical layers by setting up monitoring or mirror ports on a managed switch.)

If you intend to record most or all of the traffic moving over your network, you need to spend as much time thinking about your disk subsystem as your processor and Ethernet card. Last year Sandstorm spent several months comparing IDE drives with the UDMA100 interface to SCSI LVD-160 drives. We also explored a variety of RAID systems. The conclusion: today's IDE drives are significantly faster than SCSI drives costing two or three times more per gigabyte stored.

This is not the result we were expecting, and it goes directly against the conventional wisdom that says SCSI is inherently better than IDE. Nevertheless, it does seem to be the ugly truth, at least for straightforward read/write tests in a single-user environment. Although we saw the highest performance with a hardware-based RAID 5 system manufactured by Advanced Computer & Network Corporation, we saw nearly the same performance with a RAID 5 system based on the 3Ware Escalade 7000 RAID controller.

Long-term storage of captured data is another problem entirely. Although you can build a terabyte RAID system for less than $2,000, backing this system up will set you back $4,000 for the AIT II tape drive and $120 for each 100GB cartridge. Absent extraordinary requirements, most users will elect not to back up their capture disks, and instead archive specific capture runs to CD-R or DVD-RAM drives.

Analyzing the Data

After you've taken measures to collect the information, your next big decision will be the analysis tools that you can bring to the table. If you have built your own system, your primary analysis tools will be tcpdump and the `strings` command. You can use tcpdump to display the individual packets or filter a few packets out of a large data set. The `strings` command, meanwhile, will give you a rough transcript of the information that passed over the network. Snort will allow you to define particular conditions that generate alarms or traps. If you purchase a commercial system, your analysis will be pretty much limited to the capabilities the system provides. That's OK, though, because analysis is really the strength of the commercial offerings.

In a world in which strong encryption was ubiquitous, the monitoring performed by these network forensics systems would be restricted to what's called "traffic analysis" -- every IP packet contains the address of its destination and the address of its sender. By examining the flow of packets over time, it's possible to infer when a person is working, who they are communicating with, what Web sites they are visiting, and other sorts of tantalizingly vague information. Traffic analysis is the stuff that a lot of military intelligence is built upon, and it can be very powerful.

Unfortunately, we do not live in a world in which strong encryption is ubiquitous. Largely as a result of the U.S. government's war on encryption in the 1980s and 1990s, the vast majority of personal, sensitive, and confidential information sent over the Internet today is sent without encryption, open to eavesdropping, analysis, and misuse.

Using a network forensics tool you can spy on people's email, learn passwords, determine Web pages viewed, even spy on the contents of a person's shopping cart at Amazon.com. The tremendous power these systems have over today's networks makes them subject to abuse.

If you install a monitoring system, you should have a policy regarding who has access to use the system, under what circumstances it should be used, and what can be done with the information collected. In fact, you should have such policies even if you do not install an NFAT, since every UNIX workstation is a potential network wiretapping tool.

Indeed, none of these network forensics tools -- not even the FBI's Carnivore -- provide capabilities that are fundamentally new. Back in the 1980s, packet capture programs were available for DOS and UNIX. Using these programs, it was possible to eavesdrop on people's email, learn passwords sent without encryption, and otherwise covertly monitor information sent over networks. This vulnerability to covert monitoring is a fundamental property of most communications systems, including telegraph wires, long-range microwave links, and even semaphore.

But while monitoring was always possible in a networked environment, NFAT tools make monitoring considerably easier than ever before. On a gigabit network it is simply not possible for a human to examine each passing packet to see if it contains useful information. The power of these tools is their ability to rapidly distill down a large data set into manageable chunks.

As such, these systems are a double-edged sword for security and privacy. On the one hand, a powerful NFAT makes it possible to put a spotlight on a particular subject. You can, for example, covertly monitor all of the email messages sent between a pair of users. But on the other hand, these systems also make it possible to conduct surveillance of a network being used by thousands of people and limit the information captured and disclosed to external intrusions, system glitches, or one or two individuals under surveillance. Of course, this selective capability makes it far more likely that these surveillance capabilities will actually be used.

For example, in 1996 the FBI obtained its first Internet search warrant for the Internet backbone at Harvard University. The FBI was investigating a series of computer break-ins all over the world; they were all originating at Harvard from a variety of different machines belonging to the faculty of Arts and Sciences. But rather than record the contents of every TCP/IP connection, which would have subjected Harvard's entire community to unacceptable monitoring, the FBI used a program called I-Watch (developed by the Automated Systems Security Incident Support Team at the Defense Information Systems Agency in Washington, D.C.) that could be programmed to only capture TCP/IP connections that contained a particular keyword.

It turned out that the hacker was breaking into other computers and setting up a program called "sni256." So by only recording TCP/IP connections that contained the letters "sni256," the FBI was able to restrict the data collection to those TCP/IP connections made by the attacker. (As it turns out, during the monitoring period, two other TCP/IP connections belonging to legitimate users contained the same keyword and were inadvertently captured.)

Ultimately, the monitoring capabilities made possible by an NFAT are not a tremendously big deal to anyone who has spent time working as a system administrator, since these are exactly the same sort of capabilities granted to a person with UNIX "root" or Windows System Administrator privileges. Most system administrators regard being able to read people's email and look into their files more as an unwanted responsibility than a right. It is a necessary capability that occasionally needs to be used, but generally administrators have better things to do than to nose around through other people's business. And while there are exceptions, generally people who abuse positions of trust do not retain those positions.

From a legal point of view, your right to monitor (or to be free from monitoring) depends on who you are, where you are working, and who is doing the monitoring. Corporations generally have free rein to monitor their own networks, provided that employees and network users are told in advance that the monitoring may be taking place. (It is not necessary to inform the employees before each specific instance of monitoring, however, so most corporations generally inform their employees with a posted policy and leave it at that.)

ISPs are required under the Electronic Communications Privacy Act (ECPA) to protect the privacy of their customers' electronic communications -- they can't eavesdrop on communications or disclose intercepted contents -- unless one of the parties to the communication has given consent, or if the monitoring is needed to maintain system operations, or in cases of a court-authorized intercept.

Generally speaking, most ISPs require their users to give implicit consent to any and all monitoring as part of their "terms of service" agreement, so for most practical purposes the ECPA doesn't give ISP users any privacy at all. Law enforcement agencies have the right to monitor without the consent or the knowledge of the individuals being monitored, provided they can obtain authorization from a court. However, they have the added restriction of minimization -- they can only capture and record information specified in their warrant.

Today there is gaping disconnect between the level of privacy that most users expect and what is both technically possible and legal. That is, most users expect that their computer use is largely anonymous and untracked. At the same time, computers are getting better at monitoring, more products are being introduced specifically for the purpose of monitoring, and legislation such as the USA PATRIOT Act is making monitoring even easier than it was in the past.

Conclusions

Full-content network monitoring is no longer the province of spooks and spies -- it's increasingly a practice that serves a variety of goals for both computer security and overall network policy. These days the underlying hardware is certainly up to the task, and some of the software that's out there, both commercial and free, is exceedingly good.

What hasn't caught up is our understanding of what to do with this technology -- what it is good for, and what uses should be declared out of bounds. In particular, few of the commercial or free offerings have facilities for watching the watchers -- that is, for logging the ways the systems have been used in an attempt to prevent misuse. Likewise, few organizations have developed policies for the appropriate use of this technology, other than catch-all policies that simply allow the organization to monitor anything for any purpose whatsoever.

Although there has been a lot of public opposition about monitoring technology in general, and about the FBI's Carnivore project in particular, ultimately these systems will be used by organizations because organizations need to understand what information is moving over their network connections. As such, it behooves us as technologists to understand how these systems work, to publicize their capabilities and their limitations, and to develop standards for their ethical use.

Editor's note: Simson L. Garfinkel is Chief Technology Officer of Sandstorm Enterprises, which develops and markets the NetIntercept network monitoring tool. Garfinkel is also the author or coauthor of numerous books on computer security, most recently Web Security, Privacy & Commerce (O'Reilly & Associates, 2001).

Return to the O'Reilly Network.