chapter 12

Hackers don't want you to know that ... *they can see you but you can't see them*

Chapter summary

- How LANs work
  - Ethernet
  - Token Ring
- The inherent risk of shared media
- The pros and cons of network sniffing
- The difficulty in detecting a sniffing attack
- Some techniques for detecting sniffers
  - L0pht's AntiSniff
  - IBM's Sniffer Detector
- Ways to protect against unwanted sniffing
  - virtual private networking (IPSec)
  - extensive LAN segmentation
  - physical security

Remember the old 'party line' telephone systems? By sharing the same line with other people in your neighbourhood, the telephone company was able to reuse

their facilities across multiple customers. This resulted in a more economical service for everyone concerned. The problem was that anyone sharing your line could also listen in to what you were saying. This required that everyone on the line exercise a considerable degree of courtesy in order for private conversations to remain private. Of course, all it took was one nosy neighbour to eavesdrop on another person's call to throw the whole system of trust into a tailspin. As soon as private lines became available at a reasonable price, everyone moved to this service and the party lines went the way of the dinosaur. Or did they?

Hackers don't want you to know that your corporate network may be a lot more like a party line than a private line. They know that LAN technologies are built on the concept of a shared medium. That means that it is, in fact, possible for them to eavesdrop on what are presumed to be private data exchanges in order to glean useful information to further their attacks. They are, in essence, like a nosy neighbour on a party line.

12.1 What's that smell?

Most LANs are based on a protocol known as **Ethernet**, which is, at its heart, a broadcast technology. The idea is that if a client station wants to get data from its server (or another client station, for that matter), it formulates a data packet, tacks on the appropriate communications header, addresses it to the server, and then sends the packet out over the line to be transmitted to the intended destination. The server, and all the other workstations on that LAN segment, sit listening for any packets that might be intended for them. Packets addressed to another station are simply ignored. An inevitable consequence of this arrangement is that every station on a given LAN segment can hear what every other station on that segment is saying. Since no one would normally want data that isn't applicable to them in the first place, the system works quite well. But all it takes is one bad apple to spoil the barrel, as we will soon see.

Token Ring is another popular LAN technology which suffers a similar fate. In a Token Ring network each station is basically responsible for:

- receiving the 'token' (a specially formatted message) and any accompanying 'frames' (the data to be sent) – it might be helpful to think of the token as a bucket carrying its contents (frames) to the intended destination,
- determining to whom the frame is addressed,
- copying the data to the machine's local memory, if the frame is addressed to this station, and
- sending the packet along to the next station on the ring.

(Of course, much more goes on than this, but this is enough detail for our discussion.)

With both Ethernet and Token Ring technologies the normal mode of operation is for stations to only look at the header of a given packet to see if it is addressed to them (Figure 12.1). If it isn't, they don't bother to examine the rest of the packet's contents. There is an exception to this typical case known as *promiscuous mode*. Promiscuous mode, as its name implies, isn't as well behaved. With this mode the LAN adaptor is not restricted to dealing only with its traffic, but it saves a copy of the packet contents for *all* the packets it sees, whether they are addressed to that particular station or not. An adaptor can be switched from 'virtuous mode' (not an official term but it is the opposite of 'promiscuous', right?) to promiscuous mode through software running on the workstation.

Why would LAN adaptor manufacturers create such a security loophole in their products in the first place? Surely, they understand that they have created an environment that is ideal for eavesdroppers, don't they? It turns out that this capability has some legitimate uses in the proper hands. Network technicians often use tools that allow them to monitor all traffic on a network in order to do problem determination and performance analysis. Such tracing tools are often called **sniffers** (after the product of the same name from Network General Corp. which rose to fame in the

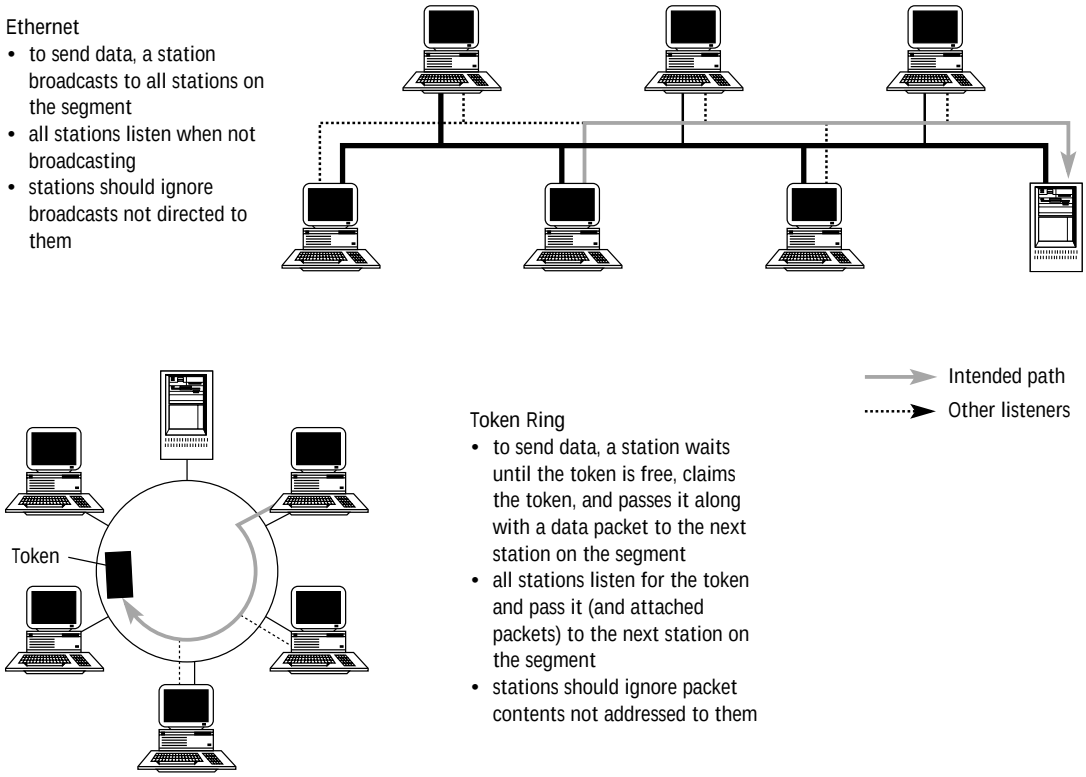

FIGURE 12.1 A comparison of Ethernet and Token Ring LAN operation

early 1990s). A minimal sniffer might consist of a PC (possibly a laptop for the sake of portability) containing a LAN adaptor capable of promiscuous mode and software designed to analyze the traffic it sees. It might classify traffic based on network protocol, workstation, size, volume, etc. In addition it would be able to show the actual byte-by-byte composition of selected packets so that protocol errors could be diagnosed.

Such information in the right hands can be a valuable asset when trying to tune a network or simply keep it up and operational. This same information in the wrong hands could spell disaster.

12.2 Aroma or stench?

What happens if hackers gets their hands on a promiscuous mode LAN adaptor (which isn't at all hard to do) and the requisite software to exploit it? The answer is that they start snooping around your network looking for anything that might interest them. It turns out that the same commercially available sniffer tools intended to help the good guys are a double-edged sword that can be exploited by the bad guys just as well. The difference isn't in the tool but in the *reason* the tool is being used. **Sniffing** can be a legitimate technique used by authorized network technicians or a means for hackers to eavesdrop on corporate communications. Since network traffic could include passwords, confidential reports, or even incriminating email, the stakes can be extremely high.

Unfortunately, the barriers to entry into the world of malicious network sniffing are quite low. Since most Ethernet adaptors support promiscuous mode (many Token Ring adaptors do, but they are somewhat less common), all hackers need is a LAN adaptor and some readily available sniffing software and they're in business at minimal (if any) cost.

One of the more effective sniffing tools for swiping passwords is L0phtCrack's built-in **Server Message Block** (**SMB**) packet capture facility. This tool takes a look at all the packets going across the LAN, saves a copy of any server logon packets it sees, and discards the rest. This way, hackers end up with a nice, neat listing of userids' encrypted passwords which they can then feed into the tool's cracking facility (see Section 11.5 for more information). In other words, if hackers are interested in just stealing some passwords, L0phtCrack makes the job easy by filtering out all the other extraneous transmissions and saving only the desired packets. When you consider that the vast majority of LAN packets don't contain logon information, you quickly realize that this weeding out process can save hackers a considerable amount of work.

A compelling argument can be made for the legitimate use of L0phtCrack's password-cracking facilities as they can be used by system administrators to test the strength of the passwords their users have chosen. However, the case for sniffing

tools such as the SMB packet filter is on shakier ground. Of course, there's the standard 'if we didn't do it someone else would' argument, but the fact remains that while sniffing for the purpose of diagnosing network problems is valid, sniffing in order to swipe passwords or other corporate communications is highly unethical and most probably illegal. The reason for including such information in this book, of course, is not to tell hackers how to steal passwords, because they already know how to do this and are probably very familiar with tools such as L0phtCrack. The objective is to alert IT professionals who, for the most part, don't know that such dangers exist, in order that they may take appropriate actions.

12.3 The 'silent attack'

What can you do to defend against a sniffing attack? You might think that you are safe since hackers would have to gain physical access to your LAN in order to do any real damage this way. It turns out that this is only partially true. In fact, even if it were completely true, it still wouldn't be reason enough to justify a sense of security. Here are just a few of the reasons why ...

Many people say 'it doesn't matter if someone hacks this particular system because it doesn't contain anything of value anyway'. Hackers don't want you to know that if they can gain access to *any* client or server workstation on your network, they can install a sniffer which could run unnoticed in the background. How could attackers get access? Wouldn't they have to actually put their hands on the keyboard of the target workstation in order to plant the sniffing tool?

Not necessarily. In fact, the entire attack could be executed from halfway around the world. Hackers could simply email a **Remote Access Trojan** (**RAT**) to their victim and let it do the rest of the work for them. The RAT might appear to be an innocuous program that produces an entertaining screen show (e.g. a holiday greeting) when, in fact, its real intent is to install a sniffer on the target system. Once installed, the RAT might send back a message to the hacker, via either email or an Internet Relay Chat (IRC) session, telling him or her where to attack. In addition to broadcasting the location of the compromised machine, the RAT might also steal a telnet password on that machine and pass it along to the attacker. At this point all the hacker would need to do is logon to the compromised system from time to time to see if anything interesting had been snared by the sniffer.

This illustrates why security must be dealt with on a corporate, rather than individual, level when considering issues such as password security mechanisms, firewalls, and virus/Trojan horse protection. Even if you do a good job of repelling outsiders, you need to remember that insiders are already in an excellent position to sniff your network. They already have physical access to your LAN and probably have a clearer idea of what information is valuable and what is not.

Complicating all of this is the fact that sniffing truly is a 'silent attack'. As you will soon see, the passive nature of this sort of eavesdropping makes it very hard, if not impossible, to detect. Since most Ethernet adaptors support promiscuous mode, any station could be copying data packets and go completely unnoticed. Initially, Token Ring adaptors were somewhat better in this regard since very few had this capability. This fact offers little comfort, though, as many of the newer adaptors are capable of promiscuous processing.

Some adaptors are conscientious enough to automatically broadcast the fact that they are operating in promiscuous mode. A LAN management tool could detect these messages and warn you of a potential intruder. However, you can't depend on this capability to save you since it's ultimately left up to the adaptor (or, more likely, the software controlling it) to generate these notifications. So while legitimate sniffers used by authorized network technicians might alert you to their presence, the kinds used by hackers won't – and it is the latter that you need to be the most concerned with.

12.4 Sniffing for sniffers

Since you can't rely on the sniffers to broadcast their presence and turn themselves in, you have to find another way to locate them. There are no foolproof methods guaranteed to sniff out the bad guys, but there are some techniques that can be used to *predict* whether a sniffer is about. Many of these are based upon the idea of trying to detect some of the tell-tale signs that sniffers unintentionally leave in their wake. In other words, you can't directly test for promiscuous mode on other machines but you can borrow a page from the hacker's book and do a little snooping on the snoopers.

12.4.1 Antisniff

One such tool that can help is AntiSniff[1] from L0pht Heavy Industries (Figure 12.2). AntiSniff combines a number of different detection techniques into a single program, which can proactively test systems that are suspected of running in promiscuous mode. AntiSniff looks for specific operating system quirks that can indicate promiscuous mode processing. For example, some versions of Linux (a popular UNIX variant) can be tricked into exposing themselves when confronted with a specially modified PING[2] request. The trick is to build a PING request using the Internet Protocol (IP) address of the suspected machine but address it to a LAN

1. AntiSniff, L0pht Heavy Industries, www.l0pht.com/antisniff.
2. PING (Packet INternet Groper) is a command which generates an ICMP echo request. This packet is sent to a specified address in anticipation of a reply (which indicates that the station is reachable by the network).

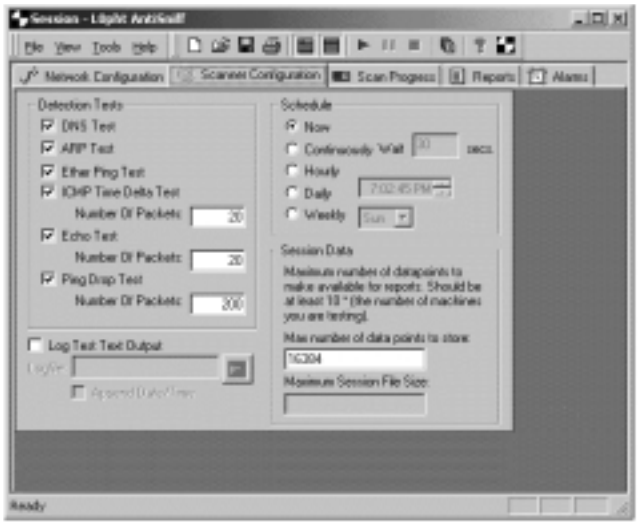

FIGURE 12.2 Sniffing for sniffers with AntiSniff (Copyright © 1994, 1995, 1996, 1997, 1998 LHI Technologies)

adaptor that doesn't actually exist. Adaptors not operating in promiscuous mode will ignore the request because it isn't addressed to them. But since a promiscuous mode adaptor copies all packets (not just those addressed directly to it), the eavesdropping station passes the request on to its own IP software which, in turn, takes the bait and responds to the PING. AntiSniff simply listens out for the PING response and, if it gets one, it concludes that a sniffer has been unearthed. This test can be circumvented, though, by hackers if they update their system with the appropriate patch or choose an operating system that doesn't make this same mistake.

Another test that AntiSniff can perform relies on the fact that most systems use the DNS to look up the numeric IP address (e.g. 10.1.2.3) for a given symbolic name (www.widgets-r-us.com). Reverse DNS lookups, which do just the opposite, are far less common. Sniffers, however, frequently make use of this function because their users often want to know the *name* (rather than just the IP address) of the workstation whose packet they just copied. AntiSniff exploits this fact by manufacturing bogus network traffic that appears to come from a station that, in fact, doesn't exist. AntiSniff then switches its adaptor to promiscuous mode and waits to see if anyone attempts to resolve the fictitious address with a reverse DNS lookup. When the hacker's sniffing tool steps out from the shadows in an attempt to figure out the name of the non-existent station, AntiSniff zeroes in on the sniffer's location.

AntiSniff can also make educated guesses as to which stations are in promiscuous mode by observing the latency or delay that occurs during a burst of bogus network traffic. The theory is that stations that aren't in promiscuous mode aren't likely to be adversely affected by 'background noise' which they quickly realize is

not intended for them. However, a promiscuous mode station will be slower to respond when the network is flooded because it will be busy trying to process all the additional traffic (which is bound for the non-existent station). AntiSniff, then, takes advantage of this behaviour by PINGing stations during normal network operation to establish a baseline response time. Then it generates a flood of meaningless traffic and tests the machines again with another set of PINGs. The second set of results are then compared to the baseline and if a significant delay is found only on certain stations, they are considered to be sniffing suspects, since they were bogged down trying to process traffic they should have ignored. Of course, this technique can be error prone and, therefore, any conclusions drawn can only be considered preliminary. Nonetheless information gathered during this test can be used as the basis for a follow-up investigation.

12.4.2 Sniffer detector

A completely different approach to promiscuous mode detection involves a bit of misdirection which, when pursued by the hacker, is a dead give-away to malicious sniffing. In this case a series of client machines randomly logon to a designated server, issue a few commands, and then logoff. Since the server performs no meaningful work and, therefore, exists only for the purpose of acting as a hacker *decoy*, it is assumed that any attempt to logon to it by anyone other than the clients assigned to participate in the charade must be the result of sniffing (Figure 12.3). The server's job is simply to record all logon attempts that originate from an IP address other than those of the decoy clients and alert a system administrator of the suspicious activity.

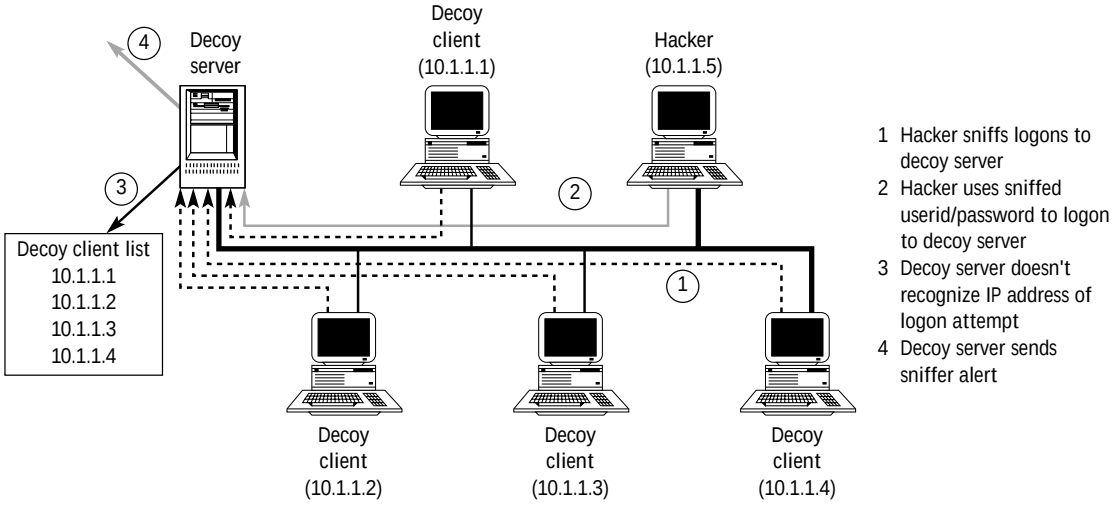

FIGURE 12.3 Bait and snitch: the operation of a sniffer detector

This indirect approach can be particularly effective because it can be easily automated and run in the background. By using particularly intriguing machine names like 'topsecret' or 'payroll' and powerful-sounding userids like 'root' and 'sysadmin', the decoy server can appear especially inviting to hackers.

A more sophisticated variation on this theme came from IBM's Global Security Analysis Lab (GSAL) and was presented at RAID '99, an industry conference dedicated to intrusion detection (Grundschober, 1999). GSAL's 'Sniffer Detector' also includes a separate Manager system which oversees and coordinates the entire process by instructing decoy client systems when and where to logon. In addition, a 'Probe' system is used to sniff out any logon attempts to the decoy servers.

As you can see, most of the techniques for detecting sniffers involve setting a trap of one sort or another and then sitting back and waiting for hackers to take the bait. Clever hackers could, of course, 'fly below radar' if they were careful to avoid leaving traces of their presence. So while there may be no conclusive method for sniffing out the sniffers, there are any number of useful ways to narrow down the search.[3]

12.5 Hanging up on the party line

So while sniffing is hard to detect in many cases, all is not lost. There are some things that you can do that will restore at least some degree of confidence into your corporate communications. Probably the most effective of these options involves encrypting transmissions as they traverse the network. As mentioned in Section 1.2, TCP/IP was not originally designed for secure communications. As a protocol for use in military networks it was assumed that all data would flow over private lines and that the endpoints would be protected by soldiers with high-powered weapons. In such an environment, the need for secure network protocols is diminished.

However, as TCP/IP has made its way into popular use in both public networks (like the Internet) and private corporate networks alike, the need for greater security has become obvious. To address this requirement the *Internet Engineering Task Force (IETF)*, the group that controls the TCP/IP standard, launched an effort to design a next-generation version of the protocol which came to be known as IPv6.[4] Among many other improvements, an increased level of security has been added. During this process, though, it was realized that the need for security was simply too great to wait for widespread IPv6 deployment (which would take many years), so they decided to retrofit these enhancements into the existing IPv4 standard.

The name given to these security improvements was IPSec, which stands for **Internet Protocol Security**. IPSec defines a framework for setting up and operating

3. For more information on sniffing see Robert Graham's 'Sniffing FAQ' at www.robertgraham.com/pubs/sniffing-faq.html.
4. IPv4 is the predominant form at the time of this writing.

a **Virtual Private Network** (**VPN**) where transmissions can be authenticated and encrypted. Authentication features allow you to know if the message really came from the person you think it came from, while encryption helps ensure no one else can read your messages. As such, encryption gives us the 'P' (for 'private') in VPN. The 'V' (for 'virtual') comes from the fact that while the communications are, in fact, private, the network is not. In other words, if a message is scrambled in such a way that only the intended receiver can read it, then it's safe to send it across a public backbone because eavesdroppers won't be able to make sense of what they've just received. A brief tutorial of IPSec is presented in Appendix B.

Of course, it's more complicated than that as there are numerous issues regarding cryptographic strength, protection of private keys, and so forth, that ultimately determine the security of such a system (see Chapter 19 for a brief treatment of crypto issues).

The point is that VPN technologies *are* available today and can be used to keep your private communications private. However, setting up the necessary infrastructure can be time consuming and expensive. Its appeal, though, lies in the fact that a VPN over a public network can be substantially cheaper than private, leased line connections. In fact, Infonetics Research estimates that VPNs can save anywhere from 20–80 per cent when used for external network connectivity (Infonetics Research, 1997). As a result, this approach is probably best deployed sparingly within the corporate intranet for particularly sensitive applications but used extensively when communicating to remote offices, business partners, and mobile employees who dial in for remote access.

Often overlooked in the deployment of VPNs for these workers, is the fact that more and more classified data will be making its way out of the corporate campus and onto the far less controlled environment of laptops and home PCs. Some have observed that using strong crypto to encrypt credit card numbers over the Internet is like using an armoured truck to deliver a paper sack full of money to a park bench.[5] While this statement is an extreme illustration of e-commerce insecurity, its basic point is just as applicable to VPNs. It doesn't do much good to use strong crypto to obscure transmissions over the network if they will be stored 'in the clear' (i.e. unencrypted) on an unprotected platform at the other end of the VPN tunnel. The increased use of high-speed, 'always connected' services (often in conjunction with VPNs) for remote workers only compounds the problem, as you will see in Section 23.3.

This means that while VPNs represent an effective defence against sniffing, they aren't the total answer in and of themselves. Sensitive data should be encrypted when it is stored on disk and frequent backups should be taken in order to avert disasters.

5. Based on an example by Gene Spafford in Garfinkel and Spafford (1997).

12.6 Moving to a private line

Another option to foil sniffing attacks on the LAN involves isolating workstations in order to minimize the shared nature of the communications media. Today's switching hubs (networking hardware which provides improved manageability for LANs) could hardly be imagined by the originators of Ethernet because the price per port was prohibitively high. Now, however, in the interest of better performance, many companies are moving toward greater and greater degrees of segmentation on their LANs. By essentially putting each workstation on its own private segment, you can dramatically improve throughput (Figure 12.4). Think of it this way: if you have to share a connection with 10 or 20 other people, you know there will be some contention for the available bandwidth (i.e. capacity). On the other hand, if you don't have to share with anyone, you can effectively have full use of the connection and its capacity.

Owing to decreasing costs, switching hubs have turned this dream into a reality. The great news, from a security standpoint, is that if you're the only one on your segment, hackers can't camp out on an adjacent workstation and sniff your packets. This isolation not only improves performance but it also improves security.

Still, even this offers no guarantee. If hackers plant their sniffers in the server, they still will be able to listen in on a great deal of what is being transmitted because many transmissions will be directed there anyway. There also exists the possibility that packets will leak out from their intended segment and end up on other segments. This really shouldn't happen if the switch is doing its job, but people who have tested these devices continue to report cases where it does, indeed, occur. Therefore, anyone intending to rely on such a mechanism for security would be well advised to test their LAN for this sort of leakage from time to time.

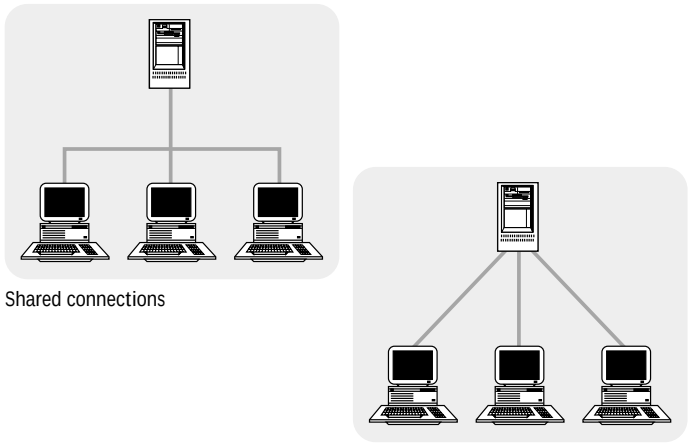

FIGURE 12.4 Before and after extensive LAN segmentation

12.7 Choices, choices, choices ...

Sniffing attacks are among the most difficult to defend against but all is not lost. Here's a quick review of your options:

- *Encrypt data using VPN technology*. This is the most secure option, assuming strong cryptography is used, but it isn't cheap. Also, the overhead of encryption could degrade performance to some extent and setup/administration is far from trivial.
- *Extensive LAN segmentation*. By moving to switched hubs and putting every workstation on its own segment you can isolate LAN traffic so that it isn't broadcast to everyone within earshot. This could be expensive if you have to replace existing equipment but, once implemented, wouldn't have the maintenance problems of the VPN option. However, once packets leave the segment and hit the server or the WAN (wide area network), hackers may be able to position themselves in a sniffing position anyway.
- *Sniff out the sniffers*. If Token Ring LANs are involved it may be possible to detect some promiscuous mode adaptors via network management tools. Such an event could then trigger an alert to notify the proper personnel. However, since there are many ways for hackers to get around this, don't put too much stock in this as a line of defence. Better still are some of the sniffer detection methods mentioned in Section 12.4 which involve baiting a trap and waiting for hackers to break their silence.
- *Maintain tight physical security of the LAN*. Of course, you should do this anyway, but the potential for sniffing is just one more reason why. The deficiency with this approach is that it's very difficult to do and, even if you succeed, hackers could still plant a sniffer on a legitimate LAN workstation. This could even be done by a hacker outside the corporate intranet via a RAT (Remote Access Trojan). Also, physical security may not help much in the case of insider attacks since restrictions often must be eased to accommodate office and equipment moves and mobile employees.