RECORD, Volume 29, No. 1*

Washington, D.C. Spring Meeting
May 29–30, 2003

Session 37OF
Electronic Security in the Internet Age

Track: Computer Science

Moderator: BRIAN M. SEPTON
Panelists: MITCHELL SCHWARTZWALD†
BRIAN M. SEPTON

Summary: This session is intended to improve your understanding of the history, current status and future of Internet security, and increase your awareness of common vulnerabilities. Topics discussed include firewalls, encryption software and how Internet networks are constructed.

MR. BRIAN M. SEPTON: We've got some material to go through, but we certainly want to hear from you, so here's what we're going to be doing. We'll start off with some introductions. Mitchell will walk through the distinct processes of an Internet transaction. I will talk about encryption. We'll then shift over to the corporate network. I'm then going to talk about a crime security survey by the Computer Security Institute (CSI). We'll look at some examples, and then we'll have some time at the end to fill with questions and answers. Let me start off with introductions.

Our guest presenter is Mitchell Schwartzwald. Mitchell is not a member of the Society. He joins us from Chicago, where he works for a company called Plumtree Software. They do Web portals for financial services companies and others. He has been at Plumtree for about two years. Prior to that he spent some time in the dot-coms, where he was working on system integration between back and front ends, Web browsers and back to inventory. Prior to that he developed and managed some systems at Whirlpool. Mitchell's company is based out of San Francisco. I'm Brian Septon. I'm an FSA (Fellow of the Society of Actuaries). I work mainly as a pension consultant, but my practice and expertise include benefits, outsourcing systems and pension plan valuation. I am also on the Spring 2003 Meeting Committee, and I am a member of the Computer Science Section Council Board. With that I want to turn it over to Mitchell. You have some questions for the audience.

---

†Mr. Mitchell Schwartzwald, not a member of the sponsoring organizations, is a consultant at Plumtree Software, San Francisco, Calif.

MR. MITCHELL SCHWARTZWALD: What we're trying to do is find out a little more information about people's roles in their organization. If you're not comfortable answering, that's fine also, but that way we can try to tailor the presentation to fit your questions and needs. I put together four questions: your role in your organization, your areas of responsibility, rough idea of the size of your organization and what initiatives you have going on in your organization. I'll start.

I'm a consultant with Plumtree Software. My area of responsibility is helping customers succeed in deploying our product. We have about 350 employees, and currently we're rolling out a new version of our product. Brian, would you like to go?

MR. SEPTON: I'm a principal of Chicago Consulting Actuaries. My main responsibilities are information technology (IT) and actuarial work. We have about 100 people, and we are looking at putting in more Internet security—trying to prevent hackers—and we're also looking to expand to a wide-area network (WAN) between our various offices.

MR. SCHWARTZWALD: Anyone else?

MS. INGE HARRISON: I'm the second employee out of six in a small company. We're reinsurance intermediaries, but one thing that we do is we get large volumes of variated data from our clients every month, and a lot of that comes over the Internet. I'm very interested in seeing what I can do to protect the confidentiality of that.

MR. SCHWARTZWALD: Before I actually go into the presentation, I want to give everyone some background about why I picked the areas to talk about today. If you were to walk into your bookstore and you started to look at information on electronic security, you'd find that there's a lot of language that's foreign to a lot of people, a lot of terms and acronyms that are very difficult to understand, and they start talking about the various pieces. Until you see how they all sort of work together, they're very hard to follow. From an end user's perspective, like a home user going to a Web site and trying to purchase something, there's only so much you can do about security. The second part of security is what's happening behind the scenes. If a hacker broke into a large e-commerce site (Borders, Amazon—it doesn't matter) and took everyone's credit card number, the hacker can get a lot further than just trying to steal your one credit card. That's the second part about the corporate network—just understanding what the corporate parts of that are, because everyone works at a company of some size. That way if you go and pick up a book, hopefully you'll have a little more education and understand the terminology it's talking about.

To start, there are the three distinct processes of an Internet transaction. The first is the sending segment. In other words, you're sending some information to a site. There's the transmission of your message, there's the receiving segment and then

there's the storage. In other words, if you send your credit card number over the Internet, what's going to happen once it's stored? I think we should talk about that a little bit, so I'm going to just continue on.

I tried to draw a diagram, which you see in Chart 1. As we go on, we'll look at different areas of this diagram in more detail. Let's start with the basics. We've got a user. Let's just say the user wants to go and pay his or her life insurance bill online. So what happens? The user is going to send a transaction because the user has a connection to the Internet over what's called an ISP, or an Internet Service Provider. All these little tower-looking things are supposed to represent different hubs on the Internet. An idea that I want to represent here is *multiple things.* One, it's not like the user's computer is connecting directly to your computer. The second part is that there are many different switches, so to speak, and routes—you can see two different routes—that a transaction can take between where it's sent and where it's received. Eventually it's received by the ISP at the company, and it goes across to the company's network. We're going to talk about this in a little more detail later, but the basic idea is the company has got a Web server somewhere and probably has a database that contains information like whether the person paid his or her bill, how much the bill was and the person's credit card number. Are there any questions?

FROM THE FLOOR: Can you talk about why there are so many little hubs before you get to the place you want to go to?

MR. SCHWARTZWALD: Let's suppose the user was in New York, and the company where they were processing the transactions was in the state of Washington. There may not be a direct connection between the two cities. Sprint and WorldCom, between the two of them, account for over 50 percent of the Internet traffic in the United States. In other words, their networks are the ones that are transmitting it. We'll actually see that between a city like New York and Seattle, the most direct route probably has two Internet connections, and it could be up to seven or eight or nine. We'll talk about some of the security benefits of that.

FROM THE FLOOR: The more security problems, the more hubs you have?

MR. SCHWARTZWALD: More hubs do not pose a security problem. Actually, if I were a hacker, and let's say I was somewhere in between here, and I knew you were going to go and pay your life insurance policy, and I thought I could get your numbers somehow, well, considering there are different routes, that's already made it more difficult to physically get at a piece of data, so it actually can be a security benefit.

FROM THE FLOOR: Then the multiple products are also part redundancy. Should one go down, you have a multiple route?

MR. SCHWARTZWALD: Yes, that's a very good point, thank you. As we go on, hopefully more of this will become clearer. We're going to talk about the corporate side in a lot more detail.

You go on to the Internet, and you go to send your transaction. You have control only over so much. One of the important things to know about is that in a browser, there are always some Internet security options and privacy options. I want to go over what a few of these things are. Under the browser and Internet security and privacy options—it doesn't matter if you're on Netscape or Internet Explorer because they all have these types of options—there are a few different things that can happen. One is, that the browser can be SSL enabled, which means Secure Socket Layer, and to make it really simple, that means your data have been encrypted. Brian is going to talk a lot about encryption a little later on. How we will recognize that something is SSL encrypted would be at the beginning of the address it will say "https" instead of "http." Not all encryption is the same. We'll talk about that in the company portion.

Along with that you can have cookies. People talk about cookies as a bad thing. Cookies don't necessarily have a user ID and password in them, and if they do, the password can be either encrypted or not encrypted. The basic idea is that they're storing some information on your computer. When we talk about the password-protected computer, if you just stored some information locally on your computer and anyone can walk up to your computer, anyone can get to that information, so you have to enter a password when you get onto your computer.

There are two other interesting things to talk about when you're looking at Internet security and browser security. One is whether it enables Java script or not. Java script is often used; it's a very good thing. It can be used, for example, so if you select a country like the United States, then under the city location or under the state location, it can get the states down to the appropriate states/region without actually refreshing your browser, which really improves performance from an end user's perspective because no one is willing to sit there for ten minutes while the states load because it loads every state/region in the world.

Continuing on, Active X controls are very powerful. They can also be very dangerous because an Active X control can talk to the operating system. If a hacker is sending you an Active X control and you installed it, the hacker could possibly send a command over the Internet and delete all the files off your hard drive or something of that nature. I'm not here to scare anyone today.

MR. SEPTON: Mitch, what are some good Active X controls?

MR. SCHWARTZWALD: Macromedia sometimes uses Active X controls in some of its versions, so that would be like some Flash and streaming media. Microsoft uses a lot of Active X controls. In fact, another good example is Citrix used to use an Active X control to be able to sort of share their whole desktop. I don't know if

anyone is familiar with Citrix. Basically Citrix is a remote client. Someone dials into it, and Citrix can launch maybe an application that's custom to your company, and by dialing in you're securing it. Active X controls are used a lot less frequently. They're very difficult to deploy.

I'm going to continue on then to the transmission and repeating segment. This comes down to the idea, when we were looking on the network diagram, that you're going to have a connection to your ISP. It may be analog; it may be wireless. Wireless is easier to break into than having to physically gain access to cable. Another thing is that a lot of companies will have more than one ISP. Through the transmission layer, there are things to think about. Are you sending things in clear text or SSL? Obviously if you're sending credit card information, you want it to be in SSL. If I'm sending an e-mail to Brian with my presentation, I don't really care if it's encrypted or not.

Then on the receiving side, there are things to think about. Who has access to the data that you've sent? If you're dealing with a small vendor, has the vendor subcontracted someone to provide the service to it, and is that person reliable? I think as we go through the company portion and the corporate network portion, you'll be able to make a better judgment whether someone is a reliable vendor or not. That's all that I have on the receiving segment.

There are things to think about with the storage. Who has access to the information? Is it encrypted? In other words, if they're taking their credit card information, are they encrypting it so that someone in their company can't just look at the database and see everyone's credit card numbers? How well is it protected? So it's risk mitigation. How many sites did you use the same user ID and password for, and how often do you change it? Is it a difficult password? If you use your birth date or something like that, someone who knows you can easily guess it.

Continuing on with that, what kind of information are you sending? Again, if I'm sending a presentation and someone gets it, I may not care. If I'm sending my credit card or my Social Security number, I should be a lot more cautious. Here's an example. If all credit cards from Bank X begin with 5555, if they're a MasterCard, the following numbers are 1111. If you enter a card that is a MasterCard from Bank X, and a hacker got in and got 1,000 credit card fields, that also said what bank they were from and whether it was a MasterCard or a Visa, all of a sudden the hacker is going to be able to use all that information to try to crack whatever encryption formula was used. You'll notice most Web sites will only ask you if it's a MasterCard or Visa and not what bank it's from, just so they don't have all that information.

I'm going to turn the presentation over to Brian at this point, and he's going to talk to us about encryption.

MR. SEPTON: All right, thank you. I want to talk a little about the history of encryption, give a little bit of detail on the question of the unbreakable cipher, or the unbreakable code, and talk about some of the contributions made by Diffie, Hellman and Merkle—that's Whitfield Diffie, Martin Hellman and Ralph Merkle. Then I'll get into some of the mathematics behind what makes RSA and PGP such good encryption tools, and then I'll just touch on how those are used in the actual Internet browser through SSL and S-HTTP.

Encryption is thousands of years old. It's actually as old as the secret. There are two different ways of hiding messages. The first is steganography, where you actually conceal the existence of a message, and the second is cryptography, where you conceal the meaning of a message.

In terms of branches of cryptography, there are both codes and ciphers. A code is a substitution at the word level. For example, if you replaced "attack" with "retreat," and "today" is replaced with "tomorrow," instead of "attack today" you'd get "retreat tomorrow," which is clearly a confusing message if someone were to intercept it. Ciphers are substitutions at the level of letters. There are transpositions where you change the order of letters in a given word For example, "FSA" can become "SFA." Or you can have substitution where you just substitute letters for one another. The Society of Actuaries can become the Casualty Actuary Society.

When I was thinking transposition, I'm not sure that's such a good encryption method for ciphers, because I was trying to think how to transpose the word "mom" or the word level where they're palindromes. I guess you get a lot of things that actually still say the same thing.

In terms of history, the ancient Greeks were big users of steganography, and the records of this date back to 500 B.C. Their idea of encryption was to hide the message. There are some stories of them writing messages on the inside of barrels, writing messages and covering them with wraps, or actually writing messages on a person's head, letting the hair grow for a couple of months and then sending the messenger over. It's still based on the idea that if somebody were to intercept the message, the person could read it, but just the pure existence of the message was hidden.

There are lots of good examples of encryption from history. Mary Queen of Scots, who ruled Scotland in the 1500s, was sentenced to death by her cousin, Queen Elizabeth of England, because of an assassination plot which was discovered by breaking Mary's encryption or breaking her cipher, which was actually the key to her arraignment or the key to her execution. But throughout history there has been an overall desire for ultimate secrecy. What we've found, and what the textbooks have found, is that codes are unbreakable, but not forever. When a code is first brought out, there's an idea that maybe this is the unbreakable code, but I think history has shown that everything is breakable to a certain extent.

There's actually a French cipher, the Great Cipher used by Louis XIII and Louis XIV, which was halfway between a cipher and a code. It had substitutions at the level of syllables in French, and it also had some tricks in it such as delete syllable phrases and stuff like that, which made it very complex to understand and complex to use in those days. It actually took cryptologists or analysts over 350 years to crack that code after it fell out of use in the 1600s.

Up until now we've been talking only about steganography, which is, except for the French example, that you have to know where a message is in order to be able to crack it. When Mary Queen of Scots was corresponding with people from her prison cell or when Louis XIII and Louis XIV were corresponding with their agents, you didn't know when they were passing pieces of paper to and from each other even if they had code or even if they didn't.

Marconi's invention of the radio in 1894 changed everything. What we now have is more frequent communication in the public domain, which means that it's not a question of where you're going to find the message, it's a question of what are you going to do with all the messages you intercept. So the need for the unbreakable cipher was redeveloped now that the communication was more easily facilitated between parties. In World War I, the French had some fantastic cryptographers who actually cracked some complex German codes and according to some sources, cracking the German ADFGVX cipher was one of the keys to turning the war around. The French cracked the cipher, found the Germans 50 miles from Paris, pinpointed their troops and were able to execute a raid on them. It changed the course of the war in driving the Germans back at Montdidier and Compiègne.

Then there was also the great Enigma of World War II, which was the German encryption machine that did triple and five times substitution ciphers at various levels. That code was actually cracked in two ways. One, a decryption device was found in a sunken German submarine, and, two, a group of mathematicians in England were able to crack the Enigma. When they looked at the transmissions they were able to figure out that the weather report was always in the first line of the transmission, so once they had a key to identify off of, they were able to read back through the encryption. What this shows is that these were actually two-way functions, that the same Enigma used to encrypt is the same Enigma used to decrypt.

We're getting into the point where we're at the Diffie-Hellman-Merkle contributions; that's Whitfield Diffie, Martin Hellman and Ralph Merkle. These men were around in the 1970s. They were here around the time of ARPNet and ARPA, the Advanced Research Projects Agency, which was the Pentagon's first initiative to create a decentralized computer system. The idea in the late 1960s and early 1970s was to create a decentralized computer system to make the Pentagon's infrastructure more indestructible in the case of a nuclear war, but the Pentagon's initiative gave birth to ARPNet, and that in turn gave birth to the Internet in 1982. By the end of

the 1980s, as we all know, nonacademics and nongovernment people were given access to this.

Diffie and Hellman, back in 1974, imagined Internet commerce, and they imagined e-mail. They had the idea that two people would dial up on their computers via phone lines and talk with each other and conduct transactions with each other. They asked, "Should there be some secure way of exchanging information between two strangers meeting on the Internet?" They were first credited with the idea of the public key and public key encryption, but they kind of stumbled on stuff. They said, "If I encrypt one message and I give you the encryption key, then I have to send you the encryption key, which means I have to encrypt the encryption key, which produces another key and so on and so on." They had first thought there might be no way to actually have two strangers meet on the Internet in a secure way. Then they thought of an example about lockboxes. They said, "If I write a message out, put it in a box and padlock it, send it to someone else, he would put his own padlock on it, he would send it back to me, I would remove my padlock, I'd send it back to him and he would take off his padlock." What we have there is an example where two people are able to exchange a message without ever exchanging information about what each other's keys are. That was a big breakthrough for them, that actually there might be a way that two people can meet and exchange information.

What they eventually got to was the idea of a one-way function. They realized that modular arithmetic or base arithmetic can actually very well be a one-way function. You don't really know what base you use. You don't really know what the underlying arithmetic is. All you're going to see is, say, your number is 4, and you're going to be able to use it as a basis for the one-way functions. I think it's kind of cool, so let me step through it for a few moments. This is the basis of the Diffie-Hellman-Merkle key exchange, which we'll see in a moment is the basis for the RSA algorithm, which is the basis of the PGP privacy, which is the basis of SSL. We picked secret numbers. Mitchell picks a 3, I pick a 4. We have a public function out there, which is Y^x mod P. What we do is we plug our secret numbers in there to get the results of our one-way function. In my case I have 7^4 base 16 or mod 16 equals 1, and in Mitchell's case his equals 8. What we then do is we exchange our numbers between 8 and 1, we stick them through a reverse function beta to the alpha mod Y and we both end up having the same results. We've communicated alpha, beta, Y and P in the public domain, but we've been able to keep a part of our component (A and B) private. This idea of a one-way function plays heavily into the RSA algorithms, which we're going to jump to right now.

Who are these RSA guys? RSA are Ronald Rivest, Adi Shamir and Leonard Adelman. They were an MIT team with substantial computer science and mathematical expertise. In 1977, I believe, they came up with the idea of a mostly asymmetric, or basically a one-way, function that kind of solves some of these public and private key distribution issues that Diffie, Hellman and Merkle had thought of or had published earlier. What they ended up doing was perfecting the Diffie-Hellman-

Merkle equations. They came across an algorithm, or they discovered the algorithm, where you can have a private and a public key in order to exchange secure information. They came up with an N, which is the product of two very large primary numbers, and those two primary numbers are P and Q. P and Q are your private key, N is your public key and this leads you to your seven steps that actually encrypt, which is your private and your public key.

FROM THE FLOOR: P and Q are prime?

MR. SEPTON: Yes, P and Q are prime, and in my example P and Q are 17 and 11. Those aren't big enough for this to really work, but when you get to 15- and 20-digit prime numbers, this begins to work. The whole idea is that once you have two very large prime numbers, P and Q, you multiply them together to get an N. It's very hard to factor that N back into prime numbers because here are two prime numbers that make it up, so you might be dealing with a 40-digit number as your N, which then you have to back into the factorization of it.

I choose two prime numbers, 17 and 11; I multiple them together to get my N in step 2, which is 187. I choose another number (E), which is the other component of my public key, and I publish them in my phone directory. Let's say Mitchell wants to send me letter X; we're going to go see *X-Men.* What we then do is convert the letter X into the ASCII equivalent, in this case 1011000.

FROM THE FLOOR: Is that base 2?

MR. SEPTON: That's base 2. Actually, ASCII means American Standard Code for Information Interchange.

MR. SCHWARTZWALD: It's one of the basic text formulas or text character sets that's used. It's been replaced a lot lately because it doesn't support all international characters.

MR. SEPTON: Now instead of ASCII, I'm sure it's some Microsoft-copyrighted name—Unicode. You take that ASCII number, and you take that base 2 number, and you convert that to a base 10 number. That's 1 x 2 to the something, plus 1 x 2 to something else, etc. You end up getting X as 88. Mitchell then determines the C based on his ASCII equivalent number and plugs it into the function. He gets $C = M^E$ base N, or N and E are my public directory numbers. What he has is his message of 88 to my E power, which is 7, base my N power, 187. Bingo: C = 11. This is what he tells me. Mitchell sends me C = 11, and then we go forward with that. What we then do is go back and courtesy of Euclid's algorithm, I have to determine the components of my private key, which are based on my two very large prime numbers, P and Q. My private key ends up to be D = 23.

Then, on to step 7, I'm able to translate. By using my D = 23, I'm able to reverse the function. Remember, my D = 23 was computed only based on my P and my Q

and my E, so I had to know my P and my Q in order to calculate my private key D. I can reverse my function. Bingo: I'm able to get at the bottom of this, M = 88, and I'm able to convert that back to ASCII X, and we're going to go see *X-Men.* That's kind of the basis of the function.

FROM THE FLOOR: What is Euclid's algorithm?

MR. SEPTON: It's an algorithm to compute the greatest common divisor of two positive integers.

FROM THE FLOOR: Is that a formula? It's not something you write out?

MR. SEPTON: No, it's not something you write out. The mathematics behind this RSA algorithm was complex. Back in the late 1970s and the early 1980s, the computing power to encrypt long messages didn't really exist, or didn't easily exist to encrypt a long message. You saw that I was doing 88^7 for some single-character message. If you get to a couple of hundred words, it could take a computer a few minutes to go ahead and encrypt that using this complex RSA algorithm, and you need to be able to support mathematics involving very large numbers. So for two large prime numbers multiplied by each other, you need to have a processor that supports that.

What Phil Zimmerman came up with in the early 1980s was something called "pretty good privacy," or PGP, which you've probably heard about. His idea was that encryption should not just protect the mathematicians, and it should not just protect the computer science people who understand how to use it, but really it should protect everyone. His thought was that there could be dissidents, there could be rebels or there could be other people out there who need the benefit of private communications. He decided that encryption should protect everyone. He was a political character as much as a genius in the mathematical world. He said, "If I take a look at the time and processor requirements for the RSA, that could be pretty tough in the regular message." He thought of a simple encryption method to encrypt the message and then encrypt the key with a very complex algorithm. Back to what Diffie, Hellman and Merkle said, that if I have the message and I have the key, I have to encrypt the key, and I have to then encrypt that key and so on and so forth. Zimmerman stops after the second piece. You encrypt the message using something simple, and then you encrypt the key using something complex. That complex key is based on the RSA algorithm. In his software that he released, he actually had a way of computing these large prime numbers just through a random generator, which was generated through random mouse movement.

PGP does a couple of things. It can protect the contents. If I encrypt with your public key, you can decrypt with your private key, and it protects the contents of the message because only you can get at it. However, if I reverse this function, if I encrypt with my private key, you can decrypt with my public key, and you can determine that only I could have created that message, and so I'm also able to

verify ownership. If you apply this both ways, if you apply this at the same time, you can both verify the ownership and protect the contents, which then would ultimately be what Phil Zimmerman wanted. He created a software package that required no knowledge of encryption to stick this in. That's the PGP software package, and those are some of the algorithms behind it.

What we now have is, just briefly, the secure sockets layer. This is all based on the idea of a public key and a private key from the RSA algorithms, but that's SSL, something that was developed by Netscape and supported by Navigator and Internet Explorer and maybe other browsers as well. It uses the public key to encrypt the data using some of the same algorithms from RSA, and it develops a secure connection between the client and the server for encryption, authentication and message integrity. As Mitchell mentioned earlier, "https" indicates SSL is in use. In SSL jargon you commonly hear people referring to things as "bits of encryption." The bits of encryption are the lengths of the session key for each transaction, and the larger the key, the harder it is to crack. Forty bits was common a couple of years ago, but now computers can actually crack a 40-bit key in less than the lifetime of the universe. Now we're up to 128-bit key; actually I was told a couple of days ago that some software has a 168-bit key. A 128-bit key is trillions of times stronger than the 40-bit key. I believe this has to do with the powers of 2—so 2^{128} versus 2^{40} is the number of times more powerful this key is. Somebody wrote once that all the computers on earth would have to take longer than the age of the earth to crack 128-bit encryption, so 168-bit encryption must be pretty good then.

Then there's S-HTTP, which is not in use as much, but it transmits individual messages securely and complements SSL. Really, SSL and HTTPS are where it's all at.

As the last thing, we've heard about export restrictions. You cannot encrypt with a foreign power. You cannot encrypt to Cuba, Iran, Sudan and a few others. The Clinton administration, back in 1996 I believe, toned down the export restrictions on encryption, but you cannot use or export encryption software to some other countries. If you want to learn some more, there's a great book out there called *Codebook* by Simon Singh. With that, I'm turning this over to Mitchell.

MR. SCHWARTZWALD: What we're going to talk about next is being able to know about and understand how the corporate network works and the different parts of it so you can make better decisions. I've done this a little backward. Usually I like to talk about the business drivers, but since this session is on security and not the business drivers, they're sort of on the bottom, and we're not going to cover them in a whole lot of detail other than what they are and how they interact.

I wanted to start off with interoffice connections to the Internet. What is an ISP? What are firewalls? What's a DMZ? What's a proxy server? What's a VPN? If you start reading security books, you're going to hear about some of these, and we're

going to cover each of them in a little more detail as we go on. Also, we'll talk about what authentication services and directory services are, and we'll talk about redundancy to a certain extent. I've taken the diagram that we saw earlier and put in a little more detail in Chart 2.

The basic idea is that out here to the left of the firewall is the public Internet, so that's where your external users would be. If you look inside this big box, that would be a firewall around a specific company for, say, the North American office, a specific location. Then you see little lines going up to something labeled EIMA, APEC, SA South America, and those would be connections coming back into the firewall, but those would be secured connections in some way—either a private connection that the company owns, or it could be over VPN. We'll talk about what VPN is in a few minutes.

Then there's the second firewall, and this one says internal firewall. The whole idea about that is that for anything that you expose to the public in general, the more layers of protection you have the better. So if a hacker broke in and took control of this Web server, he or she would still have another firewall to get through before getting into your internal network, where you hopefully have your important data stored. Then what usually happens is that this internal firewall allows communication only between these specific servers and named locations, and that's just sort of a firewall thing. You could connect from my Web server to my database only, and you give it even a specific database.

FROM THE FLOOR: Is that how the firewall works, that only specific people can come through it?

MR. SCHWARTZWALD: That's one of the ways, and it can work in a whole lot of different ways.

Then in an internal network, the idea is that you have a router or switches. The terms, especially in a nontechnical circle, I have thrown around interchangeably. The general idea is that you're going to have some device that all your computer network cables plug into, and it's going to be able to tell one computer how to get to the next computer.

We have internal users also. In some companies there would even be more firewalls. They might put multiple different rooms; it's called a DMZ. You might have your e-mail servers in one area and another firewall in between those, and Web servers for external users to pay their life insurance claim online, or whatever you're trying to do. The idea is that inside the internal firewall you're going to have other types of services, such as Web services or maybe only internal functions like a company Intranet. You're going to have some sort of applications, like accounting systems, financial systems and e-mail servers. There are going to be some database servers.

A database is a large selection of files that's been organized in such a way that information can be put in and accessed very quickly. If you look at your desktop or laptop that you might have at your office, it typically has a 20-gigabyte hard drive. Often databases get into the terabytes, so after a gigabyte, a terabyte would be 1,000 gigabytes. There are some databases that companies have that are 10 and 20 terabytes. Considering you can fit an entire encyclopedia on one CD or less than a CD, so it's only a couple of hundred megabytes, that gives you an idea that it's just very large amounts of data. The database is organized in such a way that the data can be secured, but you can send it a command to return a certain amount of data or a certain piece of data, like someone's bill. It's supposed to be able to return it in seconds. You'd have tables, and you could also do some functions in a database like summing information, so you could do a sum on all the bills that were paid in April or all the invoices that were sent in April, and that gets into more of the applications that usually drive databases.

Another concept is the directory service. It also refers to authentication services, and we'll talk about what they do. Companies often have some mainframes and file servers. The file server would be somewhere other than your local computer where you might save, say, your Word documents, so if you're working on a project, it's often an area that's backed up nightly so that if your computer crashes, someone can recover your data.

When we talk about the connections to the Internet, we talk about an ISP, like AOL, accessed via a cable modem. Often through the utility providers there are T1 lines, DSL, all sorts of different lines. There's the firewall, which is going to control the access between the ISP and into your company, and that's what we looked at a little bit. Then when we look at that, some of the ways to control traffic and access to the network would be through a port filter—we'll talk about what a port is—packet scanning and reverse proxy servers.

Packet scanning is a device that's looking at every message that goes across a network for certain key words. Maybe you're in an industry or you're in the federal government, and you're concerned about people and messages that are going back and forth about job searches. So you're going to read them, and you're going to be able to see who the top people with job search information are. If the top messages about job searches are coming from the Human Resources (HR) department, especially the hiring branch of it, you'd consider that normal. It can also scan for certain types of files. I don't know if, when you've opened a file up in Excel, you've seen something that's called like a macro, and it's said, "Caution, this could be a macro." The idea is that a macro can execute a program on your computer, so it looks for specific types of files. You can look for those macros, and you can even stop them from coming through your firewall to prevent risks of viruses; a lot of viruses have been sent that way.

FROM THE FLOOR: My company now filters out messages from certain addresses. Is this connected with packet scanning?

MR. SCHWARTZWALD: It's often a combination of packet scanning plus some specific software that they bought that looks for specific domains. Say I sent something to you from Plumtree. Plumtree is a software company, it's registered and they'll find it in a directory. But if it was coming from a place they never heard of, you won't get it.

FROM THE FLOOR: What can you do to get messages that are being blocked?

MR. SCHWARTZWALD: There are things that you can do. There's software that you can buy and put on your computer. You might even have to add someone into an address book before you can get an e-mail from them. It depends on how much your technical people want you to do.

MR. SEPTON: My wife has Hotmail, and the other day she made it so she can only receive messages from people in her address book. Her fear is that once in a while she's going to exclude something from someone she wants to get a message from, but most of the time she's going to exclude messages such as "Time To Get Fit" or "Have You Tried Viagra"—whatever it might be.

FROM THE FLOOR: Is that from service providers?

MR. SEPTON: Yes. Different Internet software or different mail programs would have different levels of available options.

MR. SCHWARTZWALD: Right. Hotmail and a lot of the companies just don't do a lot of filtering. In fact, they're trying to sell it as a service, so they think if they don't do the filtering and they sell it as a service, they can make more money.

FROM THE FLOOR: You can put screens on Internet Explorer from Microsoft.

MR. SEPTON: With some of the newer versions of Internet Explorer, you can even tell it to block stuff from a site that's considered to have adult content, and they have other ones that target more specific sites, like advertising sites.

FROM THE FLOOR: Some are specified.

FROM THE FLOOR: Just to clarify, whenever unsolicited messages have those lines about "Hit 'Reply' to get off the mailing list," don't do that, because a lot of spam messages blindly target e-mail addresses. They don't know if they're really online, and as soon as they get that reply, they say, "Oh, yes, there really is a live body at that e-mail address."

MR. SCHWARTZWALD: There has been talk in Congress about trying to put a charge like 25 cents on e-mails to get people to stop doing that. But that was a couple of years ago, and it never really materialized.

A reverse proxy server goes along with the firewall. That would be a rule that basically says that only a user, and we verify who that user is via some method(there are different methods), can get inside my company and only to this server. That would be an example of how a reverse proxy works. The whole idea is that you're controlling who can get in. Even if you have a firewall and you say people can't get in, people have broken through firewalls before. With firewalls or virus-scanning software, anything of that sort, the most important thing is to stay up-to-date on all the different updates, releases and patches.

MR. SEPTON: You've got to stay updated on the patches. There are some well-established cases in the Crime Report, which I'm going to go over later. There is a case of a company that used credit cards and purchased credit card software from a third-party vendor. A widely publicized patch came out for this credit card software. Everybody put in the patch, but this one company didn't because it didn't find out about it since it had bought the software through a third party-vendor. This company got hacked repeatedly over a period of six months until it realized that it was missing the patch for the credit card software. So staying up on the updates is important. In Microsoft, they're called Service Packs, and other providers are calling them something else. But if you have the opportunity to get on their mailing list for that stuff, that's probably something that's pretty useful. You want to hear about the updates as soon as they come out because frequently the updates are motivated by hacking activity.

MR. SCHWARTZWALD: Then there's something called a proxy server. A proxy server is used inside a company. In some companies you may find out that you can't go to Hotmail or you can't go to certain sites. A company is limiting your access to certain places.

Another thing that we're going to talk about is called VPN, or virtual private networking. Back a few years ago, everyone used to dial up to companies. When you dialed up to a company, that was considered a secure connection. That was slow, so people wanted to connect from their cable modems. The idea with the VPN is that everything is encrypted. It establishes a route and knows who you are. Because of that and because you have to have software installed on your computer and usually a password entered so someone can't just come up with the software, it's considered a lot safer. People allow access to their networks often over this method or even to satellite offices. It uses a tunneling methodology, which basically means it's going to set up a path, and every route is going to go over that path—over SSL. Again, not all VPNs are equal. A good example of that is some use only 40-bit encryptions; some use 128. If it's 128, everything is going to be slower, but it's still a lot faster than dial-up., You have to decide how much risk you're willing to take.

Take the example where I'm in Washington, D.C., and I want to send a message to Seattle. There isn't a direct route to Seattle on the Sprint network, even though they have as big of a company as Microsoft there. The closest route, if Sprint is the

company's Internet service provider, would be through Chicago and then Chicago to Seattle. You might go through Denver and eventually from Denver up to Seattle through Tacoma. There are a lot of different routes, and that provides a lot of security and a lot of redundancy, from the perspective that you don't know what route something is taking. If something happens in Chicago and the city loses power for four days, it doesn't mean the entire country and the Internet traffic is going to come to a stop. Things get rerouted. Even from the United States to South Africa there are a couple of different paths to take on the MCI network.

Next I'll talk about networking and how things work. To start off with, there are things called protocols, which are basically a set of standards that allows the transmission of data independently of a platform or hardware. In other words, it shouldn't matter if you're on Windows, if you're on Unix or if you're on a Macintosh computer. It doesn't always hold true because people call things protocols and they're not necessarily independent, but the most common protocol and most used protocol is TCP/IP, or transmission control protocol/Internet protocol. The TCP establishes the connection, the IP actually transmits the message and this is what powers the Internet. You're going to hear this come into play in other areas of companies, too, because people are looking at TCP/IP telephones. The idea is if something is digital, you don't need to have as many wires running all over the place, and you only need one network to support, for example.

NetBT, which used to be called NetBIOS, is a Microsoft proprietary protocol. I say it's proprietary because basically no one else has adopted it. It's really powerful and it's great for certain things, but, for example, if I got on a telephone line right here, I could probably reboot one of the computers at our corporate headquarters. Because I can dial in, the hacker can get to that protocol. It scares most network administrators and most security people, so most people don't even allow it into their DMZ.

IPX/SPX is similar to TCP/IP. It's even more efficient than TCP/IP, but it was developed by Novell, and again, not many people have adopted it. It's sort of fading away because you have to use it across the Internet, and if you want to talk to another company, that's what you have to be operating on.

Let's talk about packets. Going back to sixth-grade science when they talked about atoms being the building block of matter, packets are basically the building block of what goes on inside a network transmission. Everything is literally broken down into a packet that usually contains the source, who the sender is, what the destination is, what its number is in the transmission. If you're sending a large Word document, it's going to be many packets. When it gets to the other end, it has to be put back together.

Then the last thing would be a port, which is basically a connection medium. The way I like to think about it is that it's like your cable TV. You might get 200 channels, and you know that if you go to channel 5, it's going to be NBC or

whoever it is. In the same way there are specific ports that are named for specific types of transmissions, and it's a way for the computer to keep everything straight. You have to establish you know what port to connect to.

The last is layers. A layer would be like SSL. I'm referring to this layer because SSL can run over TCP/IP, it can run over IPX/SPX and it can run over NetBIOS. Something big out in the industry is HTTP/SOAP (Simple Object Access Protocol). The basic idea is that you can send a text message to someone, and that text message is going to trigger that someone to run some sort of procedure and do something and return something to you often. We've talked about Active X controls before and how people were scared to do processing on their computer—you didn't know who owned it. What's good about SOAP is you're just sending a message, and it's getting very standardized in such a way that someone can receive the message back and get some sort of processed information.

Tunneling is the same as VPN. We talked about that a little bit and how SSL can go over HTTP or SOAP, which again would be your HTTPS and your Internet browser.

Let's talk about routing. Each transaction is going to consist of multiple packets. There are many Internet routes to most locations. Large companies often have more than one ISP. There was a time when Sprint at one point during the day and MCI at another point during the day almost completely lost Internet capabilities because their networks were attacked by hackers. What companies do is have multiple Internet providers so that if one goes down, they're not out of business. If you're Amazon and no one can get to you, you might as well not be open for the day.

How does it all work, and how does it all go together? Hopefully the example will make this all come together. The basic idea is that you're sending packets across a network, they have to be routed in a certain path and they're going to get to their destination. Since there are multiple packets and they may not all go across the same path, they may not get there in the same order, in which case someone is going to have to put them back together. I think those were the challenges in the beginning too.

Here is the exercise I came up with. Today's exercise will show how e-commerce transactions are broken into packets, sent across the Internet and then put back together. I've got 20 cards, and I'm going to hand half of them to two different people. Then Brian is going to stand in the back of the room and collect them, and please pass each card to at least one other person before Brian. We're going to find out that the message comes across in a totally jumbled-up way, but the idea is to get people moving and to make the exercise a little more practical.

MR. SEPTON: I'll start reading with the one I got first: "Will show exercise today's together how transactions packets broken sent across e-commerce back put the

Internet then are and into." So, just like you're saying, each packet went through a different place, like the hubs.

MR. SCHWARTZWALD: This demonstrates a few things. For starters, you can see how confusing it is to pass it. This is the challenge the computer has to go through every time you try to send a message. So there are all sorts of protocols and things behind it. What would happen if Brian got only 19 of the 20 cards? People realized this a long time ago. When you're going across this many different people, it's possible that a connection got dropped and something got lost along the way. We'll talk a little bit later about redundancy and error checking and things of that nature.

Authentication is a big topic. Authentication, which we've talked about and I apologize if people didn't understand what it was at the time, is the process of validating that a user, computer or other devices is who he, she or it claims to be. You don't want to allow someone to just get in and log in and see a life insurance bill if you don't know whom the person is. Going down that route, there are things that have been developed, and we have them on the diagram. They are called directory services. Directory services work together with an authentication service, and they're going to store, for example, a user ID, a password, someone's first name, last name, and a company often would have your manager, what department you're in, maybe where your office is located and a telephone number. They're similar to databases in that they're optimized for searching, and in addition to that, they perform encryption functions, so they're going to protect all the passwords. You may have heard of Active Directory or NT, and then there's LDAP, which does the lightweight data. Directory Access Protocol is standard. Active Directory technically falls into it, but Microsoft decided that they were going to be different—take the standard and modify it so it doesn't exactly fit all that. Novell has one. There's Sun's iPlanet; IBM has an LDAP directory. A lot of people have them. The concept is that you want to be able to authenticate who somebody is.

An example of risk with an authentication or directory service is the University of Texas, which, I believe, in December or January had someone break into its directory service, which contained, going back five or six years, everyone's first name, last name, Social Security number, along with information about where they lived when they were on campus. That's basically enough information to get credit cards. All sorts of notifications went out, and the authorities tried to handle it, and from what I heard, the information wasn't used inappropriately. However, it shows you some of the risks involved with that, and obviously there are a lot of benefits also. When we talk about authentication, besides a directory service, you can have something that's called a user name pair, so for instance a user ID and a password. Another thing would be a token, so, for example, in a cookie, you could store a token. The first time you logged in somewhere, they would pass you maybe a128-bit character, a token, that isn't going to make sense to anyone else. It's probably alphanumeric and has all sorts of funny characters you can't even read, and that just gets passed back and forth.

The basic idea of NTLM and Kerbose is that if you log on to your network in the morning and it is a Microsoft network, a Web server can say that it believes who you are because it knows you're logged into the network based on a token that was passed in your browser. I don't want to get too much into the details because I think there are other topics that are more interesting to people.

We talked about a cookie earlier. The basic idea is that you can put a user ID and a password in there. Hopefully, it's an encrypted password, because otherwise if someone gets into your computer and the person sees your password in clear text, the same one you use to manage your bank account, obviously that person could do a lot of damage.

The thing to think about with data storage is, especially if you're managing a project or something of that nature, if you're taking credit cards or private information like a Social Security number, are you storing it in clear text? Hopefully, at least you're converting it to ones and zeros, so if someone looks at it the private information won't be in plain text. But it's best encrypted.

Then again with data manipulation, there are compression technologies. In other words, if you look at HTML and if you ever click on your browser, if you right click on a page, you go to "View Source," and you're going to see there's a whole lot of text. You may not have that much showing on your screen. HTML tends to be bulky—in other words, there are a lot of spaces for formatting and tabs. Do people use WinZip ever—a zipable file? It's the same idea. An Internet browser knows how to unzip the compression so it can compress it and get it a lot smaller. If anyone is trying to do something internationally with a Web site, it's a great way to make things a little less painful, and it's cost effective also.

In hardware encryption, there's something like an SSL accelerator. In other words, you want to encrypt something, but you don't want necessarily to take the burden on your server, so you're pushing it off to a different device to do the work. By "server," I mean the computer that's hosting the site.

The idea of caching is, again, that's putting files on your computer, but I don't care if someone finds the Yahoo logo on my computer, and every time I go to Yahoo I'd rather not take the time to download images because they're big. The idea of caching is easy access to common information that you're going to all the time when you refresh your page. They can store it on your computer the first time, and your computer knows not to request it again from the server. It's going to speed up your communications.

Let's talk about redundancy: We talked about having multiple ISPs so that if one goes down, a company doesn't lose connectivity. Sometimes companies use public networks, and they don't own a network between the main office in Washington, D.C., and a satellite they have maybe in Arlington, Virginia. Sometimes they have private networks. If they have a private network and that also has an ISP, if your

public network goes down, traffic could be routed across your private network into the other ISP. People use all sorts of combinations of that to have redundant connections to the Internet. The Internet is inherently redundant, so that's a benefit.

In terms of fault tolerance, it's like I was saying before. If I'm transmitting a credit card and the receiver got only 14 of the 15 or 16 numbers, that's going to create an issue very quickly. The most basic of the error-checking technologies was called parity checking. What it often did was basically convert everything to zeros and ones. You sum them up; it's even or it's odd. That's not very good because if you dropped a character on a credit card, it may still add up to even or odd or whatever it's supposed to add up to. People got into much more complex mathematical formulas and other ways to check, even sending something and knowing how many bits of data you were supposed to receive and seeing you receive that many bits. I don't think that's what we need to go into today, but it is an issue that people have thought about.

Then, like I said, at the end, what are the business drivers? If you didn't have any e-mail, you couldn't buy something online. If you weren't writing Word documents or maintaining accounting systems, you wouldn't have a need for a network. Application server is a common terminology. That usually means doing processing of some sort, so an e-mail server could even be an application server. But often they talk about an application server as doing the processing of your data for your accounting system for, say, an ERP system, which basically does inventory and planning. Also, you have a database server. We talked a little bit about what a database is in terms of its joint information and being able to search for that information. So, there's an e-mail server, a file server where you might store your files and they're backed up, and then a Web server. Generally it's recommended that only a Web server would be exposed in that area that we call a DMZ. DMZ stands for "demilitarized zone." It refers to the North Korea and South Korea conflict. The idea is that it's very isolated, and you have very limited access to what you can do. I'm going to let Brian talk about the CSI Crime Report.

MR. SEPTON: As an the overview of the CSI Crime Report, we're going to look at who replied to this, what some of the common breaches were, the protection, talk about worms and viruses, and then have a quick case study from the insurance industry. The Crime Report is published by the Computer Security Crime Institute. They focus security on three questions: Who are you? Can I verify this? Do you have clearance? But if you think about it, answering those questions properly doesn't tell me what I'm going to do once I answer them or once I get into the network. So this is just looking at what people are actually doing once they answer those questions.

The Crime Report is an annual survey and has been done since 1997. This year they got 481 respondents in their survey. by Looking at industry sector, financial, high tech and manufacturing make up a big portion of the respondents. They

probably also make up a big portion of the American industry, so that's no surprise. By number of employees, a quarter of the companies had 10,000 or more employees, although 16 percent of them had fewer than 100 employees, so this does span the gamut in terms of employer size. In terms of employer by gross income, we have some billion-dollar companies in here as well as some smaller, under $10-million companies. From what I gather about the people in this room, I think that this survey adequately represents the types of companies we are working for. I know I can picture my company falling into these slices as well as what we heard about Mitchell's and some of the others. I think this Crime Report survey is about us.

Of the 481 people who replied to the survey, 90 percent reported security breaches. Of the 223 that had losses, they reported $455 million in losses. Proprietary information—your code, your marketing plans, your client list, your pricing book, whatever it might be—losses were put at $171 million. It's hard to quantify what the value of that is. The way I like to think about it is, if somebody stole my rate book and I sued them in court over it, I could get tons of money over that. So you might value proprietary information as what the market value might be and some sort of court settlement. Some have reported losses to law enforcement. We've got some invasions from outside; we've got denial-of-service attacks. A denial-of-service attack is where you've got a flood of irrelevant and useless requests to your Web server, and you can't distinguish the difference between a real request and a useless request. It's as if I had 100 people dial your phone number and one real person dialing your phone number, so you wouldn't be able to distinguish the real from the fake, and your phone circuits would be overloaded. Of course, 85 percent had computer viruses.

In terms of whether there has there been unauthorized activity in the last 12 months, it looks like there was a peak back in 2000, when 70 percent had unauthorized activity at the peak of the Internet boom. We're still up there—60 percent of the people had unauthorized activity beyond a breach.

It's no surprise to us that the Internet is where most of these attacks are coming from. In terms of once the attacks are in, who these attacks are actually coming from: foreign governments, foreign corporations, hackers, competitors or employees—these are nonmutually exclusive groups. What we see is that all these groups are fairly well represented, whether it's foreign powers, competitors, employees or hackers. We're getting attacked by everyone.

Then in terms of a summary of some of the financial impacts of for example, financial fraud, $116 million; virus damage, $50 million; laptop theft, $12 million. I can think of three people who have had their laptops stolen in the past year, so that $12 million sounds a little low, but I'm not sure how they might have computed that one.

In terms of protection, what gives the government power to prosecute and get hackers? It's the Economic Espionage Act of 1996 passed by the Clinton administration, which gives the government authority to crack down on crime. It does two things. The first is that it gives the firepower and the forum to get people, and it also alerts the industry, validating that potential threat. The passage of the bill says to industry, "Hey, the government is going to take these crimes seriously, and we're going to go after them."

In terms of security technologies used, what are companies doing? Companies control access via passwords, physical security, antivirus software—almost everyone is doing antivirus software—and encryption. Only 60 percent of the respondents are using encryption; I'm not sure why that's not higher because it's really simple to use. That was the surprise of the report. Of course, worms and viruses are a problem. Worms and viruses are simple pieces of code that can be terribly devastating to a system, a server, a user, a computer, hardware or software. The losses are increasing. The average loss as computed by the crime survey is increasing to $283,000 a pop in 2002. Remember, this is just among the respondents. In terms of globally, some global estimates say that the "I Love You" virus from a couple of years back had an $8.7 billion worldwide impact, and that the "Melissa" virus is a $1 billion impact. I think both of those were e-mail viruses, where they got into your in-box, executed codes, duplicated themselves and took down systems and networks.

I wanted to read a case study from the CSI relating to Prudential Insurance. This is straight from the CSI/FBI survey. "In March 2002, U.S. federal agents working with the New York Electronic Crimes Task Force arrested Donald Matthew McNeese on charges of identity theft, credit card fraud and money laundering after he stole a computer database containing personal records for as many as 60,000 employees of the Prudential Insurance Company and attempted to sell the data over the Internet. McNeese had worked as the administrator of the database at Prudential's Jacksonville, Florida, office until June 2000. The federal complaint states that McNeese not only offered to sell Prudential employees' identities over the Internet, but he was also engaged in other activity related to credit card fraud." There are some more details in the report.

Are there any questions on the Crime Report?

FROM THE FLOOR: I was wondering about the different kinds of firewalls, how they are set up and how you get to them.

MR. SCHWARTZWALD: We can talk a little more about that. The basic idea of the firewall is you want to have limited access points. There are all sorts of different vendors of firewalls, but they all do or try to do similar things. There are two different ways to install a firewall. One would be what they refer to as a software firewall, where you would actually put it on a computer, and the other would be a hardware firewall, where it's actually a physical device. More or less it's a computer

that's designed and sold by the vendor that's basically tuned to certain types of activities.

The idea is that you control access to only certain points and only certain types of traffic. When we talked about packet scanning a little bit, we talked about ports; usually there would only be a couple of ports open into a firewall. For example, port 80 is the normal one for HTTP, and port 43 is the most standard one for HTTPS. Usually one of those two ports would be open through the firewall, and often the firewall would do some sort of packet scanning. It may be very simple or a little more complex. Then after whatever type of packet scanning it's doing, combined with a firewall often there is a reverse proxy of sorts built into the firewall. In other words, the firewall has the ability to say that it's only going to allow traffic over a specific protocol, say HTTP port 80; it's like normal Internet browsing to a specific machine or set of machines. Sometimes you get more than one e-mail server. That's very common. Then sometimes you have a second firewall. Again you're going to say, Between the perimeter and internal firewall, so then the DMZ, I'm going to allow traffic only from this machine to maybe two or three different machines. It may be over a larger set of ports because often a Web service has to communicate to maybe your database and an application server and a directory server.

Completely opposite of everything we've talked about is that a firewall is usually used in conjunction with any type of proxy server, not just a firewall alone to block access out. In other words, people use instant messaging like AOL Instant Messenger or MSN Instant Messenger. They go over a specific port again, so often those ports are blocked or access to that application is blocked because a company knows with, say, AOL Instant Messenger, you can share your desktop. If you are allowing me to see my desktop inside of your company, and I've got control to use your mouse on your desktop, that means I can also open up files on your desktop and possibly get to places that I'm not supposed to go. Some people close traffic like that; some people don't.

Another thing that was very popular a couple of years ago was the different vendors that were allowing people to download music, like Napster. People blocked Napster because they didn't want employees sitting at their desks all day long consuming all of their bandwidths to their Internet service provider downloading MP3s and their music files. That's the completely opposite side of the firewall.

MR. SEPTON: Those firewall settings weren't on your standard setting. At the corporate IT administration level, if they brought you Yahoo Messenger, no matter how hard you try or how many times you load it up, it's going to be blocked. It would have to be a central IT decision to open that port up. Mitch, you were saying the other day that there are how many ports on a firewall?

MR. SCHWARTZWALD: General TCP ports go from, I think, one to 32,000-something. I don't have an exact number, but there are many of them, and they

usually block everything by default and then open up when requests come in. So if you have a request that you want to grant access to some Internet site out there so everyone can learn about their company, they'd open up just that one port. We can talk about this at home a little bit, too. For example, I don't know if people here have high-speed Internet, like cable modems or DSL. With a simple router from, say, Linksys, that you sometimes use to share your connection to a laptop that you bring home from work, by default that doesn't allow anyone to make a request coming in. It allows a response, so I sent a request. The response can come back, but it doesn't allow anything to come in—that's the default setting. Can it be broken into it? Yes, but it's a very nice basic step, especially to keep a 12-year-old out.

MR. SCHWARTZWALD: Also, the simple Linksys type of devices that we were talking about are a great step for anyone who has, say, a satellite office and a small company and doesn't have a big budget, because you can usually get them for under $100. It's a lot better than having nothing.

Chart 1

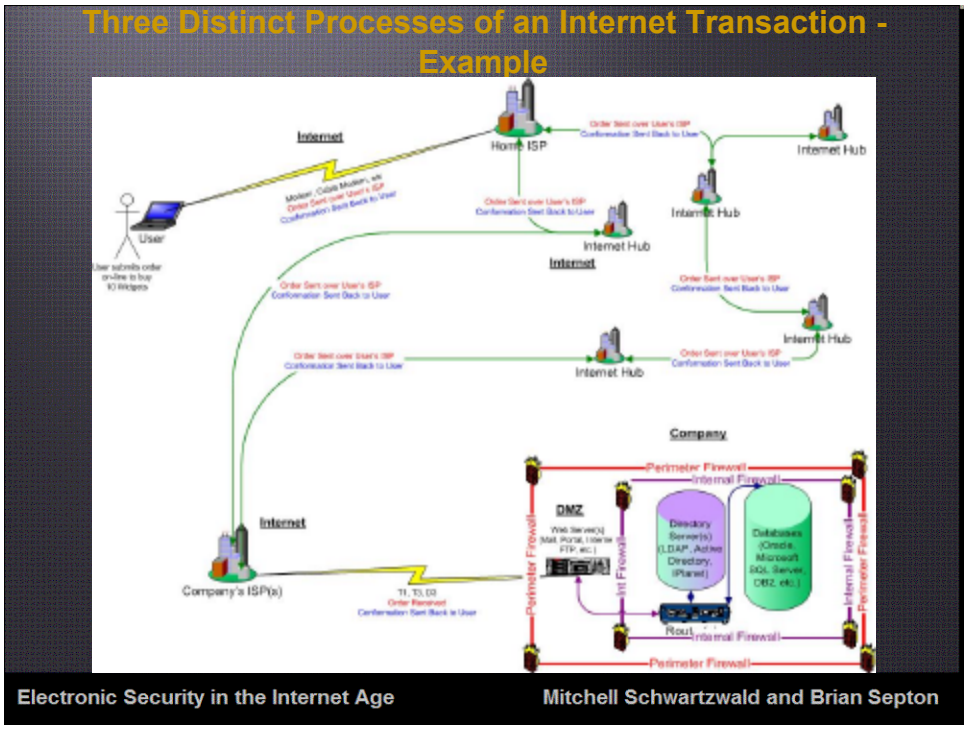

Chart 2