SECURITY NOW!

Transcript of Episode #43

Open Ports

Description: This week Leo and Steve cover the broad subject of “open ports” on Internet-connected machines. They define ports, and what it means for them to be open, closed, and stealth. They discuss what opens them, what it means to have ports “open" from both a functional and security standpoint, how open ports can be detected, whether stealth ports are really more secure than closed ports, and differences between TCP and UDP port detection.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-043.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-043-lq.mp3

Leo Laporte: Bandwidth for Security Now! is provided by AOL Radio at AOL.com/podcasting.

This is Security Now! with Steve Gibson, Episode 43 for June 8, 2006: Ports. Security Now! is brought to you by Astaro, makers of the Astaro Security Gateway, on the web at www.astaro.com.

Steve Gibson is ready to talk about your ports.

Steve Gibson: I’m so excited.

Leo: You really, I mean, I think probably for many people the first kind of introduction to the concept of ports came from ShieldsUP! and Steve Gibson.

Steve: Well, and as far as I know I coined the term “stealth.” That was one that I sort of, you know, in the whole ShieldsUP! Star Trek theme thing, I thought, okay, what are we going to call a TCP port which is neither open nor closed? And I thought, oh, stealth, like, you know, the cloaking field and all that stuff from Star Trek.

Leo: And of course, as usual, all of this stuff really was the bailiwick of business and enterprise computing, networking and all that stuff. But as more and more people have multiple computers in their home, suddenly we’re all becoming networking experts, and the topic of ports becomes very important. Now, it’s a little bit of a confusion because we’ve always had ports with PCs, but they used to be serial ports and printer ports. And that’s not the kind of ports we’re talking about.

Steve: You're right. And in fact you mentioned enterprise. And I remember the day, or the era, rather, where – and this was just – it demonstrates such an evolution in what's going on on the Internet, where it used to be, in the early days of the Internet, when there was some mischief going on, then the IT guys would block a certain port that this mischief was coming into the network on. In other words, there was sort of this presumption of everything being benign, but the exceptions were things being bad. So in terms of, like, a firewall methodology, and the way people were thinking, you had a default "allow traffic in," and then your exceptions were denying traffic.

Leo: Boy, that's changed.

Steve: And, I mean, oh my God, talk about getting fired quickly. If you were an IT guy these days who did that, you know, it'd be like, what are you thinking? Because of course today the world is completely inverted, where by default you deny everything, and you only allow traffic into your border that you know you want because, you know, the Internet's just crawling with junk. For example, we've got this legacy of Windows worms still crawling around the 'Net, probing old vulnerabilities that have long since been removed. But, you know, they're still out there trying to infect machines. And as we know, if you did stick a – if you took a computer you had just installed XP on, before doing any Windows update, before installing any service packs, that Windows XP that we were told was going to be the most secure operating system Microsoft ever created, you put one of those on the Internet and start your stopwatch to see how long it takes to just be just taken over by the junk that's crawling around the 'Net. I mean, it's...

Leo: Sasser and MSBlast and all those.

Steve: Oh, yeah. I mean, it's like a matter of minutes, and stuff is crawling in your machine.

Leo: I liken it almost to herpes or – it's an infection that's endemic. It's everywhere on the 'Net and will continue to be there. Nobody's sending it out anymore, it's just there are infected machines who continue to do it and...

Steve: Yup.

Leo: ...probably won't go away until Windows goes away.

Steve: I think we will probably never get rid of those things. You're right, they're old, unmanaged computers that are just sitting on the 'Net in Lord knows where, I mean...

Leo: Right, in corners.

Steve: ...in strange places, like long forgotten, and they've got this junk in them now. And, you know, the whole point of a worm is that it's self-replicating. So once it crawled into this machine and set up shop, it then began scanning for others. And, I mean, they're just always going to be there.

Leo: And what they're scanning for is, in fact, open ports. Maybe we should define what a port is. And I think the terminology's not great. I think a word like "channel" might be better.

Steve: Well, "channel" would be a great term because, I mean, as we've talked about what we've talked about about the issue of ports in general, what I wanted to do this week is really just focus on this issue of open ports because a lot of people are concerned about them. We get questions all the time, like I have to have this port open, is that a problem? How do I close this port? What does it mean to have a port open? So, yeah, I really wanted to just focus on this issue of open ports and sort of really cover that well to resolve a lot of these questions.

But you're completely right. As we talked about when we were talking about the basic protocols of the Internet – ICMP, UDP, and TCP – we glanced on this before, the idea that, unfortunately, a port, people associate that with a physical thing, you know, like a serial port, a parallel port, a USB port, a Firewire port or whatever. But in fact a port is nothing but a 16-bit number which is carried along at the front in the header of Internet packets which sort of specifies, exactly as you said, Leo, which channel or many channels of 65,535 possible channels this packet is aimed at.

Leo: So there's no physicality to this kind of port. It's not – there's no, like, electronic – 65,000-channel electronic switcher or changer or anything like that. I mean, there's no physical thing.

Steve: Right. Well, the physical manifestation, of course, would be your Ethernet port. And there again we've got that collision of naming. So, you know, your Ethernet connection is the way that this Ethernet traffic, of course, travels into and out of your machine. And it's often doing so through these so-called "open ports."

Leo: But it's not like a part of the cable is port 1024. I mean, it's all coming through the same electrical signal.

Steve: Exactly. Exactly.

Leo: And I guess it helps if you understand that all of the data is sent in little discrete chunks called packets; and that each packet has, as you mentioned, a header. And inside this header is information about where that packet's going, where it came from, and what port, what channel, its surfing number.

Steve: Well, yes. And that, of course, was the great breakthrough of the Internet, was instead of having a switched circuit system where actual physical circuits were being switched, we stepped back from it, and we have a switched packet system, where packet switching is the way machines talk to each other across a network of fixed circuits.

Leo: So the line's always open, and data's always going, but it's routed according to the packeting. Is that Bob Metcalfe's invention? I know he invented Ethernet. Was that kind of the part of that invention, or did that predate that?

Steve: No, that actually predates the actual – the Ethernet is just one of a number of electrical technologies that can be used to carry packets. But, for example, Token Ring was IBM's network.

Leo: And they used packets, too.

Steve: Exactly. So you can...

Leo: I think Vint Cerf might have, I mean, certainly he gets credit for IP. I wonder if – well, maybe not, though.

Steve: Yeah. Well, it was all done, you know...

Leo: A long time ago.

Steve: ...at the beginning of the 'Net, when it was all being put together.

Leo: The beginning of time. I guess Paul Baran is the guy who invented packet switching, or one of the guys who invented packet switching.

Steve: So fundamentally we have sort of like the core for a machine which is on the Internet is the operating system. But the operating system doesn't itself natively have any ports. That is, it supports the protocol and the ability for the OS to communicate by providing services, you know, like the so-called TCP/IP stack and IP services. The operating system will support IP addresses which allow it to accept these packets. But something then after the operating system, something running in the operating system is the actual entity which creates and opens these ports.

Now, for example, in the case of Windows, it may not be a separate application. It might be a service, which really is part of the operating system, but it's a separable part. I mean, for example, you can stop the service or remove the service in order to close the ports that that service opened. And then the next level of distance from the operating system are actual applications which are running in the operating system, like you and I are using Skype right now. It's running as a program on top of the operating system, using the operating system's lower level networking facilities to allow it to communicate out on the Internet so that our two Skype clients are able to connect to each other.

Leo: And Skype is kind of independent of what port you're using. In fact, you can in Skype say, no, use this port or use that port. It works exactly the same. It's not tied to the port.

Steve: It doesn't care, exactly. So there are – also as we talked about before there are sort of what's called "well-known port numbers," where, for example, DNS, the domain name system that allows web names to be looked up and matched to their IP addresses, by agreement it uses port 53. And so the DNS server is listening for packets coming into port 53 of any computer that it's running on. And your own clients are sending their data out of port 53, bound for whatever DNS service they've been configured to operate with. And of course we know the web uses port 80, and 443 for SSL-secure connections, and on and on. So there's a large array of well-known ports, the idea being that systems will by default have services running in them, listening for incoming traffic on those ports.

So if we remember what we were talking about when we were talking about TCP before, the Transmission Control Protocol, the idea is that the operating system is doing the work for the application of establishing and sort of getting the connection going. And this is where this notion really of an open port comes from because, when a connection wants to be established with a machine, a SYN – which is short for "synchronize" – packet is sent to that IP address that the computer is listening on. If it's a TCP port which is open, which is to say there's something that has said "I want to accept connections that are coming into this port," or as we've said, really sort of a virtual port, more like a channel, then the operating system will send back what's called a SYN/ACK, its own SYN and an acknowledge of the receipt of that incoming SYN. Well, that's sort of this whole key of what makes the port be open is that anybody, literally on the planet, can send one of these SYN packets at someone's machine. And if it responds with a SYN/ACK, then we know that something is there at that IP address, even if it's across the planet, which is ready to accept a connection and have some sort of transaction with us.

Leo: So cool. Every time you describe these things I just am impressed with how they thought this stuff up.

Steve: Well, and that has survived the test of time, I mean, so well. And so that's really what it was that got me thinking about ShieldsUP!. Back then – and we're talking years ago – I was setting up an ISDN connection for my computer.

Leo: There you go. That'll tell you how long it was.

Steve: Exactly.

Leo: Pre-DSL, pre-cable modem, you know, ISDN.

Steve: Exactly. It was an ISDN connection. And I was aware of this whole issue of ports and security. And so I got one of the – I just downloaded one of the free online scanners that were available on the Internet, and still are. And I just – I was curious, like, what was going on in the neighborhood of the IP address that we'd been assigned. So I just set the scanner up to, like, scan, I don't know, you know, the hundred IPs plus or minus where the IP that we'd been assigned was. And there were all these computers that had – and in fact this particular scanner was scanning for Windows filesharing. And, I mean, literally the names of machines and the C and D drives, wide open, exposed on the Internet.

Leo: That must have been a shock. Wow.

Steve: Well, yeah. And it was enough of a shock that it was – I thought, you know, nobody knows about this. This needs to receive attention because people were putting their Windows machines, hooking them directly to the Internet – this was before personal firewalls, before NAT routers – just literally plugging them into the Internet. And by default, Windows machines all had this filesharing port open. Meaning that, even if the user hadn't shared any files, they still – Windows had all these services that were running in the machine by default, accepting incoming connections from anyone on the planet.

And so it was finally, I mean, it was that, the recognition that this really needed attention, that caused me to just say, okay, I'm going to do this thing that's going to make it very easy for people to check their systems to see if they're in this kind of danger. So, you know, the very first version of ShieldsUP! primarily checked for Windows filesharing. And then I expanded it in several follow-on generations to do – like, for example, now it does a full 1056 port scan to check from ports actually even including 0, which is not a legal port, but it turns out there are some vulnerabilities in routers that will accept traffic on port 0...

Leo: Oh, wow, interesting.

Steve: ...all the way up through 1056, in order to look at

even the low client ports under Windows. But anyway, the idea was that – or is of TCP – that software running in the system will instruct the operating system to open a port. What that means, then, is that that port will affirmatively respond to incoming traffic. Well, now, an open port responds affirmatively. But it turns out that even a closed port, that is, a port for which there is no listening software associated with it, there is no program that has told the operating system, I want you to accept on my behalf traffic coming in and do the low-level housekeeping work for me of setting up a connection. In that case, a packet coming in and hitting a standard TCP/IP stack will generate an affirmative denial of a connection attempt. Normally it'll get back a reset, or sometimes an ICMP message saying there is no service available on this port at this IP. So although you haven't confirmed that you've found something potentially vulnerable, for example, a service that you may be able to exploit by virtue of the fact that it's going to accept a connection from you, and you're able to mess with it, what you have confirmed is there's a computer of some sort listening for incoming traffic on that IP. So those ports are considered to be closed, but they're still known to exist.

And of course then the next stage of this is a so-called "stealth port," where incoming traffic hits the machine. If the port is not open and would normally respond in some affirmative fashion, saying no traffic is being accepted on that port, instead the machine is completely mute. It just says nothing. So, and that's, of course, exactly the response that you generally get for a dead connection, where there's just nothing on the IP at all.

Leo: And that's your so-called "stealth mode."

Steve: Which of course has now become, like, the way to be on the Internet. It's interesting, I mean, there are people who argue that stealth is bogus.

Leo: What? Really?

Steve: Oh, yeah. You know, it's the old UNIX guys. And they also dislike the idea that stealthing a machine technically breaks the IP or the TCP...

Leo: Oh, I get it. It's out of spec.

Steve: It's out of spec, exactly, because...

Leo: Well, that's a purist point of view. But frankly, if you think about it, if bad guys come a knocking, what's the best response? We don't have any money in here, or nothing?

Steve: Yes, or, exactly, there is no "in here."

Leo: There is no "in here." Nothing exists at this address. Move on.

Steve: Right.

Leo: So, I mean, you can be a purist about it, but frankly I think it's pretty obvious what the best choice is.

Steve: Well, yes. And the fact is, since it costs nothing to be stealth, why not be stealth? I mean, since it costs nothing to be invisible, it seems to me it's better, exactly as you said, to be completely invisible on the 'Net than to say, I'm here, but all the ports you've checked so far are closed.

Leo: Now, it does come up from time to time that, well, this happened with the identd port, where a router manufacturer decided it wasn't a good idea to stealth that port because some services were still using it, and an invisible port wouldn't be an appropriate response.

Steve: Well, it's a very good point. The example you cite, the ident port, what happens is, when a user is trying to connect to a server – and this is generally, I mean, just ancient servers. I mean, there are some IRC servers, some really old web servers, sometimes some FTP servers. Part of the connection protocol is when a request comes in to the server, it sends back an ident packet to the ident port at that user's IP, because in the old days people would have these things called ident servers where they would list a whole bunch of information about themselves. I mean, who's going to do that today? I mean, nobody, because basically you're sort of saying, here, here's everything you want to know about me.

Leo: Come on in.

Steve: Exactly.

Leo: Yeah, yeah.

Steve: So it had been forever since anyone actually ran an ident service. But what the server that makes the query wants is at least to get an affirm...

Leo: An acknowledgement.

Steve: Yes, some sort of affirmative statement that, yes, there's a machine here, but nobody's home. So what normally happens is the ident uses TCP protocol. So the server will send a SYN packet, trying to establish a connection in the reverse direction, back to the client. Well, as we know, TCP is very patient about getting a connection established. It'll send a SYN packet. If it doesn't hear anything else, it'll send another one. Then it waits twice as long and sends another one. Then it waits twice as long again and sends another one. Some machines will send up to five packets, and you can end up waiting a minute before the thing finally decides, okay, there's nobody here. The problem is that all of that suspends your main connection to the server, that is, the server, everything just stops on the server while it's trying to establish this painfully slow process of getting a TCP connection. If the far end did say no, I have no ident service, by sending back an ICMP or by sending a TCP reset packet, then at least the server would know, oh, okay, no service, but there's somebody here. And it would typically just then – it doesn't really care about the ident, it's just old technology that is still in some servers on the Internet.

Leo: So what is the harm in doing that, then? I mean, now I'm going to play devil's advocate and say, well, in that case, why do we bother stealthing that port?

Steve: It's just that, well, actually I think probably GRC is at fault. I mean...

Leo: It's your fault.

Steve: I'm not kidding. I mean, I was showing everybody that their ident port was not stealthed, and stealth became a cool thing to do, and people began asking their router manufacturers and their personal firewall manufacturers, "Hey, Gibson says my ident port is not stealth. I want to be stealth." And so just really due to popular demand the router manufacturers said, okay, fine, we'll stealth the port. Well, the problem then is that some connections will stall when you are going out through a router or out through a personal firewall which stealths the ident port.

So then the next generation of this came along, and that was adaptive stealthing, or adaptive ident stealthing, where the router would be smart, and it would stealth the ident port from any source IP, that is, the remote server trying to open an ident connection back to you. It would stealth it unless it saw that you had an outgoing connection to that IP. Which is a perfect solution.

Leo: There you go.

Steve: So if you've established a connection, and it asks back, well, you got an ident server, you can respond to that. Then you say no, I don't. But at least the far end server is happy that you exist. You acknowledge that immediately, and then you get on with your main connection establishment.

Leo: Ah. But here's the thing. Would you mark that as a stealthed port?

Steve: Yes, in fact, I do it on purpose. ShieldsUP!, it checks the user's machine from an IP different than they are connecting to us from.

Leo: To avoid this thing.

Steve: Exactly. So I do it on purpose in order to give them credit for and to show that their router is stealthing ident for random sources of IP addresses out on the Internet, not the ones that they're actually trying to connect to. So it ends up being a very useful thing. Now...

Leo: We should just mention that the reason that you want it to be stealth is any indicator, even on a completely safe port like the ident port, any indicator that you exist could be a message to a hacker, well, at least there's something here you might want to keep investigating.

Steve: Well, here's a perfect example, Leo, and that is denial-of-service attacks. If you piss off somebody on the Internet who's got control of even a small botnet, and they decide they're just going to DDoS you into oblivion, well, they'll blast you for a while, and then typically stop the attack so that they can see if you're still there. Well, you know, you'd very much like them not to be able to tell that you're still there.

Leo: Yes, yeah.

Steve: Only if you're stealth can you pull that off. If they're able to ping you or to bounce packets off you or try to open a connection and get back an affirmative closed state from you, then they'll know you're still there.

Leo: And it's fairly trivial to actually test each and every of the 65,000 ports. I mean, computers are fast. So even if there's but one open, or not even open, closed but not stealth, they'll know you're there.

Steve: Well, yeah. And in fact, if you're running a system that is not stealthing you, every port will at least say either it's open or it's closed. So in order to be completely off the 'Net in appearance, you really do need the technology which is going to stealth you. And as a matter of fact, I've seen dialogues where hackers know that ident is often not stealthed. So they're specifically trying to open an ident connection because, unless it's adaptively stealthed, as all the latest firmware and personal firewalls are generally now able to do, it will look like it's closed, and they'll know you're still there.

Leo: So thanks to GRC.com and ShieldsUP!, all routers, all consumer-grade routers that ship these days, ship with stealth turned on.

Steve: Yeah. Yeah. It's, I mean, it's the right way to go. There's just no good reason not to be stealth where you can be.

Leo: Now, am I throwing all the value of stealth out, though, by having some open ports?

Steve: Probably not because you don't know what it is that might be looking for you. You might have, you know, a hacker might specifically be scanning for a new vulnerability which has just been found, like in MySQL. And so it might be looking to see whether you have a SQL database server port open, so it would be specifically checking for that port.

Leo: And if you look at the hacker tools, they usually will say, what port do you want to hit, and what range of IP addresses do you want to test?

Steve: Right.

Leo: And you could test a range of ports. But for efficiency's sake they may just be going after that one port.

Steve: Well, now, it's also necessary, since we really want to cover the topic of open ports well in this particular episode, it's necessary to talk about the fact that UDP protocol is every bit as viable as TCP. But because it doesn't have this whole introductory handshaking going on, where you send the SYN and the SYN/ACK comes back, or you send the SYN and a reset comes back, UDP ports will generally operate or may operate differently. That is to say that, as we know, UDP doesn't have this connection establishment handshake, which is really the benefit for very short-term connections. For example, the DNS protocol for domain name services, generally you just send a single packet off to a DNS server, and it sends you a single-packet reply. So it's extremely efficient. Since DNS is going to be transacting such small amounts of information, you wouldn't want to go through all the trouble of having a three-way packet handshake, then send your request, then get the reply, then have to shut down that existing or established connection through another series of packets.

Leo: So DNS uses UDP.

Steve: Exactly. Well, DNS...

Leo: I didn't know that.

Steve: It actually uses both. It'll use UDP. But there's a limit. One of the reasons that UDP actually isn't convenient is if you need to send a lot of data because generally UDP is sort of packet oriented. Now, again, all of these things sort of have caveats. For example, you and I are using UDP right now for streaming substantial amounts of data between each other during this podcast. But what's happened is a protocol on top of it, well, I mean, the typical VoIP protocol is called SIP [Session Initiation Protocol], which is used on top of UDP to sort of give it the ability to do more. But in the case of DNS it is possible to connect with TCP to a DNS server and then make your queries that way if you needed to for some reason. And for example there's something in DNS called a "zone transfer," where you basically say tell me everything there is to know about GRC.com, for example. And if zone transfers are allowed, which for many security purposes nowadays they are not, but in the old days you only could use TCP for one of these so-called "zone transfers," where you're saying I want to know about all the machines within the GRC.com domain, the MX or, you know, the email servers, and everything going on. Give it all to me. And you cannot do that over UDP.

So in general, UDP is a much more quick, simple, lightweight protocol. It also means that you might have a UDP server with an open UDP port, as opposed to an open TCP port. And you would not really be able to tell that it was there unless you asked it in its own particular protocol. For example, if you wanted to find out if someone was running a DNS server, you'd have to send a DNS query to port 53 and see if you got a response. Whereas the whole opening connection dance with TCP is generic. You do the same three-way handshake no matter what service, whether it's web or, for example, DNS over TCP or FTP or any other TCP-based protocol.

Leo: So if you're a hacker sniffing TCP ports, you could just send a SYN to every port, one after the other.

Steve: Exactly.

Leo: And say hello. And that'll tell you that port's there. But if you want to sniff UDP ports, you'd actually have to use the appropriate protocol on each port.

Steve: Exactly, in order to satisfy the server that may or may not be listening. Now...

Leo: So it's much more complicated to sniff UDP ports, then.

Steve: It's a lot more complicated. Although, again, the original spec for the UNIX machines, you know, where all this originated, does say that if a UDP packet arrives where there is no service listening and that has told the operating system that it wants it to forward packets that arrive on a certain port to it, then the operating system should send back an ICMP, a specific ICMP message saying there's nothing listening to this port. So UDP ports can, by default, show themselves as being closed, that is, you get back something saying there's nobody here. So again, you'd like to stealth that behavior. And of course that's what personal firewalls and routers do.

Leo: I think this is great. You know, when you – we deal with this all the time. When you go into your router, for instance, to port forward, to make some port work, let's say you've got a router that's, you know, rightly so, blocking all ports, but you want to use a, you know, you want to set up a server for World of Warcraft, you'll see all this. You'll see UDP versus TCP, and which port number, and all sorts of stuff. But now you know what it means. Now you know what you're doing.

Steve: Well, it's interesting, too. You were talking about port forwarding. And I remembered that that also bears on the ident port because there are still some older routers whose firmware does – it will not stealth the ident port. It will respond that there's nobody here. But a really fun workaround is to forward that ident port – which, by the way, is 113 – you forward that to a nonexistent IP address behind the router, that is, on your own network. So, for example, if your IP address was 192.168.0., you know, 1 to 100, you could tell the router, forward that to .0.200, a machine that you know will never exist. And what the router does is dutifully accept that incoming ICMP packet and sticks it on your network, aimed at an IP that doesn't exist. Well, since it doesn't exist, there's nobody there to answer the call, and you end up stealthing your ident port if your router otherwise would not do so for you. So, I mean – and of course that works for anything that you want it to. You could name any ports that you wanted to stealth, if the router wasn't, just off into the twilight zone, to an IP inside your network that doesn't have a machine listening on it, and those packets are just going to go nowhere. They just end up being dropped.

Leo: Well, I think you – have we covered the subject? Do we know everything we need to know about ports?

Steve: The one thing that I think is worth mentioning to people is that all of this problem – which has been lots of history, you know, we talked first about this notion of firewalls by default allowing traffic and then IT guys blocking only the mischief and how that's completely flipped around. Well, the old days of Microsoft Windows, and for that matter other operating systems, generally had lots of things listening because there wasn't a compelling reason not to. And of course in Microsoft's case, Microsoft always wanted to default towards allowing traffic because they just wanted things to work. I mean, and they sure did, boy. You know, you stuck your Windows machine on the Internet, and you could share your files with everybody in the world.

Leo: Worked a little too well.

Steve: Whether that's what you had in mind or not. And they want, you know, they wanted Windows so that, when you click your machines together into a network, they can all see each other, and they can all happily share files. Unfortunately, putting Windows onto the Internet was the same as putting it on your network.

Leo: Right.

Steve: So it's really worth mentioning that this is all changed now, finally. I mean, and it took – I don't know why it took so long, but it did. It's changed with Service Pack 2 of Windows XP, where there is a built-in firewall, and it is on by default. And, you know, there are people who are still downloading my DCOMbobulator and my UnPlug n' Pray utilities. Those are things which I created in a day, immediately after a new vulnerability had come out and well before, in some cases months before, Microsoft did anything to deal with it. And those things I created to, like, kill off those ports or shut down those problems because we still didn't – many people did not have personal firewalls. XP didn't have a personal firewall in the beginning that was turned on all the time. Earlier versions of Windows never did. I mean, back then people were still using, you know, 95 and 98.

But it's really the case that these problems have been solved just by, first of all, by people having NAT routers. I mean, if you've got a NAT router in front of your system, it matters to a far lesser degree what ports are open on your machine itself. And you can see that because, if you use ShieldsUP! at GRC, it'll show you everything is stealth, even if you've got open ports on the computers in your own network. The reason being, nothing gets through your NAT router. We're testing your public IP, not those private IPs that no one can access anyway because they're not routable on the Internet. There's no way I can send traffic to 192.168.0.1 in order to test it because that IP won't go anywhere. I can't, you know, tens of thousands of people have that IP on their machines behind their routers.

Leo: Probably millions by now.

Steve: Millions, I'm sure it is, yes. So, you know, many, many tens of thousands. So it really is the case that this problem with computers having open ports has really been mitigated, first by the advent of routers, and secondly, for those who are not behind a router, certainly with a personal firewall which is on and doing its job as, you know, the built-in firewall in Windows XP does. Which really means, then, that the frontier for the concern for open ports is ports opened in routers.

And so the last thing worth talking about is the people who are worried that, for whatever reason, they have to have exposed open ports. You know, what does that mean, to have an exposed open port? Something, you know, where they're just not able to be stealth because they need to have services that are available out on the public Internet. And this is interesting because it factors exactly into the discussion we've had about buffer overruns. Because unfortunately the exposure of an open port is that traffic is going to be flowing back in through your router, then to whatever machine you have designated on the router will receive that traffic. And presumably you have something there on that machine, some application which is then going to be accepting the traffic.

The problem is, as we know, it is very difficult to write perfect software. You know, the classic boondoggle of an open port was pcAnywhere, which many people were using in the early days of the Internet because it allowed them to connect to their machines at home and do whatever they wanted to. That's why it was called pcAnywhere. The problem was...

Leo: Anything anywhere.

Steve: ...it had, yeah, it had serious security problems that were being found one after another after another. Many people didn't even take the trouble to put a strong password on pcAnywhere. So everyone knew what the default password was. And people, you know, bad guys would scan the 'Net for the standard pcAnywhere port and connect to people's machines who never took the time to change the default password. So the problem is, if you've got ports exposed, if you've got ports open, it is something you need to recognize as a potential problem, and that is that you are then depending upon the security of and the proper functioning of whatever software package it is which is listening to those ports. And in fact, when I fired up Skype just now, Leo, in order to establish our connection, I got a message telling me that there was a new version available because a security problem had been found and fixed in Skype. So it's like, okay, I'm going to update myself right now.

Leo: Right, right, right. Any time you're running a service of any kind, in order for that service to work you have to open a port. And that opens up your system to trouble if the service has a bug. And as you point out, it's inevitable. There's always bugs.

Steve: Yeah. It's just so difficult not, I mean, this was the huge problem that Microsoft had with all of their services. I mean, virtually every single one of them...

Leo: Something was wrong with them.

Steve: ...had multiple problems that were found and exploited. And, I mean, that's where the worms came from that we were talking about before is specifically from these kinds of problems. So, you know, the good news is, security is on everyone's mind. Certainly security is foremost in the minds of anyone writing applications. I would say the only piece of advice, if you have to have ports open, is try to use robust, well-tested services that you have every reason possible to believe are not going to have problems. And in fact, you know, if you really wanted to go a step further, and you had the ability to, I would say run those machines separately. That is, you know, it may be the case that you've got an old computer. Let it be the one on the front line in the so-called "DMZ," where it's going to be receiving that traffic, and not run those services on your main machine, where you really have much more valuable data, and you want to make sure nothing is able to crawl into it.

Leo: And it's another reason why people should go out and get routers, if they don't already have them, and use them. And the minute you do, in fact every time I install a router, and as soon as I've changed the password and turned off Universal Plug and Play, I'll go to GRC.com and run ShieldsUP! and make sure that I don't have any unstealthed ports. And that's what a great service that is.

Steve: You also want to make sure when you're setting up a new router that you remember to turn off anything that's, like, WAN-side stuff. Many routers have, like, WAN-side administration where...

Leo: Unh-unh. That means the other guy can administrate your router.

Steve: Exactly. Anyone on the Internet.

Leo: Bad idea.

Steve: That's not a good thing to have.

Leo: Not a good thing to have.

Steve: I mean, again, if you have to use it for whatever reason, then you want to take the time to do a really good – to choose a really strong password that no one is going to be able to guess because, if your router is accepting a connection on its standard WAN-port, then somebody out there could just sit there pounding away on it, doing a brute-force password attack, trying to get control of your router. It's certainly better, first of all, not to run it on the standard port. Move it, always move those things to a different port, if you have to have them at all, and then run a really strong password.

Leo: Yeah. And you could be sure that, if it's out there, somebody's banging on it. That's the other thing we've learned on the 'Net is that you can't just kind of skate anymore. People are out there all the time.

Steve: Well, for example, the way I've got my equipment at Level 3 configured for the GRC network, I need to be able, if the worst happened and I needed to reboot a machine, I need to be able to power cycle the machine or get console access remotely. So I've got some equipment which are neat little rack-mounted boxes. But all they have is telnet. They don't have any provision for stronger authentication. I can't do SSH or SSL.

Leo: Ooh, that's not good.

Steve: No, it's horrible. And there's no provision for changing from the default telnet port of 23. So I've got these three boxes sitting there that I have to have access to from the outside. I mean, that's the whole point of them is I'm able to get to them from my home network or when I'm on the road. So the problem is, they will only listen to port 23. They do provide a password, but it's just eight characters.

Leo: Oh, man.

Steve: And, I mean, and it's my network. It's the GRC network.

Leo: That's terrible.

Steve: And if someone accessed it, you know, they could turn off the equipment at GRC.

Leo: Yeah, yeah.

Steve: So obviously the only reason I'm saying this on a podcast is I've solved the problem.

Leo: I was going to say, you're asking for trouble here. What did you do?

Steve: What I did was, I found a really nice managed switch. I have a Dell managed switch, which is surprisingly inexpensive, which allows me to filter those ports and only allow specific IP ranges to see them at all. So only...

Leo: So only somebody from your IP address can log in at port 23.

Steve: Exactly. And in fact that equipment, it doesn't exist for anyone outside of specific networks which I have pre-designated as being allowed to send traffic in.

Leo: That's a good way to do it. That's super stealth.

Steve: Well, and, I mean, you have to. Because you just can't have a service exposed on the Internet, especially a well-known service, especially from a well-known company. It's just going to get attacked. Someone's going to write something that sits there and starts with A and then B and then C...

Leo: It's not going to take long. Eight letters?

Steve: Exactly, and does a brute-force attack.

Leo: Well, I just want to circle back and say that we can thank in particular two different people for packets, the notion of packets. I did say Paul Baran. He did this research in the early '60s at Rand Corporation and wrote a paper on the idea of a packet-switch network. And a Brit named Donald Watts-Davis who simultaneously, but independent of Baran, wrote some papers on – in fact, he's the one who coined the term "packet switching" and describing that idea. And it really does go back to one of the great pioneers of the Internet, Len Kleinrock, who wrote some papers theorizing that the best way to do this would be with packets and, in fact, created the idea of a notion of data blocks to solve that issue of data flow. So it's been around for a long time. And I have a poem I want to read.

Steve: Okay.

Leo: Do you mind?

Steve: No.

Leo: This is – I'm going to tell you the story about this poem in a little bit. But it's been going around the Internet for years. I'm just going to read one of the verses: "If a packet hits a pocket on a socket on a port, and the bus is interrupted as a very last resort, and the address of the memory makes your floppy disk abort, then the socket packet pocket has an error to report.” Just thought I’d pass that along to you.

Steve: That’s pretty good. I like that.

Leo: Actually it’s quite a bit longer. It’s written by a guy named Gene Ziegler, who is at Cornell. Wrote it in ’64, and it’s been going – or ’94, I should say. But it’s been going around the Internet as written by Anonymous. But it’s a long parody of Dr. Seuss that is really quite funny. And I’ll put a link in the show notes.

Steve: I was just going to say, put a link in the show notes, yeah.

Leo: Maybe what I’ll do is I’ll read it, giving credit to Gene Ziegler, and put a copy of the recording up. I read it years ago on “The Site” as Dev Null, the virtual character. And it goes on, I mean, I’ll just read the last verse. “When a copy of your floppy’s getting sloppy on the disk, and the microcode instructions cause unnecessary risk, then you’ll have to flash your memory, and you’ll want to RAM your ROM. Quickly, turn off your computer and be sure to tell your mom.” And the page is “A Grandchild’s Guide to Using Grandpa’s Computer.” He wrote it after his grandkids messed up his Mac.

Steve: That’s very cool.

Leo: And we also, of course, want to remind people that GRC.com is available 24 hours a day to check your ports, baby. ShieldsUP! is one of the many resources, valuable security resources, Steve makes available for free. But it’s all supported by his great program, SpinRite, the ultimate disk recovery and maintenance utility, which everyone should have a copy of in this entire world. And if you don’t, go to GRC.com and get yourself one. And also, if you want 16KB versions of the show, thanks to our transcriptionist, Elaine, those are also available at GRC.com/ – I’m going to do this – securitynow.

Steve: Yup.

Leo: They’re waiting for the htm. No htm necessary.

Steve: Yup. No www, no http, anything.

Leo: Hey, I want to – we got a note from Alex Neihaus who is at Astaro, our great sponsor. And you remember that last episode we were talking about NAT traversal, maybe two episodes – no, I guess it was last episode.

Steve: Yeah, it was last episode. We talked about how NAT traversal works and then the notion of friendly versus non-friendly routers.

Leo: Right.

Steve: That would behave or not, depending upon how they mapped the ports through the router.

Leo: And of course Astaro makes, you know, the Security Gateway software. So he actually sent a note to his engineers saying do we do this, and they actually do it, it sounds like quite right. Now, I didn't fully understand that. But...

Steve: Actually they do it so right that, I mean, it's like the best way you could. What they do is – and this is the Astaro Security Gateway. When it's running in a NAT mode, they will leave the source port unchanged as it moves across the NAT.

Leo: Most routers do not do that; right? They change the port.

Steve: Correct. Most routers just make up a random port and assign it in a table, so they're always changing the source port. And what you're hoping for is that the source port will be the same, even if the destination IP is different. That's the critical feature that you need for peer-to-peer-friendly NAT. Well, the Astaro NAT is, like, the best it can be because it will leave the source port alone as it crosses through the NAT translation, only changing the source IP from the machine behind the NAT to the NAT address itself, so that the packet is able to come back. And what's very cool is that the only time when it will change the source port is if you happen to have two different machines, both communicating on the same source port, both to the same remote IP, because there it's very clear you would need...

Leo: You could have a collision.

Steve: ...you would have to change the source port in order to disambiguate those two machines from the outside. But unless that's necessary, the source port is not changed, which means that the Astaro NAT is like, I mean, it's going to be the friendliest NAT you could ever have.

Leo: That's slick. That's real – and by the way...

Steve: And it's free.

Leo: ...I'm using one. Yeah, it's free. You can get the software for free. I'm using one right now, and I'm really happy with it. I feel kind of powerful. I do want to mention Astaro is, of course, our sponsor. And we've mentioned before that you can get the Astaro Security Gateway software for home users absolutely free. For a little bit more you can upgrade it to spam, antivirus protection, and it really is powerful stuff. But I also want to mention that there is a new managed system, the Astaro Command Center. I've been looking at the screenshots of this. It is so slick-looking. ACC v1. It is free for users of Astaro Security Gateway, so I'm going to download it. And it's really designed for network administrators who have multiple gateways. It allows you to manage and control those gateways from a single slick-looking dashboard. I mean, this thing is gorgeous, really looks good. Includes a world map so you can see where your gateways are all over the world. You can, you know, it has this monitoring so you can see what the threat levels are, I mean, I don't know about threat levels, but resource usage. You can't see the threat levels, but let's hope you don't have any threat levels. But the resource usage for all the gateways in the network, and you can coordinate them and manage them, you know, startup, shutdown, and maintenance and all of that stuff.

Steve: Well, you know, I've got some friends that manage the security for, like, a bunch of small networks. And this sounds like it'd be just the thing for them.

Leo: Exactly. Exactly. So if you're already using ASG, just go to Astaro.com, and you can download the Astaro Command Center v1 from the Products section. Which, you know, I mean, I think it's one of the nice things about using software like this, open source software like this, is it gets better all the time. And it's just wonderful. I mean, you really get a real benefit from it.

Steve: It's the future.

Leo: I almost wish I had a big managed network. It wouldn't be any good for me. I just use the home version. Astaro.com. And we thank them for their support. And of course we thank the folks at AOL for supporting the show with bandwidth, which is always an issue with a show like this, with hundreds of thousands of listeners. That bill can add up, but AOL's been very generous. And we encourage you to find out more about podcasting at AOL on the AOL Radio Channel by going to AOL.com/podcasting.

Steve, I'm so glad you did this. I think ports are, you know, probably the single most confusing and interesting topic, and certainly the thing that we all have to deal with all the time.

Steve: Well, yes. And I think it's that they're so visible. I mean, it's the thing people can see, and it causes concern. So I just really wanted to cover that really well.

Leo: And it's one of those things in the computer world that really I don't think anybody ever intended end-users would have to deal with. It wasn't designed for end-users. But we do.

Steve: Oh, I mean, neither was http://.

Leo: Exactly. Exactly. Tim Berners-Lee is embarrassed that anybody has to see that. But that's how things evolve, and that's the way it is.

Great, Steve, we'll see you next week. You have any idea what we'll be talking about, or...

Steve: Absolutely. It's Episode 44.

Leo: I was going to say, it must be a Mod 4.

Steve: Yup. So we'll do Q&A. Anybody who's got any questions, they can go to GRC.com/securitynow. Down at the bottom of the page is a form. Send your questions to us.

I’ll read them, and we’ll pick from them and answer 12.

Leo: All right. And of course that’s also a good place to go to the discussion groups, the security discussion groups at GRC.com. And you can get your questions answered by Steve and other experts. It’s really a really wonderful resource: GRC.com. Thanks, Steve.

Steve: Always a pleasure, Leo. Talk to you next week.