SECURITY NOW!

Transcript of Episode #29

Ethernet Insecurity

Description: Leo and Steve discuss the design, operation, and complete lack of security of Ethernet - the LAN technology that virtually all of the world uses. They explain how this lack of security enables a wide range of serious attacks to be perpetrated by any other machine sharing the same Ethernet - such as in a wireless hotspot, within a corporate network, or even in a wired hotel where the entire hotel is one big exploitable Ethernet LAN. GRC's ARP Cache Poisoning page contains a detailed explanation of these problems with diagrams and links to readily available Ethernet ARP exploitation malware.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-029.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-029-lq.mp3

Leo Laporte: This is Security Now! with Steve Gibson, Episode 29 for Thursday, March 2, 2006: Ethernet Insecurity. From his fortress of solitude, deep within Irvine Mountain - are there mountains in Irvine?

Steve Gibson: Yeah, actually, they're Saddleback Mountains, like the big famous mountain range, yeah.

Leo: Deep within Saddleback Mountain, buried under 3,000 feet of granite: Steve Gibson. You don't have, like, this - I just imagine you in this kind of basalt batcave.

Steve: I'm surrounded by screens. That's about it. I've got monitors showing bandwidth. And it's funny, too, when you have too many screens, it's hard to find your cursor. Because it's like, okay, where did this go? You know, which screen is the mouse cursor on?

Leo: Oh, so you - the screens, you can move your mouse across the screens and...

Steve: Yup.

Leo: What are you using for that? Synergy, or what do you use?

Steve: No, I mean, I've got three screens on a single system. But then I also have other systems monitoring other things. Like I'm keeping a track on the IP turnaround between you and me right now. We're about 30 milliseconds roundtrip time for our packet, just so we can see, you know, what's going on, packet loss and so forth. So I've got screens showing, you know, bandwidth monitoring over at Level 3. I can see how many people are connected and what the CPU utilization is, you know, it's bizarre.

Leo: You're your personal lab experiment on the dangers of RF radiation on the human biology.

Steve: Yes. Fortunately, I don't think that'll be a problem, so...

Leo: Hey, he's fit, he's healthy, he's doing great. It's harmless. But you have all LCD panels by now, I'm sure, so...

Steve: Oh, a long time ago. And let me tell you, that really cut down on the air conditioning bill.

Leo: Oh, I bet, yeah.

Steve: Because, you know, those big, huge - I used to like large CRTs. And boy, did they generate heat.

Leo: Yeah. And you were getting a bit of radiation in those days. Those things were bad. Especially - they fired them out back, apparently, so it was behind the machine that was bad, not the front.

Steve: Yeah. I had some cats for a while, and they loved to curl up on top of the CRT. It was like a heating pad.

Leo: And then their hair fell out, and...

Steve: Exactly.

Leo: So do we want to - I don't think we need to. But let's, I mean, since you answered a lot of questions last week. But is there anything we need to loop back around for last week?

Steve: I think we're all clear to start.

Leo: Yeah. I mean, usually when you answer questions you don't get more. Let's talk, then, about Ethernet Insecurity. In a way, this is Part 2 of our conversation from Episode 27, yes?

Steve: Well, yeah. We have several episodes we talked about earlier about WAN technology, how the Internet works, how it uses a hierarchical addressing space. And in fact I did have somebody asking me a question, basically about why Ethernet couldn't be used on a WAN. If Ethernet's addressing is so cool because you just have unique addresses, why not just use it out on the whole Internet, and then you wouldn't have a problem with, you know, spoofing and NAT and all this.

Leo: You're talking about using MAC addresses instead of IP addresses?

Steve: That's what he was talking about.

Leo: Hmm. That's an interesting idea.

Steve: Well, because, you know, we know that MAC addresses are unique. As we know, a MAC address is a 48-bit number which is 24 bits that is assigned uniquely to a manufacturer; and the other 24 bits is, like, the manufacturer's serial number, so that none of its adapters ever have the same MAC address. And because the manufacturer has 24 bits of the total, and then they have their own serial number within those 24 bits, the concatenation, which is a 48-bit binary number, is guaranteed to be unique in the world. And so he was thinking, hey, why not just use that?

Well, the problem is, we absolutely depend upon IP addressing being hierarchical, that is, when an IP address, for example, has 68.something.something.something, all the routers in the world know that anything beginning with a 68 needs to kind of go in that direction, to go aim those packets toward that network. So it's just like the way our U.S. postal system works, where you have, like, a state and then a town and a street and a number, where the first thing that happens is the mail gets sent to the right state. Then it gets sent to the right town. Then it gets sent to the right street. Well, that's a hierarchical system.

MAC addresses are just, you know, a NIC will have any random 48-bit number. So if you had that on a WAN, that is, on a global network, you'd have no way of knowing from the address any idea of where that particular network interface card was located. So specifically it's the hierarchical nature of IP that makes the WAN work. Whereas, you know, Ethernet LAN, because it has what's called a "flat addressing space," that is, just a single 48-bit, you know, it's on your LAN, and you don't need to know where because the whole LAN is one big network, one big sort of solid chunk of wire. And you just address a packet to this 48-bit address on the LAN, and all of the network cards hear it, and then only the one that has the matching address responds to it.

Leo: So you couldn't do this because it would be too inefficient to do it globally?

Steve: Yeah. I mean, to do it, every single router would have to have the complete address list of every single Ethernet adapter in the world. And the fact is...

Leo: That's not going to make sense at all.

Steve: Well, no, it just couldn't work. I mean...

Leo: It wouldn't scale well, yeah.

Steve: Oh, exactly, exactly. And his point was, hey, you know, 48 bits is bigger than 32 bits. Ethernet addresses being 48, they've got, like, a whole lot more addresses; whereas, you know, the IP space on the Internet being 32 is obviously a lot less. So wouldn't that give us more? And so the point is, I mean, it's a really cool concept because, yes, from that standpoint that's certainly true. But it is exactly the fact that the Internet addressing is hierarchical, that is, that the higher bytes, the first bytes in the address give you location. I mean, that simple fact is the only reason the Internet works is that that way...

Leo: We'd better not change it, then.

Steve: Oh, no. That way routers, you know, it's like 68.*.*.*. That is anything beginning with that byte they send in this direction. And so, since there are only 256 first bytes, you know, possible values for the first byte, 0 through 255 - and actually a lot of those, a chunk of those are not even routable. They're like, you know, for example, we know that anything starting with 10 is a private address. So a router, if it receives something with a 10., it just throws it away. There's, you know, nowhere for it to go.

Leo: Right.

Steve: So the beauty of the hierarchical addressing is that it tells you where to send it. Whereas a MAC address is just a 48-bit number that has to be unique on the LAN. But nothing about it tells you where to send it.

Leo: The power of hierarchy.

Steve: Yeah, it's really a cool concept.

Leo: It's like ZIP codes. You don't do individuals - we talked about this a little bit before, but you don't do individual ZIP codes for every locale. You have a hierarchy: 9 means West Coast and 0 means East

Coast and so forth.

Steve: Well, a good example of sort of a ZIP code-ish thing, imagine that paper mail was addressed based on your Social Security number, which is, you know, assigned when you get a Social Security number to each person who has one. And, you know, but the number doesn't tell you anything about where they live itself. It's just, you know, it's a serial number. So if you tried to deliver mail based on a serial number, you'd have to look every single one up.

Leo: Right. That wouldn't work.

Steve: And so anyway, just wouldn't work. So, okay. So what's very cool about the Ethernet is - yeah, the Ethernet, or Ethernet technology is probably the proper way to say it. First of all, it was designed 33 years ago, back in 1972, by Robert Metcalfe. Bob Metcalfe, when he was - now I don't remember where he was.

Leo: Was he at SRI or BBN?

Steve: Oh, wait, no, he was at PARC, at the Palo Alto - he was at Xerox PARC.

Leo: Okay.

Steve: And they were experimenting with locally networking computers. And so, you know, computers cost $100,000 back then, and they were all hand wired and built with chips and things. I mean, these were real expensive machines. So the concept he had was - and it was brilliant back at the time - was to just put all the computers that they wanted to talk to each other on a single link, that is, on a single wire, essentially. And you remember that the original Ethernet used coax. It had the RG whatever it was, 57 or 70, I don't remember now what the number of the coax was. But it was that big, you know, it looked like the same kind of...

Leo: TV cable, yeah.

Steve: Exactly.

Leo: Little thinner, but...

Steve: The same thing that people's cable modems are connected to, that kind of coax. Because he needed what's called "transmission line behavior," where you'd have a single run of coax, and you would terminate it resistively at each end. And then all the computers that wanted to be on this LAN would basically have, like, a T adapter. They would just, like, have an electrical T connection that would run up to their card. And you would hook all these machines together.

Leo: And I remember how fragile that was. I learned how fragile that was because once I disconnected my computer, and I broke the whole ring, and everybody went down. By disconnecting the wrong part of the cable. You want to keep it together.

Steve: Yes, exactly. And you break that, and you could end up with two segments that were unable to talk to each other. Now, there were some other local area technologies, and IBM had one called Token-Ring, where you literally passed sort of a virtual token, which was your talking permission, around and around this ring. And so the machine that had the token was the one that was able to send data out. Bob came up with a different approach, which was substantially simpler. And as often happens, the simpler ones end up winning, although it does have some failure modes.

Leo: It's Occam's Razor.

Steve: Well, exactly. And Bob's approach, and actually I mangled this acronym the first time I talked about it months ago, was Collision Sense Multiple Access with Collision Detection - CSMA/CD. The idea is that, on the LAN, all the computers are connected to the same wire. So they listen for a time when nobody is sending data, that is, they're not receiving any data from anybody else who's on this single wire. So when the wire's quiet, they will put a packet onto the wire that is bound for some other machine on the LAN. There's a chance, though, that two computers that were both listening for silence will, when silence occurs, both put their packets on at the same time. So you could have a collision of these packets.

What happens is that the voltages that result from two cards talking at the same time is easily sensed. So they're able to each determine that, whoops, I stomped on somebody else's transmission. So they back off and wait for a random amount of time and then retransmit the same packet. Well, since they're waiting for a random amount of time, the chance that they're going to collide a second time is relatively low. So basically all the NICs - the Network Interface Cards - on the LAN are listening all the time and waiting for a time to talk. So it turns out that that approach is simple enough to work really, really well. And that's what all of our LAN technology today uses. All of this, you know, nobody in their homes has IBM's Token-Ring or any other LAN technology that fell by the wayside. Everybody's using Ethernet because it ended up just working well enough.

Now, what's interesting is it does have a failure mode when the overall level of traffic on the LAN starts to increase, as you might imagine, since there isn't any, like, formal permission-giving for people to speak on this common, shared wire. Then what happens is the percentage of collision, the probability of collisions occurring, starts to go up as the amount of traffic on the LAN segment goes up. And so what happens is, due to collisions being, you know, fatal for the data, the LAN adapters will back off a random amount of time and try again. Well, there might be another collision for their retry, if not among them, among other cards.

So if you look at a curve of collision rate versus bandwidth for Ethernet, it does, as the segment gets busier, you have a higher rate of collisions, and your overall throughput starts to drop. And of course, once people are having collisions and retransmitting, that in turn causes more collisions and thus more retransmission. So it sort of fails badly. It doesn't survive well when it really gets busy. But that's never really been a problem because most Ethernets are not that heavily loaded. And in fact the solution, if you do have an overly loaded Ethernet, is to use a switch.

Now, what I've been talking about, for example, is in the original Ethernet topology was a single coaxial wire that all of the NICs clamped onto. We went from that so-called 10Base2 technology Ethernet to 10BaseT, which is now what virtually everyone is using - T as in twisted pair. So we changed from literally a physical coaxial cable that would loop around the building, we switched to the Ethernet wires that everyone is now familiar with, where you click them into a hub.

The first technology we had was a hub technology. And the idea was that anything that the hub received, it would send out on all of its outputs. So basically, the way 10BaseT works, instead of having a single, literally a single copper wire surrounded by a shield, which is what the coaxial cable is, with 10BaseT you've got a pair of wires for transmitting and a pair of wires for receiving, a so-called "twisted pair." Thus the "T." So an Ethernet connection is a four-wire connection; even though our plugs have eight wires, those RJ45 jacks are eight-connector plugs. If you look closely and count them, you can see eight. Only four of them are used, two of them for receiving and two of them for transmitting.

So when we are using a hub, everything that everyone sends into the hub over their transmission wires, the hub simply rebroadcasts brainlessly. It doesn't do any thinking about it at all. It just basically, all of the incoming wires are received, and everything is oared together and then sent back out. Well, naturally you can still have collisions because, again, two cards could be transmitting at the same time. They would collide in the hub. The hub would send out this garbled nothing message, and they would both go, oops. Because they're listening to everything that the hub is sending, they would realize their message had not gotten through, and they do the same back-off and resend. So switching to the original 10BaseT technology did not help with this Ethernet saturation and collision problem.

However, what then happened was we changed to switches instead of hubs. A switch is actually an intelligent device, which is the reason they're more expensive than hubs, traditionally, and we're beginning to see hubs going away as the price of switches are coming down just because of manufacturing efficiencies. A switch actually learned the MAC addresses that are on each one of its segments. So whereas the original older coax and a hub had, like, a single segment, that is, every NIC in the LAN could directly hear every other one, in a switch it actually segments the LAN into individual pieces represented by its ports. And so you could have multiple computers on a switch's port because the switch actually has a table of RAM. And if you look at the specifications for a switch, it'll say something like up to 4,096 MAC addresses it's able to memorize, meaning that in there is a 4K table of MAC addresses. And it learns which adapter is on which one of its ports.

Leo: How can it be 4K? I mean, are you saying that each MAC address - it's more than a byte, isn't it?

Steve: Well, 4K would be 12 bits.

Leo: 12 bits.

Steve: Yeah. But the idea would be that you could have 4,000 computers...

Leo: I got it, I got it.

Steve: ...on the switch.

Leo: Right.

Steve: And obviously a switch that's got eight ports normally only has eight computers.

Leo: Yeah, 4,000 is quite a few.

Steve: So they got a lot.

Leo: Yeah.

Steve: But you can chain switches together. So you could have a switch that's got eight ports going to eight switches that each have eight ports going to eight switches that each have eight ports. So that switch up at the top, it would have to know the MAC address of all the computers in the hierarchy of switches down below it. But this the point: it does. So now when a computer is transmitting, its data goes to the switch. The switch actually reads the MAC address that it's addressing its packet to and retransmits it only out of the one port that it knows has that MAC address. So it does a huge job of dealing with the potential problem of Ethernet congestion, which really is, like, the one Achilles heel of the Ethernet from a fundamental technology standpoint. We're going to be talking about a much worse problem with Ethernet, but it's not about sort of the fundamental data-carrying ability. So essentially that's what switches do, is they chop this one LAN into multiple electrical pieces. And they actually learn which MAC addresses are living down which of their ports, and only send the data to the proper location.

Now, one variation on that, or a further detail on that, is what's known as the "broadcast address." Most LAN adapters will only respond to their own MAC address or to the broadcast address. Now, if you've ever looked at the underside of your router or maybe printed on the back of some computers, you'll see the MAC address. It's always expressed as six pairs, or six sets, of hex-paired digits. For example, 00:02:05 would be the starting of the MAC address, would be the manufacturer's serial number assigned to that manufacturer. And then the next three sets, which could be anything, would be the serial number of the adapter from that manufacturer. So you see, like, six pairs of hex digits, separated by colons. That's the MAC address.

So an Ethernet NIC will respond to its own address or to the broadcast address, which by universal agreement is all ones, or in that MAC address addressing it's FF:FF:FF and so on, six sets of FFs. The idea is that that's the way systems are able to find each other on the Ethernet. If through some technology or by reason of some technology it's necessary for one computer to locate another, it's able to do a broadcast on the Ethernet. And all the machines that receive that will take a look at the data. That also means that a switch, which is normally providing some isolation, must rebroadcast out every one of its ports anything that it receives that is addressed to that broadcast address. So one time that the switch looks like a hub is, if it receives something, a packet, an Ethernet packet, addressed to the broadcast address, it just sends it everywhere.

Okay. So let's talk now about how the Internet's IP technology that we've been talking about, the hierarchical addressing, how does that work within an Ethernet LAN, and what problems does that create? The biggest problem with Ethernet - and it's certainly not the fault of its designers 33 years ago because, again, they did this as an experiment at the Palo Alto Research Center run by Xerox just to sort of see if they could. And this is the technology, that is so often the case, you know, the one that worked, we just kept using, never really intending it for primetime. There is no notion of authentication. Essentially, there is absolutely no security with Ethernet. The assumption always was that it would be used in a LAN setting where you knew and trusted everybody on the network. You were one big happy company, or, you know, your little network at home, or whatever. But the idea was that, you know, in a LAN, it's local. It's inherently local.

The problem is that, as new applications for our technologies have occurred, this Ethernet LAN, which is fundamentally insecure, has been used without really giving this any thought in non-secure and sort of semi-public settings. For example, when you visit a hotel, and you plug your traveling laptop into the hotel's Ethernet connector, which hotels increasing offer, you're on the hotel's LAN. And any trouble, any security problems that Ethernet has are now hotel-wide, and you are subjected to them. Similarly in a hotspot. When you're using a wireless, you know, we've talked about WiFi extensively. Well, WiFi is running on top of Ethernet, meaning that it's vulnerable to the underlying insecurity of Ethernet when you're in any kind of a WiFi setting. And of course the same is true even in a corporate LAN. Somebody who wanted to do something malicious in a corporate LAN, which is certainly going to be connected by Ethernet today, they have a tremendous amount of latitude for being able to cause mischief by bringing a computer into the network and just clicking it into the company's network because Ethernet has no security.

Now, what I mean is that there is no authentication, meaning that the adapters are addressed just based on this 48-bit MAC address. Okay, well, we know that everyone today is using IP addressing. On our own local networks we might have, you know, 192.168.something.something, or 10. or whatever. Even out on the Internet itself, routers are interconnected very often using Ethernet links from one router to the next. That's just the way, you know, Ethernet has turned out to work so well and be so inexpensive when you integrate the circuits and all the technology, that it's just sort of the universal glue for all of our computers.

Well, okay. The way Ethernet works is that packets have to be Ethernet packets in order to travel across the Ethernet. Which means that the IP data, the IP packets that are coming into the LAN, have to be encapsulated. They're basically wrapped in an Ethernet packet that contains the MAC address of the originating adapter and the destination adapter. Inside that is the IP data that shows the source IP and the destination IP. But the Ethernet LAN doesn't use that at all. It is all based on MAC addresses. So what has to happen is that there's this relationship, an association between MAC addresses and IP addresses. So that, for example, any computer on a LAN, a contemporary Ethernet LAN, will have its IP address that it's been assigned by a DHCP server or manually configured. And inherently it'll have a MAC address.

So there's basically two addresses. There's sort of the more global hierarchical Internet IP, and also this, what we talked about, this non-hierarchical, just sort of 48-bit fixed MAC address that is how data gets to it on LAN. What this means is that the computers on the LAN, in order to send IP data to each other, which is really what they want to do, they need to have a table that says this IP is assigned to the computer with this MAC address, and this IP, another IP, is assigned to such-and-such a computer. So there's basically an association table that maps the IPs to the MAC addresses. What happens when a packet comes in from the gateway, say that it's coming in off the Internet, and it's addressed to a certain IP. The gateway, the router or whatever it is that is the way the LAN connects to any other networks, it'll look at this destination IP and look to see if it knows which MAC address is associated with that IP. If it does already know because it's sent some data to it in the past, it simply wraps that IP packet in an Ethernet packet and sticks it on the LAN, addressed to its destination. That machine will hear it and accept the packet, take the Ethernet wrapping off, and there's the IP packet that it has been sent.

If, however, the gateway doesn't know the address, doesn't have the MAC address associated with that IP address, it needs to ask the entire network who has this IP. Well, that's where this thing called Address Resolution Protocol, ARP, comes in. ARP protocol is the way MAC addresses and IP addresses are associated and essentially glued together in a LAN. The router sends what's called an "ARP request," and it sends it to the LAN's broadcast address, that FF:FF:FF, basically all ones, that has been reserved for this purpose. So it basically says, hey, who out here on the LAN has this IP? And because it's broadcast, every machine on the LAN will receive that query, will hear it. The one machine that is assigned to that IP will respond to that. And since the broadcast was sent to everybody, but it was sent from a specific MAC address, that is, from the router, the ARP reply is just sent right back to the router. The card who does have that IP sends back, hey, I've got your IP.

So what happens is, that's the way, on a LAN, the gateway is able to learn the IP address - or addresses, because it's possible for one computer to have multiple IPs - of every machine on the LAN. You know, it's a very slick and cool technology, but it's got one real problem. And that is, there is absolutely no way to know if this ARP traffic is valid, or if perhaps it's been spoofed. So imagine, just to reiterate how this works when it's working right, is that the gateway will send an ARP request, broadcast it to every machine, asking who has this IP. Well, similarly, when any one of the machines on the network wants to send a packet outside of the LAN, or even to another machine on the LAN, individual machines will also generate these ARP requests, saying who has this IP? The machine that does, responds.

The way the Ethernet works is, when receiving an ARP reply, the receiving machine simply fills out this table entry in what's called the "ARP table," not surprisingly, and puts the information in. It turns out that a malicious person anywhere on the LAN could send any other computer an ARP reply. And the computer would believe it was real and change the entry in its ARP table, basically updating the entry with this new information.

Leo: So there's no attempt to validate the sender at all.

Steve: Well, there's no way to validate. I mean, literally, there's just, like, no way.

Leo: There's nothing you could ask.

Steve: Yeah, exactly. This was ever considered, never thought about back in the original design of the 'Net.

Leo: So did they make the assumption that no one who has direct access to your LAN would be malicious, or what?

Steve: There was no assumption.

Leo: They didn't even think about it. It wasn't...

Steve: I mean, this wasn't even on the radar. Wasn't even considered.

Leo: We see that a lot with security issues, where just nobody even thought about it because nobody had done it. There were no exploits to that point.

Steve: Well, exactly. Exactly. So let me make this clear because this is really important. If a malicious person sent a computer on the 'Net an ARP reply saying "I am the IP of the gateway," that would replace that entry, the entry for the LAN's gateway IP, with the MAC address of the intruder. From that moment on, any traffic which that computer wanted to send out to the Internet would be addressed to the MAC address of the intruder.

Leo: So you've effectively stolen the connection.

Steve: Yes. You have stolen all the traffic that is bound for the gateway. Similarly...

Leo: Oh, go ahead. I just thought of a...

Steve: Okay, yeah.

Leo: ...problem, because if you're stealing it, it's not going to get to the gateway.

Steve: Well, exactly. So part two is, this same intruder then turns around and sends one ARP reply to the gateway...

Leo: Ah.

Steve: ...pretending to be the IP of the other computer that it's intercepting, that will replace the table entry in the gateway. So anything from the gateway bound for that other computer's IP will instead be sent to the MAC address, that is to say, to the computer that wants to intercept this. In order to keep the connection alive, the interceptor has to forward any traffic it receives onto the original MAC address. So basically it's spliced itself in, in what we know is called a "man-in-the-middle" attack.

Leo: Makes sense. It's impersonating both ends, one to the other...

Steve: Yup.

Leo: ...and taking all the traffic.

Steve: Well, and it is, I mean - get this. It's as simple as sending one packet to each of the computers whose traffic you want to intercept. I mean, literally...

Leo: I'm guessing there's...

Steve: ...that's all it takes.

Leo: I'm guessing there are lots of automated tools that will do this.

Steve: Yes. For example, there's a tool called Cain & Abel. And it's now at version...

Leo: Our old friend.

Steve: Yeah, Cain & Abel. It's now at version 2.81. And if I read from one of their boasting features, they say their latest version is faster - oh, good, so you can intercept traffic even faster than before - and contains a lot of new features like APR. That's their own acronym, and they say that stands for ARP Poison Routing. And they go on to say, "which enables sniffing on switched LANs and man-in-the-middle attacks. The sniffer in this version can also analyze encrypted protocols..."

Leo: Oh, boy.

Steve: ...such as SSH1 and HTTPS and contains filters to capture credentials from a wide range of authentication mechanisms.

Leo: Wow.

Steve: Oh, I know.

Leo: This is freely available on the 'Net.

Steve: Oh, yeah. Cain & Abel.

Leo: Everybody has it. I have a copy. Everybody has a copy of this.

Steve: Okay. And, I mean, actually, sad to say, it's a beautifully written piece of software.

Leo: And it's very useful for a lot of things.

Steve: Yes. Well, also on version 2.8 they talk about, you know, we've talked a lot about Windows Remote Desktop Protocol, you know, Terminal Services, RDP. They say in their features list for 2.8, "RDPv4 Session Sniffer for APR." And it says, "Cain can now perform man-in-the-middle attacks against the heavily encrypted Windows Remote Desktop Protocol, RDP, the one used to connect the terminal server service of a remote Windows computer. The entire session to and from the client and server is decrypted and saved to a text file. Client-side keystrokes are also decoded to provide some kind of password interception. The attack can be completely invisible because of the use of ARP poison routing and other protocol weaknesses."

Leo: This is a sideline, and I'm not going to go too far down this rabbit hole. But I just want to point out, people will say, well, how could this program be legal? You know, how is there any legitimate reason why you would want to own it, you know, why don't they shut it down? Well, a lot of security experts use this program. Right?

Steve: Well, for...

Leo: For testing.

Steve: Well, in fact, it does many things other than this.

Leo: Yeah, it does WEP cracking and, I mean, it does all the bad stuff. But it does - it's a very useful program. And just because somebody has it doesn't mean that they're a bad guy.

Steve: Right. There's another free tool called Ettercap. And from quoting their home page, "Ettercap is a suite for man-in-the-middle attacks on LAN. It features sniffing of live connections, content filtering on the fly, and many other interesting tricks. It supports active and passive dissection of many protocols, even ciphered ones, and includes many features for network and host analysis." And they have screen shots showing it doing all of this.

Leo: Now, you can't fault these guys, I mean, this is a well-known hole, this ARP poisoning, and it's easy to do. And so of course somebody's going to write a tool for this.

Steve: Well, and so this is the lesson, is that, I mean, you know, we see over and over and over, that which is possible will happen.

Leo: Yeah.

Steve: Now, so once again, I just want to - because I want to make sure people get this. Anybody on a LAN, any malicious software, can send two packets, one to each other computer, lying about its MAC address to each of them, and receive - basically receive all the traffic they intended to send to each other. It receives. It then forwards it on. It has spliced itself into the connection, and nobody will know. This can happen at a wireless hotspot. I mean, one bad person in a hotel could arrange to, without much work, literally intercept all the traffic going to and from the hotel's gateway so that all of the email conversations, all of the traffic of any sort that is being transacted by every other hotel guest, they're able to monitor and intercept. And in some cases, where you have weekly authenticator protocols, like Windows Remote Desktop that really doesn't provide any kind of authentication, man-in-the-middle and complete decryption attacks are easily performed. I mean, it is really bad.

Leo: And trivial to do. And if I wanted to just check into a hotel and put my system on the Ethernet and just start running these programs, it'd be, you know, it's easy. It'd take you minutes. You don't even have...

Steve: Well, now, about half - right. About half of the hotels out there are still using hubs. As we know, all the traffic a hub receives it sends back out to everyone else. So if you're on a hub, all you have to do, I mean, you don't have to do anything active. There's no active attack you need. You just plug your computer in and start sniffing.

Now, remember I mentioned that a NIC will respond to something addressed to its own MAC address or to the broadcast address. That's almost always true. However, NICs can also be put into something called "promiscuous mode." Now, all that means is you just switch the adapter into promiscuous mode, and that says just receive everything. And so that's what these sniffers do. Any sniffing tool simply puts the NIC into promiscuous mode, and everything on the wire comes in and gets logged and decoded and checked out.

There was someone we quoted a few months ago when we were talking about this briefly that said that, you know, she was a security expert. Out of curiosity - oh, and she also traveled a lot. Out of curiosity she plugged her laptop into hotels wherever she goes, turns on some of this, quote, "security software," unquote. And in the typical evening, she's able to receive several hundred email log-ons and passwords.

Leo: Wow.

Steve: She knows what they're logging onto, who they are, and what their password is.

Leo: She's probably got mine.

Steve: It's just - it's wide open.

Leo: That's why I use VPN now because of you and learning this.

Steve: Yes. Well, a VPN is the only way to solve this problem because a good, properly implemented VPN will completely defeat, not only passive sniffing, but also any man-in-the-middle attack because the problem is - and we're going to be talking about this, I think next week, where we're going to finally start talking about what is all of this encryption, how does encryption work and what is authentication and signatures and signature authority or certificate authorities and all that, you know, certificates and signing and all that. We're going to lay all that foundation.

But anytime you don't have authentication, anytime that somebody can come along and pretend to be anyone else participating in the conversation, you've got trouble. And Windows Remote Desktop Protocol doesn't have strong authentication. They've got some, obviously not very well working because there's free software that is able to just crack the conversation and log all your keystrokes. So, you know, Windows doesn't provide substantially and significantly strong authentication, so that the entire dialogue can be monitored, unfortunately. But a VPN that is designed right does.

Leo: There's no man-in-the-middle attack for VPNs or SSH or...

Steve: Well, what's interesting is that users can be exploited in this. And we'll be talking about security certificates and verifying security certificates when we're talking about state-of-the-art encryption because what can happen is that, once a connection is being filtered with a man in the middle, if you go, for example, to PayPal, and you try to establish a secure connection, there is no way for a man in the middle to truly spoof the PayPal certificate. That is, there is no way for them to not raise an alert on your browser. But many users will incorrectly respond to a fraudulent certificate, not recognizing that it is such, and give it permission to connect.

Leo: And then...

Steve: When that happens - yes.

Leo: Then they got you.

Steve: When that happens, you're in deep trouble.

Leo: Right. As long as they can continue to simulate the PayPal experience. I guess once you've entered the password and log-in, they don't need to anymore. They could say, well, connection broken.

Steve: Yeah, thanks very much, see you later. So it is really necessary not to give your browser permission to connect when you shouldn't do so.

Leo: Right. Now I have a question for you, an ethical issue. We've talked about Cain & Abel and Ettercap. Should I put a link to Cain & Abel and Ettercap in the show notes? Or not?

Steve: I have them all over my page, Leo. I think there is no reason not to. Anybody, I mean, I have...

Leo: Can you use Google? You can find it.

Steve: Of course. You just put in "ARP spoofing."

Leo: If you can't use Google, then forget it, you'll never be a hacker.

Steve: You put in "ARP spoofing" or "ARP poisoning" or "ARP cache," and all of this stuff comes rolling out. I have a really great page, which is our show notes page for this episode. I put it together actually in early December, and I've been waiting till now to really talk about it. It's GRC.com/nat/arp.htm. So /nat/arp.htm. We've got a link to it on our page. You should link to it, too, Leo.

Leo: I will. I'll put the link in show notes.

Steve: Basically, I explain all this. I have some nice diagrams that show how this works. And because I - the reason I put these links to these tools on the page is it's important for people to recognize, you know, we're not making this up. You know, I was reading just from the boasting that these tools offer. And so I think it's important for people to recognize how very prevalent this issue is, and why connecting in a hotel or in a hotspot there is - there's just no security, due to the fact that Ethernet was designed back in an era where it wasn't even on the radar. No one even thought about it.

Leo: Right, right. And it works great, but there are some issues, and now you know what they are, and you know how to handle it.

Steve: Right.

Leo: Because we've talked about VPNs in great detail.

Steve: Yes. And in fact I am at work on the VPN notes. I just don't know when I'm going to have them done. I'm doing a really thorough job, and I'm working with a neat German guy who's making some changes to some Ethernet bridging, some free Ethernet bridging technology that's going to make the configuration of OpenVPN even easier. So we're doing a little software development here in the process.

Leo: Very cool. Well, all of this information is, as Steve said, in his show notes, and I'll have a link on our show notes at TWiT.tv to that and to everything you need to know.

Congratulations, by the way, Steve. We just got stats back from America Online, and we crossed kind of an important threshold, the 100,000 listener...

Steve: Yup, I saw that.

Leo: Yeah, that's really exciting.

Steve: Very cool.

Leo: Security Now! 25 in three weeks had 100,000 listeners. Generally the way it works is we get about 60, 65,000 in the first week, and it goes half, and half again. So it takes us about three weeks to get 100,000. But what's interesting, particularly about Security Now!, is even the first episodes are still being downloaded in the thousands every week.

Steve: Yeah.

Leo: So the number climbs. And I imagine each of these episodes will be, in the long run, listened to by several hundred thousand people because they, you know, they're valuable. They form together, as a whole, a complete course in Internet security.

Steve: Right.

Leo: GRC.com/securitynow.htm. That's the place to go where Steve has everything you'd want to know. Links to these individual show notes, but also transcripts thanks to Elaine, 16K versions for the bandwidth-impaired, and lots more, of course, security help. He's been doing this for years, helping people protect their systems with ShieldsUP! and Shoot The Messenger and DCOMbobulator and UnPlug n' Pray and so many other great utilities. And of course the king of all utilities, the ultimate disk maintenance and recovery tool, I tell everybody to get it, SpinRite. All at GRC.com. Are we going to have a birthday cake or something for 100,000? I think we have to do something to celebrate. It's great.

Steve: It's just - it's really good.

Leo: Yeah, it's just been fantastic. And I'm not surprised. This is something everybody wants to know and needs to know. And I know sometimes it's a lot of work, but that's where the transcripts come in handy.

Steve: Well, and, you know, not to break your flow here. But, you know, there will be people who will write and say, oh, my God, you know, so, yes, it's possible for ARP cache poisoning to happen, and man-in-the-middle attacks, but how likely is it? And again, you know, who knows? We just want people to understand what the reality of their connections are, what the reality of the security is, and let them decide what kind of measures they want to take.

Leo: Well, I think it's so easy to do. And, you know, there are a lot of tourists who would do this just for fun, like the woman you quoted. But it's so easy to do, I think it's only prudent, especially when you're using a hotel or a WiFi, a public WiFi connection, it's just only prudent.

Steve: I'm not plugging my laptop into a hotel without some protection. Nope.

Leo: No, no. And I've started using HotSpotVPN all the time. And I just, you know, it just makes me feel better. And I think it's just the prudent thing to do. Just the right thing to do.

I want to thank the great folks at AOL Radio who always are so generous. And we really appreciate it now that we've crashed others' servers. We really know what we're getting from AOL Radio. They broadcast us on their podcast channel and, of course, provide us with the bandwidth so we can bring this to you. AOLmusic.com/podcasting for more information about that or to subscribe to, I think, really the best Internet radio out there.

Steve, we'll come back next week, we'll talk about authentication.

Steve: Very cool.

Leo: That'll be fun.

Steve: Yeah.

Leo: Have a wonderful week.

Steve: Talk to you then.

Leo: Okay. For Steve Gibson, I'm Leo Laporte. Thanks for joining us on Security Now!.

Copyright (c) 2006 by Steve Gibson and Leo Laporte. SOME RIGHTS RESERVED

This work is licensed for the good of the Internet Community under the Creative Commons License v2.5. See the following Web page for details: http://creativecommons.org/licenses/by-nc-sa/2.5/