SECURITY NOW!

Transcript of Episode #198

Listener Feedback #67

Description: Steve and Leo discuss the week's major security events and discuss questions and comments from listeners of previous episodes. They tie up loose ends, explore a wide range of topics that are too small to fill their own episode, clarify any confusion from previous installments, and present real world 'application notes' for any of the security technologies and issues we have previously discussed.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-198.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-198-lq.mp3

INTRO: Netcasts you love, from people you trust. This is TWiT.

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

This is Security Now! with Steve Gibson, Episode 198 for May 28, 2009: Listener Feedback #67. This show is brought to you by listeners like you and your contributions. We couldn't do it without you. Thanks so much.

It's time for Security Now!, the show that covers all things secure and private. And Steve Gibson is here...

Steve Gibson: Except your parts. We're not covering your private parts.

Leo: Yes. He's the general of private.

Steve: Just your private data.

Leo: From GRC.com, Gibson Research Corporation, and creator of SpinRite. Hi, Steve. How are you?

Steve: Hey, Leo. We are approaching our double golden anniversary. We're at 198. And in two weeks obviously we'll be at 200. So, and then of course we have a short eight-week jump, and that's our four-year anniversary.

Leo: That's what blows me away. It's not the number of shows, it's the four years of continuous production that blows me away. That's incredible, Steve. Well done.

Steve: Well, we've both been up for it, and we've never missed an episode. So...

Leo: Well, that's thanks to you because I've missed episodes on other shows, quite a few. So that's really cool. Well done. Bravo.

Steve: And I do, as I'm running through the mailbag from people who go to GRC.com/feedback and enter their questions into the little web form there, so many people really appreciate the fact that they can count on an episode of Security Now! every week. So I'm...

Leo: Consistency is really everything in broadcasting. I think, you know.

Steve: Yeah.

Leo: But easier said than done in many cases. So I'm really appreciative...

Steve: Unfortunately we have a fertile topic, or I guess it's a mixed blessing that we have a fertile topic. There's always all kinds of things going on...

Leo: Ain't that the truth.

Steve: ...with security, yeah, and lots of technology to talk about, too, so.

Leo: Well, this week on Security Now! it's Steve's questions - I mean, your questions, Steve's answers, Episode 198. So in just a bit we're going to get to those questions and answers. We also have security news, errata and addenda.

Steve: And errata, yup.

Leo: Steverino, let's see, I guess we should start with the news. Is there anything going on?

Steve: We got news. We got some news.

Leo: Got some news.

Steve: Yeah. Good news and bad news. Well, actually kind of all bad news. Well, no. There is some good news. Microsoft just released Service Pack 2 for Vista. So the good news there is that everyone who's setting up a new Vista system until now has needed to first install Service Pack 1, and then do the Windows Update and stand back, maybe go on a short vacation while all the individual updates from Service Pack 1, which was March of '08 was when SP1 came out. So more than a year of security fixes and changes and patches and glitches and so forth. Now, if you just - you do need SP1 installed as a prerequisite for SP2. So it's no longer the case, remember that it used to be that you could just get the most recent service pack. In this case Service Pack 2 only contains all the fixes since SP1, rolled up into a single deliverable. So if you are now setting up a new Vista machine, you need SP1 and SP2, and you have to install SP1 before SP2. But just those two actions brings you current as of today, as opposed to needing to go through all the incrementals. So that's good news.

And for what it's worth, anybody using Server 2008, this is a hybrid service pack which is also Service Pack 2 for Server 2008. Server 2008 came with SP1 built in, so there's no need for an SP1 for it, followed by SP2. Just Server 2008 followed by this hybrid service pack, SP2 for Vista and 2008, and that brings you current there.

The other sort of news, it actually was Greg my tech support guy who first noticed this, and that is that many people, as we have discussed, have had a problem with Windows XP Service Pack 3. I'm still not running it on my main system.

Leo: Really.

Steve: Because it hurt me when I...

Leo: It's been a year.

Steve: I know. And that's the point, is the blocker tool expired on May 19. And you'll remember that Microsoft provides the Service Pack Blocker Tool which will, for a period of a year, allow you to keep their Windows Update from trying to install a service pack on your system. And you can, however, tell it do not offer this to me again. And I - so what happened was, after the blocker tool expired on my own main system, it's like, wait a minute, now I see what Greg's talking about. It's for the first time since a year ago it's saying, hey, go for Service Pack 3. And it's like, no, I'm still reluctant to do that. I mean, they haven't changed Service Pack 3. And it messed up things. So I'm finding I'm cautiously putting it on additional systems that I have, and I'm not seeing problems. But on systems where I know that it seems to be unhappy I've not moved forward. So I guess I ought to make an image of this system and try it again. It'd be nice not to fall too far behind.

Leo: As I remember, the issue with Service Pack 3 was driver incompatibilities that they - well, maybe that's Vista Service Pack 1 that they were patching so that you could have - you might have - in other words, you might have had a fix in the interim. Did you do no patches after SP3, or no?

Steve: Oh, no, no. I'm keeping - see, that's the nice thing, is Microsoft allows you not to install SP3, to opt out of that, yet will give you all the other incremental things continuing forward. So I'm frankly a little cloudy on how this all fits together. That is, it seemed like when you were installing SP3, all you were doing was catching up to all the incremental things between SP2 and SP3, which I was doing all along.

Leo: Right, right.

Steve: I was staying current. So it's like, okay, why do I need Service Pack 3 if I've been doing the incrementals all along? It's certainly the case that on setting up a new system, just like we were discussing with Vista, where you would - it's so much nicer to just be able to do SP1 for Vista and SP2 for Vista and then be current. So I certainly think that installing a brand new system - and I have, by the way, for example, when I was setting up my little tablet, this new tablet that I'm using at Starbucks. I'm an MSDN user, the Developer Network, so I have access to these builds of Windows as part of the $2,500 I pay every year to Microsoft for the privilege. So I installed brand new from scratch XP, and then I think I put in Service Pack 2 and then Service Pack 3, because I think even with XP you need Service Pack 2 at least before you can install XP's Service Pack 3. So I did that, and I have no problems. And I manually put in all the drivers myself because this thing came with Vista. And it's like, okay, well, I'm not ready to do that yet. So I'm really happy with XP SP3 on that machine.

Leo: Good, good.

Steve: So it makes sense when you're setting up a new system to do that. But for whatever reason, even though this main machine I'm sitting in front of now while I do the podcast with you, it's a relatively recent install and setup of XP. Still it didn't - it got a little funky when I put in Service Pack 3.

Leo: Really, wow.

Steve: So I was able to back out of it, yeah. And actually Greg, again, my tech support guy has had several systems where, I mean, this is the reason he was really concerned. He said, hey, what do I do now? He said, I mean, there are systems I absolutely know collapse with Service Pack 3 on them. So, yuck.

Leo: So you're going to have to get it, though. No way around it now.

Steve: Yeah. Now, Macintosh security.

Leo: Uh-oh.

Steve: I don't know if you've seen this. But there is a bad known vulnerability in Java which virtually the whole industry has fixed except Apple.

Leo: Really.

Steve: Windows has fixed it. All the Linux and various UNIX builds have fixed it. Apple has not. It's been known since December of '08, long since been patched. One of the security researchers, sort of a gray hat guy, decided, you know, this is dumb that Apple has still not fixed this.

Leo: Oh, so he released...

Steve: Yes.

Leo: ...an exploit.

Steve: So he released a proof-of-concept...

Leo: Oh, boy.

Steve: ...which has been decompiled, and the source is now available.

Leo: Oh, I wish they wouldn't do this.

Steve: It is being exploited in the wild. The security community is telling people, Mac OS X people, that they should disable the Java virtual machine support in Safari. The same exploit is functional under Firefox. And if you go to a malicious web page, that is, that runs a Java applet, it can take over your machine. Proof-of-concept code exists. It is truly being exploited in the wild. So, I mean, I guess certainly Apple will now fix it. I mean, there's no way that...

Leo: Does Apple use Sun's Java, or do they have their own Java? I though they used Sun's Java.

Steve: No, Sun's, yes, and it was Sun who warned in December of '08 about the flaw, made the fix available.

Leo: I wonder why Apple hasn't pushed it, since it's available from Sun?

Steve: It's a good question. I don't know if they missed it, or if they have got some reason. I haven't been able to track that down. But I do know that certainly now the pressure is turned up.

Leo: Yeah, no kidding. It's odd that they didn't fix it in the big, you know, that 500...

Steve: Oh, the mega patch.

Leo: ...megabit patch.

Steve: The 13,000 files.

Leo: And that just came out a week or two ago. I don't understand why they didn't...

Steve: Yeah. And in fact that's what this guy was waiting for. It's like, okay, let's see if that does it. And they didn't. So I think we can - I imagine we'll see something shortly that Apple will say, okay, I guess for whatever reason we need to do this.

Leo: Yeah.

Steve: So, and there's a lot of dialogue in the community saying, well, Apple is arguably becoming a larger target as they're succeeding more, as they're moving more into the mainstream. I mean, it's no longer the case that it's such a small segment that people are ignoring it.

Leo: Apparently Apple does their own Java. They're using their own code. So that's why they're behind.

Steve: So they're not using - oh.

Leo: They're not using Sun's code, yeah. Otherwise they would have just shipped Sun's code and had done with it, I imagine.

Steve: Okay. Well, and then it's odd, if it's really their own code, that they've got the same problem that...

Leo: Well, it's probably a reference, there's a reference implementation that they modified.

Steve: That they followed.

Leo: Yeah, they followed, yeah.

Steve: Okay.

Leo: A bad reference.

Steve: Last little bit of security news is that Adobe, under the increasing pressure to get their act together with all the problems they've had in PDF format interpretation in Reader and in Acrobat, has announced what I think is sort of a mixed blessing. They said, okay, we're going to do regularly quarterly updates, synchronized with Microsoft's second Tuesday of the month update.

Leo: Hmm.

Steve: But they're not doing it monthly. So they're only going to do every third one.

Leo: Quarterly.

Steve: And it's like, well, okay. I'm not sure that that's often enough.

Leo: Right.

Steve: Does that mean that they're not going to do one mid-quarter? I mean, Microsoft does them mid-month when something comes up that's bad enough.

Leo: People are so reluctant to admit there's something wrong with their code and patch it.

Steve: Yeah.

Leo: Or maybe it's just the issue of testing it and the concern that companies have about implementing this stuff and...

Steve: Well, but Adobe's no small group.

Leo: Gosh, yeah.

Steve: And you've got to think, okay, Microsoft is managing this for this whole nightmare called Windows.

Leo: Right.

Steve: I mean, all of Windows. And here Adobe's got a reader. Basically they've got a PDF...

Leo: Yeah, what's the tough...

Steve: They've got a PDF interpreter. So anyway, they've made a lot of noise recently saying, okay, we're going to really get serious. They've come up with their own acronym for, like, their security development life cycle, which is, you know, SDL is Microsoft's, but the equivalent for them. And we're going to be looking at old code that we've had in Reader to make sure. It's like, yeah, okay, good. Get going, folks, because PDF format, I mean, it has become the standard document transfer format for the 'Net. I mean, they won that. Microsoft has tried a few of their own that never - that just sort of sputtered and never happened because, again, Microsoft would like to own that, too. But Adobe owns that. And so it's like, okay, time to make this thing work solidly.

Leo: It's just odd. It's just very odd.

Steve: And get serious about security. I just think that quarterly doesn't sound to me, it's like, well, okay, we're going to do it, but we're only going to do it every three months.

Leo: But up to now they've been doing it as necessary, as needed; right?

Steve: As needed. Well, and what they've been doing is they've been doing the most recent version first, and then catching up backwards, then, like releasing, like sort of, for example, like the patch for 9 came out first, and then 8 and 7 came out a few weeks later. I mean, like, and almost deliberately like a week later, almost like they're saying, well, you're going to be vulnerable. We're going to leave you vulnerable for an extra week if you're not using 9.

Leo: Yeah. It's like, well, that's a bad - that's a really bad, bad thing to do.

Steve: It really is. That's a fundamentally flawed policy.

Leo: Well, it's driven by commercial concerns. I mean, that's not a nice thing to do.

Steve: Right, right. Well, it had to be done deliberately, too, because they announced a month beforehand that the 9 patch would be available on this date, and the 7 and 8 patches would be available one week later.

Leo: Oh, they knew.

Steve: So they planned to do this.

Leo: Just upgrade, you'll be patched.

Steve: Yeah, huh? Yeah, we'll patch a week - it'll be patched a week sooner.

Leo: Yeah, that's not nice.

Steve: Okay. So now, I mean, watch. I'm predicting here that this quarterly update will not stand. It's - there's just - this is wrong. And so we're going to see a flaw come out that's really bad, and they will do a mid-quarter...

Leo: Well, even Microsoft does that, though; right? Out-of-cycle patches?

Steve: But they're saying they're not going to.

Leo: They say they will never do that?

Steve: No, they're saying we're going to make them - we're going to update more often, and it's going to be quarterly.

Leo: And it's sufficient.

Steve: And it's going to be synchronized with Microsoft's second Tuesday.

Leo: That I don't understand. Is that so, like - I don't understand why you would synchronize it.

Steve: I don't know. I don't get anything about this. This is just, you know, what you ought to do is, especially for something like Reader, it's like, okay, it's not like the whole OS. It's a reader. And there's alternative readers. So, like, look. If there's a problem, fix it fast and give us an update. And we'd rather have three in a row, on successive days, if that's what it takes to keep our PDF reader secure. And we're certainly not going to wait around for months while a known problem is out there. That's just dumb. So anyway, that's Adobe's big news and announcement. So it doesn't make any sense to me. We'll see. We will be watching this closely, and we'll let our listeners know what happens.

Leo: Do you remember, I mean, this latest double flaw, but have there been a lot of flaws before that? I mean...

Steve: Oh, yeah, Adobe's been having lots of problems. I mean, like almost every month or two. It's like, oh, here's a PDF problem again.

Leo: Oh. Well, they need to patch it more often then, clearly.

Steve: Yeah, they do. In reading through my mailbag, a number of readers took exception to our sort of dismissive glibness, I guess I would phrase it that way, about Ada, the language.

Leo: Ada, oh, Ada, yeah, yeah, yeah, the programming language. Did it ever get implemented right? I mean, that was our - it did.

Steve: Yes. Yes, yes, yes. I've now been pounded on. They've enumerated all the compilers. There's open source compilers, closed source compilers, there's sideways compilers, there's PDA compilers. It is everywhere.

Leo: Ada is everywhere.

Steve: So I just want everyone to know, yes, my information was dated. Oh, and people are using it.

Leo: That's the question, is it being used in the...

Steve: There are colleges...

Leo: Well, okay, but is the DoD, which invented it, using it?

Steve: I don't know.

Leo: Yeah.

Steve: But colleges are training with it because it's such a great language. And I just, okay, fine. I give up. Yes, it's everywhere.

Leo: It's a wonderful language.

Steve: We love it. We love it. I meant to tell you, or to mention last week, and I forgot, Leo, so it's in my errata list here, I also loved, but this time really seriously, "Star Trek."

Leo: Yes. Oh, we didn't talk about it. Yeah, you were about to see it two weeks ago, and I didn't want to spoil it for you.

Steve: I literally, I hung up the phone, and I went, and actually I've seen it twice now. I thought it was spectacular. Everything I could ask for. I mean, even in my own sci-fi newsgroup, we have a sci-fi group at GRC, there are...

Leo: Oh, you do? I didn't know that. That's cool.

Steve: Oh, yeah. We just - gives us sort of a place to talk about it because, I mean, so many - there's such a cross-interest in, you know...

Leo: Right, sure.

Steve: ...technology and science and sci-fi.

Leo: Is that GRC.com/forums?

Steve: No, I don't have a web-based viewer. We have a traditional old NNTP, Network News Transfer Protocol, NNTP news server. So anybody who has got, for example, has Outlook or Thunderbird, you just need a real news reader. And so it's news.grc.com is the server. And when you subscribe to that there's a whole bunch of newsgroups. One is GRC.scifi, s-c-i-f-i.

Leo: Okay.

Steve: But anyway, so the point I was going to make was that there were a bunch of sort of purists who were saying, well, this wasn't at all like TOS, which of course is the acronym for The Original Series. And it's like, well, right. Okay. So maybe they thought they were going to get what - sort of what we heard. We heard for years that this was going to be where Kirk and Spock first meet and go forward. And so they were really wanting a formal, correct prequel to the original Star Trek series.

Leo: This was a reset, was what this was. This was more than a prequel.

Steve: This was.

Leo: This is starting over.

Steve: Yes. And I will not say too much because I don't want to be a spoiler. I avoid spoilers because I want to see normally these things for myself. So I can't explain what it was that happened that really releases this new - this movie and anything subsequent from needing to follow the same timeline that we all know so well who've been following Star Trek for all these years. So anyway, I just wanted to say I saw it. I saw also "Terminator Salvation" yesterday. And I really liked it, too.

Leo: Really, because it got crappy reviews.

Steve: I know. Something was missing. I can't quite put my finger on it. I mean, the James Cameron Terminator movies, the first two were just spectacular because, you know, that's what James Cameron...

Leo: Well, they had Arnold. Arnold is the key. You are not Arnold, you are not the Terminator.

Steve: And this one was good. I mean, it was not fantastic. But again, I'm like a thirsty man in the desert when it comes to sci-fi.

Leo: Exactly, you'll take anything, yeah, yeah.

Steve: So I'll take anything.

Leo: You'll take anything.

Steve: And this was fine. I thought this was better than No. 3. No. 3 was sort of, ugh, you know, really forgettable. But...

Leo: I think I've only seen 1 and 2, to be honest.

Steve: Yeah. And I watched all of the "Sarah Connor Chronicles" on...

Leo: I have no desire to see this new one.

Steve: ...Fox. And so I'm pretty much well steeped in Terminatorness. Maybe that was part of the problem, is that maybe it's a little overexposure of all things Terminator at this point. But I did want to mention that I saw it, and I liked it. But "Star Trek" was spectacular. I mean, I was literally, I was trembling and out of breath when...

Leo: Oh. Oh, Steve.

Steve: ..."Star Trek" ended the first time. I was. I'm a sucker for...

Leo: It was so good.

Steve: No, it was great.

Leo: It was so good. That's cute. That's really cute.

Steve: Well, I have, you know, Leo, I'm - I have to tell you, I think I've mentioned to you before that my buddies and I in high school made a Star Trek movie?

Leo: Oh, yeah, that's right.

Steve: It was on 8mm back in super...

Leo: Yeah, no, you come by it honestly, yeah.

Steve: And we did beaming out by standing under a chandelier in the dining room of one of our friends.

Leo: And that was before the movies. That was after just one, the one series.

Steve: Just the original series.

Leo: So you really get a lot of credit because that original series, as I think most people know by now, was kind of a flop and canceled early on. They didn't do a whole lot of episodes, and...

Steve: Well, I had dinner with Gene Roddenberry.

Leo: You did.

Steve: At COMDEX one year. And this was after, I mean, this was obviously years later. And it was funny because I met Stew Alsop down at the bar in the Las Vegas Hilton, and we chatted about the idea of me doing a column in InfoWorld. And which of course did happen. I did that for eight years. But Stew said, "Hey, what are you doing for dinner?" I said, "I don't have any plans tonight." And he says, "Well, why don't you come with me? I'm going to meet some guys, and we're going to go have some food." I said, "I'd be glad to."

Leo: Oh, man.

Steve: So I had no idea what the group was. But we go upstairs, I believe at the Las Vegas Hilton, and there's just, like, three regular random people sitting, kind of lounging in the hotel room, I guess waiting for Stew to show up. And I was tagging along. And so these people introduced themselves, round robin, and one of these guys says, "I'm Gene Roddenberry."

Leo: Geez, Louise.

Steve: Now, okay, now I have to explain, Leo, that we stood under the chandelier with the camera, the Super 8 film camera mounted on a tripod, running the - and everyone stood still. Then we stopped the camera, and everyone got out of the way, and then we started the camera again and filmed some time with nobody there.

Leo: Yeah. Smart, very smart.

Steve: Then after the film had been developed, we went back in, and on the emulsion side found the spot where...

Leo: Oh, my god.

Steve: ...everyone in a single frame disappeared, and then we began scratching...

Leo: You scratched it. Oh, my goodness.

Steve: Yes. We scratched the emulsion.

Leo: I'm so impressed.

Steve: On both sides, diminishing in each direction, so that when you played the film back, you saw everyone get under the chandelier, and then the appropriate sound effects, and then suddenly all of the centers of them began scratching until they were completed scratched out. And then it dissolved again, and they were gone. So it looked like they transported.

Leo: That's fantastic.

Steve: And we needed a...

Leo: Did you go [transporter sound]?

Steve: Oh, we had sound effects. In fact, we even had alien-sounding bad guys. Scott Wilson was - actually this was where most of this filming took place. And he was arguably the most over-the-top Trekkie in our group. His sister had a large collection of stuffed bunny rabbits.

Leo: Oh, you had Tribbles.

Steve: Well, no, no. These were the aliens. They were evil. They were the Bunnons.

Leo: The Bunnons [laughing].

Steve: The Bunnons. And so...

Leo: Oh, Steve.

Steve: Oh, it was bad. So...

Leo: How old were you? This was, like, 14 or 15?

Steve: Yeah, exactly.

Leo: Oh, that's so cute.

Steve: And so we used stop-frame animation to have the attack of the Bunnons. But when they of course had to hail the ship and be threatening, we needed them to sound alien.

Leo: Right.

Steve: So what we wanted them to say was, "We are the Bunnons. Surrender your ship or be destroyed."

Leo: I think we talked about this last week, and you did it backwards.

Steve: Well, yes. I came up with the idea, since this was actually that brown, actual magnetic recording tape that we no longer have...

Leo: Right, Mylar, yeah.

Steve: And this was reel-to-reel deck. So we recorded, "We are the Bunnons. Surrender your ship or be destroyed." And then we reversed the tape mechanically so that it would play backwards. And we listened carefully to the way it sounded and learned how to say it backwards, which happens to be yo-sha ba-di-dro, sna-na ba-na-ni, pa-shor-yor-nar-ros.

Leo: Of course it is.

Steve: Now, if you say it carefully, because things sound a certain way backwards, it's actually more like yo-sha ba-di-dro, sna-na-ba-na-ni, pa-shor-yor-nar-ros.

Leo: Hey, if you're listening to this, Tony, could you just take that and reverse it, and let's hear what that sounds like.

Steve: So...

Leo: Now, I have no idea what that sounded like, but I think it was probably pretty good.

Steve: Well, we recorded that, and then we reversed that. So that re-reversed the reversal.

Leo: Right.

Steve: And it really, it was - it came out "We are the Bunnons. Surrender your ship or be beshroyed." And we never quite got...

Leo: [Laughing] I can't do this in real-time, but I think Tony is going to do this on the show, and you're going to hear this.

Steve: Well, I have to tell you that because, I mean, I never obviously forgot this, nor what those phrases were. Remember when I was messing around with the SIP compressor?

Leo: Yes.

Steve: And I wrote a little app that would just - you could record something and then play it back.

Leo: Right.

Steve: Just for the hell of it, I had a different shift key you could hold down, and it would play the buffer backwards. And I practiced that again, and it was just the way I remembered it.

Leo: So we can be pretty sure that that's actually going to sound like that when we...

Steve: It'll be something like that, yeah. But anyway, so yes. Serious, over-the-top Trekkies. And I've been such ever since.

Leo: Oh, that's great.

Steve: And I'm one of the...

Leo: You've earned your stripes.

Steve: ...people who think that the Next Generation was the ultimate series. That's the whole Jean-Luc group. I just think that...

Leo: I agree. I kind of agree. I think that was a really great group of people.

Steve: Well, and it was - this recent "Star Trek" movie was action. It was fun and action, and it was that sort of thing. The Next Generation was a far more cerebral show, which is why I liked it.

Leo: Yeah, I liked - yeah, me, too. I liked the strategy and the solving the problems and that kind of stuff.

Steve: And, you know, they'd meet in their little conference room and go around and give their opinions, and then Jean-Luc would declare what action they were going to take. And I just - anyway, so I sat next to Roddenberry at dinner. And I didn't want to be like the oh, you know, the annoying groupie. But I did find an opportunity to ask him what happened. And he said that what happened was they had a three-year run. There were three seasons of the original series. And...

Leo: Something like 77 episodes. I can't remember the exact number.

Steve: And after that it was canceled. And the way Gene explained it to me was that several years later they were developing the technology, the first developing technology for the TV demographic profiling, what's it called, where you're...

Leo: The Nielsen ratings?

Steve: The Nielsen ratings, yes. They were developing the original Nielsen technology, which factored demographics in for the first time. And that was the difference. Star Trek was canceled after the third season because the raw numbers didn't justify its continued production compared to other shows. But when years later the raw data was reprocessed using demographic technology, it turned out that it had the most perfect demographic profile for advertisers of any show that had ever been created. It was the yuppies, the young...

Leo: Until TechTV.

Steve: Yes. Young, upwardly mobile, married newlyweds who were buying strollers and cars and homes and, I mean...

Leo: Smart people.

Steve: It's exactly who advertisers wanted. There just has never been a better show. But they didn't know it at the time. And so they said, well, look at the numbers. I mean, look at the count.

Leo: TechTV was like that. We had the most highly educated and affluent audience on cable television except for the Golf Channel. But inevitably, advertisers are slow to respond to that. It's one of the reasons that TWiT succeeds is because the demographics...

Steve: I was just going to say, not our advertisers.

Leo: No, they get it, they get it.

Steve: Yup, exactly.

Leo: So, let's see. We've got some letters. You've got a letter, a SpinRite letter?

Steve: I just got a nice note from, not a listener, I think, this time, because he wrote to our support email, and his name is Bob Blaine. And he said, "Hi, I just wanted to thank you for your SpinRite software. I was a little surprised at the price and wasn't sure if the problem I was having was going to be fixed by it. But based on reviews and your money-back guarantee, I thought I'd take a chance. Your software was easy to use and actually fixed the problem I was having. I'm very impressed!" Exclamation point. He said, "The problem was that I had a hard drive failure on an XP machine that was not allowing the machine to come up into Windows. It was giving some obscure message that I eventually found out meant that the registry file was corrupt. Amazingly, SpinRite fixed that so that I was able to get into Windows and back up all of the data on the drive before I replaced it. Thanks again, Bob Blaine." So yes, SpinRite to the rescue. Not a surprise for our listeners, I'm sure, but...

Leo: I think it's worth emphasizing that money-back guarantee because I think people do, you know, a new hard drive is about the same price as SpinRite in some cases.

Steve: Yeah, in fact I was listening to your replay of a podcast...

Leo: Must have been The Tech Guy or...

Steve: No, it was the one that was just playing with Andy Ihnatko...

Leo: MacBreak Weekly, yeah.

Steve: MacBreak Weekly, just as we were getting ready to do this. And he made a comment about - you were talking about ripping DVDs. And he said, yeah, you know, you can get a terabyte drive now for 90 bucks.

Leo: Exactly.

Steve: And I'm thinking, yeah, and that's what SpinRite costs. And he said so, you know, there's really no need to burn all those. Just rip them all onto that terabyte drive. And I'm thinking, yes, please do. Because, please.

Leo: Why is that, Steve?

Steve: Good. Put your whole movie collection on there because I will have your money. When that $89 terabyte drive craps out on you...

Leo: We're buying - are you saying people should buy fancier drives, or just this is inevitable?

Steve: Put all the crown jewels, put everything you have on hard disk.

Leo: Well, don't throw away the DVDs. Keep them. But it really is true that, if there's data on there, it's worth more than 89 bucks. It's not a question of buying another drive, it's a question of getting that data back.

Steve: Yes. I mean, people, for a while people were saying, well, gee, Steve, $89, that's pretty steep. And I'd say, yes, I understand. And then they'd say, well, we can buy a new drive for that. Yes, but it doesn't - it's not all of the data that you've got. It's not everything that's been installed in your system before. It's not, I mean, what's your time worth to, like, recreate everything from scratch? And in some cases these are irreplaceable. These are people's entire photo libraries that have never been backed up, never put somewhere else.

Leo: Well, and the other issue is that you're not just talking about recovery, you're also talking about maintenance. And you probably have a lot of drives. Nowadays we, I mean, I have dozens of hard drives. I buy - I literally buy terabyte drives by the six-pack now, probably about once a month. And so SpinRite's really great on checking those drives before I put them in use and making sure they're in great shape and maintaining them.

Steve: Yes. I do read people who say, hey, I bought SpinRite, I run it, I'm waiting for a miracle. But maybe the fact that I'm running it on my drives every few months means that I'm never going to have a disaster that requires a miracle. And of course that's the optimal situation.

Leo: Right, I'm with you.

Steve: You want to keep the drive from getting that bad. But yes, we do sell it with a money-back guarantee. The problem is I can't give a demo because, once you run the demo...

Leo: You're done.

Steve: ...and it fixes problems, you're done.

Leo: Yeah, you're done.

Steve: It's like, uh, okay.

Leo: People often say I want to download a trial.

Steve: Yeah.

Leo: And you don't want to do the thing, and this really bugs me, the unerase programs, there are some out there, will say, oh, yeah, I can see all your data. 500 bucks, please. And that's even worse. That's like, c'mon.

Steve: The tease, yeah.

Leo: That's that tease, yeah. So, no, it's worth it. Money-back guarantee if it - and I think that's the best way to do it. That's basically, you're basically saying try it for free. If it doesn't work for you, I'll refund your money.

Steve: Absolutely. Absolutely. And, I mean, there may be people who've bought it and then asked for their money back anyway. It's like, well, if that's how you want to play the game, that's fine, too. I'm not going to, you know...

Leo: I bet that doesn't happen very often.

Steve: Not among our audience. Our audience is great. And they help make the podcast possible and make GRC possible.

Leo: That's right. We want to support Steve, that's for sure. I have in my hands 12 questions from 12 listeners good and true. All for you, Mr. Gibson, starting with No. 1, Alexandre in Quebec. He's a Quebecois. He says: You made me love you. Actually made me love Assembly language, he says. Assembly language. Hi, Steve. I'm a young 18-year-old guy from Quebec, and I just love your show, pretty much everything you made for the beautiful yet now somewhat dirty computing world. I'm also a SpinRite customer - yay - and I love it. Sorry if the English isn't very good. I'm a French guy. So I'm very - your English is better than mine. Don't worry, Alexandre. So I'm very interested in Assembly language programming for Windows. That is just awesome. I just love hearing that. And I would love to know what Assembler you use or would recommend to me. I would like to use the same one. Anyway, I know NASM, but I have become interested in FASM. I don't know much, but it looks very good. Have you tried it? And what do you use?

Steve: Okay. Again, I try to choose questions that are representative of many that we receive. And there's been an interesting surge of interest, I guess because I've been talking about it, among our listeners about Assembly language and Windows. And there have been some postings, I know, in the GRC newsgroups. And I run across questions like this many times in reading through listener email. My recommendation is to go check out MASM32.com, the website, www.masm32.com. It is a tremendous site. And I have always been programming Windows in Assembly language, and I was sort of on my own when I began, I don't know, 15 years ago or something. MASM32.com is a site that has a tremendous tool that's at v10 now, a complete IDE, an Integrated Development Environment, all the libraries and include files, lots of sample code, the source for everything. It is dedicated to programming Windows in Assembly language. There's also a forum that is available there with a whole bunch of enthusiasts that are all doing exactly this. There are all aspects of programming Windows in Assembly language. So there really is a complete, I don't want to call it really a subculture...

Leo: And this is free.

Steve: Yes, it's all free.

Leo: Wow.

Steve: Now, MASM is Microsoft's Assembler. That's what I use. That's what I've always used. I really like it because they've extended it, the syntax of it, just a little bit in order to make it more pleasant to use. For example, you can say .if eax = ecx, and then have a bunch of code, and then a .end if. And so you have a traditional if/end if enclosure...

Leo: Oh, that's neat. And that's done with macros?

Steve: Well, no. It's native to the language.

Leo: The Assembler knows this. Oh, that's neat.

Steve: Yes. It's built into the Assembler.

Leo: You're starting to get a little higher level now here.

Steve: And you can have an else, for example. So if something else. And then but the reason I love it is that it exactly assembles into the same code I would write.

Leo: So it's still very clean. There's not a lot of cruft introduced as there would be in a compiler.

Steve: Well, there's zero, zero overhead.

Leo: Yeah.

Steve: Yeah. So I would have to write cmp eax, ecx; and then jne, which is to jump not equal, then to something down below. And I'd have to invent some label for it to jump to, which clutters things up. Visually it's easier to see if there's, like, an if/end if, and then you indent the code that is inside. So there's that. They even have looping constructs, repeat until or while. I mean, so you've got those nice flow control things that are the standard structures from higher level languages that exactly assemble into the same thing you would write for yourself. And they have a really nice construction called "invoke," which is used for calling subroutines, including the Windows API. So I can say, for example, invoke space create file, followed by a list of arguments. And the way Windows works is when you call into the Windows API, it expects those arguments to be on the stack in reverse order.

Leo: Right. So it actually puts them on the stack and everything.

Steve: Yes.

Leo: Oh, that's nice.

Steve: It'll convert the lengths to something else. You're able to say address of a - if Windows wants a pointer to something, you could say addr and then the value. So my point is that, without any overhead, you're still programming in Assembly language at bare metal; yet it's a really pleasant environment. And this MASM32.com website has the core - it's at v10 that you can download, that installs basically a complete working development environment on your system, and help files, all the libraries, include files. Basically it's a turnkey, get you up and going. And so I recommend it without reservation.

Leo: When I did Assembly programming, and I used to do it on a 68000, which was a much cleaner, easier processor, I think, than the X86.

Steve: Yeah, very nice instruction set.

Leo: Much nicer.

Steve: Very orthogonal, meaning it could do most things in most instructions.

Leo: And no memory segmentation or any of that stuff.

Steve: Right.

Leo: But I used macros a lot to make it look more English like. Do you use - in fact, I noticed a lot of Assembly language programmers would build large macro libraries to kind of simplify things.

Steve: Yeah, I do that, too. For example, I have a macro called "zero," z-e-r-o. And so I'll say zero eax. And all that's doing is...

Leo: Just a move.

Steve: It's an XOR.

Leo: Or an XOR, oh, you're clever.

Steve: Yeah. You could use a move, you know, move eax zero.

Leo: Is XOR more efficient?

Steve: Yes. XOR is smaller. It does modify the flags, that is, the condition code. But sometimes you want that. But the point is, if I wrote XOR eax eax, I know that that's going to zero it because if you XOR something with itself, the ones cancel out; the zeroes were never on. And so it ends up being zero. But as I'm reading the code, if I say z-e-r-o, then I immediately know what I'm doing there.

Leo: Right.

Steve: And I've got a bunch of things. I have one, a favorite of mine, it's called roundiv, r-o-u-n-d-i-v, which is a fancy little bit of code with no - that doesn't involve any other registers. But it does a division, and then it compares the remainder to the dividend and sees if it's greater or less than half of that, in which case it adds one to the result, so it does a rounding division. And so it's just, it's very convenient. So, yeah, I've built up a macro library over time of all kinds of little tricks like that, that I use in my code.

And the other thing that I think really puts people off of Assembler is very often you'll see somebody's code which is just atrocious. It's running down the left-hand edge of the page, and it's just the stream of acronyms and weird arguments. And you look at it, and you think, what the hell is that? Just no way can I read that. But my code, I mean, I take some pride in the way it looks because I know eventually I'm probably going to have to come back and read it. And so I'm expressing this to myself and to the computer at the same time, as good coders do, and the result is something which is really pleasant. It's not cryptic and horrible.

Leo: I'm going to be - I've kind of volunteered to teach, to do a little programming tutorial for my kid's high school because they don't do anything like that right now.

Steve: Wow.

Leo: And I'm not an adept programmer, but I think I'm good enough to get them started. But I just - I was going to, you know, use Python. But I just had a thought, it might be a great idea to spend some weeks learning basic Assembler because what a great way to learn how the computer works.

Steve: Yeah.

Leo: You're working bare to the metal. And so it gives you an understanding of what registers are, moves, loads, conditionals, bits, all of these things are very, you know, conditionals, conditional bits and stuff are very useful to understand, I think, at a low level. As you then move into higher level languages, you kind of know what's going on.

Steve: I don't know if - it could be Mac based. But remember that that little PDP-8 emulator is a beautiful piece of work.

Leo: There you go.

Steve: On the Mac platform.

Leo: Every kid in the high school has a Mac laptop. They matriculate with a Mac laptop.

Steve: It's perfect then because the PDP-8, being that I only has eight op codes...

Leo: Very straightforward.

Steve: ...is a perfect little language to begin to, like, do little experiments with. And I have used that PDP-8 emulator now on my Mac and written some code with it. It just works beautifully.

Leo: I will be calling you.

Steve: Okay.

Leo: I'll be calling you for some fatherly advice. I think this is going to be great. This is going to be so much fun. Start them with a PDP-8. That is a great idea.

Steve: Yup. It's just a junior perfect little machine. And you're able to single step, I mean, it does, it disassembles the code, it single steps, you can see the contents of all the registers. It's a visible computer which is perfect for learning, for giving you a sense for how they really work underneath.

Leo: Oh, I'm so excited now [laughing]. Moving along, Question 2. Marsh Wildman in Sacramento, California says, what do you mean when you say "scripts," exactly? Dear Steve and Leo, I constantly hear you warning about the security risks presented by scripts. What kind of scripts are we talking about here? This web form I'm entering in this message into, for instance, doesn't this run on a script? I'm looking for a form such as this or an order form. There are many options out there - Java, ASP, Perl, XML. What is the best choice for performance and to give me and my users the safety we rightly expect and increasingly need? Thanks for any help. Love the show. SpinRite is on my birthday wish list. I love that. Scripts.

Steve: Well, that's a really good point. We've talked about scripts. I just sort of use it as a - I assume that it is a term that everybody understands. But I think Marsh had a really good question. That is, okay, what exactly is it that we talk about? Okay. So there's probably no really rigid formal definition. But in general a script means that something is being interpreted, that is, it's like a batch file is - you could say that that's a script because you're not compiling that into executable code, the actual code that the machine runs. Instead, an interpreter is reading the script step by step. Now, again, that definition even has kind of a problem because the original BASIC language was an interpreted language, so that it was an interpreter interpreting it. But you really don't think of BASIC as a scripting language. I mean, that's not the way it's been known. So...

Leo: Scripting languages are typically, when they're run - I'm going to try this, too. We're both going to fail on the perfect definition. But when they're run, they're often plain text, which then is run, as you say, and on the fly interpreted and executed. I'm thinking of JavaScript, Python, Perl. Sometimes, like BASIC and Python, they have intermediate steps of precompiled code. But they're still interpreted in real time, as you say, as opposed to something like C or Assembler, which is machine-translated ultimately into machine code, which is run, then - can compile once, run forever.

Steve: Right. And I guess the problem is there's a gray area where something you don't think of as a scripting language is interpreted. So it's not just the interpretedness of it that makes it a scripting language. And so I don't think there really is a rigid, clear definition. But the reason I'm upset with scripts, or scripting, is not the scriptingness itself. It's that, for example, in the case of JavaScript, that JavaScript, which is by the way a very nice language, I mean, it was developed by the guys at Netscape. It unfortunately really bears no connection to the Java language itself from Sun. I mean, they're not at all the same. So in that sense it's sort of a poor choice of naming terminology. But the Netscape guys sort of wanted to pick up some of the glamour of the Java language, so they called theirs JavaScript.

Leo: Yeah, no relation at all.

Steve: Right. What makes me uncomfortable is that browsers, web browsers interpret JavaScript. And that means, well, and for the purpose of executing code from a remote web server. So that just makes me really uncomfortable. That's the classic Gibsonian reaction. The idea that you're going to click on a link, you're going to receive a web page from some remote server who you don't have necessarily a trusting relationship with, and that page is going to come into your browser. Your browser is going to parse the HTML, find scripting commands in there, and then run code which you've just received. The idea, I mean, that's fundamentally a bad idea. I mean, I recognize...

Leo: That's really what we're talking about then here is client-side/browser-side programming, languages that run on the browser side.

Steve: Except that Active Server Pages, ASP, that's a scripting language that is server-side. So you can also - or Perl. You could say Perl is, I mean, in fact, Marsh listed Perl among his choices. Perl is...

Leo: That runs on the server as a CGI script as opposed...

Steve: Right.

Leo: It doesn't, you know, the browser doesn't know Perl.

Steve: Right. So mostly it means something, you know, in general it means that interpretation is going on. And I would say maybe the right description is to say that scripting is used in web processes. You know, you're scripting on the client; you're scripting on the server.

Leo: But generally server-side stuff doesn't seem to be anywhere near as dangerous. For one thing, it doesn't have access to the contents of your machine.

Steve: Although you also have, for example, UNIX shell scripts. And there's scripting, but this is not...

Leo: But they're only dangerous when they run on the hardware that's being compromised; right?

Steve: Correct. And I'm...

Leo: That's the issue, I think.

Steve: I'm just sort of - I was sort of like trying to say, well, so it's not just web things that are scripting.

Leo: Right, right.

Steve: It's interpreted. Mostly I think I would say scripts are interpreted.

Leo: And but the risky issue is the stuff that is running on your computer that you are receiving from a website, and that the website is getting to run on your hardware. That's the problem.

Steve: Yes. That's why I - exactly. That's the source of my continual concern about scripting is it's fundamentally very powerful, I mean, all of the Web 2.0 stuff, you know, any online forums which are accepting content, and the state-of-the-art fancy stuff, Facebook and MySpace and this next-generation technology. It relies on you - well, and all the Google stuff. Gmail is a perfect example, Google Mail.

Leo: Right, it's all in JavaScript.

Steve: And calendaring, all of that. That's all based on scripting. It's arguably doable not with scripting. So, for example, Marsh said, hey, this form at GRC - he's at GRC.com/feedback - is this scripting? No, because it's from me.

Leo: Intentionally, you wrote it, although almost everybody else uses a script. You wrote it without script.

Steve: Yes. Mine is all server...

Leo: It's just using HTML, HTML...

Steve: It's in Assembly language, Leo. It's in Assembler...

Leo: Running on your side.

Steve: On the server side, exactly. So it's just - it's a naked form which you submit, and then I take the form and do it all over on my side.

Leo: It's not unusual for people who have forms on their website to have some JavaScript around it validating input, things like that.

Steve: Right. Or in fact sometimes the actual email engine itself is on the page, and that's been used and exploited in the past. People could change the addresses that the script was using and essentially use it as a remailer back in the early spam days. So the exploits of scripting are legend. And, I mean, and not surprisingly from my standpoint because it's code. You know, you're accepting code from somewhere else and running it on your machine.

Leo: That's the real - that's the fundamental issue. And it could be Java, which is not a script, by the way. We just talked about a Java exploit on the Mac that - you wouldn't call that scripting.

Steve: Or my even bigger nemesis is ActiveX controls, where that's code that you've just accepted and your browser has run. Microsoft is getting smart finally, and the browser says, hey, something's trying to run ActiveX. Do you trust where it is that you've gone? And as we mentioned last week, Windows 7 is going to get better about telling you more about this thing that's trying to run so that you're able to make a more informed decision.

Leo: Yeah, I mean, I think really we shouldn't probably use the word "scripting" because that's not the issue. Really the issue is a website putting...

Steve: Code from somewhere else.

Leo: Code from somewhere else running on your machine. That's almost always where the security issues come in.

Steve: Right.

Leo: Because it has to run a - I can run a Perl script that runs on my server. It's hard, nay, virtually impossible for it to corrupt your machine without it getting some code on your machine that you run.

Steve: Right, exactly. And on the server side you're able to provide the same benefit. I would argue that most of what is being done client-side can be done server-side. It's not as sexy, it's not as interactive, it's not as fast. So there are...

Leo: But it's safer.

Steve: It's way safe.

Leo: So, and that's why you say use something like NoScript which turns off JavaScript because that's one of the common venues, or vectors.

Steve: Yes. I love the idea of giving the user the choice, so that their browser is not just running any code that it happens to stumble over. But when you get to somewhere that you care about, it's like, oh, look, this fill-in-the-address form doesn't seem to be working correct. Then you turn on scripting right now for that. And then, oh, look, now it knows what state my zip code is in and the other fancy things that scripting are able to do for you.

Leo: As long as we're doing definitions, let's do another one.

Steve: Okay.

Leo: Victor in Pretoria, South Africa writes. He says he's been listening and wonders what a socket is: Hi, Steve. Since you've been discussing network technologies recently I thought it might be a good idea to discuss sockets. We use the term all the time. But I don't actually know what they are and what they do. What's a socket?

Steve: Well, the term as far as I know originated at Berkeley with their implementation, their first implementation on UNIX of Internet-style networking. The idea is it's an abstraction to mean an endpoint of communication, and also sort of to mean what your program talks to when you want to communicate across the network. Perhaps a more common or familiar sort of related name is a handle. The idea, for example, in programming you call the operating system and say I want to create a file. And from that you get back this handle, and you then write to that handle sort of as an abstraction for the file. And you're actually writing to the file that you created using the handle sort of as your token.

Well, a socket is very similar, but in networking parlance. So you create a socket, and then you may give it an address on your machine which is sort of like creating, like naming it. You may then connect this socket to a socket on a remote machine. And then as you write to your socket, it is possible to read from the remote socket. So sockets are maybe connected in that sense. They don't have to be connected. There are so-called "connectionless sockets." But sort of from a definitional standpoint it's the way programmers think about the way they talk to each other over the Internet, first in UNIX, and then Microsoft adopted the sockets interface and model, modified it a little bit of course because they can't leave anything alone ever, and so theirs is not really compatible with the Berkeley standard sockets model. Later they've made them more compatible.

But so that's what it is. When we talk about sockets, we're talking about sort of the programming interface and also - that is, from the programmer's side, and also the communications endpoint from the networking side. So, for example, in the Internet case, a socket would have an IP address and a port number. And together that IP address and a port represent a unique reception point or transmission point in an operating system for it to communicate to the outside world.

Leo: I've struggled with this myself because I get asked it on the radio show.

Steve: Really, you have listeners that are...

Leo: What's a socket?

Steve: Wow.

Leo: You know what I get a lot is, "What's a port?"

Steve: Yes.

Leo: So a socket's not a port.

Steve: No.

Leo: They have kind of the same functionality, in a way; right?

Steve: I would say that the difference is a port is - when I think about it, I know exactly what a port is. A port is a number from 1 to 65535 that port-based protocols carry. I know, that didn't make it much simpler. But, for example, the IP protocol has IP addresses, but no ports. You need to then layer on top of that the TCP protocol or UDP where they have - they bring this notion of a port so that the packets that are UDP or TCP format say I was sent from this port, and I'm going to that port.

And so the beauty of that is you can have services listening for inbound packets carrying a certain port number. So really the only thing that is, is just a number. It's carrying a port number. But when it arrives at the server running an operating system, the server says, oh, this is aimed at port 80. It's coming in to port 80. Which means I'm going to give it to that process, the web server process running inside me. If it's coming in to port 110, oh, that's the POP3 protocol port, so I'm going to give it to the email service running in there. So really you could think of ports as like the final switching stage, like in a switchyard, where when the packets come in, it tells the operating system which of the various servers that those should be routed to at a single IP address, that is, in a single server.

Leo: So really a port and a socket aren't the same thing at all.

Steve: Right. The socket is more of the programming abstraction, the way programmers see the Internet. Ports are an abstraction that lives on the packet side, the way that packets get routed once they arrive at the proper IP.

Leo: But they both have to do with communication.

Steve: Yup, and they make the world go round.

Leo: Sockets make the world go round. Let's remember that. Could be a good song. All right, long one here from Shawn Poulson, Middletown, Delaware, about SSL: Hi, Steve. Another fine Security Now! podcast with a deep dive into a big technology we could all understand, or we should all understand. He's talking about SSL. Well done. I wanted to write in to comment about a remark made near the end of that podcast. You and Leo were suggesting that with the power of computing these days, SSL doesn't represent much overhead. So why not use it for everything and continuously offer more security?

Let me point out a couple of cons that were not discussed. And if you like, use them, if you think I'm on to something. First of all, the cons: Web browsers don't cache content over HTTPS. It turns off caching. So that means they have to download every image, HTML, JavaScript, every single time. Of course, browsers these days cache a lot, and that speeds up browsing considerably. Of course you shouldn't cache the secure content or it wouldn't be secure. In addition, caching proxies won't cache this content either. ISPs often employ transparent caching proxy devices that save their upstream bandwidth to the 'Net by caching what their users often access, like say the Google search page logo. So when you go to a Google page you're probably getting that logo, not from Google, but from your ISP.

So it's important to know what content is truly public and safe to disseminate to anyone listening in and which is confidential in any way. I don't think it's ideal to blanket all web content as suitable for HTTPS. To that regard your CryptoLink package might be a good solution. I've often griped to myself how insecure it is to fire up my laptop at a local Internet caf. Anyone could listen in on all my doings from a distance. That's why you would use a VPN. He makes a good point. It just would be inefficient. It's not that the computer can't decrypt it, it's just the waste of bandwidth.

Steve: Yeah, and, I mean, so yes, I wanted to share Shawn's notion because he's absolutely right that web browsers are prevented from caching secure content, which is what you normally want. I have seen the option in the configuration dialogue sometimes that gives browsers permission to cache secure content. You probably don't want to do that because you want to know that those pages which you are looking at, just you, having logged on to your banking site, for example, and where all of those pages are wrapped in the security of that communication, you want to know that, when you close the browser window, that stuff was never written to the disk. It isn't in - it's not sitting there able to be scrounged around by anybody else using your machine or any malware that may be on there. So it is the case that the browsers are smart about that in the case of SSL.

And he mentions caching proxies and how using a secure connection will bypass your ISP's proxy. That's, for example, what I do when people are entering the ShieldsUP! site. Many users are behind ISP proxies. So nonsecure connections are going through the ISP's proxy to the remote server. So if I didn't ask for a secure connection when you were using ShieldsUP!, I would see your IP as the ISP's proxy. And that...

Leo: That's not right, yeah.

Steve: You don't want me testing the proxy. You want me testing with ShieldsUP! your machine. So the entry point to ShieldsUP! gets a secure connection so that specifically to avoid intermediate proxies. So both of those things are the case. Where what we were talking about differs from these examples that Shawn was suggesting was more my sense that all connections should be secure.

Leo: Yes.

Steve: Not necessarily that the content that is being exchanged always needs to be secret. And so I would differentiate the secretness of the content from the security of the connection. So, for example, email connections should be secure, even if it's just your random old email when it gets there, something where you're not concerned about the security on arrival as arguably you are if, for example, you're doing banking, and you want your banking pages not to be written to the hard drive. So this sort of - this confusion is a function of all of this still just being immature technology. This sort of came upon us half baked and half thought out, and we're all using it and making the best of it. And how often have I said someday it's going to be better. We're going to be marching...

Leo: [Singing] Someday...

Steve: We're going to be marching slowly toward that day for quite a while, hopefully doing podcasts all the while.

Leo: And we will have a celebration when it's better.

Steve: So it's really not HTTPS that I was referring to, it's this notion that setting up SSL connections are easy and inexpensive with today's technology. So given all of the, I mean, all of the problems with insecure locations where you can be doing things, like in open WiFi hotspots, it'd be really nice if our protocols were just secure by default, and then had this extra layer of, oh, yes, obviously it's secure, but also please don't cache it because this is sensitive. And, you know, it's sensitive content, carried in a secure connection. Be nice if everything were carried in a secure connection.

Leo: I blame myself. I think I was the one who was saying, well, why don't we just use HTTPS all the time, for every page?

Steve: Okay.

Leo: And there's a reason why not.

Steve: Right, because it's certainly the case that caching really gives us a tremendous performance boost. But boy, have you noticed how big, how much space on a hard drive web browser caches start eating up?

Leo: Yeah, hundreds and hundreds of megabytes, yeah.

Steve: Well, yeah. I mean, if you empty your cache, sometimes that takes half an hour just to delete all that crap. I mean, all...

Leo: But to be honest, I mean, hard drives are big now. And it's not, I mean, even if you have a gigabyte of a cache, that's still a small percentage of your total hard drive space. We're just old-timers, and we go, oh, a hundred megabytes for your cache, that's outrageous. But there's plenty of space.

Steve: Although you do really wonder how much of that is getting reused.

Leo: Yeah, that's true, too. It's caching everything.

Steve: Obscure stuff that you're never going to be seeing again.

Leo: Well, and that raises another issue, which is you should know, if you're not on a secure page, that frequently content that you're looking at is saved.

Steve: Yup.

Leo: And that's how people can get in trouble sometimes.

Steve: Often, in fact.

Leo: Yes. Simon Iremonger - love the name - in England wonders about processor security flaws. We don't talk about that much.

Steve: Never have.

Leo: Hi, Steve/Leo. I know there's all manner of interpretation going on in modern processors creating microcode, predicting the flow of the program in advance, all manner of optimizations. In one form or another, this is using a relatively fixed logic in order to interpret a user-supplied program. Yeah, it's a von Neumann machine. So why do we not hear such things as "a flaw in the AMD64 microcode allows malformed SSE2 instructions to bypass ring privileges"? Why do we not hear of "flaw in Intel atom cache row buffer allows unprivileged process to read cached kernel memory"? Just what's going on? I don't believe processors don't have bugs. Clearly they could present security issues. Is this security by obscurity? Is the nature of the hardware logic engineering more secure by design? Do those elite teams of CPU designers simply not resemble typical application developers? Your comments are welcome. I think this would be an interesting discussion topic. Thanks for the great show. It is a good question.

Steve: It's a really great question. And I found myself having to - pausing this morning at Starbucks as I was going through the mailbag, assembling these, thinking, okay, that's a good question. There have been flaws in processors, like in processor microcode. There was a famous - remember the Pentium divide flaw, where I remember seeing it demonstrated back in...

Leo: That's exactly what I was going to bring up, yeah.

Steve: Yup, in the Lotus 1-2-3 days you could put some specific data into the spreadsheet, and the spreadsheet would give you the wrong answer. Which was a consequence of there being a bizarre little subtle bug that affected specific cases of a division instruction. And it's conceivable that that kind of thing could be leveraged into a, like, into a public exploit.

But I think the best way to explain why we don't see this so much is that the model for a processor is very much like the model that we were discussing recently of code for the shuttle, you know, the NASA shuttle, where it absolutely - or even the Rovers on Mars, that absolutely has to be right. I mean, it has to be right to a much greater degree of "has to" than regular consumer software, where they go, oh, sorry, here's an update, download this. I mean, this is a processor where, for example, in the case of Intel of AMD, they can afford to spend huge amounts of man centuries making sure this is right because they're then going to encapsulate it in a chip and start spitting out millions of these things. It has to be right. It's high profit.

It's also high damage if they mess this up. And they've got very sophisticated tools for making sure that, as they've gone from one generation of X86 to the next, that this thing still executes similarly. That is, you have to know that they've got test suites of code that they run on a new design, emulating the next-generation chipset, really, and initially it's failing. There's things they missed. And it's not until all of the so-called regression analysis passes that they let it out the door.

Now, we know that Microsoft does something similar, but we've seen that Microsoft, for example, just to choose someone, doesn't catch everything because the guys down at eEye Security in Southern California, they have found many Microsoft faults by doing what Microsoft should have done. They've got a lab full of machines running Windows, and they're just throwing junk at them. There are, like, all kinds of wacky packets and all kinds of junk. They're keeping a trace of everything that they do. And every so often they crash a machine. Well, we know that crashing a machine is the first stage in exploiting a machine. So when one of these machines crashes because some random junk packets were thrown at it, it's like, whoa, wait a minute, what exactly did we throw at this machine that caused it to die? Then they back up, and they go forward.

So chips are done in the same fashion. I just think the reason we don't have problems like we see so commonly with software is, I mean, not so much differing levels of complexity, because exactly as Simon suggests, modern contemporary processors are phenomenally, phenomenally complex with the out-of-order execution and instruction renaming and flow prediction, I mean, I've been reading in the last few months, sort of studying processor design a little bit, and it just makes your head spin how complex these things are. But they're correct because they can afford to make them correct, and the cost of a mistake would be so phenomenal that they just can't afford to make the mistake. So a lot of time and money goes into making sure that when we get the chip, it works exactly as the spec says.

Leo: That floating point error was very - billions, I would imagine, it cost Intel. I mean, huge.

Steve: And here we are talking about it today. I mean, it isn't even dead yet.

Leo: Yeah, yeah. I mean, so, yeah, they learned their lesson.

Steve: Yeah.

Leo: It's really an interesting question, though. I'm surprised we don't hear more exploits.

Steve: I think it's a function of having the right tools and the right methodology and understanding we can't - this one we can't mess up.

Leo: It's also simpler. I mean, it's not as big a - it's not Windows. There's not, I mean, how much microcode is there? It's not a huge amount.

Steve: It's - I'm reluctant to say that it's much simpler because I've looked at what's involved now with a contemporary processor. And, I mean, and it's just mind-boggling. It's just mind-boggling what's going on in there.

Leo: So props then to these guys at AMD and Intel, and all the other chip manufacturers who do such a good job of this.

Steve: They're right because it has to be.

Leo: I like that.

Steve: Just like the shuttle code.

Leo: It's kind of neat. It's neat when you see somebody doing a job above and beyond.

Steve: Yeah.

Leo: Doing such excellence. Talking again about JavaScript, we have a bunch of questions and comments about that. Matthew Srebinski of Essexville, Michigan is getting warnings. Warning, warning. This may be one of the most common questions I get on the radio show, if it's the question I think it is: Steve, I just listened to Security Now!, your netcast on SSL and TLS, 195. As soon as you mentioned the latest Acrobat Reader problem I disabled JavaScript on several of my work and home computers. The first PDF I opened after that gave me an error. It said this contains JavaScript, but I had JavaScript disabled. Since you said you'd never encountered a PDF with JavaScript in it, I thought you might want to know about it. The file was a topographical map downloaded from the U.S. Geological Survey. I'm not certain what they use JavaScript to do, but it's interesting to know that somebody other than black hats is using it. I'm an avid Security Now! listener, and I own a copy of SpinRite, which has fixed several problems for me. Keep up the good work, and let us know how well the Kindle DX works as a PDF reader.

Also Eric in San Jose, he's getting false JavaScript warning messages. He says: I disabled JavaScript in Adobe Acrobat Professional 8 as instructed. However, now when I receive and open attachments from my co-workers I get two - count them, two - popup messages to click through that say "This document contains JavaScript. Do you want to enable Javascripts from now on? This document may not behave correctly if they're disabled." So I clicked "no" twice and moved on. The files did behave themselves. The PDFs I received were created using the Print to PDF feature in Acrobat 5 or 7. I don't know why they'd contain JavaScript.

Steve: Okay. And these questions are representative of a bunch that I received. The problem is that - and this is just laziness, frankly, on Adobe's part. They're not used to running with JavaScript disabled.

Leo: And they get mad.

Steve: So, yes, so it doesn't - so the reader doesn't behave itself very well. There may be some lazy tag in the PDF that says it may have JavaScript, but it doesn't actually. And so that causes the popup. I don't know, and it would take days to weed into this and wade into it and look at the actual raw PDF code to determine what's going on. What I do know is that these are false positives. It's that they only occur because Adobe hasn't taken the time to prevent them from occurring. Because Adobe has JavaScript turned on all the time, as do all Acrobat and Reader users except those who listen to this podcast and who understand that, okay, whoops, we've got an unpatched vulnerability, let's turn off JavaScript.

So as I mentioned before, there are definitely PDFs that use and depend upon JavaScript. And I had said that I ran across one myself at the State of California website. They had something, it was a PDF form that sort of filled itself in and helped you when you were applying for a permanent mail-in balloting status. And you really did have to have JavaScript turned on. On the other hand, it was obvious that you did. But static documents like these guys are talking about don't have to have it. You can say no, and everything works just fine.

So maybe the commenting, you're able in Acrobat to, like, add annotations and things. Maybe that uses scripting. But if you're just printing to a PDF, chances are there's no scripting, and you just don't need it, even if this thing says, oh, wait, you know, you've got scripting turned off. It's like, yes, I know, thank you for confirming that. So I'm going to read this without any scripting and be just fine.

Leo: Hmm. Humph. Harrumph. Question 7, William listening in Canada is interested in the XP mode that Microsoft has announced in Windows 7: Hi, guys. Is XP mode in Windows 7 properly sandboxing Windows XP? This is a particular concern if XP mode can be used as a vector of attack on Windows 7 systems, or if malware gets into the XP virtual system. The reason this may be a concern is because of the integration technology that is included with XP mode where XP applications are able to be on the same desktop as Windows 7 applications. They're using Virtual PC for this, if that helps.

Steve: Yeah. And I wanted to post this question to sort of put this on the map because we've received it a number of times. People want to know if this is the same as, for example, a VMware-style enclosure. And I don't know. I'm reluctant to look at it too much until Windows 7 gets, like, to release candidate. I mean, I know we're at release candidate. I meant, you know, RTM, Release to Manufacturing stage. My concern is that I'd be surprised if it was sandboxing properly. That is, that would be too inconvenient for most users. They would expect things running in Windows 7's XP mode to have access to the hard drive, to be able to see their documents and the normal things that they see outside of the virtual machine. I don't know either way. But I just wanted to share the question and to let our listeners know I will - we will find out exactly what's going on with that because it's going to be an important feature of Windows 7. I'm skeptical, though, about its use as full security sandboxing, only because doing that really does circumscribe the environment for the user. So Microsoft may have done default things, like made the whole external hard drive, external to the VM, available in a transparent fashion. I wouldn't be surprised if Microsoft had done that, which would completely zero it from a standpoint of security and use as a sandbox.

Leo: Hmm. Here's a really interesting one. This is from Rick Slater in Carriere, Mississippi. I'm thinking MS is Mississippi. He's worried about Slashdot's article about the Kindle DX's kill bits. In fact, I'll put a link to the article in the show notes so you can see it [http://slashdot.org/article.pl?sid=09/05/14/1356253]. You can see that the new Kindle DX has a number of kill flags built into the hardware which allow Amazon to reduce its functionality when they want to. This was disturbing enough, in fact, so much so I canceled the order that I had placed for the new DX pending the time that this whole thing gets sorted out. I'd be interested in hearing what your take is on the subject during the next Security Now! netcast. Thanks for the great show. Haven't missed one since the beginning. Kill bits.

Steve: Well, first of all, it's not plural, from everything I could...

Leo: It's kill bit?

Steve: It's kill bit. And this is addressing the concern that either we've talked about, Leo, or I've heard you talk about, that is, the audio reader.

Leo: Oh, this is the text-to-speech stuff.

Steve: Yes. So apparently there are some Random House texts which have required of Amazon that they not be read-out-loudable on the Kindle.

Leo: Right.

Steve: And so all that this is, this whole Slashdot story is people spinning off sideways, as they so often do on Slashdot, and exaggerating what's going on. We don't know that there aren't other kill bits. But there don't have to be. There's only one that this whole issue is about, and that's a single bit which says don't allow this particular title to be read out loud. Amazon designed that into the original specification for their content when they added the text-to-speech. They thought, well, maybe there'll be some situations that come up where this book should not be read-out-loudable. And we've now run across some. So that's what this is.

Leo: Yeah, it's just it's an incendiary term, "kill bits." I think people get upset just because it's called a "kill bit."

Steve: Well, and there was some guy who did post on Slashdot who said this thing bit me. There are some books that I purchased which...

Leo: I wanted to read, I wanted to hear.

Steve: I bought them so that they could be read to me, and they won't - and he's like, and he's grumbling about I'm sure Amazon won't take them back. And I'm thinking, no, I'll bet they would. I'll bet you could say, hey, this doesn't work for me, remove these from my bookshelf and cancel them. I mean, I just bet you could get cooperation. So I think it's a little tempest in a teapot, myself.

Leo: This next one...

Steve: But be aware that not all books can be read to you.

Leo: Yeah. This next one comes from a friend of mine, Steve Vance, who's with the Golden Gate Computer Society in San Rafael, California. He's a great guy, see him every year when I speak to the club there. He wants to know about how to detect SSL proxying: Hi, Steve. You've mentioned that if I'm using a company computer, and they have installed a certificate on it, they can decode my HTTPS packets, look at them, then reencrypt them and send them on. Kind of a man in the middle almost. All right. So I want to go to my bank's website on my lunch hour, but I don't want anyone in my company to watch me do this. Is there some way to tell if they've installed the certificate and are doing it? It looks like my bank's website. The address bar is green and everything. If my company were doing this, would it somehow be obvious? How would I go about trying to detect if they're doing it? For extra credit, if they're doing this kind of proxying, is there any way I can thwart this? Would CryptoLink solve this problem?

Steve: Okay. That's a great question. And again, my discussion of this a couple weeks ago caused a whole bunch of concern, very much like Steve's, on the issue. What you can do is pretty simple. If you, for example, went to any secure site, and since I know my certificate you could go to the page I was mentioning earlier, the ShieldsUP! page. So you just go GRC.com and then choose ShieldsUP! from the main menu and go no further. That'll take you to sort of the entry page where we're saying hi, click this button to proceed, but stay there. You'll see then that you've got the padlock showing that you're on a secure page. You look in the address bar, it's https. I have not yet bellied up to the bar and purchased an EV, an Extended Validation certificate. I'm considering that when I next renew my certs because it would be a nice thing for GRC to have, to also have the green. I think that's - it'll be a good value for my money, although it is a lot of money.

So then you can, if you hover, at least in Firefox, hover your mouse over the little padlock, you'll see something that says VeriSign Trust Network. You can also double-click on that to open up the properties. What you want to do is you want to poke around in there. Various browsers have these in different places. But you want to look at the so-called chain of trust that we've talked about, the sort of a hierarchy of links for the certificate. And in the case of GRC's certificate, which I get from VeriSign, you'll see GRC.com, you'll see a VeriSign intermediate, and then that trusted root, the certificate authority, and nothing else. That's the key.

If you did this in a corporate, within a corporate region where they were proxying your SSL, you would see that you had a secure connection. But when you looked at the certificate, it would show, for example, GRC.com and then link to some non-trusted certificate authority that had been planted in your browser and that had been used to generate a certificate on the fly. It would not link back to the real certificate from GRC.

So to be sure, you could go to any website, doesn't have to be GRC, but mine's always going to be there, and do it from within a corporation, see what that chain looks like. You can probably tell for sure just there. But if you went outside that corporate environment, for example, you went home and did the same thing, you should have exactly the same chain no matter how or where you connect to the secure website like that page at GRC. So if they're different, then you know for sure that some funny business is going on.

Leo: Excellent. Simple enough. Just check. And by the way, would CryptoLink help with this? Yes; right? Because then you have a tunnel, and they can't break through the tunnel at the work.

Steve: Yes. Now, okay. Good point. I forgot that last part of his question. The CryptoLink will excel at getting out of pretty much anywhere. For example, you'll be able to use a connection to port 110, which is the POP3 port, for example, and 80, and 443, and 25, which is probably going to be blocked anyway. But, I mean, in fact, the way CryptoLink will work is it will spray simultaneously a whole bunch of SYN packets out towards different destination ports in order to opportunistically find an opening out in order to reach the server that you're connecting to, which will be looking at - actually CryptoLink has the ability to look at all ports, all there are, all 65535, because the packets are self-authenticating. So it's able to detect incoming packets that are from its matching client.

So again, I'm designing this so that it just works, no matter what situation you're in. But it's still conceivable that a corporation could have its border so locked down that nothing that's not proxied can get out. In which case nothing can get out. I mean, I would be surprised if that were the case. But it's conceivable. You know, I mean, CryptoLink will be able to use both UDP and TCP, blah blah blah, I mean, if there's a way to get out, I'm going to get out. But it's conceivable that if they're proxying port 80 and proxying port 443 for HTTP and HTTPS, and like nothing else is allowed, then you would know, for example, you couldn't be getting POP mail, you couldn't be getting email from some other ISP, for example. That would mean that port 110 was open, in which case CryptoLink would go, ah-ha, and it would find a way out.

It's conceivable that a corporate network could be so locked down that nothing but web surfing works, and all of that is proxied, in which case nothing would be able to get out of there. But seems unlikely. I hope that Steve will, if he's in such an environment, let me know while we're beta testing CryptoLink if it's able to get out because it'd be fun to know.

Leo: Yup. Ren Zhi Zhang in Auckland, New Zealand wants to verify the security of a portable router solution: Hi, Steve. I like the idea you mentioned in Episode 196, carrying around the portable router - like AnalogX does - and using it as a hardware firewall when I connect to someone else's LAN, say at a hotel, to protect against other machines on the LAN. I'd like to be cautious and extra secure by testing this setup first. However, I'm not aware of anything remotely similar to ShieldsUP! that I can run from another machine on the LAN to confirm that there aren't any ports left open by my portable router. I ask the question because ShieldsUP! - correct me if I'm wrong - would only scan for open ports on the initial router connected to the Internet, not that router in between. Any suggestions?

Steve: I have a great suggestion, and that's a great question. We've never talked about local port scanning. I don't think we've ever talked about it. There's a neat security company a little bit south of me called Foundstone. They were independent for a long time. I think they were purchased by McAfee, but I'm not sure. But they're continuing to operate independently. And in fact one of their guys hangs out in the newsgroups. And he's an author of security-related software. Foundstone has a little, free, nice, very well designed, standalone, Windows-based scanner called SuperScan. So if you just put into Google Foundstone, F-o-u-n-d-s-t-o-n-e space SuperScan, it's the first link that comes up. And it's free. You can download it. They don't ask you any questions. You have to agree to their little license agreement, but they're not asking for email address and your name and all this other nonsense that sometimes it's not quite as free as it seems. It's nice and small. It's like 100 and some K in size. I think that Robin Keir may be its author. I know that they caution a little bit that some functionality was lost when Windows removed raw sockets.

Leo: Oops.

Steve: I mean, that's the consequence of removing raw sockets to tighten up the security and prevent abuse of that technology. But they're still able to perform a lot of scans. It's got unrestricted IP ranges. There's a whole bunch of features in it. And I heartily recommend SuperScan from Foundstone. It's at v4, which requires a Windows 2000 or NT or later machine. They do still leave v3 available because it would work on even earlier machines - 95, 98, and ME - if anyone is still using those. And so the idea would be that Ren would set up his travel router, and then a machine behind it, and then from another machine on the same LAN would use SuperScan to scan all the ports on that IP. And he's right that using ShieldsUP!, ShieldsUP! being out on the Internet, its whole intention is to be an Internet-facing security test. But in this case he specifically wants a LAN-based security test, so like an intra-LAN test. And SuperScan will do that for you. And I recommend it without hesitation.

Leo: Cool. SuperScan. And is it free?

Steve: Yeah.

Leo: Wow, that's nice.

Steve: Yeah. They have a whole bunch of great free security tools there at Foundstone.

Leo: Have to make a point of writing that one down. Darius in Port Moody, BC wonders about multi-site version control: Hi, Steve. I've been listening to you and Leo for the past couple of years. I've been listening to Security Now! since Episode 99. You often talk about the code that you write for your various projects. It sounds as though you use different computers to work on these projects depending on whether you're at GRC HQ or Starbucks. I was wondering what your solution is for secure source control. How do you keep your source code safe and well protected on the move? Do you use SVN or some sort of versioning tool?

Steve: You know, I don't. I found a great Windows utility. It's not free, and not even particularly inexpensive. And I'm drawing a blank right now. I was just looking for it in my tray because I always have it running down there.

Leo: Do you use Brief? What do you use as your editor?

Steve: I use Brief. Brief is my editor. But, and I use Jungle Disk and Amazon S3 as my repository.

Leo: Oh, interesting. So it's not just a backup for you. That's where you keep updated code.

Steve: Okay, I got - I remembered. It's FileBack PC.

Leo: FileBack, okay.

Steve: FileBack PC is a phenomenally powerful, general purpose backup utility. And it does versioning. So, for example, in the configuration dialogue you're able to say, I want to keep a maximum of 20 copies of my source code. I want no two to be closer together in time than an hour, no more than seven in a day, and no - and, I mean, you're able to granularly specify exactly how you want this thing to operate. And what I love about it is I'm also able to say, okay, I don't want any .err or .map or .exe, you know, you're able to give it like a long chain of, like, file pattern descriptions of whether you do want it or don't want it to back those up.

And so what I have is essentially, answering Darius's question, is by sort of gluing together a couple solutions, Amazon S3 is amazingly inexpensive, Jungle Disk is a one-time cost of I think it's $19, and FileBack PC allows me to be working at Starbucks, to be saving the code. It's transparently all copied to Amazon. It's completely secure because we know that FileBack - we know that Jungle Disk does encryption on the client so that nothing that's ever at Amazon can be decrypted. I mean, I'm not trusting my source code to anyone. And so, and I trust this setup because it is bulletproof. Then, when I'm back home, everything that I was doing at Starbucks has been kept with multiple versions.

And boy, I'll tell you, that does come in handy because there have been times when I've sort of gone off on a tangent and deleted a whole bunch of stuff that I didn't think I was going to need anymore, and then it's like, oh, shoot, I wish I had that. Well, you know, I still do. I'm able to reach back in time and grab a file that still has something that I got rid of prematurely or grab some snippets of code. So it ends up working really well. And I just like the transparency of it. And of course FileBack PC is a general purpose file backup solution. So I'm using it, for example, to spool content among many of my systems as I'm doing work. It's able to be - you can configure a huge list of different sort of tasks. And, for example, every time I hit the hibernate button I see it briefly pop up and shoot things off to Amazon because it senses that you're going to sleep, you're hibernating, you're shutting down. And it does, like, pre-end-of-session work. And all of this is really configurable with a lot of granularity. It's a very mature tool. So that's what I use, and I love it.

Leo: Going to have to get you on Git one of these days. That's a little more modern. But if it works for you, you know, you're still using Assembly, after all.

Steve: I'm still using a DOS box. And a DOS editor, a 16-bit editor that won't...

Leo: Yeah, that's right. I don't know if Git - I don't know if you could run Git in DOS, to be honest with you. That's actually a very interesting question. You might have to stick with FileBack.

Steve: I'm not worried about it, Leo.

Leo: David Greenberg in Nyack, New York has our last question of the day. He's got his Travel Router Tip of the Week, is what he's got: Steve, I'm an avid fan of Security Now! and have been since day one. In your most recent show you discussed a D-Link portable router as a neat solution for protection against malicious traffic from other machines on the same LAN. I'm actually using my Apple AirPort Express to do the same thing. You can put in your pocket. You can plug it in anywhere, use it in the hotel. Oh, look. He says: Without any affiliation with Apple specifically, I also want to recommend their AirPort Express travel router. It has a - you know, because I was going to go out and buy the D-Link. And then I went, I've got, like, two AirPort Expresses that I'm not using.

He says: It has a similar wall-wart form factor, complete with swing-out prongs, and can be configured to serve WiFi after being connected to a wired Ethernet connection. However, it can also be used to attach to an existing WiFi AP as a client, in which case it may be used in several interesting modes. In one mode it can be used to extend the range of an existing WiFi network. That's what I used to do with my AirPort Extreme and my AirPort Express. I used it in WDS mode. In another mode, its built-in USB port can be used to attach a printer, creating a print server. Finally, the unit has a built-in D/A converter and a 1/8" stereo audio jack, can receive streaming audio directly from iTunes or, via a third-party program called Airfoil from Rogue Amoeba, any other audio-generating piece of software. In my case I use it to stream music from my desktop and laptop PCs to my living room stereo. Then when I travel I reconfigure it as a WiFi AP, take it with me for that purpose. I guess you take speakers, too. It's really quite a versatile device, with one drawback. Rather than running a web server for easy configuration, you have to run a specific Apple utility on your machine to connect with and configure the unit. And that's why it's not so hot for a Windows user.

Steve: That's just what I was going to ask you, if there was a Windows solution.

Leo: I don't know. I don't think so. Thanks for such a high-quality program and for occasionally sharing the details of your own personal background and interests. We have several interests in common, and I'm sure we'd have a lot to talk about if we ever met someday. Let me tell you, having a glass of Cabernet Sauvignon with Steve Gibson is always a pleasure. I can speak from personal experience.

Steve: I miss those, Leo. That was fun.

Leo: We're due for that. I know, we haven't had that in ages.

Steve: I'll have to get up to your neck of the woods.

Leo: Please do. I have a bottle of Cab just waiting for you.

Steve: Anyway, I wanted to share David's posting. I know we have Mac listeners. And this little AirPort Express sounds like a spectacular little unit.

Leo: It's great, yeah.

Steve: Yeah, I love the idea that its little fangs swing out, and you plug it directly into the wall to get power, so in that way it's like the D-Link that has got it built in. And I think it's very cool, too, that if you're in a hotel you can plug it into the hotel's wired network, which is exactly what Mark Thompson does when he brings his little D-Link with him here, is then it's your wireless access point, secure I'm sure with WPA. So it's a router protecting you, as we were talking about, and then you're able to use your laptop anywhere in the hotel room without having to be wired down. So it sounds very cool.

Leo: It says - I'm looking at Apple's documentation. They say to set up AirPort Express using a Windows PC you can use iTunes.

Steve: Yay. Well, that's good.

Leo: Although, y ou know, I'd have to - it's really, I mean, it is designed for Macintosh hardware. I've never tried it. I'll have to - I'll tell you what. I'll report back next week. I'll try and figure out how I can configure it from a Windows machine. I'm just, I mean, the truth is, if you set it up kind of in generic DHCP mode, you don't have to configure it at all. You just plug the thing in and, you know, say hello.

Steve: Right. I was thinking similarly that I've got Macs, so I could use a Mac to get it configured, and then use it in Windows mode with no problem at all.

Leo: There is, I guess there's a Windows AirPort setup application.

Steve: Oh, good.

Leo: Yeah. So you just use their application. I guess there might be some advantages to having a dedicated application doing this to using the browser to do it.

Steve: I don't think so, except maybe just...

Leo: For security?

Steve: ...if it lowered the cost of the device. Because, you know, you'd have to have a browser in there and a lot of web page content and so forth. So maybe it's just simpler to do it that way. I did note, by the way, when I was doing a little bit of quick pre-podcast research about this, everybody's got a travel router now. You know, Linksys, Netgear, D-Link, Asus, Belkin, and Apple are the ones that I saw.

Leo: Right, right.

Steve: So it's like, whatever brand you like. I can say that the D-Link works that way with a little built-in adapter. And that's what you really want, I think, you want something small that you can just toss into your bag and not have lots of cordage and things dragging around. It sounds like the Apple Express travel router is exactly that, too.

Leo: Pretty small. I think that APC may even make a smaller one, designed just for laptops. There's stuff around. Steve, we've come to the end of our 12 wonderful questions from our 12 wonderful listeners, and we thank everybody.

Steve: And look, we made it in less than two hours.

Leo: Just.

Steve: It's a miracle.

Leo: Barely. Barely. I am downloading right now the PDP-8 emulator.

Steve: Yay. It's beautiful.

Leo: It's designed for older Macs. It's a Carbon application. You have to run it kind of in a - but it runs, seems to run fine. However...

Steve: Maybe you got the wrong one, Leo.

Leo: Oh, where do I get it from? I'll tell you where this came from. It came from Bernhard-Baehr.de.

Steve: Yeah, that's the guy. But it just runs on my Mac.

Leo: Well, it runs. It's not a - it's a Power PC. It does run on Intel. It will run up to Leopard. It is, you know, once you get it running it's a little cryptic. It's like, well, now what do I do? I'm going to have to ask you for help there.

Steve: Yeah, I was very, very impressed with it. It knows when you, like, drop some source code onto its icon in the - whatever you call that on the Mac, the tray or the...

Leo: Yeah. Well, you know what's kind of cool, it looks like it - there was an alternate disk image which I got, as well, that had, I mean, this is - it comes with Pascal.

Steve: Oh, yeah, it's got a full OS 8 operating system. So you're able to, like, mount the OS 8 drive and run the original - and that was like the final operating system for the PDP-8.

Leo: It's got a front console, front panel.

Steve: Oh, beautiful rendered console. You're able to see the contents of the registers. It simulates all the different I/O devices. And you can see what's in them.

Leo: Yeah, I mean, when you print something to the file, to the line printer, you can then save it to a disk.

Steve: Yeah, he did a really nice job.

Leo: So, but where's the interface for pro- oh, I guess there it is, there's the console. ASR33 console teletype [laughing].

Steve: Yeah. And in fact there's a - somewhere, I think he's got it in there, is a Towers of Hanoi - or maybe I typed it in. But I remember watching it like, you know, going [mimicking teletype]. And it's like, you know, you can have it run at 30, or at 10 characters per second, just like a real teletype did. And I remember, like, I think I might have typed in the FOCAL program, which was DEC's interpretive sort of response to Dartmouth BASIC back in the day. And it sat there [mimicking teletype], typing out the height of the...

Leo: Here it is, Towers of Hanoi, yeah.

Steve: Ah, good.

Leo: FOCAL-8. So now if I open this, it loads it in. And now how do I run it? Do I type "run"?

Steve: I don't remember.

Leo: [Laughing] Typing "run" did not do it. But I'll...

Steve: I would imagine you'll, I mean, I figured it out very easily. And after you get the hang of it, you're able to make it look pretty simple for people to use.

Leo: Yeah, and that's going to be the key for me is I'm going to take a bunch of high school kids and say, guess what, you now have a PDP-8. Let's...

Steve: A 12-bit computer with 4K of memory.

Leo: Let's program it, what do you say? Oh. I think this is great. What a great way to teach computer science, the kind of fundamentals, to start with the basics?

Steve: And Leo, when then you take them to higher level language, they will be so appreciative.

Leo: They'll be grateful.

Steve: Instead of just taking it for granted, they'll go, oh, thank god, we have variables now instead of memory addresses.

Leo: Yeah.

Steve: But again, I think - I'll bet you none of them ever forget that. And it's a valuable lesson.

Leo: A great way to learn, if you ask me. I mean, maybe it's because we're old-timers. But I just think that's a great way to learn.

Steve: No, given the reaction from our listeners to this stuff, the gray-haired episodes, there's a lot of appreciation for it.

Leo: I mean, there's a disassembler window. You can see the memory contents. You can see the stack. You can see the interrupt controller. I mean, you get to see how it's working.

Steve: Yup, it's the whole machine made visible.

Leo: I have a pointer here at address 200. That must be the code I loaded in. I wonder if I can just run it from there.

Steve: And that's the default starting point because the page 0 - and that's 200 octal, which is actually location 128.

Leo: Ah.

Steve: So the first page is a special page because remember that with the instruction format there aren't that many addressing bits. So you're able to only address the page you're in or page 0. So page 0 is sort of special because you can get to it from anywhere. Otherwise you're only able to address the page you're in.

Leo: So you keep page 0 blank and there for pointers or things that you need.

Steve: Exactly, sort of like your global variables that you want to be able to access as you wander through code space.

Leo: Oh, this is so cool.

Steve: It really is. It's just...

Leo: Do you have a manual somewhere online that you recommend for people, I mean, I'd like to learn more about this before I try to teach anybody how to use it.

Steve: Yeah.

Leo: I might have to get a PDP-8 manual.

Steve: There is a - there's a bunch of stuff...

Leo: There's some tutorials, it looks like, on the website.

Steve: ...built in. And there is a - I want to say Chicago or Indiana. There's a college that is using this and using the PDP-8 for their curriculum.

Leo: That's what I need.

Steve: And that professor has a whole bunch of really nice sort of laid out like, you know, here's the memory reference instructions for the PDP-8 and so forth.

Leo: That's what I need. I need to steal somebody's curricula.

Steve: Yeah.

Leo: Yeah, the Towers of Hanoi FOCAL source code. Is FOCAL like a language? Like BASIC?

Steve: Yes, FOCAL was DEC's sort of answer to Dartmouth BASIC. And so it's a funky language. But, I mean, it was very popular, and people were using it on...

Leo: It also comes with FORTRAN and Pascal-S. So you're in pretty good shape here. This is a great little development environment.

Steve: Yeah.

Leo: All you need is a Macintosh. All you have to do is buy a thousand dollar computer, and you can run it.

Steve: It'll slow you down to 10 characters per second [mimicking teletype].

Leo: It does, it warns you, it says "This is really slow. Be patient. Interpreted FOCAL programs running on a PDP-8 simulated by a slower or faster Macintosh are very slow." This is the old days, folks.

Steve: Yup.

Leo: Thank you, Steve Gibson. For more information, of course, Steve has his show notes online at GRC.com; also 16KB versions of the show for quick download; transcripts so you can read along. There's the forums there, the security forums. You can leave feedback at GRC.com/feedback. And of course don't forget there's some great stuff on there, free to download - ShieldsUP! to test your router, Wizmo, DCOMbobulator, a lot of free stuff. And the bread and butter of the whole operation, SpinRite. GRC.com. Steve, we'll see you next week on Security Now!. I've got some PDP-8 programming to do here.

Steve: Cool. Thanks, Leo.

Leo: See ya.