



SECURITY NOW!



Transcript of Episode #85

Intro to Web Code Injection

Description: Steve and Leo begin a three-episode series to discuss and examine web-based remote code injection exploits. Commonly known as 'Cross-Site Scripting' and 'SQL Injection,' these exploits are growing in popularity and strength as hackers discover increasingly clever ways to exploit subtle defects in next-generation web-based applications.

High quality (64 kbps) mp3 audio file URL: <http://media.GRC.com/sn/SN-085.mp3>

Quarter size (16 kbps) mp3 audio file URL: <http://media.GRC.com/sn/sn-085-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 85 for March 29, 2007: Cross-Site Scripting, Part 1.

Security Now! is brought to you by Astaro, makers of the Astaro Security Gateway, on the web at www.astaro.com. And by Nerds On Site. Looking to grow your IT service business? Find out how Nerds On Site can help. Visit Iwanttobeanerd.com.

It's time for Security Now!. And that means it's time to say hello to Steve Gibson from his ultra-secure, Tempest-hardened lair somewhere in Southern California.

Steve Gibson: I hope it's ultra-secure.

Leo: You know about Tempest; right?

Steve: Oh, sure.

Leo: The military spec for hardening, what, technology devices; right?

Steve: The idea was that anything that's electronic has emissions coming from it. That is, you know, any time you've got current running through a wire, you generate a magnetic field around the wire. And when the frequencies are high enough, that ends up generating radio frequency emissions, which of course travel much greater distances than magnetic emissions.

And so the whole concept behind Tempest was, like in the spy era, was that you could passively monitor the emissions of things like computers and reverse engineer what they were doing, like literally what was on their screens and what people were typing and so forth.

Leo: They call that, I think, "Van Eck phreaking." At least that's what Neal Stephenson calls it in "Cryptonomicon." Which we were talking about before we began here. But we're going to talk about books a little bit later on.

Steve: Yeah. We have some errata from last week. Actually only really one piece. Someone commented that I messed up my ports. We were talking about SSH and Telnet. And I know what the port numbers are, but I just got myself scrambled up there. So just for the record, in the errata category, FTP uses two ports, 20 and 21.

Leo: That's where I went wrong. I said 21, and you said no, that's not, that's the Telnet port. So FTP is 21, okay.

Steve: Exactly. FTP is actually 20 and 21 because, remember, the FTP uses two channels: one, a so-called control channel, which is where the client and server talk to each other about what they're going to be doing; and then a separate port is the data channel. So they have a control and a data channel instead of...

Leo: And doesn't it create a random port for the transfer, and that's why you have to use passive FTP because it's an incoming random port that the server sets up? I seem to remember...

Steve: The way it works, actually, is that in the normal sort of – the default FTP is called "active" FTP, or "non-passive" FTP.

Leo: Normal, right.

Steve: Normal, yeah. What's tricky about it, and in fact this caused problems traditionally for people behind NAT routers, is that these connections go in opposite directions. That is, you the client, you connect into the FTP server, and then the server connects back to you. So the idea being that you would say to the server, hey there, I want a connection, and you can connect to me on this port. So you tell the FTP server what port to connect back to you on. And so the client is also a server in the sense that it's going to accept an incoming reverse connection from the FTP server.

The problem is that, if you happen to be behind a NAT router, you know, computers behind NAT routers hopefully don't have to know that they're behind NAT routers. They don't have to do anything special. So the problem was, as we know, NAT routers make really good firewalls, hardware firewalls. So here the FTP client connects out to a remote FTP server. And it creates – it sort of starts listening on some high-numbered port on its computer, saying hey, remote server, connect back to me. But when the remote server tries to connect back to it, it's blocked by the NAT router because the NAT router has no idea that, you know, what's going on. It thinks this is just an unsolicited packet.

So there were two solutions. The first was to change the FTP protocol in so-called "passive" mode so that both connections are outgoing from the client, and that solves the NAT router problem. The cooler solution, which now all contemporary NAT routers support, is they're smart

enough to see a packet going out to port 20 of the remote FTP server and to literally look inside the packet to interpret the packet in the same way that the FTP server will, where it says make my reverse connection on this port. So the router dynamically opens that port for the FTP server's reverse connection. So it allows regular active FTP by being – basically by participating in the FTP protocol. Very cool.

Leo: So, and then just to finish, we said Telnet wrong and SSH wrong. So what are the ports for Telnet and SSH?

Steve: SSH is 22, and Telnet is 23.

Leo: 23, that's it, okay.

Steve: Exactly. And it's funny, too, because I wanted to – when I was making sure that I wasn't going to get myself scrambled here, I said, well, I'll just use GRC's Port Authority. I've got a cool facility on GRC. I don't even know if you know about this, Leo. But if you just go GRC.com/port=something, like [port=23](http://GRC.com/port=23), my server sees that. And any kind of URLs of that form, it's very flexible, and it will immediately take you to one of our Port Authority Database pages.

Leo: And so we just go there, and we look it up, and now we know.

Steve: And now we know.

Leo: And we should have done that in the first place. But we were doing it off the top of our head, anyway. Hey, before we go much further I just want to mention the great folks, as you know, who are back sponsoring the show, the Nerds On Site. I think they need a jingle [singing]. I don't know, maybe...

Steve: Oh, Leo, you may not want to ask them for that. That could be really scary.

Leo: Nerds On Site is actually a neat idea. It's a guild or a, I don't know...

Steve: Federation?

Leo: Federation maybe, yeah, of tech support specialists, IT experts. So the idea is you're still an independent contractor, you're in business for yourself, but you're not by yourself; so you can focus on your passion, not the burdens of running a business. Worldwide, seven countries. In fact, I think they added an eighth country. It's Canada, U.S., Mexico, England, Australia, South Africa, Bolivia, and more. And if you're in business doing any kind of thing, whether it's fixit technicians, website designers, programmers, project managers, even sales, trainers, security experts, antivirus gurus and more – but they especially love, of course, those folks, those nerds who troubleshoot, tear apart, and rebuild their own systems in their spare time – go to Iwanttobeanerd.com and register for a Nerds Only meeting in your area today. Like minds, building a business, it's a great idea. And I'm really glad that we can help them out. Nerds On

Site. Iwanttobeanerd.com.

Steve: And I guess since they initially gave us a relatively short three-week run and now have re-upped for three months, it must be that there's a good intersection between our listeners, Security Now! listeners, and the kind of people that they're hoping to find through this campaign, Iwanttobeanerd.com.

Leo: A lot of nerds listening, I think, Steve.

Steve: Well, speaking of which...

Leo: I don't want to depress you or anything, but I think that's the fact.

Steve: No, you know, these are our people, Leo.

Leo: One of us.

Steve: I'm one, god knows.

Leo: Someday we're going to tell the story which I just recently learned about your wine cellar that you're designing.

Steve: Oh, my radio telemetry – we'll do that sometime. We've had a lot of feedback, very positive feedback, from people who have followed our mentions in the past of Peter F. Hamilton's books. We talked about "Pandora's Star" and then the sequel, "Judas Unchained," and recently "Fallen Dragon." We had a bunch of comments back from people saying, wow, I really loved "Fallen Dragon," and specifically saying, you know, what else do you know? Who else have you discovered that is equally good? And I have to say that the kind of sci-fi I like is more along the so-called hard sci-fi variety. I've never really understood why wizards and unicorns and things are in the sci-fi section of bookstores. It's so annoying.

Leo: They've got nowhere else to put it.

Steve: I guess. But I don't think that's science fiction at all. It's fiction-fiction, you know, fantasy. And I guess often you see, like, fantasy and sci-fi. Well, to me these are completely different genres. But I did want to share, thanks to the quest from people for additional authors, I wanted to share another one of my finds that I am very excited about. This is a guy who is literally a rocket scientist. He designed pumps that are on the international space station. He's an aerospace engineer located out in Arizona. And he writes really good science fiction. He's got a site, and we'll put a link to this on our episode notes. It's scifi-az.com. His name is Michael McCollum.

And actually he and I were just exchanging some email because I wanted to find out about the various formats he offers. He has his books in paperback, so you can buy them that way, but also electronically downloadable, both PDF, Microsoft Reader, and in Palm format. And he just released the sequel to a book of his that I had read, I don't know, about a year and a half ago,

that I really liked. It was called "Gibraltar Earth." And I was telling you about it a little bit before we started recording. Humanity has never encountered any other aliens, and we're out in the future exploring around, and we stumble upon evidence of a vast intergalactic race that is so powerful that it just stomps, basically it enslaves any other species that it encounters.

Leo: Yeah, baby.

Steve: And they don't know that we found out about them, so we scurry back to Earth with our tail between our legs, and the story unfolds from there. Well, he's just done a sequel to that, that I haven't read yet, that I just discovered because I was looking up his site in order to tell our listeners about him. But there's another trilogy, the Antares trilogy, which is just spectacular. So anyway, I give it my unrestrained – obviously – full recommendation. All of his stuff is really good. He also has some freely downloadable short stories. I've not read them because I've just found his regular stuff to be so good. So without reservation I can recommend this Michael McCollum at scifiLaz.com.

I don't have any sense for how much people enjoy the eBook experience. You know that I'm a major eBook reader. And in fact you and I, Leo, have both just purchased the new Sony ePaper reader. I've seen it. I have not yet used it. But I'm excited to give it a try and see how it feels.

Leo: I just got it yesterday. And I used it last night, and I read quite a bit of Neal Stephenson's "Quicksilver," which I had in hardcover. But I thought, well, I don't want to carry these to Canada. They're too big.

Steve: Well, especially Stephenson books are like, I mean, they're like carrying an encyclopedia around with you.

Leo: So I bought the Quicksilver trilogy. And the funny thing is, it's only 10MB, all three books, which is three huge books. And I put them and a bunch of other stuff, and I haven't even – there's 100MB worth of storage to start with on that eBook reader, and I still haven't filled it up. It's taken me – I'm not used to reading on an electronic device. It's small and lightweight, very thin. And it feels like a paperback. I got a little booklight for it, too, because you need ambient light.

Steve: Right, because it's an ePaper display, which is to say they actually have got bi-colored spheres, white on one side, black on the other, and they electrostatically rotate the spheres to either have their white face front or their black face front. And then the beauty is, once you've spun these spheres, they require no power because they're in a viscous medium. They just hold their position.

Leo: It's very clever.

Steve: And ePaper, we've been reading about it for literally, I think, decades. But it's one of those ideas that's been on the drawing board and hard to actually put into production. And but Sony, as far as I know, is the first commercial ePaper technology. And what's cool about it is battery life issues just go out the window. They say you can read all of "War and Peace" four times on a single charge. So there's just none of that.

Leo: Well, we'll see. I have a lot of books on there. And I have the Antares trilogy; and I'm now, on your recommendation, going to buy the Gibraltar books. And unfortunately I had the Antares trilogy in Microsoft LIT format, which is an incompatible format. But I'm going to get a PDF. I wish more authors would do it. There are quite a few books I can't get, including, I mean, only "Pandora's Star" is the only Peter Hamilton book in the Sony CONNECT library. But I don't care if it's Sony CONNECT. Just put it on PDF, and I can read it. And I would love that.

Steve: And I really appreciate this, too, and I know you will. Michael does not...

Leo: No copy protection.

Steve: Exactly. He does not – well, except I wonder about the...

Leo: The LIT format is, yeah.

Steve: Yes. The Microsoft Reader would be. But it is a nonprotected PDF. I read his license page yesterday also. And he said, look, you know, I'm doing this because I want to trust everybody. Please don't post these. Please pay for them. Don't share them. If you like my work and you want more – and believe me, I really want more. This guy, his books are just so good. He says, you know, pay for them, don't steal them, and then this can work.

Leo: Well, I've paid twice now for Antares. And I don't mind. It's so cheap. 15 bucks for three books, come on.

Steve: Oh, and Leo, let me tell you, you have a treat in store for you with this Antares series. It's just so good.

Leo: Now I have too much stuff to read. This is a bad thing. This eBook reader has literally now 20 or 30 volumes on it.

Steve: And how did you find – I have not yet had the Sony ePaper experience. And you know that I don't mind looking at a little tiny Palm Pilot screen. How did you find that screen...

Leo: It's pretty good. You know, you can set the text size to small, medium, or large. And of course I'm getting old, so medium was good for me. It isn't high contrast. I would like it if it were a whiter background with black text. It's a little gray. But it's fine. I mean, it's better than a Palm, I think. And it's certainly a bigger screen than a Palm. And the only other thing that bugs me a little bit, there are two different ways to turn a page, neither of which is very natural. At least not for me. And there's a flicker when you turn the page which I find a little annoying. But these are minor. And I did find it easy on the eyes, and I read quite a bit. I read about 20 or 30 pages last night, and I didn't have any eye fatigue. So I think it's going to work, you know, be great for an airplane and great for traveling because I can now have my whole library with me.

Steve: Yeah, I'm really fanatical about battery issues. And so if I'm reading in a restaurant and

the waiter comes over to talk, I have to, I mean, I do, I turn the power off.

Leo: You won't have that problem with this.

Steve: Well, exactly. With this you literally – you just don't even worry about it. You just lay it down because it's not consuming any power.

Leo: And it looks a little bit like a paperback. You know, it's thinner than a paperback, but it's not much bigger. And so that's kind of, I mean, you don't look like too much of a weirdo.

Steve: Yeah. And it plays music, too, doesn't it.

Leo: I'm not sure how the battery life would be with that.

Steve: Yeah, it would probably pretty much kill it. Well, I had one SpinRite fun story that I wanted to share. I'm not going to read Tom's last name because, well, you'll see when I read the story why he may not want me to identify him. He said, "Steve, I looked all over the site and could not find anywhere to leave some feedback. So I'm mailing you here." I think he must have sent it to our support email, and then Greg forwarded it to me. He said, "I'm a Windows systems administrator, and I've been dealing with bad hard drives for quite a while. I've been listening to TWiT and Security Now! since the beginning. I've been trying to get my boss to purchase SpinRite for years, but it never seems to be in the budget. Since I am a father of two small boys, and daycare costs are killing me, I've not had the funds to purchase SpinRite for myself. But I have heard and read many of the testimonials and would love to try a copy." So he says, "Finally last week a co-worker came to me with his dead laptop. He had all of his wedding photos and many hundreds of dollars' worth of downloaded music on his drive. I looked at the laptop and knew for sure that SpinRite could fix his problem."

Leo: I wonder how he knew that.

Steve: How would you – why would you think that? And so he says, "Well, I got a copy of SpinRite through some nefarious means."

Leo: Oh, shame on you, Tom.

Steve: And he said, "Sorry, Steve. And sure enough, SpinRite brought the hard drive back to a working state, and I was able to recover everything." He says, "When my co-worker found out that I got everything back for him and made backups, he was delighted. He asked me what I would want in payment. I said \$89. He looked at me strangely and said, why 89? I told him about SpinRite and how I got a copy. So right then and there he pulled out his credit card, and we went to GRC.com and purchased SpinRite. I feel a lot better now, having a legal copy. Thank you, Steve, for the great product. And I can't wait to help others in the future."

Leo: It's a nice tool to have because – for exactly that reason. When people come to you, you can say, well, let me try SpinRite on it. And it's really nice to own, I have to say. It is

well worth the 89 bucks. That's neat. What a nice story. And a happy ending. He ended up buying it. That's great.

Steve: And you know, frankly, I don't have any problem with that. I mean, it's why I read this. Technically piracy, well, yes. But I recognize that not everyone who needs it is going to buy it. I really appreciate it, obviously, when people do because it's what keeps me on the air, literally. So it's a win for everybody.

Leo: That's great. Well, thank you, Tom, for sharing your story with us. And now what are we – we haven't told anybody what we're going to talk about today.

Steve: No, 20 minutes in, and this is a big mystery.

Leo: Well, if you heard the title and you heard me announce it, I guess you probably know. We're going to talk about something that is a big security issue.

Steve: Wait, wait, wait. Are you announcing this some other time? I don't think you even gave us the title yet.

Leo: Well, you don't hear it, but at the beginning of every show I say welcome, and I tell them the name of the show and the date. Just so that if you're listening on a shuffle or something, and you don't have a readout, you know whether you've heard this one or not. So they know that this is Cross-Site Scripting Part 1. They may not know what that means. But they know what we're talking about. So what are we talking about?

Steve: Yeah. It's something that we have never yet touched on, I don't even think at all. Most of the time we'll swing by some topic when we're talking about something else, and then move on, and then later, weeks or months later, come back to really focus on something that we've talked about before. I would generalize cross-site scripting a little bit more and say that it is about web-based code injection. And cross-site scripting is one major way that that can happen.

Leo: The other one is buffer overflows.

Steve: Well, no, because that's not necessarily web-based.

Leo: That's a technique.

Steve: Exactly. Well, in fact, we've of course extensively talked about buffer overrun issues. And in buffer overrun issues a hacker is injecting executable machine code, essentially, directly into some other application over a communications connection. So you're certainly right that that is also remote code injection. In this case we're talking sort of about web-based apps which turn out to have a disturbing number of vulnerabilities to this kind of exploit. But the second type is something called Sequel Injection, or SQL Injection, SQL having pretty much become the de facto dominant database language because it's, you know, there's now Open SQL, MySQL. Microsoft of course is pushing SQL, or Sequel, as that acronym is often

pronounced. And it turns out that because SQL is scriptable – you know how I feel about scripting, Leo.

Leo: Yes.

Steve: There's all kinds of problems there, too. So for a long time I've had this on my list of things to talk about. But something happened last Saturday afternoon on the East Coast that sort of said, okay, now is the time to discuss this. So this is a big topic. And because we're doing an audio podcast, and I want to make sure people are able to visualize this, I'm not going to try to cram too much content into this first introduction. So this week is sort of introduction to this issue. This will be sort of the teaser for what I think will be the next two weeks. And then the week after that we'll be back to one of our Q&As where we'll be able to tie up any loose ends and answer any questions that have come about from talking about this.

So what happened last Saturday was that a guy named Billy Hoffman from SPI Dynamics, which is a well-known, good-reputation, security research firm, he gave a presentation at a funny-named hacker conference called ShmooCon. It's a three-day – it was Friday, Saturday, and Sunday – a three-day East Coast hacking conference. The title of his talk was "JavaScript Malware for a Grey Goo Tomorrow."

Leo: I like it. I don't know what it means, but I like it. The grey goo is in that nanotech problem that people have raised? Maybe if there were too much nanotech we'd all become grey goo?

Steve: Just be reduced to grey goo, just sort of like thoroughly homogenized.

Leo: A slurry.

Steve: Okay. So leading up to this there were a couple interesting stories. And in fact afterwards – that is, stories in the news. And I want to run through these quickly because they are full of really good quotes from Billy and other people. In eWEEK this basically was covered under the subject "Tool Turns Any JavaScript-Enabled Browser into a Malicious Drone."

Leo: Wow.

Steve: A new tool too dangerous to give away can turn any PC – Windows, Mac, Linux – or any device with a browser into a site attacker. The tool, called Jikto – that's actually "J" stands for Java. There was an earlier tool, a web vulnerability scanner called Nikto. Of course "nikto" is a word...

Leo: Nikto barada...

Steve: Exactly.

Leo: From "The Day the Earth Stood Still."

Steve: You definitely have your geek hat on.

Leo: Klaatu barada nikto. That's what you had to tell the robot to save the Earth.

Steve: Well, because the robot's name was Klaatu. And so you had to address Klaatu, and then give him this coded phrase. So anyway, this is not Nikto, this is Jikto, `JIKTO`. It's a web application scanner that searches for cross-site scripting vulnerabilities. Billy Hoffman, a security researcher with SPI Dynamics, demonstrated what the tool could do at the ShmooCon hacker convention on March 24th, namely Jikto, which is written in JavaScript. Okay, JavaScript. You know how I feel about lovely JavaScript. How you feel about Web 2.0, Leo, and I know that [audio glitch]. But this can surreptitiously latch onto a browser – and remember, any browser, Windows, Mac, Linux – that has JavaScript enabled. After silently inserting itself to run inside any browser, be it that of a PC or even a cell phone, Jikto can then search sites for cross-site scripting vulnerabilities and report its findings to a third party without the user of the infected browser being aware.

So get this. You go to a site with JavaScript enabled, which we understand most people have to have to get the functionality of Web 2.0 and all the fancy stuff that's being done these days. Or you receive email, and you're using a web-based email viewer, as most people are by default. One way or another, this JavaScript runs. Basically it commandeers your browser on the spot, turning it into a scanning tool which will then scan other websites for vulnerabilities and report what it finds to a third party. And so Hoffman says it can also replicate itself onto sites containing cross-site scripting vulnerabilities, meaning that it has the capability of also being a worm which can self-propagate and then spread via latching onto other visiting browsers. This is something that JavaScript wasn't supposed to be able to do. But unfortunately, Hoffman says, it can. And he demonstrated it.

So that was one, that's part of one story. Then I wanted – there were some more good quotes in – a guy named Joris Evers, who reports for CNET News, wrote a story before and after. And there was a bit of controversy stirred up by his first story, which was – there were, like, blogs about it because he either made a mistake or believed, one way or the other, that Billy Hoffman was going to release the JavaScript code for this at the conference. Maybe they said so; maybe they changed their mind. It's not clear. But Joris's first story says, "A security researcher has found a way hackers can make PCs of unsuspecting web surfers do their dirty work without having to actually commandeer their systems. That's possible with a new security tool called Jikto. The tool is written in JavaScript and can make PCs of unknown web surfers hunt for flaws in other websites, said Jikto creator Billy Hoffman, a researcher at web security firm SPI Dynamics. Hoffman, who developed the tool as a way to advance web security, plans to release Jikto publicly later this week at the ShmooCon hacker event in Washington, D.C. 'This is going to drastically change the scope of evil things you can do with JavaScript,' Hoffman said. 'Jikto turns any PC into my little drone. Your PC will start attacking websites on my behalf, and you're going to give me all of the results.'

"With the advent of online applications, hackers have shown increased interest in breaching web security through vulnerabilities such as cross-site scripting bugs and SQL injection flaws. Even though these have been around for years, such security problems are increasingly being reported and exploited. 'Jikto is a web application vulnerability scanner. It can silently crawl and audit public websites and then send the results to a third party,' Hoffman said. 'Jikto can be embedded into an attacker's website or injected into trusted sites by exploiting a common web security hole known as cross-site scripting flaw.'"

Vulnerability scanners by themselves aren't new. Hackers often use such tools to find holes that let them break into systems. Jikto is like Nikto, a web application bug scanning tool popular among hackers. The difference is that Nikto is a traditional PC application, while Jikto runs in a web browser and distributes the bug-hunting task across multiple PCs.

Leo: Like SETI@home for hackers.

Steve: Exactly. We're all familiar with the whole problem of bot fleets and distributed DoS attacks. Now we've got distributed vulnerability scanning courtesy of Jikto, which is just written in JavaScript, which was originally designed not to be able to do this kind of thing.

Leo: Now, he didn't release the code. He just demonstrated it working.

Steve: Correct.

Leo: We don't know exactly what it's doing. It may in fact be calling a Nikto server somewhere. In fact, I suspect it probably is. You wouldn't write Nikto in JavaScript. You couldn't really make that something you could...

Steve: Well, the code has leaked.

Leo: Oh, it has. Oh, well. Now we'll find out. And the reason I mention that is because, if it does do something like that – and this is an example of why security – this happens all the time that they demonstrate these flaws at security conferences, but they don't give enough details to really know. And for instance, if this requires a Nikto server somewhere, you would be caught red-handed at some point because it would be calling your server, and people would know who you are.

Steve: Yes. And again, I've not looked at the code.

Leo: That's an example of a potential flaw that you wouldn't know until you saw the code.

Steve: Well, and the other thing, too, certainly if it's going to be returning its results back to, for example, its originator, then certainly you're able to look at the code and track that down, too. The problem, of course, is that these could be sites where you've got an unfriendly nation or an uncaring nation with whom we don't have a relationship. Or they could use the same sort of obfuscation techniques that the IRC chat servers do. For example, it could log to an IRC chat channel and send its results into the IRC cloud, where it just disappears because it's being relayed autonomously from one IRC server to another.

And the biggest problem is that, as is always the case with these things, once someone knows it's possible, it's not hard to recreate it. That is to say, just the fact that this has made the news, that we're talking about it, it's been blogging all over the place, if you put Jikto into Google now you get just pages of this stuff. So now it's known that someone did this. And once you know that someone did this, it's just not difficult. Everybody is smart out there who's involved in this. And so it's just not difficult to recreate these kinds of results.

Leo: And you said it's leaked out. So even if you weren't smart. That's what I worry about. I don't worry about the guys who are smart enough to figure it out. I worry about the script kiddies who aren't, but will use this maliciously.

Steve: Well, yes. And in fact, it's interesting because there are papers that have been written by researchers as much as ten years ago. In fact, I've got one here that was put together by a bunch of Microsoft guys in early February of the year 2000. And I just want to read verbatim what they say about cross-site scripting. They say, under "Consequences of the Attack" – and I'll put links to this whole paper, and we'll be talking about this again. And we're going to get into the real detail, the nitty-gritty of what this is, next week and the week after, although week after we may talk about SQL Injection because it's related but not the same.

But Microsoft writes, "Assuming that a particular web server is vulnerable to cross-site scripting attacks, the attacker can run a script in the wrong security context. This means that cookies can be read, locked-down plug-ins or native code can be instantiated and scripted with untrusted data. User input can be intercepted. Any web browser supporting scripting is potentially vulnerable, as is any web server that supports HTML forms. And even HTTPS, that is, secure sockets connection, a secure connection provides no immunity. Data gathered by the malicious script can be sent back to the attacker's website. For example, if the script has used the DHTML object model, that is, dynamic HTML, to extract data from a page, then it can send it to the attacker by fetching a URL. As long as the user navigates within a given domain, that is, within the same domain, a robust exploit script can follow the user. This attack can be used against machines behind firewalls. Many corporate local area networks are configured so that client machines within their LAN trust servers on the LAN, but do not trust servers on the outside Internet. However..." So it's a way of breaching those zones of trust.

It says, "However, a server outside a firewall can fool a client inside the firewall into believing that a trusted server inside the firewall has asked it to execute a program. To do this, all the attacker needs is a web server inside the firewall, like the corporate Intranet server, that doesn't check fields in forms for special characters. Only one page on one web server in a domain is required to compromise the entire domain and network. This is true even if the vulnerable web server doesn't hold any important data." Because of course it can be used as a jumping-off point. "It can still be used as part of an attack on other machines within the same domain. All web servers should guard against this attack, even ones that don't perform critical tasks."

So, okay. In a nutshell, and this is what we're going to expand upon next week, the big problem is sort of a gotcha, that the originators of HTML and Web never thought about. And that is, what happens when, for any reason at all, users, that is, clients of a web server are able to provide anything that is going to be redisplayed. And that's the fundamental flaw in the web architecture. That is, originally, in day one of the Web, as we know, web servers were serving static pages. That is, users would go and, you know, you just clicked on links, and you looked at pages that the webmaster and his people had created for people to download and view. That is, browsers were simply browsers. They looked at static pages that never changed. Every time you went back you saw the same page.

Then of course little additions were made, like web counters, like how many visitors have been to this site, or how many visitors have viewed this page. So there we were seeing a little bit of dynamic content. But of course the whole big explosion in the 'Net came, or especially in the Web, came from making sites interactive. Any kind of a forum is an interactive site. DIGG is an interactive site where, anytime users are able to post content to the server, which it will then serve to other users, it radically changes the rules because what happens is you immediately get an interaction between the notion of trusted content coming from a server and the fact that you are displaying content from an untrusted source. That is, you know, anybody can log onto an open forum and post their comment. When they do that, text that they've provided is then being stored by the server and is being redisplayed to other people who visit the site.

Well, here's the problem. There is no separation between textual displayed content and executed scripting content. That is, because of the evolution of the Web, there was no really clear and clean delineation between text that the web server sent your browser, which you would see, and text which it would execute. In fact, all you have to do is have an open bracket, you know, "less than" sign, the word "script," and then a closing bracket. And then anything

after that, until the closing script tag, is treated as scriptable content. And arguments to that script tag tell the browser what kind of scripting it is, what language, what version and so forth, so that it's able basically to turn its scripting engine loose on whatever this text is in between these tags. And it turns out there are a bunch of different tags that will end up causing the interpretation of text as script, which is to say, you know, to run scripting; and that all you need to do is – all an attacker needs to do is figure out some way to get a web server to serve this script when it thinks it's serving just non-executable text.

And even after this problem was known, it turns out that there are still very clever ways of finding holes in the prevention of this. For example, you would imagine that you could simply tell your web server, strip out any bracketed things, that is, anything with a bracket and a closed bracket. Well, there are normally exceptions. People want to be able to put italics in their postings or underline words in their postings or have some formatting controls. And it turns out that exceptions in the way text is parsed have still allowed holes to be found. And that's, for example, what the so-called "web vulnerability scanners" are scanning for, is they're looking for any sorts of active content.

Historically, web servers even came with a bunch of default pages and default scripts which were vulnerable. So you could just have, for example, there were some versions of Microsoft's IIS where, when you installed the server, it gave you a whole bunch of, like, starter pages, which many – you know, like sample pages – which many people left alone. And they were vulnerable to cross-site scripting, so people would immediately be able to leverage them. And in fact, this is one of the ways, you know, we're always hearing about websites being defaced, you know, sites being changed, home pages changed and so forth. It has been these kinds of cross-site scripting which allows those kinds of defacement to happen. That's been the enabling technology.

Leo: Well, I guess we're going to have to rethink the enabling technology, then. I mean, it's a little scary.

Steve: It really is scary. And, yes, it's powerful that you can have inline scripting just mixed right in with your text. But the fact that you can do that, the fact that, for example, scripting isn't somehow sequestered completely separately from content, but in fact is just a form of content, and it is, I mean, to say it's active or alive, I mean, that's almost an understatement. It's hyperactive. And it really does create an opportunity for exploits.

Leo: So we're going to continue to cover this subject in the next couple of episodes, give you some more in-depth look at how cross-site scripting works and so forth. In fact, what are we going to talk about next time?

Steve: Next time will be some exact examples of all the very things that a web server – the few mistakes a web server needs to make in order to allow cross-site scripting exploits. I'm going to get very specific with some examples and walk our listeners exactly through, for example, how you could end up with another – with an attacker capturing your cookies, which is your log-on credentials for a site like Amazon or eBay, which would allow them to then assume your identity and take it from there.

Leo: And meanwhile, you might want to download Nikto and run it against your sites, see if you've got any of these vulnerabilities. I presume Nikto does a good job of it. I don't know of other programs that can do the same thing. But that's very interesting, very interesting. It makes you want to go out and get Astaro's Security Gateway right now and run it everywhere you can.

This show is brought to you, as I mentioned, and I do mention every week, by the good folks at Astaro Security. They make the great Astaro Security Gateway, which is a security appliance that runs on open source software. The newest version, Version 7, is out now, with email encryption at the server side, and so you don't have to worry about it on the client side, at the device. Also secure remote access via SSL VPN, which is fantastic. And scalability via clustering. Astaro's developed a unique brand of active clustering that enables a load distribution for as many as ten gateways, eliminating the need to install additional load balancers. It's very cool. It's a patent-pending technology which really increases the speed and reliability of network traffic. So you know when you've invested in Astaro's Security Gateway, you can grow it as your enterprise grows.

And don't forget, by the way, home user licenses are available for free for the v7 package, including all subscriptions and Astaro Up2Date. That's a great deal. Astaro, find out more by visiting on the web at www.astaro.com. Or if you're so inclined you can also find out more about Astaro. In fact, go to them and sign up for a free trial of the Astaro Security Gateway in your enterprise by going to 877-4AS-TARO. Wait a minute, let me check that and make sure that's right because I actually don't have it sitting in front of me. I have this all memorized now.

Steve: It sounds very familiar, Leo.

Leo: It sounds like the right number. But let me just look it up.

Steve: In the meantime, while you're doing that, I'm going to correct my own errata from the beginning of the show. Even as I said it I knew I was misspeaking. And so for anyone who has already sent off email telling me that I've still got my FTP ports backwards, I mean, I know so well that the server listens on port 21. And I'm sure that I said it listens on port 20, which I know is not the case. FTP is the main, is the control channel for FTP, the server listens on port 21; and then it's port 20 that is the reverse channel, the data channel.

Leo: I always thought it was a random port. I must have misunderstood the spec when I was looking at it.

Steve: Well, it's random at the client end. But because...

Leo: Oh, okay, yeah.

Steve: Yeah. So it's whatever port the client is able to be assigned by the OS. Remember that service ports are only able to be opened – okay. Service ports are those ports from 1024 and lower. Those are only able to be opened for listening by privileged processes in UNIX so that clients are unable to listen, for example, on 21 and 20. They're able to listen on high-numbered ports. So the idea is that the client makes the remote connection to the server on port 21 and says, I've just opened port whatever, 32654. Connect back to me. And so the server connects from its port 20 up to the client's specified port number.

Leo: Okay. Got it. And...

Steve: Fixed my own mistake there.

Leo: And the number of Astaro is – I was right – 877-4AS-TARO for a free trial in your home. We thank them for their support. Well, I think now we have – this is the first time we ever had an errata correction in the actual show, correcting an errata at the beginning of the show. So that’s impressive. The turnaround’s getting very quick on these errata. Steve, lots of fun. We’ll come back next week and talk more about cross-site scripting. And of course your questions should continue to flow to GRC.com. He’s got a form there. I’m sure it’s safe. Tested for cross-site scripting.

Steve: Yes, because I’m not displaying anything that anyone posts. Although, Leo, I have to say, I will confess that GRC did have a cross-site scripting vulnerability years ago.

Leo: Oh, interesting.

Steve: Back when IIS, Microsoft’s IIS was having so much trouble with its own URL parsing, remember there were all kinds of exploits where you were able to, like, put funky Unicode characters in the URL in order to get IIS to misbehave. And I was unhappily using IIS. And I didn’t have any sort of filter. So I wrote my own URL pre-parsing filter to intercept any and all incoming URLs before they got to IIS. And what I did was I had a page that would intercept and say – oh, I called it the APF, the Advanced Prophylactic Filter, because it was a prophylactic for IIS that would basically protect IIS from any badness trying to get in. And what it did was, when you got intercepted by this, it showed you the URL that was illegal. Oops. Because what that meant was that the user was providing something, anything, which could cause the server to redisplay it, even as an error message. And so that created a cross-site scripting vulnerability, which someone pointed out to me and I quickly changed. So it’s like, yeah, I mean, it’s so easy to do.

Leo: It is.

Steve: You have to be just ever vigilant. It’s very much like, although different, of course, from the buffer overrun problem where it just requires so much vigilance. And so the problem is, with all of our contemporary sort of next-generation Web 2.0 applications, which are all about, like, you know, Wikipedia and DIGG and online forums and communal blogging, any situation where the server has any reason to display what someone provides, that creates real trouble due to the fact that text and script, they’re able to cohabitate on a page.

Leo: Right. Well, go to GRC.com. Look at that form. Find a hole. Find a flaw and let us know. Meanwhile, while you’re there you can download 16KB versions of this podcast, and Elaine’s great transcriptions so you can read along. On complicated subjects like this I think it’s always helpful to read along as you’re listening to Steve. You can refer back to it, show the boss, explain why you have to recode the entire site, all of that is at GRC.com along with all of Steve’s free security software and the great SpinRite, the world’s finest disk recovery and maintenance utility.

Steve, we’ll wrap this up, but we’ll reconvene in a week. What do you say?

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

This work is licensed for the good of the Internet Community under the Creative Commons License v2.5. See the following Web page for details:

<http://creativecommons.org/licenses/by-nc-sa/2.5/>