GRC's XSS Adventure

Description: Father Robert and I discuss concerns over a significant expansion in effectively warrantless intrusion into end-user computers; the forthcoming change in Internet governance; generation of a shiny new (and bigger) DNSSEC root signing key; Google's next move in using Chrome to push for improved security; the interesting details emerging from a successful NAND memory cloning attack on the iPhone 5c; some fun miscellany. Then I share the details and findings of a recent Cross-Site Scripting (XSS) problem on GRC, including the best website security scanner I found and now recommend!

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-578.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-578-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here to go all propeller-head on breaking through Apple's NAND Flash. Is it proprietary? You're going to find out. Google has changed the Internet into being more secure. ICANN says goodbye to IANA and hello to DNSSEC and DANE. And cross-site scripting: It's the adventure of GRC.com. Security Now! is next.

FR. ROBERT BALLECER: This is Security Now! with Steve Gibson, Episode 578, recorded September 20th, 2016: GRC's Cross-Site Scripting Adventure.

It's time for Security Now!, the most secure show on the Internet, where we are guided through the security issues, software updates, and coding conundrums by our Explainer in Chief, Mr. Steve Gibson, of course from GRC.com. Steve, it's so good to be working with you again.

Steve Gibson: Father Robert, is absolutely a joy. I was thinking last week, I was hearing Leo already, even though he hadn't left yet, bemoaning the fact that he had to be back in two and a half weeks. And I'm thinking, okay, wait. So we've been hearing about this vacation for the last six months. Now, as you're getting ready to go, you're worried about having to come home too soon. So the good news is I think that in the future you and I will have probably more opportunities like this to do the podcast together.

FR. ROBERT: And you know what, Steve, I think that's actually healthy. That's the whole idea, that he's getting used to the good life. He's worked really hard to build this thing up, and it's time for him to actually enjoy it. And I've got to say, after going on a vacation for a month, if you've ever done one of those, these little two-, three-day vacations just don't - they don't mean much anymore.

Steve: Right. Takes a while to disconnect. And then you're not in a hurry to reconnect.

FR. ROBERT: That's so true. And actually I've got a Jesuit analogy for that, an analog, actually. It's when we go on retreat. So we disconnect for at least eight days a year, and by "disconnect" I mean really disconnect. There's no Internet, no phone.

Steve: Are you allowed to talk?

FR. ROBERT: Some of us - you are allowed to talk, but most of us don't.

Steve: Oh.

FR. ROBERT: It's just sort of you give yourself permission to just be introspective. But it does take a good day to three days for you to get into that mode. And it's the same thing for vacations. So, but we're not going to do that, Steve. What we're doing is we're going the other way. We're going to go full steam ahead. We're going to look at some very interesting security issues that have been popping up over the world.

Steve: So, yeah. We've got - this is going to be a fun podcast. I haven't mentioned this at all. But there was an event a couple weeks ago that unfortunately distracted me in an unexpected fashion, when a security researcher reported an unknown cross-site scripting vulnerability on GRC. So the adventure that ensued resulted in my, first of all - well, okay. Sort of chronologically, in looking for a way to determine what the problem was, I looked at three site-scanning, site security scanners. And one of the three stood out and has become the formally recommended terrific solution that I'll be talking about. And then I'm also going to talk about what it found, which is interesting because we've sort of - we've glanced over cross-site scripting problems, but this is a perfect opportunity to dig in a little bit further.

But there's also a bunch of other news. We're going to talk about the concerns over the significant expansion in what is effectively warrantless intrusion into end-user computers which, unless Congress acts to block it, will automatically take effect at the beginning of the year. There's also a forthcoming change in Internet governance which has been causing some controversy, I think largely due to misunderstanding what it is and what it means. There's a new DNSSEC root signing key on the way.

Google has moved further or is continuing to move in its push for improved browser security, well, web security using the power, the marketing power, essentially, of its browser in order to force change. And that's always been controversial because they're regarded as being rather strong-armed. But it's being effective. And then we got some really interesting details from an individual who successfully cloned the EPROM, the NAND memory in an iPhone 5c, which was the target, of course, of the San Bernardino issue, the FBI wanting Apple to decrypt the phone and so forth. And the write-up is fascinating because it gives us, first of all, proves that it's possible to use that theoretical technique which had been proposed. But there's some surprising things hidden in the details. And we have a little bit of miscellany, and then we're going to talk about the cross-site scripting as it happened on GRC.

FR. ROBERT: Wow. I guess there really must not have been much going on because we're only talking about an intimate look at cross-site scripting, about how the government's getting more spying powers, about how the iPhone has been broken into, and of course how the governance of the Internet's going to change. So I guess it was a slow week.

Steve: Yeah, well, we try to scrape some things together to talk about.

FR. ROBERT: All right, Steve. Set us loose on this wonderful, wonderful amendment to Rule 41 because this, at first glance, it seems a little scary.

Steve: Well, so what's most upsetting about this, I mean, we'll talk about what these amendments are. But the thing that's most annoying is that an obscure process was apparently deliberately used to cause this change to occur unless Congress acted to block the change. So as we know, the way the U.S. legislative process works, Congress creates, it's supposed to be the creator of laws, which are then voted on typically in the House and the Senate. Then, if there are differences in those bills, then they try to pull them together and ratify them to a single one in a conference. So the concern here is that Congress had no role in writing or approving these changes. They were developed within the U.S. court system through what was described as an "obscure procedural process."

Okay. So what happens is this proposal, or these changes, will automatically go into effect on December 1, that is, so the beginning, well, the last month of this year, which doesn't also give us much time. There's just six more working weeks until the Senate recesses, and there is a long to-do list of other things that have to happen. What the rules essentially make lawful is the government and governmental law enforcement's ability to hack any number of computers, I mean, like millions or more, with a single warrant. So the government says it needs this power to investigate, for example, botnets, as we know so well, networks of devices which are infected with malware and controlled by a criminal.

But these are typically innocent victims' machines. And I talked about this years ago on the podcast. I was involved in a big conference call with, like, the person in the government, like the DOJ person, I don't remember now who she was. I want to say Jennifer Granholm, but I don't think that's the case. But it was somebody way up. And a group of people who were involved, this was back in the Code Red and Nimda days, where innocent Windows machines had been infected by this worm and were then perpetuating that infection, were out scanning the Internet independently in order to find targets. And in this discussion we had, the security researchers sort of on our side were saying, is it not possible for a white hat hacker to fix these machines? That is, we knew the IP addresses because they were non-spoofed TCP connections.

So we had a list of the infected machines. And they had essentially an open backdoor, which is how they got infected. That same backdoor, that is, that same flaw in Windows back then could be exploited to remove the bad thing. And this was like half of the conference call was discussing the legality and, unfortunately, the illegality of doing that. It was absolutely against the law, even for the best of reasons, for a system to be modified without its owner's permission.

That's what this changes. This essentially is what the government has been wanting, what law enforcement has been wanting, they would say "needing," for quite a while. And we all feel victims from time to time, depending upon who's being DDoSed or attacked, of networks of autonomous slave computers. So essentially what this does is it changes that from being illegal to being legal in a very broad way.

Essentially, it represents, in fact Wired magazine editorializing said: "This kind of vast expansion of government mass hacking and surveillance is clearly a policy decision. This is a job for Congress, not a little-known court process. If Congress had to pass a bill to enact these changes, it almost certainly would not pass as written. The Justice Department may need new authorities to identify and search anonymous computers linked to digital crimes. But this package of changes is far too broad, with far too little oversight or protections against collateral damage."

And the biggest thing this changes is it allows a warrant to be issued by a court, which then grants search and penetration right to the requesting agency, essentially irrespective of jurisdiction. We have traditionally had jurisdictional boundaries such that a magistrate was only able to issue a warrant for a specific jurisdiction, something that they controlled. This removes that limitation. And you can understand. Like, you know, the Internet is global. We've got computers all over the place. So you can see how they want this. But this is a big change.

And again, I don't think anybody would have a problem if it was well argued and thought through in committee, by Congress. Ron Wyden is screaming at the top of his lungs, trying to keep this from happening, saying this has to be stopped. This is far too broad. The language is too permissive. And so the concern is that this has sort of happened without our normal process of figuring out is this something we want to do. Look at the problem we've had with Net Neutrality that refuses to go away, despite the fact that we keep deciding how it should be. But people, powerful entities don't want it to be that way. So it never seems to go to bed.

And now we have something sort of similar. Essentially, this does allow the FBI to implant malware, that is, their own software, in a machine that they believe they have reasonable cause to do so with, like it scanned somebody, or it sent a probe somewhere. And, I mean, it's a big change. And I wouldn't argue that we don't need to discuss solutions to this problem. But having ISPs prevent spoofing of their own addresses in packets that egress their network would be a nice, simple, technical first step to take, rather than immediately saying, oh, the only solution is to let law enforcement do anything they want.

FR. ROBERT: Steve, I'm with you. And I understand that this is one of these laws of unintended consequences, where I see the problem that they're trying to solve. I understand how they're trying to legislatively deal with it, and yet I can very easily see how this could be abused and how this could lead to horrible, horrible secondary effects.

But let's be fair. I'm going to play devil's advocate. I don't think the law should be as it currently is. But if I'm law enforcement, I could make a good case for saying, look, we need to update the warrant process because it's so difficult to know where a perpetrator might be operating out of. They may have servers in all 50 different states. They may have servers outside of the country. And I cannot, in my investigation, wait for a warrant for every jurisdiction that a mal-packet may have passed through. And so this is essentially giving me the authority to track down the bad guys where the bad guys actually live. What would be your answer to that? I mean, if you had a law enforcement officer come to you and say, what's the better way to do this, what would your answer be?

Steve: I don't have one. I mean, the other interesting thing is that cyber warfare is becoming a bigger issue. I mean, it's like it's on the radar now. The politicians are talking about it. It's moved from sci-fi into reality, like with chilling speed. And so there are two sides to this. There is, might this be itself a backdoor into providing a means by which the U.S. government would have the legal authority, A, to reach into attacking machines and deal with them; or, B, to commandeer IP-connected bandwidth generators, which is what our machines are these days, and rally them for an attack.

FR. ROBERT: The interesting part about that is I could understand the law enforcement mentality that says, okay, this server is generating bad packets, so therefore we're going to confiscate it. We're physically going to go and take it away so that we can analyze it and find out where the attacks were coming from. But that's not how the Internet works

anymore. I mean, if you're running a service somewhere on the Internet, most likely you're on a shared box. You're a virtualized machine. So there's nothing that LE can take without disrupting what is most likely dozens of other completely legal, completely legitimate traffic sources.

Steve: And we've seen stories of that.

FR. ROBERT: Yes, we have.

Steve: I mean, the FBI has gone in and yanked out racks that had completely unrelated domains that just disappeared because there was one bad guy among them.

FR. ROBERT: Right. In fact, during the MegaUpload debacle, when the DOJ actually physically took servers, there were people who were not hosting pirated content, did not have copyright content. But it was a multi-year fight for them to get the data off the hard drives because essentially the DOJ was saying, you need to prove to us that you own the data that you own. And they're saying, wait a minute. How are we going to do that? Our ownership was the hardware. You took the hardware. I don't know.

Steve: Yeah. There was a - I don't know how long ago it was. It was maybe more than a decade ago there was a relatively mild attack on GRC that was not using spoofed packets. It might have just been ICMP flood. But so I captured the traffic and looked at it, and then wrote a little Perl script to run a quick reverse DNS through the entire block of IPs. And I found four that were oc.oc.cox.net. Well, that was clearly - that was my own reverse DNS. That was Orange County Cox Cable.

And so I contacted some local friends at the FBI, and I said, "Hey, I've got four IPs belonging, I'm sure, to some people who don't know their home PCs are infected with some malware. Is there any way you could contact them and arrange for me to get permission to take a look at them?" And as it happens, my relationship was strong, and they were able to get a hold of one of the families. Oh, so I gave them the IPs because I didn't know who they belonged to. So they asked Cox for the physical real world identities of these families, after opening a case to make this legal, and contacted the families. One of them was only a couple miles away. I drove over and found the bot that was attacking me. And of course that was part of the process of unwinding the mystery of the very early attacks by some hacker named Wicked that GRC underwent.

So that's an example of the way it's being done today. I was only able to do that because I have that kind of working relationship with law enforcement that made it possible. So not everybody can do that, and they wouldn't know what to do with the information probably if they had it. But still, it involved a lot of real-world interaction with the court system, with the bandwidth provider, with the victims of the attack, and getting their permission to allow a non-law enforcement security researcher to make a house call. And so it worked, but it doesn't scale.

And I think what we're - this feels to me like at-scale issue, that is, all of this is getting bigger. Attacks are getting stronger. And existing tools that were designed for the physical world are failing the cyber world. So, I mean, I can absolutely see both sides of it, too. I guess the only problem is this didn't have any oversight. This was clearly - the only reason these amendments happened - and this, by the way, it's like an obscure rule in something that was completely unrelated to this, literally just slipped in, was to avoid this process. Now, maybe it's a problem that something as broad as necessary could not get past. But that's the way the system works. And it seems wrong to arbitrarily bypass it, if some people's interests are to do so. That's not a democracy.

FR. ROBERT: Right. And unfortunately, I think we're just in this very scary part. And we've known this. We've talked about this on multiple shows on TWiT.tv, where law is so far behind where we live in the virtual world that, when they try to apply law that was built for the physical world, it always ends up breaking something, or it always ends up just looking foolish, ham-handed. We've got people in the chatroom, like Eric Duckman and the like, who are saying, you know, it's easy to say that you're against this because of course there's going to be unintended consequences. But it's more difficult to say, okay, well, then what is the solution? What do we do? If you don't want to do it that way, and you admit that we need this authority, we need the ability to reach into a box that is generating mal-packets, that is owned, especially a box that - let's take your example. You were lucky enough to have a relationship with law enforcement. You were lucky enough to have an IT administrator who was willing to let you in and fix what was wrong with his gear.

Steve: Oh, his team, the two teams there were very happy because their big problem was they could no longer record pirated CDs because the computer was so busy attacking things that the CD burner no longer worked.

FR. ROBERT: We're trying to do our illegal things.

Steve: My music, my music, I don't have my music. It's like, okay, that's got our priorities straight.

FR. ROBERT: No, but, I mean, even in your case, that was someone you could work with. There are so many servers that have been compromised on the Internet,, that have been abandoned for 10 years. And they will never get patched, and they will never be looked at.

Steve: In the closet.

FR. ROBERT: Because someone spun them up a while ago and forgot about them. And what do you do about that? When there is no contact person? When it's paid for the next five years, and it's just going to spin at the bottom of that rack, but it is completely useless and generating traffic across the 'Net. And so I think all of us understand that there needs to be something, some provision to allow us to fix that. But I think you're right. It's just the lack of oversight really does, it makes us feel like the FISA court again.

Steve: You know, you asked me, you posed the theoretical problem, what could we do? An interesting compromise is service termination. That is, just disconnect the bandwidth to that malfunctioning machine so it's no longer able to contaminate the shared resource that is the global Internet. If somebody then is behind it who's trying to send letters to their mother, and suddenly their Internet goes down, then they contact their ISP and say, "Hey, I don't have any connectivity." The ISP says, "No, we had to shut you off because you've got malware in your machine." And then thereby get permission to proceed. So maybe just cutting the cord of these things, of these machines that are misbehaving. Again, it's not a perfect solution, and it is subject to abuse. But it's a midway between doing nothing and actively having a court issue a warrant that gives global rights to law enforcement to do what they cannot do today.

FR. ROBERT: Steve, there is one more provision in this rule change that we haven't yet talked about, and that is - this is interesting because it does sound like the lawmaker was trying to be responsible, that there is at least the mention that the best effort must be made to inform...

Steve: Notify.

FR. ROBERT: ...the owner of a server that it's been searched. I'm not sure if that was just tacked in at the end, or if that was the original intent of the bill, to make this seem balanced. That's not enough oversight, though, because, I mean, the language is, well, I mean, you should try to make your best effort. But we're not actually going to hold you to that.

Steve: Right, it's very open to interpretation.

FR. ROBERT: Yeah. So in other words...

Steve: And, I mean, Edward Snowden has shaken our confidence, I think, in an important way. And as a consequence we view what is being done on our behalf with a little more skepticism than we did before, when things were theoretical, but were assumed not to be happening. Now we keep seeing conspicuous, I think, evidence of what appears to be deliberate manipulation of equipment across the industry, all aimed at essentially, I mean, and this is illegal manipulation. Cisco didn't give anyone permission to crack into their routers using previously unknown SNMP flaws. Yet we have evidence that that's been going on. And probably from some government bodies.

FR. ROBERT: You know, two years back at Black Hat, Dan Geer, the keynote speaker, was saying how he thought that governments should be responsible for buying up zero days and then sharing them. They're the only entity that has the resources to be able to find these and then secure Internet infrastructure. But it seems more that the governments are buying up zero days and then using them.

Steve: Yeah.

FR. ROBERT: Yay. All right. You know what, let's go on to something that I consider to be a related topic because you just mentioned how our confidence in government organizations has really sunk to a low.

Steve: Eroded.

FR. ROBERT: Yes. I mean, "eroded" is being gentle. It was washed away completely. And even now, even the smallest mention of something that happens in the dark, or without oversight, just raises all sorts of red flags.

Steve: Well, even something that's being changed. It's like, well, do we really have to change anything? So, okay. This is a - the way I put the show notes together, it was as a bit of an acronym glossary. Because this is a little bit of acronym soup. The headline I put on this is "NTIA's contract with ICANN to handle IANA is expiring in 10 days."

So, okay. So we've never really on this show looked at the bureaucracy of the Internet. That is, you know, the behind-the-technology mechanisms which are important, but they're such a mess that I'd just rather talk about bits and bytes and protocol numbers, rather than the politics of it. But there is politics. So the IANA, that's the Internet-Assigned Numbers Authority, and of course that includes DNS. That's the group, the body which manages the DNS route and two top-level domains, INT and ARPA. It coordinates the global pool of IP and AS. That's the autonomous system numbers. Those are the numbers that large ISPs are given that BGP routers use for moving bulk data between top-level networks. And they maintain Internet protocol assignments. So we don't normally, you know, our normal use of the Internet only encounters that a little bit, that

is, the IANA work.

But down at the packet level, for example, an IP packet has a header field that identifies the protocol like UDP or TCP or ICMP that is carried by that IP wrapper packet. Well, those protocols are identified by a number. Who decides what that number is? Who sets that number? That's the job of the IANA. And so their job is crucial. The only thing that allows the level of interconnectivity that we have is an agreement about protocols. And you have to have an enumeration of stuff within the protocol, like which sub-protocol of IP you're talking about, and so on. So the Internet-Assigned Numbers Authority, like the name sounds, manages those sort of static definitions for the Internet.

Okay. Now, ICANN, I-C-A-N-N, is the Internet Corporation for Assigned Names and Numbers. Now, that's already today a multinational, multistakeholder body. It happens to be based here in Southern California, in Los Angeles. And it is composed of many member countries, including China and Russia. So it's already multinational in nature. And while the Internet has been growing and happening, ICANN has managed the IANA functions, which are technically its responsibility. So ICANN has been managing or providing these Internet-Assigned Numbers Authority functions under a contract with the Commerce Department's NTIA. That's the National Telecommunications and Information Administration. However, the United States has long made clear that it intended to eventually privatize the domain name system in order to, that is, sort of to release implicit ownership of it, in order to facilitate international participation in its management moving forward.

So the news here is that in 10 days, on September 30th, the NTIA, which has been the contractor under ICANN to perform the IANA services, intends to allow its contract with ICANN to expire, at which time ICANN will assume stewardship of the IANA's key technical functions. And this is only controversial in some corners, arguably with people who don't really understand what's going on. Most Internet experts and the major Internet companies - Apple and Google and their ilk - universally support this Internet governance transition because it counters the growing argument which repressive regimes can use to lobby for greater power over Internet governance, or even their own local Internet, by breaking off from the global Internet altogether. In other words, if it can be said that the Internet is a U.S.-controlled thing, then Putin over in Russia can say, we're not sharing this network with the United States that is controlling the whole thing. We're going to just sever ties and create our own Russian network that is disconnected from the global one.

FR. ROBERT: Steve, that is important. But that is really a PR thing, though. Because, I mean, oppressive governments are going to want to break off from the Internet anyway. So they want their own version.

Steve: They're going to want control.

FR. ROBERT: Right. But this makes it so that they can't publicly say, well, we're only doing this because the horrible, horrible empire that is the United States controls the Internet right now. We can't stand for that. This way we can say, no, it's under private control. So it's a private corporation, a multinational. Everybody has a stake in it. We are out. Now, okay. We are getting a little bit of a letter salad going on in the chatroom. So we know what ICANN is. We know what the IANA is. Where does ARIN fit in? So that's the American Registry for Internet Numbers. Where do they fit in, in the grand scheme of assigning IP addresses?

Steve: You know, in terms of a bureaucracy, I don't know who they report to. It feels like they would be another - technically ICANN is the overlord. ICANN is the entity that

has responsibility, which is why it was subcontracting the work that the IANA needed to do to the NTIA, which now is coming back to it. But again, you'd have to look at it like an org chart of bureaucracy of the Internet, which as I said I have shied away from as much as possible. But you're right. ARIN is another one of these bureaucratic but necessarily registries to maintain order.

FR. ROBERT: Right. And we've actually had them on This Week in Enterprise Tech, I think three or four times. Their president, John Curran, was talking about their most recent push for IPv6 and DNSSEC. So that's a good thing. I like that about the organization. But I want to talk a little bit about the transition we made between the last story and this story, when we were talking about the mistrust of government institution, and therefore that's why we're getting rid of the IANA.

But it should be noted that ICANN does have a bit of a checkered past. They had an at-large board member by the name of Karl Auerbach, who was a long-time member of the Interop team and a frequent guest of This Week in Enterprise Tech, who he wanted financial transparency for ICANN. He wanted them to publish a budget just like every other public organization on the planet, and they fought him tooth and nail. And so, Steve, I want to throw back to you, we in the United States tend not to trust the government, and we trust corporations, even though it could be argued that ICANN is less transparent about what it does and how it uses the resources at hand than the U.S. government.

Steve: Yeah, again, I have to plead ignorance about the individual specifics. I know that Esther Dyson was intimately involved with ICANN. She was a board member for some period of time. Her integrity is beyond question. But she also wasn't running the whole thing. And I've been a board member of various organizations where I've ended up resigning ultimately because I was frustrated with the way the whole entity operated. And politics.

FR. ROBERT: That's actually a very - that's a very, very common experience. I know a lot of people who have had some sort of involvement and just kind of threw their hands up and walked away and said, "I don't know how to navigate this organization. I guess it works. I have no idea how that is."

Steve: Where does the work actually get done? Who does anything here? Because all we're seeing is reports and memos flying back and forth, yeah.

FR. ROBERT: Yeah, yeah.

Steve: Well, you mentioned DNSSEC. And I am so bullish on, as the listeners of this podcast know, on the promise of DNSSEC. That is, if we, or when we, because we're clearly moving in that direction, when we have a securely verifiable Internet-scale directory, what we can do is, I mean, it's hard to think of anything you can't do. For example, the existing troubled certificate authority structure, which we bemoan just because it has become so bloated, and because we're now trusting many, many hundreds of certificate authorities who say they're going to act on our behalf and correctly, but we keep finding instances, discovering instances where they have failed in that mission.

And so, for example, there is no reason that a server's certificate has to work the way it does, where we trust a certificate authority to sign our public key to assert its validity on our behalf. After all, we created the key pair. We have the private key. We give them the public key. In some cases pay, but now with Let's Encrypt it's an automated process. Get a signature which asserts, if nothing more, that we're who we say we are, or that we do

control the domain that this certificate is providing security for.

So if we had DNSSEC, there's already a protocol - I think it's DAME, or DANE. I don't remember. I think it's DANE, D-A-N-E - which is a DNSSEC-based certificate solution. That is, you ask DNS for the private key of your site, rather than needing to accept its signature from a certificate authority. And that's just one example. I mean, just the mind boggles when we could have a scalable, controlled, caching, truly secure, global directory that can contain all manner of different information.

So I'm, you know, it's taking a while to get there. Comcast got their root signed in 2012. Google's DNS has had theirs signed since 2013. I've looked at it for GRC out of curiosity. But it's like, okay, well, there's nothing I really, at this point, that I want to secure from GRC's DNS. And at the moment the tools are awkward to use. I think Hover, which is the name server that I'm using, or the registrar, I think they can provide DNSSEC. But I'm also playing with my DNS records often. In fact, later in the story, in today's podcast, I'll explain how I had to do that a part of this cross-site scripting issue. And that would have broken the signing of my domain, and it would have been a problem because I would have had to then get it re-signed. So we're still in this awkward stage.

But the point of all this is that there is an intriguing event that is on the horizon, and that is that the famous DNSSEC root zone signing key, as it's called, is being changed for the first time. There are no known problems. It's just regarded as good security policy to rotate your keys periodically, especially for something as high value as the DNSSEC root zone signing key. And in the process they are doubling its size. So they're also, as we always do, moving to larger size keys as the ability to crack smaller keys gets worrisomely close. No one thinks that the 1024-bit key today is vulnerable. So now's a good time. As has been said in some of the coverage of this, there's no emergency. It's a quiet time. That's when you want to do this is because it's a big change.

So they're going to go from a 1024-bit key to a 2048-bit key. And the process is almost mystical. I mean, it's weird how, well, I mean, you could imagine the value of this key. If the private side of the key pair were to escape absolute control, it would allow any entity, a nation-state or anyone, to essentially spoof DNS with no ability to detect it because it would have been signed properly. It would be signed by a private key that everyone holding the matching public key would authenticate.

FR. ROBERT: Which, Steve, I think means, if the secret key were to escape, it would essentially become the DNS that we're all using right now.

Steve: Well, I mean, that's a very good point. There's no encryption by default on DNS. It's a UDP packet. Everybody can see it go out, everybody can see it come back, and your computer implicitly trusts it. It's just, okay, this is the IP for the domain I'm going to. And notice that, when we're talking about security vulnerabilities, for example like how do we know if a TLS certificate is valid, it's does it match the domain name. And when we see the name in the browser, it's like, oh, okay.

But there's an implicit assumption that the IP address behind that domain name in the browser's URL is the correct IP. And the point is that crucial in the exploitation of this is spoofing DNS. And we don't talk about it as much as we should. But it is an Achilles heel. In fact, as we secure other bits, this is the one which sort of is beginning to stick out as, boy, we ought to get - let's get moving on locking down DNS because we've been running around doing all the easy bits, and we're kind of running out of those because everything's getting pretty secure. But DNS, also, we have to trust that domain name-to-IP address mapping.

FR. ROBERT: It's not just that it's insecure, it's that it's scary insecure. We were explaining this to a group of people who dropped by Interop Labs a few years back. And the easiest way to explain it was currently DNS is a yelling contest. If you can yell louder than the other machines around you that you are the domain that you want to spoof, it belongs to you. That's how it works. Which is why we were making a big push for DNSSEC and DNS DANE. And actually we did talk with Cricket Liu from Infoblox about DNS DANE, and he was saying it's not just that it's a more secure system, it's that it makes it easier for people to actually start using certificates without using CAs that can be compromised, that can get revoked, that just might be bad actors, as you talked about on Security Now!. We've already run into at least two that they weren't even pretending to be issuing valid certificates, and yet they were still in the trusted CA list.

Steve: Right.

FR. ROBERT: And actually the real part I wanted to bring out about this, and this is what I find so beautiful about the story, is when DNSSEC was first cooked up, they built in this function, this ability to upgrade the key, into the spec, which is huge. That's important. It means that it's a forward-looking standard. They realize that we are at some point going to upgrade the encryption level from 1028 to 2048, maybe to 4096. And we don't want to break it by upgrading security. So this is the first real test of us saying, okay, well, upgrade it, and let's see what happens to those who are currently using DNSSEC.

Steve: So I love this description of the procedure. ICANN incorporates some extraordinary security measures and considers its potential threats as everything up to nation states. At its quarterly ceremonies, so-called "crypto officers" from all over the world congregate in one of the key management facilities after passing multiple layers of physical and digital security. Next month, this coming October, in a hyper-secure key management facility on the U.S. East Coast, ICANN will generate a new cryptographic key pair. One half of that pair is private and will be kept super securely by ICANN. The other is public. Internet service providers, hardware manufacturers, Linux and other OS developers, anything and anyone who needs to verify future DNSSEC records, needs to have the updated DNSSEC public key.

Then, in the first quarter of next year, 2017, two employees will transport a copy of the encrypted key files on a smartcard over to another facility on the West Coast, using regular commercial transport. I thought it was interesting that they tossed that in. I guess, what, instead of a military jet or something. Eventually the public part - or I guess a private executive jet. Anyway, yeah, standard commercial transport in full public view, but super encrypted. Eventually the public part of the key pair will then be distributed to other organizations.

And then, finally, the whole switchover will take around two years from start to finish. The new key will appear in the DNS for the first time on July 11th, 2017. So next summer, next July 11th. And then in October of 2017 the new key will be used for making signatures. So I imagine that initially it'll be cross-signed, so you'll be able to continue using your previous 1024-bit key, yet the root will also be signed with a new 24-bit key. And then any new equipment, and of course OSes will be updated, and anything using and depending upon DNSSEC will need to - there'll be some grace period, which is probably where this two years comes in, where both keys, both the old and the new, either can be used. And then eventually it will be necessary to retire the 1024-bit key.

And I'm sure they'll be able to get statistics and things and watch to see the hopefully quickly diminishing use of the 1024-bit key. If history is any lesson, there'll have to be a threat of, okay, this key is being removed on this date. Everybody, you had two years,

get your act together, update to the public key. And in embedded appliances and devices, that can be a challenge.

FR. ROBERT: You know, Steve, I think they're missing out on a great opportunity here. And if anyone at ICANN is listening, you could take advantage of my priestly experience because I think this meeting should have seven figures that are robed in brown robes so you can't see their faces. There's a big smartcard in the middle of the room, shrouded in fog. And it lights up as the key is copied to it, and it's broken into seven pieces and sent to various parts of the world.

Steve: I think it's very much like that, Robert.

FR. ROBERT: it's all about the mystic experience.

Steve: I've seen some of the videos of these ceremonies, and there's a lot - they're very serious.

FR. ROBERT: I've seen it, and it is like that, except they don't have the robes and the fog and the weird light and the Celtic music in the background. See, they're missing an opportunity, Steve.

Steve: It might freak some people out. I'm not sure.

FR. ROBERT: This is how the Internet works.

Steve: Yeah. So we've been following Google's efforts for years to strengthen the security of the Internet. And the most controversial one I would argue was their decision to move up the already planned sunsetting of SHA-1 hashed and signed certificates. But that happened. And starting at the beginning of this year, no CA is issuing an SHA-1 hash signed certificate. So, okay, that's behind us. Now Google is turning their attention to the problem that, while HTTPS connections are shown as explicitly secure, non-HTTPS connections are just kind of vanilla. That is, they don't say anything.

And Google's concern is that, in the long term, once we accept the idea that HTTPS, protected with TLS privacy and authentication, is the de facto web protocol, not HTTP - remember, traditionally, HTTP was the de facto protocol. And even as recently as a few years ago, major sites would switch their users to HTTPS, only while transacting privileged information. And, controversially, they would leave their cookies, which were maintaining session, when the browser switched back to HTTP, allowing things like the Firesheep attack, which made it trivial to impersonate people in any open WiFi environment. So Google is saying, okay, our next push is we're going to actively discriminate against non-HTTPS connections.

So in their blog post they wrote: "To help users" - oh, by the way, this is starting January 2017, so the beginning of next year, only a few months away. "To help users browse the web safely, Chrome indicates connection security with an icon in the address bar. Historically, Chrome has not explicitly labeled HTTP connections as nonsecure. Beginning in January 2017 with Chrome 56, we'll mark HTTP pages that collect passwords or credit cards as nonsecure, as part of a long-term plan to mark all HTTP sites as nonsecure. Chrome currently indicates," they write, "HTTP connections with a neutral indicator. This doesn't reflect the true lack of security for HTTP connections." In other words, you can sort of see the thinking now. It's like, it's not that this connection is secure, oh goody. It's wait a minute, why is this one not secure? That's bad.

So they say: "When you load a website over HTTP, someone else on the network can look at or modify the site before it gets to you. A substantial portion of web traffic has transitioned to HTTPS so far, and HTTPS usage is consistently increasing. We recently hit a milestone with more than half of Chrome desktop page loads now served over HTTPS. In addition, since the time we released our HTTPS report in February, 12 more of the top 100 websites have changed their serving default from HTTP to HTTPS. Studies show that users do not perceive the lack of a 'secure' icon as a warning, but also that users become blind to warnings that occur too frequently.

"Our plan to label HTTP sites more clearly and accurately as nonsecure will take place in gradual steps, based on increasingly stringent criteria. Starting January 2017, Chrome 56 will label HTTP pages with password or credit card form fields as 'not secure,' given their particularly sensitive nature. In following releases we will continue to extend HTTP warnings, for example, by labeling HTTP pages as 'not secure,' initially in Incognito mode, where users may have higher expectations of privacy. Eventually, we plan to label all HTTP pages as nonsecure, and change the HTTP security indicator to the red triangle that we use for broken HTTPS."

So, you know, any time someone uses their power, as Google is, it creates some controversy. And no one likes being told what to do. I imagine there will be people who for some reason HTTPS is a problem, is not practical. Something prevents them from doing it. I don't know what, but something. So the dicey part here is that Chrome is going to be scaring anybody who uses those servers. Now, we could argue, well, they deserve to be scared, or this needs to put pressure on that problem to get it fixed. Which may be the case.

Ultimately, everyone knows I'm bullish. GRC has been HSTS, Strict Transport Secure, for years, and known by all the browsers that way. And ultimately I think this is clearly where we need to go. But Chrome now has the lion's share of the browser market and is taking advantage of that in order to push security forward. Which, ultimately, I mean, I think the process is always painful, like what we went through with SHA-1 certs. But once you're past it, it's like, okay, this is better now. And every lesson that we have been taught by this industry is, I mean, okay, just say IPv4, is we're not moving. We're not doing it unless something makes us.

FR. ROBERT: You know, Steve, we've got a couple of folks in the chatroom. We've got Klaatu and Eric Duckman, who are saying thing like, well, you know, some sites don't need HTTPS. And you've got Eric Duckman saying, oh, what, so everyone's supposed to get a cert now? And I understand that frustration. But at the same time, no, absolutely not. The argument that only some things need to be encrypted is the wrong way to think about it. I think both of us are in agreement that you encrypt everything because you don't want people to know what's important by what's encrypted and what's not. It should be equally for them to break into my WordPress session where I'm reading a blog as it is for them to break into my session that's transferring financial information. And as far as...

Steve: We have a perfect, just interrupt you for one second. A perfect example of that is what law enforcement has called the "going dark" problem. Well, how do they know it's going dark if they weren't looking for the light?

FR. ROBERT: Precisely. Precisely. And as far as certificates are concerned, I mean, this is the DNS DANE story. We are about to get a system based off of DNSSEC, using the same key from DNSSEC, that will allow us to have certificates at a much lower price - read free - that are more secure and more easy to revoke when they go bad. So I like that Google's doing this. They're one of the only entities that has the muscle to shame sites

into doing what they should have done a long time ago. I mean, the Internet should be end-to-end encrypted.

Steve: Yes. And, well, it is an enduring problem that security is an afterthought for the Internet. I mean, we need to cut them a lot of slack because they were amazed when a packet made it between two of their processors on Day One, back when it was the Arpanet, and it's like, oh, my god, you mean this actually works? And so no one could have foreseen what was going to happen. I mean, I'm still complaining about the fact that we have to say http://. Who was this designed for? This wasn't designed for people. So, yeah, we're dragging a lot of legacy behind.

But I completely agree with you. I think that - and don't forget, too, Let's Encrypt does automate and make easy and has been a huge success with exponential growth in the number of certificates that they're issuing, providing trusted certificates at no cost. And from what I've seen, in people who are complaining, I mean, I get it. There are technical reasons. It's like they're on a shared hosting provider, and CPANEL doesn't yet support Let's Encrypt. Well, okay, it needs to. Fix that. And then those sites can use encryption, and everybody'll be happy. And so, yeah, again, there's going to be some pain. But I think where Google is taking us is the right direction. Or pushing us.

FR. ROBERT: Yeah. And I'm okay with some short-term pain for some very long-term gains. And what we are talking about is a long-term gain. I think we can all agree that everything encrypted is a better way to do business and to live on the Internet. And, yeah, it seems a little heavy-handed. I get it. I get it, folks. Some people don't like the fact that Google is using its huge market share to be able to force this change. But someone has to force the change because otherwise it just - it won't happen.

Steve: Yeah. On this podcast we're constantly talking about the reticence to change, the inertia against change. It's significant.

FR. ROBERT: Right. All right, Steve, here's a story that I know some people are tired of hearing about, but it's important. And it's important because it deals with the devices that we use each and every day and sometimes assume that they are secure, even though we know that they're not. Tell me a little bit more about the iPhone 5c NAND saga.

Steve: So what was interesting about this, I've got the links to the detailed PDF in the show notes, and there's no need to dig in. But for anyone who's interested in, step by step, how was this done, this research is just fabulous. Lots of photos showing what this engineer went through in order to pull this off. And we covered the concept of this when it was brought up, and that was you have a phone which is locked, and the person will not or cannot divulge the information. Maybe they're no longer alive. But the government or law enforcement is desperate to get into the phone.

So the problem is that, as a security measure, proper security, Apple has a lockout after X number of wrong guesses at the security code. And so there have been hacks in the past that we've discussed where guessing the code, seeing that it's wrong, and then immediately powering down the phone before it has the chance to update the nonvolatile counter in memory was an example. And that worked for a while. You would guess it, see that it didn't work, quickly power down the phone, and that guess wouldn't count against your X number of strikes. However, Apple fixed that, so that that no longer worked. And the 5c has that advanced technology where you can no longer perform this fast power shutdown/reboot in order to prevent the fact of the guess failing from being written to nonvolatile memory.

So any engineer looking at this says, oh, well, then what we need to do is clone that memory before we even start guessing. Make a copy of it. And then we'll make as many guesses as we can before the phone gets upset and then copy from the clone back to the working memory, which essentially resets the count, as if we had never made any guesses. But we obviously don't repeat those guesses. We make new guesses until we decide it's no longer safe, and we repeat the process. So the theoretical concept of cloning the iPhone's memory, that is, with a state where guesses were allowed, and then resetting the state, it's an end around all of the other security measures. But it was only until now theoretical. We've talked about it on this podcast. It's been proposed. No one had done it. It has now been done.

So the abstract of this paper, I'll just read this from the top, says: "This paper is a short summary" - but it's actually not, it's a wonderfully detailed summary - "of a real-world mirroring attack on the Apple iPhone 5c passcode retry counter under iOS9. This was achieved by desoldering the NAND Flash chip of a sample phone in order to physically access its connection to the SoC (System on a Chip) and partially reverse engineering its proprietary bus protocol. The process does not require any expensive and sophisticated equipment." Although having looked at the pictures, it's not for the faint of heart, either. You need a steady hand.

"All needed parts are low cost and were obtained from local electronics distributors. By using the described and successful hardware mirroring process, it was possible to bypass the limit on passcode retry attempts. This is the first public demonstration of the working prototype and the real hardware mirroring process for iPhone 5c. Although the process can be improved, it is still a successful proof-of-concept project. Knowledge of the possibility of mirroring will definitely help in designing systems with better protection." And in fact I will propose some of that before we're done. "Also," they write, "some reliability issues related to the NAND memory allocation in iPhone 5c are revealed. Some future research directions are outlined in this paper, and several possible countermeasures are suggested. We show that claims that iPhone 5c NAND mirroring was infeasible were ill-advised."

Now, I went through the paper, and I am really impressed with this guy. And so I just want to share a little bit of some of the inner techiness that demonstrates that Apple was aware of this and proactively worked to thwart it in a sneaky way. They're going to have to up their game with the next round of hardware. I don't know what the 6 or the 7 phones look like. But so in one part of the paper, where they get into actually making this happen, they say: "The process of cloning involves creating a fully working copy of the NAND Flash memory chip. However, as it was already mentioned in the previous section, simply copying the 8GB information from the original chip into another identical chip taken from another iPhone 5c does not give the desired result, and the iPhone does not boot." Meaning cloning was not enough.

"Some additional research was undertaken to figure out why simple copying doesn't work. For that same model of the NAND Flash chip was programmed with the data from the original chip, and then the communication was" - oh, I'm sorry. "For that" - there should have been a comma there - "the same model of the NAND Flash chip was programmed with the data from the original chip, and then the communication was analyzed with both an oscilloscope and a logic analyzer. First, some pages were accessed from addresses outside the normal 16GB address space. For example, instead of reading and writing to the block" - and they give an example, 0x00041Axx - "the CPU was accessing the block 0x00841Axx. Although such accesses wrap around and are mapped back into the 0x00041Axx block, the page numbers were different, as well as the status of those pages." Then they have a figure that shows a table of a bunch of those, just to detail it.

"Secondly, some irregularities were found in the" - irregularities - "were found in the communication prior to the access of those hidden pages. Although the data transfer during the access is performed in the SDR mode at 17 MHz, while the configuration commands use the even slower speed of 1 MHz, some data inside the commands are smuggled at an astonishing rate of 256 MBps in DDR3 mode. Also, a dummy value for data bit 7 was introduced for the period of 23 nanoseconds. Given that the data setup time in those transfers is less than one nanosecond, there is a very high chance that such information would be overlooked by most would-be attackers."

So decoding that, this means that Apple deliberately - they understood that there was a vulnerability when the NAND chip that stored the keys to the kingdom in many ways was physically separate from the SoC, from the processor, the System on a Chip. That is, there were communication lines connecting them. That exposed a bus which the processor used to communicate with the memory, which was then subject to reverse-engineering. So knowing that, they went to tremendous effort, essentially, to hide critical details of what was going on, like in otherwise much shorter and faster little bursts of data that you wouldn't see unless you really, really looked closely enough, down to a resolution of one billionth of a second.

So they continue: "After those findings, the implementation of the communication protocol was amended in the test board." Because basically they, like, through this process, built a number of different jigs in order to hold these bits with this poor-looking red case iPhone 5c with its back cut open and wires coming out of it, like a zombie. "Then the data-mirroring software was modified to include cloning the hidden pages. As a result, the newly created clone of the original NAND chip was fully functional in the iPhone 5c. It was then tested with six incorrect passcode attempts before replacing it with the original chip. After the boot process, it was possible to enter the incorrect passcodes again six times until the one-minute delay was introduced. This fully proved the correctness of the hardware NAND mirroring attack on iPhone 5c."

And then, to put this in real-world terms, they write: "Because there's no limitation on the number of such NAND clones, they can be created in advance and restored in parallel when one of them is being used for passcode testing." And I would argue that you could also do deeper reverse-engineering in order just to zap whatever little bits are being changed and put those back by comparing pre- and post. "This way, it only requires 45 seconds for six passcode attempts. For four-digit passcodes, the maximum attack time would be" - and then they do the math - "75,000 seconds, or about 20 hours. For six-digit passcodes, this time will increase to about three months, which in some cases might be acceptable." So I just - there was too much detail not to share. It's just wonderful stuff.

FR. ROBERT: Let me ask you, because the trick that they did with accessing address spaces outside of the standard 16MB block, was that randomized? So I know in order to access Block A, you actually had to access Block X, which referred back to Block A.

Steve: Right.

FR. ROBERT: Does that change, or would that be the same on every iPhone? Is it like address space randomization that you would get in a Windows box?

Steve: From something else they said elsewhere, and I didn't have it in my notes, they indicated that it was swapped-out pages.

FR. ROBERT: Okay.

Steve: So I think that that's fatigue. It's wear leveling of the NAND. So for whatever reason, they needed to clone that, that is, like they had to develop an awareness of the physical location, not just the logically addressable location, and incorporate that. And again, that may have been by design. That feels like it was another confounding factor that Apple added, just thinking, ah, this will get them.

FR. ROBERT: Well, it's just that one little extra bit where, well, if they figure out where the actual, what the address spaces are, this will throw them off because they actually have to access a different address space in order to access the address space that they want. But, I mean, it sounds like then it's not random. It's just a reassignment.

Steve: Yes.

FR. ROBERT: Hmm.

Steve: Yeah. And they mapped that out and then showed us a table of where the reassignments were. And once they incorporated that into their clone, then the 5c didn't know there was any difference. It thought it was talking to the original chip, although it was talking to a true copy.

FR. ROBERT: Now, Steve, Creamy Corn Cob in the chatroom has an interesting question. He wants to know what is the actual encryption that takes place between the SoC and the NAND itself? Because the NAND, it's not specialized NAND. They're just - they've got it from Hynix, and then they're running their own encryption on top of it. So is there encryption between the SoC and the NAND?

Steve: I don't know that - to me, from the description, it does not sound like a standard part. They were playing some games at the electrical protocol level which would violate the specifications of a standard NAND chip. So my guess is that Apple had these things custom made. They had a little, like they tweaked the NAND controller in a way that customized it for their use. And the trick of this is that, by cloning the memory, it was not necessary to decrypt the protocol.

FR. ROBERT: Right.

Steve: So, for example, both ends could have a private key, and all you would see is gibberish going across the bus because you would never see it in plaintext. But if you are able, without knowing anything about what it means, if you just copy it and use its behavior, rather than its information, then you get a shortcut.

FR. ROBERT: Well, I mean, that in itself is fascinating, that iPhone NAND is not regular NAND. It's not what every other manufacturer is buying.

Steve: If I read, yes, if I read what they wrote, the fact that they were sending a little burst of data, essentially hiding a small packet of information, well, that's not - no spec reads like that. That's special.

FR. ROBERT: That's very special. And you know what, there are some people in the chatroom who are joking that this is way too propellerhead. And, yeah, you know what, it's actually, it's getting over my propellerhead, too, because anytime someone talks about how they hooked up an oscilloscope, and they were able to figure out what it was doing just by looking at the waveforms, I'm nowhere near that. That's a lot of skill. That's a lot of knowledge. Wait, are you - wait, are you going to bust out your oscilloscope?

Steve: Well, no, I mean, it's - it's right here.

FR. ROBERT: I can build an oscilloscope, but I don't know how to use it. I mean, because all they're looking at is they're looking at the rising and the falling edges of the clock; right? They're looking at, well, what are they receiving back when they send a signal to the individual cells.

Steve: Correct. Although these things all have a lot of lines. One of the things that has happened is we've gone to a much more serial world than we used to have. Look at the slots on our motherboards. They're little itty-bitty slots. Yet it's like faster than when we had huge long slots. So the idea is we're going to serialize things in order to bring pin counts down, to minimize bus size, and also you need driver electronics at each end in order to make this all happen faster. It's much better if you have only a couple of them than if you have, like, 64 of them. Move the 64 bits down one lane, rather than all in parallel across 64. It's a huge savings in chip size, in pin count, and in bus complexity. Anyway, I understand that this is tech-y, but I thought it was fun to basically rip the cover off, so to speak, of an actual exploit, a physical exploit against a phone, which works.

FR. ROBERT: Yes, yes.

Steve: And if anyone wants to see, again, the link is in the show notes. The pictures are just captivating. The guys, they really documented it beautifully.

FR. ROBERT: That's a lot of work. And I understand, yeah, everything that's there I can build, or I probably have in my lab already. But to say that this is something that you can do easily, no.

Steve: Well, and there's no instruction manual. You know, this is all - first of all, I don't even know how you unsolder a BGA, a ball grid array, NAND. Because, I mean, it's got little solder dots on the bottom of it. It's like, and so it's done once, and it's there for good. So I don't even know how this guy got it off of the back plane.

FR. ROBERT: Well, I have the device to make that happen, but it would destroy the PCB. So again, don't try this at home, folks. Leave it to the professionals and the professional amateurs.

Steve: So I had three bits of miscellany. I got a tweet from a Will D., who said: "@SGgrc My school's DNS server was down this morning, but I remembered your DNS Benchmark tool. I am back online and faster than before. Thanks." So I did just want to remind people that that Windows app that runs under Wine, so if you've got Wine emulation in Linux you could use it there, too, or even on your Mac, it's become the industry standard DNS Benchmark. Google had something for a while, but it sort of fell by the wayside. And this little bit of freeware that I wrote, I think it was like in '06, really does the job. And so I just wanted to put it back on people's radar.

Remember we spend a lot of time talking about DNS in various contexts today. You have to, our browsers have to have the IP address, the physical Internet protocol address, of all the sites that they're accessing data from. And today's web pages are covered with resources coming from just a huge scattering of servers. If your local machine doesn't already know what the IPs are, it's got to ask somebody. And it's got to wait until it gets an answer. So if you're not using fast DNS, your entire experience of the Internet is hampered by that because that's the first thing that happens is that those ASCII names get mapped into an IP address.

FR. ROBERT: Right. Actually, a fan of the network sent me a little plugin that he's creating for Synology's router, their RT1900ac. It's their SRM operating system. It allows for plugins, just like you would have on one of their NASes. And what he created was a service that you could put an app on your desktop. And every time you do any sort of search, it breaks down how much time it took for the query to take place versus how much time it took for the actual content you requested to be downloaded. And it's fascinating to have that running up on the desktop, and you get to see, wow, I spent more time waiting for DNS than it took to download all this text and pictures. And it does kind of change your appreciation for good DNS.

Steve: Well, and you know that DNS sometimes doesn't get the attention from ISPs that it deserves. It's not sexy. It's sort of, okay, we have to provide a DNS server. Maybe they don't give it the fastest one because they want their website to run fast, but DNS, eh. And no one really complains when it's slow. So you may - your ISP, it's sort of like the unloved stepchild of Internet service providers. It's like, yeah, well, yeah, we have it. And so it's worth making sure that you're not being slowed down because your ISP just doesn't care to give you good DNS. You could certainly change it, take responsibility and find something faster. And GRC's little DNS Benchmark tells you, helps you rate all of your alternatives.

FR. ROBERT: Steve, you run your own DNS; right?

Steve: Yeah.

FR. ROBERT: And I used to, and then two moves ago I decided I just didn't want to bring the box with me anymore. And so I've used Google 8.8.8.8 and OpenDNS at, what is it, 208.67.222.222 and 220.220.

Steve: That's right.

FR. ROBERT: If you had to use a third party, what would be your favorites?

Steve: I'm with you. I still like 4.2.2.1.

FR. ROBERT: Ah, yes, of course.

Steve: And of course that's a Level 3 server, and I'm hosted by Level 3, so that works. And 8.8.8.8 is another choice. What I do is one of my FreeBSD boxes, the one that runs the GRC newsgroups, the NNTP groups, that also runs BIND. And mine is a master for two Level 3 slaves. So when they need to check in and get an updated zone, they ask for that from me. And so the public sees these very strong, big iron DNS servers that can handle any load, that are super well connected, that are right on the Internet backbone for high performance. And then, when I make changes, I'm able to send them an update command that causes them to immediately refetch an updated zone file. So it's sort of the best of both worlds. I don't actually myself handle any public traffic. They're my front servers for me. But mine has the master records that they pull from as needed.

FR. ROBERT: Let's move to a different type of DNS question now. People always ask what I use to host my services. And I'm lucky enough to actually still have access. I have a /8 that I can borrow from every once in a while in IPv4 space, which is amazing.

Steve: Oooh.

FR. ROBERT: Yeah, I know. So I still like using my quads. But some people want to use a

third-party solution.

Steve: Well, we talked last week about the whole dynamic DNS issue. We were talking about the idea - it was a Q&A episode. And one of the people asked, do you still recommend proXPN as a VPN provider? And I explained that there were certainly applications where using any of the major VPN hosts made sense. And you wanted one that was not going to be logging and tracking your traffic, just for prudence. And we know that proXPN doesn't. So it's as good as any others.

But the position I took was that the problem with that is that, very much like a Tor exit node, a VPN server will be emitting traffic onto the 'Net from all the people using the service. And that just has to be a tasty spot for any government agencies that are interested in what it is that people are doing over their VPNs. Why do they need more privacy, for example, or need to appear to be in a different location than they actually are? So the nature of its concentration of traffic is a little off-putting in today's world.

So what we're seeing now are good options for running an OpenVPN server on your own router at home. We've talked about that there's a project called PiVPN, which takes a Raspberry Pi and, using a shell script, just completely automates the process of establishing and configuring an OpenVPN server for $35 in a little Raspberry Pi, which you then plug into a port on just a generic router, do a little bit of port mapping so you can get to it from the outside, and now you've got an OpenVPN endpoint. And then the alternative is, if you have something like pfSense running on a router, or the router that's now the show favorite is the Ubiquiti EdgeRouter, all of those are FreeBSD based and have OpenVPN servers built into them.

So the problem, of course, though, is that ISPs are not obligated to give their typical home cable or DSL subscribers a static IP. That requires dynamic DNS. That means we need some service in a fixed location on the public Internet that we can refer to, that has been updated by our device about its current IP. And so I talked about last week that Google's domain name services, for a dollar a month, $12 a year, would provide domain name and provide dynamic DNS as part of the bundle. There's another registrar, Namecheap, that also offers dynamic DNS. Hover, my chosen registrar, does not.

But Kevin Garman tweeted me about a service that I wasn't aware of, and I've not used it. But I spent some time poking around, and I'm impressed. It just has the right feel to it. It's called, unfortunately, Afraid.org, A-F-R-A-I-D dot org. It's FreeBSD based. His slogan near the top of the page says: "Why is it free? It's quite simple. We wanted a challenge. That's it."

So what this lets you do is they're a DNS server and essentially a DNS host. So you can create yourname.afraid.com, that is, whatever you want .afraid.com, and point it at your home IP, and then use the dynamic DNS support in your router to keep it updated if your home IP changes. And they offer a bunch of other services. So I just wanted to sort of put it on our listeners' radar, Afraid.org, for free DNS, including dynamic DNS. And I just, as I was poking around there, I just got a good feeling about the place. It looks like the right kind of deal.

FR. ROBERT: Yeah. I still have a bad experience that I remember every time someone asks me about dynamic DNS. I was living with a Jesuit who - he wanted to host his own little web server, so I showed him how to make a web server, and then he said he was going to use dynamic DNS. I'm like, okay, obviously he knows what he's doing. He didn't. And all he did was he opened up a DMZ port, the entire - all the ports to the box. And so within 24 hours I started seeing this weird traffic running over our network. And I go over, I'm like, yeah, that was one of the fastest ownings I've ever seen. Congratulations.

So, yeah, if you're going to do this, just make sure you understand port forwarding and know what services you actually want to run. And then what I would suggest is you run ShieldsUP against your newly created dynamic address, just to make sure it's only listening on the ports you want.

Steve: Yup, exactly.

FR. ROBERT: All right. That's good to know. Afraid.org.

Steve: Yes, Afraid.org. Check it out, if it sounds like it might be interesting to people. Again, it looks like a straight-up good deal. I feel good about it, looking at it.

Simon Zerafa, who's a friend of the show and frequently sends me valuable links and tips and oftentimes funny O'Reilly covers, he sent me something he wanted me to share. He said: "Hi, Steve. Can I please request a PSA?" Which I thought, a PSA, okay, that's a - I don't know. I have to figure that...

FR. ROBERT: Public service announcement.

Steve: Public service announcement, okay, for this week's show. I thought maybe a personal shout-out or something? I didn't know what. Anyway, public service announcement, thank you, for this week's show. He says: "My wife had a serious medical emergency yesterday, and we hadn't filled in the Emergency Contact information on the iOS Health app. As a result, the emergency services were delayed in contacting me. Can we ask listeners with iOS devices to fill that info in and enable it on their lock screens? It should make it easier in the future for emergency services to contact people and won't require them to enter their iOS PIN code to access their emergency contact details."

So I thought that was a great tip. It hadn't occurred to me. It's the kind of thing you never think to do, but once you wish you had, it's often too late. So now would be a good time to do that. It makes a lot of sense. Apparently something happened to his wife such that she was unable to provide her contact information. They had her phone, but they couldn't get into it in order to know who to contact. So thank you, Simon. That's a great tip.

FR. ROBERT: I just realized that your smart device is your new medical ID bracelet, the little thing that you used to carry. I mean, I used to carry one that would say what you're allergic to and who I should contact.

Steve: Right.

FR. ROBERT: I haven't done that in over two decades because now I just carry my mobile device. And it does have my emergency contact info on the front locks page.

Steve: Nice. Nice.

FR. ROBERT: You know what, Steve, I can't do an episode of Security Now! unless I hear about SpinRite.

Steve: Well, believe me, you're not subjected to them every single week, like all of our listeners. I just have two short notes that are kind of fun. Michael Nation in Michigan referred to me as "Steverino," and he said, "Apologies to Steve Allen." He said, "I run SpinRite on all my drives all the time, and nothing ever goes wrong. I'm disappointed. If I'm paying for health insurance, I expect, well, to get sick." And so I thought, okay.

And then I thought, okay, wait. To apply the health insurance analogy, it would be more like using RAID to recover from drive problems. Using SpinRite is more like taking an adequate level of Vitamin D to prevent any drive from having problems in the first place. So insurance doesn't keep you well. SpinRite does keep your drives well. So I think that's even better than having something that is able to recover from problems or cover you if there are problems. Just don't have them in the first place.

And then an anonymous person shot me a note with the subject "SpinRite rescues a stuck Windows 10 update." And that's the first I'd heard of that happening. He said, "My Windows 10 v1607 patches got stuck today, and this isn't the first time in all, although it is the first time for a v1607 update. Retrying didn't and doesn't help. So usually I would revert using system restore and try re-updating again. But this time I tried SpinRite - and it worked."

So this is sort of a reminder that drives are, they continue to be, even SSDs we are finding, arguably the least reliable component in the system. Everything else is just - it's a lot more solid. Maybe we're demanding much less of it. Certainly the bit density of mass storage has been subjected to huge competitive pressure for decades now, with the consequence being we all see reports about drives not lasting as long as they used to, manufacturers creeping the warranties down over time, just sort of hoping no one will notice that they went from a five-year to a three-year warranty, and so forth, because they're under pressure.

I'm happy that the mechanical spinning drives are staying alive, although it looks like SpinRite is going to be able to provide a solution for non-spinning mass storage, as well, based on experience. But for what it's worth, when something doesn't seem right, just run SpinRite on the system. And a surprising number of times it actually was a small mass storage glitch that wasn't revealing itself to be that because they often don't. There's something wrong, but who knows what?

FR. ROBERT: You know, Steve, I just realized what I want out of a future version of SpinRite. I want, again, your elegant assembly code, but something that runs as a timed script in either Windows or OS X, specifically for my SSDs. I want it to do garbage collection at 4:00 o'clock in the morning when I'm not awake, just because every time I start to slow down, I will pull out my copy of SpinRite. And is it Level 1?

Steve: Level 2.

FR. ROBERT: Level 2, Level 2, where it's just - it's basically going to the drive. It's not using up any of my writes unless it needs to actually destroy something. And it's just looking at what garbage collection has not yet been done. I tell you, you've saved a couple of my Samsung SSDs from the brink of death, and you've definitely improved my performance. So personally I thank you.

Steve: Well, thank you. And we had a story just last week about exactly that. Some guy's machine was just not running well and fast, and it was an SSD-based system. Ran SpinRite across it, and it was snappy as ever. So there again, even solid state memory. Unfortunately, the marketing pressures have pushed the density to the point where it's good enough, but there just isn't any tolerance. The old engineering margin has just been squeezed out. So, I mean, it's inevitable, but it's unfortunate.

FR. ROBERT: Right, right. And actually...

Steve: Good news is we have a solution.

FR. ROBERT: If you could get a version of SpinRite that runs on my Android phone because I'm pretty sure that memory has not been leveled at all. It's horrible. All right, Steve. Big story. Big story for us and for the people in the chatroom and the long-time listeners of Security Now!. They've followed this saga a little bit. We all know that a while back you raised the alarm that there might be some weirdness going on at GRC.com, maybe some cross-site scripting. And you've put on the propeller hat and figured out exactly what was happening.

Steve: Well, so what happened was a couple weeks ago someone tweeted me, someone using the Twitter handle @tbmnull said just - it was a short tweet - "XSS vulnerability found at GRC," and then a link to a site called OpenBugBounty.org, www.openbugbounty.org, which is where this person was apparently very active because I looked at the cross-site scripting problems he had reported, and they are all over the place.

So he didn't single me out. I think he's a white hat hacker who's decided to take it on himself, take on the mission of finding vulnerabilities in websites and reporting them. He made some money from me. It's a voluntary payment, but I wanted to thank him for bringing this to my attention and for everything that transpired because I ended up with something that I think will be very valuable to a subset of our listeners, arguably anyone who has responsibility for a website. So that page didn't show me what the problem was. I had no idea where there was a problem. The fact that this person, this tbmnull, had reported so many caused me to give it some credibility. But, so, okay, what did he find?

So the first thing I did was I thought, okay, there's got to be some free or reasonably priced scanning services, sort of like ShieldsUP has always been for a person's ports, but this would be a service that would scan a website, looking for these kinds of problems. There's got to be. And sure enough, I dug around, and I found three that had kind of the right profile. I think there was like a list of 11 that I found in an article. But the other eight were sort of off target. They weren't exactly what I was looking for, which was spider my site and tell me what you find.

So the first one was an offering called Acunetix. And they offer both an online scan and a local Windows app, which I love the idea of being able to download an app. So this would be very useful if you wanted to scan a non-publicly facing Intranet server, which an outside site could not get to. Or if you just like the idea of running your own scan against your own site, rather than having a third party do it. So I signed up, downloaded the app, and they were clearly in a trial mode. But it was like, okay, they're saying that they're going to allow me to do scans in trial mode.

One of the things that you always have to do, and it's very much like applying for a certificate, you need to prove ownership of the site, not surprisingly. That prevents random third parties from scanning other sites that they don't own and control and having problems revealed. So clearly a responsible scan service would require you to prove ownership. In this case, it was put a file on the root, and I did that, and it just worked. So authenticating with them, authenticating that I was the person who had control of the content of GRC.com, not a problem.

Well, then, okay. So their online service scans - and I don't remember now which one I did first. But I tried both. And both of them reported all kinds of horrors. Oh, my god, the sky is falling. How have you not been attacked every which way from Sunday by now? What's wrong with you? But we're not going to tell you what. And I said, what? And then, you know, dire warnings, but you've got to pay us. And so, well, which did not impress me because the whole thing was a bait and switch. Nowhere was there any implication

that we're going to frighten you, but use that to get your money. Which just pissed me off. Also, for what it's worth, the local app was very poorly written. They want many hundreds of dollars a year, so it was not like it was cheap.

But, for example, one of the first things you learn to do when you're programming Windows is you create a UI thread that is a separate thread to run the UI. For example, the DNS Benchmark has one. While the benchmark's running, you can push buttons and flip pages and do all kinds of things. You know, it all just sort of works at the same time. Well, this thing had pause and stop buttons that you couldn't push when it was running, although that's a little contrary to common sense, because they don't have a UI thread, or it's not being sampled often, or I don't know what. But anyway, I ended up thinking, okay, this is not what I'm looking for. I don't want to be told that the sky is falling; but, sorry, we're not going to tell you where. Then the spam began, four or five pieces of marketing sales promotion per day from this company. So anyway, Acunetix, no.

Second one I looked at was a company called Beyond Security. Now, I had an interesting problem with those guys. I was unable to authenticate without their help. They provided me a file that I drop onto my root, but they refused to see it. And I immediately realized why. And that is, from the first day of its existence, like 15 years ago, the root redirects to a page called Intro.htm. I don't know why. And if I were to do it again, I wouldn't. But back then, 15 years ago, it was all sort of magical and new, and I thought, ooh, this'll be fun. So there's a 301 redirect which is the only response anyone gets to trying to pull GRC's root page. It doesn't deliver the full document. It bounces you to a named document.

Well, apparently Acunetix knew how to follow the 301 redirect. I don't think that represents a security vulnerability. But the Beyond Security guys' scanner didn't. So I was unable, late at night, whenever this was I was doing this, to use their service. I did send them a note, and I think they recognized me because they immediately said, "Oh, hi, yes, we'll authenticate you." And so I was able to user their service. I noted that their page required Flash for displaying summary graphs, which is a little disappointing because there's three blacked out boxes because of course Flash is not running without my permission on my browser.

FR. ROBERT: That's the first warning flag, there. A security scanner that demands you have Flash. Like, oh, okay, hmm.

Steve: They're very nice people, and they were willing to work with me. But their scanner didn't find any problems. It gave my site, GRC, a 100.00 and an A+ grade. So I thought, well, okay. I know, I'm pretty sure there is a problem somewhere. I'm trying to find it. So far one service says, oh, my god, how is this server even up? But we're not going to tell you how to fix it or what's wrong without money. And then these guys, who are fabulous people, said, yeah, no problems here. Move along. Nothing to see. So I thought, okay, I've got to keep looking.

Number three, and the third time was the charm: TinfoilSecurity.com. I am incredibly impressed with this service and with the user experience, with the user interface. It is a teaching site. It is the Security Now! podcast chosen website security scanner until further notice. I'm sold. Now, their authentication, as with Beyond Security, would not allow a resource on the site to be seen. So they're also not following a redirect to my intro page, which is my nominal home page. But they offered one additional means. I could put a text record in my DNS. And that's, like, brilliant. So that demonstrates I'm controlling the domain of GRC.com, which is tantamount to controlling the server.

So they gave me a blerch blob of nonsense. And as I was talking about earlier, I updated

the GRC.com zone with that text record and then informed Level 3's DNS servers that I had a new zone. They grabbed it. And then I said to Tinfoil, I clicked the okay, authenticate me, and they said, yay, you're authenticated. So now they knew that I was in control, and the account that I'd just created was there. They offered a single-domain 90-day trial. It's a three-month trial, which I think is generous. And believe me, I'm sold. I'm, well, and so here's the details.

Their report found 64 problems. Now, okay. And I'm going to go through them real quickly. They're finding things that are good to know, but they're not cross-site scripting or cross-site request forgery problems or anything, but useful. And our listeners will get a kick out of some of them. But this is the kind of thing where I'd rather have too many than to be told, oh, you get an A+. Nothing wrong. So because you can always look at them and go, okay, I know about that. That's not a problem.

So most of them were informational. A couple were low impact. And then there were some that they thought were high impact. For example, I was notified, one of them, they called it "credit card number disclosure." And sure enough, their spider found a string in my PDP-8 source code listing, which I have on my PDP-8 pages: 4207 0610 0000 0013. That's a credit card. Except it's also octal, and it's machine language.

FR. ROBERT: I think that's Leo's credit card. Hmm, Amazon.com.

Steve: So, and maybe they're smart enough to know that the first digit has to be a four. That's actually one of the leading digits for credit cards. But I was impressed. It was like, okay. If I actually did have somehow a credit card flapping in the breeze, I'd want to know. This thing would have found it for me. In this case it wasn't a problem.

They also identified 10 instances of what they called "missing subresource integrity protection." Now, I should explain that with every one of these, it's itemized. You can click on that item. It takes you to a page that completely breaks it down, shows you where it was, what the query was, what the response was, what they matched on, and what it is they think it means.

So, for example, in this case they said: "All externally loaded resources" - externally loaded resources - "must have their content pinned by using the subresource integrity mechanisms provided by modern browsers. This involves computing a hash of the contents of the resource, and specifying this hash when loading that resource. In the case of a script, this might look like the following." And they give you a sample of, like, here's what we mean, where they show include.js being loaded from example.com, and then the integrity is an SHA-256 following.

And so the point is that this is your declaration of exactly the code you are pulling from some remote site that you don't have explicit control over. So this is your way to tell the browser, make sure no one has messed with this. If you get it and load it into the web page, hash it and make sure the hash balances. Otherwise don't. So great advice. And there are a couple places where, on my side, I am pulling something from somewhere, and they saw that. And it's like, bravo to these guys. Again, another nice piece of information.

They also found 14 email addresses. And I thought, what? Well, it turns out in more than 11 years of Security Now! transcripts, some email addresses for one reason or another have been put in the transcripts. It found them all. It went through every single Security Now! transcript and said, hey, you've got some email addresses here. You sure you want to expose those to the public? And again, not a problem, but very nice to have something with a push of a button that can go find those for you.

And this was interesting. There are two problems which it called "found robots.txt." And I thought, well, yeah, of course. I have a robots.txt to keep spiders from getting lost in my dynamically generated code. And so when I clicked on it they said: "If you require it, try not to list important files or directories in robots.txt. Instead, password-protect them such that, even if they are crawled, the crawler gets nothing but an authentication page. A good robots.txt file includes content like image directories or locations that are generated dynamically and do not work if a search engine accesses the page, but does not include administrative areas or server logs." And so I think that's sort of a generic comment that they make when they, you know, sort of like, "Here's a tip for tightening things up on your site." Again, a nice observation.

They found seven things they called an "HTML object." And I do have some. I have HTML native video players, for example, on the various PDP-8 pages and on a couple of the SpinRite pages. And so their comment was: "This is nothing to be concerned about at the moment, as we are purely providing it for informational purposes. It often gives the hacker a beachhead of where to begin searching for a vulnerability, but isn't a vulnerability in and of itself."

They reported four insecure cookies, saying: "Set the 'secure' flag in the cookie such that all cookies are served over a secure channel like HTTPS only. If you tell us what software stack you're running on your website, we'll be able to give you more detailed results on how to fix this issue. You can do this from your dashboard." And then there's a link there to their dashboard, which is their online configuration tool. Oh, and I should say, by the way, you can dial how aggressive you want the spider to be. I think it defaulted to 50 simultaneous requests. I pushed it up to 100 just because I have a fast server. But, I mean, it did, I could tell when it was on the site, it was sucking everything out. But again, if you want a less aggressive spidering, you can ask for that, too.

It also noted non-HTTP-only cookies four times. And they said: "Set the 'HttpOnly' flag in the cookie, such that cookies cannot be manipulated via client-side code like JavaScript. If you tell us," again, what software stack you're running, blah blah blah, we'll tell you how to fix it. Again, the whole experience is handholding; explains what's going on; and, where they can, gives you samples and fixes. And I should mention, the only place I use cookies is in that - and they're crazy named cookies. I don't do it for any sort of login state ever.

Even my eCommerce system does not maintain state using cookies. It encrypts a blob of current transaction state and returns it with each phase of the transaction. And no one has the decryption key but the GRC server. So even there I don't use cookies. The only place I do is in the cookie forensics research that I did when we were finding bugs in cookie handling years ago in browsers. And so the reason I've got cookies is just as instrumentation probes, essentially. And they're right. There's no reason I don't have them marked secure because now GRC is all secure. I just never went back and retrofitted it. But here's the reminder. It's like, okay, that would be a good thing to do.

FR. ROBERT: I've been looking through their site as you've been talking, and I love the fact that they separate issues from potential exploits. I think that's very important for someone who's just scanning their page, and they're not going to freak out. And then I love the fact that you can drill down on the individual issues to see exactly what they're talking about. And, honestly, I'd never thought about that with robots.txt. Yeah. [Crosstalk] files tell someone who might want to exploit my site exactly where I don't want them to go.

Steve: Right, right.

FR. ROBERT: Oh, wow, okay.

Steve: And so the last of the insignificant things was 16 what they called "private IP address disclosures." And again, given the nature of my site, you can imagine, and in fact on the intro page to ShieldsUP I say "You should be aware that scanning probes will come from this range of IPs," from this IP to this IP. So there's two, you know, there's two IPs. So again they said these were again - oh, no, I'm saying in Security Now! podcast transcripts. And then they said: "Remove private IP addresses from the body of HTML pages." And they wrote, "These can often be left over from testing and may indicate that your website is not running in the environment it expects. These IP addresses also give information to an attacker that can be used as a beachhead for other more harmful actions." So again, if you don't have reason to know that you've got IP addresses exposed, that's something good to know.

But they did find three things. One was a false positive, but I love that they flagged it. On the GRC Fingerprints page, I provide a field that allows someone to fingerprint the certificate of any domain they choose, of any site they wish. And they flagged it as a potential cross-site request forgery, saying it was due to no replay protection. And their advice on that said that it would be possible for some entity to resubmit that form in the future because there was no - like a serial number embedded in the post data that the page was placing in front of the user, and that the user was then sending back to the server. And that's true. But that isn't an issue here.

And I do show the domain name in a - I display the domain name back to the user, which is sort of scary because anytime you display something the user provided, that's the big danger, when we've talked about things like SQL injection attacks or cross-site scripting, like for example a forum, back in the days before forums were carefully filtering what users submitted and they posted, when the forum puts up on the display what somebody has submitted. Well, if they submitted something malicious, and it was left intact, that would be displayed and then potentially running in the user's browser. So, but in this case I carefully parse out the domain name from whatever the user provided, and I have a zero tolerance policy. And so I tried to, like, mess with it and get something through and was unable to. And nobody thought that was a problem anyway.

However, they found four URLs in the discussions client, essentially. GRC, as I've often talked about, has an old-school NNTP server where we have a bunch of discussion groups. That's where all the work on SQRL has gone on, and it's where I will be returning to SpinRite to work with a community of people testing and asking for thoughts as I move forward. It's a huge resource for me. We have a read-only facility which allows people who just want to browse what's going on, that's always been there, which I did not write. It used to be called DNewsWeb. And their server was awful, so I had purchased it years and years ago, and I abandoned it and switched just to FreeBSD Unix and a real INN Internet news server. But I kept their little web frontend because it was just a handy means to allow people to browse.

Well, it turns out that it has a facility to allow people to kind of login. They can identify themselves. I don't use that. But it's called a "utag," a user tag. And once you provide that, it goes in the form. And then, in order to link successive browsing together, every response returns it. And it's not filtered. Meaning that anything you put in the utag will be found and returned. And so if something malicious were put in there, then that would come back and potentially be executed by the browser. So there are four instances of that. The bad news is I didn't write that little, I mean, it's an inelegant piece of code. It's big, and it's an EXE, and it sometimes misbehaves. But it hasn't caused us any problems.

The problem is, without source, I can't fix it. And so it technically represents a problem. So what I did was I wrapped those pages in what's called Content Security Policy, CSP. Content security policy can be delivered either as reply headers or response headers in the page, where their metadata is not seen, or in the actual HTML headers of a page. And essentially they control what the browser will do. Without any content security policy, because we need to be backwards compatible, everything is wide open. So there is no control. But you can add the CSP, the Content Security Policy rules, which will restrict what can be done.

So, for example, now all of those browser pages, every one of them has a header which is incredibly restrictive. Just enough for the page to work. I said, I allow images to come from GRC. I allow CSS. I'm blanking. CSS style sheets. I allow the GRC style sheet and images and text. But, like, nothing else. No script, because this thing does not need scripts. No inline scripts. No child frames. No fonts coming from anywhere else. No media, no other objects, and so forth. So it is just - it's completely locked down. It's not as nice as being able to filter it. But I don't want to take the time to wrap that in my own code, to provide a filter or to write one myself. And I've got work to do. So that solved that problem.

But so those things it found. And then finally they found what the attacker found. And there actually was a problem. Again, in the particular case of GRC, it wasn't mission critical because I don't have any cookies of any value. And it's hard to see how this would actually get exploited. But under the support page, or sales support, I allow a user to put their SpinRite purchase transaction ID into a form and basically make a database query. We look up that record for them, which then allows them - they get download links if they want to download another copy of SpinRite, or they want to print out another copy of their purchase receipt or whatever. If their transaction ID was not found, I was providing them the service of returning the form, saying that we did not find your transaction ID, and populating the field with what they provided. [Buzzer sound] That's dangerous because user-provided data is being returned without filtering to the site, that is, to their browser.

And what this white hat hacker found was that there was a clever way of escaping HTML characters such that he was able to insert inline script in the form field. You actually couldn't do it in the field because I had restricted its length. But a spider could do an automated posting that would have no length restriction because it wasn't being enforced by the page UI, and get their script to execute in the GRC domain of the user's browser. And you don't want that to happen because, if I were, for example, maintaining state, like session state, in a cookie, script running on a page from GRC is trusted. And it would have access to, if my cookies were not marked HttpOnly, which is the other thing these guys found, it would have access to cookie contents and could then, for example, generate a URL query to a malicious site with that cookie, the name and value embedded in the query tail, and thereby exfiltrate data that was meant to be private between the server and the person looking at the page.

So I was delighted to find this, and I immediately fixed it. The standard best practices says that less than, greater than, ampersand, double quote, single quote, and forward slash are dangerous. You cannot allow those to come to the browser unescaped. And so the update I made a couple weeks ago is to have a safe HTML send function that this and anything else that sends data back to the user runs through. So, for example, it converts an actual less-than symbol into the <, which the browser displays as less-than, but which doesn't trigger script execution. It's not like the angle brackets containing the word "script." And I also added content security policies just as belt and suspenders, as well.

So, and I notified the hacker, and I thanked him. He mentioned that he'd be willing to

accept a little donation in thanks, and I treated him very well with an Amazon gift certificate, which is the means that he suggested. And in the process I found an online website scanning service that, based on all of my experience, and even the experience of looking at the competition, I can recommend without hesitation. TinfoilSecurity.com, 30 days for free. I mean three months, yeah, 90 days, three months' trial, 100% functional and much more comprehensive than anything else I have found. If you are responsible for a website, I don't know what you're doing still listening to this podcast. Sign up, have them scan your site, and then browse through the results. There's no way, based on what I've seen, they could ever be happy, which is what you want from something checking your site to make sure you haven't had any oversights. So bravo to Tinfoil Security.

FR. ROBERT: Steve, talk about burying the lede, this should have been at the top of the show because this is a practical example of so many things that you talked about over the years on Security Now!. This is you finding a vendor that you trust. This is you finding a couple of vulnerabilities within your own site, mostly because...

Steve: I know more than I did before, precisely. It educated me.

FR. ROBERT: This was your "Iliad" and your "Odyssey" all rolled into one. And thank you. This is something I'm hoping that we hear from more podcast celebrities and more big websites because I think the more that you disclose about the vulnerabilities that you find, the more people are secure, the more that they'll patch it in their own instances, their own services.

Steve: Right.

FR. ROBERT: Of course, we have gone way over time, which I guess is actually pretty normal for Security Now!.

Steve: We broke our record, I'm sure, Father.

FR. ROBERT: No, no, no. Actually the two of us still own the record. I think we went 2:40, 2:40 something on one.

Steve: Okay, we're at 2:24 right now.

FR. ROBERT: 2:24. And they actually restructured the schedule so that we can't break our record without really messing up the shows behind us.

Steve: And we're not going to do that.

FR. ROBERT: No, we're not going to do that. But of course, Steve Gibson, you're going to find him every week here at, what is it, 13:00 Pacific time - 13:30 Pacific time. He is our security guru. You're going to find him at GRC.com, where you can also find transcripts and audio versions of the show. We do Security Now! here on the TWiT.tv network normally with Leo Laporte, but I'll be subbing for him while he is looking for the Holy Grail.

Steve: Two more times.

FR. ROBERT: Two more times. Two more times. Exactly. And I'm very much looking forward to it. And don't forget that you can find all of the back episodes. If you need to replay - and trust me, there's no dishonor in having to replay it because Steve is pretty

condensed in the information that he gives to us - you can always go to his show page at TWiT.tv/sn, that's for Security Now!. There you'll find all of the back episodes, as well as ways to subscribe to get the show automatically downloaded into your device of choice. Of course, you can also find Security Now! wherever fine podcasts are aggregated. Steve, would you like to say some final words of wisdom before we send the Internet back into the ether?

Steve: I think you've got it covered, Padre. It was a pleasure working with you for the last 2.5 hours, and we'll do it again next week.

FR. ROBERT: It's been an absolute pleasure, Steve. And on a personal note, I do have to say this. And this only makes sense to the people in the TWiT Army. I'd like to say rest in peace to Tater, and we will miss you. Until next time, I am Father Robert Ballecer. He is Steve Gibson, the man himself. And we'll see you next time on Security Now!.

Steve: Thanks, Padre.