SECURITY NOW!

Transcript of Episode #283

Bluetooth Hacking

Description: After catching up with the week's security and privacy news, Steve and Leo complete their analysis of the Bluetooth security by examining the history and current status of Bluetooth hacking exploits. They conclude with a set of recommendations for minimizing the Bluetooth attack surface.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-283.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-283-lq.mp3

Leo Laporte: It's time for Security Now!, with Steve Gibson, Episode 283, recorded January 12, 2011: Hacking Bluetooth.

It's time for Security Now!, the show that protects you online, your privacy, your security, your way of life. And here he is - he's kind of a superhero. He doesn't wear a cape. He wears a shirt that says, "No, I will not fix your computer," Mr. Steve Gibson of GRC.com. Hi, Steve.

Steve Gibson: Hey, Leo. It's great to be with you again, as always.

Leo: Always a pleasure.

Steve: Back from CES, for you.

Leo: First time I have actually ever gone to the Consumer Electronics Show without killing myself, I have to say. And I guess it's because I had such great help with Tom Merritt and Becky Worley and Sarah Lane, our editors Tony Wang and Jason Howell that edited and produced, and Eileen Rivera, who's absolutely the best producer in the world, produces this show for us.

Steve: And you came back in one piece.

Leo: And they worked, and I didn't. And Lisa Kentzell, who is our financial person.

And, yeah, it was just a - it was really fun to go with a team like that and hit the ground. Tom Merritt said, "This is the most fun I've ever had at CES. I actually got to see some of the show."

Steve: Very cool.

Leo: Yeah, so it was fun. Now, we were talking before the show, and you said you didn't see anything to knock you out at CES.

Steve: Yeah. I watched you guys, and I watched the news. And I am a BlackBerry user, so I'm interested in their little tablet. I mean, I think I like the size that they're talking about because it's enough smaller than the iPad to be more portable, and enough larger than the iPod Touch or the iPhone to give you more browsing space for the web. So that's interesting. Although I didn't see what the resolution of it was. But probably enough. I did hear it was high resolution.

Leo: Yeah, it's funny, I should know that. I'll have it somewhere, and I'll find out for you. But, yeah, just - I think that's, to me, the RIM PlayBook - I guess they're calling it the BlackBerry PlayBook - was the product of the show. But I liked the Samsung Galaxy Tab. I know you like Android, and that's pretty nice, too. I think we're going to see some great Android and other - because the BlackBerry's QNX based, of all things - operating systems in addition to iOS.

Steve: And QNX is an old-time, old-school, really solid, strong, embedded OS. So I was...

Leo: And real-time; right? It's a real-time OS.

Steve: Yes. It was what people, it's what engineers would license when they needed to do, like, real heavy-duty, process-control, real-time work. And so I was delighted when I saw that RIM had grabbed up QNX. It's like, oh, there could be some good results from that. So basically it was a kernel that was very mature that they've built this GUI tablet on top of, so...

Leo: Well, wait'll you see the multitasking capabilities. And I guess this is where having a real-time OS makes a big difference. It's incredible. And they've added a great touch interface. It feels very modern. But you're right, it's stable and well-tested.

Steve: And I do love that they don't miss an opportunity to mention that it runs Flash applications.

Leo: Yeah. Everybody says that. I actually didn't look at the - I should have - at

Flash performance. It does have - get this, Steve - dual 1GHz processors. So if it can't run Flash, what can? And it's a Tegra 2 chip. They wouldn't - it's funny, BlackBerry said we don't talk about that. But AMD, I'm sorry, Nvidia, which makes the Tegra 2, has been trumpeting it all week. That's an amazing processor with Nvidia-class GeForce GPU in addition to the two GHz processors, and a gig of RAM. That's a real computer.

Steve: Wasn't it around the CES timing that we first really learned that Microsoft Windows was being ported over to the ARM architecture?

Leo: Yeah.

Steve: Yeah. So that's interesting, too, where they're saying...

Leo: That was one of the stories, I think, of the show is the decreased dominance of Wintel. And Intel's got to scramble at this point. I mean, now Intel's completely in charge. I'm not saying that. It's funny because Microsoft and Intel still are the big guys. But what you see is a movement away from them to ARM instead of Intel. And frankly...

Steve: Yes, to Android.

Leo: To Android and iOS and maybe even QNX instead of Windows. And I think that that's telling.

Steve: Yeah.

Leo: Anyway, very - there were, as always, it's one of those things where you go to the show, and it's hard to see the forest for the trees. And maybe when you come back you start to say, hmm, ahh, hmm. And one of the things I thought was, Microsoft had a big booth; Intel had a big booth. But that's not where the story was.

Steve: Right.

Leo: What else is on? Shall we get to Security Now!?

Steve: Maybe we should do that.

Leo: All right. It's your show. I'm sorry. What...

Steve: I was just about to start talking about the odd Nintendo warning about young kids who aren't supposed to look at their 3D display. And a lot of orthopedic - not orthopedic, pediatric ophthalmologists have said, what? We're unaware of any such phenomenon that would affect eye development. And so it's like, okay, it just sounds like Nintendo's hedging their bets and...

Leo: It's a CYA, yeah.

Steve: Exactly. They just want to say, hey, if anything ever does turn up, they'll say, hey, we warned you, we told you. So it's like, okay.

Leo: I'm almost tempted to put a warning on all our shows saying, if you're epileptic, don't watch because there might be strobe action. I don't know. It's funny, on The Screensavers we used to get emails from people saying, "I'm epileptic, and whatever that was you did really is dangerous." And I'm sensitive to that, but I don't know what it is; you know?

Steve: Please don't wave your arms around at 13Hz.

Leo: Right. So maybe there are some people who are at risk of the 3DS. But who knows.

Steve: Well, eventually we're going to talk about hacking Bluetooth.

Leo: Hmm. That's our subject of the day today.

Steve: That's our topic for the day. Three weeks ago we covered all of the, in propellerhead-winding detail, the technology of Bluetooth, how the protocols operate, how the crypto operates, the nature of pairing of Bluetooth device addresses and all of the minutiae of that. And I promised then that we would next talk about the dark side, that is, the hacking side. And there has been a huge amount of work done by the bad guys, and some of the gray hat hackers, too, people who say, well, we don't want to really do anything bad. We're just curious what we could do if we wanted to. So our topic today is hacking Bluetooth. And of course we've got updates and news and even a little bit of errata.

Leo: So let's talk. Patch Tuesday happened while...

Steve: Yes.

Leo: Yesterday.

Steve: Yup, couple days ago. We had a relatively uneventful Patch Tuesday. Just two different things were fixed. There was a problem with Microsoft's Data Access

Components, which was a critical rated vulnerability across the board. So all versions of Windows that Microsoft is currently supporting, because that's a common component across their platforms, had some problems that they fixed. And then a so-called "important" as opposed to "critical" vulnerability in the Windows Backup Manager. So both of those are a little bit of a yawn.

What's interesting is that that means many of the zero-day vulnerabilities which have just surfaced, there's the CSS stuff and some other exploits that we've talked about in recent weeks, were not addressed by this. There are, as we said, some quick-fix, single-button, click-on-the-announcement fixes for those things, which Microsoft has suggested while they're working on permanent fixes. But I was surprised that actually we had as little done for this first update of the new year. But that's all. And nothing from anybody else. Adobe didn't have anything happening...

Leo: Because everything's perfect; right?

Steve: Uh, yeah. Actually, there was news, but I thought, well, I'm not sure this quite makes the bar. There was news of someone who found a way around the Flash 8 sandbox in Flash. Adobe has put a sandbox around, and there was an information leakage exploit. But it's like, eh, okay. We're all at Flash 10, and there's a sandbox there that is not workable around in the same fashion. So I said, okay, we'll leave that one, see if anything develops there further. But otherwise, not that much.

We talked a little bit last week about the topic in two weeks, which is the so-called "browser fuzzing," which Michael Zalewski, who works for Google, did, which revealed more than a hundred problems across the board. Every single browser that he fuzzed, and we'll be talking about what that is in two weeks, was found to have problems. And you'll remember that there was some confusion because he provided Microsoft with his proof-of-concept code back last summer in July, and they were unable to reproduce the problem, so they said. Then he ended up telling them, I mean, with lots of notice, that he would be going public with this at the beginning of 2011, as he did. And it wasn't until halfway through December that they apparently woke up and said, wait a minute, we don't want you to do that because now we're able to reproduce the problem.

Well, it turns out, I was curious about what was going on. And Microsoft has now said more about why they're not considering it a huge concern. And again, we'll talk about this a little more in two weeks when we go into this in detail. But it turns out that it takes a series of specific HTML pages to be loaded, one after the other, in order to incrementally destabilize Internet Explorer to get it to the point where the final problem of the last page to be loaded manifests in this exploit. So they're saying, well, yes, you were able to destabilize IE, and we agree that's not a good thing. But if you just do the page that makes it hurt, like all by itself, there's no problem.

Leo: No big deal, yeah.

Steve: You have to precede it with all these other things. Which, to his credit, Zalewski's fuzzing program does find. And so, yeah, hopefully Microsoft's going to take this seriously and find out what's going on. But that sort of explains how this fell through the cracks and what the controversy was between Microsoft's position and his. And for what it's worth, I think he really did give them sufficient notice and that they probably should have been paying a lot more attention to this, rather than trying to stop him here at the last minute.

Just in sort of generic worrisome news that affects our listeners sort of tangentially, I did pick up a little note that the California Supreme Court came to a troubling ruling on Monday, January 3rd, right off the bat here in 2011. They ruled that police can search cell phones without a warrant.

Leo: Oh. You know, it's funny, at CES I met three different cell phone forensic people, including the guy who did the Scott Peterson cell phone. It's a big - it seems like that's the thing you want is a cell phone.

Steve: Well, yeah. And exactly, because people have - think about our own use of the cell phone, which is why I wanted to bring this up from a privacy standpoint and technology impact. It's really not a phone as much anymore as it is a computer. I mean, there's email, there's texting logs, there's all your contacts. I mean, a chunk of who you are and what you have done in the last couple years, in some cases, can be stored there. And what is very troubling - and I don't know that this California Supreme Court ruling is going to stand. The defendant Gregory Diaz, who was caught in a sting operation purchasing drugs from a police informant, the defense attorney intends to appeal this to the United States Supreme Court. Because what happened was, at the time of his arrest, police seized his cell phone and found text message logs implicating him in additional drug-related activities. And it's the Fourth Amendment to the Constitution is what protects us from so-called "unreasonable search and seizure."

The California Supreme Court disagreed with his defense, stating that cell phones are similar to personal effects such as clothing, which can be searched by arresting officers. Which many people, I mean, this has generated a huge backlash on the 'Net and in privacy rights blogs, and even other judges have felt that, well, in fact the dissenting opinion, it wasn't a unanimous opinion, the descending opinion from the California Supreme Court argued that this really raised some concerns. Other people are saying, well, if this, if cell phones qualify, then what's to prevent it from being PDAs and laptops, if the person's carrying them at the time of their arrest? So anyway, it's worrisome.

Leo: Yeah, no kidding.

Steve: It turns out that they likened it to a previous ruling that equated it to police inspection of a cigarette pack taken from a subject, which was...

Leo: No, that's not.

Steve: Exactly.

Leo: It's not the same.

Steve: I mean, a phone is now a computer. So...

Leo: That's like, you know what? Liken it to the wallet. How about taking my wallet and searching it? How does that feel? Is that legal? Hmm, wow.

Steve: Yeah.

Leo: This just shows you how clueless the courts are.

Steve: Well, the good news is we've got people standing up for our rights. And I'm sure this will not, I mean, this is too, too troubling a ruling to go unchallenged. So I think it's probably likely that, I mean, to me it feels like something that might interest the Supreme Court from this standpoint. I hope that that just doesn't make it set further in stone.

Also, there was buzz since we last talked about something that's still very ill-defined and is worrisome, which is this so-called "identity ecosystem" which the current presidential administration, our Obama administration, is continuing to talk about. Now we're apparently a few months away from them unveiling something which is, first of all, it's not mandatory, it's optional, whatever it is. But I was reading some statement from Obama talking about how, with the increase in eCommerce - this season it was up, I think, 5.5 percent over the holidays over a year ago, which is a huge increase. And so it's people in the government beginning to awaken to many of the things that we've talked about on the podcast, like OpenID and multifactor authentication. And we've been dealing with the technology side of, absent any legislation, absent any formal solution, what can we do to solve the problem now? Well, even my own little Perfect Paper Passwords and similar things. So now we're beginning to see, at the governmental level, them rummaging around and saying, well, we need to do something about this. We need to provide solutions to our citizenry. I just hope they don't really screw things up. So we'll keep our eye on whatever this "identity ecosystem" is.

Many people who are following me on Twitter sent a note about something that Reuters picked up, and then it was echoed by many other news feeds, which talked about a new WPA vulnerability. It was really troubling because it isn't a new vulnerability. The Reuters headline said: "A security researcher" - I'm quoting. "A security researcher says he has figured out a quick and inexpensive way to break a commonly used form of password protection for wireless networks using powerful computers that anybody can lease from Amazon.com over the web." So the question is, is WPA vulnerable to this new password-cracking tool? And the answer is no.

So the story behind this is interesting, though. A German computer security consultant named Thomas Roth essentially used the Elastic Computing Cloud, which is one of the services that Amazon offers - we've talked about AWS, the database in the sky, in the cloud, for example, which Jungle Disk is able to use. Well, they have the ability also to do something called Elastic Computing Cloud, they call it EC2, which allows you to essentially grab or commit a large number of processors and get them all working at your beck and call.

So essentially what this guy has done is he's demonstrated the fact that, with this cloud computing, where we're actually talking about computing resource, not just storage resource in the cloud, that you could grab a bunch of computing resource that could potentially be used for brute force attacks and get the benefit of parallel computing without having to spend tens of thousands of dollars to do that. We've talked about in the past other people have used, like, walls of PS3, PlayStation 3 systems which have very powerful graphics processing units, GPUs, to, like, do brute-forcing attacks on crypto. So what he did was he - and I don't think I would have admitted this, were I this person. But he says he cracked the crypto of a WPA-protected WiFi network in his neighborhood in I think he said 20 minutes. And then he subsequently improved the technology so that it ought to be able to do it in six minutes.

So the reason this is not a concern for anyone who listens to this podcast is nobody who listens to this podcast will still be using, hopefully, an easy-to-crack password. This is all about the password being easy to crack. If it is random letters and numbers, and if it is long, then it will not be easily cracked by a brute-force cracking tool. And again, the number of bits per character in a large alphabet password, that is, a password whose characters are upper and lowercase, special characters, and digits, basically just looks like just jumbly gibberish, the kind of thing that my own password generator at GRC generates for people. Take a chunk of as much of that as you want and use that as your key, and you're safe against this.

But this does demonstrate something that actually we'll come to a little bit later when we're talking about hacking Bluetooth, and that is that assumptions have been made in the past about the available computing resource that were available, not to nation states, but to individuals. And we've seen the walls of PS3s that have gone after cracking. The good news is that we have so many bits, and many bits create so many combinations, that there's concerns about the future of quantum computing that's supposed to be so much more powerful.

Well, this guy, using a chunk of computing resource, was able to test 400,000 potential passwords per second. So that's a lot more than you can test on a PC. But it's still, that doesn't even begin to get close to the number of passwords that we're able to test or we're able to have as potential when you have a huge number of bits that is potential in the password. So this generated a lot of news. Anybody who's got an insecure password was already insecure. This demonstrates, though, now that individuals for - apparently this was costing him 28 cents a minute to use that much computing resource. So the lesson being that computing resource is now elastically available, it's rentable, and it does up the ante for the need to make sure that your passwords are bulletproof.

Leo: Yeah, as if you needed an encouragement.

Steve: Exactly. But as I was going to say, anybody who's listening to the podcast already will be using a password that is not in a dictionary and would just take, still, hundreds of thousands of years, even at this level of computing, to try your random gibberish passwords. It's just not going to happen.

And lastly, I just thought I would check in on Firesheep and note that it has crossed a million downloads. When I looked this morning before sending the notes to you, Leo, it was 1,043,468 downloads of Firesheep, which is the...

Leo: Oh, man, oh. It's not slowed down.

Steve: No. It's continuing to crank along. So that's our download add-on for Firefox which allows people to go into any open WiFi hotspot and, unfortunately, with shocking ease, hijack many of the social networking sites and other services that people are using, unfortunately, without encryption on open WiFi networks.

I did have one little bit of errata from a Swedish listener of ours who's actually located in Orangevale, California, Peter Jakubicki. He wanted just to mention that Sony had not bought Ericsson, that Sony Ericsson...

Leo: Partnered.

Steve: ...is a joint venture, exactly.

Leo: They corrected me on - they have a new phone, the Xperia, and I said the same thing. And they said, no, no, we're a joint venture.

Steve: Yup. I'm sure back then when I first heard it, what I read was Sony bought Ericsson. I had this clear image in my mind that big fish had swallowed little fish. But nope, not the case. And so he said, "We Swedes care about these details, so please fix this." And I said, "Okay, Peter, I will definitely do that."

Leo: Even though it's a Finnish company, isn't it? Oh, I'd better not say that now.

Steve: Yeah, that's a really good point, yes. Oops. And I did have a fun note from a Martin Parrott, who wrote to say SpinRite had saved another system. And he said, "Steve, I've written before, but SpinRite has done it again. And I wanted to send another email to say how much I appreciate a great product. A friend of mine had a machine with SCSI drives" - don't hear that that much these days anymore - "with SCSI drives installed that are around six years old. One drive recently started showing errors in Windows' event viewer, and I took my SpinRite disk over to check it out. Indeed, it appeared the drive was starting to fail. The drive has always had blocks marked as bad, ever since it was new.

"I ran a Type 3 scan" - that's a SpinRite level, SpinRite Level 3 scan - "on the drive, and four hours later it finished. I checked the status of the drive, and not only had it refreshed the entire drive, all the bad blocks had been recovered, and the map showed all sectors as good and working. We rebooted the server, and Windows is happy again. After a few hours of use there were no more warnings in the Windows event logs. The machine was stable enough now to allow full backups, which could not be done before. And it appears it will see a bit more use until new drives can be ordered and installed. If only other software was as dependable and useful. Thanks again. Martin Parrott." So thank you, Martin, for the report of SpinRite's success.

Leo: We're going to get to - it's not snarfing - Bluetooth hacking. It's kind of - some of it's snarfing; right?

Steve: Oh, well, we've got BlueJacking, BlueBugging, BlueSnarfing, BlueDiving. We've got something called HeloMoto.

Leo: Hello, Moto.

Steve: The Carwhisperer, BlueTooone, BluePrinting, and Redfang.

Leo: Oh, my. I love it.

Steve: The hackers have been busy.

Leo: Yes, they have. Now, let's see. Let's snarf, or whatever it is. Bluetooth time.

Steve: So when I began three weeks ago to dig deeply into Bluetooth, my eyes just crossed when I ran across the history of hacking of Bluetooth. And I thought, okay, there's no way we can cram all of this into the same podcast. So I decided to do the deep technology and operation of Bluetooth first and then swing back around and talk about the dark side, the hacking side, the history of all this.

If it was six years ago, in 2004, 2005, we'd be in trouble because there were, looking at sort of the characterization of the problems that I found when I looked at what had been possible in the past, it was clear that it was a case of people throwing Bluetooth onto existing platforms, like PDAs, that were not about Bluetooth, they were about being a PDA. But then they said, oh, the competition has Bluetooth. We'd better add that on. So they threw in a little Bluetooth radio and dropped in a Bluetooth stack for providing the insanely complex protocols that the committees had worked out. And they said, oh, yeah, we've got Bluetooth also. The problem was that, I mean, our listeners, again, who are getting a sense for the nature of security and how much a conscious effort security requires, could already guess that security wasn't even a consideration, unfortunately, for these people. Functionality was; getting it out the door; making it sell.

And so what was discovered back then was that there were all kinds of problems. One of the developers or hackers who was looking at this in '05 had a PDA which did not need to be paired, but was able to provide filesharing functionality when it was just in discoverable mode. So it was essentially wide open, if he left his PDA as discoverable. So none of the security which Bluetooth always provided from the beginning was engaged in many of these early devices.

The concern is that, and I'll remind us from three weeks ago, that any connection to a Bluetooth device, even absent any pairing, does actually create a protocol flow. It's called L2CAP. L2CAP is the lowest level protocol which has to be established before you can do pairing, which implies that there is a handshake and a protocol level connection, even without pairing. What pairing does is establish, allow, essentially, the establishment of a secret shared symmetric key which the Bluetooth devices on each end of the connection use in order to drive their crypto system, in order to turn the plaintext that they would be sharing into ciphertext. But the packets always, even when they're encrypted, they contain the MAC address, the Bluetooth essentially ID, the hopefully unique ID of the Bluetooth device.

It turns out that one of the problems has been that many of the Bluetooth IDs historically were not as unique as they should be. In some cases the manufacturer shipped them all with the same one, which was a problem because the first 24 bits of the ID identifies the manufacturer. The second 24 bits is supposed to be unique for that manufacturer. But that meant that, if someone noticed the make and model, or just the brand of device, they could often guess your ID. And it turns out that even a nondiscoverable device will respond to its correct MAC address even when it's in nondiscoverable mode, which allows you to establish a connection to the device if you know its MAC address, which normally you're only able to get because the device won't respond to an anonymous query saying, hey, can anybody within my coverage range hear me? So, okay. So things have gotten a lot better since then.

BlueJacking, which unfortunately was - when we think of BlueJacking, we would think "Bluetooth hijacking" because that's what is implied by the use of the suffix "jacking." Turns out it has nothing to do with hijacking. It's that it was done first by a guy whose name was Ajack, so he named it BlueJacking because he had come up with it. It turns out that was nothing except the ability to send unsolicited text messages to someone. So some of the early devices would accept unsolicited text messages. It was just the ability to pop something up on someone's screen. And often these text messages would identify the name of the sending device. So if you named your sending device something, for example, like "I'm watching you," then that's what would pop up on the screen and upset people. So BlueJacking turns out to be much less of a concern than, for example, BlueSnarfing, which...

Leo: Which sounds so much worse anyway.

Steve: Exactly. Now, snarfing is 'Net slang for unauthorized copying. When you snarf something, you're essentially sucking in information which is unauthorized.

Leo: And afterwards you have to go "num, num, num."

Steve: Precisely. So there's an interesting site, www.bluesnarf.blogspot.com, that talks about BlueSnarfing and how, when this was available, when this could be done, the Bluetooth calendars, contact lists, email, text messages, photos, basically the contents of people's phones were available. And, for example, we all remember, in fact we've talked on the podcast about how Paris Hilton famously had her phone BlueSnarfed. That's what was done to it. She left it in discoverable mode. And in her case there was a bug in the implementation of Bluetooth on her make and model of phone that exposed it to BlueSnarfing. So in this case it required both being discoverable and having a problem with its Bluetooth protocol stack.

It turns out that the Bluetooth technology implements something that will be familiar to us old-school modem users. You remember, Leo, the famous Hayes AT command set. The "AT" was short for "attention." And the idea was that you wanted to be able, with a modem, to mix commands and data through the same channel, that is, that you didn't have, like, a control command channel separate from the data channel, so that you needed a way in order to mix the commands and the data together. And so the so-called AT command set was originally conceived to allow commands to sort of be intermixed in data and have them treated properly. And that AT command set has survived and been extended over time.

And so a lot of what I ran across when I was looking in detail at this hacking, many of the hacking tools require Linux. They were all done on the Linux platform, and tools were developed; source code is freely available. But you saw a lot of AT commands passing back and forth through, essentially, the term that they used for all this was BlueBugging, which was sort of a catchall for the ability to read/write the phone's SMS store, get read/write access to the phone book. And this was done through access to the Bluetooth AT command set which, if the phone was discoverable, and if there were some known problems with it, that sort of gave a low-level hacker access to this. Interestingly, there are not many high-level tools, no simple-to-use GUI tools for this. It all sort of stayed down in the hacking level.

Most of these things have now failed to be useful. I would say anyone who's purchased a Bluetooth phone in the last couple years, or who's kept their firmware, their phone firmware up to date, really doesn't have much to worry about. There are penetration-testing software suites available, which you can either install on phones or on personal computers running Linux, which will poll the area, look for discoverable phones, and then essentially fingerprint the phone or, as they call it, BluePrint the phone in order to determine...

Leo: Of course they do.

Steve: Of course they do - the make and model of the phone. There's a protocol called SDP, Service Discovery Protocol, which, if your phone is discoverable - you're able to make that first low-level connection over that L2CAP protocol. Then the service discovery protocol enumerates which services that Bluetooth device offers. And turns out that there's a lot of additional information that is leaked there which gives the hacker more of a foothold.

I talked about this HeloMoto attack, which is a classic example of the bugs that were unfortunately in the early Bluetooth implementations. Reading from the description that I found of the HeloMoto attack, it says that it "takes advantage of the incorrect implementation of the 'trusted device' handling on some Motorola devices. The attacker initiates a connection to the unauthenticated OBEX Push Profile, pretending to send a vCard." And you may remember, Leo, remember that OBEX was that vCard protocol...

Leo: Oh, yeah. Right.

Steve: ...that was around where you were able to, like, beam somebody else essentially the contents of your business card.

Leo: It was a Microsoft thing; right?

Steve: Don't remember.

Leo: Seems like it. Maybe not.

Steve: Probably they had some partners because I remember certainly non-Windows devices did support that. And so it turns out that, if you initiated a connection through OBEX, pretending to send the user a vCard, and then interrupted the sending process prior to it being finished, there would be no alert on the phone because you hadn't finished what you started. However, this required no interaction on the receiving end. And as a consequence, the attacker's device was stored in the Motorola phone's list of trusted devices.

Leo: So it's done. You always have access from then on.

Steve: Exactly. Then you're essentially trusted, and that allows you to bring up a trusted connection. And then you're able to use those AT commands to dump the contents of the contact list, to send and receive SMS messages. Basically, you turn their phone into your modem. And also the database in their phone you have full access to. By initiating that OBEX send them a vCard and abort it before its being done, because this wasn't happened properly in Motorola devices, you were then added to their trusted list.

Now, the one thing, I mean, so basically I'm sure Motorola has fixed this years ago. That's what the HeloMoto hack had been. There are people who do something called BlueTooone to, essentially, they're talking about tuning as in tuning an antenna, where they use a high-gain antenna to hugely increase the range of Bluetooth dongles, where they'll take a little Bluetooth dongle apart and essentially hook up a coax cable to it, to a high-gain antenna. We have talked about how Bluetooth sort of uses its limited range as one aspect of security, the idea being it's sort of, people feel comfortable with the, oh, well, this only goes 10 meters, so I don't need to worry about attacks from further away than that. The idea being that, if this was global in distance, you'd be much more subject to hacking than just being a personal area network. So this BlueToooning does weaken the argument that Bluetooth has limited range because there have been people successfully hacked over distances up to, for example, a mile away. Now, the one thing...

Leo: That's a - wait a minute.

Steve: Yeah, a mile.

Leo: I thought Bluetooth was 10 meters.

Steve: Yes. However, there are three different classes of radio power, Class 1, 2, and 3. And Class 1 devices do operate over a larger distance by default. But just as in the case with WiFi, we've also seen this with WiFi, you can take - remember the famous Pringles can with WiFi. And essentially it's just the size that you want in order to create a directional antenna. So inherently, Bluetooth is radiating in all directions. And so when it's omnidirectional, you get about a 10-meter radius. But if you took the same amount of power and made it unidirectional, that is, aimed in a tight beam, then instead of the power radiating in all directions, you're forcing it down a specific direction. And you can, just by doing that, get much more distance out of the same radio.

Leo: Interesting. Wow.

Steve: Redfang is the name for...

Leo: I love the creativity of hackers. I just...

Steve: Yeah, Redfang. Don't know why it's not Bluefang. I guess they got tired of blue, finally. Redfang finds devices that do not want to be discovered. In other words, it finds Bluetooth devices that have not been left discoverable.

Leo: Ooh. That's a problem.

Steve: That's a problem, and it's still a problem today. Now, what's interesting is there's been a huge evolution since Bluetooth was designed in what hackers have access to. We've talked a few times about the notion of a wideband receiver, that is, the frequency hopping that Bluetooth does, the so-called FHSS, Frequency Hopping Spread Spectrum technology, where the master Bluetooth device uses its own MAC address, coupled with the clock and the shared secret, which it shares with the slave device, which then knows the clock and knows the MAC address of the master. That determines the pseudorandom sequence of hopping. And Bluetooth hops 1,600 times per second over 79 different channels, the idea being that it was believed that once upon a time this would, like, render it impossible or, you know, much more difficult to hack.

Well, it turns out that this is all being done within a relatively narrow band. So all you need is a radio which is able to simultaneously receive all the signals within that band. And some digital signal processing makes this whole frequency hopping a joke. I mean, it doesn't even care that it's doing frequency hopping. It's essentially listening to the entire Bluetooth spectrum and sucking the entire spectrum in at once. And using very inexpensive, off-the-shelf technology now, it's possible to essentially monitor all 79 channels at the same time so that this frequency hopping spread spectrum, as a security measure, is rendered completely useless.

Now, the attack that Redfang uses is a brute-force attack on the MAC address of the Bluetooth device. The MAC address for Bluetooth is identical in size and composition to the MAC address we're familiar with with Ethernet, which is most significant 24 bits is manufacturer, least significant 24 bits is the device ID. However, it turns out that, when the device IDs were looked at more closely, it was found that the manufacturers, many manufacturers had gotten sloppy. They were, for example, reusing only a subset of the potential 24 bits. 24 bits is 16 million possibilities. So 16 million, while a lot, is not an impossible number. Meaning that, if you knew, if you could, for example, see that somebody was using a BlackBerry or was using a Sony or was using a whatever well-known brand of phone, hackers have indexes of all of the MAC addresses associated with a phone.

Now, if the person is on the phone, then you have no problem discovering their MAC address because the MAC address is in the clear. It's never encrypted. It's part of sort of the wrapper outside of the encryption. So Redfang is only necessary to be employed for devices that are not discoverable and not in active use. But if you know the manufacturer number, then you've got the first 24 bits cold. Then you only need to deal with the second 24, which gives you 60 million possibilities. The problem is, for example, all of the Sony Ericsson phones start out with an E, that is, the E is the first nibble of the three bytes that compose those 24 bits of device ID. Well, if you know it's an E, that eliminates those four bits. So now you're down from 24 bits to 20 bits, and you've eliminated 16 times the number of possibilities, so you've dropped it down to a million. And we know that a million, that is, 20 bits, is no security in this day and age.

So given a relatively short time, it's possible to get access to a phone through a tool like Redfang, even if the phone is not discoverable. Once you have that, though, you are limited to what you can do without having established a pairing with the phone. And the good news is, as far as we know, there are no existing security compromises for very popular present-day Bluetooth-enabled devices. The stacks have solidified, the protocols are established, and Bluetooth, to a much greater degree, is not something being thrown into the mix afterwards. It's something that, especially in today's security climate, there's much more attention being given to.

So that brings us to Carwhisperer, which is still in use and still a problem. You may remember that three weeks ago, when I was talking about pairing, I mentioned that, in order to do secure pairing, we must have an exchange of information out of band, that is to say, through a means that an eavesdropper cannot detect. And the way Bluetooth devices do that is they'll put up like a six-digit code on one of the screens and ask you, the pairer, the human person doing the pairing, to enter this code into the keyboard of the other device. What you've done by doing that is you've synchronized the devices. You've informed them of something unique which an eavesdropper cannot know. So you've seen the screen, and you've manually moved that information, not over the air wirelessly, but using your own body you've moved that information to the other phone. That allows them to perform a maximum security pairing. And as far as everyone knows, there's no way to break that.

The problem is that Bluetooth, the Bluetooth spec deliberately scales security down as necessary to deal with devices that lack keyboards or screens. Now, if you're pairing a Bluetooth keyboard, just a keyboard with no display, then you're still secure because, if you're pairing it with a computer that has a display, then that's where it's able to say, type the following thing into your keyboard. So you type it into your keyboard. Now, that's not going through the air, that is, no hacker can capture your keystrokes. That's going into the Bluetooth stack on the keyboard, and it's being absorbed by the pairing technology, the pairing protocol. Then the result of that, mixed with the computer's MAC address and a long pseudorandom number and the master device's clock, that generates the pairing. So no attacker eavesdropping is able to crack that.

The problem is, if we come back another notch to a device that even lacks a keyboard, and/or devices with no display, so they're unable to show you a number. Now, the good news is, to deal with this weakness, there are devices now which do not use a fixed pairing number. But unfortunately, just the other day I was messing around with some Bluetooth speakers, and the manual said, when asked for the ID, the passcode of this device, enter 0000. That's what it is.

Leo: I see that all the time. Or 1111, yeah.

Steve: 1111 or 1234. So the problem is that Bluetooth headsets are probably the biggest culprit.

Leo: Yeah, because they have no keyboard.

Steve: Exactly. They've got no keyboard. They have no display. And it turns out that, even today, if they are left in discoverable mode, or if cars, if the technology in the car for doing, like, hands-free phone is left in discoverable mode, and it turns out that, for the sake of user convenience, many of them just have that on by default, this

Carwhisperer technology, which people have been successfully using from overpasses on freeways, are able to eavesdrop on the occupants of the car, just with cars driving by. And so that is...

Leo: That's really scary.

Steve: Isn't that? Yes. I mean, because we were talking about, we sort of laughed it off three weeks ago, saying, oh, well, if you're close enough to have a Bluetooth connection, you're close enough to hear the person anyway. Except, if they're inside their car, and they've got Bluetooth technology for doing hands-free, and that's discoverable, then because there isn't, probably isn't a unique ID, it's possible to pair. And unfortunately, even 0000, well, what? That gives us 10,000 combinations - 0000, yeah, four zeroes to four nines, 9999, so that's 10,000 combinations. So in a relatively short time you can try all of those pairings with a device. And even if it was a random number, if it's not a long random number, you're vulnerable again.

And so that is an existing problem. If any of these devices that used a fixed or a short pairing key are left in discoverable mode, then you can pair them, typically without any notification of their owner. And I saw demos while I was researching this of people sending audio out and receiving audio back from those devices.

So the conclusion from all this is, with Bluetooth, pairing, the security of pairing, is our only line of defense. The technology was designed with that understanding. That is, remember from three weeks ago that Ericsson, who initiated this originally, then generated a handful of followers until this thing exploded, and there are more Bluetooth radios now in existence than there are 802.11 WiFi radios.

Leo: Oh, that's interesting, wow, yeah.

Steve: Yeah. I mean, it's just been phenomenal. They deliberately wanted to create a consumer-friendly system. So the only thing that I could wish for is that there was better discipline on the part of the manufacturers for not allowing pairing to stay enabled by default. Yes, it's an ease-of-use thing. I can see why manufacturers do it, because they want to minimize their tech support calls. But it makes them vulnerable to any mistakes that they may have made anywhere else in their software. You're just much better off if you're not pairable.

Certainly you are trackable if you're pairable. And that's one thing to remember is Bluetooth tracking is being done. There are devices being sold which, like by marketing-related companies, where they advertise that, stick this in your showroom window, or stick this by the door, and we will log all the Bluetooth-equipped people, customers, who come in and out, and we will let you know if they come back, having visited once before. So it's very possible for someone to say, "Oh, hi, you were here a week ago," which would be a little unnerving, actually. And the way that would happen is you had left your cell phone Bluetooth on and discoverable. So pairing is the only line of defense.

Historically, pairing could be bypassed, just due to mistakes that had been made in the software, in the Bluetooth stack. I did run across some references where hackers were talking about how it was easy to pair hack if the device had ever been paired with anything else. Meaning that if the pairing database had an existing pairing in it, then it was easier to hack. So what that argues for, and this is another piece of advice I would counsel, is remove any unused pairings because, apparently, existing pairings which you're not using do create a little wedge point for someone wanting to attack your phone. So rather than just letting them accumulate forever - which they would otherwise probably tend to do. I noticed the other day I had four or five in a little laptop of mine from some mice that I was experimenting with once, that I had long since stopped using. So now it is useful to remove pairings you do not need.

Leo: Good to know, yeah.

Steve: The other exploit against Bluetooth involves someone briefly having access to your device. So, for example, kids or spouses or ex-boyfriends or girlfriends or something, you want to make sure that you have removed - again, it's another reason for removing unnecessary or unexpected pairings from your device. If someone briefly had access to your phone, paired it with their Bluetooth radio, and then gave it back to you, then that pairing persists. So one vulnerability is that pairings would sneak into your Bluetooth-equipped device without your knowledge, which would then give them access to that device in the future. So it's worth looking to see - and as far as I know, all display-equipped devices will allow you to look at the enumeration of devices that have been paired with it. And it's worth just removing any that you don't need. It's very much sort of like changing your password after employees that knew it have wandered off. You just don't want to allow that access to persist.

Let's see. So remove any old ones. And finally, as I said three weeks ago, turn off the Bluetooth completely if you're not needing it. Just not having it on at all will save battery power. And that's the only real way to be completely invulnerable to any kind of mistakes that the manufacturer may have made, any mistakes, human error, pairings that were left in that create vulnerabilities. Just having the radio off completely is really your final line of defense.

Leo: Yeah, most people don't do that because you want it to join automatically, when you get in the car or when you get within range, you want it to join up. So turning off the radio is, hmm.

Steve: Yeah, I know. Again, it's a function of use. If you're a person who's not using a Bluetooth headset all the time...

Leo: Turn it off.

Steve: ...then turn it off. If you do need it on for convenience, you really do need to leave it on, but absolutely make sure that it is not discoverable because that can potentially create some problems.

Leo: Yeah. I mean, if you have a smart phone, you're probably turning - if you want to save your battery, you're probably turning off Bluetooth whenever you can anyway.

Steve: Right.

Leo: You shouldn't have it running for no reason.

Steve: So the takeaway from all of this, from the research I did into the dark side, is that it's technology which is mature enough now that, if you take a few precautions, you're really safe. There are no known really bad security problems today in Bluetooth. I was carefully looking at the dates of all of the hacking stuff that I was finding, checking to see whether these problems had been solved. Just across the board, it's a technology that I'm actually very bullish about. I think it's very useful, very handy. If you keep your phone or your devices nondiscoverable, you're really going to be very safe.

Leo: Good to know. Steve, you're so good at this stuff. Thank you for giving us the - the thing is, I know when people listen to this show, and you cover a topic like this, they know they've got it; that, you know, that's the landscape. There's nothing left off. So thank you for doing that. Next week, Steve Gibson, it's a Q&A.

Steve: Yep, we'll do a Q&A. I'll ask all of our listeners, again, to send any thoughts, questions, comments, stuff, to GRC.com/feedback. And I will check the mailbag and update everybody on any other news of the day. And we'll have some fun dialogue with our listeners.

Leo: Always. I love the mailbag episodes. GRC.com/feedback. While you're at GRC, of course, check out SpinRite, the world's best hard drive maintenance and recovery utility. Check out all of Steve's freebies, too - ShieldsUP!, Shoot The Messenger, DCOMbobulator, Perfect Paper Passwords, the DNS benchmarking utility and a whole lot more. It's all at GRC.com, along with show notes, 16KB versions for the bandwidth-impaired, and transcripts thanks to Elaine. It's all there, GRC.com, Gibson Research Corporation.

I should give you a little plug, since you mentioned your Twitter handle. Steve is @SGgrc on Twitter. He also has a - I don't know, did you tweet with the CES stuff at...

Steve: No, I really didn't find anything that I felt was tweet-worthy. I do want to reiterate that I'm getting great feedback from my Twitter followers. I crossed the 16,000 level a couple weeks ago. And, I mean, I read all of the @mentions that are coming in. And, for example, some of the things that we talked about today I first picked up on from Twitter folks. So I really appreciate that. It's a great way of getting little thoughts and notes back to me.

Leo: And, by the way, Ericsson is Swedish. Nokia is Finnish.

Steve: Oh, that's right. Yup, good.

Leo: That was me, not you. But we just want to forestall some emails for next week. Thank you, Steve.

Steve: Thanks, Leo.

Leo: Have a great week, and we'll see you next time on Security Now!.