



IronKey

Description: Leo and I spend 45 terrific minutes speaking with David Jevans, IronKey's CEO and founder, about the inner workings and features of their truly unique security-hardened cryptographic hardware USB storage device.

High quality (64 kbps) mp3 audio file URL: <http://media.GRC.com/sn/SN-135.mp3>

Quarter size (16 kbps) mp3 audio file URL: <http://media.GRC.com/sn/sn-135-lq.mp3>

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

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

This is Security Now! with Steve Gibson, Episode 135 for March 13, 2008: IronKey. This show and the entire TWiT broadcast network is brought to you by donations from listeners like you. Thanks.

It's time for Security Now!, 135 episodes strong, and a good one ahead today. Hello, Steve Gibson.

Steve Gibson: Hey, Leo. Great to talk to you again.

Leo: We are going to be talking in just a little bit with the folks from IronKey.

Steve: In fact, that's the title of this episode, IronKey.

Leo: Old IronKey. And Raymond Burr need not apply. So this is a - we've talked about it peripherally, but I know you wanted to get the guy who invented it on the show. This is a USB key that has special built-in encryption and special hardware protection to keep your data private. And it's very interesting.

Steve: And we learned that it's more than we thought it was.

Leo: Yeah, that's kind of neat. So that's coming up. Do you have any addenda or news?

Steve: Oh, yes, I do. You bet. First of all, it was a relatively quiet week over in Microsoft Land. However, after our podcast went public last week on Friday, Sun released updates to their so-called JDK and JRE, their Java technology, which had some serious vulnerabilities that anyone using the JDK or the JRE really need to take a look at. They had both privilege elevation and buffer overflow vulnerabilities. And, for example, reading from their announcement, they said, for example - "A vulnerability in the Java runtime environment may allow JavaScript code that is downloaded by a browser to make connections to network services on the system that the browser runs on through Java APIs. This may allow files that are accessible through these network services, or vulnerabilities that exist on these network services, which are not otherwise normally accessible, to be accessed or exploited."

So the whole idea, of course, was to create a sandboxed environment and keep Java stuff within its own environment. Well, this says that there is leakage from that, and that an attack vector is the user's browser with JavaScript. So the vulnerability, of course, is you go to some website, and they're able essentially to bypass your router and firewall and get a connection to your internal network. So that's not good.

Leo: Microsoft did have a patch for Mac users. Macintosh Office 2004 had a serious vulnerability that allows an attacker to overwrite the contents of computer memory with malicious code. So if you use Microsoft Office, it's not an auto update. You've got to run the Office updater.

Steve: Oh, and in fact there were, you're right, there were also some updates to the regular Windows-side Office suite, as well. So there were some Office updates across the board, but not mainstream Windows OS.

Leo: You can add, and I think it's a good idea to do that, if you use Office you can add Office updates to your overall Windows Update. I can't remember how to do that. But I think if you go to WindowsUpdate.com you can figure that out.

Steve: What you do is you actually switch from Windows Update to Microsoft Update.

Leo: Ah, okay.

Steve: And Microsoft Update is sort of this omnibus now that does SQL stuff and Office and Windows. And so it's absolutely what you want to do in order to keep basically the whole Microsoft suite current.

Leo: Good. Also issues in Office 2008, mostly bug issues, stability. So that's for Mac, as well. Worth updating if you have Office on the Mac.

Steve: Okay, there were a couple TrueCrypt things, as well. We talked - one of our Q&A questions last week, and you and I actually talked about this after or as part of our Q&A, remember the guy that wanted automatic update of his TrueCrypt volume, but the TrueCrypt timestamp was being maintained at its original creation date, not showing that it had been updated. So Jungle Disk was not seeing a change, and the Jungle Disk backup was not triggering. I got a nice note from the creator of Jungle Disk saying, hey, Steve, several of your listeners contacted us to ask about that. So he said, I wanted to let you know that there's a timestamp option offered in TrueCrypt. It's "/m ts" for - obviously for timestamp. And what that does is that tells TrueCrypt not to fudge the modification time, but to leave it real. In which case Jungle Disk then does pick up on the change and will do the backup. Also last week we had a question about random hashing and rainbow tables, you'll remember. And someone posted over on the GRC forums that because TrueCrypt was using salt in their hash, there was no problem with rainbow tables.

Leo: But I always put salt in my hash.

Steve: I figured you did. You seem like the kind of guy who would want to salt his hash.

Leo: So by salting your hash, the rainbow tables can't be - you can't make a standard rainbow table.

Steve: Exactly. And, for example, that's something that WPA also does as part of its cipher suite. And it makes sense, the idea being that part of the randomness, the entropy that you're creating when you're creating the volume, is to create a chunk of entropy which is so-called "salt," which is mixed in with the data that you're hashing so that essentially you end up with custom hash results that are not like anybody else's hash results. So unlike rainbow tables, for example, where there'd be a rainbow table for MD5, the generic MD5 hash, all you have to do to completely destroy those is add a little bit of salt. So that's what TrueCrypt is doing to maintain itself. And then the big news.

Leo: And then.

Steve: We are now at TrueCrypt v5.1, which has added hibernation file support.

Leo: Wow, that was fast.

Steve: It did not take these guys long.

Leo: See, open source ain't so bad. There's some advantage to open source.

Steve: So we now - the one real thing that concerned people was that the hibernation file would be a nonencrypted snapshot of RAM. Now it is encrypted. And they've even solved the annoying Adobe/Macromedia problem. They shrunk the size of the bootloader, and they're now able to put two copies of it on the first track of the hard drive. If one of

the copies is damaged by something like Macromedia or Adobe, I think it was some PDF component that we learned about last week, well, if one of them gets damaged, the other one is automatically used. So you don't end up being crippled by doing that. And they reimplemented AES in my favorite programming language.

Leo: What's that?

Steve: Assembly language. They took it from C to Assembler, and it now runs between 30 and 90 percent faster than it did before.

Leo: Wow. I guess, you know, people still can benefit from optimizing in Assembly. You used to do that all the time, especially I/O. You'd optimize those little things than ran - the loops and stuff that ran a lot in Assembler. And then it got kind of to the point where these processors were so fast you could do everything in C or whatever high-level language you use. But, you know, in something like encryption there is a lot of processor activity, isn't there.

Steve: Well, Leo, that's my favorite language. It's all I write in, so...

Leo: I know, I know.

Steve: And then there was one little weird blurb I just want to sort of bring to our listeners' attention. And that is that CNN covered a story of they met with three Chinese hackers in their 20s who have claimed to have hacked many of the world's most secure and sensitive sites, including downloading information from the Pentagon after hacking into the Pentagon's network. They claim that they have been paid by the Chinese government to do this.

Leo: Wow.

Steve: The Chinese government, of course, denies that and says that's ridiculous, we've never paid anyone to do this kind of work. Okay, couple that with the fact that the FBI released the news recently that they've participated in over 400 seizures of counterfeit Cisco networking equipment totaling more than \$76 million, which has been filtering into the United States from China. So this is counterfeit, for example Cisco routers, not made by Cisco, coming into the U.S. from China.

Leo: And you can bet they have backdoors.

Steve: Well, that's exactly the problem. In 2004 some counterfeit Cisco switches ended up in one of the Navy's secure facilities. So, I mean, I use Cisco routers. I know how sophisticated they are. They call it IOS is the OS that runs in these routers. There could be anything in those routers, and there would be no one way to know that there weren't backdoors in these routers. So, I mean, it becomes a little frightening.

Leo: No kidding. Wow. No kidding. So make sure you're getting a real Cisco router. I have to believe that the Chinese government - that's cyber warfare. And you have to believe our government's doing the same kind of thing to other hostile, potentially hostile governments.

Steve: It's sad and freaky, but I think you're probably right, Leo.

Leo: Well, and in a way it's not because at least, you know, I mean, it's a kind of warfare that people don't necessarily die from. And it's certainly where it's going to happen. It's more like an economic warfare. It's not a good thing, but it's certainly - I would expect it to be going on.

Steve: And I guess that's where the world is headed.

Leo: Yeah. Yeah.

Steve: Speaking of which, SpinRite fixed a WiFi problem the other day.

Leo: All right, I like that. How could it possibly - I'll bite. How could it possibly do that?

Steve: We got a fun email on March 9th from Steve - I'm afraid to pronounce his last name - Diyorio. I think that's it, Steve Diyorio. And for Elaine's benefit I'll spell it, D-i-y-o-r-i-o. Anyway, so his subject was "SpinRite Fixes Weird Wireless Problem." And he said, "Hi, Steve. Just wanted to drop you a note to say thanks and thought this might be an interesting one to mention on a future episode of Security Now!." Yeah.

He says, "I have Ubuntu Linux installed on an HP laptop and have been extremely happy with it for quite some number of years now. I've noticed that my system has been booting and running a bit slower lately (a feature of Windows that I thought I'd left far behind)." He said, "And more recently my wireless suddenly refused to connect. I tried everything within my knowledge to get the wireless working and spent countless hours scratching my head, trying to figure this mysterious problem out. In light of the slowness, I figured I would run SpinRite at level 4 to do some maintenance on the whole hard drive. Lo and behold, after fixing some errors on my drive, my wireless is working again. Thanks for a wonderful product. I will share my experience with everyone I meet." And we've shared his experience with everyone who listens to this podcast.

Leo: Why would a - I guess was there a flaw maybe in the file or...

Steve: Well, just, I mean, this is a bizarre circumstance where there were problems on his drive in a sector of something to do with his wireless networking. And SpinRite fixed the sector that was erroneous and brought his wireless back online.

Leo: You know, one thing I do notice, and I get a lot of calls from people who say everything's slowed down, slowed down, what's going on? And of course naturally you say spyware or viruses or, you know. But one thing is, if a drive is flaky, the operating system can spend an inordinate amount of time trying to read it. It may eventually read it and continue on, but that can really slow a system down, can't it?

Steve: Well, in fact, yes. One of my very good friends from many, many moons ago was experimenting with some servers. He listened to Security Now!. And he asked for a copy of SpinRite to sort of see what was going on with hard drives. He learned that the vibration of the fans in the server chassis were enough to throw the hard drive heads off track. So that even though everything was working, the system was running much more slowly because the drives were having to go around and around and retry their reads in order to get the data off the drive. And the problem is, of course, the track densities are so high now in order to get these multi-hundred gigs of data in such a small space, that a little bit of vibration will end up coupling into the drive. And so it turned out that just by taking the drives out of the, like, off the server chassis and suspending them, suddenly the servers ran much faster.

Leo: Now, how would he use SpinRite to find that out?

Steve: Well, he was able just to perform more experiments with it and watch SpinRite run in sort of a uniform fashion. And I think he told me he pushed down on the drive, coupling it mechanically more tightly to the server, and he saw it slow down.

Leo: So Dynastat, if Dynastat is running you can kind of get an idea of the reliability or performance of the drive.

Steve: Sure.

Leo: Yeah. Oh, that's a clever diagnostic. I think as computers get more complex we're going to have to see more and more of these kinds of people who are really good at this kind of deduction because they're such complex systems, and there's so many interactions. You can't just say, oh, well, it's got to be that anymore. You really have to, I don't know what, try stuff, I guess.

Steve: Well, and you know, when we talk about network security, cyber terrorism, you know, the notion, I mean, these three Chinese hackers said that no website is safe. They have to know what the software is that the website's running. Then they dig into their bag of tricks, and they know where vulnerabilities are. And the problem is, as this stuff gets incredibly complicated, it becomes more like an analog domain than a digital domain because it gets soft. It's just not as hard. It's not a yes or no, this packet can or cannot go. It's like, well, wait a minute, we'll use this protocol and come in through some network that this has an affiliation with and hop around a few times and get in.

Leo: I kind of hate it when there are errors on my server, and the server announces

what it is, its exact version number, what software is running - this is Apache 127, and you've got a problem with the Perl module CPAN_43. It's like saying to the hacker, this is what we're running.

Steve: Oh, you mean like it actually spits it up on the web page.

Leo: Yeah, Apache does that. Kind of bugs me. I mean, I guess with Nmap or Nessus or something you could figure out what server's running anyway. But I don't know, it seems like they should be a little more quiet just for security reasons.

Steve: Yeah. The less you say, the better.

Leo: I learned that from stealth mode at ShieldsUP!, GRC.com. And we're going to talk a little bit about Audible, and then we're going to get to our guest. We're going to talk with Dave about IronKey. He's the CEO of IronKey and then really kind of an amazing amalgam of security technologies all built into one USB key. I think you're going to be interested in this. So introduce our guest here.

Steve: Well, this is David Jevans, who is the founder and CEO of IronKey Corporation, who we're going to spend a nice chunk of time with, talking about IronKey, what it is, how the technology works, and all the things that it is way beyond just being a secure USB flash drive.

Leo: Very cool.

Steve: So, yeah, David, I wanted to get directly from the creator's brain what the genesis was, the motivation for IronKey. And I guess really, I mean, it competes with software-based encryption solutions. So I wanted to get a real good sense for our listeners, who've been very interested in IronKey, and in fact we're interviewing you and talking to you and doing a whole show on it because of demand from our listeners. What has IronKey got that no other solutions do?

DAVID JEVANS: Well, I appreciate the chance to speak with you guys. And you've thrown quite a lot of good questions out, so let me try to take that stuff one by one. I think the first thing you asked was around how we came up with the idea and what we were trying to accomplish. And I'll be quite honest with you. What we started out with is certainly not what we ended up with. And I think that's probably true of most businesses out there. I've been in the security business now for over 10 years. And one of the things that I do is I run a nonprofit organization called the Anti-Phishing Working Group.

Leo: You run that? That is a great group.

DAVID: Yeah, thanks.

Leo: I've referred people to that site many, many times, especially because of your archive of phishing emails. For people who don't believe in phishing or say, oh, I would never be fooled by that, they go to those archives, they look so real that it's very convincing.

DAVID: Yeah. I mean, it's a group effort. We now have over 2,000 member companies and government agencies and law enforcement groups that work with us.

Leo: How did that come about? Was this your idea, or...

DAVID: It was. In 2003 I was working at a company called Tumbleweed which makes antispam and email encryption. And, you know, I started seeing that strange spam that looked like it was coming from PayPal, and it wasn't; or looked like it was coming from Citibank, and it clearly wasn't. And I looked into it, and we decided to publish a little report because we saw quite a big spike of it at the end of 2003, just before Christmas there was a huge spike. And we got a lot of interest. We had quite a number of banks and major ITs come together, and we started the Working Group. And to be honest, I thought back then phishing would have been solved by probably mid-2004.

Leo: Why hasn't it been? Because first of all, I can see why - I'm sorry. I don't mean to hijack this, Steve. But I think it's an important subject. We'll get to IronKey in a second. I can see why end-users, consumers are fooled by them because if I get an email that looks like it's from PayPal, smells like it's from PayPal, and it says they're going to cancel my account unless I click this link, I click the link and it goes to a web page that looks like PayPal, I could see how you'd be fooled by that. But in every case that link has to go to a bad website. You would think that those sites would get shut down so fast that it would just not work very well.

DAVID: Well, I'll tell you, back in 2004 there were no shut down companies, and nobody admitted there was a problem. People didn't understand it. There was no email authentication protocols. It wasn't on the radar of anybody. And I think that's one of the things that we've been successful with is building a lot of awareness. One of the things that we have done, though, is we've tracked the bad guys and how they've changed and how they've grown and how they've become more and more professional about stealing data, stealing credentials, stealing passwords. I mean, they have gotten so effective. So to your point about takedowns, now there are takedown companies and people know how to get sites taken off the Internet. So what the bad guys do is they go and abuse CMS. They implement something called "fast flux" where they basically change the server every 10 seconds where it's being hosted.

Leo: Oh, wow.

DAVID: Right. So everything you do, the bad guys who are making hundreds of millions of dollars a year off this cyber crime, they move a whole 'nother step ahead in distributed systems technology, massive botnet armies to get around spam filters,

I mean, it's a continuing evolution against well-funded bad guys.

Leo: Well, thank you for the Anti-Phishing Working Group. It's AntiPhishing.org, phishing with a "ph." And obviously it still needs to be there, which is sad to say.

DAVID: Yeah.

Leo: So that was the inspiration.

DAVID: Well, yeah. So one of the things that I get out of that, you know, it's a nonprofit group, but one of the real values I have out of it is I get to see what the bad guys are doing probably in advance of most other people because I know a lot of the security researchers. I work with the financial institutions. And so you get to see the trends and the sophistication. And it became clear to me that to really start protecting the infrastructure and to protect people's privacy and their passwords and really the stability of the financial system, you need some pieces of hardware. You can't just do everything in software. So you need either hardware to protect against keyloggers or hardware to encrypt data or hardware to do strong authentication on the Internet so that, if they steal your password, it doesn't matter, they still can't get in. And that the person you're logging into, they know who you are, and it's really you.

And so that was really the genesis of the whole IronKey project was, you know, we see where it's going. Let's put something together that can be used by millions of people because it's easy to use, it's not expensive, and it does something that they know how to use, and that we can build it as a platform to help protect against more and more of these threats. So, you know, the first thing was, hey, there are lots and lots of flash drives out there, in fact well over a hundred million of these sold every year. Why don't we make a really great one with hardware encryption, no software needs to be installed, make it cross-platform, but add some magic in there which is authentication so it can be used to do a lot more than just a regular flash drive.

Leo: So why flash drives, though? Do you think that they're particularly vulnerable?

DAVID: Well, so there's two aspects to it, I think. The first one is that, if you look at an enterprise context, people use flash drives all the time to back data up. They go on sales calls. They're going from hospitals to clinics. They're brokers moving all around the country. There's lots of different use cases for why people use them. And as you know, people lose them. They get stolen. The laptop gets stolen. And, you know, there's just increasing regulations about making sure everything's locked down.

Leo: Right.

Steve: And of course we're also more and more seeing the requirements at various

levels to make sure that anything that can be stolen is encrypted.

DAVID: Absolutely. And, you know, we can get into why hardware encryption is the right answer versus software encryption in a minute. But that was clearly one market requirement right there was make these things really easy to use, and you can't mess it up. Anything you write on it is for sure secure.

Steve: One question I saw was that you talk about being able to do a secure backup. And so I'm wondering whether that works by pulling the encrypted data off of the flash, not going through the decryption on the way out.

DAVID: It actually does not work that way, although that is a very good idea on how to do it. The reason it does not work that way is because the encryption keys on the IronKey devices are not exportable.

Steve: Good.

DAVID: You can't pull them off. Right. Which means that malware can't copy them. A malicious user can't steal them. You can't have a - they're not vulnerable to cold boot attacks that we've seen out of Princeton, some research in the last two weeks on that about hacking software encryption. So if I can't pull the encryption key off the device, but the data is sitting on your computer encrypted with it, if I lose my IronKey, and I go buy a new one that's got an encryption key, how do I decrypt the data?

Steve: Right. So you have to run it back through the cipher chip in order to decrypt it and then reencrypt it with something that's outside of the chip.

DAVID: That's correct. You have to generate another AES key that is a strong AES key, a random AES key, generated by the hardware random number generator. You encrypt the data with that, and then you further encrypt the AES key with several rounds of a hash based off of your password. And you have to store that with the encrypted data. It is how all basic software encryption has to work.

Steve: So the cipher chip is a little bit like the TPM technology that we have discussed at length in that you are limited in what you're able to ask it to tell you. And for example, it internally generates these ciphers, I mean the keys, which are used by the IronKey. And so you could tell it to do encryption and decryption, but there's no facility for you to say give me the key that you have inside.

DAVID: That is correct. And you can't have the hardware allow that and then pretend like I won't expose that API.

Steve: Right.

DAVID: Because I know from everything I see in security, if it exists, people will try to hack it. And they will find a backdoor if it exists. You cannot have any backdoors because they will get found.

Steve: And I guess the other thing that I found intriguing, certainly you make the point that you're going to prevent brute-force attacks by counting down the number of incorrect attempts to access the key through a password. And there's language in various places on the site and in the help files about this thing self-destructing.

DAVID: Right. So let's talk about the brute-force prevention first. If I find a hard drive, a

flash drive, a computer, and it's been encrypted with a software encryption package, you know, there are freeware ones and there are ones that you can buy, effectively the key is stored somehow with the data and then encrypted with some derivation of your password, or the key itself is in fact a derivation of your password. Which means if I want to break an encrypted drive, I don't go try and break AES encryption because, I mean, I've done the calculations. Even with 100 million computers I can get it down to, you know, a couple hundred thousand years maybe. It's not practical. What you do is you guess passwords. You can build a machine for \$10 million that will guess 40,000 billion passwords a second.

Leo: Really?

DAVID: Yeah, you can crack strong, you know, you can crack things with 10-character passwords in three hours.

Leo: Wow.

DAVID: Absolutely. And in fact with some of the NVIDIA supercomputers it's probably less than \$10 million now.

Leo: Wow. How many a second? How many a second?

DAVID: You can guess 40,000 billion passwords a second for about 10 million bucks.

Leo: 40,000 - that's 40 trillion.

DAVID: Yeah. So that's the way to attack software encryption is password...

Leo: How can you get it even to respond that fast? I guess you have to...

DAVID: Well, you just do this all in parallel. You just build a chip with a hundred cores on it, and you put a hundred chips per board. And then you could - and, you know, each chip costs maybe five, 10 bucks. And then you just put 10,000 boards in a computer room.

Steve: And then you stick it next to Niagara Falls for power and cooling.

DAVID: Well, yeah, I mean...

Leo: But that's why you said the NSA has to do something like this, Steve. It's just prohibitively expensive for the Russian mafia to do it. But the NSA...

DAVID: Right, or you set up an online service and offer...

Leo: Yeah, that's right. Hey, it's the new SETI@home. Brute-force cracking at home.

DAVID: Yeah. So the point of it is that the key - the way to crack it is through cracking passwords. And to be quite honest you can crack most passwords much faster than that because most people don't use a strong, random, 10-digit password. And so there's commercial password-cracking tools available on the Internet that also work with hardware. And in fact you can get forensic - you can buy computer systems that are multiprocessor purely for cracking passwords. So that's the way to do it. So the defense is you have to have hardware-implemented, brute-force prevention. And the keys have to be managed in hardware and not exportable. And so what we do on the IronKey is the Cryptochip itself manages the password count and the password verification so that when you try to log into it, it will check itself, is this the right password. And a decremptive counter which is stored on-chip in nonvolatile protected memory under layers of metal with differential power attack protection and things of that nature. So you cannot go and replay it. You can't put wires on it to turn it off. You can't recopy NAND Flash. You basically have 10 tries, and that's it. So that effectively makes it uncrackable by password guessing means.

Steve: And when you say "and that's it," what happens after a failed attempt to guess number 10?

DAVID: So on, of course, on try number nine you're warned extensively, this is the end, your last chance. We mean it for sure seriously. You're about to get a shiny nice little doorstop if you do this. But if you enter incorrectly 10 times, you're an attacker and you enter in 10 times, basically what happens is the Cryptochip is disabled, the encryption keys are killed, and then we do our Flash Trash where we basically, as a further step, we erase at a hardware level, at a very low level in the NAND Flash, we erase all of the encrypted data, including all the wear leveling and any ID data for the AES encryption. So it's a much lower level, higher speed way to erase all data than you could ever possibly do with software.

Steve: Very nice.

DAVID: Yeah. All encryption keys, by the way, on the device are also further encrypted. So if somebody did find some way to strip it down and get an electron microscope in there and somehow defeat the anti-tamper and the layers of metal, you still have to go break AES. So we believe it's got multiple levels of protection.

Steve: The one thing that you're not doing that occurred to me, because I've seen reference, and you've mentioned it, to keystroke logging protection, is that this is the same password every time.

DAVID: Yeah. So we're looking at a couple of different things on the keystroke side of things. So what you have to do is you have to look at the threat model about what you're actually defending against. So the first one is I've studied many, many keyloggers; and as part of some research we did for the Department of Homeland Security we analyzed over 60,000 pieces of malware and different loggers. Most of the keyloggers, modern high-end ones, also do screenshot logging. Because they're designed to defeat virtual keyboards. So if you take a look at the good keyloggers out there, they will actually take

a click of - they will take a small screenshot, oftentimes, of the screen around where the cursor is, or a full black-and-white image, so they can see if you're clicking on a virtual keyboard. So if that's an issue, you know, you can't just solve that by a virtual keyboard. But that is one thing that we're looking at.

You could do complicated things like the password's different every time. People have suggested you hold a button, and every third character is your password, or you do a challenge/response and say what's the third letter of your password. And, you know, you could do all that. But it's a usability nightmare. And is it really worth investing a lot of time in writing the software to do that, versus looking at other things, for example, physical external entry of your password.

Steve: Right.

DAVID: So we're looking more down those lines. The keylogger thing you can - and here's the other point, is if it's a generic keylogger, okay, well, it's on your computer, its getting at your flash drive is probably the least of your worries because this thing's on your computer getting your key logs, all your Internet stuff and all that kind of thing. So, and also, if it really wanted to attack you, it's going to copy the - it's custom targeted, and it's going to copy the data off your computer anyway once you unlock the device. So there is certainly a threat, but you have to kind of look at it and go, okay, what is the real threat, what are they going after, and are there other ways to mitigate it that are not quite so obvious.

Leo: So what this is really designed to do is, if you lose the USB key, or somebody takes it from you, not somebody who has access to your system.

DAVID: Yeah. If somebody has access to your system, they can install malware on your computer, I mean, you're dead.

Steve: Right, right. One thing that surprised me, as I was establishing the password for my key, I got - you know how you have the little red and green tag that shows whether your password is okay, it turned green for me after I'd entered four characters. Or five, maybe. But, I mean, I was surprised what a short password it would allow me to use for the IronKey itself. And I thought, well, maybe that's because of this 10 strikes and you're out, guessing doesn't really work approach.

DAVID: Right. So there are two answers to it. Well, I guess there's probably three. But one is usability. So we wanted to make a product that a lot of people could use. So if you put it at eight characters with three upper and two lower, lots of people are not going to remember that, or they're going to write it down someplace, which is not that secure if you write it down, especially if you write it on the back of your flash drive or in your wallet.

The other one is, you're quite right, because there's brute-force password-guessing prevention, it's not the same as a software attack where on software I can guess thousands or hundreds of thousands or millions of passwords in a second. With the IronKey you've got 10 tries. So you've actually restricted your attack surface down to be five or four or 10, whatever characters you choose, but there's only 10 tries. There's not a thousand or a million. So you can effectively use shorter passwords, and your odds of being attacked are actually lower. Now, I do recommend longer passwords. But I personally, my personal recommendation, it's not a corporate one, but a personal recommendation is use a passphrase that's easy to remember, like "The quick brown fox

jumped over my IronKey" or something. Because, you know, you don't have to have weird numbers and letters and upper and lowercase because, again, you've only got 10 tries to guess. Now...

Leo: That's kind of a nice - that's kind of nice. You're right, that is totally a usability issue. And this issue is raised by Bill Gates, who said passwords don't work because people can't remember them. Either they use a password they can remember, in which case it's no good, or they can't remember them and they put them on a sticky on their screen, and then that's no good either. But this was his argument for smartcards.

Steve: And the point David is making is that the only way you can really do this and make it secure is if the counter is on the hardware because any time, I mean, you could certainly have a counter in software, but all you have to do is have some other software that resets it back, and again you're vulnerable to brute-force attacks.

Leo: Now, once something like that is on hardware, though, it's still software, it's just it's written to firmware. Can't it be modified with a jump or something like that?

DAVID: No, well, you'd have - so now, well, that brings up the question of can someone load malicious code.

Leo: Right.

Steve: Right.

DAVID: So the firmware that does the counter is not modifiable when it comes out of the factory. So it's actually not upgradeable. It's actually a metal layer in the factor. So that firmware itself is not - it's actually part of the silicon.

Leo: Right.

Steve: So you're never going to change your mind and wish that it was 20 tries instead of 10. That part is...

[Talking simultaneously]

Steve: Exactly, it's physically locked at 10.

DAVID: Well, actually the logic which implements the counter is locked. The counter can actually be changed in our enterprise version, which is just coming out in the next couple of weeks. And that can be changed once you've logged in. And you have to submit the existing password. And you can then change the password try count based on policy. So an enterprise could say, you know what, I want the try count to be three, not 10.

Steve: Right.

DAVID: And I want to enforce a nine-character password with three upper and two lower and a number.

Leo: That's excellent.

Steve: Nice.

DAVID: And so we've put that in because there's environments where you want a homogeneous password policy across all devices. The other thing, just one point I wanted to bring up about the firmware, is we do have the ability for parts of the engine, which implement other kinds of logic, to be upgraded. And that brings out the question of malicious firmware, can someone put malware onto the device itself. And we prevent that by digitally signing all firmware updates with a hardware signing module that's securely managed. And those firmware updates, when they're loaded onto the device, are then verified by hardware with a 2048-bit RSA signature. So if you try to load malicious firmware, it will not load onto that device.

Steve: Nice. And it's only you guys who have the matching public key at the other end that allows you to securely sign the firmware after verifying that it's what you want to load onto the keys.

DAVID: Right. The public key is actually burned into the silicon and can't be tampered with. And then the private keys are stored in an HSM that's accessed by two [indiscernible] with two different smartcards. So you can't get rogue firmware.

Steve: Very nice. We should talk a little bit about the additional services that you guys offer that go along with the key. I mean, we've in the past covered multifactor authentication a lot. Is there any provision for the key being sort of a general purpose hardware token for other authentication?

DAVID: Absolutely there is. And that really gets back to why we decided to do a USB device in the first place. So yes, there's this issue of USB storage, and there's these cool things that are coming along, like portable applications and people putting virtual machines on IronKeys. But the other thing that we really were cognizant of is that strong authentication, I believe, is an imperative. I frankly believe that passwords are not enough for most applications. And that will continue to be proven true over the next 10 years. And one way to do strong authentication is to have it on a USB token that can be inserted into any computer, that can be removed from that computer, that requires a strong password to get into. And that's what is also part of the IronKey.

So if you just want to use it as a flash drive, that's great. You can do that. It's the world's most secure flash drive, and hopefully people like it a lot. But there's more to it. These devices are full crypto engines. They have full capabilities to do RSA encryption. They can do SHA-256 hashing. They have a variety of different crypto algorithms on them implemented in hardware. They have strong encryption keys on the devices for private and public key operation. And that allows us to do a lot of really cool things that are actually really easy and seamless to the user, but add a ton of security for two-factor.

Steve: And is any of this API published?

DAVID: So. There is a PKCS#11 API that actually is available with every device. And that is how we make the onboard Firefox talk to the crypto hardware now. So you could

actually taken open applications like Thunderbird or anything else that uses PKCD#11, which is an open API, and you could actually use crypto ops for your own application.

Steve: And I did notice that you had the DLL there as part of the files as you were loading.

DAVID: Exactly. The other thing that we've done is we are now in a sort of limited beta of a more broad software developer kit. So that you can actually, as a developer, access more functions and do things like load your own software on the device so that you can create your own custom application container or what have you, using the IronKey devices.

Steve: Very nice.

DAVID: We will be announcing some interesting partnerships in the next couple of weeks around other forms of two-factor authentication that are on the device because it really is designed as a general purpose authentication device to support not just PKI-based authentication, and to support multiple different credentials on the device at the same time.

Steve: So there's some provision, then, for user storage, either in the IronKey or protected by the cipher chip.

DAVID: That's correct. There are areas where applications can have private encrypted storage areas with their own AES keys and their own access control, yeah.

Steve: Very nice. And we should talk about the other services, like the safe web surfing and privacy and TOR and so forth.

DAVID: Okay. One of the cool things about having strong authentication on the device is we can offer real neat web services, and we hope that third parties will over time develop them, as well. And so you know it's really that device. So one of the things that's very simple that we offer, it's completely optional for users, is self-service password recovery. So if you're using a strong password, and you're the forgetful type, you have the option when you fire up the device to register with our online service, and we will store your device password for you on our service. And if you forget it because let's say you, I don't know, go out and change your password after a couple of beers and forget the next day what you changed it to, or you don't use it very often or what have you, you're a busy doctor and you only use it once every week or two, you can actually come back to the online service, answer a couple of secret question challenges, we send you an email to also confirm your identity, and then we will actually present you with your password so you can unlock your device. Completely optional, but it leverages the strong authentication of the device so that we know it's not somebody else trying to spoof you and get your password.

Steve: I was just going to say, I wanted to make sure that our listeners picked up on the fact that it is only possible to do this if the IronKey in question is physically mounted on the computer at that time.

DAVID: Yes. You have to have the IronKey. We do the challenge response. You have to log into the device. You have to answer secret questions. You have to have possession of the email accounts, as well. And again, completely optional. And we're looking at stronger authentication, as well, like potentially we could ask things where you ping the mobile phone for an authentication code.

Leo: I like that. I always like that. So it really is interesting. So once you have this hardware encryption capability or this hardware random number generator capability, it really is a useful - it's more than just a USB key.

DAVID: Yeah. Really it has the potential to become a way to manage your identity, to manage your passwords, and also over time we believe to carry applications around. So speaking of passwords...

Steve: Go ahead.

DAVID: One of the applications that you have on the device, if you choose to use it, is a password manager which, you know, there's lots of password managers. You can download freeware, and hopefully you don't download one that's actually malware. But there are some good real ones out there, as well. Just check the reputation of what you're downloading. But, you know, it's a password manager for your Internet passwords. Couple of things that are different about it, though, are, one, the passwords are not stored in a data file on the USB key that could be copied. They are stored inside of the USB key in protected memory area, so they're not on the file system. And, two, we offer the ability to do an online encrypted backup of your password database in case you lose your IronKey. If you get another IronKey, and you reauthenticate and prove your identity, you can get back and you can recover your password database. So it starts to be the beginnings of ways to manage identities and passwords beyond just storing a file.

Leo: Could you use it as, like, the football dongle, Steve, that we've talked about, the PayPal dongle? I guess you could put software on there that would do the same thing; right?

Steve: It's a good question. We've talked about, David, the SecurID and VeriSign's six-digit LCD one-time password system where you prove that you're in physical possession of the device by pressing a button, and it gives you a sequential six characters that you then enter in, and the server at the other end knows the key in your device and so is able to confirm that you must be in possession of it.

DAVID: Yeah. I've got a keychain full of those.

Leo: Would it be possible to use an IronKey for that kind of thing at some point?

DAVID: So I hate to do this, but I have to use this moment as a teaser to invite people to come to our booth at the RSA show in San Francisco the second week of April. We are making no announcements here today. But yes, this is a general purpose authentication device.

Leo: That's what I like. Eliminate all those dongles, yeah.

Steve: It's also worth noting, too, that when you use the IronKey, for example, on a foreign machine, and this differentiates, it's one of the thing I noticed, it differentiates it

from, for example, TrueCrypt that we've talked about, you do not need admin privileges in order to run the client-side software in order to access the key, which, you know, can really be a good thing.

DAVID: I cannot tell you how difficult that was to do, by the way. That was so hard to make that work without installing drivers and software. Because we're not just, you know, copying files. We're doing cryptographic operations, control operations. There's a whole bunch of information going between the device and the control panel software. To make that work without installing drivers on XP, in non-admin mode, was unbelievable amounts of work. So difficult to do.

Steve: Well, I really think it's worthwhile because, again, on a system you don't control, you may not have admin privileges. But you do need access to your data.

Leo: How about Vista? Were you able to do it on Vista, too?

DAVID: It works fine on Vista. We are right now in beta testing of Windows 2000 SP3.

Leo: You'll like that, Steve. Steve's the last man still using Windows 2000.

DAVID: We launched a beta of Linux support this week.

Leo: Great.

DAVID: So we'll be doing...

Steve: And a Mac, too.

DAVID: We have basic Mac. It's not as much as we want. But once you initialize it on Windows or Linux, you can then unlock it, at least, as a flash drive on Mac. But we will be adding more fully featured Mac support. If anybody's a super hot Mac programmer, we're hiring.

Leo: Yeah, yeah. Well, that's - and so the capability to do that driver thing, is that a requirement for all the functionality of the IronKey? I think you said that you don't have to have that for everything.

DAVID: Which capability?

Leo: The driver support.

DAVID: Well, the driverless support is needed because you have to enter your password into the device, and we have to be able to securely communicate that to

the device. And that goes over, obviously, not a data connection over USB.

Leo: So that's the piece that has to be installed onto XP or...

DAVID: No, we don't have to install anything.

Leo: Oh, you don't.

DAVID: No, it's completely driverless. That's the magic.

Leo: I see what you're saying.

DAVID: Right. That was, I mean, that was probably six or seven months of work, seven days a week, 15 hours a day by super, super smart people. It's nonintegral. And that connection, by the way, is fully encrypted. The control connection between the software and the device, not for data, but for all the control connections, passwords, crypto, APIs and all that is a fully encrypted CLS-like stream.

Leo: So do you use the mass storage class driver?

DAVID: Yeah.

Leo: You don't sound happy about it. But that's how you'd see it as a hard drive. Is that sufficient for you to communicate all this other stuff, too? Or do you have to do something else?

DAVID: The theory is no. The reality is yes. Magic hand-waving happens here.

Leo: Interesting.

Steve: And then obviously you must also be protecting the machine's client software from any kind of tampering and messing around because it has to be trusted in order to establish the TLS dialogue with the IronKey itself.

DAVID: Right. So we do as much as we can. All that software is locked on a virtual CD-ROM that's digitally signed. But at the end of the day, to be quite honest, you cannot trust the host computer ever. You just have to assume the host computer is not trustworthy, and you have to do your best. But, you know, doing your best includes, yeah, making sure you can't tamper with the software, you're doing PKI operations between the software and the device to establish CLS-like connections, things of that nature.

Leo: Well, let's assume worst-case scenario somebody's got a host computer that's just, you know, completely owned. Obviously that means they can see any data before it's copied to the IronKey. But does it soften the IronKey afterwards? I mean, when you unplug it? Now, the IronKey is still secure, yes?

DAVID: That's correct. Yeah, and that's because we're doing things like, for example, software updates and firmware updates all have to be digitally signed and are verified in hardware. So being able to put low-level malware on the device is not really feasible. I think some of this thing, I mean, we also control the autorun, so malware should not be able to automatically run off it if it's copied onto the device. I think there's more that can be done, for example putting AV and antispayware scans when you unlock the device. So we're looking at a bunch of different options in that area.

Leo: Are you using U3? How are you - what are you doing the - how are you doing the autorun?

DAVID: We're not. U3 was really a sort of a proprietary thing that actually has been sold off to Microsoft. It ended up in my view being kind of a shareware distribution thing. I mean, if you look at what was the killer app for it, I never quite figured that out.

Leo: But it sounds like you're doing something similar because you're mounting a locked CD-ROM image and so forth.

Steve: Yeah. Essentially the IronKey has several profiles. One is that, when you initially insert it into the machine, it looks like a CD-ROM. David was saying that they have also taken great pains to protect that from being altered in any way. And then the CD-ROM starts up, and it invokes the client, which then mounts another device which is your encrypted drive.

DAVID: That's right. So it comes up with two devices, one a CD-ROM, runs the software. Then we have to deal with drive mapping issues because you might be in an enterprise environment where you've gone and mapped a F: drive to some network drive, which you're not supposed to do, but people do all the time. And so you have to then negotiate where you're going to mount that new secure volume. And that comes up as a removable media. And once you're authenticated to the device, we mount the media on there. So there's your, you know, 4GB of writeable storage. That also protects against certain attacks, for example cloning an offline attacks. So imagine you've got a software-encrypted flash drive, you could copy the contents off of it and then go and farm it out to a botnet of 10,000 computers and for free go and attack it. Or just run common attack tools on your laptop. And the IronKey you can't do that because we don't mount until you've actually successfully logged into the device.

Leo: Very interesting. I love the challenge that you faced and the work that you put into solving all these. It's really great.

Steve: And then also you guys have what you call a Secure Sessions service.

DAVID: That's right. We basically have, and I think you mentioned it, it's based on TOR technology. We're big fans of the TOR project. But basically one of the problems with that system, which is effectively a - it creates encrypted tunnels out to the Internet so that people can't easily spy on your traffic. But there are a certain number of problems with the experimental, what they call the "experimental network" out there, which is that it's slow. And you don't know who's actually running the endpoint, so oftentimes you get malicious people running endpoints, injecting malware, injecting code, injecting trackers, redirecting DNS or what have you.

So what we did is we built a private network based on that technology where we establish an encrypted surfing connection from your computer out to our servers. It's a three-layer encryption model, so we can't actually track your activities and record it because it goes through multiple hops at different datacenters. We also run the DNS for you. So you're protected against pharming attacks if let's say you're on a WiFi network and someone's running malicious DNS on there, trying to route you to their server instead of when you typed in PayPal. They could take you to somewhere else. So we protect against that by running the DNS queries through our network, as well. So you're getting privacy. You're getting accurate DNS information, as well.

Steve: It's tremendous.

Leo: Yeah. I had no idea. I thought it was just a USB key with, like, a cool, brushed-metal finish. I, you know...

DAVID: The other thing we built into that was an anonymous subscription protocol. So we can, you know, you're not just going to get everybody piling onto that network. But again, it's an anonymous cryptographic protocol that was validated by a number of different cryptographers. So, you know, we know you're allowed to be on it, but we don't know who you are.

Leo: Wow. So you're not doing your own TOR thing. You're still a gateway to TOR.

Steve: No, no, no, it's their own network.

Leo: It's your own TOR.

Steve: Yes.

Leo: Oh, wow.

DAVID: Yeah. And then...

Leo: And you control the exit nodes, so you're really...

DAVID: We control the - yeah. So we can control security on the exit nodes. We know where they are. And we control the DNS out of them and things like that. So you can provide a lot of different services on top of that.

Leo: Wow. Really neat. I'm going to have to buy one of these, Steve.

Steve: I have three of them now.

Leo: I know. Did you break one? Did you actually see if it self-destructed?

Steve: No, no, no. I don't want to lose it. It's just too cool. And we ought to also mention, and the IronKey site talks about this, is that it's deliberately potted solid with an epoxy compound. So it is extremely hard to get into. But it's also - it means that if you run over it or back over it with your car or something, it's also uncrushable. So it's really strong physically, as well.

DAVID: And a degree of waterproofness as well. So if you leave it in your pants, and it goes through the wash or what have you, if you dry out the connector, it will work again. All of the electronic components are sealed. And it does make it highly tamper resistant. Meaning, you know, you'd have to get into the device and grind through it and get at the chips. And your chances of getting at them without destroying them are much more difficult than with any other regular USB-type device.

Leo: And all this for \$79.

DAVID: Right.

Leo: Really you could charge twice as much. That's really amazing. What's the next...

DAVID: It's actually a good deal. I mean...

Leo: It is a good deal, yeah.

DAVID: They're not cheap to make, and there's a lot of technology in it. And there's a lot of thought into it, too.

Leo: What is the largest capacity?

DAVID: We're currently shipping 4GB devices. But we hope to have 8GB devices out by the end of the month.

Leo: And how much is a 4GB?

DAVID: \$149.

Leo: See, that's very reasonable, I think.

Steve: And Amazon has it for \$138 at the moment.

DAVID: Grrr. The other thing about these things is one of the reasons that they cost a little bit more than a regular device, other than all the magic we've been talking about tonight, is that these devices use a type of flash memory called SLC Flash. And most of the really cheap ones you can get use something called MLC. And MLC is really designed, like, for an iPod or something, where you copy your files on it, and you almost never, ever write them again. It's almost, you know, it's slow, and it's not very reliable. Whereas SLC memory is designed for applications like an IronKey, where you're putting critical files on it, you want them to last a long time, and you may be running applications where you're doing a lot of write cycles. So it's very much faster than MLC memory. But it also lasts up to 20 times longer. So you can get 100,000 write cycles out of them.

Steve: And I also heard you talking about wear leveling. So it looks like you did not short the actual technology over on the physical storage side, as well. I mean, you did everything you need to to make it not only secure, but really reliable.

DAVID: Right. So we put - there's hardware wear leveling in it, so that as you write, as data is written to it, it's randomly mapped around to reduce hotspots in the memory to make it last a lot longer. There's error correction in the NAND Flash, as well. So there's full error correction on the chip. So if there's any kind of glitches in the memory we actually detect it and will go move the block and rewrite it correctly. And then the other thing we do, which actually turns out to be difficult, is implementing AES encryption correctly for large blocks is actually very difficult. It's called CBC-mode encryption. And you actually have to store data inside of the NAND Flash away from where your real data is to make sure that the crypto stuff's not actually replayable if you write the same block. And that's actually really hard to do.

Steve: Yeah, cipher block chaining we've talked about on the show, so our listeners know what that is. But you're right, I hadn't thought about the difficulty of doing that in hardware.

DAVID: Right, where are you going to go store the initialization vectors? You've got extra metadata about every block of data. Every 512 block of data's got extra metadata that's got to be stored somewhere. And so thinking about where that's going to be mapped is actually a real challenge.

Steve: I've always wondered about wear leveling. What's the mapping page size that is

typically used?

DAVID: There's different page sizes. And actually I'm not a flash engineer, so that I'm not sure I can accurately answer that. But 1024-byte pages. There's small maps, there's large maps, there's sectors, there's...

Steve: Right.

DAVID: Now I'm talking beyond my area of expertise.

Leo: But it sounds good to me.

DAVID: [Indiscernible].

Steve: Well, I'm very impressed.

Leo: Yeah. It's really great to talk to you, Dave. Thank you so much for taking the time to join us. I'm not kidding, it sounds like something I want to run out and buy.

DAVID: IronKey.com.

Leo: There you go. We'll give you the plug.

DAVID: Thanks. Thanks, guys. I really appreciate the interview.

Leo: Sure.

Steve: Can you make them smaller? I wish it were smaller. That's the only complaint.

Leo: No, it's cool. Brushed metal. It looks like kind of a cool retro cigarette lighter almost.

DAVID: Yeah. And we put a lot of work into the physical case. So let's talk about smaller. Yes, they can be made smaller. That is, with everything there are compromises. So the first one is, it's the thickness that it is, 9mm, because you need to accommodate varying chip heights. It's a double-sided board. The way we get a lot of the speed is we're writing parallel channels at the same time. We actually have two memory chips in there, not one. Now, if we had one memory chip, it would run half as fast, but we could make it probably 12 or 14mm shorter. So we could actually make it quite a bit shorter. But you're going to sacrifice speed because you're taking away one channel of memory.

Leo: So you're interleaving, you're interleaving the writes?

DAVID: Yeah.

Leo: Wow, that's neat.

DAVID: Yeah, yeah. There's a whole high-speed DMA engine in there that's parallel.

Leo: Is that common in a USB key, or is that something you guys...

DAVID: In high-end ones.

Steve: It's the way they get the writing performance up is they're writing in parallel to many chips because each chip has a maximum bandwidth.

Leo: Yeah, even among USB 2.0 keys there's a dramatic difference in writing.

DAVID: Oh, yeah. Well, the difference is the single versus dual and the MLC versus SLC. Your main speed comes from MLC versus SLC. So you can get an MLC device with a single chip, it'll write 4 Mbps, maybe 3. And we can write up to 18 to 20 Mbps because it's much faster, more expensive memory, and because it's dual channel.

Leo: Is the read faster, as well?

DAVID: Yeah, it's like 30 Mbps or more on the bigger ones.

Leo: These are faster than the hard drives of a few years ago. That's what blows me away. I can't believe it. That's just great. Very cool.

Steve: And it's got a nice blinky light, too.

Leo: Ah, well...

DAVID: It's got a little blinky light. And it actually - the colors actually mean things. There's firmware verification at the beginning which is when it changes different colors, lets you know if the correct firmware is in there or not, all of these nondocumented features. And actually when they're dead there's a little blinky-light session, too, that lets you know they've actually cleanly erased. And for certain applications where you need to know it's safe to leave it dead, that's helpful.

Steve: Oh, that's interesting. I may have to kill mine, or one of mine, after all, just to watch that other light blink.

DAVID: Different firmware versions may or may not have it, so I can't guarantee how - I don't know how old yours are. But that's one of the newer features, the ability on the light to signal that the device is safe to be left.

Steve: That's very nice.

DAVID: That it's completely killed itself, yeah.

Leo: And now, finally, the unannounced feature that will be unannounced at RSA, will that be an upgrade, or will it require a new IronKey?

DAVID: So there's been some discussion about that.

Leo: If you were to add some features...

DAVID: [Indiscernible]. So we design these devices to be completely field upgradeable to where you can actually download software and firmware updates from us. So we don't want to require new hardware for that, and we want to make these types of upgrades downloadable. I will tell you guys that the challenge has been the constant battle between people demanding new features and people wanting these upgrades, and also QA of the upgrades.

Leo: Well, you also want to lock down the firmware; right? You don't want to make it too easily upgraded.

DAVID: Oh, no, yeah, trust me, the firmware is pretty locked down. But the question, the challenge is that we need to do it, you know, if we make that upgrade happen, and you have a device with a certain version of firmware, but not a different other one, and then you did this thing to it and stored this data here or there, it's going to be so different across the user base that the testing matrix that we have to do is extremely difficult. And so we're actually kind of working through that right now over the next month or so, you know, getting the process down where we can actually release updates a little more frequently, maybe smaller updates more frequently. So that's our hope this year is really to get into more of a regular pattern. And some people would say, well, gee, I can get an update to Firefox every three months or what have you. And yes, it's true, but that's one piece of software.

Steve: And also, for example, in the case of Microsoft, people are getting updates every month whether they want them or not because there are serious security problems that Microsoft is continually fixing. And it would obviously defeat the whole point if the IronKey were not really, really, really well designed from a security standpoint.

DAVID: Yeah. I mean, we're in a lot better position than Microsoft given that we control the hardware. And we came in it with digitally signed hardware and firmware and software. So you're right, we're in a much cleaner position. But then, you know, to their

credit they've had 25 years to develop their software update process. And they can now do it about once every month. Ours is, you know, we don't quite have 25 years of history under our belts. And I think the other one is we don't want to do so many updates.

Steve: Right.

DAVID: And we're updating firmware as well as software. So it's actually far more challenging than might actually appear on the surface.

Leo: Yeah. Good, well, we'll look forward to see what we can do new with our IronKeys.

Steve: In about a month.

DAVID: Yup.

Leo: Dave, it's really been great talking to you. And now I'm sold. I tell you, you make an excellent representative.

DAVID: Thanks a lot.

Leo: Which is good because it's your company. So that's good.

DAVID: Yeah.

Leo: Yeah. Dave is the CEO at IronKey, also has a blog you can read on the IronKey site if you want to know more about the IronKey and its uses and so forth. That's IronKey.com. And your blog is blog.IronKey.com. Dave Jevans, thanks so much for joining us. We really appreciate it.

Steve: Thanks, David.

DAVID: Thanks, I really enjoyed it.

Leo: Wow. Okay. You're a cagey fellow, Steve Gibson. Because you said, oh, we're going to talk about IronKey. I didn't realize it was - did all that stuff. That's really neat.

Steve: Well, what I learned during this also, which for me obviously as a developer and with my Perfect Paper Passwords and all that, is that it's a multifactor authentication dongle that is having an open API on its way. So that becomes very interesting in addition to everything else it can do.

Leo: Security is very interesting. I love it. Well, what's interesting is, unlike a lot of programming, although I guess all programming now has to be security aware, you have an adversary. So it's a more kind of challenging form of programming where it's, I mean, it's really an intellectual challenge to think this through.

Steve: Oh, Leo, yeah, it's like everything - for example, when I was writing our eCommerce system, there's, I mean, everything I did, everything I thought about was Spy vs. Spy. It was like everything I was doing was, okay, now - or even when I did the Perfect Paper Passwords system, in order to create secure roaming for myself and Greg and Sue, was okay, now, wait a minute, what if, what if, what if. And so you're constantly challenging yourself with what could someone do to get around this.

Leo: Man. I guess all programming nowadays, if it's on the web, is going to have to be aware of that. Anything...

Steve: Oh, and Leo, that is the huge vulnerability. Every single newsletter that I get from SANS, for example, they talk about the phenomenal number of insecure web apps that have known vulnerabilities in them because people are just rushing these things onto the 'Net. And security is the last thing they think about rather than the first thing they think about. Which it really needs to be. And certainly it's the only thing these guys thought about.

Leo: Clearly. Well, this was a fun episode. I'm just...

Steve: I want to just inject one thing. I just got a big kick out the fact, listening to Dave, who's steeped in security and security technology, all the things he talked about we've covered on Security Now!.

Leo: Yeah, I know. You know, it was neat. It was like we've been building up to it. You could listen to this, if you'd listened to all the previous ones, and understand what he's talking about.

Steve: Yeah, I mean, the acronyms, TOR and cipher block chaining and public key and private key and symmetric crypto and, I mean, just it was everything we've done here. It was just really neat.

Leo: And it's what impresses me about it is it answers all those questions that you would have. And it has capabilities that just blow me away. But I think running their own TOR servers, wow. I mean, these guys obviously are into this stuff, and they share your sensibilities about the whole thing.

Steve: Yup.

Leo: All right, Steve. We're going to wrap up. Next week it's a Q&A segment.

Steve: Yes, yes, yes, yes, yes.

Leo: People should go to...

Steve: And then the week after we've got our RAM hacking episode.

Leo: Oh, we're going to talk about the Freon freezing thing?

Steve: The freezing RAM and Firewire backdoors and USB boot dongles for sucking RAM contents out and all kinds of cool stuff.

Leo: Oh, great. So we know we've been getting a lot of email about that. We'll talk about that. Next week we'll answer many of your questions. And of course you can go to Security Now!'s website - well, Steve's website - GRC.com to submit a question, to ask a question. You can also go there to get 16KB versions of this show, the really small ones for people who have dialup. You can also get transcripts. A lot of people like to read along, highlight as they go. All of that at GRC.com/securitynow. And that's where you can find all of Steve's great freebies, including the new updated Wizmo that automatically turns off that wireless 0 config. That's a very useful...

Steve: The new wanlock command.

Leo: Wanlock. Now with wanlock. I love it. It's Wizmo, Now With Wanlock. You can also, of course, get SpinRite there, which is Steve's bread and butter and a great program, a must-have program that is the ultimate file and disk recovery utility. It's a maintenance utility for your hard drive. If you've got a spinning disk, you need SpinRite. GRC.com for that, too. Steve, we'll see you next week.

Steve: Right-o, Leo.

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

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