RAM Hijacks

Description: Steve and Leo plow into the detailed operation of static and dynamic RAM memory to give some perspective to the recent Princeton research that demonstrated that dynamic RAM (DRAM) does not instantly "forget" everything when power is removed. They examine the specific consequences of various forms of physical access to system memory.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-137.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-137-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 137 for March 27, 2008: RAM Hijacks. 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! with Steve Gibson, everybody's favorite security guru at GRC.com. You're a guru.

Steve Gibson: I guess I am. I'm a year older guru now.

Leo: That's fantastic. Congratulations and happy birthday.

Steve: Yeah. Well, you know, I am not a big birthday person, but it's sort of nice to note that I've survived another one and got - I think I'm about halfway done, so...

Leo: That would be good. That would be good.

Steve: Yeah.

Leo: You're halfway there?

Steve: My grandfather lived to 103. I'd like to beat him.

Leo: You're kidding.

Steve: Like to beat him by one year, so...

Leo: 103.

Steve: Yeah.

Leo: So you have good genes. And you're wiry and thin, which bodes well. You don't drink, you don't - well, you drink a little wine, but that's good for you.

Steve: Cabernet is good for you. And I work out 64 minutes a day on the stair climber. I had my annual physical the other day, and the doctor said, well, if all my patients took as good care of themselves as you do of yourself, Steve, he said, I wouldn't have a job.

Leo: That's fantastic.

Steve: And I told him we'll be doing this in 20 years.

Leo: That's so great. Well, congratulations. Happy birthday.

Steve: Thanks.

Leo: Let's see. Today we're going to talk about something that is a very, very hot topic, something that actually we missed a little bit because we did that TrueCrypt drive encryption story a little early, we taped it early, and then the news broke that you could in fact crack drive encryption with some very arcane techniques.

Steve: Well, arcane, and I'm not sure whether this is a hot story or a cold story because a lot of it involves spraying Freon on RAM chips in order to extend the length that they - or, well, to slow down their rate of decay. We're going to talk about all of that and also about the various means of accessing RAM and what it means. These guys at Princeton just did a fantastic job, so it's going to be really fun to catch everybody up on on what's been going on with that.

Leo: All right. Let's get some errata and addenda and all...

Steve: Well, actually, yeah. We don't really have any errata, but we have a very important security issue which has just come up. It is an exploit which is being actively exploited that's been acknowledged by Microsoft. It involves a flaw in their Jet Database which is part of the Office Suite and especially comes along with Word. What's happened is, okay, first of all, the vulnerability affects Windows 2000, XP, and Server 2003 SP1. This vulnerability does not affect any computers running the SP2 of Windows Server 2003 or Vista because they run a different edition of the Jet Database.

It turns out that the MDB extension is the file on the database file. There's an exploit which was found in the wild, it was zero day, no one knew about it, at least it has never been acknowledged publicly in any forum. And most systems will not run anything with a .mdb. So the way this exploit functions is that users receive an email message with two attachments. One is a Word doc, and then the other is a file with the extension changed. And when you open the Word doc, the Word doc causes the database file to be executed, which makes the exploit occur. And so what's happening is this is a targeted attack. For example, it's been used in industrial espionage and attacks on government systems where, rather than just spraying spam, people who are using this are deliberately sending email to specific recipients, hoping that they will open this attached Word doc and get themselves infected. So there's no fix for this. It is being exploited. There's really nothing anyone can do at the moment, unfortunately.

Leo: Well, except not open email attachments.

Steve: Exactly, except follow the standard guidelines and just do not open email attachments. There is a scenario also where both files arrive in a ZIP file instead of being separate attachments. So but either way, basically the exploit functions the same way. The Word doc causes the database file to be executed, which wouldn't otherwise be executable. That makes the exploit happen.

So I expect - Microsoft's advisory acknowledges this. They know it's in the wild. They're trying to figure out what all the possible exploit vectors are to make sure they nail it down. And they're saying they will either deal with it in the second Tuesday of April - hopefully it would be April - or they may do an - this may be bad enough and serious enough that they will do an out-of-cycle patch. We're going to be unable to advise people next week and the week after because we're having to record those episodes early. So there may be news of this in a week or two. If we're not able to talk about it, that's why, because we're already had to record our podcasts. So I just wanted to give people a heads-up about this.

And also, I mean, I've been noticing Windows Update has been very busy lately. There was an update to an Excel patch which was patched on March 11. But there was what they call a "regression error." They broke something else in Excel when they made the patch. So they had to patch the patch. And that came out a few days later. There is now vulnerability code, exploit code for some of these problems that were patched. Remember we had the big Office Suite patch earlier this month. So there is now released and in the wild vulnerability code which generally means we're going to see a lot more attempts of that. So again, the standard guideline is make sure that Windows is staying patched, at least for the patches that we have right now.

Leo: I was doing a search for Jet vulnerabilities and there's a lot of them since going back years. This is kind of a continual problem. There was one last year. There was one in 2004. I'm having a hard time finding the one we're talking about. Oh, here it is. Security Advisory 950627. I'll put a link...

Steve: That's exactly the one, yes.

Leo: I'll put a link in the show notes. You know, people often yell at me because I say don't open attachments. And they say, well, come on, not all attachments. And then I say, well, I guess technically it's don't open executable attachments. The problem is, people can't tell what's executable. And this is a really good example. Here you are getting a doc file and an mdb database, neither of which are executable technically.

Steve: Well, exactly. And in fact it used to be that the file extensions would be changed in order to fool the filtering software. And Microsoft added technology to open files by content rather than by extension in order to solve that problem. Here we have a problem, though, that Word is able to run scripts, and documents are able to contain scripted executable code, which is just like a web page.

Leo: So this is, I mean, you know, don't open attachments. Really it's very straightforward. I wish there were another way. People say, well, how am I supposed to do business if I can't send attachments?

Steve: Well, and to give us a little more strength, when I was researching the details of this I ran across an interesting sort of summary from the Security Focus site. And quoting from their site, it says - they wrote, "Flaws in Microsoft's Office productivity applications have become standard weapons for fraudsters conducting targeted attacks aimed at high-level managers and executives. While 10 or fewer high-severity flaws were reported in the five major component applications of Microsoft Office each year from 2002 through 2006, at least 26 high-severity flaws were reported in Office applications last year, according to data from the National Vulnerability Database. And earlier this month, as we know, Microsoft patched dozens of flaws in Office applications."

Leo: Yeah.

Steve: So we're seeing an increase in the rate at which these problems are surfacing in Office.

Leo: That's interesting because for a while it looked like the vector had shifted from email attachments to web-based vulnerabilities, web-based exploits. Guess that's not the case.

Steve: Well, the Office exploits are slightly more targeted, as this said. They're not spraying them out to everybody because the likelihood of finding victims is smaller, and

they would rather not have their actions discovered as quickly. They'd like to keep these exploits a secret. I mean, this Jet Database exploit that we're talking about here, the longer it stays unpatched, the better for the bad guys. So they're not wanting to spray it all over the place.

And the one last little bit of news I wanted to mention on the security front was you may have heard, Leo, you and I hadn't talked about it, about there was a bunch of furor, I guess it was late last week, about - first it was Barack Obama's passport file that had been breached on three occasions in January, February, and earlier this month, in March. And then they found out that both Hillary Clinton's and John McCain's passport files, all three of them had been opened by contractors working for the State Department. The cool thing is that it was State Department monitoring software, security monitoring software that caught these breaches. So...

Leo: Too bad they caught them and couldn't block them.

Steve: Well, actually they caught them and notified people, who then didn't talk about the fact that they had been caught.

Leo: But it might have been better, instead of having the monitoring software, have some sort of security on there.

Steve: Yeah. Well, now the problem, of course, is that this was an unauthorized access.

Leo: Well, they shouldn't allow unauthorized access.

Steve: Well, but no, I mean, it was something that these employees by virtue of their job had to have access to. They had to be able to do it. But they shouldn't have looked there. In fact I'm reading stories in the security space like this all the time. Police officers or law enforcement people are poking around in databases, in files, in other people's lives that they're curious about. And they have the authorization to access the database. But it is a violation of privacy rights for them to be using their curiosity to direct their searches. And so that's the problem is these employees, by virtue of their job, had the access. But they abused their access in order to satisfy, apparently satisfy their curiosity.

Leo: Well, at least they had a monitoring system. They didn't have one before, so that's...

Steve: Well, and that's my point is that - and it's a rare thing still in this day and age for companies to have monitoring software. Normally companies just rely on policy enforcement and don't have that backed up by something verifying that the policy is being followed. So, yes, this was a good thing.

And then I have one quick little short fun SpinRite story that was different. I always look for, try to find things that are different. The subject was "SpinRite Rules." And this was - looks like Ravi Keecheril. I hope I'm pronouncing his name right. He says, "Hello, Steve. SpinRite rules. I have a MIT TV box made up entirely" - I love this - "of discarded hard

drives."

Leo: Oh, wow.

Steve: "In my company, whenever a hard drive fails, it immediately goes to the dark storage room on its way to the hard drive graveyard. I've mentioned SpinRite many times to them, but they're more comfortable taking a new drive from inventory. There's always several new drives in stock, usually bigger than the last one that just failed, of course. So I asked them, can I have the old ones? And my boss said sure. So periodically I go into the storage room, take the old drives home. Of the seven drives I've taken so far, six have been completely resurrected by SpinRite and are working happily ever after."

Leo: See, that's an interesting point, that a lot of times what looks like a failed drive isn't a failed drive, it's just an error on the medium that can be either repaired or blocked, and the drive is fine.

Steve: Sure.

Leo: That's a large percentage of them, I mean, that's a huge number.

Steve: Exactly.

Leo: Do you think that holds true across the board?

Steve: Well, given SpinRite's proven track record, its ability to recover drives which have died in one way or another, I mean, it really does seem to be the case.

Leo: Very interesting. All right. Shall we talk about these exploits, these very interesting - Ed Felten is a brilliant security researcher at Princeton. He's always pushing the envelope.

Steve: Yep, in fact his name is very familiar to me because I see him being cited and quoted all over the place.

Leo: Oh, yeah. I think he was first known in the copy protection wars. He's done some really interesting research there.

Steve: Over in DRM stuff.

Leo: In DRM, yeah. He got in trouble for, and I think bowed out of - oh, he's done

analysis at Diebold, the voting machines? He's really an interesting guy, interesting researcher.

Steve: Yes, and in fact I just saw an article about that. Somebody wanted him to analyze - he and Princeton were going to analyze another voting machine issue where there had been some concern raised. They were all set up to do it, and they got threatened with a lawsuit, saying that it was a viol- by the company who had the apparently defective voting machines, preventing them from analyzing what was wrong. It's an example where our DMCA really does us no real service.

Leo: Yeah. He did a really good paper on the digital music initiative challenge. He bowed out of it because he couldn't publish his results. But he cracked it in a few weeks. I mean, it was like - these guys are bright. So what's the latest?

Steve: Well, what they did was - and many of our listeners literally flooded us with reports of this when the news first broke. The great concern was that - there was a little bit of hype, what I consider hype. And I think maybe our listeners, once they have all the facts in front of them, will agree. Because the email that I was getting made it clear that people who were writing to us at GRC.com/feedback, using the web form that I have there, they were clearly given to believe that whole drive encryption had been cracked, it had been broken, it was a serious problem. That's in fact not the case, although what this group did was extremely cool. Essentially what they discovered was that the contents of RAM stays available for longer than was believed.

Now, historically people have understood that RAM could have sort of a latent image, essentially, that there was - that data stored in RAM, until it was expressly and explicitly cleared, would linger for some length of time. But no one in the literature that had been surveyed ever really sat down and figured out, okay, how long is long? Is this seconds? Is this minutes? Is this hours? No one thought it was days. But people sort of - there was sort of this general concern floating around that oh, you know, memory doesn't - it isn't, like, immediately lost, even when power is removed.

Now, let's back up a little bit and talk about technology because that's always the underpinning behind what we talk about that I really enjoy and I know our listeners do. There's two types of memory, essentially. That is to say, volatile memory. We've talked a lot about Flash RAM and how that works. Volatile memory is either static or dynamic, which are the two terms to broadly differentiate memory. The original memory that was created for early computers was static memory. And what that means is not that it survives power being turned off, but that it does not need to be continually refreshed. Refresh of memory - and people may have heard, like, RAM refresh terms if they've been in the business for a while - is something that dynamic memory needs, and I'll explain why in a second. But static memory doesn't need it.

Now, the way static memory works is kind of cool. If you think of a piece of digital logic called an inverter, we've talked a lot about binary data. An inverter turns a zero that it receives on its input into a one. And conversely, it turns a one that it receives on its input into a zero. In other words, just whatever you feed it, it sends out the other bit, the reverse. Give it a zero, you get a one; give it a one, you get a zero. So imagine a very simple circuit where you connect the output of the first inverter to the input of the second, and the output of the second back to the input of the first. So you've got two inverters sort of in series. They're each connected to the other in a loop. Well, that's a

stable, logically, that's a stable configuration. If say that the first inverter has a zero coming into its input, so it puts out a one, which goes into the second inverter, which because it's getting a one puts out a zero, which is connected back to the first one, giving it that zero that we started with. So that thing can sit there forever, essentially. As long as power is up, those inverters are going to just maintain their state.

Now, imagine that we briefly imposed an external influence on this. We forcibly yanked the input of the first inverter, which is a zero and has been sitting there, we yank it briefly up to a one. We just force it up to a one. Well, it puts out a zero then, which goes into the second inverter, which now puts out a one. So that one that we briefly yanked up is now again stable. Well, this is another thing, a common term people have heard of called a "flip-flop." Essentially it's called a "bistable multivibrator," also known as a "flip-flop," and it's the basis of static RAM. So to turn that into a chunk of memory, basically you just have a ton of these little inverters, these little inverter pairs, all connected to each other, connected back to back like that. And you provide an ability for reading out the state of any of these little flip-flops and also for forcing them into, like, change from their otherwise stable condition over to the condition that you want to store. So that's how static memory works.

The problem with it, which our semiconductor industry ran into after a while, was that every single cell, that is, every single bit, essentially these two inverters, this flip-flop or this bistable multivibrator, it takes a lot of space in terms of silicon to create the inverters with the transistors and resistors and the addressing logic and things that you need in order to force it into either state, in order to read its status out, and just for its own little bit cell takes up a bunch of real estate. What that means is that as you try to increase the density, and of course that's what we're always doing in the computer industry is trying to store more data in a smaller space, you start having your chips getting too big, or you're just not able to put as many bits on a practical size chunk of silicon as you would like to.

So scientists, these brilliant engineers that come up with all this stuff, thought okay, how can we make this simpler? How can we somehow reduce the size that's required to store a bit? Well, they came up with something very clever, which is dynamic RAM. What dynamic RAM is is essentially a capacitor, that is, that's all there is, with a little bit of logic around it. But just a capacitor. A capacitor is a component in electronics which is able to essentially electrostatically store a charge. You put a charge on the capacitor by pulling electrons off of one of the plates or pushing them on. And as long as you leave it alone, it doesn't change. So it's a very simple way of maintaining memory. And there's a bit.

Now, the problem is, as you make capacitors very, very small - and again, density is our goal, we want to cram as many of these little microminiature capacitors onto a chunk of silicon as possible. As we make them very, very small, their capacitance, that is, their capacity for storing electrons, diminishes. And leakage effects begin to creep in, just sort of thermal effects, you know, electrons tend to wander off the reservation. And so the capacitor won't be able to keep its charge indefinitely. And again, as is always the case, we're trying to make these capacitors as absolutely small as possible.

So we start running into tradeoffs. What the engineers figured out was that they could make the capacitors incredibly small to get a whole bunch of them on a chunk of silicon, but they couldn't do that and have them keep their charge, for example, indefinitely. So they came up with this notion of refreshing. And the idea is that all of the capacitors, all of the memory bits in a chunk of dynamic RAM are continuously being scanned. That is, what's happening is you write something into memory which either charges or discharges the capacitor. If it's charged, it immediately begins discharging back down to its so-called

"ground state." It starts to just self-discharge due to electron migration. So as long as you come back and read that before it discharges too far, you can see whether you had originally stored a one or a zero there. And as it's draining, as long as you come back and read it in time, you can go, oh, well, this is only 50 percent full. But that means it must have once been 100 percent. So you refresh the data in the memory, essentially recharging all the little capacitors that have been trying to discharge since you last swung by. That's how dynamic memory works.

So now, if you imagine suddenly cutting the power to this, you have stopped refreshing, but you've got this whole grid of little capacitors which are, at their own speed, and based on variations in the specific physics of the material and temperature, they are all beginning to discharge as soon as you stop refreshing. So the guys who did this research, they said, okay, what happens if at normal operating temperature we cut the power, count to three, and then turn the power back on again? You know, what percentage of these capacitors will decay, and what can we do about that?

So the research they did showed that very much as a function of temperature that dynamic memory would hold its data for - oh, well, okay, first let's talk about normal operating temperature. Normal operating temperature, which is pretty hot actually inside a laptop or inside a computer, we're blowing air on all this stuff to try to keep it from melting down, but still it's very hot. What they discovered is there's a great variation in decay rate based on the technology being used. The newer RAM, being even more dense, meaning the capacitors are even smaller, tend to decay to, like, maybe 10 percent will decay in as short as one or two seconds. So you have, like, 10 percent loss of information in one or two seconds without refreshing, that is to say, without power on the DRAM. And again, at normal operating temperatures, they found a couple older dynamic RAM chips that, oh, maybe you could get as much as 10 seconds before you lost - oh, I'm sorry, the charts that I was looking at, we're talking about 50 percent loss of information in one or two seconds or 10 seconds. So in 10 seconds, even, I mean, the best these guys found was at normal operating temperature half of the capacitors had discharged to their ground state within a maximum of 10 seconds.

Now, we talked about how the reason these capacitors are discharging is electron migration through the dielectric, the insulation of which is what makes the capacitors possible. Well, naturally, as we know from physics and chemistry, temperature has a substantial effect on the rate of all these kinds of processes. So what the researchers did was they said, okay, let's - what could we do to extend this time for some sort of - whatever purposes. We want to see how much time can we get. So they just took those little spray can of air bottles, and it turns out when you turn them upside down and spray them the Freon comes out, and it cools these things down way far. They were cooling them down to, I think, -50 degrees C. And there, not surprisingly, by freezing the DRAM, they were essentially able to dramatically slow all of the physical processes going on in the DRAM which would otherwise be facilitating the capacitance discharge. And they were able to come back an hour later, that is, have this little DRAM chip out of a computer, sitting on a desk, spraying it with Freon to keep it cold, and then an hour later plug it into a computer and read out the majority of its data by freezing it down. And in fact they also dunked it in liquid something, hydrogen, nitrogen...

Leo: Probably nitrogen.

Steve: I'm not sure what it was.

Leo: They had to do that pretty quickly because you only have a second or two if you want to save everything; right?

Steve: Oh, yeah. Well, and the other thing is, Leo, I mean, they sprayed this while the machine was on. I mean, so...

Leo: Right, well, you want to cool it down before you remove power; right?

Steve: Exactly. So...

Leo: Then maybe you have some time, you could dunk it and get even more time out of it.

Steve: Okay. So, okay. So now we have a good foundation for understanding what they did. They were using Freon to slow down the loss of data from dynamic RAM. We understand how dynamic RAM works and why you get this, you know, bit errors. What they did that I think was the coolest was they said, okay, we're going to, after some number of seconds, we're going to as quickly as we can take a snapshot of what we've been able to maintain in the presence of known bit errors of RAM. Now the question is, how can we use that data? What can we find in there?

And one of the things they did, and their paper that you and I both have links to, you on your show notes for this episode and me over on mine, what they did was they said, okay, let's go after encryption keys. Let's look at BitLocker and TrueCrypt and major whole drive encryption, which is because it's exciting and it's fun. You know, what can we do? Well, okay, take for example a strong 256-bit key. And let's talk about AES because we've covered Rijndael, the AES standard, at length recently in talking about exactly how that works. So we take a key of a certain length. Well, we know that as you start changing bits in that, I mean, you change one bit, and you've got something that doesn't work at all. So given some percentage of bit drift caused by the dynamic RAM being disconnected from its refreshing for some length of time, you would think, okay, you're screwed immediately. I mean, this key changes at all, and it's useless.

But remember that in the details of the way AES works there's something called "key expansion," which we talked about. The idea is that, for example, in Rijndael and in virtually all other symmetric ciphers there are some number of rounds, that is that essentially a round is a reversible bit scrambling, meaning that it maps any set of bits that you're inputting that you're going to encipher or encrypt. It maps them into exactly one other pattern in a way that is reversible. That's the whole point, of course, is being able to decrypt what you encrypt. But that mapping itself, doing it once is not secure. So the ciphers work by iterating through this, doing that some number of times. Well, every one of those rounds requires some data from the key. And, for example, in the case of AES, we take the key, and there is a chunk of entropy, a big table of data, which is part of the AES spec, which has been chosen. And all AES implementations use the same big chunk of entropy. The key is mixed in a cryptographically secure fashion. And data is taken from that table. That generates the data which is, for example, XORed with the output from each of the rounds every time through.

The point is that the key, the original Rijndael key is expanded through this key schedule

or key setup, as it's called, to create this chunk of data which is then used, each piece of it, for each round of the cipher. What the guys realized was that data is like error correction code. It's like ECC that we've talked about on a hard disk, meaning that the key is expanded to something much bigger and inherently has much more redundancy in it. The individual bits in the key have no redundancy. But when you use them to expand this into the key schedule, you've got a tremendous amount of redundancy. It could be used like error correcting code. And they worked out all the details to reverse engineer the exact key from the key expansion and the key, under the assumption that there were unknown random bit errors. And they did the math, and it works.

So it's extremely cool. Essentially they said, okay, we're going to experiment with decay rates and dynamic RAM. We're going to figure out what kinds of levels of bit errors we can expect. We're going to experiment using temperature, using cold, to slow down the decay rate. And then we're, in the presence of known errors, we're going to see whether we're able to reconstruct a key knowing that Rijndael was used here or whatever it is. Well, in fact BitLocker uses Rijndael also, but in a different way. And they also do the same thing with DES. It turns out that DES's key schedule is extremely straightforward. So it was very simple for them, even in the presence of a high degree of bit loss, of bit decay, to reconstruct an original DES key, and even Triple DES.

So essentially that's what these guys did. It was promoted, of course, as look, we're able to take a chunk of DRAM which was briefly without power or without refresh or, for example, went through a cold boot, which is what they often did was they simply hit the reset button, so the RAM was not being refreshed for some length of time. And then the system came back up. And then so there was some level of loss. And so they were able to say, we're able to come back after a brief period of time, again, like maximum of a few seconds at normal operating temperature, or if the situation permits it, if we're able to cool the DRAM down, I mean, and really cool it down, we can go as much as hours or days and then bring the RAM back to life, capture its data, and even though we know we've got errors, we are able to, by taking advantage of what we know about the way symmetric ciphers work, and they did also some work with public key crypto as well because again the idea is that while you are using the cipher, that key is expanded.

So that key expansion, all that extra redundancy, is there in memory because it needs to be used dynamically. You just, in terms of performance, you cannot afford to expand the key on the fly every time you want to, for example, read or write a sector to and from the disk. That overhead would just be really prohibitive. So it's done once. And the point is if you capture the system, if you're able to get a snapshot of memory, even in the presence of errors, they've demonstrated that it's possible to reconstruct the keys of virtually all of the whole drive encryption products that they attempted. So it was very cool.

Leo: Well, it's very impressive, too, and they really do good work. So but I also gather from what you say that this is not something that's going to be easy for a hacker to do to you.

Steve: Well, okay. Now, there's some other things we have to cover when we're talking about RAM hijacks. But I wanted to first discuss...

Leo: How it's done, first, yeah.

Steve: Well, exactly. Well, this particular type of RAM hijack, which is, you know, the very intrusive, get your hands on the DRAM, I mean, you would hardly allow anyone to grab your laptop and turn it over and spray Freon on the memory, if they were able to even find the memory in your laptop. I mean, it would be obvious to you that this was going on.

Leo: Yeah, no kidding.

Steve: So I'm not sure...

Leo: And then let's just also, I want to emphasize this as well, they have to get your laptop while you're logged on. Once you're logged off the key is gone; right?

Steve: Yes. In the case of - we know that it's the case with the Mac because they did some work with a Mac. When you log off, your keys are scrubbed. I'm happy that this came to the attention of TrueCrypt...

Leo: TrueCrypt will fix this; right.

Steve: I was just going to say, yes. I mean, I'm really happy that this came to the attention of the industry at a period of time when TrueCrypt is under active development and has moved into v5.0. Because, I mean, Leo, you could imagine, if I got a lot of email, can you imagine the amount of email that the poor TrueCrypt guys got when this thing surfaced? So anything that the TrueCrypt guys can do to minimize the danger - and essentially what it means is it means when you're not actively needing to have access to the encrypted resource, whatever it is, TrueCrypt or BitLocker or the Apple drive encryption technology, it wants to actively scrub these keys from memory. You can either write nonsense over it or zeroes, I mean, there isn't the issue that we've discussed several times with hard drives where you're actually able to find what was stored there before. As you can imagine with these little capacitors, I mean, they're doing all they can just to hold onto the charge they've got. There isn't any notion of what was there before, although there has been some study that showed that memory also has a bit of burn-in features in the same way, remember, that you know that screensavers were originally created to, quote, "save the screen" because if an image was sitting on a screensaver for a long period of time, the phosphors were aging, well, because there's a physical process there.

Well, as we've just been saying, RAM, dynamic RAM has a physical process going on. And there has been some studies that showed that, if the same data was always being stored in the same place, that it might actually be possible to come along and take advantage of long-term physical changes in the memory. So, which is sort of interesting. It's like, well, that's really interesting. I mean, it would take a lot of research, and it would mean that memory would literally have to be burned in the same location. In modern operating systems that's probably unlikely because there's a whole layer of paging which occurs which associates physical memory with its logical addressing. And it probably means that just normal data is going to be moving around in physical memory and not in the same place all the time. But it's something for really security-conscious designers and developers to keep in mind.

Leo: Yeah, yeah, yeah.

Steve: Okay. So the other interesting things which are sort of fallouts from this, there are a couple. For example, you could take a USB dongle which is bootable and create a very small boot OS which takes a snapshot of memory. And you want it to be a small footprint in the OS because of course the OS is going to have to run in the same memory that you're trying to take a snapshot of. So you don't want to do too much. But, for example, these researchers did experiments using PXE, which is Intel's specification for network booting, saying okay, let's reboot a system and use the network boot ROM that is on the motherboard's BIOS to essentially install a very small footprint OS, just enough to do essentially a remote RAM suck through the network interface. So that was one thing that they did. You could also do the same thing with a small USB dongle that is able to boot and use it to snapshot the system memory.

And then the one other really interesting aspect of RAM hijacking which has actually been floating around for years is Firewire. It turns out that Firewire, as part of the spec, it's OHCI, the Open Host Controller Interface. And it turns out that the open host may be a little more open than these people intended. It turns out that the Firewire spec supports direct memory access. It really is a bus. It is a bus just like the bus that you plug cards into, through the 1394, the Firewire controller. And so it is possible for someone to create a Firewire gizmo which would, when plugged into a laptop's Firewire port, it would declare itself as needing accessing to direct memory, that is, Direct Memory Access, DMA. And it can then suck out the system's RAM through the Firewire port.

Leo: Wow. So if you had such a device, like a USB key, it would actually do the same thing reading the RAM as you have to do with all this freezing activity.

Steve: Exactly. Well, and in fact you may have some loss depending upon what the USB key does. I mean, certainly when we - we've talked about autostart and how much a concern it is that when you plug a USB key in, the OS will automatically run things. It always has made our listeners uncomfortable, and deservedly so, that an OS is configured by default, that is, modern OSes, where you plug the USB key in, and it has the chance to run code. I mean, that's the convenience, for example, of using traveler mode on a TrueCrypted drive is you plug it in, it is running the TrueCrypt driver automatically when you do that. Well, there's nothing to say it can't be running a RAM-sucking little executable that just simply copies all of the system RAM out to the thumb drive, I mean, literally to...

Leo: That's amazing.

Steve: Yes, to hijack your system. Nothing prevents that. Now, it's worth noting, though, that there are other vulnerabilities, for example, in typical laptops. For example, many laptops have a PCMCIA card or an ExpressCard. Those are the system bus. So, similarly, nothing prevents you, I mean, it is access to the system's bus. If you plug something in there, that device is on the bus, which gives it access to the system's memory. And for that matter, laptop docking connectors. You know, they're all different based on laptop make and model. But they're also the laptop's bus. So I sort of wanted to create a little more perspective on this whole issue and say, you know, physical access to our systems is almost never secure. That's the case. If you're going to allow someone

to spray Freon on your dynamic RAM...

Leo: What're you going to do?

Steve: ...and then take it with them, okay, that's not very secure. You can hope that they lose more bits than they plan to. But I guess the point is it's certainly of academic interest that data can be slowed down without power from loss in dynamic memory, and that clever algorithms can be used to reconstruct a low amount of bit loss in order to reconstruct someone's keys. But if you turn your laptop off and you count to 10, your data is gone. And it's certainly the case that now that this has gotten all the attention it has, and I'm glad it has, if there were any vulnerabilities, or if, for example, in the case of the TrueCrypt guys whose intentions are very clear, if there's anything they can do to minimize the window of exposure, they will. For example, you know, make it incredibly fast and easy to dismount TrueCrypt, a TrueCrypt volume, and wipe the key. And so that would require that you reauthenticate. But people who want that security may want to, for example, when you hibernate or when you suspend your system. It's certainly possible for the TrueCrypt drivers to see that that's gone on and then deliberately unmount as they're able to and then require you to reauthenticate in order to regain access to your now unmounted partitions.

Leo: I mean, I understand what you're saying when you say if somebody's got physical access to your system you're in trouble. But I guess the point of these full disk encryption systems is to protect them in that eventuality because otherwise you don't really need full disk encryption. I mean, if somebody doesn't have physical access to your system, what are you worried about?

Steve: Okay. Right. My feeling, though, is that the proper way to think about this is, if someone - if you lost your laptop, if you left it in the airport, or someone swung by and sneaked away with it in an area where it was unsecure, you absolutely want to make sure that, when the laptop is not in use, it cannot be put back into use without requiring full-strength reauthentication of its user. So it may be the case that, for example, just putting it into standby, as we know a laptop on standby is having its dynamic RAM refreshed. That's what standby is. It's the reason it's using a little bit of battery power. Unlike hibernation, where the whole contents of RAM is copied to the hard drive, and then the system is shut down, it is powered off, and you can pull the battery out of it for as long as you want to. That hibernating image exists then on the hard drive.

Leo: Now, you didn't address the issue of the hibernating image, did you? I mean, can you do the same kinds of tricks to that to get the key out of it?

Steve: No. We know that the hibernating image is encrypted. I don't know whether a decrypted hibernating image requires reauthentication. That would be something that we would want to check on. But, I mean, I would bet it does. I would bet that the keys are wiped, then the image is encrypted and written to the drive so that when it comes back you need to reauthenticate coming out of hibernation. I'd bet anything that that's the way it's got to be.

Leo: Well, until we find that out, though, the most prudent thing would be to shut it down.

Steve: Certainly powering off your machine and then not handing it to a stranger...

Leo: Immediately...

Steve: ...immediately...

Leo: What would be a prudent amount of time?

Steve: And especially a stranger who's holding a can of Freon.

Leo: Would a minute be enough?

Steve: Oh, 10 seconds, Leo, I mean, really...

Leo: It's pretty quick.

Steve: And anything even more recent is down at a second or two. I mean, they managed to find some old, less dynamic, dynamic RAM, less dense, where they were able to get, like, 10 seconds kind of. But, I mean, really...

Leo: Okay. So to be prudent, if you're using full disk encryption, and you shut your system down, and you hold onto it for 10 seconds, you're safe.

Steve: Yeah. Then give it to anybody you want.

Leo: Then it's a gift. Well, no. More to the point, then if you leave it in a taxi or it gets stolen you don't have to freak out because none of these techniques work unless the system is logged in.

Steve: Yeah, in fact I would argue that it's probably the case that with TrueCrypt you are more safe because given that TrueCrypt wipes the keys and encrypts the RAM to the drive, then you're getting explicit wiping of the mount of TrueCrypt. So the moment the hibernation is finished, it's safe. You no longer have that 10-second wait. If you powered down leaving the system mounted, then you powered down with the keys in service, in use, then you've got to wait a few seconds. I mean, most people are going to turn the machine off, then they're going to be wrapping up their cables and putting it into its laptop case and putting things away. I would say turn the machine, turn your laptop off first. And then when you're busy doing everything else to get it stowed, it's completely

forgotten everything it ever knew.

Leo: Right. So it's not sensationalistic, I mean, but in fact it's a valuable research project to let people know that this vulnerability even exists is saying something. I'm particularly interested in the USB key vulnerability. That's amazing.

Steve: Leo, it is a perfect episode for Security Now!. That's about what it is. I mean, it's really intriguing academically. But it isn't going to affect anyone's life because memory - DRAM doesn't forget what it knows instantly, but it does very quickly.

Leo: Fast enough.

Steve: Yes, very quickly.

Leo: Okay. Very good. I'm glad. I think there was a lot of concern, in all seriousness. There was a lot of - people were worried. And so nothing to worry about.

Steve: Well, or at least we...

Leo: We know what to do.

Steve: Yes, yes. I was going to say the threat is now understood. And I'm glad the TrueCrypt guys know. I'm glad the BitLocker folks know. And apparently there was some work that Microsoft did in awareness of this for the design of BitLocker. The researchers made the point, though, that there is - I think they call it Basic Mode with BitLocker where it works with the TPM module that we've talked about on the motherboard, the Trusted Platform Module. So I guess saying "TPM module" is redundant. I got two modules in there. It works with the TPM. The problem is, in the basic mode the TPM by default provides BitLocker with the keys on the fly without requiring authentication. So...

Leo: Then you've got a problem anyway. You turn on your machine, and hello.

Steve: Yeah, I mean, exactly. So anyone could take it and turn on and get what you want. So don't - I'm sure any of our security-conscious users understand that requiring authentication, while it's a bit of a headache, I mean, it's protecting you from exactly that.

Leo: Right. I never mind logging in. I know that I'm...

Steve: It feels good.

Leo: Yeah. It's safe, unless you - so I'm glad we did this. I want to reiterate that the problem was that we did the TrueCrypt episode ahead of time. And so I think the day before the TrueCrypt episode came out, but a week after we recorded it, this whole storm broke. So I'm glad we could, it took us a little time, but we could get around to it and talk a little bit about it. And as you can see, it was nothing to freak out about, nothing to worry about.

Steve: Well, and again, it did raise a lot of people's concern. I don't want to say nothing to worry about. I just want to say, well, probably, and...

Leo: Something good to know about, though.

Steve: Absolutely. It makes a perfect episode of Security Now!.

Leo: Well, as always. You are now going to pull ahead of This Week in Tech because I'm going to Australia. And so we'll - by the way, don't worry, there's a Security Now! episode next week and the week after. We're going to pretape. But not of TWiT, not of MacBreak Weekly. You're now pulling ahead. The only show that is ahead of you now - there are two shows. One's The Tech Guy, but just because it started in 2004, and it does two a week. And you can't - you'll never beat the Daily Giz Wiz because he does five shows a week.

Steve: That's cheating, Leo, Giz Wiz. I mean, could you not have numbered them, like, 1.1, 1.2, 1.3?

Leo: If we did that, you'd be winning.

Steve: Okay.

Leo: So next week Episode 138. TWiT is in the dust. Security Now! will be back next Thursday with the great Steve Gibson. And don't forget Steve's site, GRC.com. That's where you can go to get SpinRite, of course, the world's best, the world's only, really, true disk recovery and maintenance utility. It's a must have for recovering those hard drives that the boss is putting in the dark closet.

Steve: I hope Boss is wiping the contents of those drives. I was thinking about...

Leo: Employee is I'm sure.

Steve: Yes, yes.

Leo: If he's a good employee, he wouldn't dream of...

Steve: Yeah, he's filling it with MIT TV video.

Leo: Isn't that funny, he's got every episode of "CSI" on there. You know, I'm sure the boss is not wiping them. You know he's not.

Steve: Well, especially when the drive dies. It's a little hard to wipe a dead drive.

Leo: Yeah, how are they going to wipe it, yeah. So Security Now!, then wipe it.

Steve: SpinRite.

Leo: SpinRite. Security Now!, then SpinRite, then wipe it.

Steve: There you go, there you go. Then you're really covered.

Leo: When you go to GRC.com you also find many great tools like Steve's fun - and these are all free - Wizmo, which now has that, what do you call it, LAN Wipe, LAN...

Steve: Lanlock.

Leo: Lanlock feature.

Steve: I'm sorry, wanlock.

Leo: Wanlock, to turn off zero config, which is really handy. It does a lot of other stuff, too, fun stuff. And then all his great security utilities. It's all GRC.com, including 16KB versions of this show for the bandwidth impaired, and full transcripts so you can read along as you listen. I know many of you like to do that, as well. Steve, next week what do we do? It's I guess Q&A time, yeah?

Steve: We have a Q&A time and an interesting little bit of news. Someone has created a paper enigma machine. We've talked about...

Leo: Oh, cool.

Steve: ...the German Enigma machine. There's a paper enigma machine. You can

download a PDF and make one. So we'll be talking about it at the top of the show, then do our Q&A.

Leo: Oh, that's so cool. And if you don't know what an Enigma machine is, you're going to find out.

Steve: Yup.

Leo: All right. Well, that will be next week on Security Now!. We hope to see you then. Until then, Leo Laporte and Steve Gibson. Stay safe. We'll see you next time.