



Once Upon a Time

Description: For this special year-end holiday edition of Security Now!, Steve digs down deep into his video archives. He takes us back 22 years, to 1990, to share a 45-minute presentation he gave, once upon a time, on the inner workings of the "megabyte-sized" hard disk drives that gave birth to the PC industry.

High quality (64 kbps) mp3 audio file URL: <http://media.GRC.com/sn/SN-384.mp3>

Quarter size (16 kbps) mp3 audio file URL: <http://media.GRC.com/sn/sn-384-lq.mp3>

SHOW TEASE: It's time for Security Now!, a special edition on this day after Christmas. We're going to take the week off and let Steve do the talking from 1990. You want to know how hard drives used to work? Guess what, they still work the same way. Stay tuned. A trip back in time, next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 384, published December 26th, 2012: Back in Time.

It's time for Security Now!, a very special edition of Security Now!. We're introducing this the week before Christmas, but it will air on December 26th. Happy Holidays, Steve.

Steve Gibson: Hey, Leo. Thanks very much, and Merry Christmas. This is the day after the Christmas holiday.

Leo: Boxing Day.

Steve: Yeah, I guess probably many people who are normally listening to the podcast while they're commuting are probably not doing so this week.

Leo: We hope you're in your jammies at home, is what we hope.

Steve: That they're off. And actually that's a good thing because this is our special holiday Security Now!. We agreed to strongly endeavor never to have a dark week again. We made that mistake once and heard about it for quite a while. So the reason it's good

that people may be home is that this particular one really has to be seen and not just heard.

For some strange reason, my company recorded a series of presentations that I gave 22 years ago, when I would have been 35 years old. These were in Chicago for what was called the SofSel SofTeach, which was a multicity road show that I was invited to attend for our software distributor. This is back well before the Internet, when we copied software onto diskettes, and there were manuals, and they went in boxes, and you got them at Microcenter and Egghead and Fry's and regular boxed software retailers. So SpinRite at the time was new. And I was explaining to these people how hard drives work. And I ended up sort of having it down pat, with a patter, which when I watched it again a couple weeks ago, I thought wow, I was pretty funny.

Leo: You were a showman.

Steve: There's a bunch of stuff that I had forgotten I had come up with, like at the front of the audience, standing there with my arms by my side, my fingers sticking outwards, declaring myself to be a screw, and explaining the difficult job of a screw, which is holding down the stepper motor in the hard drive because it dare not allow it move, or the drive alignment will change. And so there's a lot of pantomime. It's very physical. I'm taking advantage of the fact that I'm there in body in front of the audience. So this podcast I'm always very conscious of the fact that the majority of our listeners are doing so in audio, so it isn't heavily graphics dependent. Back when you and I were doing the TechTV stuff, there was a lot of graphics because it was TV. Everyone was seeing it who was hearing us.

Leo: Right. There was no audio version.

Steve: Right. So I would urge our listeners for this episode, I think it'll be worth your while. You'll get a kick out of it. And I'd forgotten the level of detail that I went into in order to describe the things that SpinRite does. I would bet that every single listener learns something even now about hard drives that they didn't know.

Leo: It's actually funny because I would have, if you'd asked me in, what was it, '95, '96...

Steve: No, '90, 1990.

Leo: '90? If you'd asked me...

Steve: Yeah, 22 years ago.

Leo: ...in 1990 if in 20 years we'd still be using hard drives, I would have mocked you. I would have said, oh, no, that technology, that can't continue. We're going to be using holographic memory cubes, of course.

Steve: Well, actually, I even talk about those. I'm not kidding you, Leo.

Leo: Really. Really.

Steve: Those holographic memory cubes are in that video presentation 22 years ago.

Leo: But what's interesting is the same technology that you were describing 22 years ago is pretty much how it works today; right? With a few added details.

Steve: Yes. It's amazing how little of it has changed. Some has. But I just - it's a perfect, fun, wacky, holiday podcast. But again, let me urge people, I mean, maybe you'll not believe this, and you'll listen to it, and you'll go, okay, I have to see what he was doing when everyone was laughing at him. So in that case you'll probably end up watching.

Leo: You spinning around like a screw.

Steve: Yeah.

Leo: All right. Well, we're going to get to that in just a little bit, Steve Gibson explaining how hard drives worked in 1990. I can't wait to - do you have hair?

Steve: Yes, it's black. Oh, my god. And a huge black mustache.

Leo: I can't wait to see this.

Steve: Jenny just looked at it, because I showed it to her a couple weeks ago, she's like - she was just staring at me with her mouth open. She says, "I don't think I would recognize you." I mean, she knew me years before then, but then we had a gap in our relationship. And she says, "I don't think I would know who you were." So, yeah. It is quite a different look.

Leo: I love it. Is there anything you want to say before we roll tape? And I mean tape?

Steve: I just want to say Merry Christmas to all of our listeners. Thanks for being with us for this last year and heading into 2013. I'm sure we're going to have a lot more fun and great podcasts.

Leo: So let's spin the time machine wheel as we go back to 1990...

Steve: Where I had hair.

Leo: ...and a hairy Steve Gibson.

[Begin 1990 recording]

Steve: Let's talk about hard disks. If you ask somebody who has been associated with personal computers for a while what the least reliable component of a PC is, if you make the exception of the operator, they'd probably say, oh, the hard disk drive. The common wisdom now is not, is this drive ever going to die, but when? And then of course you ask the subquestion, yes, but exactly when? Because I'd like to back it up just before that, please. Of course they don't give you any notice, typically. And backing up is not a fun thing to do. It's a very noncreative experience.

And most people will tell you, "Oh, well, I know that I should have done that, but I just sort of never got around to it." Or, "Oh, after I had my computer for two weeks, I backed it up." And you say, "Yeah, uh-huh, when was that?" "Oh, two years ago." "And you haven't used it since?" "Oh, no, I've been using it, but it's been fine." Yeah.

Why do drives die? Why have they got this bad reputation that, sooner or later, they're going to give up? We know how to make things. We're a society that's very good about building things. We're putting men on all these other planets. And what we don't know how to do, the Japanese guys do. So together we've got the bases covered. Yet we've got drives dying. And the Fuji drives are dying just as quickly as the Seagate drives are. So what's the problem here?

Well, let's look at it. Let's step back for a minute and look at the technology in the drive and answer the question. So here's the drive, a MiniScribe something or other, 3650. It's got three surfaces, or rather three platters, six surfaces on each side. The engineers haven't figured out yet how to do a Mbius disk, but they probably will. That'll save them one head for each disk. So in this case we've got three platters and six surfaces. And one head for each surface. This is a so-called "head tower," where the heads are all mounted. And each of the little heads are sitting on one of the surfaces. Then we have a little rack-and-pinion mechanism down here that translates the stepper motor's rotary motion into the linear motion for moving the heads in and out. And that's basically it. And this of course all spins around in order to allow us to get to all of the area on the different tracks.

What we need to do in order to store data in this thing is to establish addressability. Right now it's just sort of this empty domain of magnetism, potential grazing land for a magnetic field. We need to establish addressability. And we want to also protect ourselves against a problem which these drives have by virtue of the technology in them.

Let's look at that. If we have perhaps multiple surfaces, each with a head, so we need to select a head. So that's one dimension in our address. Then we need to know where on the surface, in or out [indiscernible], which track or cylinder on the surface we want to be. So that's a second dimension. And, finally, where around the selected track are we wanting to do our reading and writing? So it's a third dimension. So it's sort of half of a 3D coordinate system. Which is not surprising because we live in a 3D coordinate system, and the drive occupies a volume of space. And so we need to talk to specific areas of it. So we have 3D.

Well, in the cable that goes between the controller and the drive are some wires known

as the head-select lines. And taken in combination, they determine which of these surfaces is the active one at any given moment. It would be a problem, though, if the controller said I want to be on surface No. 4 and sent that signal through the cable, through a couple of these head-select lines, if there was a break in the cable or a bad connection at one end or some goo on the connector that kept it from mating correctly. You wouldn't want the drive to hear the head 4 as head 2 because then it'd be working on the wrong surface. So we need some way of knowing that we really did get the surface over here that we asked for over on the controller side. So that's reliability in that first dimension.

Secondly, we need to make sure that we're on the right track. These little stepping motors are nifty gizmos. They are getting the speeds of them up so that they're actually cutting into the market that the older voice coil actuators used to have. Voice coils tend not to be as reliable as the stepping motors. But stepping motors have been known to misstep. Naturally there's pressure on the engineers who design all this stuff to make them get where they're trying to go as quickly as they can so that the manufacturer can say we have a 63ms seek time, or 25, or whatever it is. So they push them maybe a little too far, enough so that every so often they misstep, so they don't actually get to the track that they were asked to go to.

Well, if we were on track 25, the controller said I want to do something out on track 100 so go out 75 steps. And if we didn't, if we threw in a freebie there and went out 76, or only 74, and didn't end up on track 100, that's a problem. It's not good enough just to write our data somewhere in the neighborhood. We really need it to be on the right track. So we need a scheme for allowing us to address the data in the drive that prevents these kinds of confusions.

Also, we need to chop our tracks up into smaller pieces. A track formatted with the least dense technology, called "modified frequency modulation," has about 80,000 bits around its circumference. That's about 10,000 bytes. Well, that's a lot of bytes to just allocate in one lump. If we had a little 40-byte config.sys file that said buffers=20, files=20, and we could only allocate with the granularity of one track, well, it would take up 10K of the disk just with that little 40-byte file. So we chop the tracks of the drive up into pieces. In fact, these pieces, it's nice, are all 512 bytes. It's nice because it turns out that's a galactic standard. Aliens have chosen 512 bytes also for their packet size. They don't have all this mechanical. They've got those neat little cube things that are all solid, and they work with laser beams and things. But they use 512-byte molecular-string DNA polymer manipulation. So, still, that means that they'll have an easier time getting to IBM compatibility.

So, where was I? So we need to label these sectors uniquely and give us some prevention against any kinds of problems. So that's where low-level formatting comes in. Low-level formatting establishes addressability on a drive. Basically we go to every single track on every surface on every cylinder and everything and lay down little signposts. These are called "sector IDs." And they're just periodically popped out on the disk as it's spinning. It spins at 3600 rpm, so that's 60 rps. It's whizzing. And so the controller in low-level formatting just blasts these out of the head every so often, and that spaces them evenly around the track. Then it goes to the next head, blasts those out; the next head, blasts those; the next head and so forth, until it's done all of them on that cylinder of tracks and moves to the next one.

Each of these contains three vital pieces of data: the cylinder it's on, that is, that they track in and out; the surface it is a member of; and which sector it is around the disk. Now, that may seem a little bit redundant because, after all, all the sectors on this surface, they all have the same surface number written into them. Correspondingly, all

the sectors on this track have the same track number. Except that there's a benefit here. If we encode the complete address of each sector in the sector's header, then all of them are unique. So we've completely uniquely labeled all the sectors in this 3D storage volume. Because of course you can't have two sectors that are in different places at the same location, by definition.

Okay. So how does this work? The computer says, "I want to write into Sector No. 1, please," to the controller. The controller, that's what it's here for. It says, "Fine, whatever you say, boss." So this is Sector No. 1. Well, we never know where the disk is at any given time as it's spinning around. So we just start listening. We put the head on the right track first and say, okay, we're on the right track, so to speak. And we start listening now. We open what's called a "read channel" from the proper surface, listening to everything going underneath the head. It's a very, very delicate magnetic listening device. And as each of these little signposts comes along, we check it and see, are you the one? Do we get a match here? No? Okay. And wait for the next one. You? A match? No? Okay. And so on. Eventually we're going to find it.

So imagine the head running along here. It finally hits this sector, this sector ID. It runs through it and says, ah, this is the guy. I got a match on everything, exactly what I was hoping for. In a little moment called the "right switching interval," right afterwards, we turn the juice on to the head. We switch it from a read-mode head, where it's a delicate magnetic listening device, into a big magnetic plow, where we pour juice in it, and it goes [vocalizing], leaving ones and zeroes behind in its wake. 4,096 of these ones and zeroes form a 512-byte sector. So we have essentially written data into the gap between these two sector IDs, which is really the sector region. And the controller reports to the motherboard, okay, got it. Motherboard, tending to be suspicious, says, oh? Uh-huh? Well, read that back for me. So it's fine.

So again, it's down here somewhere by that time. It again just starts listening from all these little sector IDs as they come along until it finds the right one, runs across Sector No. 1 and says, okay, here we go, and then starts paying attention to the data. And it sucks in these 4,096 little bits into a buffer that's on the controller. Generally it's this chip here. And it gets them all together, that one sector's worth, and then adds them all up, adds the bytes up to make sure that, with the addition of a checksum here at the end, a little gratuitous byte thrown in, that we end up with a zero sum. That's just a quick way of knowing, hey, all the bits are probably okay.

So we say to the motherboard, okay, I've got them all here. Motherboards says, well, give them to me. So they set up something called a DMA transfer, Direct Memory Access transfer, to move the data down through the I/O connector onto the motherboard and over to some little buffer or something sitting there. And then we're really done reading this one sector. So the controller says, "Okay, now we're done." Computer says, "Well, yeah, but, what, one sector? 512 bytes? What can you do with that these days? Have you seen new Lotus?" "Okay, I'll get busy."

So the problem is now the computer wants a second sector. But while we were doing - we got this all sectored in, and then the disk kept on spinning while we were doing our checksumming and stuff and our DMA work, until we got to here somewhere, and then it said, "I want No. 2." Well, No. 2 went by. We were on it for a while, then - now we're on No. 3 even, probably. But 2 is gone. But that's not a big problem. Columbus demonstrated how this works. You just wait long enough, it's going to come back around, and we'll be fine. So it starts looking for Sector No. 2, sure enough runs across it and reads in that sector. Then the computer says, okay, we've only got about another three megs to go, so I need Sector No. 3. Okay. The problem is same as before. We've moved past the beginning of 3 at least, by several sectors, perhaps. We've got to go all the way

around again.

Well, so the system is working. We're reading and writing data. But it's non-optimal. We're spending much more time waiting for the sector that we've been asked to find, after just finding one, than we are actually transferring data.

Well, IBM invented all this stuff. And they had a solution for it. They said, let's not put Sector No. 2 right after Sector No. 1. No one says we have to. Remember, it's just what's written in the ID that says I am Sector No. whatever I am because we just listen. We listen for them to see which one's going to go underneath the head next until we find the one we want. Let's put Sector No. 2, oh, let's give it some time. If it needs still down here somewhere, let's put Sector No. 2 here, and a couple more sector spacing, and No. 3, couple more sector spacing, and No. 4, and so on, and continue skipping a few, running around until we finish numbering all 17 of them. Essentially we're interweaving all of these guys.

Okay. Now the computer says, let's try this again, see what you can do. So last time we were only getting one sector per revolution. Every time it asked for it, the one we wanted just went by. So 17 sectors meant 17 revs. Now, it's worse, in fact, if you have a higher density drive. This MFM, remember, gives us 17 sectors. We have RLL encoding, stands for Run Length Limited encoding. It's a somewhat more aggressive means for encoding the data. That gives us 26 sectors around the track. And then we have even one step further, ERL, stands for Enhanced Run Length Limited encoding. Actually, in my experience, this stands for Expensive Run Length Limited encoding. But so it can get bad if your interleave is wrong as you crank up the number of sectors that you've got on track.

Okay. So now the computer says, "I want to go again. Sector No. 1, please." So average about half a revolution, and we find Sector No. 1 and suck in the data during Sector No. 1, and then do the checksumming, set up a DMA transfer, move down to the motherboard, get everything finished, and we say we're done. Computer says, okay, I want No. 2. Perfect. Here's No. 2 in front of us and heading towards us. It's the next one we're going to encounter. So we read through No. 2. And then after being done, do our checksumming, use the DMA transfer, move down to the motherboard, then the computer says, "I want No. 3." Again, perfect. So in comes No. 3. Well, this is a much-improved situation. We're now reading almost all the time. And in fact we're reading every third sector each time around. So clearly, in three revolutions, we've read them all. Much, much better.

Well, IBM was creating the PC XT. They knew about interleaving. But being IBM, they really didn't have a clue what the right number was. But they didn't worry about it too much. They figured, hey, you know, we're IBM. We don't have to be right. We just sort of have to be here. People are going to buy this stuff anyway. And they did know one thing that was critical. They knew that it was better to have the sector that was next in line and was going to be asked for out in front of you and heading in your direction than to already be standing on it or to have it just having passed by and leaving.

Well, that suited them fine because conservatism is their nature. And they said, well, how about six? Six sounds good. That's certainly going to be out there in front of us, no matter how slow we make this XT, so that'll give us lots of time. So in fact that's what we got. They shipped the IBM PC XT, the blue one, with a hard disk interleaved at 6:1. So it took six revolutions to read one track of data.

Okay. Then along came clones. And hard disk controller maker to the clones was Western Digital, who said, oh, let's make some controllers here because these IBM PCs or

compatibles are going to be hot sellers. So they made a family of controllers, the 1002-WX1s and that whole lineage. They said, we want to compete with these Big Blue guys, and we're in California, so we're expected to be faster and crazier. Let's set our interleave to 3:1. That's much better. In fact, it's twice better. IBM is interleaved at 6:1. Six revolutions to read one track of data. We, Western Digital, will have our disk controllers default their interleaves at 3:1. So we can read the same data on the same track as IBM in half the time, meaning that our throughput is twice as fast. So that was pretty great.

Well, then I came along, and I had been writing a column for InfoWorld for about six months at that time. That was about two and a half years ago. Now about three years I've been writing the TechTalk column every week for InfoWorld. And I wanted to write - I wanted to address the issue of performance, hard disk performance, but a different dimension of it than had been spoken of before. Everyone knew about average seek times. You open up your mass mail catalog or your mail order outfit, and they show you, oh, the 80ms worm drive drive, I mean, at the low end, or this 17ms screamer at the high end. You know when you have one of these slow ones because you can kind of hear it arrive at the right track. And you know when you have a fast one because your bank account is empty. And then there's everything in between.

So people knew what their average seek times were. That was no mystery. But for me, I wanted to know, hey, once we get to where we've gotten with whatever the seek time is, once we get there, how long do we have to stay before we can leave? I always find myself asking that when we're going to visit my mother-in-law. How quickly can we get this over with and get this data transferred and get on to the next track? So that was really the question that had not been addressed.

There was a utility out there called Coretest that everyone sort of knew of. It was in the public domain, Coretest. And it had been around since the dawn of man. I mean, it was originally named the Rocktest, and cavemen used it to see how round their rocks were. And they upgraded it finally at, I think, rev. 9.7 and made it IBM compatible, and they renamed it the Coretest and then put it back down to rev. 1. Well, you could run it, and it would tell you something. It would say, you have 47,926 bytes per second throughput. Is that good news? I don't know. The only thing you could do, I guess comparatively you could have all your friends run it. And if theirs was bigger than yours, then you weren't so happy.

I wrote something that delivered the information in something that I felt was what was really happening, with a little thing I also put in the public domain called SPINTEST, SPINTEST and SPINTIME, two little - it was a little 183-byte thing. I write everything in assembler. Even SpinRite is 100 percent 500K of assembly language. So I took SPINTEST and stuck it into my PC, my PC XT from the Blue people, and it said six revs. It was to be expected. Yeah, there's six, blah blah blah. Went over, and I stuck it in my clone computer, and it said 17 with my Western Digital controller.

Now, it always did seem a little slow. I figured that was sort of generic cloneness. But now I thought, wait a minute, this is just mis-interleaved. Western Digital chose this 3:1 interleave. It must be that it needed 4. It needed a little more time. The computer wasn't getting done somewhere here before Sector No. 2. It's probably just barely standing on it or somewhere in it, and it needed No. 2 to be moved back here. So it was getting the worst possible interleave. I mean, if it's standing on it, I'd rather have two back over here, where it would have been if I didn't have any interleave because I wouldn't have to wait quite so long. It can't be any worse than the one too tight. That's as bad as it gets because you're on the one you've just been asked to get.

So I said, well, let's find out if this is true. So I got 75 floppy disks and formatted them all. I waited for morning so I was fresh. Then I started pumping those floppy disks into this puppy and backed up my whole hard drive. Then a real experience of user-friendliness. I typed "debug." I got the famous minus sign prompt. You bet it's a minus sign, not a plus sign. And I wanted to override the default low-level format, which was 3. I was going to force it to give me a 4 and see what would happen. So I put a 4, RAL4, put a 4 in the AL register of the AX accumulator and a 1 in the AH register of the AX accumulator. Then I typed G=C800:5, and it said "Welcome to Western Digital." I said, yeah, right.

And then it spit out something about ooppeerrrruuu, and I was supposed to figure out what my right precompensation cylinder was, and where my right curve was to be reduced, I don't know from where to what, and about my stepping rates and something. And I figured, you know these engineers. This is just probably bullshit. It doesn't matter. So I typed in some numbers, and off it went. So everything seemed fine, and in about 10 minutes it was all done. And so I needed to put all my data back. So I spent the last half of the day with my 75 floppies, putting them back in, and then rebooted. And it did seem faster. So I did a few little things, familiar things. And it was definitely running better.

So I took SPINTEST, stuck it in there - 4 revs. Had been 17 that morning, this same morning. Now 4. So I said, my goodness. 425 percent difference in throughput. Think if I could do that with my mother-in-law. I could have four of them. So I thought, how widespread is this disaster? To how many systems did WD do this? So I had since published my column and put the program out into the public domain. And I was getting letters back: Steve, it's 17. Steve, mine's 17. Steve, it's 17. I went out on the road with my own disk - this is before you had fear of viruses and things - stuck it into all the computers I could find, without fear. And I tell you, Epson was at 17, and Kaypro was at 17, and Leading Edge was at 17. They were all at 17 revs. They were all wrong. I think Compaq might have been at 6, which figured. They were sort of beginning to follow IBM's conservative approach.

So I thought, my goodness. If I could come up with a way of nondestructively low-level reformatting a drive, I could probably drive any kind of car I want. And CompuServe, the guys on CompuServe sort of heard about this. I was talking about it. Because I was enough of a figure by then, with the InfoWorld column, and I had FlickerFree, which was my first little TSR product for the IBM that eliminated the flicker from the scrolling of the CGA adapter and sped up other stuff, as well. And I had done light pen stuff back in the Apple II days, a high-resolution light pen for the Apple II. So they kind of knew me.

And they said, Gibson is talking about doing a non-destructive low-level format. What has he been smoking out there in California? Low-level formatting is a wipeout. And of course they're right, it is. I mean, the worst thing that can happen to your computer is that it catches a software virus somehow, and then you innocently do something to piss off that virus, and it sticks it to you in your low-level formatting. It's over.

What they didn't know is that there's another command than low-level format drive that the viruses know about down in the low-level IBM BIOS. There's one called "low-level format track." Just one track. Any one you want. IBM probably put it in there because they thought maybe someone would come along and write some software that would fix a blown track. No one ever did, as far as I know. But I thought, yeah, I can use this. I'll start with the first track. I'll read all the data off of it, low-level reformat it at the right interleave, after figuring out what that right interleave is, and put the data back and go to the next one. I thought, how hard can this be? A couple late nights? A long weekend?

Well, a year later I was wrapping up because it turned out there was a lot more to it, but

more importantly, that I could do a lot more good. Peter Norton was the first person to figure out that, when you deleted a file, it really wasn't gone; and that, if you asked for it back before waiting too long, you undeleted it, you could have it back. As a consequence of that little discovery, I guarantee you Peter is now driving any kind of car he wants, and in whatever town he owns.

And actually we did lunch a couple months ago because he wanted to acquire this. And he said, "You know, Steve, when you came out with this, I thought you were just going to crash and burn. You were going to just nuke yourself on the spot." He said, "But then the reviews started happening. People were talking about it." He said, "So we went out and got a copy." And I said, "I bet you did, Peter. What's in here?" And he said, "You put a ton of technology in there to make it safe and to really create some value." He said, "You want to sell it?" I said, "No. No, thank you. I want to do more good things instead."

What I figured, I've got the data off the track. I've just low-level formatted the track. Is this a safe place to put the data back? Why not find out? Because there's this whole world of defects on drives. It's not good news, but they have them. The manufacturers can't be very happy about it, but they print a little list of defects. MiniScribe has a small list. Seagate you just sort of scroll. See page 9. Oh, okay. Where is that? They're not happy about this, but they've got defects. So I figured, hey, let's just check on some defects while we're at it, see if this is a good place to put the data back.

I did a lot of R&D. I'm fundamentally a hardware guy, which is I think why I write good low-level software and I'm an assembly language fanatic. So I got out all my equipment, and I learned a lot of other interesting things, too. I found out what the biggest nightmare that these drive designers and makers and manufacturers have. Their biggest nightmare they call "long-term drive alignment drift." Long-term drive alignment drift. Look at this. It's all made out of metal. It's got yellow metal, silver metal. It's got real nice sort of expensive-looking chrome kind of metal. It's got some tastefully matte finish metal and yellow and silver and all kinds. It's all screwed together with screws, little screws down here in the motor and everything. And these get very hot when you run them. You put your hand on a drive after a couple hours, whoa, baby, I mean, this is where your 220 watts goes in your power supply, goes to heating this puppy up. They make good irons. Actually the Seagate ST225s with the [indiscernible] edges, oh, boy, they won't catch on your pockets and things. That makes a great iron. Or a doorstep.

So it's all made out of metal. Well, what does metal do when you heat it up? It expands. And in fact, different metals expand at different rates. There's something called a "coefficient of thermal expansion" which describes the rate at which metal expands as it gets hot. So here we have this thing made out of all kinds of different metal. We don't have the alien technology yet to just make it one humongous lump. So it's complicated. It's all screwed together.

Okay, now imagine I'm a screw. These are my threads. And I'm this screw, the one that's responsible for holding down this corner of the motor, keeping it in place. Now, you've perhaps never spent any time contemplating the life of a screw. It's a very simple life. A good screw just really - calm down there - just wants to be the very best screw possible. That's its whole purpose. It's a simple request, I know. During manufacture it got very well screwed. And it's going to try to, by definition, to hold everything in place. Yet it's made of a different metal than this base plate that it got screwed down into, and yet a different metal than this massive black intimidating kind of motor which it's been told it has to hold absolutely motionless. The motor's made of iron. The screw is made out of steel. It's in an aluminum base plate.

All of these things are doing their own thing as you heat this up. And we've already seen

it gets very hot in there. So it heats up. The screw expands. The hole expands that the screw is in. This motor is tugging on it because it wants to expand at a different rate. The three other screws, they're all trying to do their own job of being good screws and hold that motor absolutely motionless. Well, it's amazing to me that it works both hot and cold. And in fact many people who specialize in recovering data from drives do so by deliberately changing the temperature of the drive because it's known to have an effect. The point is the alignment is not even constant from morning until night.

Now, the end of the day happens. The computer gets turned off, begins cooling off again, and our little screw is getting pinched. The motor is shrinking. It is shrinking. The hole is even shrinking. All these little threads that have been trying to hold on for dear life all day are getting stretched back down. Is this in exactly the same place, exactly as it was in the morning? No. It can't be. We've got molecules, after all, and they're kind of grainy down on that level. It's amazing to me that it works the second day after one of these thermal expansion/contraction things.

The fact is it doesn't work after a couple years. It begins failing. This is a mechanical device. Wear and tear. It's got a little rack-and-pinion system back down here with gears that are wearing and rubbing on each other over time. So the alignment is going to drift. Long-term drive alignment drift. Now, of course you've got gravity. Gravity applies. It's a force. So amid all this expanding and contracting and all these little screws trying to hold themselves exactly where they are, there's a siren call of gravity - hello, come over here - kind of pulling at everything. And these little screws are saying, okay, I'm trying, and they're going to succeed to some degree, to varying degrees, varying from drive to drive. Long-term drive alignment drift.

So let's say, for the sake of argument, that our tracks are drifting inwards. Sort of a function of the way the drive is designed, how it's aging, the relative tightness of the various screws who are tugging on each other, and who's winning in the long term. But what does that mean? We saw that we've got defects. So there are defects wandering around in here, in various places on the surface. Clearly, drifting alignment affects defects because defects are on the surface. They're surface defects. They are of the surface. And they're not moving. But the tracks are moving relative to the defects because, after all, the thing that determines where a track is, there's no grooves in the surface. It's a smooth, uniform surface.

What determines where the tracks are is where the stepper motor ends up with all its mechanical connections, holding the head. That's the track. And if, over time, this is aging and altering its alignment just subtly, then the head is going to be in a slightly different position on track 100 than it was when it was made. So if a defect was right there, but the track has drifted away, the defect has floated, essentially. So that, for example, this defect is in between tracks. The manufacturer didn't find it. But a year later this track has drifted inwards and is now smack dab in the middle of that defect. So we have a new defect. This one was found. But because all the tracks were migrating as a whole, that track has correspondingly drifted inward out of danger. No more defect there. Fact is, defects are not a static phenomenon of a hard disk. They are inherently dynamic in nature.

Now, MiniScribe has this little chart that says "As Shipped Defect Criteria." I always thought that was interesting. "As Shipped Defect Criteria." One good UPS man can triple the length of this list. But I don't think that's what they're referring to. I think they paid some attorney \$20,000, or on retainer, to say, you know, "You're telling me that the alignment is changing on these drives?" "Yes, sir." "Well, what does that do to the defects?" "Oh, well, they change, too." "Then we'd better not say 'defect criteria.' We'd better say 'as shipped defect criteria.'" "Whoa. You're worth your money, aren't you, Mr.

Attorney." "Well, that's why you pay me." They know this is going to change. Doesn't help their sales any if they advertise it, so they don't talk about it much. Besides, they've got our money and cashed our check. And this is probably good for a year or two, six months. If you have a Seagate, you measure it in weeks. I'm just kidding about Seagate, of course. They're just the biggest.

What about our data, though? That's why we're here is our data, not defects. Well, something interesting happens with the data. If the head drifts a little bit from where it originally was, but it's still able to read the data, and it never writes it, the data gets realigned. And if there was a gradual long-term alignment drift, and we were periodically realigning the data, well, it tracks with the drifting alignment. So it balances out. Over time, the tracks are drifting. But the alignment is drifting.

The data is tracking [indiscernible] sort of an overall upheaval of the drive that rewrites it all will do that. Oh, my god. There is an empty cluster at the beginning of my drive. Well, we can't let that stay there because that would make a fragment. So that's where our optimizer can pick up 70MB and move it over. One cluster. Got that little sucker closed off now. The little opening is now over there with the rest of the free space. In the process, we read and wrote the entire drive and inherently realigned all the drive's data. So the data is tracking with this drifting alignment.

But there's something which isn't. There's something which has absolutely no opportunity for this realignment. That's the low-level format. It's written during the first minutes of this virgin drive birth and never again. We just read it. Even when we're writing data into the sector, remember, we find the sector by reading the low-level ID first. Over time, the alignment drifts so far that the head passes right past, and you get "sector not found, abort retry ignore." Drives need some respect. They're going to get it from you sooner or later. It's less painful if you give it to them sooner.

SpinRite does three things. The first time you use it, it optimizes the interleave for you. If you didn't have the next sector arriving as exactly the one it was being asked for, it will change it so that you do. And perhaps you get a free performance boost for all time, just by numbering these sectors properly. More importantly, I think, it is a long-term, low-level format maintenance tool for hard disk drives. Run it three times a year, four times a year, weekly if you use Seagate drives - I'm just kidding - and it will keep the drive aligned. It will scrub the surface for defects that are coming and going in and out of the tracks, literally tracking them as the tracks drift across them over the years. There's no reason our drives have to be unreliable. This is why they are dying over time.

SpinRite has proven effective over and over and over in keeping drives alive. It's simple to use, simple to sell. It won't ask your customers about cylinders and heads and sectors. You can't make it ask you those things. You just run it, and it works. Thanks very much.

[Applause and SpinRite giveaway]

Leo: Wow. You did have hair. Oh, I loved that. Now, tell me again, that was a lecture you gave at SofSel?

Steve: It was, yeah, it was a series of presentations. And, frankly, what I heard was - there were many vendors that were hosted by this big software distributor. This was the largest software distributor in the country at the time. There were many vendors who were hosted. So people had to - and there were only so many slots during the day when you could attend one. And so attendees, who were retailers - you'll have noticed that I

referred to what makes SpinRite easy to sell because I recognized that my audience were resellers. SofSel was distributor. These guys had computer stores, and so they would be buying my product from my distributor, SofSel, and then reselling it to the end user.

Leo: Ah, the good old days of shelf space. You don't do that anymore.

Steve: Exactly. And so - I just completely lost my...

Leo: I'm sorry, I didn't mean to throw you. So you were just saying this was to the distributors and to the end - I didn't realize it was also to shop owners who would buy from SofSel. And so they would record this because not everybody could get in to see it. I think that's where we were headed with that.

Steve: Oh, I know what I was saying, right, was that there were many more vendors making presentations than there were time slots. So it was necessary to select those which you would see. And what I ended up hearing through the grapevine was that the word spread like wildfire that you had to come see Gibson.

Leo: Oh, that's great.

Steve: So our little room was like standing room only, packed with people all lining the back and the sides because there weren't enough chairs because I ended up draining the audience from the other presentations. And in fact the other vendors wanted to come to see what the hell was going on in this one room. So it ended up being a lot of fun and some good laughs, too, as we all saw.

Leo: I just want to show you. In that year, in 1990, this is what a Mustang GT looked like. Now, and I should show you my Mustang. But let me tell you, this is a good moment to take a break. When we come back I want to talk a little bit about what you would amend from that speech, what's changed in hard drives.

Steve: Ah, right.

Leo: All right. So what is, I mean, the fundamentals you describe in that video are the same; right?

Steve: Yes. Probably the major change has been, well, obviously density. So I talk about MFM and RLL and ERLI encoding, not too much, but that's all in the past now. Now we have PRML, which is as spooky as it sounds, which is Partial Read Maximal Likelihood, where our drives are literally guessing at the data because the bits are so tightly packed they really can't see them any longer.

Leo: Wow. Wow.

Steve: So that's one change. But also the addition of servo information. That was the big change. It used to be that we had what I would call "dead reckoning" head positioning, where there was like a stepper motor and a rack-and-pinion that would just go out a hundred steps, and where the head landed was where the track would be. But that meant that you had this long-term drive alignment drift problem, which was one of the things that SpinRite really excelled at because, as I explained in the video, only low-level reformatting the drive before the alignment got too bad could cure the problem. And of course SpinRite famously was the only nondestructive low-level reformatter. That's what it did. And that's why Peter Norton, when he wanted to buy it, said he thought he was going to look down towards Irvine and Laguna Hills and see a big mushroom cloud coming up because he just didn't think you could do that safely.

So that's the big change. But when I talk about sector marks and sector headers and - of course interleaving disappeared as the data channel got fast enough so that we were able to suck in all the data off the drive in one revolution, rather than needing, like, a 3:1 or a 4:1 interleave, so it would take three - you could only get every third sector or every fourth sector per revolution. So that changed. But the underlying technology is the same. And so many things about computers are like that. I mean, when you were first saying that, Leo, I was thinking, yeah, word processing. We have GUI, but we almost had it back then. I mean, we had Windows 3.0 back in that era. And we had Micrographics Designer was that great graphics drawing program. And we've got more buttons now, but it really hasn't changed.

Leo: Well, maybe it's changed because you can put it in your pocket. That's a change.

Steve: Yeah. That, and run it on batteries.

Leo: And run it on batteries. Yeah, that's...

Steve: And it costs about one one-millionth the cost for the equivalent power.

Leo: Per MIP, yeah. Really fun. I think we should do more of this. You have more of these?

Steve: I do.

Leo: Because I think this is, well, it's educational. I mean, there's a lot of people listening who were kids in 1990. Probably most of them. And so I think that a lot of people - this is good stuff. This is fun to see. Anyway, I think it a perfect thing for our holiday episode. We will be back on January - will we be, January 2nd, doing a show?

Steve: I'll be. I hope you're here with me.

Leo: I'll come in. Might be a little hung over still. No, I won't be. I'll come in. January 2nd will be our next episode. That's a Wednesday, 11:00 a.m. Pacific - that's when we do Security Now! - 2:00 p.m. Eastern time, 1900 UTC, on TWiT.tv. We would love to see you live because we have the chatroom. But if we can't, that's okay. Download a copy. We have on-demand versions available a lot of places.

Steve has 16Kb versions. That's the smallest audio format. He also has transcripts. Those are even smaller. Most people get those and listen, I think. We have higher quality audio and video, too, at TWiT.tv. Steve also has SpinRite. Now that you know how hard drives work, maybe you ought to get a copy there. That's the world's finest hard drive maintenance and recovery utility still, to this day. What version of SpinRite were you talking about on that?

Steve: Oh, that was...

Leo: 2?

Steve: That might have been 1. Or maybe 2. I mean, SpinRite was not very old at the time, so this was an educational tour. This was to say, hey, folks, there's something you haven't seen before. And, I mean, it was all...

Leo: You know what's cool?

Steve: Yeah.

Leo: If you'd bought SpinRite in 1990, you'd still have an upgrade path to SpinRite 6; right?

Steve: That's true. That's true.

Leo: Free upgrades.

Steve: Even the very first person 22 years ago, we'll give you a discount on 6.

Leo: Discount. Okay. Not free. Let's get that straight, Leo. GRC.com. Go there right now and see - and also lots of free stuff there from Steve. Steve, I'm so glad we could talk. And have a great New Year, a happy holiday - because we're recording this before Christmas - and we'll see you in 2013.

Steve: Can't wait.

Leo: On Security Now!.

Copyright (c) 2012 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/>