PROFESSOR: So welcome everybody, and I actually used to be at MIT in the '90s, so it's good to be back. And so we're going to talk today about a different kind of security. It's going to be less on the technical mechanism side, and more on the, well, what happens when all this technology gets put in place in something where there's high consequences? Not quite so high-consequence as, say, an airplane in the sky, but getting pretty close.

Just to let you know where I'm coming from. So I used to be part of the midnight coffeehouse club myself, but this is Michigan, actually. We're not quite as big as your campus here. But a short while ago somebody decided to put a hot tub on our computer science building, so they're doing research inside there. But what we're going to talk about today is some of the research that bubbled out of that.

So we're going to talk about everything from exploding defibrillators to other issues of privacy in medical devices. And this mainly is going to be related to just one thread of research from one of my former graduate students here, who is actually at this point sanitizing explanted pacemakers. But we're going to mostly talk about the security of medical devices today.

Got a bunch of acknowledgements. There it is on tape. This work is by tons of people, and I'm going to try to summarize for you some of the modern bits about medical device security through all sorts of places. I'm also required to put up this boilerplate slide of my potential conflict of interest, so now you can know about any potential biases on my thinking. But I'd like to think that I am less biased than the average person. OK.

So moving on. So an interesting thing happened about a year ago, when FDA-- the Food and Drug Administration-- released a draft document saying they are now going to be expecting manufacturers to consider cyber security-- or as we call it, security and privacy-- not only in their implementation of the medical device software, but in their design of their software. Before a single line of code has been written. And so we're going to talk about how this has affected the thinking in the medical device manufacturing community.

Their final guidance came out just a couple weeks ago, and we just held a conference call. FDA held a conference call, and over 650 people decided to join the teleconference. So there's a lot of interest in the manufacturing community about how to take some of the concepts you're learning here in your class and actually apply it to the medical community.

But it's really hard. And I noticed one of the questions up on the website was about how to get the culture change in the medical community to understand security. And this slide illustrates that.

So, who washed their hands this morning? OK. Oh, this is not MIT, everybody.

So actually about 164, 165 years ago, there was a famous physician, Ignaz Semmelweis, who was looking into something called childbed fever. And he discovered that his medical students who were working in the morgue in the morning who later went to work with patients, well, those patients tended to die more often. And he discovered if you washed your hands, then statistically you were less likely to pass on some kind of probability of not living longer. So he recommended that physicians wash their hands.

And the reaction from the physician community was, doctors are gentleman, and therefore their hands are always clean. And to some extent we're seeing some of those kinds of attitudes toward security today, so it's not too surprising. But I'll try to draw some parallels with that throughout the talk.

I've got a lot of material to cover so I'm going to whip through some things. But first thing I'm going to do-- anyone a physician? No? OK, well you're all going to be able to have some good material for cocktail parties with your doctor friends.

We're going to talk a little bit about implantable medical devices. Actually I'll pass this guy around. You can feel free to touch it. It's been de-dangered. Just don't lick it.

This is a defibrillator from a former patient. And actually this is a device here-- about 50 years ago, some of the first pacemakers started to appear on the scene. They were external. You had to have a burly nurse to cart it around.

And then as the decades wore on, they became small enough to be implanted, completely implanted in the body. And here you see a picture of what's called a wand that's using inductive coupling. It's technically wireless. There are no wires. To wirelessly program the device to be 60 beats per minute.

But interesting to me as a security researcher was that in around 2003 or so, we began to see defibrillators, such as the one I'm passing around, that started to embrace wireless technologies and networking that you'd be more used to as sort of general computation. And I was thinking what could possibly go wrong?

Luckily there are a lot of engineers also thinking that same question in companies, but security, it takes a different mindset. And I'm going to tell you a little bit about how that mindset is changing.

So if you were to open up one of those devices, what you find inside are vast resource constraints. If you want a hard engineering problem, pop open one of these devices.

So about half of the device is just a battery. A very high quality battery. These cost about $40,000 a pop on the market. Silver vanadium oxide. And you've got little microcontrollers at the top. Typically you have some antennas where you can do your communication for your control of the device as well.

This is all hermetically sealed, implanted in your body. We're talking one of the harshest environments possible. You want to recharge a battery in your body, good luck. Did you know that batteries give off heat and gas? So there are very challenging constraints to engineering the device. When you want to add security, it gets just a little bit hard.

So there is, however, a very good reason for having a wirelessly controlled medical device. There are good reasons, but there are these subtle risks. So to illustrate that, I want you to see what pacemakers used to look like.

So this is a pacemaker from the Medtronic Museum up in Minneapolis. And can anyone guess what that little piece of metal is on the right hand side? What its function is?

Antenna? Control? Control is very close. Any other guesses?

So this is a device before there was wireless communication to control a pacemaker. In the old days, when you want to change the settings on your device, the physician says, "Patient, please lift up your arm. I'm going to put a needle through your armpit to twist the dial to change your heart rate. "

So one of the great reasons for wireless is that it actually reduces infection rates, because the more you put foreign objects in your body, the more likely you are to contract an infection. It is a serious risk. Actually, 1% of implantations have major complications, and of those, about 1% are fatal. So controlling infection is one of the most important things you can do in the implantation and changing of the device.

Of course, if you go the other extreme and just say, I want to put wireless everywhere, you'll get different kinds of risks. So I've sort of dubbed this the bacon theory of wireless. Now my mother's from the Midwest, so she used to say bacon makes everything better.

And I've noticed there are some device manufacturers who seem to be putting wireless everywhere without necessarily thinking through all the risks. It does have its benefits, but you need to very strategically think before you add this to a safety critical device. What are the security risks for instance that are going to be opening up?

Oops, I had one misplaced slide, but I guess I'll just say it anyway. I'm not going to talk a whole lot about networking, but I thought this quote was just too good not to mention. Does anyone remember the ship off the coast of Italy? The captain says, "These days, everything is much safer, thanks to modern instruments and the internet." And there's his ship that turned over there.

So you add internet connectivity and wireless to your medical device, there are going to be new risks. And you don't need to be afraid of them, but you just need to have appropriate mitigating controls.

So I'm flying through this. But what I want to give you is paint a picture of what's a typical day in a medical device, and how it's used in clinical care, and how that might change your mindset if you come from a security background, and how you think about risk.

So first going to talk about the world where there aren't real threats, just unsafe practices and some carelessness. So the FDA maintains a database of near misses, malfunctions, injuries, and deaths. This is all public. You can go look this up yourself. It's called MAUDE.

And one of the devices was called this volumetric infusion pump. This is a device that infuses drugs into your body through an IV mechanically. And this patient died.

And if you look carefully, it says one of the causes was a buffer overflow. I think you learned about buffer overflows in your first lecture. So they are very real and they happen and in every sector.

So in this particular case when the buffer overflow occurred, it was actually caught in their error checking in the software, but the action it took was to shut the pump down. To bring it down to a safe mode. What they didn't realize was that for some patients, shutting down the pump is basically a death sentence. So this patient died after the increase in intracranial pressure, followed by brain death because of the buffer overflow.

So there's nothing really complicated here, right? You all know you don't want to have buffer overflows in your software. There's no adversary at this point. So this kind of illustrates the state of software, at least for this particular device. It's very challenging.

The other challenging part that doesn't come up a whole lot in a security course is the human side. So there are few universities that focus on the human element, but I think there ought to be more. So I set out on some life experience of my own.

My wife asked to remain anonymous, so she said as long as I don't reveal her name. So that's me, that's our infusion pump in the back, and that's our baby in there. And for us luckily the pump worked just fine. But pumps are great for delivering medical care, but they have resulted in over 500 deaths due to various forms of malfunctions.

So I'm going to tell you about one more malfunction. There's also an implantable kind of pump. They're not just bedside pumps, the kind you see on daytime hospital dramas. But here's an implantable pump, and it's got this semipermeable membrane where you can replenish the drugs. And this is a user interface that the nurse or the clinician uses to change the dosage rate.

So does anyone see where you type in the quantity of drug? You've got to kind of squint, right? So you squint really closely.

And one thing you'll notice is here by number six it says we're going to dose this bolus-- bolus is a quantum of drug-- over 20 minutes and 12 seconds. We're going to dose this into the patient. And this is implanted, so you don't feel it. There's no nerve.

And this user interface is actually after an FDA recall went into effect for the software quality. So what was missing before the recall were eight key elements. In particular HH:MM:SS. So what do you think happens, or what you think could happen, if that label were missing?

It's really easy to get the units wrong. Make an order of magnitude error. So unfortunately for this patient, who later expired, he or she had his or her pump reprogrammed, and the health care professional noticed that the bolus was given over 20 minutes instead of 20 hours after the fact. Unfortunately the patient left the facility, got into a motor vehicle accident, and then later died after the family removed life support.

But if you look at this from a technical perspective, the problem is pretty simple, right? In terms of you didn't have the label there. But human factors is very easy to overlook. It's not always right there, front and center, in the engineering process.

Do you have a human factors part in this lecture? See what I mean? Blame Nickolai. No, Nickolai is great.

But it's a very important element of improving the trustworthiness of devices that rely on software. So I encourage you to think about better human elements and human factors for your software, even if it's on something non-critical. So that should begin to paint a picture of the typical problems in medical device failures post [INAUDIBLE] 25.

And the other thing I want to talk about is the exciting world of management. Management, exciting. I used to collect all these little dialogue boxes whenever my computer would get a software update, but this all happens in the background now. Like my iPhone's constantly getting updates and drawing more power. But now it just sort of happens.

But medical devices also take software updates. They're not really fundamentally different from traditional computing devices. They just happen to control vital functions of your body.

So there's an interesting case. It's now been about four years. So McAfee-- there are a number of antivirus companies that produce products that hospitals use-- and in this particular case, McAfee had this software update that actually misclassified a critical Windows DL as malicious, and then decided to quarantine the system. So when it quarantined, let's see.

[COMPUTER SOUND]

That always happens, right? OK. So, ha ha ha.

In this particular case with McAfee, when they quarantined this critical Windows DL as malicious, the machine just started rebooting. Blue Screen of Death and cycling. And in Rhode Island, they basically stopped admitting patients at one hospital, except for severe cases like gunshot wounds, because their admission systems weren't working properly. So clinical care depends heavily on the function of software, and we sometimes forget about the role of security.

On the topic of depending on other people's software, Microsoft has one of the largest footprint of operating systems. And believe it or not, there are a lot of medical devices that run on Windows XP. Windows XP, in case you didn't hear, went out of service half a year ago.

So you should not be using this, because there are no more updates, security updates, function updates. It's antiquated software. But there are still medical devices today being shipped brand new with Windows XP. The software life cycles are a little bit misaligned.

If you're used to downloading updates for your open source software on a daily basis, well, think about medical devices. You might not be able to get to it, say, for a year. It might be in the field for 20 years. So it's very difficult to locate software that's appropriate for a 20-year life cycle. It's basically flying in space.

So the Food and Drug Administration has now released some guidance-- actually, this was just exactly a month ago-- on what they expect to see from manufacturers. Think of it as a design project. As you're writing down all the requirements of your medical device, they're asking manufacturers how have they thought through the security problems.

How have they thought through all the security risks? How are they mitigating it? What risks are they accepting as what they call residual risk, things that they don't solve? But they expect them to least be aware of all the risks and ideally mitigate them.

So with the management of software, when no one person is accountable, all sorts of crazy things happen. But there is some guidance now that's beginning to emerge to help the manufacturing community to better integrate security into their products. So I think we're making some pretty good time. All right.

So now we're going to be able to go into the security side. I wanted to get the non-security stuff out of the way for the context. So let's put on our gray hats and black hats.

Before I begin this, though, I guess what I want to say is this is a very challenging area to do research, because there are patients. And if I were given a medical device, for instance, today, I'd still take it even if the security problems weren't all worked out, because I know I'm much better off with that medical device. But that said, of course, I'd prefer to have medical devices that are more secure. So there is the emergence of more and more secure devices, but today, if you have to choose between a device and no device, I'd strongly recommend taking it, because you're going to be in a much better position.

But that said, let's take a look now. If we consider the adversary, and if the adversary wants to cause problems to a medical device. So who's got the defibrillator at the moment? Oh, it's right over here. Good.

So I'd like to tell you a little bit about how these defibrillators are implanted. This is a very special device because, well, number one, it's implanted, therefore it's very high risk. It's life sustaining. If it's pacing your heart, for instance, and it fails, the results can be catastrophic. So it's very interesting from an engineering perspective. It needs to work 24/7 for many years.

So this is a programmer. Not a person, but a device. It's basically a ruggedized computer, and attached to it is a little wand. That's not a mouse. That's a transmitter/receiver speaking a proprietary wireless signal over a licensed spectrum. We're not talking 802.11, we're talking specially-licensed spectrum there.

And what happens is it takes about 90 minutes. The patient is awake, just slightly sedated to remain calm, and there's a local anesthetic. A small incision is made beneath the clavicle. And then the team-- typically it's a team of about six people-- will weave electrodes through a sacrificed blood vessel that then terminates inside the heart.

And actually I have one of them right here. This was not previously used. You can pass this around. You see the little tines on the end.

And on some of the devices there's both a sensor, so it can sense your cardiac rhythm, and there's also actuation. You can send shocks, both small and large, to either pace the heart or to basically reboot the heart if there's a chaotic rhythm. It's a very highly advanced device. It's a steroid-tipped piece of metal on the end, so it doesn't bind to the tissue. You can pass that around. It's basically a USB cable, right?

So after that's implanted into the body, the patient is sewn up. They do some testing. And typically the patient will receive what looks like a little base station. Like a little access point. It's very proprietary.

Typically they speak a proprietary RF to the implant so it can gather all the telemetry, so that it can send it back up through the cloud-- typically through a private cloud, for whatever private means-- so that the health care professionals can keep tabs on their patient. So for instance, if you notice that there's some odd measurement coming from patient Mary, you might call up Mary and say, "You should really make an appointment and come in, because I'd like to see what's going on with your defibrillator." So one of the nice things about the wireless is they're able to have more continuous care rather than come back in a year.

We had a team of students at several universities get together, and I gave them one of these defibrillators and an oscilloscope, and they went off into a cave for about nine months. And they came back and said, "Look what we found!"

So this is a screenshot of the communication between a device and the programmer. And what you can see is first of all, it's in the clear. There's no cryptography, at least none that we could find. You'll find inside here the name of the implanting physician, the diagnosis, the hospital. Basically a complete electronic health record.

This is an older device, from about 10 years ago. But that was the state of the art about 10 years ago. There didn't appear to be any use of encryption, at least for the privacy of the health information.

So when we noticed this, we thought, well then, we definitely need to look at the security side about how the device is controlled. How do they ensure the authenticity of the control? The integrity? And that's when we decided to do the following experiment.

So we started learning how to use something called a software radio. Probably some of you have played around with these. There are a bunch of them now. About 10 years ago, the most popular one was the USRP and GNU radio software.

So we took an antenna from a pacemaker that we didn't need, created a little antenna, and we recorded the RF communication of inducing a fatal heart rhythm. And then we replayed that communication back. And then the device happily emitted a large-- something on the order of a 500-volt shock. On the order of about 32 joules in one millisecond, which I'm told if you were to get that on you, it's like being kicked in the chest by a horse. So it's a rather powerful shock.

And the interesting thing was how we discovered this. So I was in the operating room, and recall back, I said that when you're a patient and the procedure is ending, the health care team tests if the defibrillator is working properly. So how do you end-to-end test if a defibrillator's working properly if the heart is beating normally? Right?

So what's built into the defibrillator is a command to induce the very fatal heart rhythm that the defibrillator is designed to restore you from. It's called a command shock. So when I asked the physicians about that, they didn't seem to understand the concept of authentication. And that's when we decided we'd really need to look more deeply into how to solve these problems.

So in this particular case, we were able to send the command to the device, and we weren't authenticated, and we could induce that shock. The good news is these devices have been able to solve these problems through some software updates. And they've been aware of it for quite a while, so they're able to spin out devices that now take into account some of these more adversarial conditions.

Where are those tines going around? Over there? OK, great.

So that's the implant side. There's a huge amount of innovation going on with implants. It's not really science fiction anymore, but there are real people and patients behind it. And most people care deeply about delivering quality health care. But sometimes they just don't realize how to fit security into their designing process. So it's a challenge culturally.

Another stakeholder are the people who provide health care in the first place. Hospitals, primarily, or small clinics. If you want to find malware, go to a hospital. You're going to find some interesting malware. And here's why.

So here's a screenshot from a colleague who used to work at Beth Israel Deaconess Medical Center here in Boston. And he gave a map of his network architecture. There's nothing particularly earth-shattering about the architecture. What was interesting, though, was he listed the number of operating systems in his hospital on what were considered medical devices.

And I looked at him-- I like to add up numbers and insanity check things-- and I said, "Well, you've got Service Pack one, two, three of Windows XP, zero 15 plus one. That equals 16. That doesn't equal 600. Your addition's wrong."

And he looked at me and he said, "No, Kevin, that's 600 Service Pack zero machines in the hospital." So these are medical devices where they've been unable to get the manufacturer to provide patches and update it to the modern software. Which means it's that old software, vulnerable to all the old malware that's been hitting Windows XP for 15 years.

So it's very difficult in the clinical setting to keep yourself protected, because the product life cycles are just completely out of sync. They think in terms of decades in health care, but in the fast hockey stick world of Silicon Valley, we think about days, weeks, or months for software updates.

You can see down here in their clinical systems, average time to infection is about 12 days when they don't have any kind of protection against malware. And they can get almost up to a year if they're able to get an antivirus product on there. But even that's not perfect.

And feel free to ask questions too, by the way, if you want to know more. Go deeper dive on any of these incidents. But one of the interesting things I found was that one relatively common source of infection is the vendor themselves. Sometimes they don't even realize it. So I'm going to go over a few cases where the vendor has sort of accidentally been the carrier of the malware.

I was talking with the chief field security officer for the Veterans Administration, the VA. They have about 153 clinics in the United States. And one day there was a vendor showing up to do software updates on some of their clinical medical devices. And her intrusion detection software was just chirping away everywhere-- I think his name was Bob-- everywhere Bob was walking and plugging in his USB drive to update the software. He was infecting the machines with malware by accident, because somehow malware got onto his USB drive.

So there's a perception out there that if you're not networked, you're safe. But if you think about it for moment, very few people used the internet 20 years ago and there were still computer viruses.

So in a hospital, a common infection vector is the USB drive. I'm even aware of two manufacturers-- I can't tell you their names-- but they almost shipped malware-infected medical devices. And they caught it by chance, by luck, before it went out into the product line.

Who's done any work on the programming with the cloud or software distribution? A few of you. So the medical community is also embracing the cloud. It gives them more distributive control.

But it also comes with risks that are qualitatively different from your typical software. If you want to get the newest word processor, that's one thing. But if you want to get an update for your ventilator, completely different.

So I noticed there was a recall on the firmware for a ventilator. And the manufacture sent out a handy dandy website where you could download an update. Now I was going to go check their PGP signatures. Couldn't find those, but what I did find was a little link down here. It says, "Click here for your software update." I thought, oh, goody, let's go do that.

So I did that and up popped this dialogue box. It says, "Warning-- Visiting this site may harm your computer. This website you are visiting appears to contain malware." Has anyone seen this before? Do you know what it was what it is? What's going on?

AUDIENCE: So that's probably your antivirus software, correct?

PROFESSOR: Close. It's not my antivirus software, but it's sort of a similar concept. In the back, I heard.

AUDIENCE: I would bet this is Chrome.

PROFESSOR: Chrome. Yeah, so in this case I believe I was using Chrome. But effectively what's going on is Google has something they call the Safe Web Browsing service.

So actually, the guy who did this is Neil [INAUDIBLE]. He's one of the lead programmers for, I believe, OpenSSH. He's actually from Michigan. But he created this service at Google that goes around the internet just downloading random executables and then running them.

And what's interesting is they create a whole bunch of virtual machines. This is my understanding. I may be misrepresenting it, but my understanding is they create a whole bunch of virtual machines, download those executables, and just run it and then see if the virtual machine gets infected. And if the virtual machine gets infected, you flag that website as distributing malware. They don't know the intentions necessarily, but it's a participant in the malware distribution.

This is what you might call drive-by downloads. It's a very common way of getting malware to you on the internet, especially with the spammers, and some of the organized crime. But in this case their website appears have been infiltrated, and instead of sending me the ventilator software update, they were giving me malware. And at least according to the Google website, it says that over the past 90 days, that's what the website was resulting in.

So all I could think was, all right, so if there's an FDA recall, and you're a biomedical engineer working for a hospital, and your job is to keep your hospital medical devices safe and effective. You're going to go download that software. So which box do you think they clicked? Do you think they clicked close or ignore?

Right? I am sure, I would bet you dollars to donuts, 99% of them clicked ignore. Right? And so all I'm imagining now is we've got thousands of clinical engineers and biomedical engineers walking around with malware on their laptops in hospitals. Hopefully not on the ventilator, but most likely on their local computer.

So other fun things you can do is you can go search the MAUDE database for keywords like computer virus and see what's in there. And these are all narratives submitted by hospitals and manufacturers. One of the more interesting ones is something called a compounder.

So I have one of these in my lab. It's kind of hard to get. But it makes liquid drugs. So it has I think on the order of 16 ports on the top, where you can have the little serums, and then it deposits it into a saline bag. And then you can use IV delivery to deliver it directly to your veins.

So many hospitals will have these for custom, just in time drug delivery, special cocktails of drugs for patients. And what was interesting is here, there was a report that the compounder was infected with a virus. OK?

So we bought that compounder, and we found it runs Windows XP embedded. Surprise. And so it was vulnerable to malware, all the malware that any other Windows XP box would be vulnerable to.

But what was a little bit surprising to me was manufacturer response at the time. I hope they changed their tune, but at the time they said, "Well, we do not regularly install operating system updates or patches." This struck me as whoa, what? What do you mean? I said maybe they had a bit flip.

But there's a huge misunderstanding about expectations of software updates. Let me be clear. FDA expects manufacturers to keep the software up to date. But many manufacturers will claim that they are not able to do updates because of some FDA nonexistent rules. So if you ever run into a medical device manufacturer, and they claim that the FDA rules prevent them from doing software updates, just tell them, no, actually that's untrue.

And Professor Freeman created a poster for this. So here we go. "Homework prevents me from passing class, eHarmony prevents me from getting dates, and yes, FDA rules prevent software updates. Yeah, right. Bull."

So it is true that issuing a software update takes effort. It takes engineering time. It's not a simple process. It's not like-- I don't know what course it's called these days, 6.170, what it's become-- but it's not as simple as typing "make" and then submit to the auto-grader. There's a huge amount of verification and validation that goes on.

But that's what you're expected to do if you're in the medical device manufacturing game. If you're in that industry, that's the expectation.

So a question that often comes up is, do we need to worry about this? And are there any intentional malicious malfunctions? How significant are these? And the good news is, I'm not aware of any specific instance where there's been a targeted attack, and I hope none ever happens. But I think it'd be foolish to assume that bad people don't exist.

So if you look back in history, in 1982, actually, there was an incident in Chicago where somebody deliberately tampered with extra-strength Tylenol on the shelves of pharmacies and inserted cyanide. A number of people ingested it and died. A short time later, at the funeral, additional members of family used the same bottle. They also died.

Within days, the US had pulled Tylenol from all the shelves in the United States. You could not find Tylenol in the United States. And within one year, Congress had passed new legislation requiring tamper-evident packaging and physical security of over-the-counter drugs. This incident is the reason when you open up your medicine, you see a little metal foil.

So we know bad people exist. The cases that we are aware of are more about tomfoolery, but still dangerous. So this woman said she had one of the worst seizure she's ever experienced when somebody decided to post flashing animations on an epilepsy support group website. So quite malicious. It was probably someone who didn't realize the ramification of their actions, because you can actually severely harm a patient who's sensitive to those kinds of things. But again, bad people do exist.

So one of the problems with the culture gap is that much of medical device manufacturing thinks statistically, and they think about past performance of a device predicting future performance. So in the security world, we know that actually, if you see no security problems, that might be because there are a bunch more to come soon.

So if you take a look at the Mac, for instance, right? Before two years ago, basically no malware was on the Mac. But then one night over half a million Macs got infected by Flashback.

So one of the problems is bridging that culture gap. To move from, well, there haven't been any reported problems yet, so we don't need to worry about it, to explaining more about how to fit security into the risk management thinking of medical device manufacturing. So hopefully we can avoid this, and keep that to be on the *Weekly World News,* but it could happen.

So trying to bring that analogy home now. Before we get into a little bit more on the solutions here, is that way back when, there was a lot of denial that hand washing was a problem. But there was a real reason for that. In the 1800s, running water was not exactly common in hospitals. Latex gloves did not exist yet. So to ask someone to merely wash their hands for each procedure was actually a pretty tall order.

And the same thing can be said of security today, in almost any context. There's no magic pixie dust you can sprinkle. There are no magic latex gloves you can put to somehow magically add security. So when you ask a manufacturer or clinician to, say, keep your device secure, it's a pretty tall order. So it's going to take some time, I think.

But if they were alive today, they might be saying medical devices should be secure, and doctors are gentleman and therefore their computers are secure. But I'm optimistic we're going to get there, because most manufacturers I talk to now realize it's a real problem. They're just not necessarily sure on what to do next. So maybe they'll be hiring you people for the future, to help them solve these security problems.

But what it all boils down to is it's very difficult to add security on after the fact. Bolting it on is very challenging. It's possible in some cases, but it's really hard, and often very expensive.

And you've really got to design it in from the beginning to get it right. So FDA is expecting manufacturers to get it right when they're still working with pen and paper, on whiteboards, before they've actually manufactured the medical device.

So how are we doing on time? Oh, quite a bit? 40 minutes, awesome. OK. I'm going faster than I thought. Sorry if you're taking notes. I'll talk slower now.

I want to talk a little bit about technology to make a medical devices actually more trustworthy. So I'm going to try to blow your mind, all right? So why do you trust the sensor on, let's say, your smartphone? You've got a smartphone there. Do you know what sensors are on that smartphone?

AUDIENCE: GPS.

PROFESSOR: There's GPS? Accelerometer, I heard. Any other thoughts? What else would we find on a phone?

AUDIENCE: Compass.

PROFESSOR: Compass? Light?

AUDIENCE: [INAUDIBLE].

PROFESSOR: Electromagnetic field? Everything's temperature-sensitive. Camera's technically got a CCD sensor. So there's sensors all over the place. Medical devices have sensors, too.

Now, why do you trust what the sensor's telling your processor? If you write software and your sensor tells you it's 77 degrees today, or 25 Celsius, why do you believe that? So at least in my lab, we do a lot of work on sensors. So I try to pass this one around.

This is a batteryless sensor. It's got an MSP430 microcontroller. But there's no battery. It actually runs off a 10 microfarad capacitor, and it harvests RF energy to power up that microprocessor. I'll pass it up this side, I guess.

And it's got all the fun little things like a 3D accelerometer, temperature sensors, light, all that fun stuff. But it's really hard to power up.

But again, how do you trust what's actually coming into that sensor? Something's translating it from all these physical phenomena to little electrical pulses. So one thing I want to highlight is why you might not want to trust what's coming out of that sensor.

So this is work from one of my post-docs, Denis Foo Kune here, who's kiteboarding on Lake Michigan. But in his other spare time, he likes to interfere with sensors. So let me tell you about-- forget security for a moment, to safety-- there was a gentleman in 2009 who reported that every time his cell phone rang in his kitchen, his oven turned on. So you can go find this in the *New York Times*. It just happened to be that that resonant frequency was just perfect to get that ignition to go off in the over.

So there's interference all over the place. It's a constant battle, because we have different devices speaking in the same spectrum. But there are technologies to reduce that interference.

The problem is, what happens when the interference is in the baseband? I'm going to go a little bit analog on you for moment. So does 6.003 still exist? It does? OK, good. So I encourage you all to take it if you haven't. It's one of the most awesome courses for a CS person, because you don't have to go too deep into the circus.

So what was interesting to me was, I was trying to understand why I should believe what a sensor's telling me. And so I started to look at the block diagram. And so for instance, if you've got a Bluetooth headset, what you're going to find inside that Bluetooth headset is a microphone, piece of wire, an amplifier-- right, 003-- some more wire, or some traces on a PCB. It goes to an analog/digital converter. There might be some filtering. And then it goes to your microprocessor.

But there's all this other stuff that gets in the way before it gets to your software. And for some reason, your software just believes anything this wire says. So what was interesting to me was, well, you know what? That piece of wire from the microphone to the amplifier, it has a length. It also has a resonant frequency.

So what would happen if somebody generates custom electromagnet interference that's optimized to latch onto that resonant frequency of that piece of wire? Well, it would go into the amplifier and it would get amplified. And then it would go into that analog/digital converter, and you'd pass onto the microprocessor.

One of the questions we had was, was this possible at all? And if so, how hard would it be? What kind of power would you need to do it? And what would be the quality of the signal that actually reaches the microprocessor?

So the fundamental reason why this is even possible is because we're talking about intentional, as opposed to accidental interference, we're throwing it into the baseband. So here's an example. Imagine that your medical device is designed to accept physiologic signals in the low hertz. Like your heart doesn't beat that fast. We're talking a few hertz or less.

So if your electrodes were to pick up some high frequency signals, you'd just put in some analog filters. You'd say, that cannot be real, right? If your heart's beating that fast, you're probably just picking up something like an electric mixer while you're making your lunch. So similarly you can filter out pulses in the high frequency.

But if you send interference that's in the baseband, those filters are going to be meaningless. Because those analog filters cannot get rid of if it's in the same frequency area as what you're expecting. So it's hard to filter in the analog.

So I'm going to go through a couple examples. We're going to start with a Bluetooth headset, and then work our way up to a medical device. So Denis, he built a bunch of homebrew dipole antennas and transmitters and amplifiers.

Now what he's got up here is you can see he's got a webcam. I guess not too many of us need to buy these anymore, because they're built in. But that webcam has a microphone, and then it's got a little USB cable to deliver the audio to the computer. So what he's done is he's set up the computer to record the video and audio and then play it back.

So what's interesting is-- you'll see this now. He was in a completely silent room. It sort of sounded like this. All you could hear was the ventilation system.

He's got the camera. He removed the housing, just so it's easier to tap in and measure the interference. And then he's got a software radio about a meter away, generating custom electromagnetic interference. He writes it in Python, and then sends over his signals.

So here's what the computer on the left thought it heard, even in this silent room.

[AUDIO PLAYBACK]

[MUSIC - WEEZER, "ISLAND IN THE SUN"]

[END PLAYBACK]

PROFESSOR: So yeah. The last time I did that, somebody in the back actually started dancing.

So it's actually relatively high fidelity. And it actually turns out that in the manufacturing community, they're so cheap. They use really cheap microphones with poor frequency responses. So we actually got higher quality audio through interference than going to the microphone.

So if you ever don't like your Bluetooth headset and you want to play classical music, just do it with interference. But don't tell the FCC I told you to do that, because you're not supposed to.

But the point is if you're talking intentional magnetic interference, it's kind of outside the security model. And so your processor just trusts it.

So some interesting things you can do. Let's say your office mate decides to call up his bank to make some deposits. Well, you can insert DTMF tones. That's kind of fun. So we were just playing around. You can change the language as the person's trying to make deposits from account to account.

But there's all just interference. And actually the person on the Bluetooth headset didn't hear it. Because remember it's coming from the person, so that it doesn't actually get echoed back to them. But the bank heard it and made all the transactions.

So there are ways to do this. It doesn't take a whole bunch of analog skills. We're mostly computer scientists. But you do need to somehow convert the signal you want to have appear at the microprocessor into something else that's easier to transmit.

So the first thing you can do is think about just overwhelming the thing with a very strong signal. That's the brute force approach. It doesn't work so well, but it works a little bit. So if you send something out that matches the resonant frequency of that little piece of wire, yeah, that'll get the job done to some extent.

The problem is a lot of these signals are low frequency, and it's more difficult to transmit. It's got less power, basically. So it's going to be harder to send the signal.

So what you really want to do is send a higher frequency signal, and it's going to be easier to deliver the power. But if you send a really high frequency signal, that's going to be outside the baseband, so all the filters are going to go at it.

So here's what you do instead. You treat this circuit as an unintentional demodulator. So what you do is, we had that original sine wave we wanted to transmit.

Instead we modulate it onto a higher frequency sine wave. And we send it in to the amplifier, and eventually it's going to work its way in because of sampling theory. You can think about Nyquist and all that.

So up on the top is the interfering signal we're actually sending, and then on the bottom is what the microprocessor sees. Because remember the analog-to-digital converter is not continuously sampling. There's an interrupt on the processor. Wake up, take a reading, wake up, take a reading. So it's actually going to sample, and then try to infer the signal.

So as we're sending out our really fast signal, it takes a sample, it takes a sample, it takes a sample, et cetera, et cetera. Your microprocessor thinks it got this nice low frequency sine wave, but we actually used a high frequency one, because that allowed us to transmit more easily.

So I'm not going to go through all the nitty-gritties, but one another kind of cool way to do this is to muck around with the non-linear components of the circuit. But this is all about violating security models, right? So we're completely violating what the circuit designer had intended. It turns out that if you send in, say, in this case you're sending in 826 megahertz is the resonant frequency of our wire. But I can't speak that fast.

So what we do is we modulate our voice on an 826 megahertz carrier. Problem is it's going to get, for instance, all this replication of the signal. You're going to see the frequency. Here we're looking at frequency domain. It gets repeated.

But it turns out because of the filters built into most of these devices, it's actually going to chop off the repeated copies. So the end of the day, what the microprocessor sees is our original 1 kilohertz signal we were trying to send in. It's been unintentionally demodulated.

So that's the easiest example that I've been able to come up with to explain the idea of this intentional interference. And now we're going to try to apply it to defibrillators and medical devices.

So again, the defibrillator's implanted into the clavicle. And it has these electrodes-- you can kind of see them here-- that go into the chambers of the heart, and it's used for both sensing and actuation. So it's just a signal. So this is the time domain, and this is the Fourier transform, effectively.

So this is a single heartbeat, and the heartbeat is actually quite intricate. The physicians have actually labeled the different components of the heart rate. You've got the QRS complex, which is typically what you would think of as the heartbeat. The actual beat is this giant R here. That's the one you'll feel. But there are also these other smaller waves, as your tissue is energizing and relaxing.

So if do a Fourier transform on your cardiac rhythm, you're going to end up with most of the signal in the tens of hertz. You're not going to see things a whole lot beyond 100 hertz in a typical cardiac signal. So most of these devices are designed to filter out things that are really low frequency or really high frequency.

But if you choose to insert intentional electromagnetic interference on the baseband, then it gets through all the analog circuit filters. And now the only approach to [INAUDIBLE] that would be things more on the computer science side.

So this is where my students began to have a little bit of fun. So we wanted to test this in as realistic a situation as we could. We couldn't get volunteers.

So instead we discovered there's actually a national standard. This is a body. This is you. It turns out that we're all just bags of saline solution. And so if you have a highly calibrated saline solution, that's the best way to simulate human tissue.

The other thing we've done is we used the synthetic cadaver. She's actually anatomically correct. She's got all the same vital organs as anyone else would have inside, and a working circulatory system. So it has all the surface properties of the RF.

So here we're doing radiation fluoroscopy to do 3D imaging-- 4D. We see light imaging as we're implanting the electrodes into our synthetic cadaver.

So what we're going to do now is generate some electromagnetic interference and then try to see what the device is perceiving as a trustworthy signal. So a couple ways we did this. In the saline solution, we used just a spool of magnet wire. Here we have the wand that's reading out the telemetry to see what the device thinks it's seeing, and then another experimental case.

So I had some leftover pipes from plumbing, so we created a dipole antenna. And on the back there on that poster board, we created a 2D version of a patient. You can see that's the curvature of the electrode, that's the electrode, and then the pacemaker is right underneath the tape. And we're transferring to it.

So here's what the device thought it saw, even though it wasn't happening. So keep in mind this should have been a flat line, because there is no patient. There is no heart beating.

So we tried a couple different signals of interest are we pulsed a sinusoid. So that's really a sine wave, but it's so fast you can't quite tell. But we pulsed it like a heart beat. So every one second we sent out a pulse.

And then we also did one that's modulated. That's a little bit noisier.

So this is a screenshot of the pacemaker programmer which tells us live what telemetry is going out. And it's hard to read, but the little green up there, VP, says that the device sent out the ventricular pace. This is the pacemaker sending an artificial heartbeat, basically, to make the tissue contract.

What's interesting is when we started sending our interference, it got what's called a VS, a ventricular sense. The little purple VS, there's three of them. So the pacemaker thought that the heart was beating on its own, so it chose to inhibit the pacing to save power. And then when we turned off the interference, the pacing began again.

Similarly over here you see where the interference starts, and it's sensing ventricular sense. It says, oh, the body's pacing itself naturally. I don't need to waste my energy pacing the heart. So we're able to induce that interference, and then trick the microprocessor into believing the long state.

There is a silver lining, though. The good news is, that only works in vitro. Whenever we would do this in saline solution or in anything that approximated the body, it basically didn't work.

And that's because your body absorbs a lot of that RF energy, and it doesn't actually get to the sensor. So the closest we were able to get this to work was, with the saline, like three centimeters. So that basically means there's no worry for this particular kind of interference from an implant.

However, an externally worn device, we don't know. We hadn't done any tests on insulin pumps yet. There are plenty of different kinds.

There's glucose sensors, for instance, that are percutaneous. I wouldn't be surprised if someone here has one. They're pretty common. But we just don't know yet.

But one of the approaches we're taking to solve this follows the end-to-end principle to some extent. A lot of these, I just don't think the analog is able to distinguish good from bad signal. And so you have to do it closer to the application layer.

So one of the defenses that we tried out was the following. It has its own limitations, but here's the basic idea. So imagine you're a pacemaker, and you want to know, are you getting a trustworthy signal?

So what you do is you selectively choose to send test pulses every now and then, to basically keep the adversary in check. So here's the interesting thing we discovered when we worked with electrophysiologists. We learned that if you send a pacing pulse to a heart that recently was beating, within about 200 milliseconds, that cardiac tissue is physically incapable of beating again. It's also physically incapable of sending out an electrical response, just because of the polarization-- the way that cardiac tissue works.

So we said, so what would happen if we send an extra pacing pulse right after a ventricular sense? He said, oh, well, if the heart actually had beat, as your sensor told you, then you should get no response. Because it would be incapable of sending a response.

So therefore, if we saw the heart send us an electrical signal back, we knew-- then that proves to us that we were fooled on the previous heartbeat. And there we raise our warning signs that it appears to be we're getting intentional electromagnetic interference.

So the basic idea is, again, we probe it, and we make use of some of what we know about the physiology of the body to have better trustworthiness. Another approach we didn't look into too deeply was looking at propagation delay.

Because if you have electromagnetic interference coming at you, it's basically light, right? Speed of light. And if it's hitting you all at once, if you have two sensors, and you simultaneously see the same cardiac signal at the same time, something's wrong. Because there's an electrochemical propagation delay from your vagus nerve as the electrical signal is traveling down through your heart.

So there are other ways to try to tease out whether the physiologic signal is trustworthy, but this is new ground. There's not a lot going on in this space yet. A lot of fun projects for graduate and undergraduate research.

We end at-- oh, 25? Oh. So I want to tell you about another project, and that is detecting malware at power outlets.

So a few years ago, one of my students, Shane, he said, hey, I built this power outlet, and I can tell what website you're browsing. So he put a little sense resistor in here, and he measures what's called the phase shift in the reactive power. It's basically a proxy for the load on your computer.

And he can basically tell how your computer-- how your processor is changing the load as it's going out onto the AC power system. This is not new. Has anyone heard of Tempest? Tempest protection? A few of you.

So Tempest has been around for years. Basically signals leak all over the place, and so there's a whole fine art to stopping signals from leaking. What was interesting to me was, I like to keep all my old computers-- actually I have an exokernel machine. And it's an old-- I think it's a Pentium 4.

And this was before there was advanced power management. So if you measured the power coming out of this old Pentium, it was constant. Just doesn't matter if you were doing anything. If you have a spin while loop, doesn't matter. It's the same thing as actually doing processing.

But if you buy a modern computer, whether it be desktop or phone, your workload is being revealed over the power line in subtle ways. And so what he discovered was that what's going on here. If you have an embedded system that's very difficult to change, and you want to retrofit security onto it, what you can do is put in basically a power strip. An intelligent power strip.

And it uses machine learning classification of the frequency domain. Actually looking at the frequency components of your power consumption. It's not looking at how much power you consume. Instead it's looking at how often do you consume it.

So let me give you some intuition here. So imagine you have a medical device that gets infected by malware. Let's say this malware is going to wake up every few minutes to send out spam. How might that change the power consumption? Yeah.

AUDIENCE: [INAUDIBLE].

PROFESSOR: Yeah, every few minutes that interrupt's going to go off, and the processor's going to wake up. It's probably going to power up its memory. It's going to do some cycling, or it might actually insert a few extra cycles in what used to be a very constant set of instructions.

Medical devices generally do a small set of things, as opposed to a general purpose computer. So it's a very regular pattern. So when you suddenly have malware getting in, it just changes the behavior of its power consumption patterns.

So you can pick that up. You do a Fourier transform. You do some other magic involving machine learning. The devil's in the details. But you can basically use the machine learning to identify with very high precision, very high accuracy, low false positive, low false negative, the presence of malware and other anomalies.

And so that's a project he had been working on for a number of years. He initially created this project, though, to identify what website you were browsing. And unfortunately, he submitted it to a bunch of conferences, and they all said, well, why would you ever want to do that?

But it's kind of interesting, because he picked the top 50 Alexa websites. And then he profiled his computer, used that as a training set for the machine learning, and then again, with very high accuracy, very high precision, was able to identify which website you were going to. And we were really confused why it worked at all. And we still don't know exactly why, but we have some strong hunches. And [INAUDIBLE] Drupal.

So there's been a movement over the last 10 years on websites to move from-- who still writes in Emacs HTML? All right, me too. That's why I have all these mistakes on my website. But there has been a large movement, especially in institutions, to have code automatically generate a web content file that follows a very regular structure.

So can you imagine if you go to CNN.com, and they always have an ad in the upper right-hand corner, with flash animation that lasts exactly 22 seconds? So your GPU might kick in at a very regular rate. So some very interesting things bleed through into the power consumption patterns from the web browser, and from other things in your operating system, as a result of your activity.

The only website we couldn't classify too well was GoDaddy. We still don't know why, but we don't care.

So this is going to branch a little bit further out from the security side. But one of the things you find is that when you're helping your colleagues in the hospital system, they often ask back for some of your help. And one of the interesting projects that we got involved with, purely by accident, from some of the pacemaker security work was in some humanitarian aid in developing countries, especially Ghana, to give patients new life. Literally new life, because it turns out if you don't have a health care system, it's very difficult to, say, get a $40,000 pacemaker plus the surgical team. Very challenging.

So what they've been doing is they've been recovering discarded pacemakers and defibrillators, and then sterilizing them. It's actually pretty interesting. You have to use-- well, you don't have to-- but what's typically used is ethylene oxide. It's a gas chamber to sterilize and remove all the pyrogens, things that cause fever.

But these devices are sterilized, and then reimplanted in patients. So here's a gentleman. I believe he was suffering from a slow heart rate, which was basically a death sentence for him. But because he was able to get a pacemaker, it gave him extra years of life.

So the problem they came to us with was, how do they know if the devices are still safe? They weren't even used. So obviously you can look at the battery life. So that's one thing you do. And if the battery is too low, you would not reimplant it, because that wouldn't last too long.

But then what about some of the other things? Has some of the metal corroded? How do we do an end-to-end check to see if you can still detect arrhythmias properly?

So the students in my lab created a special tester that sends out what you would see from the electrical components of cardiac arrhythmias. Things other than sinusoid. Cardiac rhythms that you wouldn't want to have, right? So, anomalies.

And it replays these against the pacemaker leads. The pacemaker thinks it's connected to the patient, and then it responds. And so we check that response to see if it's actually diagnosing the cardiac arrhythmias, and whether it's actually sending out the lifesaving shocks properly.

So they're now starting to test this through the whole FDA process to get their blessing. And that's a work in progress. But it's called the My Heart Your Heart program. You can go look it up if you're curious about it.

And then we also interact quite a bit with the medical device manufacturing community. We bring them in each summer out to Ann Arbor, and we have the manufacturers sit down, while some of the persons who are in charge of running hospitals sit down next to them, and they start sharing their gripes and problems with medical devices.

We had one company come in and just reveal all the problems that none of the people would respond to at the medical device manufacturer. And one guy in the corner was like, that's my team. And so they decided to go out for lunch, have a beer, and just work out the problems. So a lot of it is cultural.

So I don't know if anyone here has done any security analysis work, or reverse engineering. Anyone here? Couple people. So it's really delicate. It's almost an art, because you're dealing with the social elements of the manufacturing side.

And it's even more so in medical device manufacturing, because lives are at stake. And so it can be very, very tricky to share these kinds of problems with the people who are most able to fix it. So it often results in in-person meetings and actually going to their facilities.

So I want to save some more time here. Hopefully we'll have some questions, because I think we have five or 10 minutes. But I want to dispel a couple of myths.

You'll hear a lot of newspaper headlines and TV shows talking about hackers breaking into medical devices. Let me say it is a problem, but it's not the problem. It's not the only problem, and it's probably not the most significant problem.

And it's hard to say that, especially when you enjoy doing security analysis. It's hard to say that, because there's actually two problems that I think are more important. One is preventing wide-scale unavailability of patient care.

Because forget adversaries-- what if you just have malware that accidentally breaks into a medical device in a monoculture where they're all running the same operating system? What happens when you lose 50,000 infusion pumps all at once? It's very difficult to deliver patient care.

One of my colleagues wrote to me saying that his cath labs were shut down. Catheterization lab is a relatively new specialization. It's a special kind of operating room for minimally invasive surgery.

And at his hospital, they had to shut down the cath lab, because turned out a nurse had accidentally brought in a USB stick. Something about transferring family photos up to Yahoo. And somehow malware had gotten in and infected their cath labs. So they had to shut the thing down.

So if you're waiting to get an angioplasty, that particular center's not available to you. You'll have to use one of the backup centers. So availability is, I think, one of the key things that is often forgotten about in security.

Second one is the integrity of the sensor. So if your medical device gets infected by malware, or any kind of malicious software, things are going to change. Things are going to change in a way that the designers didn't anticipate.

So a very simple example. Let's say some malware gets in, and adds a timer to every now and then wake up, send some network packets, and send out some spam. This took some time.

Well, what happens if your medical device assumed that it had complete control over the interrupt handler, and suddenly now it's missing interrupts? Maybe the sensor has some data to supply to the medical device, but because of the malware, it missed the interrupt. You may actually start misdiagnosing patients now, because that device is going to be getting bad data. So I'm very concerned about the integrity of the medical sensors.

There was actually a reported case of a high-risk pregnancy monitor getting infected with malware and giving out incorrect readings. The good news is a highly-trained clinician can look at the device and say, that makes no sense. That's not a sane number coming out of my medical device. But we're basically cutting down the safety margins when we can't have the integrity of our medical sensors.

As I mentioned, very difficult to bolt on this stuff after the fact. You think changing software's hard on an internet scale? Try it on a medical device. So I met a guy from one hospital where his MRI is still running on Windows 95. I have a pacemaker programmer that runs on OS/2. And they recently upgraded to Windows XP.

So they have some really old stuff out there, so changing things for security after the fact is going to be difficult. Not impossible, but difficult. And the other reason is the interruption of clinical workflow.

If you ever go off and want to start implementing medical devices or doing something security related for health care, I encourage you to go off and call up some people, and say, hey, can I go into your operating room? That's what we did. Because you'll see some weird things happen.

I took all my students to live surgery, pediatric surgery. And as they were watching the surgery, they were watching one clinician checking Gmail on one of the medical devices. And they're like, oh OK, so, drive-by downloads, check. At the same time, they wanted to calm the patient, so they logged into Pandora to play music.

Actually I was just at my dentist the other day, and she was playing Pandora. And these ads for various beers started coming up on the screen as she was looking at my dental x-rays. And I was trying to figure why Dos Equis was on my-- I was like, did I drink that much? She's like, no, we just play Pandora here. It's the same web browser, and just click here.

So there's a lot of mixing going on. Maybe it's not malicious, but it's opening cracks. It's out of sight, out of mind.

The hand washing sterile technique is driven into the mindset of anyone who's a clinician. Wash your hands. Don't touch the gloves after you put them on.

But when it comes to security hygiene, it's really out of sight, out of mind. It's not part of the culture yet. They don't even realize they should be asking these questions. Should I be running Pandora on the same device that's controlling my x-rays?

So but the important thing is on the designer side is not to interrupt the clinical workflow. Because that's when mistakes happen. You want to keep the clinical workflow regular, predictable, easy for them to make decisions. And if you add a new dialogue box to enter a password, what do you think a problem could be in the operating room if you ask the clinician to enter a password, say, every ten minutes?

Distractions? You're sitting there, doing this, right? Scalpel. Oh, yeah, let me walk over here and type in my pass-- oh, no, I got my gloves on. Let me take those off. Oh, I've got to resterilize now. Nurse!

So if you're a security engineer, you have to take into account all the rather special conditions of the clinical setting with infection control. Which, surprisingly, not everybody knows about. There are definitely some very talented engineers who know about it, but not enough.

The other big problem is I've noticed that security people tend to specialize in the mechanisms to control security. You can wield crypto. I know CBC mode this, and I know public key crypto this. That's great. And you know how to prevent the problems. You know how to detect the problems.

The issue is from the medical world. Most people in the medical world are coming from a very different mindset, one that's called risk management. Let me try to explain it. In risk management, you look at risks and benefits, and you ask yourself, do they balance? If I take an action, does that improve my risk management outlook?

So if you're going to decide, for instance, am I going to deploy a password system on all my medical devices. A security person might say, duh, of course you're going to deploy passwords, because you need to authenticate.

The safety person might say, well, wait a minute. If I required we have passwords on every system, we're going to worry about sterilization. How do we know how often to time out? And what about emergency access? What if we forget the password? We want to make sure we can get a response time in 30 seconds.

So they might actually make a different decision. They might actually decide not to have passwords at all. Actually many hospitals don't have passwords. Excuse me, many hospitals don't have access control on medical records.

Instead they have what's called audit-based access control. After the fact, if you look at something you shouldn't look at, they come get you. Because they know that it's very difficult to predict what you're going to need in your routine of your clinical workflow.

So the risk management kind of way will depend upon deploying the security controls and all the technology learn about. But in the risk management picture, you might actually decide not to deploy something, because it could cause harm somewhere else. But trustworthy medical device software is going to require both.

So I'll just finish up here. I think there's a lot of interesting things to do. So you're taking this cool security course. I encourage you to go out and use those tools.

But as you're thinking about where to go afterwards, whether it's industry or graduate school, think about medical devices, because they need your help. They need a lot of smart people there. And so there's just one thing missing-- you are. And I think there's a lot of interesting stuff still to be done.

So I think we have five or 10 minutes or so. I'd be glad to take some questions. Or I could go more into depth. I got some fun videos I could show. But I think I'll at least take a break for a moment to see if you have any questions. Yes?

AUDIENCE: So that pacemaker or whatever it was that you were passing around, does that [INAUDIBLE].

PROFESSOR: Oh, the defibrillator.

AUDIENCE: Yeah. How does that interact with the fact that they are [INAUDIBLE] these kind of things.

PROFESSOR: OK so a couple things. So there are defibrillators and there are pacemakers. They're very related. This is a defibrillator. It sends out large shocks. Pacemakers send out small shocks.

But in the US, it's illegal to reimplant these. So it doesn't matter if you can. It's just illegal. But in many developing countries it's not illegal.

And if you look from-- let me back up a slide. If you look from not the control mechanism, but the risk management side of equation, it might actually lead to better public health outcomes to allow reimplantation and reuse in developing countries where they have no other choice. And this is not my project. This is just a project we're assisting on.

But in that particular case, the patients really have no choice. It's basically a death sentence. To sterilize it is pretty tricky.

There's a whole lot of science and engineering that goes into how to properly sterilize to get rid of the pathogens. Because these were in blood, so first abrasive cleaner, but the ethylene oxide is one way to destroy most, if not all, of the pathogens. There's a whole testing procedure. You actually put special little-- I forgot what they're called. They're little wafers-- with known quantities of pathogens. And you put it in alongside some of the devices as it's going into the chamber, and when you pull it out, you test to see if all those organisms have been killed.

Did that answer all your questions? You had a follow up. OK. Yes?

AUDIENCE: So what you're saying, integrity of sensors is a bigger risk for hacker attack, because most of the examples of sensory interference you showed are intentional interference. So it's kind of [INAUDIBLE].

PROFESSOR: Oh, so the question is why focus on integrity of sensors rather than hacking, because everything I showed was about hacking? That's selection bias. I selected those cases, but that doesn't mean that's statistically relevant. I divided up into two cases-- maybe three. The past, the present, and the future.

So at the present, most of the problems we're seeing from malware in our very rudimentary surveillance of medical devices has to do with malware that accidentally gets in, and then causes near misses and malfunctions. But we're no dummies. We know that there could be an intentional adversary in the future. They just haven't materialized yet.

The closest example would be-- this is just from the news reports. I don't know if it's true, but it was from I believe the *New York Times*-- that there was a hospital. CHS, I think, was the hospital, Where they brought in a security company, Mandiant.

And they believe that a nation state had actually come in to steal the medical records. They don't know exactly why, but nation states. And nation states are powerful adversaries, right? If you run up against a nation state, you might as well just give up, because none of these controls are going to help you.

But here's my concern. If the nation state, for instance, is very dedicated on getting one piece of information, what if they make-- they're human too, right? What if they make a mistake along the way and accidentally hit a medical device as they're trying to extract whatever kind of information they're trying to get at? And that could affect the integrity.

In the future, there could be instances of custom malware, but I think it takes that one more step of someone really wanting to cause harm. And I'm hoping that there aren't too many of those kinds of people. But there are people who write malware who don't realize that malware gets into medical devices in hospitals, and it's still causing problems.