Your question is a very healthy reminder that a robust privacy program and an assurance program that supports that cannot stop at the boundaries of the technology system.

MS. GARRISON: With that, we're concluding this panel.

Please be back at 3:15 for panel four, and I would like to thank very much each and every panelist here this afternoon for their contribution to this discussion.

Thank you.

(Applause.)

(A brief recess was taken.)

PANEL 4: Designing Technologies to Protect Consumer Information

MR. SILVER: Welcome back, everyone, to this session, which is not only the final panel of today but the final panel of this pair of workshops which began in May.

This panel will consider how to design technologies to protect consumer information.

Are the microphones working? All right.

And to that end, we've gathered an impressive group of engineers and policy experts.

First, we have Edward Felten from Princeton

University, Alan Paller from The SANS Institute, Richard Purcell from the Corporate Privacy Group. Howard Schmidt is with eBay. Toby Levin will be helping me moderate. Ari Schwartz is back for more from the Center for Democracy and Technology.

Tony Stanco is with George Washington University. We've got Vic Winkler from Sun Microsystems, Kathy Bohrer from IBM Research, and Peter Neumann from SRI International.

I will begin with Peter by asking him to define the problem that we're facing in this area of technologies and designing them to better protect consumer information.

MR. NEUMANN: Thank you.

I would begin by saying that I am a technologist in my 50th year in this field, so I've been around a long time. I'm also an anti-technologist in the sense that I am very concerned about the misuses of technology. I will draw on both facets of my life in what I have to say very briefly.

I go back to Multitex, which was probably the most secure commercially available system ever produced, from 1965 to a couple of years ago, when it was finally decommissioned. In 1972, we did the first very reliable fly-by-wire system for NASA.

So I've been heavily involved in really high-tech technology.

On the other hand, I think we seriously tend to over-endow technological solutions, and I'd like to follow up a little bit on that.

If you think about the repeated statement about defense-in-depth, what we really have is weakness in depth, and I'd like to point out that we have flawed requirements to begin with.

We have flawed evaluation procedures.

We have flawed systems, including legacy systems and systems that require hundreds of patches.

We have flawed administrative procedures.

We have a tremendous burden that we're putting on systems administrators for the very simple reason that those systems are so difficult to maintain.

In fact, the U.S. Government is now widely out-sourcing system administration, as well as software re-deployment.

If you remember the Y2K problem for the air traffic control system, the entire upgrading of the system was out-sourced to the People's Republic of China, unbeknownst to the technical people at the FAA. This is a very strange example of out-sourcing.

We have flawed procurement processes where the

government folks, in particular, are severely constrained by the procurement processes.

We have the risks of un-trusted outsiders and trusted insiders who are not trustworthy because of the fact that the systems themselves are not adequately secure, and we have an enormous lack of accountability.

We talk here about privacy problems and security problems.

The identity theft problem is one that typically comes to mind, where the average individual doesn't think that they have anything to hide, and yet they are vulnerable to identity theft.

But I would like to give you an example of one prototypical or paradigmatic example of a system that requires privacy, security, integrity, and accountability, and a lot of other things -- prevention of denial of service and so on -- and that is the electronic voting problem.

In all of the electronic voting systems produced by the major vendors who are, in fact, providing something like 70 percent of all of the voting machines in the country, there is absolutely zero accountability that your vote goes in correctly and that it's counted correctly.

This is an appalling situation. The fact that

we're trying to make your votes private and provide some sort of assurance to you that nobody can figure out how you voted has resulted in systems in which the integrity and accountability and security issues have been essentially completely ignored.

The Federal Election Commission standards are lame. They're inadequate. They're fundamentally flawed. The evaluation procedures are almost non-existent. There are certification procedures, but they're based on flawed standards in the first place. The result is that we have systems that effectively have no assurance that they're going to do the right thing.

So, I think the confluence of security and privacy and accountability and availability and survivable systems that don't fall apart all by themselves without attacks suggests that there is a problem where we have, in a fundamental way, fallen short of what is needed.

Counter to the very rosy glasses picture that we heard in the previous panel, I wanted to throw out this contrary view that there are some systems that are fundamentally flawed. If we look at, say, the critical infrastructure protection problem, where we see that all of the critical infrastructures are dependent on telecommunications, on computers, on power, and in many

cases on the Internet, which may surprise some of you, and the fact that all of this is completely interrelated, and the fact this was pointed out long ago by the Marsh Commission in '97, it suggests that we are not progressing as fast as we should.

Now, the standard free enterprise version is, oh, the marketplace will solve all these problems. I claim that the marketplace is not solving the problems that I have been working on for the past half-century, namely very survivable, very secure, very reliable systems.

They're certainly good at producing lots of features and whiz-bang Power Point systems and things of that nature, but I think from the point of view of what can be done to make these systems robust, the marketplace is simply not driving it.

Now, you might say, well, gee, there's the open source world. Perhaps if we made the voting machines open source, it would solve the problems. Of course, they're all proprietary. The vendors say that if anybody could ever look at the code, it would decrease the security of the system, therefore nobody is ever going to look at the code.

I happen to have looked at the code for one of these systems for New York City over a decade ago, and my

conclusion was, even if this code was perfect, here are a couple of dozen ways in which the election could be rigged using this system.

So, I think the fallacy there is that, gee, if only we could look at the code, it would solve the problem. It doesn't solve the problem, and there are many examples.

For those of you who are techies, you remember the Ken Thompson Trojan horse that gets installed in the system with absolutely no evidence of anything in the source code. It happens to be an object code modification to a compiler so that the next time your source code is compiled, this Trojan horse is planted in your system.

The bottom line here is that we're dealing with end-to-end holistic problems, whether it's privacy or security or reliability or safety or whatever, and the weak link phenomenon is really one in which we are dealing with weakness in depth.

Frank mentioned snake oil in the previous session. We have a lot of smoke and mirrors, placebos, bait and switch, shell games, and certainly in the electronic voting machine case, the vendors are all saying, look, we test these things. We have a pre-test before the election and a post-test, and that proves that

the system must be doing the right thing.

For those of you who are computer scientists, you realize that that's sheer and utter nonsense. Yet, the claim is made that, because these systems are certified, they must be secure.

Now, it turns out that for one of the main vendors -- after the system is certified, the way they install the ballot face for a particular election is they change the code, after it's been certified, and they put this new software into each of the precincts' systems, which is different for each ballot face in each precinct, and they say, oh, but it's been certified. Okay?

I suggest again that we have a weak link phenomenon which has too many weak links in it.

So, very briefly, given the holistic nature of the problem and the tendency that we have to grossly oversimplify problems, I think the issues that we have to deal with suggest that we really need to look at technology as a holistic problem.

If somebody tells you that they have certification procedures or they have best principles or whatever it is, this is one piece of the puzzle, and all of that is good, it's useful, it's helpful, if you remember that it's only one piece of the puzzle. The real problem that we're dealing with is that in most of

the critical applications that I happen to deal with all the time with safety, reliability, security, and so on, ultra-critical systems, any weak link is enough to demolish the integrity of the system. Yet, if we have a system which is nothing but weak links, we have essentially no assurance.

So, I offer you as a paradigmatic example of this whole thing this election system, the all-electronic voting machine, with essentially no assurance that your vote goes in correctly. I suggest that you try to apply all of the wonderful techniques that we heard about in the previous session and try to seriously apply them to that problem.

Open source would help a little, maybe, but it's competitive. Everybody is writing their own systems.

At the moment, there is no way of telling when something has gone wrong whether it was an accident or whether it was fraud, because there is no accountability.

It is impossible to do a recount, because the bits are already there. If you do a recount, you get exactly the same result, even if it was completely flawed.

This is the bottom line that we're dealing with, and I can go on for another five minutes, but I

think I'd better stop at that point.

MR. SILVER: Thanks very much.

Howard Schmidt, how do you view this problem?

MR. SCHMIDT: Well, I'll start with the piece that I agree totally with what Peter said, and that's the fact that this is not just a technology issue. We've said for a long time it's the other PPT -- the people the process, and the technology.

As Peter related to, some of the early operating systems were very secure. We've seen some A1 systems that were secure.

No one bought them, because they were that difficult to use.

So, consequently, there was always that sort balance point that people were looking for. But oftentimes, as I look around and I see intrusions in the systems, I see flaws in systems, I see the way things occur, and sometimes it's about the coding itself. The errors that are made in the code, which we've been dealing with since -- 1976 is the first one I'm aware of, in which an intrusion took place due to a bad code in a proprietary operating system. But we also see, in many cases, configuration mistakes, and that goes to Peter's point that I'm in agreement with that these things are way too hard. They're designed not to be simple anymore.

And thirdly, the other piece that we see are errors that occur not just because of configuration, but because of an inability to maintain a system. It's interesting, because I try to put things in the analog world and compare to what we've seen over the evolution of automobiles.

In the very beginning, those that owned cars were people who could fix them themselves. I think back into the early days of the PC revolution in the early '80s. Those of us who could were doing it because we could fix them ourselves. Since then, like cars, we've made PCs easy to use. We can all do things with them, but we can't fix them.

We can't do our own brakes anymore. We can't, in many cases, repair our own computer systems. So, consequently, we can do more with our cars and computers. We can go faster in a car, we can do a lot more with a PC, but it's more complex to fix them.

Now, I do want to switch for just a moment and discuss something that I am not in full agreement with Peter on, and that's about the role that the market plays in this.

I think, significantly, having been there from the early days in the Marsh Commission to the private sector, back to the government and back to the private

sector, I see a tremendous desire, true, genuine desire by industry to do better, to the extent that people are spending millions of dollars of research and development from all of our major companies. Some of them sitting here at the table with us, some of them in the audience today. They are putting real dollars behind the problem, but the problem is it's not going to happen overnight.

We have built a system that has some flaws built into it. We're not going to be able to repair it overnight. We're not going to be able to, as I mentioned once before, even if we were to turn around tomorrow morning and hand everybody a CD with a secure everything, from a web server to an operating system to a word processor. If we were to turn around and do that tomorrow, we would still take three to five years before everybody would upgrade, because everybody has to migrate and remediate and do all these other things.

I'm not in concurrence with the view that market forces aren't working.

In closing, I just want to, once again, look at the broader perspective that Peter brought up about all the different ways one can do things. Once again, you're looking at this in the analog perspective.

There are ways to break into a home. You can kick the door down, smash a window, mess with the garage

door opener and get the door to open, wait till somebody takes their car to a automobile place, make a pass key for the home.

There are a lot of ways to do this in the physical world, and we've not solved those problems yet. They're a lot more tangible and a lot easier to solve, I would think, than in the electronic world, where many of the folks that are using the things don't even understand what's under the hood.

So, consequently, it goes into an area where we need to continue to work, because they are working in the private sector -- to make the technology self-healing, self-repairing, and self-configuring, to where security and privacy are, indeed, part of what we're doing.

Thanks.

MR. SILVER: Thanks very much.

Kathy Bohrer -- I know you have some slides, as well, if you'd like to go to the podium.

MS. BOHRER: Can you hear me? Okay.

So what I was going to do is just give a little taxonomy of privacy research areas, to give a broad view of technology that we look at when we look at privacy.

I'm from IBM Watson Research. I work with research teams, also, in Zurich and Almaden and Tokyo, plus we have a privacy institute that's made up of

external members from academia, from governments, and from companies that helps guide our research and set our agenda each year.

Anyway, this is just the little chart we use. It's got several areas in it.

The first one is privacy enabled services and applications.

That's where we would look at very high-level privacy problems like new services or new applications, new ways of doing things that would just give people improved privacy over what they have today. So, it's at the top of the stack.

It's a long way from the physical security that people have been talking about, at the opposite end of the spectrum, just how could you do things totally differently that would not intrude on people's privacy as much?

Federal identity management is one of those things. We heard about that in the first panel. Anonymous payments is something David Chaum has been working on for some time.

We have done a little research in something you might call privacy rating services, which is, you know, how do you help users understand privacy policies and be able to actually decide whether they would consent or

not, opt in or not, to something that's presented to them on the web?

Well, one way that some researchers experimented with was you start accumulating a body of evidence of what people have agreed to.

You start tracking what policies people consented to, and didn't consent to. Then you start providing that information in summarized form, both to enterprises and to individuals, with comparison, so they can see, well, is what this company asking for in terms of the policy they're promising and the consent they want -- how does that compare to what everyone else has agreed to or what other companies ask for that are trying to provide the same service? That's one way to start getting a handle on what the social conscience is around what should be acceptable and permissible and what shouldn't.

This next area of privacy management is some of the things we've heard already in other panels. It's the more concrete stuff about helping your enterprise classify their data.

Of course, unless you know what personal information you keep in your systems, or outside your systems, for that matter, as somebody brought up in the last panel, in Rolodexes or whatever, it's hard to figure

out what privacy policies you should apply to it.

Possible extensions to databases to push privacy control down to the same level that we push security access controls on data.

Negotiation of policies. P3P. When they first started out, they tried to do more with that standard than what it has actually ended up to be. I think there will be more as time goes on, but the idea is that it shouldn't be so one-sided.

Companies shouldn't just say what the policy is and then users have maybe some opt-in, opt-out choices. Otherwise, their only other choice is to find a different company to do business with. Perhaps there should be a little more negotiation.

But of course, one of the problems with that is most consumers would be overwhelmed if you really gave them a lot of choices to set the policy. So, we also study user models and user interfaces and how to try to get some of the complexity out of helping users know what rules to set.

That turns out to be particularly important in collaborative applications. Calendaring systems is an example. Location services through your PDA is an example.

Those are cases where it would make sense and

most users want to say who they're willing to have locate them on their PDA or in their car, who can actually look at their calendaring system, and all these kinds of things. To a small extent today, some of those systems allow users to make those choices. But if you imagine extending that to the richness of a privacy policy over all of your personal data and what companies can exchange the data with each other and use it for what purpose, it can be overwhelming.

Data minimization. I actually think this is a really interesting area, because it's totally different from the idea that, well, what we're going to do is we're going to set privacy policies, enforce privacy policies, help people understand privacy. This is saying, well, let's just get away from using personally identifiable information. Let's try to redo our business processes wherever possible so that we don't need personally identifiable information.

Let's randomize it for purposes of analysis, saying we're just trying to analyze data to determine our market direction in some products or something.

We may have no need, really, to know whose data that is. There are algorithms to randomize large amounts of data like that, so, in fact, it's impossible to go back and figure out whose data it was. Yet, the accuracy

of your data mining results is still good enough for the results that you need.

The anonymization work, anonymous transactions, and cash, and things like that, I think are also an example of this, where you just get away from having the personal information, and therefore, you get away from the problem.

Privacy is protected by either anonymizing information or summarizing it or randomizing it or some approach like that.

There is, as many people have said, privacy at what I consider the hard level that relies on security.

If you don't have security, then you can't have true privacy.

There's also research in extending security mechanisms to handle privacy concerns, and one of the ones I've personally worked on is access control.

You can think of enforcing privacy policies as just another kind of security -- access control. It's just that it's much more fine-grained, because you might want to have a different rule for how people use your business phone number from how they use your home phone number. So, that's a very detailed thing.

Plus, I might be willing to have my phone number used in a different way than Peter might have

wanted his phone numbers to be used. So, it just gets to be very much more fine-grained in most security access controls, which would generally be on the type of data, phone numbers, and the same rule would apply to everyone's phone number.

Different people might have access to phone numbers and other people might have no access to phone numbers, but it's unlikely you'd have security policies that said, well, you have access to Kathy's phone number but not Peter's.

MR. NEUMANN: Unless you're unlisted.

MS. BOHRER: Yes. So, that's an example we actually do have today, probably one of the very few examples we actually do have today.

Then the other part of privacy where you need to extend access control is, of course, with purpose, and we heard that a lot.

Since this is about misuse of data, you want to know what the data is going to be used for. By that, we don't mean just whether you're going to read it, write it, or delete it.

We mean what you're going to do with it after we give it to you. Are you going to give it to someone else? Are you going to use it in order to fulfill the order that I asked you to fulfill? Are you going to use

it to sell it to somebody else because they want to send me marketing material I don't want? Things like that.

Cryptographic protocols are another area of security technology, but it's also very important to privacy when you start talking about trying to anonymize things or de-personalize things.

Violation detection -- I think we've talked about that.

Steve Adler presented one of IBM's products that helps you enforce privacy policies in real time or to create an audit log where you could go back and analyze it after the fact.

Finally, I don't know how many people are actually doing work in this, and maybe this is getting at some of what Peter said -- you could do all this technology with the kind of software and hardware controls that I would probably come up with, because I'm really an engineer, not a researcher, but some scientists would say, well, yeah, but I could find a lot of holes in that unless I do a formal certification and verification, perhaps formal languages would help. So, there are things we can do to make the solutions we come up with much more rigorous.

That's what I had.

MR. SILVER: Thanks very much.

Ari Schwartz, are the technologies we've described so far up to the task? What else is needed?

MR. SCHWARTZ: Well, I think everyone, so far, Howard and Peter, in particular, talked about the fact that technology alone is not enough to do this. Howard said people, procedures, and technology, PPT. Nuala Kelly, earlier today, said P4P -- people, procedures, policy, and practices, adding the policies and practices side. I do think that that does get us a little bit closer to what is needed, a full framework there.

Good policies are, in some ways, more important than the technology, because they're what the technology gets framed around.

So, the policies really do have to be in place, and procedures have to be in place before the technologies can really kick in and work.

And I just want to give one quick example of what I mean by this, so that we can get to the point where the technology and the market forces really do kick in and improve privacy and security. That's in the ID management area.

You can have the new ID management technologies, but they have to be based on something, and right now, our ID management structure out there is broken.

If you look at the breeder documents, the documents that create other documents -- that is, driver licenses, Social Security numbers -- they are documents that, right now, are fundamentally corrupt in some way or another. The fact that we have to base other systems on these old systems that are broken causes problems down the road. No matter how good a technology we create for identity management, if it's based on this quick-sand model, it's going to be flawed.

Insider fraud remains a problem because of those other issues involved in ID management, and the security is still weak in ID management.

Now, technology can help solve especially those two latter problems to some degree, but they can't answer all the problems.

So, it goes back to what we've been saying ever since the FTC's been looking into the privacy issue in the first place.

Technology does play a role, a very significant role, but it's got to be teamed along with best practices, self-action by industry, including education and training, and lastly, baseline legislation that really does protect individuals.

Without all three working together, the technologies will not do enough to secure privacy or

security, for that matter.

MR. SILVER: Richard Purcell, do you care to weigh in here?

MR. PURCELL: Yes. I'll represent the people today on this panel.

Oftentimes technology is developed to function in ways that it does just because somebody figured out that it could do it.

My example of that would be peer-to-peer file sharing, particularly for music swapping. You know it could happen, right?

People figured out you could do it. You could listen to everybody else's music. Everybody else could listen to your music. Great.

Now, cool technology is the kind of technology that fills a purpose, but I've never driven a Porsche. So, would it be okay if somebody invented a technology that allowed me to drive somebody else's Porsche? Well, no. That's using somebody else's property without necessarily their permission. So, why is it okay to do music swapping?

We often overlook the fact that people have a reasonable sense of what's right and what's wrong, and technology simply overrides that, just because it can override that. It's so easy to do.

So many of our privacy and security violations aren't really because of flawed security practices. The technology actually works exactly the way it was written. It's not broken. It works that way.

And it works that way not because the security around it is flawed. It's because the individual said, geez, you know, I can either take a shortcut, which is a completely human kind of approach to problem-solving, or it's because they said wow, cool, I think it could do this, but I'm going to be very obscure about putting this in, because it's just because I can do this. Nobody is going to know about it. I'm the only one who is going to know. This is the old security by obscurity model that says, essentially, there's a back door into this thing but nobody knows about it but me, so that's cool, that's okay.

Well, there are a few vulnerabilities now that have exploited those back doors, and now we know that that's not okay to do any longer.

I've had personal experience that was rather dramatic and psychically damaging, when a grid was placed on the electronic registration process in Microsoft products, and it was placed there because it could be.

A developer, without documenting it, without saying anything about it to anybody -- it wasn't on the

spec, believe me -- said, hey, you know, we could do this, and maybe it will be useful someday.

Well, of course it's useful some day. It's useful to spy on people.

So, the point is I'm here to represent the people, both internally and externally, both the perpetrators, as well as the victims.

Perpetrators often just don't know better. A lot of developers that I know are not socially gifted and fully implemented human beings in a lot of ways. So, it is our job as individuals who have a policy framework, who have the ethical framework, who know what the long-term vision is -- not just can I ship this code on time, can I make it do all the whiz-bang things it's supposed to do -- but go beyond that.

Those are the people where I think the flaws are stemming from.

Those are the people who aren't providing oversight.

Have you seen the specifications for most software? I mean, really, the real specifications.

MR. NEUMANN: Typically there aren't any. Typically it's I want to make it do this.

MS. LEVIN: Richard, what about quality control processes? Is this an industry that doesn't have as much

quality control as we think there is in other industries?

MR. PURCELL: Well, I'd say that the level of quality control is completely commensurate with the way that we specify what it's supposed to do. Okay.

So, I want a lock on that door. Somebody puts a lock on the door. Well, damn, I can't get through that door, because the lock only operates during working hours, and I have legitimate reasons to go through it at other hours.

Is that a quality problem? No, it's a specification problem.

So, most software works the way it's designed to work.

Software can't work against its own design, right? Is that right, Peter?

MR. NEUMANN: Pretty much.

MR. PURCELL: It pretty much can't do things that it isn't designed to do without being modified. So, if it is vulnerable, that means it's designed to be vulnerable.

Now, that might be through negligence, it might be through shortcuts, it might be through stupidity, it might be through maliciousness, who knows? But pretty much it works the way it's designed to do.

So, it's a question of planning and oversight

in the first place. Quality control is certainly part of that, but it's also the specification.

We have to start thinking about this world not as a landscape.

Landscapes have trees and mountains and streams and things like that, but we essentially will sacrifice parts of that landscape, because we're only thinking of that part. But you cut the forest, it erodes the hill, it clogs the stream, and it kills the salmon. It's not a landscape. It's an ecosystem. It all works together.

So, you can't say it's okay, fine, I don't care, just shortcut this, just do that, it will be okay, because we think of those decisions as isolated decisions that only have the impact over the things that we are conscious of at the moment.

The problem is it makes guys in this room, in this panel, get old really fast.

Howard's 19 years old.

(Laughter.)

MR. PURCELL: The problem is that we're not thinking long-term very often. We're not thinking very far in the future.

Howard just said, look, even if we produced technology that was perfect, it would take it a long time to deploy it.

Why is it that privacy and security have rather suddenly, in social terms, in time, become a screaming issue. Why can't technology, which we all think of as incredibly rapid, solve this issue very fast?

Well, it's because technology isn't that rapid, honestly. It really isn't. It takes a while to build. I don't know about you, but I've witnessed how operating systems are built, and it's like sausages and law; you don't want to look.

It takes a very long time. There are a huge number of compromises.

People actually do this. These aren't made by machines. And people have a bad night or somebody yells at them and they come in the next morning and they're coding.

How good is that code that day, really. Have you ever driven a car that was built on a Monday? Don't buy a car built on a Monday, if you can avoid it. It's generally not that good quality.

So, all of these procedures just are indicators to me that we think about it wrong. We think about it not as an ecosystem which has mutually dependent parts, and where failure in one part almost always and necessarily is going to create failures in a different part.

MR. SILVER: Thanks very much.

Vic Winkler, do you have any thoughts here?

MR. WINKLER: Yes, I do. The first one would be to listen to Kathy about the microphone.

MR. SILVER: Excellent.

MR. WINKLER: So, I agree with many of the things that were stated here.

The difficulty for the products and the decision makers really comes when you don't have enough information to begin with, and you may not be aware of other choices, right?

The open source initiative is taking big advantage of that.

But as you take individual products and compose them into an infrastructure, for instance, for a small business or a larger business that manages information about me, I've come to be very suspicious of the level of skill on the part of the people doing this.

I think many of them don't really understand what it is that they're doing.

They've learned about these products maybe just by walking into the consumer stores and these products weren't necessarily designed to be put together in a manner that improves or even maintains a level of security, and that's what we have with sophisticated

solutions in infrastructure.

So, there are a number of different levels to the problem, and quality is certainly one.

I take a much more charitable view towards the people writing software, maybe because I work for Sun, right? But all humor aside, writing software is a defective process, and it's not fair to people who are engaged in it to write it off simply as a function of human beings engaged in a human process, although that's quite true.

But what comes out of the process are logical specifications that machines then execute. The tools that we use to write those specifications aren't really enabled to allow for the resulting products to be complete and correct.

Kathy mentioned formal methods before, and I'm a real believer in the need for the software industry to change towards one where we specify the logic and not the code, and where a process that itself has been designed and tested then converts the logic specifications into things that are executed, and then it doesn't matter who does it. The software will either succeed or it won't in terms of its evaluation by the process.

MS. LEVIN: For those of us who aren't technologists, what do you mean by saying let's work on

the logic and not the code?

MR. WINKLER: Okay. It's hard to talk as an engineer without slides.

MR. NEUMANN: Could I stick in a word on that?

Back in '73, when we did the fly-by-wire system, it was formally specified in a formal, logically defined language, and we mathematically proved properties about the layering properties, the synchronization, the distribution of information, the voting scheme.

This is a seven-processor system where everything was two out of three voting on the critical tasks, and there was a great deal of formal analysis, mathematically, logically sound formal analysis that showed that the algorithms were correct, the specifications were consistent with the requirements, the code was consistent with the specifications.

So, there's an example.

MR. WINKLER: Yes.

MR. NEUMANN: A 30-year-old example, but it's still an example.

MS. BOHRER: In maybe more layman's terms, if you think of mathematics as being extremely precise and everyone agrees that one plus one equals two, all right? And you think of expressing a policy or directions on how to get somewhere in English to someone and the chances

that it would be mis-communicated. Formal languages are much closer to mathematics than programming languages, which are a little bit closer to English.

MR. WINKLER: Absolutely.

My wife and I found that out when we spent about 10 minutes sitting on opposite sides of the living room about a year ago, each thinking that we're talking about the same thing. After 10 minutes, I said, Rebecca, it's astonishing. I don't think we're talking about the same thing. She said what? And we clarified it, and it was absolutely the case. So, the room for error in English and then in programming languages is significant.

As a former software developer, very few times do I see programmers doing anything more than rudimentary testing to see if the code will work as they think it should work versus testing it against unusual boundary conditions or under circumstances that it wasn't really designed to operate under. So, adequate testing is one of the problems.

That's an opportunity for somebody with a great deal of talent or even minimal talent, a hacker -- but there are some wonderful cases of incredibly creative exploitation of how to manipulate a piece of executable code to do something it wasn't designed to do and thereby take advantage. So, this kind of thing has to be

reduced.

That's not, however, where most of our problems lie.

Most of our problems do come from mis-configuration or systems that were designed predominantly with functionality in mind without taking care of other considerations.

So, engineering is really last on the list when it comes to most developers, most vendors, and most of the technology that you use.

If you want to continue to encourage the propagation of dangerous code, please continue buying technology that causes most of the problems.

I think that maybe the electronic equivalent of what happens at your firewall on a periodic basis, Frank.

MR. SILVER: Howard, do you have a point to add?

MR. SCHMIDT: Yes, a couple of points, if I could.

First, on the use of quality assurance in software development, this is a relatively new phenomenon, because quality assurance has been changing over the past years. It used to be the two major criteria were does it work and does it break something else, and is it functional. But what we've seen recently

is what I see as the paint-by-number scheme when it comes to IT development.

I failed stick figures 101 in school, but yet, I can do a paint-by-numbers thing and make it look pretty good, because all the pieces are there. All I have to do is fill in the blanks, and that's some of the modular libraries that make coding easy for us. If there is an inherent flaw within that particular library, it also becomes an inherent flaw within the application.

The other piece that relates to this, quickly, is the fact that we talked about how IT would make our lives easier. We've actually moved in the realm where, in a lot of cases, we've created a humanization of every IT system to where I've had identical hardware running identical bits on a operating system, and it does different things.

It's almost like the core DNA. You may be allergic to penicillin, I may be allergic to milk, but yet, we're still humans and adults and males and so forth. Consequently, we've seen this DNA-building of the IT systems, which in some cases is very unpredictable, just like it is in the human body.

MR. SILVER: Have we reached the point of negligence actions based on inadequate IT implementations? Does anyone have any thoughts?

MR. PURCELL: It's coming.

MR. WINKLER: Yes.

So best practices are being defined in all different vertical areas -- finance, health care, et cetera, right?

And over time, as these best practices become clearer to not just the practitioners in those areas but to the end users, the patients, the banking users and so forth, I think it's quite clear that the lawyers will take advantage.

MR. SILVER: Tony, I know you have comments on open source for later, but with regard to security right now, do you have anything you want to add?

MR. STANCO: I think I will keep my time for later.

MR. SILVER: All right.

Edward Felten, any remarks here?

MR. FELTEN: Yes. There are two things I wanted to say, although much of what I had planned to say has already been said.

First, although the discussion earlier in the day focused a lot on best practices, benchmarks, and so on, and there's been less of that discussion on this panel, it's important to recognize that best practices are incredibly worthwhile and really foolish not to

follow but also to recognize that they'll only get us so far. I think we're going to realize over time that best practices alone are not going to get us to where we want to be, best practices in the use of technologies of the sort that we're accustomed to using, because those approaches are fundamentally reactive.

They react to vulnerabilities that have already been found, that people have already been burned by, and it's a good thing to not get burned in the same way that someone else has been burned before. But it's also the case that new problems, new vulnerabilities, new exploits are always coming along.

The rate of new vulnerabilities being discovered, being exploited, is as high as always, and unfortunately, the speed with which the bad guys can exploit problems is only increasing to a really scary rate. We're going to have to become more pro-active about dealing with security problems, baking it in, designing it in, and that's what a lot of the panelists on this panel have been talking about. That brings me to the second thing I wanted to say, which is that it's important to recognize that all of the talk about better design, better quality assurance is right. That's what we need to do. But it's not the case that we know how to do that at scale for realistic systems -- and we're not

doing it.

There really are fundamental unanswered basic questions in computer science that we have to answer before we know how to do real quality assurance on big complicated software systems, and it's going to be a long time before that happens. I think one of the reasons the market is not providing that high level of quality assurance is just that no one is even close to knowing how to do it.

MR. SILVER: Richard Purcell, how do we go about protecting information better? What is the way out of this problem as you see it?

MR. PURCELL: Well, I think Kathy did a good job of laying out a framework that's useful. I think data minimization is one of the keys.

In the off-line world, we're very used to having collected, historically, a huge amount of information for every purpose.

This harkens back to a few weeks ago in the prior workshop where we talked about the example of how technology is so cool that states now can essentially encode your driver's license information more thoroughly onto an instrument, a driver's license, and make it retrievable instantly.

Well, so I want to go to a bar, and I don't get

carded anymore. I wish -- but they card me. Fine.

So, when you're carded to purchase alcohol, what is the data point they're actually looking for? And the data point is simply that you're over 21, period, end of story, not who you are, not where you live, not your weight and height, not your picture, not anything like that, simply that you're over 21.

However, the new technologies, the digitization of driver's license information combined with our legacy habit of using a driver's license to collect the age information mean that bars are now scanning driver's license, where possible, and collecting and databasing your entire identity, as well as the time that you came there, perhaps even some sequential number that associates you with other people who are also there, and all kinds of things like that.

So, why? Why are we doing that? Well, it's because we're used to it. It's because we've always done it that way.

So, what we're doing is we're not saying the technology, the digitization, the ability to apply technology to current issues gives us the opportunity to change our behaviors.

We just take the same old behavior and apply the technology, and we end up in these kind of messy goos

where there's just too much data. We have the opportunity to undo that.

So, data minimization is one of the keys, I would say, as well as the privacy management practices that are bi-directional, corporate and individual.

MS. LEVIN: Let me follow up with this question, use of Social Security numbers. Historically, we'll agree that they were started for one purpose and now they're used ubiquitously.

You can't even go to a doctor's office now without being asked to give your Social Security number, even though you're giving your insurance number and they're going to pay for it. There have been bills proposed on regulating Social Security numbers, and they're pretty complicated. Some of them talk about authorizing a lot of other uses because we're so used to using them. Businesses are very used to using them for a lot of purposes. It is, I think, a microcosm of the problem.

How do you see us getting out of some of these older systems and yet we realize there's a great need for people to be identified in various contexts? We talked a little bit about this at the last session, about data minimization.

But you have these tensions from government and

commercial entities that want the data.

MR. NEUMANN: There is a huge educational problem here.

One is that if your Social Security number and your mother's maiden name and other information that is essentially public record, such as your birth-date, are used as authentication information instead of identification information, there is a fundamental security flaw as a result of that.

Data minimization is part of the answer to that, but I think the burden -- again, maybe we get back to liability.

Anybody who uses a fixed password, a four-bit PIN, for example, that goes in in the clear and can be shoulder surfed, if you will, or photographed is vulnerable.

One of the most secure cryptographic devices that was created for public use was the clipper chip. The PINs on the clipper chip went in in the clear, and the idea that this is going to be a super secure system was, in that sense, a joke.

So, again, it's back to this oversimplification. We stick our head in the sand and believe that all of the stuff that we've been using is fine, and yet, we have practices -- this has nothing to

do with the technology, in a sense.

It's an administrative thing, the idea of using a password that is going to protect you, even though it's flying around the Internet in the clear or it's being given over a telephone, or a Social Security number that's used as an identifier, which is being used in the clear over the telephone.

This is a very foolish way to run a business, and I think there is a fundamental need for things like cryptographic tokens, for example. Then we get to PKI and then we'll open up another hornet's nest, because Carl and various others do not believe that PKI is a sound way to base an infrastructure, and yet, this is what is being done. The same thing can be said for SSL.

If the operating systems on which you're building your castles in the sand are fundamentally flawed, then your whole environment, your whole enterprise is potentially fundamentally flawed.

MR. SCHMIDT: Peter and I are in complete concurrence with this, because when you look at digital identities or PKI, which is something we've been very, very slow to move to -- I mean two-factor authentication is long overdue.

We have multi-levels of two-factor authentication, and for those of you who may not be

familiar, two-factor is something you have such as, in the case of my military ID card, a smart card chip and a PIN number, something you have -- or something you know, which means they have to put the two things together. This is very, very rudimentary, it works perfectly, but yet this has been around for a couple of years. I lament every time I go to a military installation or a government agency, I have yet to find a terminal to plug this thing into and utilize it.

We have it, the technology is there, but I have yet to find anywhere, including some of the offices that create these things and issue them.

So, consequently, when you look at it from a societal standpoint, that is one way we could go.

Once again, not everybody is going to be sophisticated enough to be able to walk in, get their card, understand that there's a level that is totally anonymous that gives them access to health care information that they may have concerns about, all the way up to INFALC on occasion so you can transmit security clearances for government meetings.

There's various levels we can provide, but what happens, every time we have a conversation, it's too difficult, the unsophisticated user won't understand it, so we do nothing.

MR. NEUMANN: And then the dependence is on the high-tech solutions. For example, the smart card, which is seemingly a high-tech solution, is itself vulnerable. We have friends in the community, good friends who are good people -- Paul Cotcher, for one, various others -- who have broken essentially every smart card that exists today, extracting the secret key out of the smart card in a very short time, but yet, a lot of technology will be built on that concept.

MR. SILVER: Let's talk now about convenience and the importance of convenience.

Alan Paller, is this something that's going to possibly lead us out of this problem, at least in part?

MR. PALLER: Clearly, building security in so the user doesn't have to be an expert and the system administrator doesn't have to be an expert is an essential first step. That was in the first panel in May. Nobody disagrees with that, I don't think.

A few panels ago, we had a member of the panel who, in an earlier life, sat in his dorm room at college and broke into systems and stole things and was really a bad guy before he figured out he could make a lot of money acting like a good guy. I thought it would be useful to take people very quickly through what he would do to old people's database and then what technology

would fix that real quick.

I just think it would be a nice way to pull our discussion together.

So, he wants the Social Security numbers. He wants some other stuff, too, because -- there are lots of reasons to steal people's data, but the one you can turn into money fastest is credit card numbers, because they sell for between 20 cents and $1.40 depending on whether you also know that three-digit code that you're never supposed to put in the computer and the expiration data. He wants other things, but he wants their credit card numbers.

So, how's he going to get them? I'll just take you through.

He's lazy. Not lazy. He wants to find the easiest way of attacking.

So, the first thing he does is he knows, as Peter said, the operating systems are fundamentally flawed. There are actually two problems in the operating system.

One is they had mistakes in them.

A CIO from one of the Federal agencies was sitting at Microsoft, and Balmer bounces in the room, and news had just broken about another buffer overflow, and he says damn it, I thought we'd figured out how to fix

that problem years ago.

So, the operating systems are fundamentally flawed because the programmers make errors -- that's a small problem.

The big one is they're fundamentally flawed because people install them configured unsafely, and they do that because that's the way their friendly vendors told them to install it.

There's no end user stupidity here. That's how I got it from my vendor.

So, the first thing I do is I just check to see if any of the common vulnerabilities are there, because the common services are there. I do a real quick check. No trouble. I'm in.

Okay.

So, that's the easy one. I get by that one.

Maybe they've configured it right so I can't get in that way.

Then I decide, well, all right, they've got a database accessible, meaning I'm a user, I want to get into the database, attack, the same thing. The database people make mistakes in programming, and even worse, they make mistakes in configuration, exactly the same as the operating system people.

So if I can't get in on the operating system, I

can come in at the database, and the third level would be the application.

I could do both of those attacks at the application level.

I want to say something about configuration.

We expect the system administrators to configure the system safely. All of you who work in large organizations hire people to do that.

Just a short time ago, one of the largest system vendors was running a training class for law enforcement people in Washington. On the night of the first day, the guy who paid for it walked in and said this is great, we love learning how to run the systems, but what we really want to know is how do people break in and what should we know about blocking those kinds of problems. Because you are the experts, you're the people who would know, please teach us that.

He said I'll come back and tell you by 10:00 in the morning.

He came back the next morning and he said it is corporate policy not to teach that to students. This is one of the largest vendors.

It's true of all of the vendors.

If you have a person who has a certification from the vendor in system administration, he has never

been taught security, never.

To the extent he has been taught security, he's been taught how to run the for-sale security products that that company sells but not how to secure the basic operating system.

So we have a situation where we're expecting people to do things that they can't do.

So that's why Dell's move is so important.

MR. NEUMANN: There's one other fascinating problem there.

IBM is doing a phenomenal job in their autonomic computing program -- that is, a system that basically doesn't require a lot of system administration, because it's going to keep on running no matter what happens to it. It's going to diagnose the fact that it's under attack and reconfigure itself and so on.

The problem there is that suppose you get rid of all your system administrators, or most of them, and they get lazy because things don't go wrong anymore, and now something breaks.

You're in real trouble, because you have either got to out-source your critical system administration to some third world Beltway bandit subcontractor or you have to have a guy on staff 24 hours a day on call, or a team of people, who could come in and be skilled enough to

repair the system under conditions that you've never seen before.

MR. PALLER: Yeah. Nothing I was trying to imply said that you don't still have phenomenally skilled system administrators.

It's just you can't expect all of your system administrators to know how to install it safely in the first place. That's what I'm saying is the error.

We have to train the system administrators. We have to get them up to speed, because they're going to have to deal with new problems as they come up. But day one is where we shouldn't make every single human being who ever buys an operating system from anyone be a security expert. It ought to come out of the box safely, and the idea that it doesn't is malpractice.

I mean it's just stupid, and they've known it for years.

Sorry.

Okay.

So those are the easy attacks.

Let me give you an attack a lot of people don't know about.

We're still stealing their credit card numbers.

Now, this won't work at eBay, because they know how to solve this problem, but there are places where

this will work, like 100 or 200 thousand other places.

It turns out the person who sold you the storage devices on which you put the data in the database is not the person who sold you the database or even the person who sold you the computer.

This is the guy who sold you this raid box or the switches and the storage devices that you stick it on.

So it's the hardware, the servers that the data is on, all right?

Well, it turns out that a lot of them have a dial-up port, because they want to make it easy to maintain it, because up-time is the single most important thing. So, they have a dial-up port, and some of them have a dial-up port that has no password on it, and the ones who do have passwords on it have known passwords on it, and you wouldn't want to change the password, because then the maintenance guy couldn't get in, all right?

So, what's the general solution to that problem? What's the general solution? Encrypt it, so that even if they get the data, they can't -- that's why Howard doesn't have the problem, I hope. So that even if they get the data, they've got to go to some of Peter's best friends, and if you make the price high enough to break it, you'll lower the barrier.

MR. NEUMANN: I've got a story I've never told in public, and I think it's time.

Probably 18 years ago, I went up to Alyeska in Alaska and did a security review of their pipeline control system, and I discovered that every node in the network used the same dial-up password for their switch in the router -- I should call it a router, I guess, but it's a one-way router, and it was the same password that was being used by the vendor everywhere in the world.

MR. PALLER: That problem is not limited to Alyeska. Cisco classes teach you to use one of two passwords, which I won't name, and almost everybody thinks because it's in the manual as an example, that they should put that in their routers.

So, those two are in some reasonably large percentage of all routers.

Okay. Two more quick ones, and then I'll get out of here.

Say you've got the systems and they're okay, the hardware and the software and it's okay, but you still want to get in.

The organization has set up, because it's smart, a VPN that allows people to work at home over the Internet, but it's all encrypted channels, so it's all safe as can be.

Most people don't understand the VPN is not a security system. It's a pipe. It's a pipe with a hard wall. The hard wall is the encryption. But if the PC at the other end is used by the person's teenage children, what are the odds that it has a file-sharing program on it with access. Once you have that on it, the VPN is a pipe into the system, and you are a validated user of the system and you've gone around all the things. If that doesn't work -- and say I really do want to get into eBay -- then what I'd do is I'd spoof an e-mail message from Howard to 50 of his system administrators.

"Spoof" means send them a letter with the return address on it that says Howard Schmidt and you can do that really easily, really easily. So, you send them lots of e-mails, and they all say, wow, my friends at Microsoft -- everybody knows he used to work at Microsoft, so "my friends at Microsoft" sounds right -- just told me there's a big bug in Internet Explorer and we've got to get it fixed. They haven't made it public, but they've set up a special web-page for us to download the patch. Click here.

Well, the "click here" works. It just doesn't take them to Microsoft.

Would this work?

MR. SCHMIDT: No, because everything I would do

would have a digital signature. It would not. But in a lot of instances, though, you are correct.

MR. PALLER: And that one takes training.

So if we fix everything on the hardware and software side, we haven't fixed more than 50 percent of the problem.

The other 50 percent of the problem is I can fool you into opening that. Almost no one else uses digital signatures, even the guys who sell them. So, I can fool you into going to a website thinking you're going to Microsoft, download a patch, put it on.

That patch actually opens that computer, bypasses the firewall, and the computer goes to a website looking for commands. So, you're not getting in, it's going out.

There's absolutely nothing to stop it.

Those are the ways I would get you. There's technology fixing all of that stuff.

MR. NEUMANN: I had a wonderful thing in my "Inside Risks" column from some Russian guys who pointed out that if you put the "O" in Microsoft in cyrillic instead of in our alphabet, it was indistinguishable, because the "O" is identical in appearance on the screen, and so, microsoft.com with the cyrillic "O" gets you a very different website than the one you'd think you'd get

to.

MR. PALLER: That's a hard one to fix.

Okay.

So, just quickly, what Dell's doing is absolutely the most important stuff that's happening. We have to have that kind of configuration baseline in every application, every operating system, every piece.

The other reason Dell's work is so important -- and it is the one that people miss -- is that a lot of the reasons the operating system can be broken into is because the applications force you to undo security, meaning the application was written on an unsecured operating system.

So, if you want to install that application, you are forced to make your computer un-secure. Even if you installed it with Dell's technology you have to turn it off. IBM's got some products that do this to you, because the developers wrote it for an unsafe version of Microsoft or for Windows.

You want to do that, but the guy wrote it for the system the vendor sold.

Once Dell starts selling a system that people say it’s a safe configuration, then buyers can say I'd like to buy my applications and I want you to certify that it runs in a safe configuration, but until somebody

as big as Dell or as big as Microsoft makes that kind of move, nobody can act sensibly, because they don't know which configuration to match to.

It's a wonderful year for progress.

The vendors are really doing a lot of work.

They're making some moves that are purely pecuniary.

Like Microsoft does this thing where they'll automate a patching, which is absolutely essential for all of the grandmas in the world, but they won't do it for anything you already have. You have to buy their new operating system.

So, it's pecuniary, but it's moving us forward in the process. If people want to know more, I'll be happy to fill in all the good things that have happened, but it's been a very good spring for improving, not getting us around the fact that we still have problems, Peter.

MR. SILVER: Tony Stanco is here to talk about security, privacy and open source.

MR. STANCO: Actually, I guess it's appropriate that I'm going at the end, because open source is almost a parallel universe that really doesn't touch a lot of these other places.

I'm going to talk a little bit about open

source, which is really a completely different way of doing things, and like the flight of the bumblebee, it really should not be working, except it is.

Open source is gaining momentum around the world. Basically, all the major companies have some kind of open source strategy.

This isn't a coincidence, because Wall Street requires it.

They don't, they actually get penalized on Wall Street, and if you've got a mixed message, you get penalized, too.

Europe, China, India, South America -- they're probably ahead of the United States. The United States has the risk that it might fall behind, except just last week, DOD issued the first, for the Federal government official policy statement. It’s in the package.

It was dated May 28th, and it really just got off the press yesterday.

What the memo does is just basically level the playing field between proprietary and open source. So, the government isn't picking on anyone who's here.

That also shouldn't be very exciting or surprising except because of the lobbying that's been going on for the last couple of years. Ptech October 2000, basically said the Federal Government should level

the playing field for open source, except between then and now, there's been a lot of activity, let's say, at the political level.

Also in the package, there's a Mitre report on the use of free and open source software in DOD, and what it said is that if you try to yank out open source from DOD, you basically lose your security. It actually is even stronger than that. It actually says you can't plug into the Internet, because most of the Internet runs on open source software.

So, open source is important. That's the basic message there. Open source security.

All right.

NSA -- I'm sure everybody here knows about the NSA. They started a security-enhanced LINUX project, SC-LINUX. NSA has been worried about the critical cyber-infrastructure for a long time, but really, in the last decade, they were very concerned.

In fact, they're concerned that there isn't even a secure operating system, and you need to start at a very fundamental level.

What they tried to do is they have this architecture, mandatory access control that's used in certain military installations. They tried to give it to the proprietary companies about 10 years ago. Before

9/11, there wasn't a market for security, as some other people have mentioned. So, nobody adopted it.

The technical people thought it was a great idea. The marketing people said it's a cost center and nobody is going to pay for it.

So, it didn't work. It didn't get vectored into any of these mainstream products.

So the NSA said, hey, let's give it to the open source people; maybe they'll take it.

Well, they took it, and there's a lot of activity in the security enhanced LINUX through the open source community, through the university where we are through a lot of universities around the world, in fact.

All right.

Let's talk a little bit about security. Security really is still very misunderstood. I think there was a sense at this event that there's a lot of ambiguity and a lot of misconceptions.

I've heard some of the same things here.

I was at a CIO council web services working group meeting just recently, and they talked about securing the web services applications. And they didn't worry about anything below the stack. But the NSA has made it very clear that you really need to start as low as you can go, because otherwise, doing it at the web

services level, you're really talking about bulletproofing the third floor of your house and leaving wide open the doors and windows of the first and second floor.

In fact, there's an NSA colloquium on secure systems going on this week, and there was somebody from Australia who said forget about the first floor. Threats to security are working below that. They're going to the real foundations. They're working in assembly language. They're working at the hardware level. They're working at the BIOS level. So, if they want to get you, you can even have a secure operating system, and they can get you.

But the point is that's a good place to start. That's a nice dividing line, because that's where the software starts, for the most part.

Unless we get at least that low, nobody should have a sense of security. It's all smoke and mirrors. The vendors will tell you that it's secure. They'll tell you that they have great products. But you know, they're just selling you products.

MS. LEVIN: Tony, you're saying the level you would start out would be the operating system?

MR. STANCO: That's what NSA said.

QUESTION: The BIOS?

MR. STANCO: Yes, you should, but let's start with the operating system. You can always go lower, but that's a nice place to start, and that’s where NSA wants to start. That’s what they're trying to do with the SC-LINUX.

They're trying to get the secure architecture up there.

All right.

Let's talk about open source security. I'm not here to say that open source security is going to be any better than proprietary. There's no definitive study. I'm not going to make that claim.

You know what? It doesn't matter anyway, because they both aren't good enough.

Security is not something that is baked in, as somebody said, or architectured inside the development process, and this is very key.

Neither open or proprietary is doing a very good job.

The good news is both are starting to look at it. SC-LINUX, a lot of the proprietary companies -- Microsoft, IBM, Sun, Oracle -- everybody's looking at security at this point.

The bad news, again, is that none of these are going to be usable products for the next three to five

years, as somebody mentioned, because you have traditional product cycles that really rev about that speed.

All right.

The other good news -- and there are some pieces of good news -- is that there's some other things happening -- Common Criteria -- NIAP, which is the National Information Assurance Partnership between NSA and NIST. They require at this point, as of July 1st last year, though there's still some wiggle room since there wasn't enough product in the pipeline, that sensitive software, military systems, has to be evaluated and certified.

Now, this is good news, because once they basically debug the process, the CC-NIAP process, everybody expects this to go to the civilian side of the government and then to everybody else, here and international, because at CC, the common criteria part of that is really international. So, the future is starting to look a lot brighter if you have a far enough horizon.

But let's leave all this aside, too, because open source is different, and it really goes to fundamental ideas of not only technology but society and organizational structure.

The bigger question that I want to raise here

that I don't think anybody else has raised is who do you want to protect, who do you trust to protect citizens? Are you going to trust companies? Are you going to trust government? Or do you have to find somebody else? Is there another group?

Well, let's talk about companies. They have fiduciary duties to maximize profits for shareholders. That's not a bad thing. I used to work for the Securities and Exchange Commission. I mean that's a good thing, right? They created a lot of wealth in the last 300 years. But we just have to realize that their mandate is not to protect consumers or citizens.

Now, the theory, how the free market relates to societal benefit is that free market competition among the companies checks the ambitions of any one particular company. So, the competition and the market regulation has, through this competition mechanism, achieved the societal goals.

So, you have this invisible idea. I'm not saying that's wrong, because we know it's right. You can't say that it didn't work.

You have eastern Europe. You had East Germany. You had West Germany. I mean, come on, same people. The only difference was the legal system and the ideas, the principles of free markets and democracy.

So, there's a real test case there that says this -- there's something there.

But the key point is you have to have a dynamic market. You have to have the competition. And software has network effects, especially once you get to the Internet. Hopefully, everybody knows what network effects is.

The value of the system or the product increases exponentially with every person who gets added to the system.

So, that creates monopolies. It creates situations where a particular consumer cannot choose, because you could choose to unplug from the electrical grid or you can choose to unplug from the phone system or you can choose to unplug from the computer infrastructure, but you don't have choice beyond that. The choice is in the system or not in the system.

Market regulation -- we can probably cite two or three cases that point this network effect out in the antitrust area.

Let's just assume that markets aren't sufficient. We don't even have to conclude that. Let's just assume for argument's sake.

So, what happens then?

We can't look to the governments -- to the

companies, let's say. Can we look to the government? Well, the government usually steps in. That's the usual solution when there's a market failure. But in the past, government stepped in in slow-moving capital-intensive industries. So, you generally regulated the assets, which is feasible.

But software, IT -- that's not how it works. It's a fast-moving, innovative industry.

Industry will always, in my opinion, outstrip government's ability to do oversight. They have more assets. They can incentivize. They can give stock options to even the best in the government to bring them into the other side.

Can government really provide effective oversight when it relies on industry, in the first case, to constantly innovate?

Again, who do you trust to protect citizens?

The problem actually gets a lot worse. If that wasn't bad enough, it actually gets worse, because software in cyberspace is functionally equivalent to law in physical space.

Basically, law regulates interactions between people, between businesses and people, between businesses and businesses, between people and businesses and government. That's really what all the rules are all

about.

Software does exactly the same thing in a cyber world as that, exactly the same. You will interface not with people directly but through your machine. People are already talking about these mobile agents that go out and actually do the contracting. There's a real indication that this is not completely out in left field.

These agents are supposed to set up your contracting terms, and go out into the Internet and actually execute the contract.

So if that isn't law, I'm not sure where we're left.

Let's extend this a little further. Let's say we can arguably say that it's like law.

Now, the creation of law, as everybody here knows, especially in this town, is a very complicated organization, carefully structured with checks and balances, because it's fundamentally too important to society, too important to democracy, to free markets -- it's the most basic layer.

So, we have legislatures, courts, executives, executive agencies, the legal profession, legal schools, political journalists. We have think tanks. As somebody mentioned, there’s this ecosystem that, works out the legal rules.

So, if software is like that, where are the checks and balances in the creation of software for protecting the consumers and the citizens?

And if you look at it from this perspective, do you really want to leave it to the market, which doesn't seem to be able to control the appetites of business in the first place?

You can obviously have a company -- if we thought it was such a good idea, we can have a company, for efficiency reasons, create our laws.

Why is that different? Why would we not accept that?

If we leave it to the government, is that a good idea? Because it's a fast-moving industry. It’s not clear that they can do it.

What I'm saying here in this roundabout way is that the issue may not be at the level that was proposed in this panel, because the question might not be how do you design technologies to protect consumer information at this particular time or at this particular place, but it's probably fundamentally how do you design a system that will design technologies, that will protect consumers, because the dynamics of the environment are such that a solution isn't going to help. You need a system that will adapt.

If you leave it to the industry and if you don't want to go down this road, these institutions lack the checks and balances. I would suggest that you're constantly going to be where we are, which is always behind industry, trying to catch up.

Industry is going to exploit and harm consumers, and there's going to be an outrage at some point. They take a lot, but at some point, they become upset and they complain, and then policy people like the people in this group, like myself, come up and try to find a solution for that problem.

By the time we cycle through that problem, industry has said fine and they're off to the next problem and the next exploitation of people.

It's not a problem of a technology. It's not a problem of policy. It's a problem of structure. And unless we solve that problem, this is an ongoing thing.

All right.

I'm here to talk about open source. Where does open source fit in this?

Well, like open government and transparent law creation, as a first step, you would expect, if software is law, that you would need open inspection of software. But I'm not going to say that open source at this time has the necessary checks and balances to protect

citizens.

Yes, it's better than companies, in my opinion. Yes, it's more capable of government, because they're technologists that obviously can duke it out with all these companies on the same terms. But it still lacks, for a system, the appropriate accountability that society would require for legitimacy. The appropriate accountable structures still need to be created even if you're using open source.

But realizing the past responses, what we've done in the past, how we've looked at things in this new cyber-world, it isn't going to work.

That is, itself, a first step. Open source, in my opinion, is a partial answer. It's a starting point. But you really need to get to the point of thinking and laying out and designing accountable open source development systems.

That's where the time should be spent, in my opinion, not designing, as I said, the particular policies of the moment and not just trying to play catch-up with industry.

So, that's where I'm going to end.

MR. SILVER: Dr. Neumann, any comments on open source?

MR. NEUMANN: Yes. That was quite a speech.

Let me make a couple of comments.

One is that you're absolutely right. Open source by itself is not a panacea.

Without the things that seem to be not present in the proprietary development process as much as they should be -- namely, attention to system architectures, attention to good software engineering practice, avoiding some of the problems of legacy system backward compatibility with every system that's ever been built in the past or monster cut-overs through architecture for distributed systems -- one can achieve, I think, very high security reliability and so on. But that applies to both the proprietary world and the open source world. Without that, it is very difficult for us to have the kinds of systems that we need.

Now, your argument is good in the sense that the open source world has an opportunity to do things that are much more difficult to do in the proprietary world.

I'll give you one example, the DARPA program called CHATS, which is Composable High Assurance Trustworthy Systems, of which I happen to be one of the contractors. It is purely open source. Everything in it is open source. It's taking LINUX VSD variants --

MR. STANCO: We're part of that, too.

MR. NEUMANN: -- and making some truly considerable improvements in what can be done in open source by itself.

But without the discipline that is required to develop systems, the open source thing is not going to go anywhere either, and I think --

MR. STANCO: Can I respond to that?

MR. NEUMANN: Yes, sure.

MR. STANCO: Granted.

But I'm just not sure how using proprietary methodologies solves the problem.

In fact, I would think if you have open source, you teach open source, you teach architecture that bakes in security to the students, who then go out in five, 10 years and implement that, you're in a much better position than having students work on a closed system, a black box, you know, click here, click here, click here and it will be secure and go out and work on that.

MR. NEUMANN: I agree.

The point I was going to make was, in fact, the exact opposite, that the stuff that has come out of the CHATS program -- for example, some of the tools that came out of my project done by the Berkeley team for finding all kinds of security flaws based on formal methods, oddly enough, are perfectly applicable to proprietary

software, as well, if only they would use them.

MR. STANCO: If only they would use them, exactly.

MR. NEUMANN: Let me finish my comment.

Multi-level security was mentioned here. I want to point out that there are some potential open source solutions to multi-level security that the marketplace has not picked up on.

One is work we did back in the '80s on showing how you could put an off-the-shelf Oracle on top of a security kernel and the result is an A1 -- effectively, a very secure multi-level secure database management system without having any trust in the database management system for security.

MS. LEVIN: Peter, why did the marketplace not pick up on that?

MR. NEUMANN: Well, Oracle discovered they could do something on their own.

We worked with Oracle, actually, on that, and they discovered that they could modify their kernel a little bit and come up with something that was multi-level secure. Nobody wanted an A1 system at that point. It was not practical. It cost too much to develop it. And the evaluation procedure was so complicated that it took years, and by then your software had gone many

levels beyond it.

There's an architecture that Norm Proctor and I came up with in 1992 on how to build multi-level secure environments out of single-level components and some trustworthy multi-level servers.

So, all of the trustworthiness is in the servers for multi-level security. That's something that can be done essentially off the shelf, with a few open source trustworthy servers and anything else you want to use, and you actually can wind up with a multi-secure environment.

The tools that have come out of the CHATS program I think are very important and very applicable to open source, but they're also applicable to proprietary stuff. The key argument comes back to the question that we raised earlier of whether the research community is having a real influence on the marketplace, and I think there may be arguments. Howard made the case that, in fact, the marketplace is becoming much more aware of security.

Certainly, Microsoft has made a huge effort in the last year-and-a-half. They spent, what, 1,200 man years in February of last year alone, although maybe some of that was just a half-day course on how to make secure systems, I don't know. But the point is that there is a

need for a cost-driven marketplace where there is a real incentive, whether it's financial or jawboning or whatever, to the mass-market software developers to produce stuff that is much more robust.

If you look at the buffer overflow problem which was mentioned earlier, buffer overflows have been around for 30 years.

We've known how to get rid of them for 30 years, but they are pervasive, and they keep appearing and reappearing and reappearing. CERT keeps showing that half of the breaches in securities laws over the past four or five years are attributable to new buffer overflows. They keep recurring.

But we know how to get rid of them by using intelligent architectures and intelligent software and intelligent use of programming languages and programming style. It's easy. But it's not in the interests of a marketplace whose primary goals are not to develop secure systems.

So, if that's changing, I welcome it, I think it's wonderful, but it's a very slow process.

MR. SILVER: Are software development contracts being written at all to shift risks to the developers in case of security breaches?

MR. NEUMANN: Ed would be a good one on that.

MR. SILVER: Professor Felten.

MR. FELTEN: Actually, I think someone else on the panel would be best equipped to answer that.

MR. SILVER: Go ahead and make your remark. Maybe we can save the question for later.

MR. FELTEN: I just wanted to amplify a little bit on the point Peter made about buffer overflows. As he said, it's a very common category of bug. It accounts for half of the CERT advisories, and it's a problem we know how to solve. Yet, both proprietary and open source software is still rife with buffer overflows. This should be telling us something, that, in fact, there is an awful lot of inertia in the software development process and that it's not the case, I think, that industry has been lax in picking up the knowledge that does exist about how to develop more secure software.

I think it's just much harder to transition basic knowledge about security into practice and especially into the software development process than many people realize. I think that although it's true that commercial software has not improved all that much in security, that's more a reflection of the fundamental difficulty of improving security as opposed to anything that's broken about the process itself.

MR. SILVER: Tony, then the last word to Alan.

MR. STANCO: I'd just like to respond to Peter on four basic points that he brought up, or themes.

Okay.

The research community -- it seems to me that open source follows the scientific method of allowing everybody to share code, results and experiments and everything else.

I don't see how there's a conflict with open source. It seems to be a reinforcement. It seems to go back to first principles. And I'm reminded of a story where people didn't used to share ideas.

In fact, a few hundred years ago, heart surgeons didn't share their techniques, and society at some point said, you know what, I don't think you should die with those techniques, because there are other people who can be saved. Maybe this is the same; maybe it's different.

You talked about coexisting, I think, or one or the other.

I'm not sure this is an either/or situation.

I think the government, as a policy, should say it’s a level playing field, which is what the DOD memo said. I'm not concerned about it.

I personally think that open source has been under-estimated from its beginning.

People, 10 years ago, never would have imagined it would get where it is, and I think they're still under-estimating.

So, I'm not concerned about a level playing field. I'm concerned about de facto or de jure prohibitions. But if we can level the playing field -- for example, de facto would be that procurement officers must consider allowing is open source software procurement. A lot of the software lobbyists were being dropped into state legislatures to oppose procurement officers from even considering open source -- not just buying it.

You talked about security and I talked about the fact that there's no definitive study between open source and proprietary that would sway people, reasonable people one way or the other, but there's still anecdotal evidence that open source is more secure.

What is this? Basically, every military establishment around the world uses open source. They don't trust proprietary.

Now, there might be a lot of reasons for that. Some of those might be social reasons. Some of those might be nationalistic reasons. But those are still security issues.

Let's pick on one of our enemies, like France,

and you're not sure if NSA sees all your documents. From France's point of view, it's a security problem if there is something in there that redirects all your information.

And the last thing -- I think this is a very valid argument that you brought up, the business model. I don't think you called that a business model, but you said these people have to be paid or something to that effect. Otherwise, there's no incentive.

That I agree is very important, though I have a lot of faith in the free enterprise system, the free market system.

I think if government stays out of the way and says everybody play this out, things will rise to their appropriate level and bad solutions will fall to their appropriate level.

I think, yes, business models are currently lacking from open source, but I also think that people are working on open source business models. I actually think that they're going to develop them pretty quickly, because this reminds me of what happened with LAN's and the Internet. The same arguments, right, that you can't use a public property Internet to really do anything. You've got to buy up proprietary LAN's, because you need to have incentives. You need to have a company behind

these solutions. Who is going to support a public good Internet? Well, that's not how it worked out.

MR. SILVER: Alan, you had a comment?

MR. PALLER: Yes. It was in answer to the question you asked.

MR. SILVER: I think you and Howard both had responses to my question on contracts.

MR. PALLER: The question was, is anyone doing something contractually to require --

MR. SILVER: Right.

MR. PALLER: -- safer systems, and the one example that I know about, although I've heard of four -- I just didn't write them down.

The one I know about is Virginia Tech has required for the last year that every software vendor that sells them a software package certifies that that software package has been freed of all 20 of the 20 most common security vulnerabilities, and of 620 vendors, only two have not been willing to sign.

Probably that means 300 are lying, but it definitely is a method. The reason I wanted to make the comment wasn't just to answer the question. I think that's the lever.

If you wonder how are we going to get more secure systems, given what Dell is saying, that customers

are actually beginning to ask for it, there is one software vendor, big software vendor, that just rails against benchmarks, just, oh, no, we don't want that. Everything's different. The whole world is different. Everybody's different, therefore no security benchmarks.

And one of their customers came to them with $100 million and said we want to buy a lot of your software, but only if you'll deliver it according to these benchmarks. Oh, sure, absolutely.

I mean publicly angry about it; privately, of course we'll do it.

And I think that's the lever. As Dell proves the vendors can do it, as the customers prove there's a market for it, I think we roll over, and then the other really wonderful thing is at the FTC.

People are now promising security. The FTC has a spectacular role in saying if you're going to promise it, please deliver it. I think that combination of the market moving and the FTC saying put up where you said you were putting up is really wonderful, and thank you for running this workshop.

MR. SILVER: Howard. Then we'll take questions.

MR. SCHMIDT: I didn't know there was a "please," but thank you for doing it anyway.

A few quick points.

One, yes, there are a number of instances where there are contractual agreements, service level agreements, whatever capacity you want to call them, that say you will do this certain level of security, and if there's a failure, you will notify, you will contact. There's a whole plethora of issues that are going into contractual agreements now on that issue.

A couple of quick points on Tony’s remarks, and I have a tremendous amount of respect for Tony although I disagree with a lot of what he says.

On the market forces, there has not been a market failure.

If there was a market failure, the government would have stepped in. There has not been.

The market has shifted. The market has corrected. The market is doing a lot more but once again, as I think we're all in agreement, this is not a motor boat we're turning around. This is a 600-foot tanker we're turning around to get these things going.

Also, the National Information Assurance Partnership (NIAP) doesn't do much to level the playing field.

NIAP is very expensive. It's very time-consuming. Only the big companies have the ability to

participate. They do a tremendous job. It's very valuable. But we were called when I was at the White House as the President's Special Advisor for Cyberspace Security to look at NIAP and see how we can make that a better tool to improve security.

And lastly, the evolution of things -- I remember back in the early days of CPM, for example, there was a lot of free-ware that evolved into share-ware that evolved into commercial software.

So, what may be an open source today indeed may be proprietary and commercial software later on, which is not a bad thing.

And in closing, it's tough to have it both ways, Tony.

Either the government needs to be in or the government needs to be out.

If the government creates a playing field, that's government intervention in what I think a free market economy should do.

On the other side, you said the government should not be be meddling in these things, and I truly believe that's the case.

The government should keep a hands-off approach, provide some technology, and provide some research, which is vitally needed across the board to

make this better.

Thank you.

MR. SILVER: Thanks.

MR. SCHWARTZ: Can I just ask a follow-up question of Howard?

MR. SILVER: Sure, one quick one.

MR. SCHWARTZ: At the beginning of this, you were saying that, contractually, a lot more companies are asking that when there's a breach, that it be known. How much of that is due to the California law and how much of that happened before that law? Were we moving that way already, or has California law pushed that over the edge?

MR. SCHMIDT: I don't have any hard numbers, but from what I've seen, this was taking place long before the California breach occurred, because companies were looking at this issue, as part of the business process -- I need to know these things.

I know I was working on these issues two years ago. If we do a joint venture, business partner, merger and acquisition, that was part of the criteria for establishing the arrangements.

MR. SILVER: First question, please.

QUESTION: Vincent Schiavone, from ePrivacy Group. I had a couple of points to make. First of all, I think we've done a little bit of a disservice here

today to answer the question, designing technologies to protect consumer information, to get into a religious argument about open source and closed source.

When we talk designing systems, designing closed systems, proprietary systems and open source systems, there's some basic fundamentals that we did not discuss today.

When we look at technology, technology is not what makes things secure.

Technology can enable us to monitor security. It can enable us to enforce policies. But there has to be the requirement for secure systems and accountability, trust and accountability of consumer information.

Right now, you can build systems much more securely than we are building for consumer information. There is no accountability required for tracking information as it shared outside of the systems, okay?

That's the fundamental nature, and the question comes down to should it be designing technologies or are we going to require technologies to protect consumer information?

Some will argue that we already have the laws in place to do that.

Two examples I'd like to talk about.

One is standard of due care and how this plays

in software development.

We heard an example today about spoofing of e-mail addresses.

We have eBay and ex-Microsofters up there.

It happens every day of the week with very large companies.

We're talking about corporate identity theft. We're talking about individual identity theft. We're talking about real theft and fraud. Yet, there is no requirement that they use the systems that have been around, as Peter said, for many, many years to make this trustworthy and accountable.

So, we can't design a trustworthy system until we require that there be one built that handles consumer information.

The other point I'd like to make on standard of due care is that after events happen, how are we holding people accountable?

The FTC has a role. Technology has a role. Best practices has a role.

But until we have a standard that's acceptable and required, there won't be a change.

Bits are bits.

When we look at technology for security, some of the best security is in digital rights management. We

have new things coming out that can protect my song across the Internet so Richard can't copy it and share it with Tony. This is very interesting technology.

Yet it's not being applied or being required to apply to our personal information that is no different than the song.

So I'd like to ask the panel, where does standard of due care fit in and requirements for designing systems securely?

MR. SILVER: Who wants this one?

Go ahead.

MR. FELTEN: I believe pretty strongly that the approach you suggested of using digital rights management technology is the wrong way to go for privacy. The reason is that digital rights management technology, although it's loudly promoted, doesn't actually work very well, and it never has, and for fundamental reasons, I don't think it will. I think it's a mistake to think that we can rely on technology to keep someone who wants to use information maliciously from doing so.

I don't think technology is able to do that, and I think it's a mistake to try to use technology in that way. It's particularly a mistake to require people to do so. If we were to require that, we would be requiring people to use a technological approach that I

think is doomed to failure.

MR. SCHIAVONE: We're currently now at zero security on much consumer information and not ideal security on digital rights, but from the baseline to where we can get with privacy rights management and how there must be an audit trail for information sharing, it is just very far away from where both ends of the argument are.

MR. SCHWARTZ: Kathy gave a whole list of new technologies that are being built in exactly that area. I mean I don't think it's that far away. One thing that came up is the idea of a vocabulary and how we need a more robust vocabulary than we have today to make that happen, though.

MR. PURCELL: One last comment on this. One of the things that I'm concerned about here -- I'm here for the people.

We have a long and robust history of security specialization and training.

We have no history whatsoever for privacy specialization and training.

We'll hire just about anybody off the street and put them in charge of a database. One of the reasons system administrators aren't very good at their job is because there isn't a lot of training.

Neither is there a lot of hiring rigor that goes into that kind of personnel work and resources.

What I'm concerned about more than anything else is where are the credentials for the people that are handling this data?

We don't have a credentialing program that is very useful.

There's some for security. It's basic, but it's there, it’s something.

There's nothing for privacy.

One of the questions that I have is who is accountable?

And isn't, in some sense, the personnel department, the HR department, somewhat accountable for hiring people and training them, who actually have skills and experience and knowledge about what the hell they're doing, which I don't think is happening.

MR. PALLER: I think the safeguard program actually specifically requires that. They're not doing it, but we can start getting that.

MR. STANCO: Can I just make one comment? Because I think you brought up something that's terribly important, the standard of care.

I think this is a line of argument that will do wonders, because why don't we have a standard of care?

Why don't we hold companies to some kind of warranty?

It was fine when computers were just doing word processing, but when they are maintaining infrastructure, critical infrastructure, why is it that they don't have to give a warranty?

MR. PALLER: Don't you destroy the open source movement then? Because then there's nobody to sue.

MR. SCHWARTZ: No accountability.

MR. STANCO: No, I don't agree with that. What I was trying to say before is the government should make rules for everybody, then everybody rises and falls, and I think open source is going to do fine. It's a better model, in my opinion.

If it wasn't a better model, how could it possibly compete with billion-dollar companies when open source has no corporate structure, has no real structure except the Internet and a license, has no friends in high places, anyway, until recently, and still, it competes. Not only does it compete, the whole industry is going that way. In fact, it looks like UNIX is going to drop off and it's Microsoft versus open source -- or LINUX.

I'm not worried about how it will compete. My concern is I think we should have competition, I think we should have incentives as a set-up by the government. Then the government should really back off, and I think

open source has to create its organization. It's still in the formative stage, but once it does, I think it should give warranties, because I think people should be held accountable.

How can you possibly build an infrastructure that everybody in the whole world depends upon, and these people just are basically saying, well, don't look to us. That doesn't make any sense.

And if we do that, if we set up the standard of care, I think what happens eventually is you have metrics that will play into that, and more importantly, you'll have an insurance industry that can come into play and then really enforce.

MR. SILVER: Kathy?

MS. BOHRER: I want to address your original question a little bit.

I think technology can do a lot to really put into place something that tries to meet requirements for appropriate use of data, as long as the data is in the system. Of course, there's always a limitation, because at some point, the data goes outside of the system. It's displayed to some person. It's printed out. Some person sees it and now knows it.

And at that point, if there's misuse outside of the system, then you need accountability because –

MR. SCHIAVONE: But is there an audit trail to that?

MS. BOHRER: You can have audit trails. In fact, I thought that if you turn around some prophecies -- and data minimization is part of that but not the only thing you can imagine.

If you actually automate more, you could actually protect privacy more, because you could eliminate humans dealing with personal data to a larger degree.

So, for example, if I place an order, my address goes into a system. No person sees it. When the box with my order comes along the manufacturing line, some label gets printed out, it gets put on that, and it gets shipped to me. No person ever saw my address.

That's just one example that occurred to me today as I was thinking about this, but it is interesting.

There are limits, but there's still a lot we could do a lot better than we are today.

MR. SILVER: Next question. Please keep them concise.

QUESTION: Yes.

There were a number of references today to best practices, and I am a great fan of having people follow

best practices.

The trouble is, about four or five months ago, I was on a panel considering security technology for the health care industry, and two of the people on the panel were IT people from major health care providers, HMO's in California, as it turns out. I remember the debate I had with one of them, who wanted to know what are the best practices, and he capitalized the "B" and the "P", because from his point of view, HIPAA was the threat.

Attackers were not the threat. HIPAA was the threat. The danger to him was that his company would be sued. The danger to him personally was that he would be held responsible.

What he needed to know are the five simple things that he had to do called best practices such that, if he did these, then he was not legally responsible anymore.

So, if that's what we mean by best practices, I'm totally against it.

MR. NEUMANN: Ideally not. That's the lowest common denominator phenomenon, and that's clearly a disaster, but best practices themselves are useful. If you look at the generally accepted security principles that came out of our National Academy study from 1990, they're useful, but if they're not applied by people who

know what the hell they're doing and who have a set of meaningful requirements in the first place and who have an architecture for the system that they're developing that is evolvable and inter-operable and so on, then the best practices are inherently not very useful.

So, it's much more than best practices.

MR. SILVER: Next question.

AUSTIN HILL: There's been a lot of discussion about the marketplace for technologies for protecting consumers' information and I think, in the security area, we've had a long history of seeing this.

There's active threats, so it's a very easy, provable thing saying we're being threatened, so we need a firewall.

People got through the firewall, so now we need IDS, now we need patch management.

Companies can come in and say there's risk management, we have to spend so much to manage this risk of being attacked, and in the privacy side, if I look at the history of the privacy industry, which, I've been around a few years now, I haven't seen that evolve. A few years ago the FTC started announcing they were doing a great initiative, checking websites for policies. So, everyone threw up a policy.

All of a sudden you should have a CPO.

So, a whole bunch of CPO's were named, but generally they were lobbyists, to make sure no more privacy laws were assigned.

If you actually talk to CPO's about what's your budget, how many IT projects have you initiated, have you changed your database handling, it’s non-existent.

Same thing in Europe. This is by no means only a problem here.

Even in Europe, where legislation was passed and there was heavier legislation, without some enforcement or oversight into what companies actually are doing to change their practices, how they handle data -- that didn't exist until recently when we've seen it start happening. In the Netherlands, they've started doing spot checks on companies and reviewing their data handling practices, and in the last six months, we got more inquiries from the Netherlands than I have had from the United States for privacy management products.

When I start to look at the evolution of a marketplace, what exists to try and create that? We've seen safety belts, air bags. Those markets evolved because there were some standards set, there was some liability standard or regulation that said you have to be at least this safe, either through civil litigation or some other mechanism.

I just don't see that happening at all in privacy. So, generally, it becomes let's just put our head in the sand, put up a privacy web-page and hope no one calls or comes looking.

MR. NEUMANN: Austin, even though your question is very different from Carl's, my answer is exactly the same. It requires a great deal more than this litany of simplistic non-solutions.

It's a holistic problem. It requires an end-to-end solution.

It requires an understanding of architectures, software engineering, of having requirements that are meaningful in the first place, of submitting to some sort of evaluation process, of submitting to open review, perhaps, or at least having teams beating the hell out of your system, of understanding the privacy requirements before you go into building the system in the first place. There are no easy answers.

If you look on my website, you'll see lots of reports on how to build systems properly.

Nobody pays any attention to them, as far as I can make out.

MR. SILVER: I would add that the FTC Safeguards Rule went into effect recently, so please stay tuned.

And the last question, please.

QUESTION: Thank you for indulging me. I hope it's worth it.

Alan Wilcox. I work for the Vanguard Group.

I'd like to mention, also, that we don't have a CPO. We don't even have a CISO, because that spells N-o-t-h-i-n-g.

The regulations require a mature information security program, and that's what our goal is, to have a mature program.

I've got a comment and then a question.

Several comments have been raised that seem disparaging of overseas development. It's exactly the same criticism of foreign cars, when foreign cars were first being made. The issue is, if they can write code better than the processes and programs that we have in place, I welcome overseas development, if they have better checks and balances, if they have a more mature product development cycle.

Ultimately, American cars got a lot better, because we had a lot of Hondas and Toyotas around, and now we have a lot better GM's, Fords, and Chryslers. I think the same thing might bear out with overseas development.

Also, if you don't think foreign nationals are

already writing a lot of your software, you haven't been to a lot of software conferences.

I won't try to do my Indian accent impersonation.

Finally, how applications are being used is often completely left out of vendors' equations. Within my company, we see a lot of vendors saying, well, yes, here's a great database application. It has to run with elevated privileges. It has to run as the root user on your system.

Well, that's bogus. That's a practice that absolutely must not be tolerated.

Vendors should not have the ability to dictate the security environment of the customers. It goes the other way around.

Thanks.

MR. NEUMANN: That was a question. Very good question, actually.

MR. SILVER: Howard, go ahead.

MR. SCHMIDT: Just one really, really quick comment, and that's in reference to the comment on foreign nationals writing code.

The most severe intelligence threats against this country have been by born-and-bred U.S. citizens such as the FBI guy and Aldridge Ames and company, and

this has been an issue that pops up from time to time.

We have got phenomenal foreign nationals writing code, doing trustworthy things, doing good work. So, I wouldn't look at where they come from but look at the product they're putting out and the quality control and the engineering that goes into it.

MR. PURCELL: I would also comment on who writes code.

There may be an advantage to a less mature software industry emerging from another national sphere or geographic sphere. One thing that you might have heard today is that it may be the maturity of the process that's our biggest problem to overcome -- the Windows code bases, 10 million lines, 50 million lines, I don't know, some extraordinarily huge number of lines of code, which has been patched and cobbled together over a long, long period of time. It may be that one of the reasons that open source works well today competitively is because it doesn't have that maturity, because it is starting over again.

One thing that we don't do -- and nobody should ever think that this is happening -- is for most software that you're using, you don't sit down and write new requirements and write new software.

It's an adaptation of what's been written

before. The requirements are simply, okay, it didn't do this very well before, so make it do this now. So, it's re-jiggered for that, and then here's some new stuff it can do. It's kind of like your '57 Chevy spiffed up. So, I would be very careful to say that it may be the maturity of our industry that's something we have to overcome in many ways.

MR. NEUMANN: I would like to bring the foreign national argument back to my electronic voting machine. Suppose that the software and the systems were built by, say, the Russian mafia or the Bin Laden Research Institute. I think you would be very concerned about using those systems in your elections.

MR. PURCELL: No question. I would be very concerned.

But I would bet that, if they were built from scratch, that they worked very well according to the interests of the builder, right? And that is what I'm saying.

I'm not saying who should or should not build our code. What I am saying is very little of domestic code is actually being built from scratch.

MR. NEUMANN: My comment is also that you would never find the Trojan horses that they put in there.

MR. PURCELL: Right. I agree.

MR. SILVER: Well, it's getting to be about 5:30. How about a hand for our panelists?

(Applause.)

MR. SILVER: I also want to introduce my boss, who is here with some closing remarks. He's the director of the Division of Financial Practices, Joel Winston.

(Applause.)

CLOSING REMARKS

MR. WINSTON: I guess I get the final words, and I want to thank all of you hardy souls for sticking out the day. You're rewarded by having stayed here all day, now you get to go outside when it's not raining. So, congratulations.

I want to thank the panelists and the FTC staff for their thoughtful work and enlightening discussion today. This workshop had a different focus than the one last month, but in many respects, the lessons are the same -- that security technologies need to be easy to use, compatible with other systems, and applications, and built into the basic hardware and software consumers and businesses use.

In addition, the two workshops together have raised larger themes of how people, in general, can better use technology to protect sensitive information, whether they're engaging in commercial transactions or