Testimony by Herbert Lin

Senior Research Scholar, Center for International Security and Cooperation

Research Fellow, Hoover Institution

Stanford University

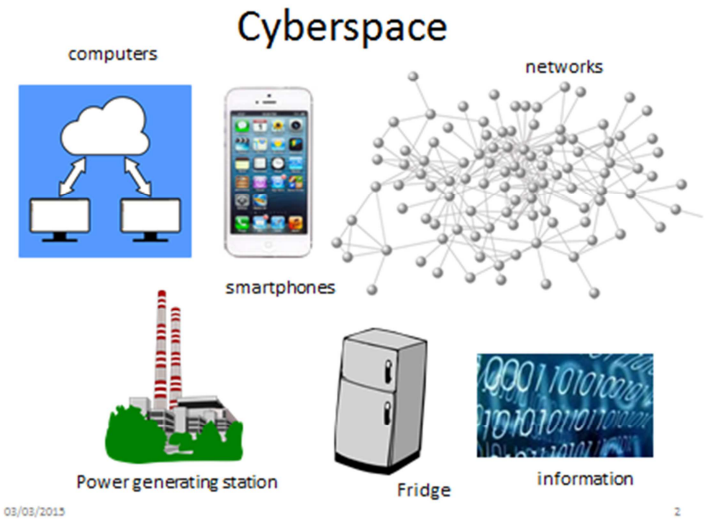
Chief Scientist (Emeritus), CSTB, National Research Council

House Committee on Energy and Commerce

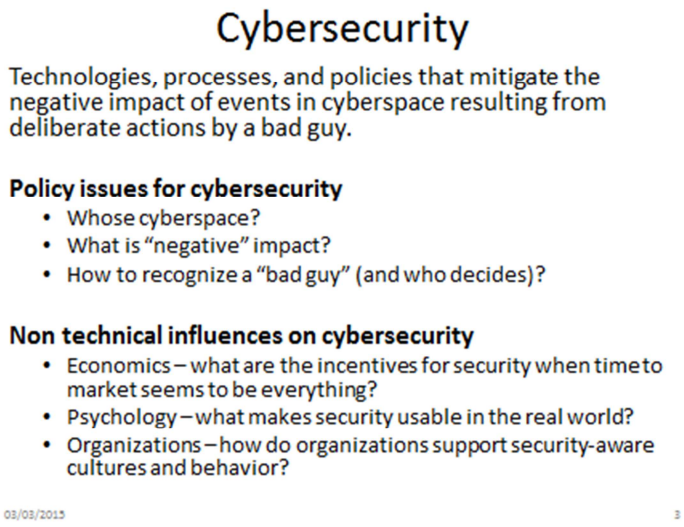
Subcommittee on Oversight and Investigations

March 3, 2015

Fundamental cybersecurity challenges to public policy



Cyberspace includes computers, networks including but not limited to the Internet, things connected to the Internet, and things with computers embedded inside them as well as the information that these technological artifacts use, store, handle, process, or transmit. And our society is becoming more and more dependent on cyberspace.



Cybersecurity

Technologies, processes, and policies that mitigate the negative impact of events in cyberspace resulting from deliberate actions by a bad guy.

Policy issues for cybersecurity

- Whose cyberspace?
- What is "negative" impact?
- How to recognize a "bad guy" (and who decides)?

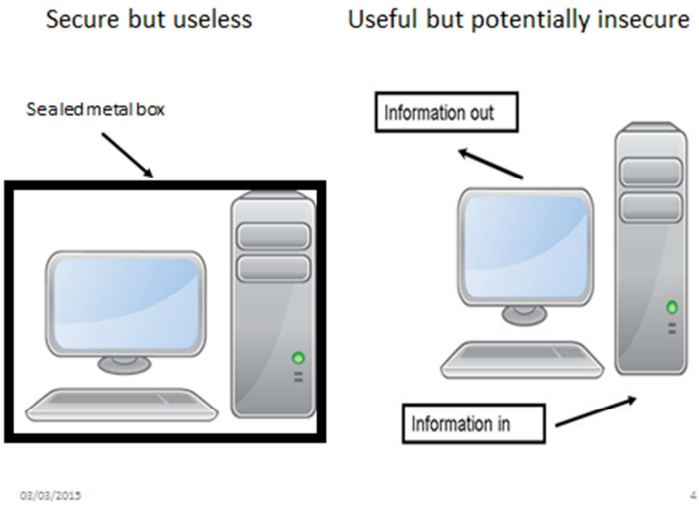
Non technical influences on cybersecurity

- Economics – what are the incentives for security when time to market seems to be everything?
- Psychology – what makes security usable in the real world?
- Organizations – how do organizations support security-aware cultures and behavior?

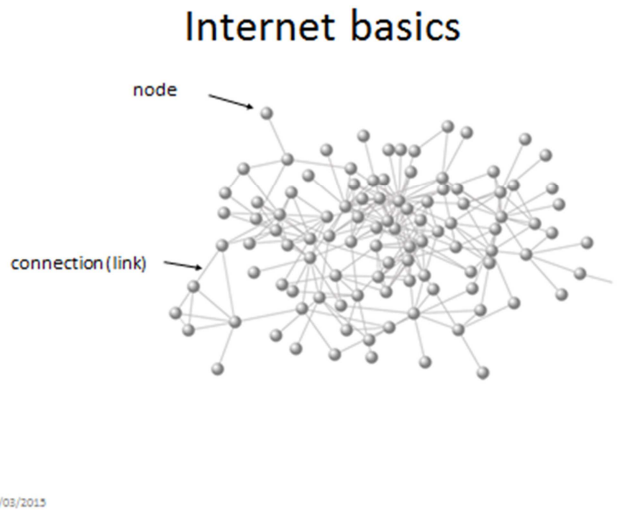
03/03/2015 3

Cybersecurity consists of technologies, processes, and policies that mitigate the negative impact of events in cyberspace resulting from deliberate actions by a bad guy. Note that this definition begs important questions, such as "whose cyberspace" (a company's? a nation's? an agency's?), what counts as "negative impact", and how we recognize a "bad guy"? All of these questions, of course, are policy questions rather than technical ones.

Also, cybersecurity is not just technology. Economic issues play out when vendors of products and services have to move very quickly in a very competitive market when time-to-market is everything in building a business and the imperative for speed precludes spending time on security. Psychology is apparent when you look at what makes security usable in the real world. For example, many passwords are easily guessed, and yet passwords have stuck around for decades even though we know how to do better. Why don't we? In large part, it's because these better methods are more of a hassle or cost more to use. Organizations and their cultures can shape behavior as well. For example, an organization that penalizes users for bad security behavior and one that rewards good security behavior are different, and the security posture of each organization may well be different.



Perfect security is possible only if you make the computer useless. We see a computer in a sealed metal box – you can't hack it, but you also can't use it for anything. Once you allow information (to include both data and programs) in, someone must make a judgment about what information counts as "good"—and that judgment is fallible, especially against a smart adversary. Computers generally can't do it as well as people can, and people do make mistakes in judgment as well.



The Internet can be regarded as a network of nodes. At each node is a computer or another network. Roughly, the Internet has been designed to have just one function—to make its best effort to transport information from A to B—and the technology does not care about the nature of that information. Furthermore, it is deliberately designed to put all of the useful functionality that you and I expect from computers at the end nodes. The internet was originally designed to be an unregulated marketplace in which anyone with a good idea could put up an application at some node with minimal regulatory burden—and this design principle is what enabled to the Internet to grow so fast in the past 30 years.

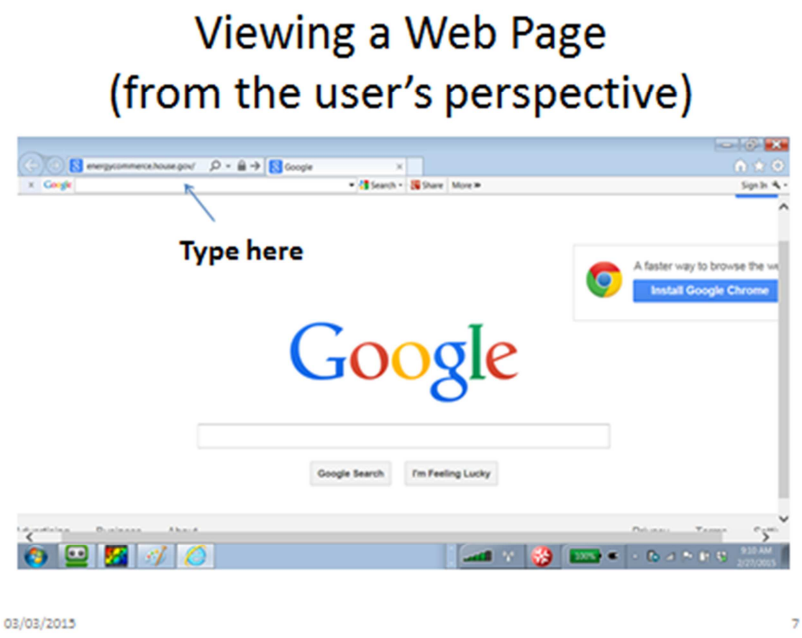
With this design, security issues must be handled at the end nodes rather than in the middle. One could, in principle, change the Internet's architecture to require that security issues per se to be handled internally, but this change would drastically reshape the nature of the Internet experience for those developing end-user applications, subjecting them to a far higher degree of interference with the traffic they want to send and receive, and likely reducing the freedom they have to innovate. Also, this change alone would not be likely to solve the entire cybersecurity problem, as it would not improve the security of the systems connected to the end nodes.

There are modest exceptions to the description provided above, but they do not change the basic story line.

Complexity is the enemy of cybersecurity. We ask a lot of our information technology, and to get that functionality, we have to integrate many different components. We put these components into a system, and when the system is complex enough, no one understands that system very well. And so even if the components are themselves individually secure (and they are not), the system may not in fact be secure.

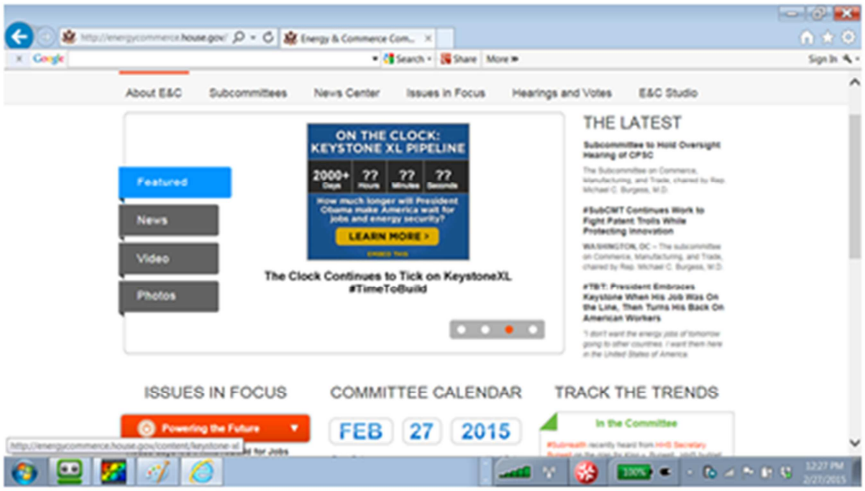
There's a sense in which it's a lot like crafting good legislation and regulation. For example, you know how hard it is to develop legislative language that covers every possible case. Often, new legislative language may interact with other legislative language already on the books, resulting in some surprising and undesirable outcomes. Then you need to pass even newer legislation to fix those problems—in the computer world, that's called a bug patch.

The next several slides provide an example of how complexity manifests itself.



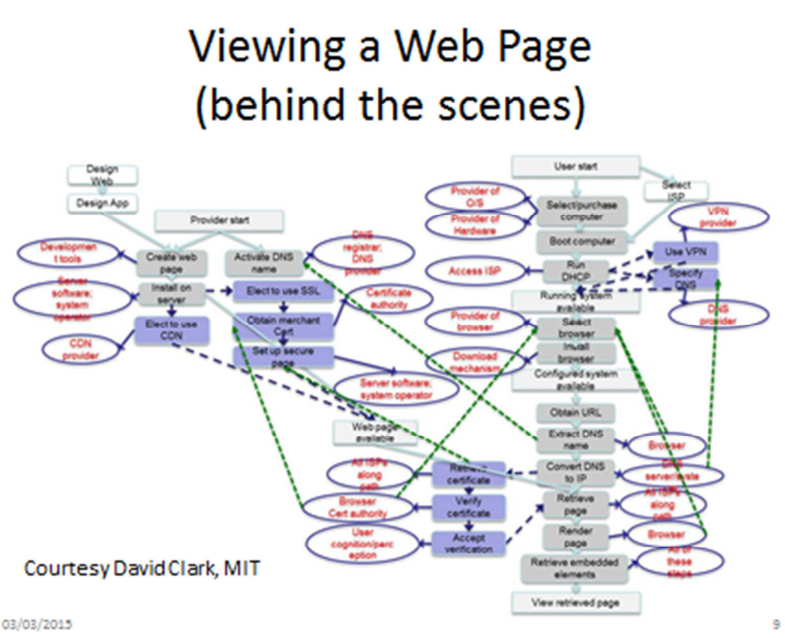
Here, we type in the name of a web page, and the page usually appears in the next slide in less than a second.

View Page Here

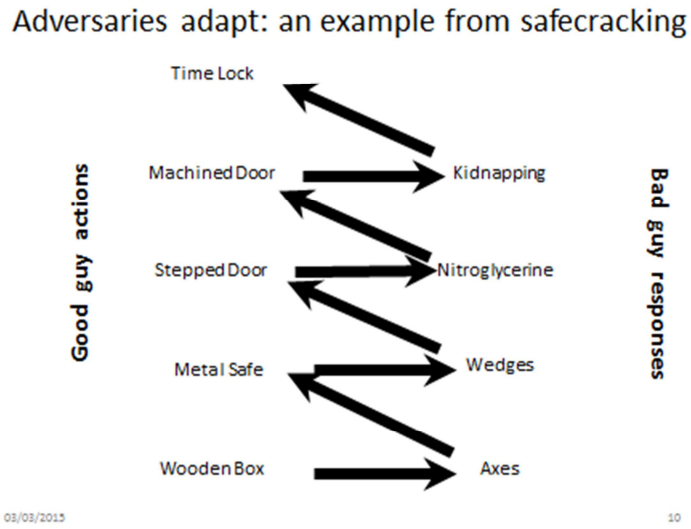


03/03/2015 8

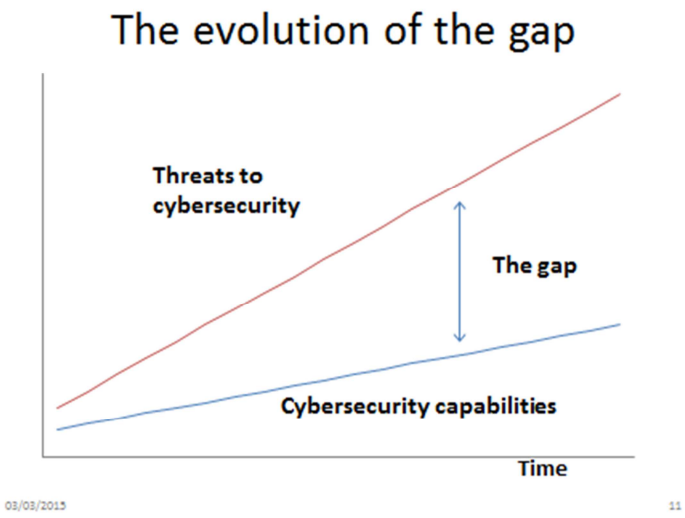
And here it is.



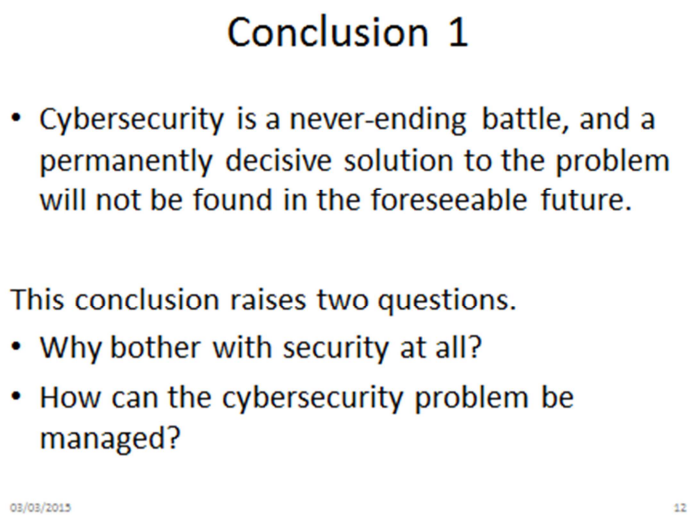
This is what is going on behind the scenes. It's not worth going over every one of these elements, but it's obvious that behind the scenes is a great deal of complexity. Every one of these rectangular boxes is a place where a bad guy can take a deliberate action to interfere with your web experience.



Also, adversaries adapt. They are smart, and the good guys don't get the last move. Indeed, there is no last move. Reading the slide from the bottom up, the good guy does what's on the left and the bad guy responds by doing what's on the right. And note that, in the physical world, we still haven't entirely stopped bank robberies – and now the bad guys have moved to the Internet to rob banks.

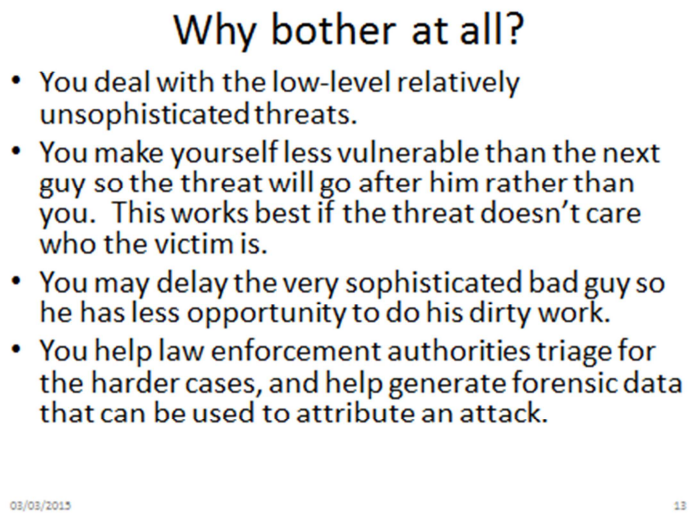


Here's the net result. It's true that over time we have gotten better at cybersecurity—that's the bottom line. But the top line – how much we depend on cyberspace – has grown even faster, and with that growing dependence the threats have grown commensurately.



Based on previous slides here's the first basic conclusion – Cybersecurity is a never-ending battle, and a permanently decisive solution to the problem will not be found in the foreseeable future. Thus, the public policy question is not how the cybersecurity problem can be solved, but rather how it can be managed at an acceptable cost in dollars and effort expended by the various stakeholder parties who have something to lose.

This conclusion leads to two important questions.



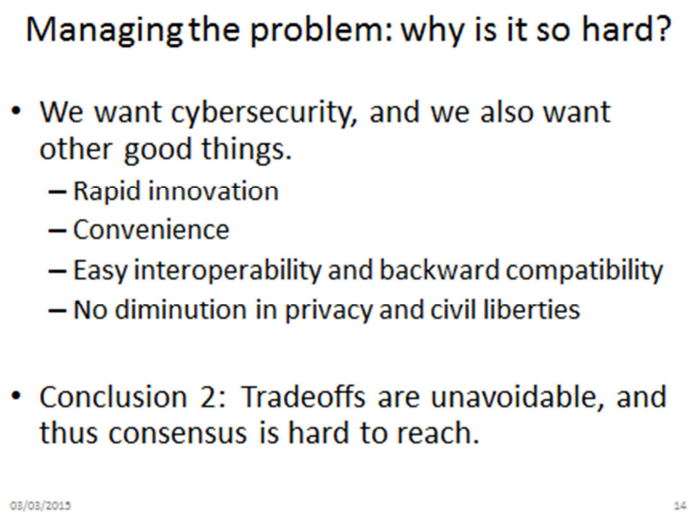
Given this conclusion, Slide 13 asks and answers the first question – why bother? If the good guys will never win decisively, what is the point? There are at least 4 reasons:

- You deal with the low-level relatively unsophisticated threats.

- You make yourself less vulnerable than the next guy so the threat will go after him rather than you. This works best if the threat doesn't care who the victim is. If he cares a lot, this won't work at all because he will try again and again. (A consequence of this point is that if the US government has information that can be obtained only from the government, the bad guys won't go elsewhere.)

- You delay the sophisticated bad guy so he has less opportunity to do his dirty work and make it more expensive for him, so he can do less of it.

- You help law enforcement authorities do triage for the harder cases, and you generate data that can help with forensics supporting attribution of an attack.



Question 2 is how we should manage cybersecurity as a public policy problem, and why is it so hard?

The fundamental reason is that we want cybersecurity, yes, but we also want many other things.

We want rapid innovation, and it's always easier and faster to do something without paying attention to security. And in a world in which being first to market has many economic advantages, it's entirely rational from a developer's point of view to ignore security at the start when the concept has yet to be proven. And once the concept has been proven, the right thing to do from a security standpoint is to start over again, this time integrating security into it.

But few people work like that. What they do, if they do anything, is to treat security as an add-on—and any design decisions they made with bad security consequences don't get fixed.

From a user standpoint, this is also rational behavior, at least in the short-term. They get a new application that does something useful for them. They don't face much of a threat because the application is new, and few hostile parties know about it, so the security environment is relatively benign.

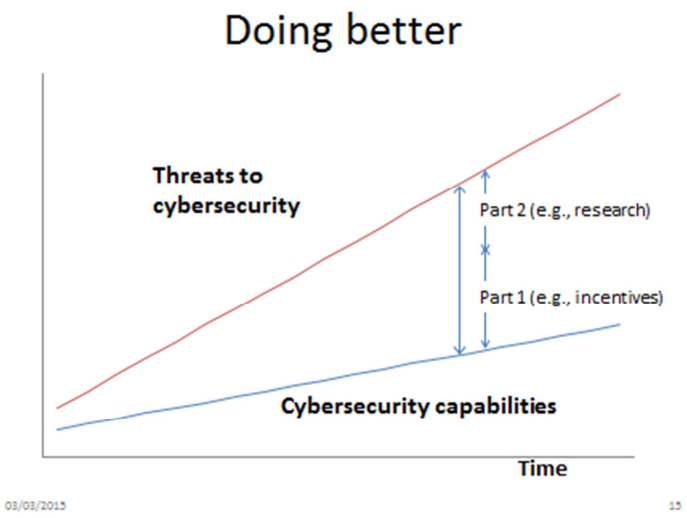
As the app grows in popularity, so do the incentives for hacking—and so the threat grows. But now the developer is *really* locked into his original design, because changing it now runs the risk of starting over again and losing his (large) customer base. So he is forced into a situation of patching – fixing problems as they appear.

Users also value convenience, and cybersecurity measures are the antithesis of convenience—mostly, cybersecurity gets in the way of doing useful work. How often have you been kept off your computer because you forgot a password? The whole point of security is to make your computer totally inaccessible to a bad guy trying to pretend he is you, and sometimes the automated mechanisms set up to differentiate you from a bad guy don't work so well.

The same is true about interoperability and backward compatibility. As you use an application, you become familiar with it. When you upgrade, you don't want to lose your investment in it, e.g., you don't want to be unable to use your old data files with the upgrade. But sometimes it happens that putting in backward compatibility means that you can't fix a known problem in the upgrade, because if you fix it, you'll break something else that depended on that problem being present.

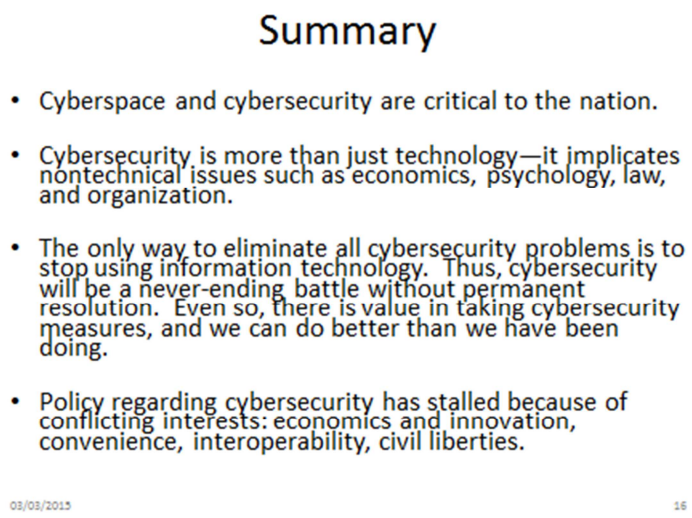
And we cherish our privacy and civil liberties as law-abiding Americans—but not for the bad guys. Again, sometimes it's hard to tell the difference, especially since smart bad guys try to look like law-abiding Americans. As we try to collect data that will help to identify bad guys in cyberspace, we sometimes gather data—inadvertently—on good guys. That is, we unintentionally violate their privacy rights and their civil liberties. This tradeoff is also unavoidable—we have to decide how much inadvertent violation we are willing to tolerate in order to gain whatever security benefits we are seeking, and we have no consensus on how far we're willing to go.

All of these examples lead to Conclusion 2: Tradeoffs are unavoidable, and thus the consensus needed to take action is hard to reach.



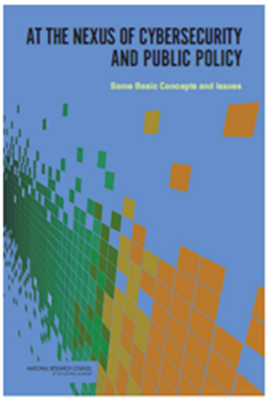
Reducing the gap is a two-part effort. Part 1 says we should reduce the gap between the average cybersecurity posture and the best possible cybersecurity posture. Part 1 is primarily nontechnical in nature, involving things like developing incentives to use known and better technologies and practices and applying already-known technical knowledge about cybersecurity.

Part 2, which we do simultaneously with Part 1 says we should reduce the gap between the strongest posture possible with known practices and technologies and the actual need. Part 2 is primarily technical, and involves developing new knowledge about cybersecurity.



This is the one page summary of my key points.

A useful reference



- David Clark, Tom Berson, Herbert Lin
- Released in final book form June 18, 2014
- Available free in PDF at www.nap.edu or for $ in hard copy

03/03/2015 17

A reference to be incorporated to the hearing record.