schneier.com

Essays: Click Here to Kill Everyone

Bruce Schneier

With the Internet of Things, we're building a world-size robot. How are we going to control it?

Last year, on October 21, your digital video recorder — or at least a DVR like yours — knocked Twitter off the internet. Someone used your DVR, along with millions of insecure webcams, routers, and other connected devices, to launch an attack that started a chain reaction, resulting in Twitter, Reddit, Netflix, and many sites going off the internet. You probably didn't realize that your DVR had that kind of power. But it does.

All computers are hackable. This has as much to do with the computer market as it does with the technologies. We prefer our software full of features and inexpensive, at the expense of security and reliability. That your computer can affect the security of Twitter is a market failure. The industry is filled with market failures that, until now, have been largely ignorable. As computers continue to permeate our homes, cars, businesses, these market failures will no longer be tolerable. Our only solution will be regulation, and that regulation will be foisted on us by a government desperate to "do something" in the face of disaster.

In this article I want to outline the problems, both technical and political, and point to some regulatory solutions. *Regulation* might be a dirty word in today's political climate, but security is the exception to our small-government bias. And as the threats posed by computers become greater and more catastrophic, regulation will be inevitable. So now's the time to start thinking about it.

We also need to reverse the trend to connect everything to the internet. And if we risk harm and even death, we need to think twice about what we connect and what we deliberately leave uncomputerized.

If we get this wrong, the computer industry will look like the pharmaceutical industry, or the aircraft industry. But if we get this right, we can maintain the innovative environment of the internet that has given us so much.

We no longer have things with computers embedded in them. We have computers with things attached to them.

Your modern refrigerator is a computer that keeps things cold. Your oven, similarly, is a computer that makes things hot. An ATM is a computer with money inside. Your car is no longer a mechanical device with some computers inside; it's a computer with four wheels and an engine. Actually, it's a distributed system of over 100 computers with four wheels and an engine. And, of course, your phones became full-power general-purpose computers in 2007, when the iPhone was introduced.

We wear computers: fitness trackers and computer-enabled medical devices — and, of course, we carry our smartphones everywhere. Our homes have smart thermostats, smart appliances, smart door locks, even smart light bulbs. At work, many of those same smart devices are networked together with CCTV cameras, sensors that detect customer movements, and everything else. Cities are starting to embed smart sensors in roads, streetlights, and sidewalk squares, also smart energy grids and smart transportation networks. A nuclear power plant is really just a computer that produces electricity, and — like everything else we've just listed — it's on the internet.

The internet is no longer a web that we connect to. Instead, it's a computerized, networked, and interconnected world that we live in. This is the future, and what we're calling the Internet of Things.

Broadly speaking, the Internet of Things has three parts. There are the sensors that collect data about us and our environment: smart thermostats, street and highway sensors, and those ubiquitous smartphones with their motion sensors and GPS location receivers. Then there are the "smarts" that figure out what the data means and what to do about it. This includes all the computer processors on these devices and — increasingly — in the cloud, as well as the memory that stores all of this information. And finally, there are the actuators that affect our environment. The point of a smart thermostat isn't to record the temperature; it's to control the furnace and the air conditioner. Driverless cars collect data about the road and the environment to steer themselves safely to their destinations.

You can think of the sensors as the eyes and ears of the internet. You can think of the actuators as the hands and feet of the internet. And you can think of the stuff in the middle as the brain. We are building an internet that senses, thinks, and acts.

This is the classic definition of a robot. We're building a world-size robot, and we don't

even realize it.

To be sure, it's not a robot in the classical sense. We think of robots as discrete autonomous entities, with sensors, brain, and actuators all together in a metal shell. The world-size robot is distributed. It doesn't have a singular body, and parts of it are controlled in different ways by different people. It doesn't have a central brain, and it has nothing even remotely resembling a consciousness. It doesn't have a single goal or focus. It's not even something we deliberately designed. It's something we have inadvertently built out of the everyday objects we live with and take for granted. It is the extension of our computers and networks into the real world.

This world-size robot is actually more than the Internet of Things. It's a combination of several decades-old computing trends: mobile computing, cloud computing, always-on computing, huge databases of personal information, the Internet of Things — or, more precisely, cyber-physical systems — autonomy, and artificial intelligence. And while it's still not very smart, it'll get smarter. It'll get more powerful and more capable through all the interconnections we're building.

It'll also get much more dangerous.

Computer security has been around for almost as long as computers have been. And while it's true that security wasn't part of the design of the original internet, it's something we have been trying to achieve since its beginning.

I have been working in computer security for over 30 years: first in cryptography, then more generally in computer and network security, and now in general security technology. I have watched computers become ubiquitous, and have seen firsthand the problems — and solutions — of securing these complex machines and systems. I'm telling you all this because what used to be a specialized area of expertise now affects everything. Computer security is now everything security. There's one critical difference, though: The threats have become greater.

Traditionally, computer security is divided into three categories: confidentiality, integrity, and availability. For the most part, our security concerns have largely centered around confidentiality. We're concerned about our data and who has access to it — the world of privacy and surveillance, of data theft and misuse.

But threats come in many forms. Availability threats: computer viruses that delete our

data, or ransomware that encrypts our data and demands payment for the unlock key. Integrity threats: hackers who can manipulate data entries can do things ranging from changing grades in a class to changing the amount of money in bank accounts. Some of these threats are pretty bad. Hospitals have paid tens of thousands of dollars to criminals whose ransomware encrypted critical medical files. JPMorgan Chase spends half a billion on cybersecurity a year.

Today, the integrity and availability threats are much worse than the confidentiality threats. Once computers start affecting the world in a direct and physical manner, there are real risks to life and property. There is a fundamental difference between crashing your computer and losing your spreadsheet data, and crashing your pacemaker and losing your life. This isn't hyperbole; recently researchers found serious security vulnerabilities in St. Jude Medical's implantable heart devices. Give the internet hands and feet, and it will have the ability to punch and kick.

Take a concrete example: modern cars, those computers on wheels. The steering wheel no longer turns the axles, nor does the accelerator pedal change the speed. Every move you make in a car is processed by a computer, which does the actual controlling. A central computer controls the dashboard. There's another in the radio. The engine has 20 or so computers. These are all networked, and increasingly autonomous.

Now, let's start listing the security threats. We don't want car navigation systems to be used for mass surveillance, or the microphone for mass eavesdropping. We might want it to be used to determine a car's location in the event of a 911 call, and possibly to collect information about highway congestion. We don't want people to hack their own cars to bypass emissions-control limitations. We don't want manufacturers or dealers to be able to do that, either, as Volkswagen did for years. We can imagine wanting to give police the ability to remotely and safely disable a moving car; that would make high-speed chases a thing of the past. But we definitely don't want hackers to be able to do that. We definitely don't want them disabling the brakes in every car without warning, at speed. As we make the transition from driver-controlled cars to cars with various driver-assist capabilities to fully driverless cars, we don't want any of those critical components subverted. We don't want someone to be able to accidentally crash your car, let alone do it on purpose. And equally, we don't want them to be able to manipulate the navigation software to change your route, or the door-lock controls to prevent you from opening the door. I could go on.

That's a lot of different security requirements, and the effects of getting them wrong range from illegal surveillance to extortion by ransomware to mass death.

Our computers and smartphones are as secure as they are because companies like Microsoft, Apple, and Google spend a lot of time testing their code before it's released, and quickly patch vulnerabilities when they're discovered. Those companies can support large, dedicated teams because those companies make a huge amount of money, either directly or indirectly, from their software — and, in part, compete on its security. Unfortunately, this isn't true of embedded systems like digital video recorders or home routers. Those systems are sold at a much lower margin, and are often built by offshore third parties. The companies involved simply don't have the expertise to make them secure.

At a recent hacker conference, a security researcher analyzed 30 home routers and was able to break into half of them, including some of the most popular and common brands. The denial-of-service attacks that forced popular websites like Reddit and Twitter off the internet last October were enabled by vulnerabilities in devices like webcams and digital video recorders. In August, two security researchers demonstrated a ransomware attack on a smart thermostat.

Even worse, most of these devices don't have any way to be patched. Companies like Microsoft and Apple continuously deliver security patches to your computers. Some home routers are technically patchable, but in a complicated way that only an expert would attempt. And the only way for you to update the firmware in your hackable DVR is to throw it away and buy a new one.

The market can't fix this because neither the buyer nor the seller cares. The owners of the webcams and DVRs used in the denial-of-service attacks don't care. Their devices were cheap to buy, they still work, and they don't know any of the victims of the attacks. The sellers of those devices don't care: They're now selling newer and better models, and the original buyers only cared about price and features. There is no market solution, because the insecurity is what economists call an externality: It's an effect of the purchasing decision that affects other people. Think of it kind of like invisible pollution.

Security is an arms race between attacker and defender. Technology perturbs that arms race by changing the balance between attacker and defender. Understanding how this arms race has unfolded on the internet is essential to understanding why the world-size robot we're building is so insecure, and how we might secure it. To that end, I have five truisms, born from what we've already learned about computer and internet security. They will soon affect the security arms race everywhere.

Truism No. 1: On the internet, attack is easier than defense.

There are many reasons for this, but the most important is the complexity of these systems. More complexity means more people involved, more parts, more interactions, more mistakes in the design and development process, more of everything where hidden insecurities can be found. Computer-security experts like to speak about the attack surface of a system: all the possible points an attacker might target and that must be secured. A complex system means a large attack surface. The defender has to secure the entire attack surface. The attacker just has to find one vulnerability — one unsecured avenue for attack — and gets to choose how and when to attack. It's simply not a fair battle.

There are other, more general, reasons why attack is easier than defense. Attackers have a natural agility that defenders often lack. They don't have to worry about laws, and often not about morals or ethics. They don't have a bureaucracy to contend with, and can more quickly make use of technical innovations. Attackers also have a first-mover advantage. As a society, we're generally terrible at proactive security; we rarely take preventive security measures until an attack actually happens. So more advantages go to the attacker.

Truism No. 2: Most software is poorly written and insecure.

If complexity isn't enough, we compound the problem by producing lousy software. Well-written software, like the kind found in airplane avionics, is both expensive and time-consuming to produce. We don't want that. For the most part, poorly written software has been good enough. We'd all rather live with buggy software than pay the prices good software would require. We don't mind if our games crash regularly, or our business applications act weird once in a while. Because software has been largely benign, it hasn't mattered. This has permeated the industry at all levels. At universities, we don't teach how to code well. Companies don't reward quality code in the same way they reward fast and cheap. And we consumers don't demand it.

But poorly written software is riddled with bugs, sometimes as many as one per 1,000 lines of code. Some of them are inherent in the complexity of the software, but most are programming mistakes. Not all bugs are vulnerabilities, but some are.

Truism No. 3: Connecting everything to each other via the internet will expose new vulnerabilities.

The more we network things together, the more vulnerabilities on one thing will affect other things. On October 21, vulnerabilities in a wide variety of embedded devices were

all harnessed together to create what hackers call a botnet. This botnet was used to launch a distributed denial-of-service attack against a company called Dyn. Dyn provided a critical internet function for many major internet sites. So when Dyn went down, so did all those popular websites.

These chains of vulnerabilities are everywhere. In 2012, journalist Mat Honan suffered a massive personal hack because of one of them. A vulnerability in his Amazon account allowed hackers to get into his Apple account, which allowed them to get into his Gmail account. And in 2013, the Target Corporation was hacked by someone stealing credentials from its HVAC contractor.

Vulnerabilities like these are particularly hard to fix, because no one system might actually be at fault. It might be the insecure interaction of two individually secure systems.

Truism No. 4: Everybody has to stop the best attackers in the world.

One of the most powerful properties of the internet is that it allows things to scale. This is true for our ability to access data or control systems or do any of the cool things we use the internet for, but it's also true for attacks. In general, fewer attackers can do more damage because of better technology. It's not just that these modern attackers are more efficient, it's that the internet allows attacks to scale to a degree impossible without computers and networks.

This is fundamentally different from what we're used to. When securing my home against burglars, I am only worried about the burglars who live close enough to my home to consider robbing me. The internet is different. When I think about the security of my network, I have to be concerned about the best attacker possible, because he's the one who's going to create the attack tool that everyone else will use. The attacker that discovered the vulnerability used to attack Dyn released the code to the world, and within a week there were a dozen attack tools using it.

Truism No. 5: Laws inhibit security research.

The Digital Millennium Copyright Act is a terrible law that fails at its purpose of preventing widespread piracy of movies and music. To make matters worse, it contains a provision that has critical side effects. According to the law, it is a crime to bypass security mechanisms that protect copyrighted work, even if that bypassing would otherwise be legal. Since all software can be copyrighted, it is arguably illegal to do

security research on these devices and to publish the result.

Although the exact contours of the law are arguable, many companies are using this provision of the DMCA to threaten researchers who expose vulnerabilities in their embedded systems. This instills fear in researchers, and has a chilling effect on research, which means two things: (1) Vendors of these devices are more likely to leave them insecure, because no one will notice and they won't be penalized in the market, and (2) security engineers don't learn how to do security better.

Unfortunately, companies generally like the DMCA. The provisions against reverse-engineering spare them the embarrassment of having their shoddy security exposed. It also allows them to build proprietary systems that lock out competition. (This is an important one. Right now, your toaster cannot force you to only buy a particular brand of bread. But because of this law and an embedded computer, your Keurig coffee maker can force you to buy a particular brand of coffee.)

In general, there are two basic paradigms of security. We can either try to secure something well the first time, or we can make our security agile. The first paradigm comes from the world of dangerous things: from planes, medical devices, buildings. It's the paradigm that gives us secure design and secure engineering, security testing and certifications, professional licensing, detailed preplanning and complex government approvals, and long times-to-market. It's security for a world where getting it right is paramount because getting it wrong means people dying.

The second paradigm comes from the fast-moving and heretofore largely benign world of software. In this paradigm, we have rapid prototyping, on-the-fly updates, and continual improvement. In this paradigm, new vulnerabilities are discovered all the time and security disasters regularly happen. Here, we stress survivability, recoverability, mitigation, adaptability, and muddling through. This is security for a world where getting it wrong is okay, as long as you can respond fast enough.

These two worlds are colliding. They're colliding in our cars — literally — in our medical devices, our building control systems, our traffic control systems, and our voting machines. And although these paradigms are wildly different and largely incompatible, we need to figure out how to make them work together.

So far, we haven't done very well. We still largely rely on the first paradigm for the dangerous computers in cars, airplanes, and medical devices. As a result, there are

medical systems that can't have security patches installed because that would invalidate their government approval. In 2015, Chrysler recalled 1.4 million cars to fix a software vulnerability. In September 2016, Tesla remotely sent a security patch to all of its Model S cars overnight. Tesla sure sounds like it's doing things right, but what vulnerabilities does this remote patch feature open up?

Until now we've largely left computer security to the market. Because the computer and network products we buy and use are so lousy, an enormous after-market industry in computer security has emerged. Governments, companies, and people buy the security they think they need to secure themselves. We've muddled through well enough, but the market failures inherent in trying to secure this world-size robot will soon become too big to ignore.

Markets alone can't solve our security problems. Markets are motivated by profit and short-term goals at the expense of society. They can't solve collective-action problems. They won't be able to deal with economic externalities, like the vulnerabilities in DVRs that resulted in Twitter going offline. And we need a counterbalancing force to corporate power.

This all points to policy. While the details of any computer-security system are technical, getting the technologies broadly deployed is a problem that spans law, economics, psychology, and sociology. And getting the policy right is just as important as getting the technology right because, for internet security to work, law and technology have to work together. This is probably the most important lesson of Edward Snowden's NSA disclosures. We already knew that technology can subvert law. Snowden demonstrated that law can also subvert technology. Both fail unless each work. It's not enough to just let technology do its thing.

Any policy changes to secure this world-size robot will mean significant government regulation. I know it's a sullied concept in today's world, but I don't see any other possible solution. It's going to be especially difficult on the internet, where its permissionless nature is one of the best things about it and the underpinning of its most world-changing innovations. But I don't see how that can continue when the internet can affect the world in a direct and physical manner.

I have a proposal: a new government regulatory agency. Before dismissing it out of hand, please hear me out.

We have a practical problem when it comes to internet regulation. There's no government structure to tackle this at a systemic level. Instead, there's a fundamental mismatch between the way government works and the way this technology works that makes dealing with this problem impossible at the moment.

Government operates in silos. In the U.S., the FAA regulates aircraft. The NHTSA regulates cars. The FDA regulates medical devices. The FCC regulates communications devices. The FTC protects consumers in the face of "unfair" or "deceptive" trade practices. Even worse, who regulates data can depend on how it is used. If data is used to influence a voter, it's the Federal Election Commission's jurisdiction. If that same data is used to influence a consumer, it's the FTC's. Use those same technologies in a school, and the Department of Education is now in charge. Robotics will have its own set of problems, and no one is sure how that is going to be regulated. Each agency has a different approach and different rules. They have no expertise in these new issues, and they are not quick to expand their authority for all sorts of reasons.

Compare that with the internet. The internet is a freewheeling system of integrated objects and networks. It grows horizontally, demolishing old technological barriers so that people and systems that never previously communicated now can. Already, apps on a smartphone can log health information, control your energy use, and communicate with your car. That's a set of functions that crosses jurisdictions of at least four different government agencies, and it's only going to get worse.

Our world-size robot needs to be viewed as a single entity with millions of components interacting with each other. Any solutions here need to be holistic. They need to work everywhere, for everything. Whether we're talking about cars, drones, or phones, they're all computers.

This has lots of precedent. Many new technologies have led to the formation of new government regulatory agencies. Trains did, cars did, airplanes did. Radio led to the formation of the Federal Radio Commission, which became the FCC. Nuclear power led to the formation of the Atomic Energy Commission, which eventually became the Department of Energy. The reasons were the same in every case. New technologies need new expertise because they bring with them new challenges. Governments need a single agency to house that new expertise, because its applications cut across several preexisting agencies. It's less that the new agency needs to regulate — although that's often a big part of it — and more that governments recognize the importance of the new technologies.

The internet has famously eschewed formal regulation, instead adopting a multi-

stakeholder model of academics, businesses, governments, and other interested parties. My hope is that we can keep the best of this approach in any regulatory agency, looking more at the new U.S. Digital Service or the 18F office inside the General Services Administration. Both of those organizations are dedicated to providing digital government services, and both have collected significant expertise by bringing people in from outside of government, and both have learned how to work closely with existing agencies. Any internet regulatory agency will similarly need to engage in a high level of collaborate regulation — both a challenge and an opportunity.

I don't think any of us can predict the totality of the regulations we need to ensure the safety of this world, but here's a few. We need government to ensure companies follow good security practices: testing, patching, secure defaults — and we need to be able to hold companies liable when they fail to do these things. We need government to mandate strong personal data protections, and limitations on data collection and use. We need to ensure that responsible security research is legal and well-funded. We need to enforce transparency in design, some sort of code escrow in case a company goes out of business, and interoperability between devices of different manufacturers, to counterbalance the monopolistic effects of interconnected technologies. Individuals need the right to take their data with them. And internet-enabled devices should retain some minimal functionality if disconnected from the internet

I'm not the only one talking about this. I've seen proposals for a National Institutes of Health analog for cybersecurity. University of Washington law professor Ryan Calo has proposed a Federal Robotics Commission. I think it needs to be broader: maybe a Department of Technology Policy.

Of course there will be problems. There's a lack of expertise in these issues inside government. There's a lack of willingness in government to do the hard regulatory work. Industry is worried about any new bureaucracy: both that it will stifle innovation by regulating too much and that it will be captured by industry and regulate too little. A domestic regulatory agency will have to deal with the fundamentally international nature of the problem.

But government is the entity we use to solve problems like this. Governments have the scope, scale, and balance of interests to address the problems. It's the institution we've built to adjudicate competing social interests and internalize market externalities. Left to their own devices, the market simply can't. That we're currently in the middle of an era of low government trust, where many of us can't imagine government doing anything positive in an area like this, is to our detriment.

Here's the thing: Governments will get involved, regardless. The risks are too great, and the stakes are too high. Government already regulates dangerous physical systems like cars and medical devices. And nothing motivates the U.S. government like fear. Remember 2001? A nominally small-government Republican president created the Office of Homeland Security 11 days after the terrorist attacks: a rushed and ill-thought-out decision that we've been trying to fix for over a decade. A fatal disaster will similarly spur our government into action, and it's unlikely to be well-considered and thoughtful action. Our choice isn't between government involvement and no government involvement. Our choice is between smarter government involvement and stupider government involvement. We have to start thinking about this now. Regulations are necessary, important, and complex; and they're coming. We can't afford to ignore these issues until it's too late.

We also need to start disconnecting systems. If we cannot secure complex systems to the level required by their real-world capabilities, then we must not build a world where everything is computerized and interconnected.

There are other models. We can enable local communications only. We can set limits on collected and stored data. We can deliberately design systems that don't interoperate with each other. We can deliberately fetter devices, reversing the current trend of turning everything into a general-purpose computer. And, most important, we can move toward less centralization and more distributed systems, which is how the internet was first envisioned.

This might be a heresy in today's race to network everything, but large, centralized systems are not inevitable. The technical elites are pushing us in that direction, but they really don't have any good supporting arguments other than the profits of their ever-growing multinational corporations.

But this will change. It will change not only because of security concerns, it will also change because of political concerns. We're starting to chafe under the worldview of everything producing data about us and what we do, and that data being available to both governments and corporations. Surveillance capitalism won't be the business model of the internet forever. We need to change the fabric of the internet so that evil governments don't have the tools to create a horrific totalitarian state. And while good laws and regulations in Western democracies are a great second line of defense, they can't be our only line of defense.

My guess is that we will soon reach a high-water mark of computerization and connectivity, and that afterward we will make conscious decisions about what and how we decide to interconnect. But we're still in the honeymoon phase of connectivity. Governments and corporations are punch-drunk on our data, and the rush to connect everything is driven by an even greater desire for power and market share. One of the presentations released by Edward Snowden contained the NSA mantra: "Collect it all." A similar mantra for the internet today might be: "Connect it all."

The inevitable backlash will not be driven by the market. It will be deliberate policy decisions that put the safety and welfare of society above individual corporations and industries. It will be deliberate policy decisions that prioritize the security of our systems over the demands of the FBI to weaken them in order to make their law-enforcement jobs easier. It'll be hard policy for many to swallow, but our safety will depend on it.

The scenarios I've outlined, both the technological and economic trends that are causing them and the political changes we need to make to start to fix them, come from my years of working in internet-security technology and policy. All of this is informed by an understanding of both technology and policy. That turns out to be critical, and there aren't enough people who understand both.

This brings me to my final plea: We need more public-interest technologists.

Over the past couple of decades, we've seen examples of getting internet-security policy badly wrong. I'm thinking of the FBI's "going dark" debate about its insistence that computer devices be designed to facilitate government access, the "vulnerability equities process" about when the government should disclose and fix a vulnerability versus when it should use it to attack other systems, the debacle over paperless touch-screen voting machines, and the DMCA that I discussed above. If you watched any of these policy debates unfold, you saw policy-makers and technologists talking past each other.

Our world-size robot will exacerbate these problems. The historical divide between Washington and Silicon Valley — the mistrust of governments by tech companies and the mistrust of tech companies by governments — is dangerous.

We have to fix this. Getting IoT security right depends on the two sides working together and, even more important, having people who are experts in each working on both. We need technologists to get involved in policy, and we need policy-makers to get involved in technology. We need people who are experts in making both technology and

technological policy. We need technologists on congressional staffs, inside federal agencies, working for NGOs, and as part of the press. We need to create a viable career path for public-interest technologists, much as there already is one for public-interest attorneys. We need courses, and degree programs in colleges, for people interested in careers in public-interest technology. We need fellowships in organizations that need these people. We need technology companies to offer sabbaticals for technologists wanting to go down this path. We need an entire ecosystem that supports people bridging the gap between technology and law. We need a viable career path that ensures that even though people in this field won't make as much as they would in a high-tech start-up, they will have viable careers. The security of our computerized and networked future — meaning the security of ourselves, families, homes, businesses, and communities — depends on it.

This plea is bigger than security, actually. Pretty much all of the major policy debates of this century will have a major technological component. Whether it's weapons of mass destruction, robots drastically affecting employment, climate change, food safety, or the increasing ubiquity of ever-shrinking drones, understanding the policy means understanding the technology. Our society desperately needs technologists working on the policy. The alternative is bad policy.

The world-size robot is less designed than created. It's coming without any forethought or architecting or planning; most of us are completely unaware of what we're building. In fact, I am not convinced we can actually design any of this. When we try to design complex sociotechnical systems like this, we are regularly surprised by their emergent properties. The best we can do is observe and channel these properties as best we can.

Market thinking sometimes makes us lose sight of the human choices and autonomy at stake. Before we get controlled — or killed — by the world-size robot, we need to rebuild confidence in our collective governance institutions. Law and policy may not seem as cool as digital tech, but they're also places of critical innovation. They're where we collectively bring about the world we want to live in.

While I might sound like a Cassandra, I'm actually optimistic about our future. Our society has tackled bigger problems than this one. It takes work and it's not easy, but we eventually find our way clear to make the hard choices necessary to solve our real problems.

The world-size robot we're building can only be managed responsibly if we start making

real choices about the interconnected world we live in. Yes, we need security systems as robust as the threat landscape. But we also need laws that effectively regulate these dangerous technologies. And, more generally, we need to make moral, ethical, and political decisions on how those systems should work. Until now, we've largely left the internet alone. We gave programmers a special right to code cyberspace as they saw fit. This was okay because cyberspace was separate and relatively unimportant: That is, it didn't matter. Now that that's changed, we can no longer give programmers and the companies they work for this power. Those moral, ethical, and political decisions need, somehow, to be made by everybody. We need to link people with the same zeal that we are currently linking machines. "Connect it all" must be countered with "connect us all."

Categories: Computer and Information Security

Tags: New York Magazine