



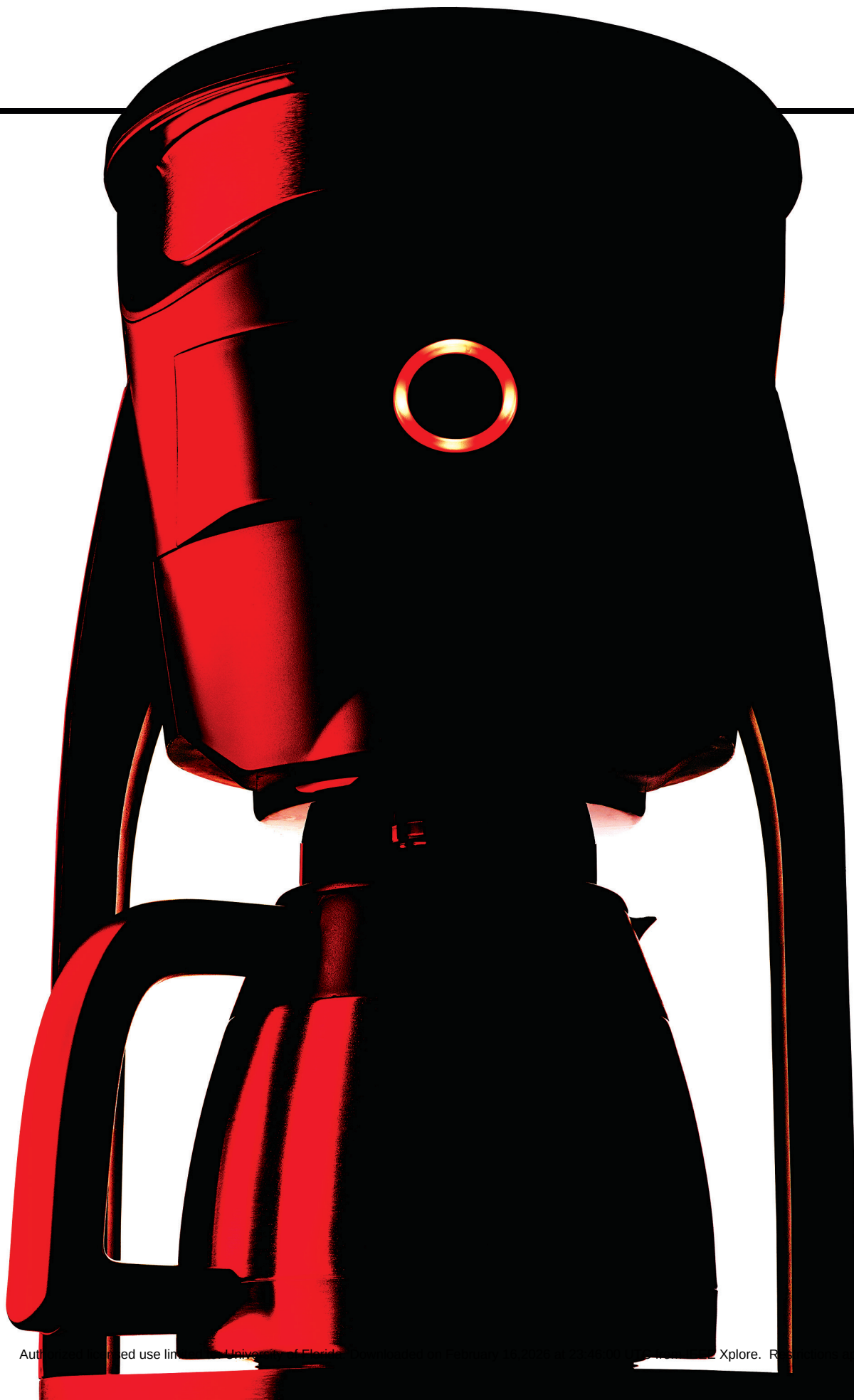
© By Sandip Ray, Abhishek Basak & Swarup Bhunia

Flexible hardware
will protect the
world's network of
smart devices

The Patchable Internet of Things /



/ 30 | NOV 2017 | NORTH AMERICAN | SPECTRUM.IEEE.ORG



ON 21 OCTOBER OF LAST YEAR,

a variety of major websites—including those of Twitter, PayPal, Spotify, Netflix, *The New York Times*, and *The Wall Street Journal*—stopped working. The cause was a distributed denial-of-service attack, not on these websites themselves but on the provider they and many others used to support the Domain Name System, or DNS, which translates the name of the site into its numerical address on the Internet. The DNS provider in this case was a company called Dyn, whose servers were barraged by so many fake requests for DNS lookups that they couldn't answer the real ones.

Distributed denial-of-service attacks are common enough. But two things made this attack special. First, it hobbled a large DNS provider, so it disrupted many different websites. Also, the fake requests didn't come from the usual botnet of compromised desktop and laptop computers. Rather, the attack was orchestrated through tens of millions of small, connected devices, things like Internet-connected cameras and home routers—components of what is often called the Internet of Things, or IoT for short.

For several years now, the number of things connected to the Internet—including phones, smart watches, fitness trackers, home thermostats, and various sensors—has exceeded the human population. By 2020, there will be tens of billions of such gadgets online. The burgeoning size of the Internet of Things reflects the fastest economic growth ever experienced for any sector in the history of human civilization.

For the most part, this development promises great excitement and opportunity for engineers and society at large. But there is a dark cloud hanging over the IoT: the concomitant threats to security and privacy, which will be of a scale never experienced before.

Our digital systems are vulnerable to malicious hackers attempting to gain unauthorized access, steal personal data

and other information, hold the information they steal for ransom, and even bring systems down completely, as happened with the attack on Dyn. The result is an ongoing arms race between hackers and computer-security experts, forcing the rest of us to live on a treadmill of security updates to the software we run on our various computers.

The current paradigm—a cat-and-mouse game of increasingly sophisticated hacks and software patches—presents a particularly thorny challenge for the Internet of Things. One reason is that security attacks on the IoT can have catastrophic consequences for our power grids, water supplies, and hospitals, to name just a few pieces of critical infrastructure that are vulnerable. The other reason for worry is that mass-produced smart devices may simply not have hardware capable of being programmed to resist all the threats that will arise in their lifetimes. These realities cast into doubt whether we are indeed ready for the regime of pervasive, ubiquitous computing devices. And yet it's upon us.

Here we explore what might be a way out of this predicament. In a nutshell, we propose that the various gizmos making up the IoT should be built so that their very hardware can adapt to future security threats. That won't be easy to engineer, but in our view, it's the smart way to design smart devices.

WHY ARE IOT DEVICES SO VULNERABLE TO HACKING?

One obvious reason is just their numbers. With billions of devices around, there will always be a multitude—many millions—that behave maliciously or are otherwise compromised. And each compromised device connected to the Internet may attempt to infect many others. So on the Internet of Things, the barrage of attacks will be vast and relentless.

The personalization factor is another reason to expect that security vulnerabilities of these devices will have especially catastrophic results. We now have small digital systems that track and record many of our daily activities: our sleep, our contacts with one another, our health care measures, our browsing patterns, and so forth. The informa-

tion from these devices is typically communicated through the Internet to central repositories and servers for storage and analysis, and an adversary breaking into that communication at any point can access some of your most intimate personal information.

Still another concern with attacks on these devices pertains to their interaction with the physical world. A smart toaster at home or sensors in a factory, when hacked, can lead to disastrous consequences that affect the machines being controlled.

Traditional mechanisms used to keep computers secure will probably not be adequate. For one thing, most of those protections—designed for laptops, desktops, servers, and even phones—consume significant amounts of power. They won't work for a tiny device such as a watch or sensor node, which must operate using very little energy.

What's more, protection mechanisms are typically designed for computing systems that remain in operation for only a few years. People tend to replace their desktop and laptop computers every three to four years, and they upgrade their smartphones and tablets even more frequently. But a smart car, Internet-connected power meter, or smart traffic light may have a significantly longer life span, in some cases measured in decades. So you can't just expect that the replacement will fix any security problems the old one had. Nor can the manufacturers anticipate what sort of hardware resources their devices might need to thwart the kinds of attacks they will experience in the distant future.

Indeed, it's hard today to imagine how exactly these devices will be used, much less what the threats are going to be 10 or 20 years from now. Perhaps by that time your refrigerator will communicate with your autonomous car so that it can automatically fetch groceries when they're needed—but a compromised smart lightbulb in your kitchen will also be able to eavesdrop on that communication and make mischief with the information. We simply don't know enough to predict the uses of different smart gadgets in the future or the repercussions of their being compromised. So we need somehow to design these systems to protect us against attacks that we will only discover later.

How then can engineers possibly make the Internet of Things secure? In seeking solutions, we enter into highly uncertain territory, with lots of unknowns and very few concrete answers. Consequently, while security experts need to do the best they can to develop protections to known threats, they should also design devices so that they can be configured and upgraded in response to unanticipated vulnerabilities and compromises. Our approach to achieve that is called hardware patching.

AN IOT DEVICE such as a smart lightbulb can easily be hacked and infected with malware, which could then spread to other devices on the network. Future, patchable hardware could reduce this threat.



“PATCHING” IS A FAMILIAR CONCEPT IN COMPUTING,

at least in the realm of software. People today are painfully aware of the need for software updates to ensure the continued secure operation of their phones and computers. Most of us get a steady stream of notifications informing us that new software is ready to be installed—reminders that become increasingly persistent the more we ignore them until finally the thing that’s complaining refuses to operate, at which point we give up and agree to the update.

Often, some applications stop working afterward and need to be updated themselves, taking more time and sometimes causing significant disruptions, which is why most users agree to software updates only grudgingly. And yet, these security updates are necessary because a typical computing device is exposed to dozens of new vulnerabilities every month.

Up until now, patching has been done only to software or “firmware,” which is how people often refer to the system code running on small devices. The underlying hardware is itself immutable. We submit that engineers must permit not just the software but also the hardware to be patchable on devices intended to become part of the Internet of Things. Why? Because it may not be possible to fix all security vulnerabilities simply by modifying the software. For example, the hardware might implement a cryptographic encryption algorithm that is secure now but could become outdated long before the end of the system’s lifetime. The only way to address that possibility is to have hardware that can be reconfigured after the device is manufactured.

Another reason to make the hardware patchable is that small connected devices frequently must operate using very little energy, and software implementations of a given functionality

typically consume more power than do hardware implementations of the same thing. So engineers often can’t design a small, low-power device that does what it needs to do just by using software running on some generic hardware—the devices must use special-purpose hardware for the job. As a result, software patching will probably be insufficient to make the needed security upgrades.



SMART WATCHES OFTEN
hold the owner’s name,
address, date of birth, weight,
gender, heart rate, and other
health information. And they
can be troublingly vulnerable
to hacking.

Clearly, one requirement for a patchable hardware design appropriate for the IoT is that it must work under highly aggressive energy constraints—some wireless sensors, for example, draw on average just a few microamperes. Our work provides a design that can operate well under such constraints. One way to achieve that is with something called a field-programmable gate array (FPGA), a general-purpose chip on which the logic can be configured after manufacture. Our contribution to this line of research is to develop an architecture, one built around an FPGA, that can satisfy different security requirements.

To understand how this would work, consider a chip to be used in a small device destined for long-term deployment on the IoT. It could be for a smart lightbulb, a refrigerator, or whatever. In our proposed architecture, a centralized hardware block, called the security-policy engine, manages a comprehensive set of security-critical events, including communication among other design blocks in the system and with the outside world.

As an example, the security-policy engine may require that a secret cryptographic key used for communication should be accessible only to certain specific hardware blocks. To enforce this rule, the security-policy engine must manage the sharing of secret keys between blocks, forbidding exchanges that don't satisfy the specified security requirements. What if one day you learn that one hardware block contains a security vulnerability and should no longer be allowed access to a cryptographic key? If the hardware is immutable, there may be nothing you can do.

Now imagine that this security-policy engine is built using an FPGA. Because an FPGA is upgradable, you can patch it. In particular, if you need to protect the device against a newly discovered threat, you patch the hardware to enforce a new set of security requirements while still using just a few microamperes.

In principle, this sounds simple enough. In practice, there are a lot of details that need to be worked out. That's because even small digital devices typically contain many diverse blocks of hardware designed by different third-party sources. In the lingo of the trade, these are referred to as intellectual properties, or IPs. A security-policy engine needs to track communication between the different IPs so that it can enforce security requirements and identify violations. What's more, a security-policy engine needs to have access to the security-critical events going on inside each of those IP blocks so it can properly flag those and react.

What's needed for that is a special interface that allows each IP vendor to use a common mechanism for its block of hardware to communicate with a security-policy engine. That doesn't exist now, and it may take a very long time for standards for such an interface to emerge. Fortunately, most

IPs already have what's called a debug interface, which is normally used to check whether the IP is working as it's supposed to after it's fabricated in silicon on a chip. If we connect the security-policy engine to this interface, that engine can track a large number of different events going on inside the block in question. And if a new security requirement involves monitoring or reacting to an event that is already considered critical for validation of the IP, which it most likely will be, a hardware patch can allow the security-policy engine to track the relevant events straightaway—no need to alter the IPs at all.

Of course, if a new security requirement involves events in an IP that are not accessible through its debug interface, you're out of luck. Our hope is that this situation won't arise very often and that over time IP vendors will only make their debug interfaces richer—at least until a standard security interface for IPs is developed.

In the long run, as security experts understand the needs better, they will develop even more flexible protection mechanisms for these low-power devices. Like the software security patches so common now, hardware patching will become a routine occurrence on the Internet of Things. The challenge will be figuring out a way to have these systems upgrade themselves on a regular basis, in a painless manner, without all the fear and loathing that accompanies software upgrades today.

There are already efforts to provide automated software updates for a variety of small Internet-connected devices like phones, a process called over-the-air, or OTA, updating. Such mechanisms must of course ensure that only authentic software upgrades get loaded, and they must be robust enough to handle the loss of power or communications during an upgrade without “bricking” the device. Similar considerations will apply when hardware-configuration updates are made automatically. And those requirements will be difficult to meet in small IoT devices, which don't typically have the hardware or software necessary to support such complex tasks.

While there will certainly be challenges to keeping the Internet of Things secure without asking users or system managers to manually upgrade dozens if not hundreds of different devices, we are hopeful that progress will be made and that systems for automatic upgrading will proliferate along with the devices themselves. If that happens, and if the components making up the Internet of Things are sufficiently flexible, we believe that reasonable levels of security can be maintained, even as the number of such devices grows into the trillions. And that's probably going to happen sooner than you think. ■

➤ POST YOUR COMMENTS at <http://spectrum.ieee.org/iotsecurity1117>