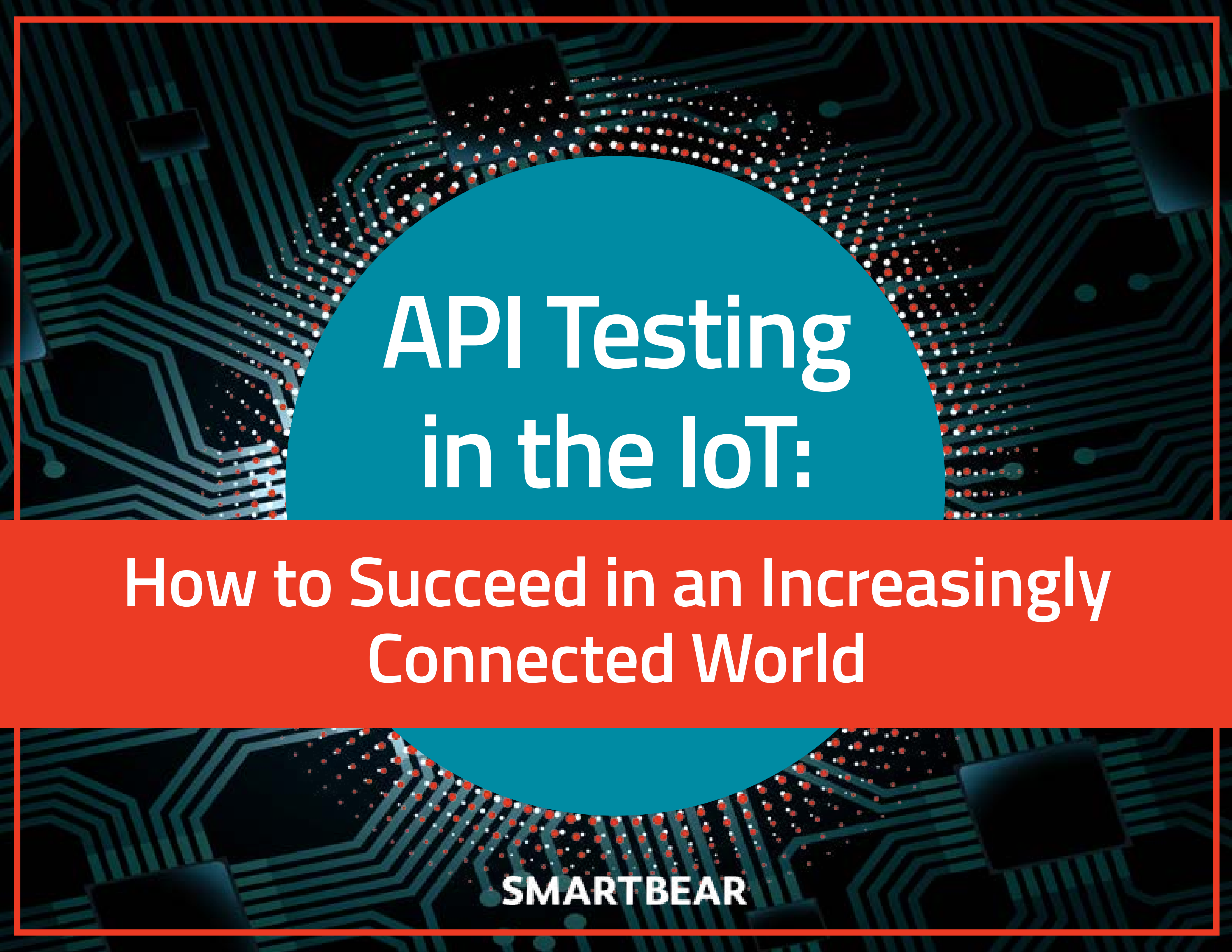


API Testing in the IoT:

How to Succeed in an Increasingly Connected World

SMARTBEAR

Table of Contents

Introduction

It seems not too far off until fridges will be auto-ordering our milk and cars will be steering us around. What we've appreciated for years in the Software as a Service (SaaS) space as integrated workflows, driven by application programming interfaces or APIs, is now driving our entire integrated, automated, connected world.

The idea of connected devices isn't a new one, but the magnitude of what's called the Internet of Things or the Internet of Everything has the potential to impact our lives like never before. IoT brings with it awesome opportunities to improve our lives through healthcare, education, safety, parenting, and so much more. But it also comes with its own staggering set of security and privacy risks.

"Last year, the popular term is Internet of Things. Internet of Everything is taking over. My prediction is, in a few years' time, they'll just say 'The Internet'," said European QA testing expert Paul Gerrard.

It's changed from us going online to us being online. According to Gartner, by 2020, the number of connected devices lumped together as the Internet of Things will grow to 25 billion, generating $263 billion in revenue.

This means designers, developers and testers have more of an opportunity than ever before to impact the lives of millions, or even billions, in an increasingly significant way. Likewise, how people use our solutions will dramatically change the way we design, develop, and test. There's no doubt that the importance of the role of the tester will continue to increase with the Internet of Things.

But that doesn't mean it'll get any easier. In fact, the job of the tester is becoming several layers and stacks harder. Although we may not know exactly what IoT testing will look like, and we can't know what the future of IoT will hold, there's no doubt that the tester is at the forefront of the Internet of Things.

Here at SmartBear, we may be experts in helping other experts test their APIs, but that's just one of many layers in this complicated, connected stack. That's why we've listened to dozens of IoT and testing leaders — exploring the current state of testing, their experiences with testing IoT devices, and their predictions and demands for the future of the space, to create this comprehensive guide to the ever-evolving state of testing the Internet of Things.

When we started this, the response from the tester community and the development community as a whole was a ravenousness demand for more information. The Internet of Things is no longer an *if* but a *when*, and the community wants to be prepared. But, for all the hype, there's not a lot out there on the topic. That means the current state of IoT testing is, at best, in flux, and we are looking at scale, breadth, and risk like never before.

This eBook, and particularly the IoT influencers referenced throughout, will help us answer: **How on earth can you perform testing in the Internet of Things?**

Where is IoT testing today?

What is the current state of Internet of Things testing?

A better way to title this first chapter may be: Is there a current state of IoT testing? The Internet of Things is in its infancy, making testing it just as green, if not greener.

Gerrard says IoT testing is still pretty undefined: “Hardly anyone has mentioned it. People don’t understand the risks yet. They don’t know what goes wrong. Most people are thinking of IoT as just an extension of the mobile app, like the Fitbit on your wrist, but that’s kind of the most trivial.”

He says it's so much more than that — it's about the dramatic impact the Internet of Things will have on society, where we're looking down the pipeline at 50 to 100 devices in every home, completely connected or sporadically and unpredictably connected smart cities, interacting with autonomous smart cars and things on or within our bodies.

According to Gerrard, "People haven't really figured out the social impact of all this. In many respects, these devices can't do any harm, but the infrastructure is still evolving, the standards aren't stable yet. We are still in the 'Betamax versus VHS stage' where we just don't know where it's going to go."

The risk is that IoT itself is still being released out into the wild in more of a beta version, where just about everything is being created and tested in isolation. But what happens when we bring different brands, manufacturers, protocols, standards, and use cases together?

> "The first struggle with Internet of Things testing becomes knowing what to test because it's no longer just a website, a server, and a mobile app coming together."

But as the pioneers and the tinkerers of the tech world are giving IoT a go, a lot of the discovery finds that, while it's undeniably more complicated, IoT brings to head the best practices of software testing that have been touted over the last forty or so years.

According to IBM API and IoT expert Andy Thurai, The concept of connecting sensors/devices to a broader network (or even Internet), is not all that new.

There were connected devices/sensors — we called them M2M or machine-to-machine — since the seventies and eighties, particularly in the Industrial Internet areas such as factory assembly lines, utilities, and power industry. They used to be controlled by PLCs in the localized private networks. Hence there was no need to worry much about security or constantly patching/upgrading firmware or replacing it with newer devices.

"It never used to be an issue until now because they were all on a private network, completely controlled, completely isolated, never on the Internet, and not accessible to the bad guys."

Thurai notes that only a smaller portion of the modern Internet of Things will fall within this controlled environment, as it will be too expensive to maintain private networks. The rest will be out in that unknown we'll refer to throughout this eBook.

Gerrard makes the comparison that, in the Internet of Things, the Internet is just acting as an implementation of Client-Server. "Suddenly what you were testing is distributed, but now it's the system of systems that are interacting. It's now hundreds of thousands of devices running thousands of different applications on different networks."

Most of the Internet of Things is not the same type of controlled environment you'd experience when developing mobile apps. You can be reasonably confident of the specs you're dealing with when building an app for an iPhone. There are many, many more factors that need to be considered if you want an application to be compatible with a variety of IoT products, said Mike Kruk, CEO of Crowsnest, customer support analytics for the Internet of Things.

As exciting of a time it is for rapid innovation, it all makes for a risky investment. "Manufacturers have no control over the entire stack," said Brian Knopf founder of BRK Security and 20-year veteran of security research and testing.

"If you have to update something, [it] has to be done across whole devices. It's going to lead to recalls."

With no clear protocols, no clear standards, and hundreds of devices, across every vertical, there will be millions of devices you won't have access to for upgrades or patches. And even when you can be almost certain of your own device or software's security, you can't be sure what it'll be integrated with will be anywhere near as secure.

Too Many Cooks in the Kitchen: The challenges of testing in the Internet of Things.

"You can't know what all the connections are and you can't really test for what all those connections could be," said Bruce de Grazia, program chair of the cyber security management and policy department at University of Maryland, University College.

Grazia succinctly describes perhaps the most overwhelming part of tackling IoT testing in an up-to seven-layer stack. These stacks aren't dominated by one provider either but rather they are many companies integrating together, with or without their own knowledge or consent, which sparks the questions:

- **Who is in charge of testing which layers?**
- **Who is responsible for enforcing the protocols and standards that weave the layers together?**

As Aditya Gupta puts it, IoT testing is about constantly asking "What network it's accessing and what data they are getting?" Gupta is a mobile and IoT security researcher and founder of Attify, which helps organizations secure their IoT devices and code.

Diwakar Menon, CEO of Last Mile Consultants says that it all depends on the context of what you are testing. He points to four different possible interactions the device makes in IoT:

- **With sensors**
- **With aggregate gateways**
- **With the cloud network**
- **With the application itself**

For his consultancy, Menon says it all depends on what they're looking for.

> **"If I am testing at the gateway level, I would test it differently than sensors, than apps."**

He calls the "human level of testing," going through chains, constantly having to be aware of user context, of what the application does, and of what data is exchanged between the users and the sensors.

As a tester — as well as a designer and developer in the IoT space — you must constantly consider these different facets:

- **User information: kind of, amount, nature of, who can see it**
- **Security**
- **Functionality**
- **Performance**
- **What happens if you lose connectivity?**
- **What happens if you lose power?**
- **Data aggregation, security, corruption, storage**

Stacey Mulcahy, technical evangelist at Microsoft, said,

> "In general, you can't predict your conditions or your environment, so you've got to do the best to replicate that, and prepare for downtime and security."

However, not everyone can afford to simulate many of the unreachable and pricey devices and environments out there.

Perhaps even more unpredictable than conditions is the human variable in the Internet of Things, when you have a group of users that are so far from the testing space using tools in unknown ways.

"Up to now, we are guessing usage. We always really guess or try to guess in terms of how it's going to be used, but the bigger challenge is that at the end of the day we can't really know if we can," Menon said.

"There are too many facets and one of the challenges of the Internet of Things is that you don't know where to begin and end," he continued. "With so many users, you have to be aware of the fact that something is going to be used in other ways.

We are dealing with a whole new beast that will continue to surprise us for a while.

You can uncover some really obscure patterns, said Vlad Trifa, co-founder and VP of research and development, EVRYTHNG IoT platform and of WebofThings.org and the book of the same name said,

> "It requires a whole new way to think about that. It's really hard to test IoT digitally. It's not like software where I have a piece of code: OK, pass, fail."

Menon pointed to how a security device could have medical applications or school safety applications. "How on earth would we take that platform and say we've tested it for everything?"

But, this is one of the many examples of how the tester's job in IoT becomes more interesting, with a side of a greater sense of responsibility.

Trifa said, "Because it's real people and real cities, the impact is much bigger and the complexity of what can happen is massive. It's pointless to test a smart lock in a lab. Put that in hundreds of houses where you can uncover information like how it reacts with metal doors, when it rains, when the lock is actually outside the geofence of your house and, if you lose your phone, will you have to sleep on the street?"

The Prevailing Internet of Things Protocols.

The three components of Internet of Things architecture are:

- **The hardware**
- **The mobile app or web-based device**
- **The protocols and standards**

The protocols and standards vary by device and use case. This can include near-field communication (NFC), WiFi, 3G and 4G, Bluetooth, and radio frequency standards and protocols. Beyond existing protocols and standards, more are emerging and competing to become the prevailing IoT protocol or protocols.

For the foreseeable future, IoT testers, as well as programmers, will continue to ask, "Which protocols?" The Internet of Things begs questions like: Does it connect to a different physical network, through 3G or 4G, wireless, or all of the above? Are you defining the wireless communication or the physical standard level protocol?

We're talking about the rapid acceleration of the generation, accumulation, and consumption of data at an inconceivable level and we need standards to manage it all.

And it doesn't just end with the protocols and standards set up for the different layers of the stack, but for the different industries a device can be used in, having to comply with HIPAA, PCI, Sarbanes-Oxley and other guidelines. De Grazia says that soon there will be cyber security standards emerging.

He says that at UMUC, when talking about cyber security in IoT, "We're not telling them how to write software, but then we're just telling them what they need to achieve when you can have many protocols as long as you have certain standards. It's the Wild West out there."

The answer for the Internet of Things may be to mix and match protocols depending on the device, situation, and use case. Mike Amundsen argues that "We're trying to build too much intelligence into one device. That's why we have so many protocols." Gupta continues,

> **"Testing in [IoT] is much different from testing other mobile applications or platforms. The framework is different and the way they talk to each other is different."**

"Whenever you are testing IoT, it's very hard because there isn't one particular standard that all the devices follow. Most of the standard IoT devices might follow one standard, protocol or a different testing scenario for each of the different protocols."

Richard Parker, founder of Altitude Angel software for the Internet of Flying Things, said it's "very much reminiscent of the dot-com boom — everyone is rushing to compete the standards," but that for his company, "we aren't trying to compete, we want to work with everybody. Integrate with us and, in addition to your drones being safer, everyone else's drones are safer." This mentality falls more within collaborative trends like open source and microservices and may be the best way for a tester to prepare herself for IoT — not anticipating the newest protocol but rather being ready to adapt to them all.

It's not unusual for protocols to compete and eventually one or two will surface to the top but, as it stands, there are at least a dozen fighting to be dubbed the prevailing IoT protocol. There aren't enough pages nor enough depth to dive into each of them but our sources referenced a few, most often potential protocol frontrunners Constrained Application Protocol (CoAP) and MQTT**.**

Paul Bruce, SmartBear's API product marketing manager and a veteran performance load tester, points out that protocols are important because they allow you to be interoperable in the often uncertain and asynchronous IoT testing environment. SmartBear and Bruce are both looking toward MQTT and CoAP "to build a first generation of formal testing around the Internet of Things, both functional and load testing. MQTT provides certain semantic — Websockets, HTTP, UDP [low-level protocols] — asynchronicity in a publish-subscribe model," Bruce explained.

Gupta looks to apply CoAP to low-resource devices where battery consumption is critical. Menon says that testers looking toward the IoT space should start playing around with small protocol generators and open source platforms and get familiar with protocols like MQTT, which he says seems to be one that is widely adopted at this point. But he also reminds that "There is not just MQTT, there's even XMPP [Extensible Messaging and Presence Protocol]. All that is important to understand is not to bet on just one — I'm not the betting type. If you look at it, they operate in different planes, devices to server, server to device, etc. The ability [for] grounding in a certain programming language like Java will allow me to shift to another." For now he says that you need to condition yourself to how different programming languages and constructs are functioning in the Internet of Things.

Perhaps in a sea of protocols, developing standards in different areas is more essential, at least for the time being. Some consortiums looking to develop standards for certain stacks are:

- Wireless Power Consortium
- Industrial Internet Consortium (Working on IoT data initiatives, including ATT, GE, Cisco, Intel, and IBM.)
- Open Internet Consortium (Including: Intel, Samsung, a few others.)
- Open Interconnect Consortium

Then there are alliances and consortia dedicated to regulating in certain use cases and industries. Some of the most well-known are the Zigbee Alliance for home automation standards and Z-Wave for smart energy standards.

In such a new space, each of the protocols has its own weaknesses and untested holes. Knopf calls Zigbee still "extremely weak, from a theoretical standpoint." He explains, "The way Zigbee works, you have the devices, the trust center, and then routers to send the signals further around your house. The speck didn't work to stop the router from going through or changing it. Zigbee says the door is locked, but it's really open."

Like all things in testing in the Internet of Things, standards and protocols are still in the early stages. "I don't think any of the protocols are mature to go for a long period of time," Gupta predicts.

He says that when developers are engineering, they must decide how efficient and productive the protocols and standards they use are. "Each has their own pros and cons. Some have their own security issues and power issues."

So the question is, if there aren't standards or widely accepted protocols, how can you as the tester do your job? API Evangelist Kin Lane argues that:

> "We need maybe not standard protocols for IoT the tech side, but we need some standard practices to the business and policies about how these devices work."

He advocates for international web literacy.

Thurai agrees that "It's going to be close to impossible to have one standard. Consolidation will happen and we will move toward some standardization but it's going to take a while," comparing it to Cobol programming which was thought to disappear in the seventies but is still alive today.

"Yes, there can be some consolidated standards that can emerge, but I don't see an overarching standard that can involve IoT. It's going to be close to impossible to have one standard."

In the meantime, one interesting resource is Postscapes which is an IoT research group that's tracking the manufacturers, platforms, and protocols in the Internet of Things. It's a good way to keep up-to-date on the industry, as well as to help you decide what protocols and standards are a better fit with which hardware.

How to test for security in the Internet of Things

First rule of security: there is no security.

From the basics of our one-password-fits-all world, where we post answers to our security questions on Instagram, to the deeper threat of nefarious hackers actively trying to destroy the world one device at a time, the Internet of Things security tester has to be more creative than ever before.

"We are walking towards disaster, apocalypse. Not there yet, but walking towards it. It's a time bomb waiting to explode. Why? As with any new technology, when a new technology comes in, you're obviously in a lab mode to experiment and then you're in an innovation mode to make it work for your environment. Then when you find something with value, you latch onto it and tell the world 'This is what's going to be the next big thing!' In Go mode, nobody thinks about security." Thurai continued, "Security is always an afterthought with any innovation scheme, which is OK if you move from the Hype to Reality and take care of the security, privacy, governance, compliance, first, before you [have to] live with it." By the nature of how close it is to our daily lives, IoT testing and design must be all about understanding what risks we can actually manage and what we can do about them, as well as identifying the risks we can do nothing about.

IoT technology is advancing at a rate that's outstripping enterprises' ability to secure internal and cloud resources. Organizations are increasing cyber security budgets but not at a fast enough rate in line with this rapid-to-market innovation. And it's not just security, particularly enterprise systems have to combine security testing with testing for compliance, privacy, security, authentication, keeping records, and keeping track of security, too.

Thurai strongly advises, "Don't move from hype to reality without fixing those holes, moving from that lab exercise to a reality mode. When you release to the public, out in the wild, you have to have security in mind."

When you are selling a non-disposable product with a long lifetime, like a washing machine, you may not have access to it to patch up security holes. All usage and security must be considered and tested before a device is designed and developed for the market.

Security has to be tested before any release.

"When you put the gadgets, the sensors, the metric collectors, traffic sensors, motion sensors so far away from your network, the majority of the IoT devices are completely away from your enterprise Fort Knox," Thurai said.

Add to that, he says, "When cost is cheap, security isn't a priority." Thurai argues that many of the sensors and devices are extremely small and inexpensive by design. "If I'm making a sensor that's two bucks apiece, the last thing I'm going to worry about is security for the device."

He points out that in this situation, often "the liability issue moves from the producers of all those components into the operator. And then the operator shifts the liability to the consumer by making them agree to terms and conditions, or asking for permission, most times without the consumer even understanding the full implication of the agreement."

According to Thurai "When you're talking about devices being used unpredictably and connecting to unknown, potentially unstable sources, bad stuff is bound to happen — the question will soon surface: Who exactly is liable?"

Still, while we talk about how different testing the Internet of Things is, testing for security usually comes down to software security testing. Of course, all hardware also needs to be tested for functionality, usability, user experience, and safety, but the security issue isn't usually in the hardware.

De Grazia pointed out that just about every time you read about technical vulnerabilities, it's in the software end of IoT. "There are hardware failures but considerably fewer of them have vulnerabilities that will give you access to something in the machine or system itself — a switch could go bad but it's usually the software."

At UMUC, "We teach the importance of making sure the code doesn't have bugs and all the testing that needs to be done on the software side as opposed to the hardware side. Software testing and software development for the Internet of Things is on the cutting edge because that's where the vulnerabilities are," he said.

Perhaps because it's the simpler piece for security testing, Gupta recommends that when testing security in IoT, "Start with the device. Think about the security for the mobile app and the communication between the device and the mobile app, what protocol they use," constantly keeping in mind the device and the server.

There are certainly parts of Internet of Things testing that can be automated already, but IoT security testing will probably never be able to be fully automated, at least as long as humans remain more cunning than computers.

Manual testing will remain essential to IoT security.

> "I don't think everything can be automated. I think there has to be a human kicking it off, being a part of it, Lane said."

He says this is one of the goals of API JSON, to make sure your device and cloud APIs are all indexed and available in one location for testing security, performance, and load balancing. "So the theory is that at the home level, every IoT device you bring into your home— Fitbit, drone, etc. —they all register themselves with your home router and simultaneously there should be some sort of JSON router, that creates an overall index of all the surface area."

Lane referred back to the three-legged OAuth promise of a partnership between the platform, the developer, and the end user, where the user should be able to use a software to learn about their own security and privacy.

He envisions one day soon a software or device that you can put on your home network that scans and reports back: "Here's all the cloud-based APIs, devices…look for vulnerabilities, look for it in the device and cloud. Send back a report: Your firmware has to be updated. That Internet site doesn't use SSL. You've got this one device on your network," creating individual security reports per home, organization, or even person.

Gupta's Attify offers up what could be the answer to Lane's request. Their mobile testing automated platform AppWatch will soon be evolving into a full IoT security platform, with hardware that sits on your network, between the router and the devices, monitoring what kind of data is going on through those devices and checking if that data is secure. "If a new device tries to connect with this particular WiFi, it evaluates" if it's secure.

But it's also the tester's job to do her best to test all security scenarios and vulnerabilities before the IoT software, app, or device is shipped.

With great power comes great responsibility.

As the role of the tester increases, so does the sense of responsibility. And the voice of the tester must be heard clearly throughout any IoT-related organization. It becomes the tester's job not only to test for desired results and functionality, but to explore further, asking 'Why?' at all times. Not only should the tester be heard but he needs to push to be heard now more than ever.

Director of API architecture at API Academy Mike Amundsen puts it this way: "Security is a context-based conversation."

As tester, you need to be careful in your thought process, always asking why when you say: "This system needs to connect to this system." What is the purpose? Does it add an important value? In the case of former U.S. vice president Dick Cheney's implanted defibrillator, it was clear that connecting to WiFi leaned well toward the security risk over its intended purpose to deliver information to his doctor. Sometimes it's a case-by-case situation and sometimes it's just dumb for any default connection to certain systems.

Knopf says you need to ask, "Why did you have it open? A gimmick. You want to be able to monitor them when you aren't with them," referencing the very public hacking of a child's insulin monitor. In general, there's a need to have the WiFi turned off by default in the majority of devices.

For Knopf, everything IoT should be:

- **Secure by default**
- **Secure by design**
- **Secure by deployment**

He and Lane both talk often about how it is the responsibility of the IT community to educate the public on risk factors. This is why Knopf co-founded I Am The Cavalry to develop a five-star rating system, similar to that of the automobile industry, to educate the general community on the cyber security, safety, and privacy of IoT devices. He says "Every device should be tested and results published publicly."

For this, he decided to similarly apply OWASP's Top Ten to IoT. "Let's list the top ten vulnerabilities and allow people to learn how to protect from them. We've been talking about the same ten types of attacks for decades and no one's fixing them. And now you have people building these IoT devices and it's even worse, and they are low security."

While the devices are getting more complicated, the risks we face have been the same for a while. In the end, all IoT security testing is as Thurai says:

> "Don't trust, always verify."

And we need to remember that everything we have now is just the beginning. "It's just so prototype-y — the whole Internet is a massive playground," Trifa warned.

"We've gone from no computers to everyone having a computer everywhere in 20 years. It's going to be so fast. It's going to be a lot to change, as the Internet of Things come into our houses, it's just going to exacerbate things."

Threat Modeling: What to do when a human guinea pig simply isn't an option.

Even if you had all the resources in the world — and for at least the first few years of the IoT revolution, you won't — sometimes you simply can't run full testing. This is especially true when you're dealing with implanted medical devices that, if you sling a testing curveball at them, you could make the device fail or worse. When as a quality analyst you find that you cannot do a full testing because it's dangerous in some way or there simply aren't the resources available, you should get used to threat modeling — thinking about how an attacker could perform reconnaissance and exploit your environment, and then layer protections against the threats identified in your model.

From an Internet of Things perspective, Knopf says that when you are going to test Internet of Things security via threat modeling, there are three key factors:

1. **Sensors to collect and measure data**
2. **Connectivity to connect & communicate**
3. **People and processes to integrate and innovate**

"We're not talking about your IoT toy. We're talking about a person who needs something for quality of life," Knopf said. He says his QA security job used to be like "Hey Brian, I need an alarm system in my home. Come tell me what I need. Bars on windows, alarm sensors, a siren, a sign." He says that now, in the IoT space, "Those windows and doors are just interfaces — wireless, ethernet, [asking] what are the interfaces."

When something fell on Knopf's wife's foot, causing irreparable nerve damage and pain levels that she could no longer function with, eventually, the only solution left on the table was to have an implant put into her back to manage the pain. But being a security assessor, when the doctor told her, "We can implant a pain management device in your back that you can control and charge wirelessly," Knopf became skeptical.

The tiny pain management device generates electricity in the body to block pain. It's controlled by a remote about the size of a pager and is charged magnetically, with a battery pack that you actually plug into the wall, with his wife having to charge herself weekly for multiple hours at a time, not able to fall asleep during for fear of overheating the device.

"Unlike some of the other medical devices where they can say 'You know what, it's vulnerable, take it out,' we're talking about an operation at this point. It's tied into her spine with anchors and the initial operation costs $30,000, so that's not something that we're just going to go ahead, 'Hey, why not? Let's update it', " Knopf explained.

Of course, Knopf didn't think it was a good idea to go through a full QA test or to reverse engineer the device because, well, it's risky to run experiments on your wife in more ways than one.

"I had no intention of doing a security audit of a device in my wife's back. I didn't want to break it," he pointed out.

Therefore he decided it would be much safer to threat model his wife, in order to understand what impact it may have.

What did he threat model for?

- **LIFESPAN:**

Scar tissue usually develops around these devices, so those that only had a couple years' lifespan didn't make the cut.

They chose one that should last for nine years, in which time her nervous system could reset itself.

- **VOLTAGE:**

This is an internal device that doesn't require higher voltage like an external defibrillator, but it still has a range between 0 and 10.5 volts. To put it into perspective, anything higher than 3.5 volts causes his wife physical pain.

- **LEAKAGE:**

One of the three main manufacturers of the about 40 different device options had to do a recall because the battery could leak into the body. Of course, in this situation, a recall involves repeating a major surgery. "If you think about it, at this point, it's no longer updating firmware," Knopf said.

Instead of going through how a general threat model works, we think the results of Knopf's threat model and the potential risks Knopf flagged explain the logic behind the risk, mitigation, and likelihood of an attack.

Risk #1:

Damage to neurostimulator caused by strong electromagnetic (EMI) interference.

Mitigation #1:

EMI shielding and an MRI-safe mode.

Likelihood:

Highly unlikely.

Risk #2:

Via wireless signal, someone could change stimulation profile, causing the user to be in pain, which in turn needs more medication and potentially overdose.

Mitigation #2:

Remote only works when directly against the skin. External signals don't change this.

Likelihood:

Highly unlikely.

Risk #3:

Attacker turns stimulation on high voltage.

Mitigation #3:

Remote only works when directly against the skin. External signals don't change this.

Likelihood:

Highly unlikely.

Risk #4:

Overheating of skin during charging causes burns.

Mitigation #4:

Neurostimulator monitors skin temperature and its own device temperature. Stops if unit or skin overheats.

Likelihood:

Highly unlikely.

Risk #5:

Riskiest, based on damaging leads with high radio frequency causing scarring, electrocution, shock or death.

Mitigation #5:

New devices have much thicker lead dispersing RF across whole length of lead.

Likelihood:

Highly unlikely.

Because of these strong mitigations of the risk, Knopf, his wife, and his doctor decided to move ahead with the operation to insert the device. But why did he choose to use only a threat model?

- **Device was $30,000.**
- **You shouldn't pen test inside your spouse.**
- **Wasn't sure if they'd even sell him one. (But he would continue on-device research if donations for purchasing one are received.)**

Threat modeling is becoming a popular way to address the distance problem that we will increasingly have when more devices come to market, particularly with big-ticket devices and those embedded in our body, but threat modeling is a compelling way to kick off any testing for IoT security.

"Many of the organizations or even the developers want to build a particular product and then test security before market. Whenever you build a particular product, you should start thinking of the security from the very start built into the framework. Create a threat model from the start" Gupta said. It's a better role for the developer to have the security mechanism in place before the testers actually test it.

Security is an essential part of the human experience.

When you're talking about devices we interact with every day, have in our homes, or on or in our bodies, it's important to think of security as essential to user experience and, really, to human experience.

> "If I knew, for example, that Facebook was sharing a lot of data I wouldn't want shared, it could impact how I view the app."

Menon says that user experience isn't all elements of security, but personal privacy and personal security does come under the UX umbrella.

So far, most data transfers have been between humans, services, and businesses but, as Sean Hargrave of The Guardian writes, when machines start to collect data, it becomes a regulatory minefield, and what data you are sharing becomes part of the security and privacy questions we must be asking.

Similarly, while security is part of the user experience, human error is part of the security experience.

You have to test both for "the engineering failures and the human failures. It's the perfect example of whatever can go wrong, will." De Grazia said that testers "need to think of more than just the technical failures. They need to think about the people failures — what could somebody do that could make this vulnerable?"

Perhaps healthcare is where there needs to be even faster innovation in Internet of Things security testing to keep up with the rapid innovation of the marketplace. Thurai points out a possible solution is to have all the connected medical devices in a hospital connect to only a specific network that's been secured, authorized and authenticated.

"Also, before it can send any information anywhere, both parties need to identify, authenticate and authorize each other," he said. "While the worry of someone connecting to your device and manipulating it to harm a patient is an issue, allowing unknown devices to connect to your network to feed the data or pollute your data collections should be considered as an issue too," Thurai continued.

Of course, healthcare, like all emerging IoT focuses, will need more standards and protocols that we will have to follow.

How the Internet of Flying Things will change security testing.

Perhaps one of the most challenging "things" to test is the hottest: drones, which makes looking at how Altitude Angel has solved much of safety testing them even more compelling.

What Parker has trademarked as the Internet of Flying Things (IoFT) involves things or devices that are permanently in motion, which have a very different set of requirements and capabilities than in the fixed world. "I have to deal that the drone gets turned off, driven 300 miles and then switched on," he explained.

Add to this the complications of connecting through a combination of local WiFi, short-range radio and, increasingly more popular, cellular networks with potentially unlimited range. And then there's not only the things in the sky but on the ground — other drones, trains, reservoirs, power stations — that are dangerous to the drones, infringing on their escape vectors.

All IoT testing is about risk management, but perhaps even more so in the IoFT, which is why Altitude Aircraft wants to be put directly on the drones as a sort of air traffic control. He says Altitude Angel is completely tested via their in-house flightpath simulator.

For Parker, it isn't just about testing if the Altitude Angel software works with certain devices, you have to test to make sure they are providing information that's important to those things in the air — where people are, which mobile providers, the location of hardware that doesn't move that the drone needs to find like charging and fueling stations, and how to navigate around points of interest.

They use simulation and machine learning as much as possible because "We don't work for the FAA [Federal Aviation Administration], so we can't plan for other vehicles," and "from our perspective, Altitude Angel is 100 percent focused on safety." In this context, safety is first and foremost defined as collision avoidance.

But testing IoT in a more traditional industry like air and space comes with its own challenges. "In the area of aviation, the industry is very old and very traditional. The view is that the drone industry needs to fit in, to co-exist. It ends up holding us to very high, very different safety standards as an aircraft company."

Altitude Angel typically tests for three situations that could apply in most Internet of Things testing:

- **Normal scenario**
- **Abnormal scenario** (Example: Introducing some sort of GPS failure, acting erratically, or offering no readings at all.)
- **Failure scenario** (Example: The drone loses complete contact or is somehow damaged by another drone.)

The Internet of Flying Things isn't just about the things up there, but the people down here. On average it takes ten to 12 minutes for an ambulance to respond to a heart attack, while a drone could cut the time in half to get a defibrillator, staving off significant brain damage. It's for this and many reasons that all of Altitude Angel's safety features are necessarily free.

> "We want it to be open, transparent, and safety should be free."

The breadth of the Internet of Things means it's essential that organizations are more open, forthcoming and sharing.

In IoT, testing for privacy is really testing for security.

With the Internet of Things, we will need a whole new term for what will become the too minuscule buzzword of "big data." It's for this that now more than ever we need to separate the identity of the person being measured by a sensor from the data they generate.

John Taysom, a fellow at the University of Cambridge and co-founder of privacy company Privitar, said in an interview with *The Guardian* that he believes this data-identity disassociation is key because companies and governments can take advantage of the data without taking advantage of the person or any risk to privacy. He fears, though, that organizations might rush in too soon before realizing the potential for compromising an individual's private details. "There's obviously a lot of concern about privacy but I think we're in one of those situations like smoking or sugary foods", he said in the article.

"The gain to getting all that data is very instant but the problems seem a long way off, and so you end up not being firm enough with guidelines until further down the line and governments have to step in to set rules.

You shouldn't forget that ultimately the machines taking the readings and transmitting them are owned by companies which want to use that information. Although we're talking about sensors, we're really talking about the people and companies that own them."

> **"Tech companies must be responsible for questioning if too much information is being transferred or if it's private in nature."**

Most of IoT testers will experience this similar moment: "The data you know it collects seems inoffensive, but then you step back and you see the huge amount of data you have, it's really hard to see that data and the guy that has the device that has no idea the complexity of the device," Trifa said. "Someone stealing my password on Facebook sucks, but this is something where people could have accidents and worse This is the emotional and physical hurting someone can do with the Internet of Things."

And with this, the importance of IoT security and privacy testing continues to grow.

The intersection of functionality testing and interoperability in the Internet of Things

Yes, the themes of functionality and interoperability are big enough to write two books on, but interoperability is so essential to the functionality of connected devices that the two testing concepts can't be separated. "I wanna know with 100 percent certainty when I say a device goes on, it goes on. When you think of the hierarchy of needs, security is great but it doesn't mean anything if I can't get it to function properly," Knopf said.

Amundsen agrees that we are all so focused on formats and protocols, but not enough is being done to check if things are actually being turned on and off as planned.

Everything in the connected world has to come equipped with a back-up plan for when it's disconnected. "As these systems become more integrated into houses, there's a wow factor that people don't stop and say, 'How does this affect me if the Internet goes out. Can I turn on my light?' There's an outage and then no one can control their devices," Knopf said. "I need to make sure that when a customer tries to turn something on, they can actually turn it on manually if the Internet doesn't work."

A good tester will always wonder, if the Internet does go out, are you able to control your devices locally? Or does it come back up when it's restored?

Parker said that "It's not just about software that works and when it doesn't. It's about how it fails. We need to get better, as an industry, at defining how things work in each of these scenarios:

- **When everything is great.**
- **When something is up.**
- **When everything's gone terribly wrong.**

He referenced testing a device in a smart home:

> "If something's wrong with grand-mom's breathing, it's not OK to simply say everything is fine until it's not fine."

And load testing has to go right along with functionality testing because the testing environment will have limits on the amounts of data that can be generated, and, because of this, we must make sure that the techniques used to simulate data — including amount — are as close to real-world situations as possible. Similarly, interoperability has to be tested not just to see if things are functioning in a synchronized way, but questioning if they should be in sync at all. In a connected car, it's important to make sure the entertainment system and the braking system are separate from each other, just as it's important to make sure what events really happened and in what part of the car.

In the Internet of Things, manual testing is an essential cost.

While we are all about automation here at SmartBear, this guide would never be complete without a large section on the role of manual testing in the Internet of Things, something that just about every person interviewed talked about extensively.

Guillaume Gimbert is one of three co-founders of StarDust Mobile, which tests functional performance, with a focus on multiple-device problems, testing "things" like speakers, household appliances, tennis rackets, NFC tags, and bracelets. His 40-person team manually looks for bugs on real devices for more than 300 customers. StarDust focuses on testing interoperability before the device goes to market, seeing what happens when it's connected to a smartphone, when it's disconnected, and then when it's reconnected. It's easier to test the software before putting the software into production. It's more difficult with the object because you have to manufacture the object, Gimbert said, pointing to a greater financial risk in creating an IoT device than just the software for it.

He says that at least a quarter of devices StarDust has tested present a problem with a pairing issue. "Sometimes it connects, sometimes it disconnects, and then you can't reconnect again. Sometimes if it doesn't work it'll never work. Sometimes there are a lot of differences between devices and you are not able to do the pairing."

He says maybe these errors are "because IoT companies tend to focus on the latest platform that has the latest version of Bluetooth, for example, but you are in the wild where the user is using a different number of devices and in that range it doesn't work." This is why his office boasts more than 3,000 different mobile devices for testing.

Gimbert contends that the most important thing in testing an app or connected device is its stability and all the things linked with interoperability. "When an application crashes, it's dead for the users. If you buy an application, sometimes it's two euros, four euros, or it's free, so if it doesn't work, it's OK. You've wasted your time, but when you buy an object that's a hundred euros or more and can't connect, it's an issue," he said.

> "When the customer acquisition cost or CAC is so much higher in the world of IoT, You have to be careful about connectability before putting your object on the market."

Like with all software, IoT testing shouldn't just happen before a product is launched, but it has to happen with all releases, where you are partner testing automation with manual testing.

"Once it's launched, you have to listen to your users or your customers. You have to go on doing some automation testing, and when you're implementing a new feature or are changing the developer, it's really important to do more manual testing. When you do some automated testing, they will detect some bugs," Gimbert continued.

Gimbert does admit that "The problem with manual testing is that it's expensive and it requires a loss of time." He offers up crowdsourcing of testing as a way to lower the price of manual testing dramatically, although it increases the time to market.

While everyone wants testing automation in the IoT space, opinions vary on how much can be automated. Michael Bolton, founder of Developer Sense and co-creator of Rapid Software Testing classes, contends that "Machines don't learn and machines don't have intelligence. There's a tendency to throw words about — to use words very imprecisely — 'learning' is a big one. Machines are not for any human value of learning. [Machines] collect and, by an algorithmic process, they refine their own algorithm."

He argues that manual testing will always be needed because "For humans, learning is about fulfilling a social purpose."

Automation will play an increasingly important role in IoT testing

There's no doubt that a certain heightened level of the human element is necessary to IoT testing. But having more to test means that you need to prioritize automating whatever you can.

In the future of the Internet, there will never be enough testers because there will be numerous layers and devices to test for functionality, interoperability, security, and more. This means the demand for better testing tools and much more sophisticated testing models will only grow in the connected world. Modern QA will be increasingly automated to the point that manual testing will still happen but it will sometimes diminish in importance and necessity — although that day is still a long way out.

To start with Internet of Things testing automation, Gupta points out that "If someone is an IoT developer and if they are already using the testing automation tool, they can build their particular tool on top of that protocol."

The general consensus is that testing the API that connects our connected world should be automated as much as possible because when that breaks down, you're left with a dumb phone and the same remote-controlled planes we've had for decades. Without a fully functioning API, there's simply no more Internet in those things.

"There is a part of testing that can be completely automated: checking," Bolton said. By our definition, checking is the process of applying algorithmic decision rules in order to observe and evaluate some function in the product. You can run a program that inputs a two and then calls a plus function and then inputs another two and returns a four. That part can actually be automated. You can automate a series of checks and activities."

Bolton reminds us that we need to judge our automation tools continuously to answer these questions, among others:

- **Is the programmer able to use it easily?**
- **Does the function have its own level of error checking that programmers can rely on?**

Knopf offered more parts of IoT testing that can be automated, including:

- **Discovering URLs**
- **Doing port scanning**

He goes on to say, "I'm a big believer in writing test code and having more test code than actual production code." However, Knopf strongly warns that "For me, automation is critical, but, from the security standpoint, you can't automate anything," but "from both the functional and performance, all those things can be automated." He said that, If you do black box testing, run through these scenarios and you verify the data. If you've got an API, a mobile application, and a web interface, how do you know where they break down?

- **Stub out the applications**
- **Automate each part of the testing**

Knopf says you have to keep asking, "How do I know my API is working right? I want to understand if I can get anything improper out of the API. I can stub the mobile application because everything I'm testing is the API, and send every range of inputs into it."

In another IoT automation use case, Altitude Angel performs static testing which covers code before it ships, including:

- **Run-time testing for behavior based on dynamic inputs, simulating realistic drone paths**
- **Security A/B testing and fuzz testing: the API expects certain inputs, what happens when they are different?**
- **A testing simulator followed by human verification**

Altitude Angel also runs penetration testing, which acts as a sort of insurance. They outsource this before releases because they run scripts, but "They're still not as good as my team who are building my software and know how to attack it." They use pen testing to find things that are open, as well as internal load testing automation. Pen testing helps them find issues to hone their A/B testing. Gupta recommends that penetration testing should be performed between the device and the mobile app and between the device and the server it speaks to, referencing Fitbit's very public security vulnerabilities that they weren't sending data securely to the server.

> **"We perform extensive penetration testing of the entire IoT device. We assure that at that particular time, it doesn't have any other security issues. We give a certification."**

Admitting that in this rapidly changing world, "maybe the next week a new security vulnerability can come out. It's not secure for the lifetime. You have to do the security testing over a period of time depending on how often the device updates code, making sure you do security testing every month or every two months."

Gupta recommends that you "start testing for the mobile app and web app for which they probably have a tool for," but he warns that "the rest of the problem for the hardware device, there aren't a lot of testing tools for how the device actually works," offering up that maybe you can "plug your device into the debug ports to test that." He points out that, for many of his clients, "the overall architecture of IoT is something a bit different from the architecture. They might not be familiar with testing the mobile device and [it takes] a lot of manual effort. Their functional testing and security testing were mostly manual steps around the test cases."

Test design must evolve in the endlessly scaling Internet of Things.

Redesigning our lives online necessitates redesigning testing as well. Trifa says that you can automate "the really nasty low-level engineering bits, yes, [but] the complexity of data is going to be hard. The more data you put in, you have the exponential of mixing data, harder to automate." Gerrard observes that, "What we haven't got are good enough tools to create test design at scale. If I'm testing a traffic management system for a medum-sized town, you can't just say 'Let's create some random locations and destinations for cars.' They can't just be randomly placed," when you have countless variables like one-way streets, holes in the road, traffic lights, and not being able to drive through walls.

In the Internet of Things, test design shifts from hand-crafting a few tests to "designing tests by patterns and then randomizing within that legitimate pattern which becomes a test model."

Speaking of the current testing automation tools, Gerrard continued that "The models we have are viable for actuaries, but we need to do that for cars, trash cans, pharmacists, fire and rescue, and police. Historically we only tend to model for the purpose of requirements and throw them away. We now need to create trusted models for both developing and testing. The testers' contribution is to challenge those models and refine them," Gerrard said.

"We don't have millions of tests. It'll shift from running their tests manually to shifting tools to automate it." He offered the example of how could Google or any other company ever test driverless cars in a bustling city like Shanghai? You have to do simulations. Offering up the example of air traffic controllers, he says we already have sophisticated simulation technology, but that it's "ferociously expensive."

Gupta contends that to design tests for the Internet of Things, "You have to write the test cases for all the possible things that any user could do with the device." If you have four buttons, you can't just do the workflow of one, two, three, then four. "Not just test for the normal working flow but also check for all the probabilities and see how the device acts for them."

> "Again, with the heightened level of unpredictability, out-of-the-box testing becomes the primary form of testing in the Internet of Things, even as you try to automate it."

Menon echoed Thurai's earlier point that many of "the players that are coming out aren't the big players. They can't be tied down to an expensive product." That means testing automation has to be used to make it easy and inexpensive enough to complete as thorough a testing as possible.

Menon says you can start by asking "How can you leverage existing, probably open-source tool sets?" There also needs to be protocol generators that allow you to test your own systems as well as simulate multiple devices. Gerrard says we "need a performance-testing tool which can generate a lot of traffic and we need to feed those agents meaningful tests that have value. We are very good at crafting tests one-to-one — you and the machine — but we need to do that by the million and that's a different game."

The next step is to find the existing test tooling to help it all.

How does IoT fit in with existing testing automation tools?

Let's walk through testing automation of one of the more popular contemporary examples of the Internet of Things — the Philips connected light bulb.

In this case you can test the lightbulb using an API testing automation tool like the SmartBear Ready! API SoapUI NG Pro. Ready! API publishes when the smart bulb first turns on, automates a workflow test and a functional test, and then puts it all through a LoadUI test.

"When you turn on a smart light, the base hub has to know that it's now available so that it can initialize that light accordingly," Bruce explained. "To do that for IoT asynchronous protocols like MQTT and Ready! API or SoapUI, you often have to prepare the tests starting from the end and finishing with the first action, like setting up dominoes and then knocking them down."

He describes it as the typical back and forth from the owner's smartphone, where the "device publishes that it's available, the hub grabs it and publishes that back to the device. Now the device confirms the data and will receive what the hub gave it and publish what to do with that information."

These are all tests that you can automate within SmartBear's commercial API testing tools because of the MQTT and CoAP plugins shipped with Ready! API.

“The trick is, that thing is listening. With IoT, you have to actually start with the last thing, simulate it in a testing environment like this, prove that the software on the device is responding with what it’s supposed to,” Bruce said.

“If you don’t have the latest version of the hub, you can work with developers to simulate it, figure out ‘What parts of this thing should I simulate?’ You may have physical devices but not the hub,” so you can build a simulation with SoapUI NG Pro. “As the device, it satisfies the listening agents in order. It’s just a chain and I can start from the end and work to the beginning. The final step receives the information as the hub. The device heard what the hub said and published a final status check,” he continued explaining. This is the sort of automating that can be done in the Internet of Things to test for functionality and API stability, making testers more efficient and more able to focus on making sure that the device is suited to its audience with the right user experience and the right level of privacy and security.

Kruk said that “SmartBear provides testing for the glue” that holds the Internet of Things together.

Fuzzing the Internet of Things.

Fuzz testing or fuzzing is a software testing technique that’s highly applicable in the Internet of Things, where you just throw anything unexpected into a test, having fun and getting creative while trying to break or infiltrate the software. “You’re throwing all this data in until you find something bad. ‘OK, I got something back that I shouldn’t have got back,’” Knopf explains fuzz testing.

“What can I do with that? How can I use it in an attack?’” offering the example of “I can see if I can get around the login function somehow and send different data to it.”

SOME FUZZ TESTING TOOLS THAT KNOPF SAYS COULD APPLY TO THE INTERNET OF THINGS ARE:

- **Peach Fuzzer, which allows you to send all kinds of data to APIs**
- **BurpSuitePro, which captures all the packaging you are using for testing and then manipulates it through the data and sends it in different ways to test functionality like log-in.**

Parker and Altitude Angel use fuzz testing in their suite of testing. "We can't trust that [a drone is] only going to be within the expected range or altitude." He says that as improbable as it would be to have a drone fly 30 miles high, they have to be sure their software can deal with that improbability because when you bring in the human element, you just never know. "We perform different types of testing so, for us, every device is untrusted. We don't vet the owners of the device."

Altitude Angel has a machine learning analytical component that is "recording all the flights that take place so we can pick out a particular zone and consumer — some will do fun things," allowing for fuzz testing and testing to be more and more based on the real-life actions of real people.

> **"We are able to take real-life flights that are recorded in Altitude Angel and then play them in our test simulations."**

Their simulation system applies machine learning in order to tweak the avoidance algorithm constantly, reflective of emergent behavior, maneuverability, and changing regulations.

PARTICULARLY, THEY DELIBERATELY TEST FOR:

- **collision of drones**
- **inner warning systems**
- **when a drone collides with a no-fly zone**
- **what to do when inclement weather approaches**
- **if you lose connectivity, what should happen**

As part of their testing automation and machine learning, Parker's small team even runs in-house contests, splitting the teams into red and blue, attacking each other, which not only builds a good team ethos but has them competing to attack and defend, finding flaws that are added into every test from now on.

"We've programmed the system to detect when a drone is in flight and is sending us updates, and then stops." Altitude Angel has drawn a ring of responsibility around the drone with an outer midpoint and inner warning system, each with its own dynamic trigger points for risk calculations, each type of device having its own set of minimum depending on location and circumstances. Altitude Angel can increase and contract those boundaries every second. "It is the drone's responsibility to get communication with Altitude Angel, but it's our job to keep drones safe," which means not only keeping drones powered by their software safe, but to notify other drones to back off when needed.

Using root-cause-analysis to detect patterns of error.

Kruk and his team at Crowsnest are working to automate the testing of the big variable in the Internet of Things that we haven't touched on yet — the hardware.

With Crowsnest's root cause analysis, he explains that "We monitor the window of time leading up to each issue. When we're monitoring millions of your units and products, we can start to identify patterns that typically lead to problems. Once we've seen a particular pattern enough times we can start to flag it and notify you before it causes problems on your other 995,000 units: 'This type of event is guilty by association 90 percent of the time, that might be a good place to look for the root of the problem'."

The idea is to notice trends across products out in that Wild West — like that certain covers around a connected light bulb can cause it to overheat or that firmware needs updating — and then notify customers for how to optimize their experience. They're even partnering with another app that could automate certain fixes, like rolling back updates when necessary.

"If we're monitoring millions of units, and, let's say, we see 500 of them all have the same issue. We'll notify QA with the information they need to recreate the scenario in their lab." The information is always available in a Trends dashboard, which ranks your most common issues and the events that preceded them.

Crowsnest analyzes the data being generated by the operating system running inside the device itself, identifying where, when and how the devices are failing in the field. It also has a dashboard that allows the client to identify which customers have been affected as a result of these failures. Crowsnest — named after the lookout atop a pirate ship — is a monitoring tool, which means they track results, but don't know any information about the customers.

By tracking all of this, companies are able to automate a big chunk of customer service, which is a clear cost cutter.

> "We're actually monitoring the code that's inside these products. We know immediately if something goes wrong, often before a customer knows."

Of course, we all know big data, not the sale of devices or of many, many more data plans, is predicted to be the real way to make money out of the Internet of Things.

"Because we're monitoring so many different units, we can collect a wider variety of data and use that to train a more intelligent service. With websites, you're monitoring one server that interacts with tens of thousands of customers. Often in

IoT the relationship is closer to one device per one customer. We realize this means opportunity for better customer support and customer service. We're optimized for this type of relationship and it's where we're focused on adding more value," Kruk said. "Our customers are seeing new opportunities to improve their product and they've told us we cut their customer service time in half."

A root cause analysis may have to be done to get to the basics of all software testing. "With things coming online, there is a need for a team to go in there and do a root cause analysis of everything there and take it down to the absolute basics and then build it up again," de Grazia said. "Instead of saying you out in field tell me what do you have. When did you connect it to the Internet. Why did you connect to the Internet? What kind of security did you put on there?"

> "The number one objective of IoT manufacturers is to avoid another recall."

"We're really good at finding critical things that you should be paying attention to in the code, but it's not only the code. More often, it's things in the environment or unexpected user behavior that causes issues," Kruk said.

Crowsnest uses natural language processing, which is certainly a growing trend and a strong candidate for sorting out the complicated world of the Internet of Things. The sheer volume of data that is already coming with the Internet of Things creates an opportunity to provide more intelligent service and to create more predictive analytics, not only like Crowsnest in devices, but in human behavior overall.

As Kruk said, the future is about predicting errors before they happen, which is why they will start to proactively automate the notifying of the end users as well.

How does the Internet of Things change the role of the tester?

If the impact of the Internet of Everything will really be as Cisco predicted, ten times the impact of the Internet itself, that means it doesn't just change people's lives, but it also dramatically changes the whole development cycle. Developers, designers, testers, management, and entire companies will be reorganizing as they try to fit the Internet of Things into their workflow and, more likely, have to adapt their workflow to IoT.

While we can't fully predict how, we're pretty certain that the role of the tester will become even more central not only to the product cycle, but to business development as a whole.

"We are fascinated by technology for technology's sake." Bolton calls testers the important investigators and essential critical thinkers of the tech world, needing to be constantly conscious of how technology changes us. It's their job to remind developers that it's not just "Oh look, shiny!" but that the shiny especially should be investigated.

"I think there will always be a role for investigators, journalists, critics, and that to a great degree is a role that testers fulfill in a project. Software will increasingly be checked through tools; through automation. What can never be automated is the investigation of social fit. Does this product fit into society? Is it good enough? Does it fulfill our intentions? Are there undiscovered intentions?"

Field testing becomes essential for QA to test user experience.

Bolton describes the user experience with all technology in a way he finds essential for testers to think of while they are investigating UX: "A new piece of technology is a new medium in the Marshall McLuhan sense. It enhances, changes a human capability. Every piece of technology gets between humans — it doesn't completely replace like an artificial heart." He offers up that "it is the job of human testers to repair the difference between the device and the purpose, the position."

"The tester finds and reports on important differences between what the product does and the human purpose that it is inserted into. And there's a complex social decision in that: Will humans be sufficiently happy in the repair work they'll have to do to adjust?" Bolton further explains.

The role of the tester is no longer just testing for specific, measurable outputs, and, in many ways it never has been. He continues, "There always has to be testing also for how well does this product fulfill the social purposes and to what degree does that product not fulfill."

Many roles will shift and you will even see QA in the field because what better way to test usability, user experience, and for use cases you never would have dreamed of than to use a bunch of actual human beings. After all, testing for IoT will never work if twenty-something guys are the only ones testing devices meant for elderly women. Field testing will become a much more important part of the testers' world, not just, as Trifa put it, "having one guy at one desk with a device but literally have a much wider, real-world test scenario."

Menon and his consultancy agency specialize in the human experience around IoT, working with how devices communicate and how users use them, testing for the UX and backing it up with end-to-end process experience testing.

"For IoT it's really about understanding how people are using them. So many different protocols, so many different options," Knopf said. "If you are going to be in the IoT space, you really have to understand the pain of the customers," which necessitates more than ever dogfooding your products.

This is especially true because IoT devices “are real. They are 3-D. They are both more visible and at the same time more invisible. If it’s on your hand, it’s tangible and you see it all the time,” Trifa said, making user experience vital to the success of an IoT product or service. People use and try out wearables and other devices on their body much faster which means that they will be much faster to return it. “It’s physical. They have to really use it, sleeping for months with that thing on their arm, and then actually, after a few weeks, it gets hard because I sweat or whatever.”

Trifa says that testing IoT involves a bunch of tools and methods that “normal QA people don’t have because it’s actually design and ergonomics. Some might be really good when it comes to usability, but this is about people, not data and numbers. It’s not just, ‘Oh Yes it looks nice,’ but ‘Hold on a sec, my grand-mom can’t use that device’.”

Trifa says that IoT testers need to be like Apple, which he says their “result makes it look like instead of spending their budget on marketing, they spend it on testing,” pointing out small things like never having seen an obscure error message on his iPhone like he has on his Android. “To get something so right and reliable and robust is of course an amazing engineering feat and also certainly an amazing testing feat,” which is why similarly he suggests that IoT have longer release cycles to ensure better things are getting released.

“The more intimate those products become, the better you want to feel about them,” he said. Plus, “If people are spending more, they expect better QA. That’s why they can charge more for the iPhone.” Of course, dogfooding is more challenging when you can’t really create a full test simulation.

Trifa points to how South Korea’s fully connected Songdo is perhaps the best test bed out there, “because you just cannot test or foresee everything that can happen is connected to the internet,” he explained. “When you’ve built such a city from the bottom up without understanding the problems or risks, those are the best test beds,” pointing out that, “Yes there will be nasty moments but it’s better than testing in the lab.” “The world should be your own testing bed,” Trifa says. “So get out there and play!

Gerrard echoes the importance of testing IoT at its massive scale.

> “Here’s a device, I’ll do my hundred tests, but then I have to test that device in an environment where there are thousands of other devices in hundreds of other potentially interfering networks.”

Knopf pointed out that all sorts of testing, whether for security, user experience, or something else must involve building realistic testing areas, which address the different types of users. He lumped them into the three sample user bases:

1. **college dorm — low number of devices, small place.**

2. **medium-sized family, with some tech.**

3. **über tech with 50 plus IoT devices.**

Now ask yourself: Can we get the device to work in all three scenarios? He pointed out that "Scenario Three is hard because you have a lot of things dependent. It can only handle eight [devices], but we are doing 30, 40, 50. Routers aren't made to handle those" loads.

Of course, for user experience to be an important part of IoT testing, that means that QA has to work with designers and developers from early on.

"As QA, you have to know more about the application than developers do," Knopf said. "Developers are responsible for a small part and you have to be able to tie that into real life."

IoT pushes QA to the left.

There's a lot in the press focusing on the negative consequences of the Internet of Things — and surely it raises some privacy and security concerns — but there are positive results besides how it will integrate and automate our world. IoT is changing the way teams design, build, and test. IoT is helping to build better, stronger, more efficient and thorough teams.

IoT further eliminates what Bolton calls the "Hot Potato Theory of Testing" — the you-had-it-last syndrome. "To suggest that a tester is responsible for a bug, that suggests that the tester is responsible for every aspect of the product," he explains. "Responsible people understand the premise that a product is a result of work by many people [and] problems in a product are the result of many people."

He says this is a part of a greater shift in software development, where programmers are also taking responsibility over more of their work.

Gerrard said that the tester should be treated like a trusted advisor. He compared the relationship between developer and tester to a young king and an older advisor or a pilot and navigator system. "In some ways, the developer's the hot shot. The developer just wants to write code, he's not thinking beyond. Now the tester must ask: How will it work in isolation? In an integrated system?"

He echos Bolton that, in the Internet of Things, there is just no room for what he calls classic waterfall: "It's not my problem, just pass it down the production line" because in IoT that'd be simply too much of a financial risk. IoT forces testers to move further left and enables more cross-team collaboration than we've ever seen before. Like in scrum framework, and in the increasingly popular microservices trend, the roles between developers and testers will become more blurred and testers will be sitting in on development meetings and even joining development teams.

> "Testers are there to protect the interests of the stakeholder but also to protect the interest of the developers, Gerrard said."

Conversely, developers for the IoT space have to design more with the testing in mind. Gerrard says to "Design for testability because the problem won't be in building them but in our abilities to test them," pointing to the fact that it's not hard to design sensors, but to build them for a smart city, it is a bit like an operating system — immensely complex.

"I see the role of testers shifting from people who design and run tests of existing software to becoming Worriers and they just spend their time asking kind of awkward questions," even joining DevOps day-to-day.

Menon observes that "the horrendous complexity that's on the way is promoting a new model of testing as a way to get into the heads of what testers do. It's a picture of the different thought processes in your mind — 'If that's the way testers think, crikey, now on left hand side that's exactly how developers think too'."

And it's not just developers and testing coming closer. "Marketers are getting the budget to build digital transformation projects, going from bricks and mortars to online. Their mentality and approach to life is quick market testing. The implementation cycle is very short." Menon says that the whole business world will experience a change with rapid release cycles and continuous delivery, "shifting left and bedding with developers and joining their world." And testers won't just bed with developers but designers too.

Trifa argues that, at least in the IoT space, quality analyzing starts with testing the design. "Don't read the spec at the end but you want to make sure the QA guy that's going to be testing is in the same room as the designer up front. Equally, the QA person understands better who the people are who are going to use it. It's all about a much better collaboration between the two."

He went on to say that it's not just going to be QA at the end but it's going to be many smaller cycles throughout the process, starting with QA on-board with prototyping: "It's going to be many more smaller cycles, literally in a giant scrum process of design," putting QA into the early phases of design and development. "Iterative design and improvement will just make things so much better."

Bruce echoed his contemporaries saying,

> **"For the Internet of Things, testers are your superstars. They're the ones that are going to be identifying what went wrong."**

He continued that

> **"getting that feedback from those testers is going to be vital to be sure that the dev and the ops processes are truly rapid release."**

Even at giant Microsoft, there's movement away from trying to monopolize the market and toward everyone participating. "We don't care what devices you're using, and if we can get Windows on your devices, that's really cool. We just released Windows for Raspberry Pi 2. That's cool too — we can meet you in the data analysis in the cloud even though we are a device company," Mulcahy said. "I think in the long haul, the Microsoft story is 'Yes, we are very much aligned with enterprise and we have the ability to support thousands of devices without messages being able to be dropped, support[ing] security'."

But companies like Microsoft have to be aware of what those devices that connect will be and how people will use them. Only then can they test for almost every situation. This is where testing doesn't seem so changed at all.

How IoT will bring testing back to the basics.

“These days, we’ve got everybody using machines, but the basic principles for that remain the same but so much of this has already happened and now it’s just a matter of scaling it a little bit more,” Bolton said, admitting that the “problem of distance and height of stack is always getting worse.”

He argues that the object and concept of testing evolves a bit with the Internet of Things, but the basic principles behind testing don’t change — you plan it and build it, and then you study what you built — plus complex technological systems have been around for a while anyway.

Justin Rohrman, VP of the Association of Software Testing, sees the Internet of Things as the impetus to return to our more philosophical routes and the testing fundamentals, where there will be so much needed to be tested that testers need to take logical steps and even shortcuts to get the desired result without missing anything.

“A common notion in software testing is that you have to have some form of expected result to determine if there’s a bug or not — most frequently a specification. Heuristics allows us to discover bugs without specifications,” Rohrman said.

Heuristics isn’t a replacement for manual testing or automation but a set of rules of thumb or shortcuts that are occasionally wrong but very well compliment testing, especially when there’s such a rush to market.

“By observing your behavior by testing and also the information around you [which] comes from developers, product managers, documentation, competing products — all of these feed us information of what and how we should test and what is important and not important,” Rohrman explains.

He pointed out that some of these heuristics we already know will apply to IoT, while “there may be specific new things that we learn that only relate to IoT, [like] what is a problem and what is not.” Rohrman argues that going back to the basics will enable testers to deal more easily with the vast differences they are facing in the Internet of Things.

“Testing a smart thermostat is going to be way different than testing some kind of embedded system in a car [which is going to be] different than healthcare.”

> “The fundamentals of software testing are what will make people strong in this area: problem solving.”

Lane also pointed out that many of the values of basic API management apply to the Internet of Things, including the need for a clean, simple portal, easy documentation, and code libraries in multiple languages. But the risk is that many developers are coming from the sensor and hardware business.

> **"They don't really get APIs and it's not just RESTful, how do we not build black box hardware and SDKs. We need to crack 'em open. This new API world lets them open it and tinker with it."**

Again, this means more power to the testers.

Where do we go from here? Collaborate or perish.

The Internet of Things will either polarize us as we play the "blame game" or it will open up software collaboration forever. Repeated throughout this eBook is the blurring of the lines of separation. "Building security is about getting researchers and manufacturers together and letting them work together without budgets," Knopf said. He cofounded I Am The Cavalry as a way for quality analysts to collaborate over testing cars, medical devices, fire alarms, and "things that could really impact people's safety." BuildItSecure.ly is another volunteer organization that's trying to help Kickstarters in the IoT space: Why don't we help you, send us a couple devices and we'll tell you how to make it secure, Knopf said, pointing out how they get to fill in the swiss cheese security holes, while getting to play with cool new devices.

SmartBear's own Bruce advises that "in order to really benefit from open standards, you need to contribute to them, provide feedback. It's our responsibility to do what we can [and] think about how the devices and services are going to be used and misused." And in the open source spirit, Internet of Things testing is about publishing and sharing as much information as possible. Lane says when you are testing,

> **"publish your strategy and plan and share it with others so they know that it's executed and so they can emulate it."**

By sharing it actually makes us all more competitive with greater potential for success. "There's more eyeballs on what's going on, thus we can find bugs faster, [with] more feedback loops for security flaws," said Lane. He is optimistic that as long as organizations are properly open and sharing, security can be enhanced in the Internet of Things. "Open communication, transparency in your pricing, how you operate, discoverability via API JSON. Using SmartBear, I can just discover all API JSON files in a network and where they point to it."

Lane advocates for a focus on testing automation, security, and, especially, discoverability through openly communicating through an ecosystem. "A big reason why systems get hacked is because it is just one loan machine sitting on a network. You may have two thousand APIs. That one that isn't known gets hacked."

"The platform providers should have standard practices that they test, and they should share and be transparent about what their practices are. 'Here's what we scan for,' passing the torch to the developers, consumers or platform integrators." Lane contends for publishing a standard map of each interface, using an API JSON index or directory "so I can automate some stuff, but then look for other holes because I don't want it to trickle over into my business."

So what are you waiting for?
Get out there and share!

Start by tweeting to @SmartBear your Internet of Things testing experiences!

In the Internet of Things, it's never been more important to have the right tools for testing your API.

Whether you're looking to automate functional testing, load testing, or security testing — SmartBear's suite of API testing tools will ensure your team is ready for the challenges and opportunity of IoT.

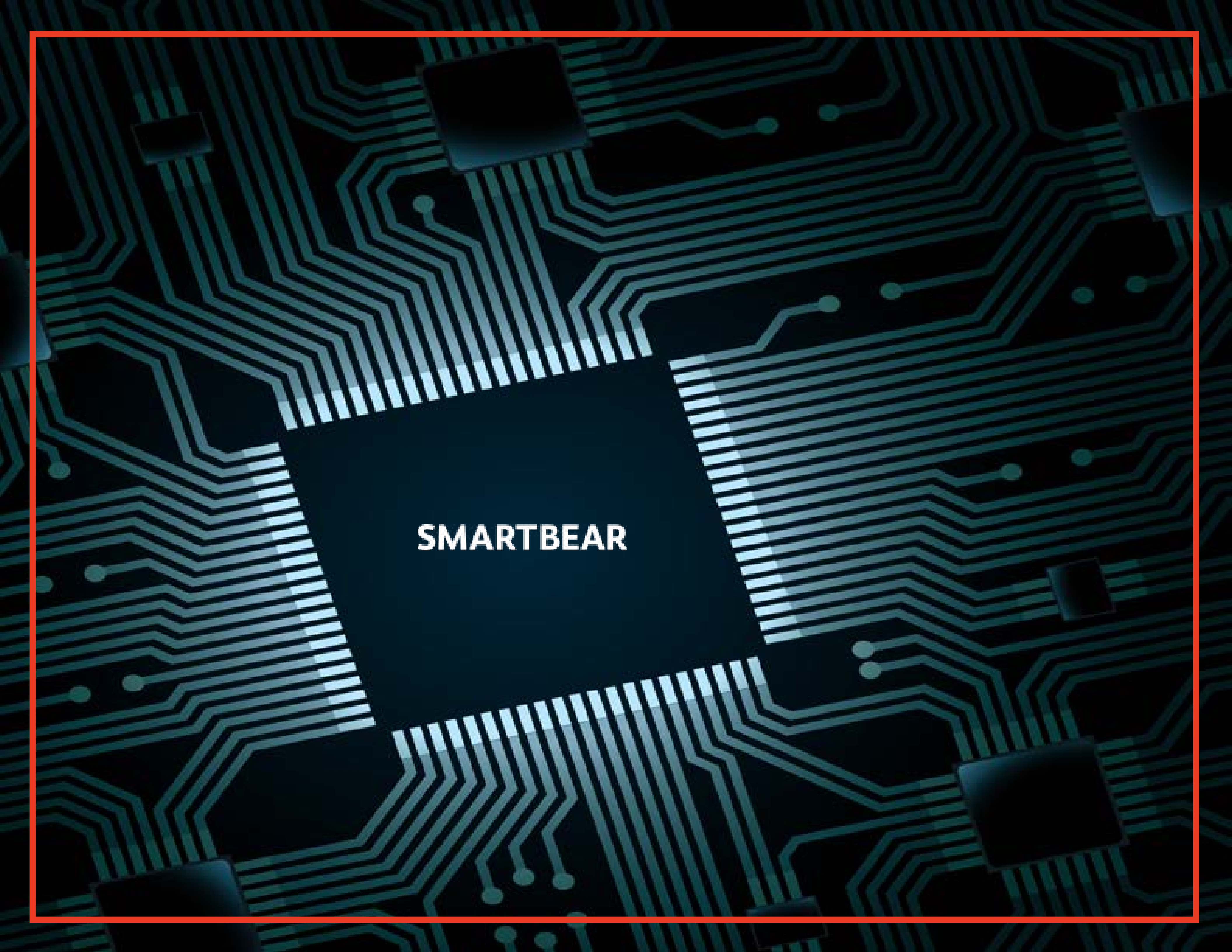
VISIT SMARTBEAR.COM TO LEARN MORE

Author’s Note

Have fun! If there’s an opinion I developed from this adventure in interviewing all these very different testers and IoT influencers is that while testing in the Internet of Things is as green as the industry itself, the principles of testing are more important than ever. More than anything, curiosity is key.

For IoT, us writers are going to need a whole new way to say “think outside the box” because that’s truly the best way to describe your job. But that shouldn’t looked at as a problem, but more like a really fun change to get creative and enjoy your job, while having the bonus of really affecting people’s lives.

Jennifer Riggins, ebranding.ninja



SMARTBEAR