Altium®

IoT Security Best Practices

IOT SECURITY BEST PRACTICES

Printed Circuit Boards for IoT systems require special attention. Everyone knows how vulnerable IoT can become without the right security measures in place to ensure safety and compliance with regulations. Discover best practices to help avoid issues and vulnerabilities during board development and learn about security gateways, design options like IoT self-testing, and authentification.

Join us as we discuss a variety of topics to help you with IoT Security, including:

- Internet of Things Security Best Practices: Passwords, Patches and Portals
- Internet of Things Security Issues Prompt Government Intervention
- Internet of Things Security Vulnerabilities: All About Buffer Overflow
- PCB Technology Trends: The Benefits of Internet of Things Security Gateways
- Security by Design: Internet of Things Authentification and Self-Testing

INTERNET OF THINGS SECURITY BEST PRACTICES: PASSWORDS, PATCHES, AND PORTALS

If you've ever created devices for military applications you're probably used to designing with security in mind. If you're more like me, and just build the odd Internet of Things (IoT) doodad, that's probably the last thing on your mind. However, recent cyber attacks are highlighting why engineers like us should be a little more concerned with security. These design best practices will show you how you can safeguard your systems.

WHY SECURE THE IOT?

On the face of it, most IoT devices seem harmless. Now, though, even the most innocuous gadgets can become a weapon in the hands of a hacker. These malicious programmers can use malware to infect poorly secured devices and add them to massive botnets. Botnets are then used to perform distributed denial-of-service (DDoS) attacks, which can take down web services.

Hackers don't need a sophisticated computer to perform a DDoS assault. They just need a device that connects to the Internet. Interestingly, IoT devices make up a huge portion of these botnets, because most have poor security. The Mirai botnet, which took down some DNS services in 2016, used thousands of Internet-connected cameras and DVRs. If you don't want your inventions to end up as a soldier in some hacker's computer army, you should implement a few safety measures.

Botnets, or zombie armies, take down web services with DDoS attacks.

NO HARD-CODED PASSWORDS

If you design embedded systems, you've probably used a hard-coded password before. Depending on your software architecture, putting passwords in read-only memory is sometimes the most straightforward choice. Unfortunately, it's also dangerous.

In the realm of industrial IoT, many supervisory control and data acquisition (SCADA) systems have hard-coded passwords. This presents a huge risk, as these systems often control things like power plants and energy infrastructure. In fact, in 2010, hackers were able to attack one of Iran's nuclear facilities because they knew some of the passwords that were hard-coded into the equipment. If you're designing an IoT device for a critical system, like a power plant or a self-driving car, you need to let the user change the password. Otherwise, the results could be devastating—even deadly.

Even if you're just designing light bulbs, you should still allow users to change passwords. Many of the IoT devices in the Mirai botnet had hard-coded passwords. Since these things are being made en masse, a hacker can take over thousands of devices simply because they know one password.

Hard-coded passwords make devices easy targets.

ENABLE PATCHING

Even if you rigorously test your device in-house and think you have covered every base, vulnerabilities will be discovered in the field. If you can't patch, update the code for your device over the air, or make patches available to users, your gizmo could be a sitting duck.

There's currently no financial incentive to build your device so that it can be patched, though in the future there might be. The US Senate recently introduced legislation that gives designers guidelines on how to make our devices more secure. One of their requirements is that IoT systems be patchable. If we're going to have to design for this in the future, we may as well do so now.

USE A GATEWAY

Up until now, many IoT devices were designed to connect directly to the Internet or to a user's computer or phone. As the IoT grows, this method is untenable for multiple reasons, including complexity, increased processing, and—most importantly—poor security.

When thinking about cybersecurity and the IoT we need to consider the "attack surface." In a large, low-power wide-area IoT network you may have thousands of devices or sensors connected. If they're all connected directly to the internet, you will have an enormous attack surface, and a hacker could gain access to the network through any one of those devices. Using an IoT gateway can help you consolidate security and present a smaller target to potential intruders.

A gateway acts as a communication node for an IoT network or system. Usually, these will be able to connect to devices using a variety of standard protocols, like Bluetooth, and may have some onboard processing. Since most or all information is routed through the gateway to the Internet, it presents a smaller target than thousands of different devices. Then you can focus on securing a handful of portals instead of an army of sensors and other gadgets.

IOT SECURITY BEST PRACTICES

The Internet of Things will certainly be an amazing tool for societies around the world, but it is also becoming a dangerous weapon. Poor security features have allowed hackers to attack critical infrastructure and co-opt devices into massive botnets for online assaults. As a designer, you can help mitigate this threat by allowing users to change device passwords, making your products patchable, and integrating your devices with IoT gateways.

Admittedly, you have many things on your plate, and these fixes aren't always easy to implement. That's why you should think about using PCB design software like CircuitStudio. It has a host of tools that were created to help you during design.

Have more questions about IoT security? Call an expert at Altium.

INTERNET OF THINGS SECURITY ISSUES PROMPT GOVERNMENT INTERVENTION

When you look at an Internet of Things (IoT) device like a fork or a juicer, what do you see? Like me, you probably see a gadget that was designed for a particular purpose. Less upstanding citizens may see a digital weapon instead of an innocuous gizmo. In the past several years, there have been several high-profile distributed denial-of-service (DDoS) attacks that were enabled by poorly-secured IoT devices. Hackers infiltrate light bulbs and other smart household items and incorporate them into huge botnets that can then be used to take down online services. These assaults recently led to the introduction of the Internet of Things Cybersecurity Improvement Act of 2017 in the US Senate.

A RECENT WAVE OF IOT BOTNET ATTACKS

If you're not familiar with botnets, they may sound like something from a science fiction novel; however, they're quite real. They can shut down company websites. This past year hackers were even able to bring down huge parts of the Internet using IoT botnets. This threat is not coming but already here.

First, let's talk about distributed denial-of-service (DDoS) attacks. Every year my college football team sells its season tickets on a single day. They always sell out, so people are usually waiting at their computers for the sale to open so that they can buy their tickets. However, with so many people trying to use the service at once, it usually crashes. Think of it like a traffic jam on a roadway. If

there are too many cars (or computers) nothing moves. This is the basic idea behind a DDoS assault. Hackers will take over computers or, in our case IoT devices, using malware. These malicious programs let the hackers control the gadgets (also known as zombies) and overload a target website.

When my college's ticket website goes down, it's not really a big deal. They still sell all the tickets, but it just takes longer. If hackers cripple a payment website, though, companies can lose hundreds of thousands—even millions—of dollars while the website is offline. These attacks can also be used to incapacitate portions of the Internet as a whole.

In 2016 the Mirai botnet attacked a company called Dyn, which runs many US Domain Name Server (DNS) services. As a result, many people in the US were not able to access the Internet. To mount such a massive strike, the Mirai botnet used hundreds of thousands of poorly-secured IoT devices, in this case mostly webcams. Many other botnets are being built from smart gadgets with substandard security as we speak. The United States Government has been slow to acknowledge this threat but is now taking action.

Botnets can be used to overload websites or other online services.

US LEGISLATION TRIES TO HOLD BACK THE TIDE

Recently, the US Senate introduced bipartisan legislation to correct this problem, which would require designers to meet certain security standards in order to sell their products to the federal government. These standards compound the challenge of meeting the FCC's radiated and conducted emissions standards. However, the proposed act's standards highlight weak points in IoT device design and indicate some key areas for designers to consider.

Primary standards of the proposed legislation:

- Devices should be able to receive and install a software patch. Either enable over the air updates for your product or make it possible for the user to install patches.

- Designers should avoid integrating known vulnerabilities into their products. If they discover a weakness during design, they should disclose it to the appropriate federal agency. Don't intentionally include backdoors into your hardware or software.

- IoT communications should rely only on standard protocols, such as Bluetooth or 5G.

- No device should have a hard-coded password, which is part of what allowed the Mirai botnet to infect hundreds of thousands of devices, like WiFi light bulbs.

The legislation does not appear to be extremely rigid. There will be allowances on a case-by-case basis for gadgets that don't meet their requirements. That being said, the risk of IoT devices being infected and grafted into massive botnets is significant. Even without that legislation, designers should seek ways to more effectively secure software and hardware alike.

Don't let your light bulb become part of a zombie army

Designing a PCB is never easy, and these new government requirements won't make it any easier. Fortunately, great PCB design software can help you stay ahead of such legislative action to secure your designs. CircuitStudio boasts a wide variety of tools that will help you manage the details and craft IoT devices that reflect increased security standards.

Have more questions about IoT security? Call an expert at Altium.

INTERNET OF THINGS SECURITY VULNERABILITIES: ALL ABOUT BUFFER OVERFLOW

I follow a bunch of animal and nature publications, and recently the phrase 'zombie ants,' kept popping up in my feed. I decided to do a little digging and discovered that there's a type of fungi - Ophiocordyceps - whose life cycle involves infecting ants that walk across its spores with fungal cells that infiltrate the ant's central nervous system and essentially take over the ant. Once they have the ant under their control, they make the ant climb a plant stalk, bite down on a leaf, and then grow more of the fungus from the ant's body to then spread more of its spores, overflowing the region with more potential fungus and zombie-ants.

It seems a lot like the way hacker groups steal their information from large companies and even governments. They attack websites and services using botnets made up of thousands of Internet of Things devices. These gadgets have been turned into "zombies" that do their master's bidding by malicious software (malware). Defending our networks and products from becoming part of these armies is a daunting task as software exploits can come in all shapes and sizes. Luckily for us, though, there is one vulnerability that is often used which we can protect against: buffer overflow. This error allows hackers to inject their code into our PCB's memory and then execute it. Careful programming can reduce the risk posed by buffer overflow, and following IoT security best practices can limit attackers ability to attempt buffer overflow.

WHAT IS BUFFER OVERFLOW

Buffer overflow was first widely acknowledged during the "Code Red" attacks in 2001. These assaults used buffer overflow in Windows to take control of computers, one version infecting hundreds of thousands of machines in a matter of hours. Once infected those computers were then used to launch a distributed denial-of-service (DDoS) attack on the White House. Since that time buffer overflow has become one of the methods of choice for hacker groups seeking to infect devices. In order to defend against this vulnerability, it's important to understand how it works and why it poses a particular threat to the IoT.

This kind of error occurs when a program tries to write a value that is too large into a buffer. A buffer is simply a piece of memory allocated for certain values. When the program attempts to fill it with more data than it can hold, the buffer "overflows" into other sections of memory. This usually results in a system crash but can also open the door for hackers to gain access to the system. Hackers can use a buffer overflow to do two things: inject their code into a system and run the injected code. The first can be quite complicated and is system dependent. The second, however, is quite easy to understand. If malware has been inserted into memory and the hacker knows where it is, they can simply overflow the buffer next to it in order to run that program.

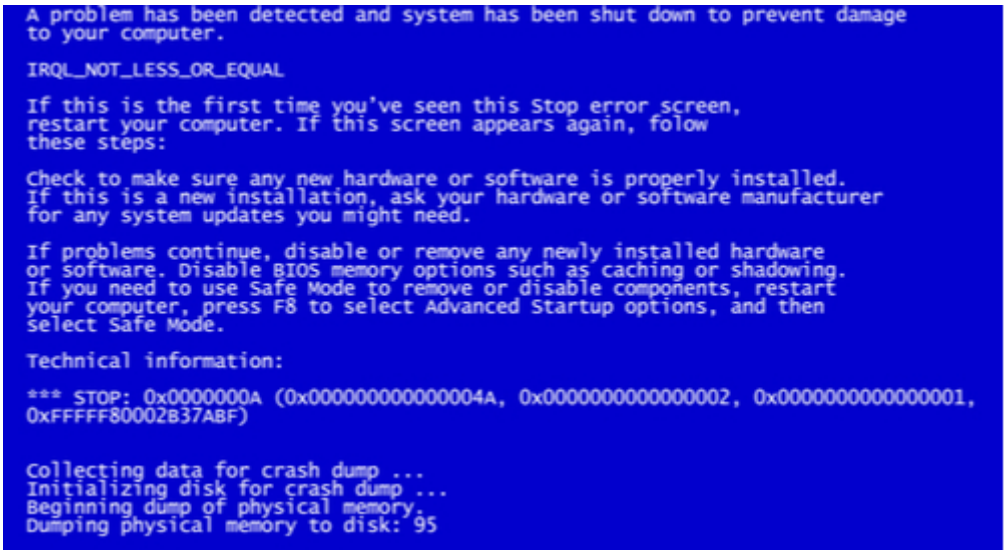
THE WEAK LINKS TARGETED BY BUFFER OVERFLOW

Buffer overflow poses a particular risk to IoT devices because of their limited memory, the languages they're programmed in, and the commonality of programs.

- Memory: IoT devices usually need to conserve power, which leads to small amounts of energy efficient memory. The smaller the buffer, the easier it is to overflow, which makes the IoT the perfect arena for these kinds of attacks.

- Language: Most programs for the IoT are written in C or C++. Neither C nor C++ has a "garbage collector" which increases the risk of buffer overflow. Also, these languages use pointers, which can be used by hackers to determine the location of critical code in memory.

- Commonality: The convenience of buying ready-made, inexpensive, programs for our IoT devices can be too tempting to pass up. When you use the same code as everyone else, though, you run the risk of having common vulnerabilities. One such exploit, known as Devil's Ivy, was found in software used by thousands of IoT devices and just recently revealed. Many machines escape infection through obscurity. If your product can be infected along with thousands of others because of common code, it's much more likely to be targeted.



A buffer overflow can be used to gain access to or crash your device.

BAR YOUR PROGRAMMING WINDOWS, PATCH YOUR SECURITY GATEWAY WALLS

Now that we know the dangers that buffer overflow poses, how do we defend against it? There are several ways to mitigate this risk. If you're writing your software yourself, careful internal programming can keep your device safe:

- Check Input Sizes - If you know how large an input should be, check to make sure it is that size. Buffer overflow occurs because a value that is too large is written to memory. If you can detect the size before passing it to memory, you can reject values that are too large and will cause overflow. This may not be applicable to all systems, for example where incoming sensor data may be an unknown size.

- Make Memory Non-Executable - As stated previously hackers will often hide malware in memory and then use a buffer overflow to execute it. If the part they inject code into is non-executable it could keep them from activating their program. Due to the diversity of buffer overflow attacks, this can stop some intrusions, but not all.

- Use ASLR (Address Space Layout Randomization) - As the G.I. Joes always said, knowing is half the battle. If a hacker knows where critical code is stored, they may be able to overwrite or delete it. ASLR randomizes memory locations, making it much more difficult for attackers to find their targets.

Even if you use someone else's programs you can insist on a few best practices or enable them yourself that will defend your system. Some of these may soon be mandated by the US government:

- Enable Patching - If you find out that your software is vulnerable you'll need to be able to patch it. If you're not able to your gizmo could become a "zombie" in a botnet for the rest of its existence.

- Gateways - If you're designing a peripheral or a sensor made to operate in a large network, consider designing for interoperability with an IoT security gateway. These can reduce the likelihood that a hacker will assault your device directly, and will instead have to deal with a portal that is designed for security.

- Authentication - Many buffer flow attacks are attempted using "Man in the Middle" (MITM) schemes. Authentication will ensure that your system only receives inputs from trusted devices, not malicious pretenders.

You can defend your system with careful code and best practices.

The IoT can benefit mankind greatly, but also presents a huge new attack surface for hackers. Buffer overflow assaults have been used extensively in the past, and are perfectly positioned to affect the IoT. Fortunately, we can either write code that protects against this vulnerability, or incorporate a few best practices into our design to mitigate this risk.

Cybersecurity is a daunting issue, and one that will likely take up more of our time than we'd like in the future. With this issue looming, there's no time to waste on inefficient design. That's why you should use the best PCB software available. Altium Designer comes with a range of tools, and even optional add ons, that can speed up your design process.

Have more questions about IoT cybersecurity? Call an expert at Altium.

PCB TECHNOLOGY TRENDS: THE BENEFITS OF INTERNET OF THINGS SECURITY GATEWAYS

Do you ever like to take a break from social situations for a few days and spend some quality time with your television? I do that maybe a little more often than I should, and last week I spent that time binge-watching the new season of Stranger Things. If you haven't seen the show, it's about a girl with psychic powers who opens a portal to another dimension (the Upside Down) filled with creatures. The gateway to that world lets monsters loose, but in the Internet of Things (IoT) gateways, or edge devices, can help you control monstrous devices. These gateways can help connect modern and legacy devices together, and they can funnel information from local area networks (LAN) to the wider Internet. There are three main advantages to directing IoT traffic through one portal: enhanced security, simpler connection protocols, and centralized processing.

WHAT ARE IOT GATEWAYS?

You don't have to be as smart as "Bob the Brain" to understand IoT gateways. These devices have been invented to control the imminent flood of IoT gadgets. They can help us make sense of a distributed network of sensors or other devices by acting as a connection node for the system. In order to be useful as a gatekeeper, these gizmos generally have processing capabilities, storage, battery backup, an Internet connection, and can communicate via multiple different protocols.

The diversity of the IoT is one of its greatest strengths but could become a crippling weakness. With so many companies making

different devices that use a variety of communication protocols, creating a coherent system could become difficult. Gateways help solve this problem for designers by giving us a single access point that we can route information through.

In order to do its job effectively, an IoT portal will need several features including: local processing and storage, connection to the wide area network (WAN), reliable power, and multi-protocol communication.

- Local Processing and Storage - While you may want to use the cloud for your heavy processing needs, your gateway will need an onboard microcontroller and storage in order to parse and collect data.

- Internet Connection - Most large IoT systems will eventually route information online so that users can view their data. Every device hooked up to a portal may not have connection itself, so it's critical that the gateway be able to upload to the Internet.

- Reliable Power - Most sensors in an IoT network are low power so that they can operate for days, or even years, without having their batteries replaced. Your edge device will be constantly communicating with the sensors and parsing data, which will take a lot of energy. A good portal will have a battery backup or a low power state so that it never completely loses functionality. You don't want your edge device to flicker like Joyce's Christmas lights.

- Broad Connectivity - The IoT will use a variety of networks and protocols to connect sensors. In fact, you may want to mix and match protocols in your system, which is why it's imperative that gateways be able to handle any and all communication methods out there.

An IoT gateway can act as a security guard for your networks.

BENEFITS OF EDGE DEVICES

IoT gateways provide a variety of extremely important benefits for sensor networks. They help improve the security of systems that can be as vulnerable to attack as poor Barb. Since edge devices act as a data node, they can also be used for centralized processing. Lastly, their multi-protocol communication capabilities can allow you to connect any sensor you choose, even extremely old ones.

- Security - Cybersecurity is extremely important in all embedded systems, from cars to sensors. The primary benefit of using a gateway is that it reduces the exposure of your system. It also lets you secure a few high-level devices instead of a network of dumb sensors. Using a hub will let you easily secure far-flung gadgets, add on new ones, and rest easy knowing your entire network won't turn into a botnet (like the shadow monster hive mind).

- Processing - You may want to use something like a multi-sensor platform with AI to inject some common sense into your network. The problem, though, is that machine learning takes a lot of processing power. Far more than can be mounted on a typical low power IoT device. Gateways can either have onboard processors or use the cloud to connect to deep neural networks to add intelligence to your system. By using a portal to do the heavy lifting, your devices can get brains without batteries.

- Connectivity - I've already mentioned that most edge devices will be able to communicate on all common protocols. Let's say, though, you also have old legacy devices on your system that you don't want to get rid of. It's unlikely they'll be able to communicate directly with newer sensors. However, if you add that capability to your gateway you can seamlessly integrate aging sensors.

I think we're all a little envious of Steve's hair. Editorial credit: DFree / Shutterstock.com

You may not be able to have Steve Harrington's hair, but you can imitate Hawkin's Lab with a gateway of your own. Instead of ending the world, your portal should enhance network connectivity, process data, connect to the Internet, and always stay on. The right edge device will enhance the security of your network, allow you to centralize processing, and could even let you integrate those old devices that are still running.

Now that you don't have to worry about someone hacking your network, you can create all those IoT devices you've been dreaming of. Even Steve needed a bat to fight the Demogorgon, so you might want a little assistance in your designs. CircuitStudio boasts a wide variety of advanced features that will make PCB design easy peasy.

Have more questions about IoT gateways? Call an expert at Altium.

SECURITY BY DESIGN: INTERNET OF THINGS AUTHENTIFICATION AND SELF TESTING

Have you noticed how many organizations are getting hacked lately? Businesses like Equifax aren't the only targets of these attacks; government institutions like the NSA have also been hit. Even the Internet of Things (IoT) has been targeted. Hackers infect and control devices in order to use them as a tool for distributed denial-of-service (DDoS) assaults.

I'm sure you've gotten your credit card hacked, as well; I have. The difficult part about a world whose interconnectivity is growing exponentially every year is the creativity with which hackers are using to find new entryways into your personal devices. It might seem impossible to properly defend yourself from a determined hacker, but that's not true. The more security that you add to your devices, the less likely you are to be hacked.

One of the many ways to protect your gadget from these kinds of intrusions is to require authentication for device communications. If we only look at protection, though, we're missing half the picture. The sheer number of breaches in recent years show that even if we take all the precautions, our boards still might get hacked. That's why it's important to self test both hardware and software in the field to mitigate the effects of an incursion.

SECURE YOUR DEVICES FROM GETTING ATTACKED

We all use authentication on a regular basis with our passwords for various sites and services, even if we forget them now and again. This kind of identity verification is going to be critical for the IoT, though not just between users and their devices. Machines also

interact with each other often without any user behind their interactions. Without proper authentication, this interaction becomes an open door. Designers like us need to work in machine to machine authentication protocols to protect against unauthorized access.

There are already billions of IoT devices around the globe, and the numbers keep growing. These gizmos are being arranged into large networks that can enable things like smart cities or connected highways. In these systems, thousands of sensors can be hooked together and may either communicate directly with each other or talk through a gateway. If your device doesn't use authentication, it could be accessed by a hacker's computer pretending to be a sensor on the network. Future IoT systems will be incredibly complex and will need authentication on multiple levels. If your product doesn't use authentication, it might be the weak link in the chain and allow unauthorized access to entire networks.

Both users and other devices need to be authenticated in the IoT.

SO YOU GOT ATTACKED? NOW PARRY!

Protecting our systems from attackers is important, but so is what we do after we find out we've been hacked. Maybe I should say "if" we find out we've been hacked. In August 2013, Yahoo's system was breached, and in 2016, they reported that over 1 billion accounts had been accessed. It turns out the intrusion actually affected all 3 billion of their users. If our boards can't self-test out in the field we may end up like Yahoo, wondering if we've been hacked or how badly. If my PCB is part of a botnet that's trying to take down the Internet, I want to know.

There are two main ways to self-test your machine: through its hardware and through its software. By enabling self testing to occur in both, you are ensuring that your machine is capable of catching any unnatural intrusions, errors, or disabled processes. This will be able to help in allowing you to catch your machine if it has, in fact, become infected.

FIRST YOU THROW A JAB

Sometimes things just fail on their own, and sometimes someone breaks them. It can be hard to tell which it is, but it's important that you know when your hardware isn't working correctly. Things like memory are prone to corruption, but luckily there are some fairly simple ways to test if your storage is operating properly. For the specific PCB you're designing, you should be able to find ways to see if everything is working correctly.

It's helpful if you provide the results of that self test to other devices in the network as well. If a hacker wants to disable a system, they could feed pieces of it false inputs until the controller assumes those parts are broken and shuts them down. Then the hacker can continue their attack. If a sensor or peripheral can self-test and show the master that it isn't broken, that can alert the system to a potential attack. After the alert goes off, you've already gone a significant way to disabling potential attacks.

THEN YOU HIT WITH AN UPPERCUT

Similar to hardware, your software may have self induced errors as well as malicious ones. Finding these errors can be more important than detecting physical malfunctions, though. If a hacker can load malicious code onto your device it may never register as being broken. Instead, the hacker's program can make everything appear fine while using the device for their own purposes.

For software you can write a part of your program, known as sentinel code or a watchdog, to watch other parts and ensure they're operating properly. That will certainly let you detect accidental errors, like data corruption which can be caused by pointers. The more nefarious hackers will often use code-injection attacks in an attempt to gain access to your device. These can be difficult to detect because they may look like normal data to your sentinel code until it's too late. It is possible to design for code-injection attacks, but you may need to use a static analysis tool to track how data moves through your program. Once you understand exactly how your device handles information, you can make your watchdog much more effective.

You need to know if your device or its peripherals aren't working.

When creating a PCB, most of us designers are more focused on getting it to operate efficiently than worrying about how to secure it against hackers. That problem, though, is becoming increasingly important as the list of large scale breaches continues to grow. Protecting your device on the front end through authentication can ensure that your gadget is only talking to authorized users or systems. These are not uncompromising protections though, and it is possible to still fall victim to an attack. If that's the case then we also need to focus on detecting assaults as they're occurring, or after they've happened. Hardware and software self testing can help us find out when our products have been compromised.

Between conventional design and new security concerns, designers like us have a lot on our plates. That's why it's important to use great PCB design software that lets you use your time economically. Altium Designer comes with a great range of tools and add-ons that can help you work as efficiently as possible.

Have more questions about IoT security? Call an expert at Altium.

ADDITIONAL RESOURCES

Thank you for reading our guide on IoT Security Best Practices. To read more Altium resources, visit the Altium resource center here or join the discussion at the bottom of each original blog post:

- Internet of Things Security Best Practices: Passwords, Patches and Portals
- Internet of Things Security Issues Prompt Government Intervention
- Internet of Things Security Vulnerabilities: All About Buffer Overflow
- PCB Technology Trends: The Benefits of Internet of Things Security Gateways
- Security by Design: Internet of Things Authentification and Self-Testing