
Symantec
An Internet of Things
Reference Architecture
When you can do it simply, safely, and quickly, you can do it all.

Executive Summary

The Internet of Things (IoT) already helps billions of people. Thousands of smart, connected devices deliver new experiences to people throughout the world, lowering costs, sometimes by billions of dollars. Examples include connected cars, robotic manufacturing, smarter medical equipment, smart grid, and countless industrial control systems. Unfortunately, this growth in connected devices brings increased security risks. Threats quickly evolve to target this rich and vulnerable landscape. Serious risks include physical harm to people, prolonged downtime, and damage to equipment such as pipelines, blast furnaces, and power generation facilities. As several such facilities and IoT systems have already been attacked and materially damaged, security must now be an essential consideration for anyone making or operating IoT devices or systems, particularly for the industrial Internet.

How can anyone secure the IoT? IoT systems are often highly complex, requiring end-to-end security solutions that span cloud and connectivity layers, and support resource-constrained IoT devices that often aren't powerful enough to support traditional security solutions. There is no single silver bullet. Locking doors but leaving a window open isn't enough. Security must be comprehensive or attackers simply exploit the weakest link. Of course, traditional Information Technology (IT) systems often drive and handle data from IoT systems, but IoT systems themselves have unique additional security needs. Fortunately, IoT security can be covered with four cornerstones:

- **Protecting Communications**
- **Protecting Devices**
- **Managing Devices**
- **Understanding Your System**

These cornerstones can be combined to form powerful and easy-to-deploy foundations of security architectures to mitigate the vast majority of security threats to the Internet of Things, including advanced and sophisticated threats. This paper describes these cornerstones, their necessity, and strategies for easy and effective implementation. No single, concise document can cover all of the important details unique to each vertical. Instead, this paper attempts to provide advice applicable to all verticals, including automotive, energy, manufacturing, healthcare, financial services, government, retail, logistics, aviation, consumer, and beyond, with examples spanning the majority of these verticals. The cornerstones themselves can be described briefly.

Protecting Communications

Protecting communication requires encryption and authentication for devices to know whether or not they can trust a remote system. Fortunately, newer technologies like elliptic curve cryptography work ten times better than predecessors in resource constrained chips like 8 bit, 8 MHz chips of IoT. This leaves the core challenge of managing all of the "keys" for authentication. A number of leading Certificate Authorities (CAs) have already embedded "device certificate" keys into more than a billion IoT devices, helping mutually authenticate a wide range of devices including cellular base stations, televisions, and more.

Protecting Devices

Protecting devices against attack requires both code signing, to be sure all code is authorized to run, and run-time protection, to be sure malicious attacks don't overwrite code after it is loaded. Code signing cryptographically ensures code hasn't been tampered after being "signed" as safe for the device, and it can be done at "application" and "firmware" levels, even in devices with only a monolithic firmware image. All critical devices, whether a sensor, a hub, or anything else, should be configured to only run signed code and never run unsigned code.

Still, devices must be protected long after code begins running. Host based protections help here. Host-based protection provide hardening, lockdown, whitelisting, sandboxing, network facing intrusion prevention, behavioral and reputation based security, including blocking, logging, and alerting for a variety of IoT operating systems. Recently, some host-based protections have been adapted for IoT, and now run well without requiring access to the cloud, and without undue strain on limited devices.

Managing Devices

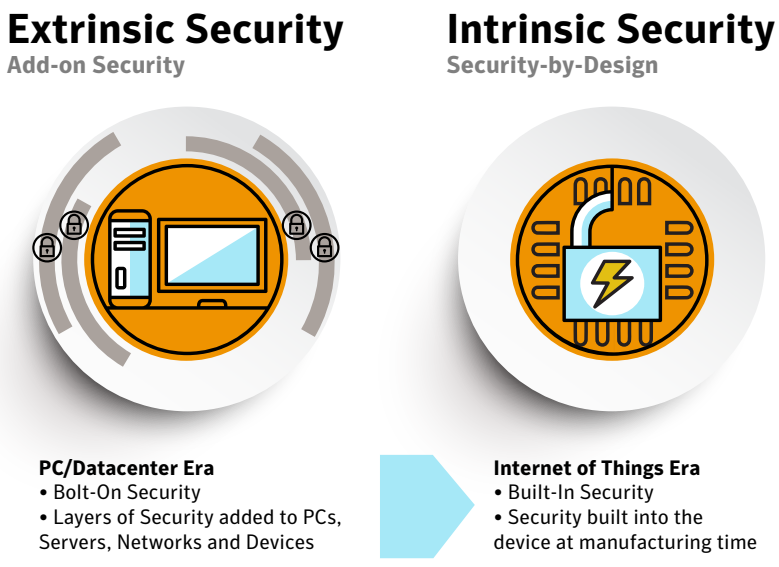
Unfortunately, vulnerabilities will eventually be discovered in valuable devices that will then need to be patched long after they shipped. Even obfuscated code for critical systems will be reverse engineered, vulnerabilities discovered, and updates required. Nobody wants to send employees to physically visit each device for updates, especially if that involves a fleet of trucks. For such reasons, over-the air (OTA) manageability must be built into devices before they ship.

Understanding Your System

Of course, no matter how well you lock everything down, and no matter how well you manage your systems, some threats can defeat all of those countermeasures to establish a toehold in your systems. For such reasons, it's crucial to have an IoT Security Analytics capability that helps you best understand your network by helping you flag anomalies that might be suspicious or dangerous, malicious or not.

Crucial Context: Critical Evolution

Most IoT devices are "closed." Customers can't add security software after devices ship from the factory. Often, such tampering voids the warranty. For such reasons, security has to be built into IoT devices so that they are "secure by design." In other words, for IoT, security must evolve from security just "bolted onto" existing systems such as servers and personal computer (PC) laptops and desktops. Security must evolve to security that is "built in" to the system before the system leaves the factory. For most of the security industry, such "intrinsic" security, built-in at the factory is a new way to deliver security, including classic security technologies like encryption, authentication, integrity verification, intrusion prevention, and secure update capabilities. Given the close coupling of hardware and software in the IoT model, it's sometimes easier for IoT security software to leverage advanced security hardware features often overlooked by traditional security vendors who must simply build "extrinsic" security layers to run on "least common denominator" hardware. Fortunately, many chipmakers already build security features into hardware. Unfortunately, the hardware layer is just the first layer required in comprehensive security, required for hardware-backed security in protecting the communications and protecting the device. Comprehensive security requires clean integration of the key management, host-based security, OTA infrastructure, and security analytics mentioned above. Failing to address any one of the cornerstones of security leaves your fate to the whims of aggressors.



In short, as the Industrial Internet and IoT bring networked intelligence to the physical things around us, we must approach its security carefully. Our lives depend on the planes, trains, and cars that move us every day. Our lives depend on healthcare infrastructure and the civil infrastructure that makes it possible for us to live and work so closely together in cities. It is not difficult to imagine how unauthorized manipulation of traffic lights, medical equipment, or countless other examples can cause fatalities. It is also becoming clear how citizens and consumers do not want strangers hacking into their homes or cars, and the kinds of damage that can be done through lost productivity in disruption of automated manufacturing. In this context, we've attempted to offer the guidance of this paper to define end-to-end security for IoT while making it both effective and easy to deploy.

Not cyber myths: Hacking oil rigs, water plants, industrial infrastructure

Security researchers explain that hacking oil rigs, pipelines, water pumps, industrial facilities, and the power grid are not myths born in the cyber-mist, but realities.

For More Information: http://www.symantec.com/iot

Protecting Communications

A Strong Trust Model for IoT

Security rests on fundamentals. Encryption, authentication, and "key management" are invariably the foundation of meaningfully resilient security. Fortunately, some great open source libraries perform encryption really well, even in resource constrained IoT devices. Unfortunately, most companies still take dangerous risks attempting to do the key management for IoT entirely on their own. In contrast, roughly $4 billion per day of e-commerce transactions are protected by a simple but strong trust model serving billions of users, and serving over a million companies worldwide. This "trust model" helps their systems safely authenticate systems of other companies and safely start encrypted communications with those systems. This "trust model" is the cornerstone of secure interoperability in computing today, and it is a "trust model" grounded on a very short list of extremely strong certificate authorities (CAs). These very same CAs already embed certificates in billions of devices every year. These device certificates enable the authentication of mobile phones in safely connecting to the nearest base stations, authentication of smart meters for the electrical power industry, and authentication of set top boxes in the cable television industry, among countless other examples. Strong CAs make it easy to safely and securely generate, issue, enroll, manage, and revoke the certificates, keys, and credentials that are crucial to strong authentication. Given the volumes of security certificates involved in IoT, most device certificates are sold in high volume for dimes each, not whole dollars each.

Why does authentication matter? It is dangerous to accept data from either unverified devices or unverified services. Such data can corrupt or compromise your devices, and give control of those devices to some malicious party who wishes to harm you or harm others through you. Using strong authentication to restrict such connections helps protect your devices from such threats, while helping you keep control of your devices and services. Regardless of whether a device is connecting to another device as a peer, or connecting to a remote service, such as a cloud based service, the communications must be protected. All such interactions need robust mutual authentication and trust. In that context, skimping on device certificates seems foolish.

Fortunately, many standards have been developed to make deploying strong mutual authentication relatively easy. Standards exist for certificate formats, and strong CAs support both standard and custom certificate formats. In most cases, certificates can easily be managed over the air (OTA) through standard protocols such as Simple Certificate Enrollment Protocol (SCEP), Enrollment over Secure Transport (EST), and Online Certificate Status Protocol (OCSP). With a strong CA helping to handle certificates, keys, and credentials, the actual authentication can easily be done by strong standards like Transport Layer Security (TLS) and Datagram TLS (DTLS)—akin to SSL. Mutual authentication, where both endpoints authenticate each other,

is crucial to the end-to-end security of IoT systems. As an added bonus, once TLS or DTLS authentication is completed, the two endpoints can exchange or derive encryption keys to start communication that cannot be decrypted by eavesdroppers. Many IoT applications will require absolute privacy of data, and this requirement is easily met through use of certificates and TLS/DTLS protocols. However, where privacy isn't a requirement, the data can be authenticated by any party if it's signed on sensor at "time of capture," and this approach cuts the burdens of link level encryption, which can be particularly important in multi-hop architectures.

It is very common to encounter concerns over the cost and power of IoT chips for cryptographic operations. However, it is important to recognize that Elliptic Curve Cryptography (ECC) has been proven 10x faster and more efficient than traditional encryption in resource constrained chips, such as IoT chips. ECC achieves this 10x improvement in speed and efficiency without reducing the level of security. ECC has even demonstrated "industry best practice" levels of security, equivalent to RSA 2048, and demonstrated such equivalent levels of security on extremely resource constrained chips such as 8-bit processors, and megahertz and even kilohertz speed 32-bit processors, some of which run on such low power as to be viable even in many micro-watt energy harvesting use cases. The DTLS variant of TLS was developed specifically for low-power devices that operate intermittently between sleep cycles. Last, the financial cost for such 32-bit chips can be in the 50-cent range, so it is not appropriate to use cost or power as reasons for skimping on security below reasonable thresholds where security matters. For these reasons, we have proposed the following guidelines on key lengths for IoT device authentication where security matters: (a) 224-bit ECC at a minimum for end-entity certificates, with 256-bit and 384-bit preferred; (b) 256-bit ECC at a minimum for root certificates, with 384-bit preferred.

Today, we cannot imagine the inconvenience of manually installing our browsers with certificates for each web server, nor can we imagine the damage from blindly trusting any certificate. That's why each browser has a few "roots" of trust against which all certificates are evaluated. Embedding these roots into browsers enabled security to scale to millions of servers on the web. As billions of devices come online annually, it is equally crucial that we embed both roots of trust and device certificates into these devices.

In managing data for the IoT, data needs to be kept safe and secure at all times. Our lives frequently depend on the correctness, integrity, and properly functioning availability of these systems more than on the confidentiality of the data. Authentication of information and devices, and provenance of information, can be critical. Unfortunately, data is often stored, cached, and processed by several nodes; not just sent from point A to point B. For these reasons, data should always be signed whenever and wherever the data is captured and stored. This helps mitigate risks of anything tampering with the information. Signing data objects once at capture, and relaying the signature with the data, even after the data is decrypted, is an increasingly common and useful engineering pattern.

The Internet of Things to be prime hacking targets in 2015

Automobiles and transportation systems will be likely targets for hackers in the coming 12 months

Protecting Devices

Protecting the Code that Drives IoT

In powering up, each device boots and runs some code. In that context, it is crucial that we ensure devices only do what we programmed them to do, and ensure that others cannot reprogram them to behave maliciously. In other words, the first step in protecting a device is to protect the code to be sure the device only boots and runs code that you want it running. Fortunately, many chipmakers already build "secure boot" capabilities into their chips. Similarly, for "higher level" code, a number of time-proven, open-source, and client-side libraries like OpenSSL can easily be used to check signatures of code, and accept code only if it comes from an authorized source. In that context, signing firmware, boot images, and higher-level embedded code are all increasingly common, including signing the underlying software components such as any operating system, and not just applications, but all code on the device. This approach can ensure that all critical components, sensors, actuators, controllers, and relays are all properly configured to only run signed code and never run unsigned code.

An apt rule might be, "never trust unsigned code." A corollary would be, "never trust unsigned data, and especially don't ever trust unsigned configuration data." With today's signature verification tools, and with hardware support for secure boot improving, the next challenge for many companies is "managing the keys," and "controlling access to the keys" for code signing and protection of embedded software. Fortunately, some CAs also offer hosted services that make it easy to safely and securely administer code-signing programs that ensure tight control over who can sign code, who can revoke such signatures, and how the keys for such signing and revocation are protected.

In an embedded context where software may need to be updated for security reasons, and where battery impact of updates must be handled carefully, it can be very important to sign and update individual blocks or chunks of such updates and not force anyone to sign entire monolithic images, or even an entire binary file. Instead, having such software signed at the block or chunk level can enable such updates to be done with much finer granularity without sacrificing security and without having to sacrifice the battery for security. Instead of always requiring hardware support, this flexibility can often be achieved from a small pre-boot environment, which might run on lots of embedded hardware.

Symantec.

If battery life is so crucial, why not simply configure a device with a permanently burned image that can't be replaced or updated by anyone? Unfortunately, we must assume that devices in the field will be reverse engineered for malicious purposes. When the devices are reverse engineered and vulnerabilities are discovered and exploited, the vulnerabilities will need to be patched as quickly as possible. Code obfuscation and code encryption can considerably slow down the reverse engineering process, and deter the majority of attackers, but not entirely prevent reverse engineering. Attackers with nation-state levels of resources, or the resources of sophisticated transnational malicious organizations, may still be able to reverse engineer programs including programs protected through obfuscation and encryption, particularly since code must be decrypted to run. Such organizations will find and exploit vulnerabilities that will need to be patched. For these reasons, over the air (OTA) update capabilities must be built into the devices before they leave the factory. Such OTA update capabilities, including software and firmware updates are crucial to maintaining a strong security posture for a long list of reasons that we'll elaborate in a section further below, "Managing Devices." However, obfuscation, granular code signing, and OTA updates may eventually need to be tightly joined for all to work both effectively and efficiently.

Fortunately, both granular and monolithic code signing leverage the same certificate-based trust model described in the section on "Protecting Communications," and the use of ECC in code signing can provide the same benefits of high-security with fast performance and low-power consumption. In that context, we propose the following guidelines on key lengths for IoT code signing where security matters: (a) 224-bit ECC at a minimum for end-entity certificates, with 256-bit and 384-bit preferred; (b) 521-bit ECC at a minimum for root certificates, because signed code is generally expected to be in use years or even decades after signing, and the signatures must be strong enough to remain secure for such a long time.

The Internet of Things to be prime hacking targets in 2015

ATMs will be likely targets for hackers in the coming 12 months

Protecting Devices

Effective Host-Based Protection for IoT

The cornerstones above describe fundamentals of key management and authentication for IoT, as well as code-signing and configuration signing to protect the integrity of the device, and the basics of managing such code and configuration, "OTA." Unfortunately, even after protecting communications and protecting the secure boot of a well managed device, the device still needs protection long after boot. Host-Based Protections address those needs.

IoT devices face many threats, including malicious data that can be sent over authenticated connections, exploiting vulnerabilities and/or misconfigurations. Such attacks frequently exploit many weaknesses, including but not limited to (a) failure to use code signature verification and secure boot, and (b) poorly implemented verification models which can be bypassed. Attackers often use those weaknesses to install backdoors, sniffers, data collection software, file transfer capabilities to extract sensitive information from the system, and sometimes even command & control (C&C) infrastructure to manipulate system behavior. Even more disturbingly, some malicious data attacks can exploit vulnerabilities to install malicious software directly into the running memory of "already running" IoT systems in ways that the malware disappears on re-boot, but does tremendous damage between reboots. This is particularly scary as some IoT systems, and many industrial systems, are almost never rebooted. Sometimes such attacks come through an IT network connected to an industrial or IoT network. Other times, the attack comes over the Internet, or through direct physical access to the device. Of course, regardless of the initial infection vector, if not detected, the first compromised device remains trusted and then becomes the avenue for infecting the rest of the network, regardless of whether the target is the "in-car" network of a vehicle, or a plant-wide operational network of a manufacturing plant. For such reasons, IoT security must be comprehensive. Closing a window but leaving a door open, "isn't adequate." All of the infection vectors must be mitigated.

Symantec.

Fortunately, when coupled with a strong code signature and verification model, host-based protection can help secure the device against all of these threats by using a number of technologies including system hardening, whitelisting, application sandboxing, reputation-based technology, anti-malware, and encryption. Depending on the needs of the specific system, a combination of these technologies can ensure the highest level of protection for every device.

System hardening, whitelisting, and application sandboxing can provide network protection, closing back doors, limiting network connectivity by application, and restricting both inbound and outbound traffic flow. This can also provide protection against different exploits, restricting app behavior, protecting the system from buffer overflows and zero day attacks, while preserving control of the device. Such solutions can also be used to prevent unauthorized use of removable media as well as locking down device configuration and settings, while also de-escalating user privileges where needed. Such solutions can also provide auditing and alerting functions, helping monitor logs and security events. Policy based technologies can even be run in environments without the connectivity or processing power required to run traditional signature-based technologies.

Reputation-based security technology can be used to put files in context, using their age, frequency, location, and more to expose threats otherwise missed, as well as provide insight on whether or not to trust a new device, even when successfully authenticated. Such techniques can also identify threats that use mutating code or adapt their encryption schemes, still separating files at risk from those that are safe, for faster and more accurate malware detection, even despite such challenges.

Of course, the mix of technologies will depend on use case, but the options above can be combined to protect devices, even in resource-constrained environments.

Symantec.

Managing Devices
Safely and Effectively Managing IoT

As mentioned above, devices will be reverse engineered, vulnerabilities discovered, and devices will need to be updated OTA. Of course, OTA update mechanisms add complexity, so many engineers attempt to avoid them at their peril. Fortunately, a good OTA mechanism can be used for many purposes, not just software / firmware security patches and functionality updates, but also:

- **Configuration updates**
- **Management of security content and security telemetry for security analytics**
- **Management of telemetry and control for proper system function**
- **Diagnostics and remediation**
- **Management of Network Access Control (NAC) credentials**
- **Management of permissions, and countless other examples**

Of course, all of the above must be done safely and securely, and requires more than securely signing code and performing file transfers. Fortunately, strong standards exist for managing software and firmware inventories on each device, as well as device configuration, and many vendors support such standards including the Open Mobile Alliance (OMA). Some of these solutions scale to managing billions of devices.

Managing security for each device can include managing configuration of host-based security technologies that we described in the preceding section. Of course, some security technologies need OTA updates of security content such as blacklists, whitelists, heuristics, intrusion prevention signatures, and reputation data. Fortunately, some security technologies depend on policy based mechanisms that only need updates when the software on a device is re-imaged for other purposes, such as adding functionality. However, both types of security technology can generate security telemetry that is valuable in facing Advanced Persistent Threats (APT). For such reasons, the security telemetry should always be aggregated from those host-based (device-based) technologies for more central analysis.

Of course, the security components are not the only components on each device that need to be managed safely and securely. Most devices generate sensor data or telemetry that needs to be safely and securely collected and transmitted to a safe and secure place for storage and analytics. Many devices also actuate control features that need to be managed carefully with configuration parameters that need to be safely and securely kept up to date. Fortunately, infrastructures that use safe and secure general device management protocols can be used for safely and securely managing the device's primary functionality as well as the device's security content and telemetry. In fact, such frameworks are already being adapted for OTA management of cars, and already used to safely and securely manage in-store interactive marketing kiosks, as well as vending machines. Some of these management frameworks use a mix of agent based and existing agentless IoT management protocols where the device is built to support standards-based management for simpler management functions. Additionally, some management frameworks can even couple those techniques with insights collected from network sniffers.

In this context, IoT systems must have update capabilities built into them from the beginning. Failing to build in OTA update capabilities will leave devices exposed to threats and vulnerabilities for the entirety of their lifetimes. Of course, such update capabilities can be used to manage device configurations, security content, credentials and much more. Similarly, such update capabilities can be used to push functionality and collect telemetry in addition to collecting software inventory information and pushing security patches. However, with or without such additional functionality, basic update capabilities and the ability to manage the security posture of each device must be built into the device from the beginning.

The Internet of Things to be prime hacking targets in 2015

Buildings and cities will be likely targets for hackers in the coming 12 months

Understanding Your System

Security Analytics to Address Threats Beyond the Above Countermeasures

Of course, no matter how well you protect the device, protect the code, protect the communications, and no matter how well you manage your security posture, even using the best possible OTA management framework, some adversaries still have the resources and capabilities to rise above those defenses. For such reasons, strategic threats require strategic mitigation technologies. Security analytics can leverage security telemetry from devices and network hardware to help provide an understanding of what is happening in the environment, including detection of stealthier threats.

Equally importantly, "monitoring" and analytics can often be deployed as an interim solution in environments where upgrading devices to conform to the first three cornerstones above will take years. Examples of such environments include legacy devices such as industrial control systems (manufacturing, oil and gas, utilities) that cannot be modified until an end-to-end replacement system is ready, automotive cars already on the road whose deeply embedded microcontrollers obviously cannot be "torn out and replaced," and healthcare environments where suppliers prohibit hospitals from modifying the equipment to add security. In such cases, anomaly detection solutions can be extremely valuable. The deterministic nature of many IoT networks allows the system to be baselined and deviations quickly identified. The wide variety of industrial and IoT protocols can make the problem harder, but newer techniques using advanced machine learning can allow the problem to be solved. Considering that many IoT systems have high demands on availability, this solution is less invasive in "detect" mode while ensuring that any false positives do not bring down the system.

Other examples include gateways, such as between legacy environments and better-protected environments, particularly as an attack in one part of the ecosystem or environment could be transmitted across the entire network if not caught early. Similarly, other high priority targets for distributed monitoring and centralized analytics include gateways between "industrial" networks and "IT" networks, residential gateways separating homes full of devices from the rest of the internet, the head unit of a car as a gateway between the "in-car" networks and the cellular network, and gateways between automotive drive system instrumentation and infotainment networks in the car.

Symantec.

For many of these examples, customers can work with security vendors to utilize existing big data security analytics infrastructure and large threat intelligence gathering systems to collect, analyze, and share information across entire networks and ecosystems. Some of these efforts are already ongoing in different verticals such as retail and critical infrastructure, and these efforts can ensure that a system as a whole can be quickly updated to protect itself against any emerging threats.

In many cases, the caliber of "data science" and security expertise required in the analytics for detecting extremely advanced threats can be beyond the capabilities of companies who do not specialize in those areas. For such reasons, many companies are turning to "managed" solutions akin to managed security solutions so that they can count on experts doing the monitoring and analytics. In other cases, companies are building their own repositories of IoT security telemetry, and controlling access to that repository so that they enable multiple analytics partners to help them find such advanced threats. Some analytics products and platforms are even exposing API and SDK to both enable such sharing, and to ensure safe control of such sharing, such as ensuring only relevant data is shared with the right partners, and ensuring that each partner's access is appropriately restricted.

For examples bridging industrial and IT networks, we recommend creating a single data plane spanning both environments to get a prioritized view of different threats, and to best mitigate risks of threats tunneling from one environment to the other. Such solutions should work across different vendors and different devices and protocols to ensure that each customer gets a holistic view without blind spots in their network.

Through such analytics, "detection and response" can complement strong protection technologies to provide security against the vast majority of attacks, as well as mitigating risks of the most serious and capable adversaries.

Symantec.

Understanding Your System

What To Trust

Today, countless IoT technologies and systems are really no more than “intranets of things.” However, as more and more of these systems will need to connect with each other, it becomes increasingly critical to “know what to trust.” Device certificates can establish pedigree and lineage of a device. However, questions on whether or not that device should still be trusted, will eventually need to be answered by other services, such as reputation based services, or a “Directory of Things.” Such a directory could track not only security information regarding each device and IoT system, but could also help track and manage the permissions and entitlements that devices and systems grant each other. In fact, as we each find ourselves surrounded by more and more IoT devices, such directories could also help with “discovery” of devices in areas of interest, and with features of interest. In such a model, it might even be possible to quickly find a remote device through such a directory, and quickly agree to purchase data or services from that device. Even if you’ve never seen the device before, the devices details including it’s capabilities and reputation could all be listed in such a directory. In fact, when you consider that the device will want to know whether or not it can trust a user, perhaps a “Directory of Things” isn’t enough. Perhaps we need a “Directory of Everything,” including devices, systems, users, and perhaps even a kitchen sink, if the sink is “internet connected” like a recent Stanford project monitoring water usage given California’s drought in 2015.

Of course, not many people have smart sinks or even smart refrigerators “yet.” However, many of us have a car that fetches traffic information over the internet, a Smart TV or Blu-ray players that stream video over the Internet, a fitness wearable, and we use ATM machines and digital Point-of-Sale machines more often than we can count. In that context, we might each want our own “Directory of Things” to manage them all sooner rather than later. Still, where protecting the communications, protecting the devices, managing the devices, and security analytics for addressing strategic threats, are all absolutely required for IoT Security, we have to admit that the “nice” concepts of “Directories” for “knowing what to trust” are still more formative and visionary, and neither a cornerstone, nor a required ingredient in “understanding your system” today, at least not for most parties. We include these “nice” concepts of “Directories” for “knowing what to trust” only to give a preview of challenges ahead for many in trying to manage such complexity at such scale. We include them as some companies are already facing such challenges, as they are already responsible for protecting more than a billion devices. For them, that “future” is already here, and they are not alone.

Symantec.

Why IoT Security Must Be Comprehensive

An example

Taking just one example of why none of the cornerstones above can be neglected, we could consider trains.

In the example of trains, electric motor controllers not only control acceleration of such trains, they often also control regenerative braking of trains. Even if mechanical brakes are included as a safeguard against uncontrolled acceleration, no such mechanical safeguards prevent a maliciously programmed motor controller from sudden and disproportionate braking that could cause harm to the train and its occupants. For these reasons, it is essential that all code executing in such controllers, brakes, switches, and more—all code driving any kinetic aspect of the train—be properly signed and all such components properly configured to never run anything but signed code.

Similarly, if communications aren't authenticated, both within the train, and from the train to other infrastructure, the consequences can be severe. It's not hard to imagine the consequences if control signals within the train for acceleration and braking could be spoofed, nor is it hard to imagine the consequences for spoofing an all-clear when danger lays ahead.

Further, without host-based protections, the controllers themselves could be hacked, and malicious parties could achieve any of the same evil objectives without needing to defeat the authentication or code signing mechanisms.

Moreover, the necessity of such comprehensive security is not limited to trains. As cars become increasingly connected, they require similar host-based protections. Such protections can be deployed on the head unit of a car even if the car is running a real-time operating system. Of course, as the code is updated OTA, these policies can be updated OTA using the same OTA system. Without the ability to "adapt" security posture OTA, adversaries will quickly adapt to find your weaknesses and exploit them.

Symantec.

However, even if all of the above is done correctly, the most sophisticated adversaries can still defeat such countermeasures. For such reasons, backend security analytics are required to mitigate these strategic threats. Such systems can continuously collect data, forming baselines for trains, planes, cars, manufacturing plants, point of sale systems, nearly anything. With such baselines, IoT security analytics can quickly detect anomalies, helping detect stealthier threats, and feeding advanced threat correlation as part of broader security analytics in helping fight these strategic threats.

Last, it's important to note that IoT security does not exist in a vacuum. Many of these devices need "physical security," and the type of physical security will depend heavily on the use-case. An IoT device in the home might simply need an enclosure that prevents a maid from tampering with the device to spy on employers. However, IoT devices in a manufacturing plant often need layers of physical security that include key-card access to each room, and similar restrictions out to a fence distance determined by electro-magnetic risk decisions. Personnel security needs will similarly vary dramatically. However, physical security and personnel security are not unique to IoT. Most companies already address these well today, and must do so simply to protect normal factory production, and protect their traditional IT systems. For those reasons, this document has focused exclusively on the requirements for getting IoT Security "right" in and between IoT devices and their communications. Of course many of these devices frequently interact with traditional backend IT systems often running in a datacenter or in a cloud. We assume that you will get security "right" for those systems. However, please bear in mind that where those "traditional" IT systems either drive IoT devices and systems, or handle data from the IoT devices systems, failure to get security right for those "traditional" IT systems can completely undermine all of the security that you have built into your IoT system.

As IoT becomes increasingly commonplace, and particularly as life critical systems like cars, planes, and industrial equipment increasingly leverage IoT, security must be correctly built into these systems, so that they are "secure by design" with security "built in" from the beginning. The stakes are simply too high for mistakes in most cases. Toward that end, to help others build security into their systems, and toward helping achieve industry consensus on a minimalistic set of cornerstones that could provide adequate security against today's threats, we hope this paper helps.

Symantec.

Summary

This paper advocates a simple and effective reference architecture for IoT security that should be easy to deploy and scale.

- The architecture mitigates malicious code by ensuring that all code is cryptographically signed and authorized for the device, and ensuring that unsigned code is not permitted to run.
- It protects communication through mutual authentication and encryption, leveraging time-proven certificate authorities and time proven trust models already protecting more than a billion IoT devices, but leveraging newer ECC algorithms to provide that level of security in resource constrained IoT devices.
- The architecture further mitigates malicious data through host-based protection and further mitigates all remaining threats through security analytics.
- As vulnerabilities and threats are discovered, they can be mitigated through effective, safe, and secure dynamic management of the system.

This reference architecture is grounded on time-proven fundamental tenets of security. At the same time, in stripping the architecture to a minimalistic "required" level of security, we have excluded substantial security features and security functionality that would be very "nice" to have, even if not required to the same degree as the elements described above. We have stripped this security reference architecture into its barest form for several reasons. As engineering professionals, we need to establish an appropriate and easily reached minimum level of security for any IoT system where security matters, and it is valuable to everyone if the same architecture can be applied across many different verticals, particularly with protective security talent in such rare supply. Still, some companies may choose to go "above and beyond" the level of security that we describe here. We applaud that as often good, even when not obligatory. In many cases, we hope that verticals, top suppliers, and service providers in those verticals go far beyond the minimum established above. More importantly though, "skimping" in any one of the four cornerstones invites harm in all of the forms that could be done through misuse of your system.

Symantec.