SCTE·ISBE CABLE-TEC EXPO 2017
DENVER, CO
OCTOBER 17-20

2017 Fall Technical Forum
SCTE·ISBE • NCTA • CABLELABS

Device Risks to Network Operators from IoT

Exploring the Critical Aspects of Onboarding, Authentication, Authorization and Accountability

A Technical Paper prepared for SCTE/ISBE by

Brian A. Scriber
Principal Architect, Security
CableLabs
858 Coal Creek Circle
Louisville, CO 80027
b.scriber@cablelabs.com

Table of Contents

Introduction

The promise of the Internet of Things (IoT) is to develop systems, sensors, and rules to help automate our environments in a way that brings connectivity between people, security and protection to individuals and families, and allows us to bridge gaps in geography. Security in this realm has been widely discussed, often bemoaned and still in need of improvement.

Grouping like concerns around IoT Security results in seven groups of closely related concepts: Identity, Onboarding, Confidentiality, Integrity, Availability, Lifecycle Management, and Future Security. Each of these can be further broken down into subcategories, and each comes with its own particular threats as well as techniques to help secure it.

Content

1. Identity

Device identity is the foundational aspect of the security model for the IoT environment. Without a solid device identity model, spoofing attacks can allow malicious devices to masquerade as trusted entities. When mitigation efforts attempt to find the source of traffic or commands, a device capable of spoofing its identity can simply change to another one, complicating and delaying efforts at neutralizing threats which endanger devices, people in the world those devices operate, or even the network itself.

Device identifiers are often an attribute akin to a serial number which the manufacturer grants to the device, sometimes it is a Media Access Control (MAC) address, and in some cases it is a collection of attribute values from the device which offer some semblance of uniqueness. The problems begin if these addresses are not *unique*. If two devices can share an identifier, then the identifier is useless from a security context. When manufacturers opt to duplicate MAC addresses in products intended for different networks (or even different parts of the world), we lose the promise of uniqueness, and we lose the ability to hold devices accountable for their actions within a network.

When an identifier can be changed, when it can be switched between different values, or when it is based on aggregating attributes which can be modified, we also lose the ability to drive toward accountability within a network. If the device identity is not stored in a protected read-only location, if buffer-overflow attacks can reach this memory, or if the software delivering this address to requesting parties can be corrupted, we have no way to trust the device is what it claims to be or to trust what it will do within or beyond the network. When we see the ease of which MAC addresses can be spoofed, and the difficulty in tracing back to compromised devices participating in botnets, it becomes clear that the *immutability* of the device identifier is critical to security.

Even if a device identifier cannot be modified, and it appears unique on the network, if that device can switch between authentication or authorization credentials, it can still do a great deal of harm and may appear to be a normal device, operating within expected usage definitions, but still harbor malicious code which could be capable of acting on other devices across the network. It is for this reason that we must be able to *algebraically attest* that the device identity is uniquely associated with the credential used for both authentication and authorization.

All devices on a network, whether they can be classified as "IoT" or not, must have identifiers which are simultaneously *unique*, *immutable*, and *attestable*. With all three attributes satisfied, only then can additional trust be built upon the relationship these devices have within the network.

There are privacy concerns that arise from devices that broadcast their IDs prior to network onboarding, and one way to support identity as well as Confidentiality is to use a pre-onboarding technique to generate temporary IDs. Once a device has been onboarded and provisioned within network, it should be using only its unique, immutable and attestable identity.

While there are different ways to satisfy the three requirements of identity, certificates backed by a Public Key Infrastructure (PKI) provide a solution to all three.

2. Onboarding (AAA)

During onboarding and provisioning, a device transitions from a position of no trust within the network to an entity with at least some level of trust. Three aspects of this process include Authentication (confirm the device is what it says it is), Authorization (establish what the device is allowed to do in the network), and Auditing (sometimes referred to as Accountability). During this process, both the device and the network are vulnerable to different types of malfeasance, revoked or expired credentials, network credential exfiltration, and others, but there are specific concerns around Man in the Middle (MitM) attacks where either the device or the network is being misrepresented to the other by an intermediate party. Reducing MitM attack opportunities can be accomplished through using aspects of the Identity section, above.

Identity is obviously a key part of Authentication, proving you are who you say you are and not a MitM imposter. Using a certificate and public key is one way in which the device can prove to the network that it is in possession of a private key. That private key and certificate combination can also be used to validate the chain of Certificate Authorities (CAs) that have permission to grant certificates on behalf of the root authority for the ecosystem. Each of these can be validated algebraically and, using specific asymmetric cryptographic protocols such as a Diffie-Hellman Exchange, cannot be spoofed by a MitM attacker. Additionally, services offered by the CAs include validation not only of the provenance of the certificate being validated, but also confirms the validity period for that certificate and can verify that it has not been revoked since it was issued.

The recommendations for Authentication are to

1) Use strong authentication backed by an attestable authority
2) Guarantee unique credentials (do not share or rely upon "default" credentials)
3) Verify the credentials against a Certificate Revocation List (CRL), against an Online Certificate Status Protocol (OCSP) responder, or against a blockchain registration authority
4) Use similar tools/techniques to confirm the issuance of the credentials and match those to the expected device type being onboarded.
5) Confirm (again through the CRL or OCSP) that the credentials being used are currently valid.

Ecosystems and like-devices that wish to talk to each other have developed communication protocols and norms for discovery, onboarding, interoperation, and with that they have authorization schemes. These strategies are what grants permissions to devices to access data on other devices, to monitor other devices, and to send data or commands to other devices. One common way to manage authorization is to create Access Control Lists (ACLs) on devices allowing specific relationships with other devices. Some ecosystems enable not only inbound ACLs, but also outbound ACLs, others use roles and assign roles in a way that enables or disables access control. Ecosystems should

1) Ensure their access control is restrictive rather than permissive (default to no access, rely on explicit grants to enable access)
2) Protect credentials and valuable resources on the device
3) Restrict proxy behavior (device A using device B to talk to device C)
4) Severely limit unauthenticated discovery or introspection of devices

Auditing and Accountability are areas which are severely lacking in many IoT devices today, the argument used against it usually revolves around the devices being constrained in terms of storage/power/computation/cost, not having a way to easily interact with a user, or component libraries not supporting auditing. Each of these arguments balance cost and capabilities against the benefits of an audit record, and when systems are compromised, the best way to learn how to protect other systems in the future is to understand the chain of events that occurred during the compromise. While that can help with retrospective analysis, audit records and accounting can also help to recognize deviations from expected behavior to trigger real-time responses a la Intrusion Detection or Prevention Systems (IDS/IPS).

Audit records should be in a standardized format to allow for automated reading and responding to events or actions recorded within the logs. Each action should have an auditable link between itself and the Authentication of who triggered the action and the Authorization of which ACL or role allowed the action to take place. Every log should be immutable; even if it gets overwritten with a new log after a period of time, the log, while it exists, must allow only the event logging mechanism to modify it. In a perfect world, the log would be distributed (to ensure multiple copies in case one is corrupted/compromised), private (encrypted to capture both the privacy as well as the log message integrity), and they should trigger alerts which a human could interact with to identify potential network penetration concerns.

3. Confidentiality

This aspect of security is the part most people think of when IoT Security is discussed. The topic that ends up coming up in those conversations is usually encryption and this tool is hailed as the way to solve whatever security problem is being encountered. Encryption is certainly an important tool for security, and shouldn’t be overlooked, but it relies critically on the Identity and Authentication aspects discussed above. Identification of sensitive information is the first step in protecting it, this information could be Protected Health Information (PHI) or Personally Identifiable Information (PII), it could be credentials used by the device, or operating data. Each individual grouping of data may not be PII on its own, but it may become protected in certain regulatory environments if it is stored with other groups of data which in aggregate become PII or PHI. Protecting data can be done in three states of the data: at rest, in use, and in transit. When the data is at rest, it can be encrypted and kept in hardened areas of the device. When important data like private keys or credentials are being used, they should be operated upon from within a protected storage module like a Trusted Platform Module (TPM) where the storage should allow certain operations (e.g. signing a message) to take place within the module, ensuring that the keys never need to leave their storage where they would be subject to exfiltration attacks. Data in transit should be protected using application-level (also known as “end-to-end”) encryption for traffic.

4. Integrity

Once Identity has assured, Authentication established, Authorization approved, Accountability ensured, and Confidentiality protected, attention is turned toward making sure the device itself is protected and that it performs within the boundaries set forth for it. For critical devices such as those which act as hubs within the home, or those which bridge between an ecosystem and the larger network (or “cloud”), certain elements (listed further below) help to mitigate threats. The threats at this level come from attackers with physical access to the device which “top” the chips, which solder connections and use tools to extract data from device memories. The attacks at this level also come from those trying to upload malicious code into the devices, they could also be scans that look for open ports with known software (libraries or custom code) that have vulnerabilities, scans that look for web servers with default passwords to enable full access to the control states of the device, and more. The recommendation to focus on the critical devices should not be interpreted to mean that all devices are not worthy of protection, but rather assistance in the prioritization of efforts.

The recommendations to protect against these types of advanced attacks are to:

1. Use AAA to confirm that device identifiers, the execution environment, configuration data and communications are all authorized and appropriate.
2. Harden the device by developing a Secure Execution/Executable Environment (SEE), use a TPM, follow the Joint Interpretation Library (JIL) or Federal Information Processing Standard (FIPS) guidance.
3. Minimize the attack surface by closing unnecessary ports and disabling unnecessary services, particularly those services typically used for engineering access, but which are occasionally left installed (and in some cases, enabled) on devices.
4. Use a secure bootloader to ensure that attacks on the state transition, on the software update, or on the configuration data are stymied.
5. Validate configuration data; if this isn’t encrypted, it should at least be signed by an authority that the device trusts (using certificates allows for the root to be persistent in the device trust store).
6. Use non-repudiation for critical communications. Non-repudiation is the act of requesting and receiving a receipt acknowledging a communication sent to another device and having that device sign the receipt with its private key or network credential. The act of returning a signed receipt goes to Auditability and proves that the receiving device did receive the message and cannot claim ignorance of it, or act in such a way as to deny that the message was delivered.

5. Availability

The Availability aspect of IoT Security can be broken into two parts, the availability of the actual device, and then the availability of shared resources (such as the internet) when groups of infected IoT devices participate in botnet attacks. From the perspective of the former, when you go to your garage door and it doesn’t open, it’s unavailable. When you can’t access the status of that garage door from your phone, it’s unavailable. If the power is out in your house or garage, it’s also unavailable. Obviously, there are some aspects of unavailability that aren't as related to security (e.g. the power being out), but how the device reacts and recovers from them is critical to IoT Security. From the device availability perspective, manufacturers need to expect jamming attacks (frequency flooding, TCP/IP traffic attacks, etc.), there should be a process for how to deal with a loss of power or a loss of network connectivity. From either of these outages, auditing and recovery, notifications to device owners, and potentially re-onboarding, are all recommended. These are processes manufacturers should not only know, not only share, but physically test these as well to confirm that a jamming attack doesn't allow for the front door to be opened without any alerts, alarms, or notifications ever being sent to the owner of the home. Ecosystems need to limit the anonymous requests that can be made, particularly those which are multicast and have the ability to trigger each device to do work (such as introspection or discovery). Finally, ever outage must be audited, recorded, and communicated. In a perfect world, any change that occurred from one of these recoveries would be triggered as suspect and notifications sent as appropriate.

The set of devices now known as IoT wasn't always as connected as they are now, they weren't as capable, didn't include as many vulnerable libraries as they do now, and often have default usernames/passwords used as credentials which enable root-level access to the devices. Because of this legacy, Distributed Denial of Service (DDoS) attacks are possible through the use of armies of corrupted IoT devices known as botnets.

As of June 2017, botnets on the dark web could be rented for DDoS attacks at a rate of $5 for a 1-hour sustained 100Gbps attack and $10 for a 200Gbps attack[i]. With the Mirai botnet capabilities, and the release of the code for exploiting device vulnerabilities to enlist them into a Mirai-type botnet, attacks close to the terabit per second range are no longer just theoretically possible, but we have seen 600-800 Gbps attacks in 2016.

To help mitigate against botnets (as opposed to mitigation of DDoS, not covered in this paper), there are a few recommendations. First, use restrictive (rather than permissive) access control and default network traffic. Second, monitor for inappropriate/unusual traffic patterns (e.g. if the lightbulb suddenly wants to send tremendous amounts of data to a site online with questionable bona fides. A third option is to segment internal traffic and devices into subnets and manage those, keeping an eye on boundary traffic and confirming that it flows within expected norms.

6. Lifecycle Management

Manufacturers of short-lived devices (cell phones can have a 1-2 year support period) that are moving into IoT devices with differing support periods are encountering challenges to their business models. When a consumer purchases a lightbulb there may be different durability expectations than when a consumer makes a purchase of a refrigerator. When both of these devices are "smart" devices, when they connect to the home network as well as the internet, they require support because security isn't a fire-and-forget industry. Security engineering is different than some other types of engineering; security engineers aren't concerned exclusively with weather, stresses, friction and time, they face intelligent, active, adversaries who are constantly searching for weaknesses. With such adversaries, security engineers need to adjust practices (encryption, algorithms, hardening, identity, protocols) to ensure continued confidentiality and integrity of the devices they work to protect.

A "smart" light bulb is different from a standard (classic?) light bulb in that it has computational power, memory, a radio, and, importantly, it has network credentials. It also has an operating system, drivers, firmware, and a networking stack which all rely on libraries that have weaknesses. Those libraries may have been updated against new threats and improved against unknown threats, but the likelihood that the firmware in the light bulb has been updated is small. For many devices, updates aren't even available from the manufacturer (who may not be incented to spend time or money on supporting a product after sale). If they are available, there may be only complicated methods to install the updates – some involving the use of a soldering iron. With knowledge that nearly half of the homes in the United States do not have working batteries replaced in their smoke detectors[ii], the idea that these homes are going to update the firmware in their light bulbs seems to be a stretch.

From the lifecycle management perspective, consumers should expect the following technical elements from their devices:

1. Secure, standardized, automated updates
2. Devices that have clearly defined end of life functionality (such as the light working from within the local network, but perhaps after the support period ends the light can no longer be activated over the internet).
3. Devices should allow for credential renewal and revocation

From a procedural perspective of lifecycle management, we have inconsistent support for disclosure of:

1. Vulnerabilities and remedies
2. The support period to consumers prior to purchase
3. Functionality of the device available after the support period ends

7. Upgradeable Security

Regardless of the security algorithms or encryption schemes shipped with the IoT device, adversaries may find weaknesses in embedded libraries or in the operating systems before the product even hits the shelves. A method for secure software update is necessary, but the device itself must be designed to support the update procedure and be prepared to address the changes. If longer key lengths are required in the future, there must be enough secure storage to retain those keys and if algorithms need to change, devices must have the capabilities to have cryptographic libraries updated. Adversaries may also change over time, as could their motivations, the security principles (or principals, for that matter) that protected devices when they were designed may not be the same as when the device is used in practice.

When considering security for devices, it's easy to fall into the trap of thinking that some devices are in need of less security than others, but the manner in which attacks take place is often to target the weakest part of the perimeter of a network, and continue using credentials of a weakly protected device to extend the attack. No device with computational power, memory, a radio, and network credentials should be weakly protected, and the credentials should be protected behind a hardened hardware element such as a TPM if possible.

Category	Description	Goal for Every IoT Device
Device Identity	Require an attestable, immutable, and unique identifier for each device.	Use a secure certificate-based approach with centralized management (PKI) to provide an attestable, immutable, and unique identifiers for each device; certificate issuance; lifecycle management; and revocation.
AAA/Onboarding	Require the use of strong Authentication, Authorization, and Accountability (AAA) methods for provisioning and management of devices.	Use the established device identity to enforce strong authentication (identifying the user or device), no shared default credentials; use authorization enforced through established mechanisms to provide access network and device resources, using techniques such as ACLs, RBAC, etc.; incorporate accountability mechanisms that associate device actions with authentication and authorization.
Confidentiality	Protect data so that they are not made available or disclosed to unauthorized individuals, entities, devices, or processes.	Identify sensitive information (PHI, PII, credentials, etc.) and protect that data appropriately. Use of encryption for locally stored sensitive information. Provide protection of sensitive information (e.g., private keys) while in use. Use mutual end-point authentication and application-level encryption (end-to-end) for sensitive data in transit. Provide an ephemeral identifier for anonymous discovery requests and limit information available to anonymous introspection and reflection requests.
Integrity	Assure the device is trustworthy and the processes, data, and communications associated with the device are accurate.	Confirm that the device identities, execution environment, configuration, and communications are authorized and appropriate using the AAA methods. Harden the device to minimize the attack surface by closing unnecessary ports, disabling unnecessary services, and using a secure bootloader with configuration validation. Consider use of non-repudiation methods for critical communications.
Availability	Safeguard devices and associated communications for proper functioning.	Use restrictive, rather than permissive, default network traffic policies to limit communications to expected norms, guarding against both unintended as well as malicious denial of service attacks. Plan for appropriate device behavior in the event of network or radio communication failures, overloads, or outages (e.g., jamming).
Lifecycle Management	Support sufficient secure operation, update, and communications throughout the life of the device.	Provide for secure, automated, update mechanisms during the disclosed support period and publicly disclose vulnerability remedies, EOL functionality changes, and credential renewal and revocation.
Upgradeable Security	Plan for security improvements required to support equivalent device and network security in concert with Lifecycle Management.	Include support for longer key lengths, stronger cryptographic algorithms/cipher suites, and hardware based security over the supported life of the device.

IoT Security Categories

Conclusion

The seven categories of IoT Security are intended to help guide thoughts around protection of devices and the networks upon which they operate. The following table provides an overview of these principles and the key elements of each. These are intended to be a good place to start and not a comprehensive list of every security vulnerability that could ever arise. They are also intended as a second step on the security front, after engineering portals are eliminated from devices, after default passwords have been removed, and after open ports allowing tools like telnet to connect to devices have been closed.

Abbreviations

AAA	Authentication, Authorization and Accountability
ACL	Access Control Lists
CA	Certificate Authority
CRL	Certificate Revocation List
DDoS	Distributed Denial of Service (attack)
FIPS	Federal Information Processing Standard
IDS	Intrusion Detection Systems
IoT	Internet of Things
IPS	Intrusion Protection Systems
ISBE	International Society of Broadband Experts
JIL	Joint Interpretation Library
MAC	Media Access Control
MitM	Man in the Middle (attacks)
OCSP	Online Certificate Status Protocol
PHI	Protected Health Information
PII	Personally Identifiable Information
PKI	Public Key Infrastructure
SCTE	Society of Cable Telecommunications Engineers
SEE	Secure Execution Environment
TPM	Trust Platform Moduule

Bibliography & References

[i] CableLabs security research on the Tor-addressable network known colloquially as the “dark net”, 2016 and 2017.

[ii] http://www.nfpa.org/news-and-research/fire-statistics-and-reports/fire-statistics/fire-safety-equipment/smoke-alarms-in-us-home-fires