Draft NISTIR 8267

Security Review of Consumer Home Internet of Things (IoT) Products

Michael Fagan
Mary Yang
Allen Tan
Lora Randolph
Karen Scarfone

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8267-draft

National Institute of Standards and Technology
U.S. Department of Commerce

Draft NISTIR 8267

Security Review of Consumer Home Internet of Things (IoT) Products

Michael Fagan
Applied Cybersecurity Division
Information Technology Laboratory

Mary Yang
Allen Tan
Lora Randolph
The MITRE Corporation
McLean, VA

Karen Scarfone
Scarfone Cybersecurity
Clifton, VA

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8267-draft

October 2019

U.S. Department of Commerce
Wilbur Ross, Secretary

National Institute of Standards and Technology
Walter Copan, NIST Director and Undersecretary of Commerce for Standards and Technology

National Institute of Standards and Technology Interagency or Internal Report (NISTIR) 8267
41 pages (October 2019)

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8267-draft

Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by NIST, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

There may be references in this publication to other publications currently under development by NIST in accordance with its assigned statutory responsibilities. The information in this publication, including concepts and methodologies, may be used by federal agencies even before the completion of such companion publications. Thus, until each publication is completed, current requirements, guidelines, and procedures, where they exist, remain operative. For planning and transition purposes, federal agencies may wish to closely follow the development of these new publications by NIST.

Organizations are encouraged to review all draft publications during public comment periods and provide feedback to NIST. Many NIST cybersecurity publications, other than the ones noted above, are available at https://csrc.nist.gov/publications.

Public comment period: *October 1, 2019* through *November 1, 2019*

National Institute of Standards and Technology
Attn: National Cybersecurity Center of Excellence (NCCoE)
100 Bureau Drive (Mail Stop 2002) Gaithersburg, Maryland 20899-2000
Email: home-iot-nccoe@nist.gov

All comments are subject to release under the Freedom of Information Act (FOIA).

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analyses to advance the development and productive use of information technology. ITL's responsibilities include the development of management, administrative, technical, and physical standards and guidelines for the cost-effective security and privacy of other than national security-related information in federal information systems.

Abstract

This report presents the results of a project that conducted a technical review of security features in different categories of consumer home Internet of Things (IoT) devices. The categories of IoT devices included smart light bulbs, security lights, security cameras, doorbells, plugs, thermostats, and televisions. The purpose of the project was to better understand security capabilities of these IoT devices and to inform general considerations for manufacturers for improving the security of consumer home IoT devices. This report provides those considerations, along with observations of IoT devices' security features, to indicate current practices and how these current practices could be improved.

Keywords

consumer home Internet of Things; cybersecurity; Internet of Things; IoT devices; smart home.

Acknowledgments

The authors wish to thank the many people who assisted with the development of this document, including our NIST colleagues Katerina Megas, Eric Simmon, and Barbara Cuthill. We would also like to thank the technical review team from The MITRE Corporation for their support on this effort: Emmanuel Bello-Ogunu, Brian Johnson, Kayla Kraines, Anthony Louie, David Mann, Joe Mansour, Elliot Mudrick, and Jay Vora; and our summer interns, Jalalah Abdullah, Sean Wagner, and Tim Weir.

Audience

The main audience for this report is the manufacturers of consumer IoT devices used in smart-home environments. Owners and users of consumer home IoT devices may also find portions of this report useful for better understanding some of the security implications of adding consumer home IoT devices.

Trademark Information

Call for Patent Claims

This public review includes a call for information on essential patent claims (claims whose use would be required for compliance with the guidance or requirements in this Information Technology Laboratory (ITL) draft publication). Such guidance and/or requirements may be directly stated in this ITL Publication or by reference to another publication. This call also includes disclosure, where known, of the existence of pending U.S. or foreign patent applications relating to this ITL draft publication and of any relevant unexpired U.S. or foreign patents.

ITL may require from the patent holder, or a party authorized to make assurances on its behalf, in written or electronic form, either:

a) assurance in the form of a general disclaimer to the effect that such party does not hold and does not currently intend holding any essential patent claim(s); or

b) assurance that a license to such essential patent claim(s) will be made available to applicants desiring to utilize the license for the purpose of complying with the guidance or requirements in this ITL draft publication either:

    i. under reasonable terms and conditions that are demonstrably free of any unfair discrimination; or
    ii. without compensation and under reasonable terms and conditions that are demonstrably free of any unfair discrimination.

Such assurance shall indicate that the patent holder (or third party authorized to make assurances on its behalf) will include in any documents transferring ownership of patents subject to the assurance, provisions sufficient to ensure that the commitments in the assurance are binding on the transferee, and that the transferee will similarly include appropriate provisions in the event of future transfers with the goal of binding each successor-in-interest.

The assurance shall also indicate that it is intended to be binding on successors-in-interest regardless of whether such provisions are included in the relevant transfer documents.

Such statements should be addressed to: home-iot-nccoe@nist.gov.

Executive Summary

A *smart home* is a home with a collection of internet-connected devices that a homeowner installs and operates in their home environment. A home Internet of Things (IoT) deployment allows a homeowner to remotely and more effectively control physical aspects of the home. For example, a homeowner might want lights to turn on or off, a thermostat setting to change at certain times of the day, or a security camera to send an alert when someone is around the house. While IoT devices introduce great conveniences to the homeowner, it is important to understand the cybersecurity implications of adding IoT devices to a home network.

This document reports the results of a technical review of security features of the following smart-home device categories: light bulbs, security lights, security cameras, doorbells, plugs, thermostats, and televisions. For each device category, the project reviewed a minimum of three devices from different manufacturers that were readily available from major retailers. The review enumerated the devices' technical properties and behaviors, by conducting open-source research and performing hands-on technical reviews. More intrusive review techniques, such as disassembling an IoT device to study its internal components in detail, were out of scope.

The purpose of this project is to review the security features available on a small sample of consumer home IoT devices and develop general considerations for IoT-device manufacturers to improve the security of consumer home IoT devices. This review focused solely on the security aspects of the IoT devices and did not include a security review of other IoT components or the ecosystem. Though many popular categories of IoT devices were sampled, due to logistical limitations, each sample was relatively small compared to the scale of IoT devices available for purchase, and not all product categories for home IoT were included.

The review showed that security feature implementation varied from IoT device to IoT device. For example, in general, different types of encryption were used for communications between the IoT device and other components of the ecosystem, such as communicating with the manufacturer's website when setting up a device. The results provided insights into areas where manufacturers did not use security features and encryption that are considered best practices. Preliminary versions of draft NISTIR 8259 [1] were used as the basis of defining and characterizing best-practice security features, because draft NISTIR 8259 was being developed at the same time our reviews were being performed.

The following is a list of the general considerations to improve IoT devices' security based on the project's findings:

- Password requirements for some companion mobile application and web application logins were weak. Manufacturers should consider requiring the user to establish a new application password, with strength requirements consistent with NIST Special Publication (SP) 800-63 best practices, upon a device's initial configuration [2].
- Mobile devices have settings that allow for a man-in-the-middle proxy. More than half of the consumer home IoT devices allowed someone to view all the data between the companion mobile application and the device by using a man-in-the-middle proxy tool, which could be exploited by a malicious attacker. Manufacturers should consider using certificate pinning [3], which associates a host with its expected certificate or public key;

this would help to mitigate man-in-the-middle attacks or certificate impersonation techniques used by attackers.

- Some devices used older versions of Transport Layer Security (TLS) encryption or no encryption at all for communications or software/firmware updates. Manufacturers should use TLS encryption suites as recommended by NIST SP 800-52 Revision 2, *Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations* [4], to protect updates and other sensitive data being communicated to and from devices.
- Some devices had open ports that attackers could manipulate. Manufacturers should close or otherwise prevent access to all of a device's unused physical and logical access ports, including physical accesses such as universal serial bus (USB).
- IoT devices commonly have a physical reset button, which attackers could leverage to gain access. This is problematic for security-related IoT devices placed outside the home. Manufacturers should not implement device reset buttons on security-related IoT devices outside the home.
- Though updates were posted by manufacturers for some of the devices we observed during the study period, there were known vulnerabilities for which updates were not provided. Manufacturers should develop and implement processes to make software and firmware updates for devices available and to notify users in a timely manner, consistent with best practices.
- UPnP [5], a plug-and-play communications protocol, was used by some devices for communications, but by default it does not use authentication. Manufacturers should implement additional device protections to secure UPnP communications.
- Keeping a device's cybersecurity features user-friendly for nontechnical users is a challenge. Manufacturers should consider applicability and best-practice implementations for all features in their devices, to support strong cybersecurity objectives.

Other considerations that may be specific to certain categories of IoT devices are highlighted in Section 3 of this document.

Table of Contents

List of Appendixes

List of Figures

1 Introduction

1.1 Purpose and Scope

This document reports the results of a project that conducted a technical review of the security features of consumer home Internet of Things (IoT) devices, also known as smart-home devices. Reviews were conducted on devices from the following categories of consumer home IoT devices: light bulbs, security lights, security cameras, doorbells, plugs, thermostats, and televisions.

For each IoT-device category, the project team reviewed a minimum of three devices from different manufacturers. The project team selected these IoT devices based on open-source research gathered from well-known retail and manufacturer websites. Information gathered included:

- device availability: devices selected were deemed to be easily and widely available through multiple sources
- device installation complexity: preference was given to devices a homeowner could install independently
- device price point: consideration was paid to all price points in each category

Selected IoT devices represent a small sample of consumer home IoT devices that are readily available to consumers. Many more product categories exist, as do product options within each of these categories. Therefore, this report is based on non-exhaustive samples of some categories of home IoT devices.

The reviews enumerated the IoT devices' technical properties and behaviors by conducting open-source research and performing hands-on technical review, but did not use more intrusive review techniques, such as disassembling an IoT device to study its internal components in detail. Analysis of the information collected by the review methodology focused on the security features available on consumer home IoT devices. This produced general considerations for device manufacturers to improve the security features offered on consumer home IoT devices, to meet cybersecurity best practices, but the observations and considerations in this report may not apply to all IoT devices or device categories.

IoT hubs, which fulfill a variety of services, including connecting IoT devices to the manufacturer's backend solutions and voice-recognition functionality, are out of scope for this project. Cloud-based services and other services, often used by manufacturers for IoT-device operations and maintenance, are also out of scope for this project. The security of these external components is important to the overall security of the consumer home IoT ecosystem and should be explored.

Throughout this document, the terms *consumer home IoT device*, *IoT device*, and *device* are used interchangeably.

1.2 Document Structure

The remainder of this document is organized into the following major sections and appendixes:

- Section 2 provides an overview of the IoT-device security-review methodology used in this project.
- Section 3 details the observations in the review for each category of IoT device included in the project.
- Section 4 summarizes findings and identifies considerations for cybersecurity features that all consumer home IoT devices should support.
- The References section provides a list of citations and relevant work associated with this report.
- Appendix A explains the review methodology in more detail.
- Appendix B provides a list of acronyms used in this document.

2 IoT-Device Security-Review Methodology

The consumer home IoT-device security-review methodology used in this project included two types of review: 1) open-source research focused on reviewing publicly accessible documentation, and 2) hands-on review in a lab-based "home" environment to observe or identify cybersecurity features in consumer home IoT devices. More intrusive review techniques, such as disassembling the IoT device to study its internal components in detail, were out of scope for this project. Additional information about the two types of review can be found in Appendix A.

The project team performed the reviews to:

- understand the technical and cybersecurity features of consumer home IoT devices
- understand how those features compared across the IoT-device category (e.g., how a single light bulb compared with the other light bulbs reviewed)
- determine if all categories of reviewed devices offered similar cybersecurity features

Consumer home IoT devices were deployed in a lab-based "home" environment, as depicted in the high-level notional architecture diagram in Figure 1. These IoT devices generally connect to the home wireless network to communicate with manufacturers' servers on the internet. Smart functions can be managed by companion mobile applications or web applications within the home or remotely.

Technical reviews were then conducted and based on a set of usage scenarios. The scenarios were modified as needed to account for the unique characteristics of each IoT device and the information already gathered during the review. The scenarios addressed the following objectives:

- review the IoT-device communications and authentication mechanisms, as well as other devices or networks with which the IoT device communicates
- explore the available security settings for configuring the IoT device, its data collection, or both
- analyze the IoT-device's security features, based on information collected during review

Preliminary versions of draft NISTIR 8259 [1] were used to guide the security review of the observations gathered through the two review methodologies, because draft NISTIR 8259 was being developed at the same time our reviews were being performed. Given the breadth of devices explored across categories, the *Core Features Baseline* presented in Section 4 of draft NISTIR 8259 was used to drive this analysis.

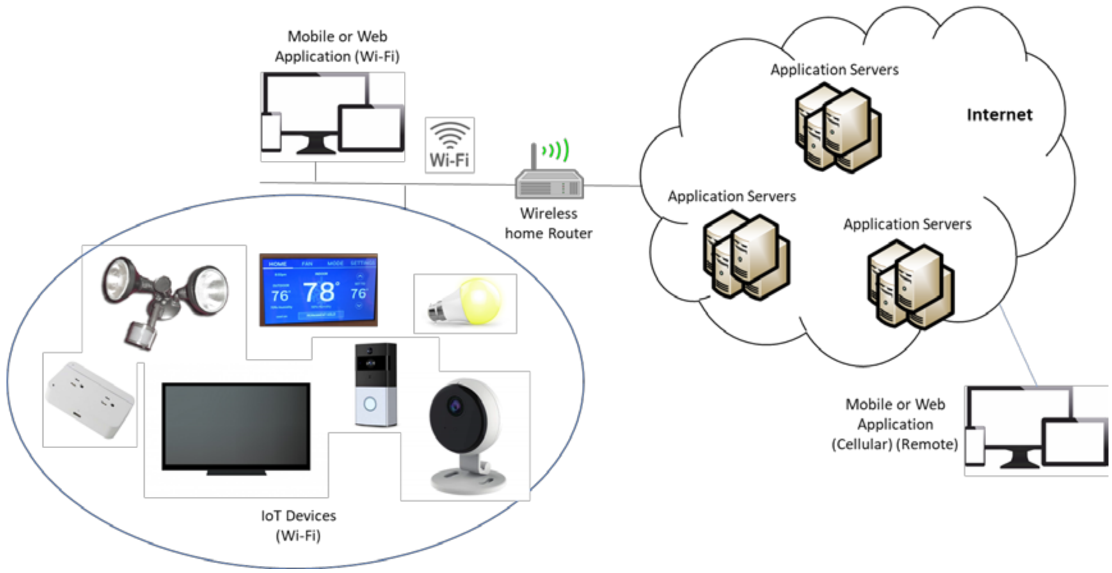

Figure 1: Notional Consumer Home IoT Architecture

3 Observations

This section reports noteworthy observations made by the team during the open-source research and hands-on review. Each subsection addresses a different category of consumer home IoT devices. The structure of each subsection is the same:

1. A summary of findings for the products through open-source research (i.e., information about networking protocols supported, options for device controls, and any available security information about the device). Because the open-source research yielded limited information, only identified security characteristics are mentioned.
2. Observations from the hands-on review, including information about wireless network usage, connections the devices make to Internet Protocol (IP) addresses and domain names, the devices' use of encryption for communications, and any other noteworthy observations.
3. An analysis of security features based on the information collected through open-source research and hands-on review.

3.1 Smart Light Bulbs

The team reviewed several smart light bulbs, each from a different manufacturer. All the light bulbs required a companion mobile application that was provided by the manufacturer, which the user would use to set up and communicate with the light bulb. Some light bulbs required hubs to realize certain functionality. The scope of this project, however, was limited to just the light bulbs.

3.1.1 Open-Source Research

The open-source research yielded the following information:

Networking: Most of the light bulbs reviewed supported Wi-Fi for networking. One light bulb supported Zigbee.

Device Control and Capabilities: All light bulbs could be controlled through manufacturer-provided iOS and Android companion mobile applications and by voice commands issued to certain other IoT devices (e.g., smart speakers). To set up each device, the user was required to create an account with a username and password through the companion mobile application.

Security: Some password length requirements were found. Many of the light-bulb manufacturers reviewed posted patch notifications of security vulnerabilities on their websites. Firmware updates were automatically pushed to the light bulbs.

3.1.2 Hands-On Review

The hands-on review identified several characteristics of interest:

Wireless Networks: The light bulbs with Wi-Fi used Wi-Fi Protected Access 2 (WPA2) for data protection and to secure network access to the home Wi-Fi network. However, these bulbs also included their own Wi-Fi access points that were used without any protection for the bulbs' initial setup and configuration. Once the bulbs joined the home Wi-Fi network, they disabled their own Wi-Fi access points. For IoT devices without a physical user interface (e.g., USB port or button), this is a common feature to support initial setup. Also, one light bulb would not connect to a Wi-Fi network unless the network had some form of security, such as Wired Equivalent Privacy (WEP) or WPA encryption.

Connections to IP Addresses and Domain Names: Each light bulb connected to numerous IP addresses, but often several IP addresses resolved to the same domain name. The number of domain names interacting with each light bulb ranged from four to 10, and the average was six. In all cases, the manufacturers' application servers were hosted by cloud service providers. Exploring these aspects is out of scope, as noted in Section 1.1. Other domain names were also identified that suggest services for mobile application crash reporting, marketing, and data analysis.

Communications Protection: The light bulbs used standard protocols, such as Hypertext Transfer Protocol (HTTP), Hypertext Transfer Protocol Secure (HTTPS), and Transport Layer Security (TLS), for communicating with other devices and protecting those communications. Not all communications with the light bulbs were protected, but the vast majority were. Half the bulbs protected all of their communications with TLS 1.2 [4]. The other bulbs did minimal HTTP communications without any encryption, and one bulb used TLS 1.0, which has been deprecated [4], for communicating one piece of data. The information exposed via HTTP did not include user data. Cryptographic suites could not be identified for most connections, but each light bulb had at least one connection where the encryption suite could be detected, and in all cases, the suites were consistent with best practices. Interestingly, one of the light bulbs could accept stronger cryptographic options than the server offered. This information was observed during the TLS handshake exchange between the light bulbs and other devices, such as application servers and companion mobile applications.

One bulb's companion mobile application used certificate pinning [3], which mitigated man-in-the-middle attacks and thus limited how much of its network communications could be examined during the review.

Communications Observations: Some light bulbs clearly had specific parts of their communications occurring with different domain names, such as login credentials, bulb control, smartphone information, and software and firmware updates.

Other: One light bulb had no strength requirements for passwords created on its companion mobile application, but creating an account through the manufacturer's website to interact with the bulb did require meeting password strength requirements that align with best practices. For all bulbs, a complete reset was available through physical means only. For some bulbs, a soft reset was available, but it did not erase data available for viewing on the companion mobile application. There was no method to identify or confirm whether user data was erased from the manufacturer's servers for complete resets and soft resets.

Only one of the bulbs could still be controlled by a companion mobile application when internet connectivity was lost (assuming the device running the application was on the same local network as the bulb). All bulbs that lost power were able to return to their previous secured state when power was restored.

3.1.3 Security Features Analysis

These are the results of analyzing the information collected during open-source research and hands-on review:

Device Identification: The light bulbs did not have unique physical device identifiers; however, they all had media access control (MAC) addresses that could be used as unique logical device identifiers.

Software and Firmware Update: Updates could not be automatically downloaded and installed by any of the light bulbs; all light bulbs required a human to use a companion mobile application or web application and specifically authorize each update. All light bulbs used TLS 1.2 to protect their update communications. All but one of the light bulbs required an authorized user to be logged into their corresponding application to update the light-bulb software. The other light bulb had the option of updating through a web application that did not require an authorized user. For most bulbs, their companion mobile applications could initiate or ignore the update. However, security configuration options for updates were limited, and none of the light bulbs offered a rollback capability to restore the previous software version if installing an update caused problems.

Device Configuration: Many of the light bulbs required password-based authentication to log in to their applications and change the bulbs' configuration settings. None of the bulbs had default passwords. Most of the light bulbs had reasonable password strength requirements, such as minimum password length with uppercase letters, lowercase letters, and numbers. One light bulb allowed trivially short and simple passwords that could easily be guessed by brute force. None of the bulbs offered configuration settings for disabling unneeded services and ports.

Device Reset: All the light bulbs offered a device reset capability that wiped data from the device, but the extent to which the data was wiped could not be determined without using invasive review techniques.

Data Protection: Most communications were protected using TLS 1.2, but one bulb used an old TLS version (1.0) for some of its communication, and another bulb used no encryption for certain portions of its communication. Data-at-rest protection was not observed for any of the light bulbs. The review did not include using invasive or destructive memory review techniques.

Security Event Logging: No security event logging capabilities were available to the user. The only type of information logged by any of the bulbs was usage statistics, such as when the bulb was on or off, which were accessible on the bulbs' companion mobile applications.

Interface Access: None of the light bulbs had physical user interfaces. The companion mobile application allowed a user to control the bulb locally or remotely, which required a user to log in to the application by using a valid username and password. There was no way to disable unneeded network interfaces, such as open ports, on any of the bulbs.

Application access varied by manufacturer. For one light bulb, the account that initially set up the bulb and connected it to Wi-Fi was the owner and primary account. Other user accounts could control the light bulbs but needed the application and access permission from the owner account. For another bulb, anyone on the Wi-Fi network with the companion mobile application could see the bulb and control it after setup, but only users signed into the main account would be able to edit the bulb's settings and access them remotely. For a third bulb, only one account could access the bulb, but that account could be used on different mobile devices.

3.2 Smart Security Lights

The team reviewed several security lights, each from a different manufacturer. All the security lights required a companion mobile application that was provided by the manufacturer, which would be used by the user to set up and communicate with the device.

3.2.1 Open-Source Research

The open-source research yielded the following information:

Networking: Most security lights supported Wi-Fi. One supported Bluetooth Low Energy for communications.

Device Control and Capabilities: All the security lights could be controlled through manufacturer-provided iOS and Android companion mobile applications and by voice commands issued to certain other IoT devices (e.g., smart speakers). One could also be controlled by web applications. To set up some of the security lights, the end user needed to first create an account login and password through the security light's companion mobile application.

Each security light could turn on or off based on its sensors and on demand by using its companion mobile application. In addition:

- One could change its light colors and how often it turned the light on and off.
- Two of them had cameras they could activate.
- One of them had an audible alarm.

Security: One security light did not require a password for local network access. Another required a password of at least six characters but did not specify additional strength requirements. A third security light also enforced a minimum password length of six characters, but it required a mix of character types (uppercase, lowercase, etc.) to help improve password strength.

3.2.2 Hands-On Review

The hands-on review identified several characteristics of interest for the security lights:

Wireless Networks: One of the security lights used WPA2 to protect its communications, while the others had their own open Wi-Fi networks during the initial setup. Once those security lights joined the home Wi-Fi network, they disabled their own Wi-Fi access points.

Connections to IP Addresses and Domain Names: Each security light connected to numerous IP addresses, but often several IP addresses resolved to the same domain name. The number of domain names interacting with each security light ranged from eight to 15, and the average was 12. In all cases, the application servers were hosted by cloud service providers.

Communications Protection: The security lights protected their communications with TLS 1.2, except Network Time Protocol (NTP) traffic. All the security lights supported a number of cryptographic suites that were consistent with best practices, although one light also supported suites such as TLS_RSA_WITH_RC4_128_MD5 that are not considered best practices. One of the security lights used a virtual private network (VPN) to establish a protected tunnel for its video-camera data stream. The VPN used TLS 1.2 with a cryptographic suite consistent with best practices.

One security light did not protect its communications for firmware updates, which does not follow best practices.

One security light's companion mobile application used certificate pinning [3], which mitigated man-in-the-middle attacks and limited how much of its network communications could be examined during the review.

Communications Observations: The security lights clearly had specific parts of their communications occurring with different domain names, including:

- time servers (all lights)
- initial light setup (some)
- statistics and metrics (most)
- firmware updates (all)
- user-behavior tracking (most)
- command and control (all)
- video-camera feed (some)
- login credentials (some)
- technical support (some)

Three servers were used by one security light, and their purpose could not be determined.

Other: Inspection of the update of one security light showed there was no verifiable cryptographic means of preserving the integrity of the update file. Knowing the upgrade path and file name, a malicious user could masquerade as the update server, push out a file, and install custom firmware on the device.

3.2.3 Security Features Analysis

These are the results of analyzing the information collected during open-source research and hands-on review:

Device Identification: The security lights all had MAC addresses physically labeled on them as physical device identifiers. Some also had unique serial numbers printed on their cases, and these serial numbers were used as both unique physical identifiers and unique logical identifiers. One light used its MAC address as its unique logical identifier.

Software and Firmware Update: Most of the security lights used TLS 1.2 to protect their update communications. One used unprotected communications for some of its update communications. All the security lights required an authorized user to be logged in to the companion mobile application for the device's software/firmware to be updated. None of these applications had security configuration options for updates. Also, none of the security lights offered a rollback capability to restore the previous software version if installing an update caused problems.

Device Configuration: Most of the security lights required password-based authentication to log in to their applications and change the lights' configuration settings; one used authentication only for remote access from outside the home network. None of the security lights had default passwords. For the lights that required passwords, most had strong password strength requirements, such as an eight-character minimum that must include at least one uppercase letter, one lowercase letter, one number, and one symbol. Others had minimum requirements of six characters with no strength requirements, which does not follow best practice. None of the security lights offered configuration settings for disabling unneeded services and ports.

Device Reset: All the security lights offered a device reset capability that wiped data from the device, although the extent to which the data was wiped could not be determined. Most of these device resets occurred through the lights' companion mobile applications, while the rest were through a physical reset button on the light. For the security lights that had open Wi-Fi networks during initial setup, a device reset triggered the initial setup process, and data was removed from the companion mobile applications.

Data Protection: Most communications were protected using TLS 1.2, but a small amount used no encryption at all. Sensitive information was not exposed for communications that did not use encryption. As for protection of data at rest, none of the security lights provided any visibility into the state of their data storage, so this could not be analyzed without using invasive review techniques.

Security Event Logging: One of the security lights did not have any security event logging capabilities, either through its companion mobile application or through the manufacturer's website. The others performed event logging of the physical security events monitored by the security-light devices, but cybersecurity events were not available on either the manufacturer websites or the companion mobile applications.

Interface Access: One of the security lights did not have any physical user interfaces; one did not have any physical user interfaces exposed once it was wall mounted; and one had local interfaces with no protection for them. Remote access to most of the security lights was restricted by requiring a valid username and password for the corresponding application. There was no way to disable unneeded network interfaces, such as open ports, on any of the lights.

3.3 Smart Security Cameras

The team reviewed several security cameras, each from a different manufacturer. All the security cameras required a companion mobile application that was provided by the manufacturer, with which the user would set up and communicate with the device.

3.3.1 Open-Source Research

The open-source research yielded the following information:

Networking: The security cameras supported Wi-Fi for networking, and one could also connect to Ethernet.

Device Control and Capabilities: The security cameras could be controlled through manufacturer-provided iOS and Android companion mobile applications and by voice commands issued to certain other IoT devices (e.g., smart speakers). Most of the security cameras offered access through a web application. To set up each device, the end user needed to create an account login and password through one of the applications (either mobile or web).

Security: Some password length requirements when creating the user account were found. One device had a unique username and password for logging on to the application programming interface (API), and the API then provided a token for each device.

3.3.2 Hands-On Review

The hands-on review identified several characteristics of interest for the smart security cameras:

Wireless Networks: The security cameras with Wi-Fi used WPA2 or WPA-Temporal Key Integrity Protocol (WPA-TKIP) to protect their communications. However, these security cameras also included their own Wi-Fi access points that were used without any protection for initial setup and configuration. Once the cameras joined the home Wi-Fi network, they disabled their own Wi-Fi access points.

Connections to IP Addresses and Domain Names: Each security camera connected to numerous IP addresses, but often several IP addresses resolved to the same domain name. The domain names interacting with each security camera ranged from four to 10. In all cases, the application servers were hosted by cloud service providers. The types of servers common across all the devices were NTP, user login, application, and firmware/software update servers.

Communications Protection: The security cameras protected their communications with TLS 1.2. Similar cryptographic suites were identified for most connections. One device primarily used TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384, while the others used TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256. These suites are consistent with best practices based on NIST SP 800-52 Revision 2 [4].

One security camera's companion mobile application used certificate pinning [3], which can mitigate man-in-the-middle attacks, and limited how much of its network communications could be examined during the review. Another camera's companion mobile application used an older

API and accepted the user's proxy certificate, allowing HTTPS traffic to be viewed using the proxy. A third camera's companion mobile application used an API where user proxy certificates were not enabled unless the application was modified to do so.

Communications Observations: The devices had specific parts of their communications occurring with different domain names, such as login credentials, streaming, smartphone information, and software and firmware updates. One device did not use TLS encryption for its firmware update. Another device used the Session Initiation Protocol (SIP) [6] to establish a connection without TLS encryption.

Other: Most of the companion mobile applications for the devices did not communicate directly with the device. One of the devices used User Datagram Protocol (UDP) to communicate with both the companion mobile application and the application servers in the cloud.

Any person with physical access to the device could gain complete access to the device by resetting it. For all devices, a complete reset was available through physical means only. There was no means to reset the devices through the applications. Each application could remove the device but could not reset the device itself.

3.3.3 Security Features Analysis

These are the results of analyzing the information collected during open-source research and hands-on review:

Device Identification: The devices had unique serial numbers labeled. Most of the devices had MAC addresses that could be used as unique logical device identifiers. The other devices used the serial number as the logical identifier.

Software and Firmware Update: Most of the security cameras used TLS 1.2 to protect their update communications, while the rest did not use any encryption. While most of the devices required an authorized user to be logged in to their companion mobile application to update the software, the other devices performed updates automatically. None of the companion mobile applications could cancel the update. However, security configuration options for updates were limited, and none of the devices offered a rollback capability to restore the previous software version if installing an update caused problems.

Device Configuration: The security cameras required password-based authentication in order to log in to their applications and change the devices' configuration settings. None of the security cameras had default passwords. Minimum password requirements were six characters, eight characters, and six characters, with at least one uppercase, one lowercase, and one number. One application had a login/password for the API, which provided a token for accessing the device itself. Access to this device was lost once the device was removed from the application or was reset. None of the security cameras offered configuration settings for disabling unneeded services and ports.

Device Reset: The devices offered a physical device reset capability. However, it could not be determined if data was wiped cleanly from the devices. In one device, previous recordings were not erased from the local micro Secure Digital (microSD) card after a reset. With resets, the process of initial setup needed to be performed again.

Data Protection: Most communications were protected using TLS 1.2, but two specific sets of communication from two security cameras were not encrypted. For one device, the SIP setup for video was not encrypted. For another device, communications to cloud servers and download of firmware were not encrypted. As for protection of data at rest, most of the security cameras provided no visibility into the state of their data storage, so this could not be analyzed without using invasive review techniques. The security camera with the microSD card did not encrypt the data; someone could pull the videos from the microSD card to view or edit them.

Security Event Logging: None of the security cameras had any security event logging capabilities available to the user. The only type of information logged by any device was motion event logs, which were accessible on the companion mobile application.

Interface Access: One security camera had a local interface for the microSD card. Any person with physical access could retrieve the microSD card. The method for restricting remote access to all security cameras was requiring a valid username and password for the application. There was no way to disable unneeded network interfaces, such as open ports, on any of the security cameras. All security cameras could appear on only one account at a time.

3.4 Smart Doorbells

The team reviewed several doorbells, each from a different manufacturer. All the doorbells required a companion mobile application that was provided by the manufacturer for the user to set up and communicate with the device.

3.4.1 Open-Source Research

The open-source research yielded the following information:

Networking: The doorbells supported Wi-Fi for networking. One also had Bluetooth capabilities.

Device Control and Capabilities: The doorbells could be controlled through manufacturer-provided iOS and Android companion mobile applications and by voice commands issued to certain other IoT devices (e.g., smart speakers). Each doorbell included a camera to record activities, and most of those cameras included night-vision capabilities. Each doorbell also had a microphone and a speaker for two-way audio communications, and a light-emitting diode status light. Most of the doorbells offered motion detection.

Security: Password length requirements were found for creating the user account for one doorbell.

3.4.2 Hands-On Review

The hands-on review identified several characteristics of interest for the doorbells:

Wireless Networks: One of the doorbells included its own Wi-Fi access point that was used without any protection for initial setup and configuration. Once it joined the home Wi-Fi network, it disabled its own Wi-Fi access point.

Connections to IP Addresses and Domain Names: Each doorbell connected to numerous IP addresses, but often several IP addresses resolved to the same domain name. The number of domain names interacting with each doorbell ranged from five to 10, with seven as the average. In all cases, the application servers were supported by cloud services. The types of servers common across all the devices were video transmission and firmware/software update servers. Other identified servers included NTP, audio transmission/streaming, and doorbell press notification. The purpose of several servers could not be determined.

Communications Protection: Most of the doorbells protected their communications with TLS 1.2 cryptographic suites that followed best practices consistent with NIST SP 800-52 Revision 2 [4]. One doorbell used the TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 cryptographic suite, and the other used the TLS_RSA_WITH_AES_128_CBC_SHA suite. One doorbell used TLS 1.0, an older form of TLS, and used the TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA suite. Another doorbell used AES_CM_128_HMAC_SHA1_80 to encrypt its video and audio streams.

Other: If internet connectivity were lost, the doorbells could no longer be controlled by their companion mobile application. One of the doorbells used UDP to communicate with both the application and the application servers in the cloud.

3.4.3 Security Features Analysis

These are the results of analyzing the information collected during open-source research and hands-on review:

Device Identification: Most of the doorbells had unique serial numbers labeled, and the rest had the MAC address printed on the device. The same identifiers were used for logical identification for each device.

Software and Firmware Update: Most of the doorbells used TLS 1.2 to protect their update communications, while the rest used TLS 1.0, which is deprecated. Devices had to be registered to an account by a logged-in user to get an internet connection, which facilitated their automatic update process. None of the doorbells offered any security configuration options for updates, and none of the doorbells offered a rollback capability to restore the previous software version if installing an update caused problems.

Device Configuration: The doorbells required password-based authentication to log in to their companion mobile applications and change the doorbells' configuration settings. Requirements for passwords were eight characters, with only one device requiring a mix of letters, numbers, and symbols. None of the doorbells offered configuration settings for disabling unneeded services and ports.

Device Reset: The doorbells offered a physical device reset capability. Previous recordings were no longer accessible from the companion mobile application, but it could not be determined if the data was wiped cleanly from the devices. With resets, the initial setup needed to be performed again. The reset would reinstate the open Wi-Fi access that the doorbell uses for setup.

Data Protection: Most communications were protected using TLS 1.2, but some communications used TLS 1.0, and some were not encrypted. As for protection of data at rest, none of the doorbells provided visibility into the state of their data storage, so it could not be analyzed without using invasive review techniques.

Security Event Logging: None of the doorbells had any security event logging capabilities available to the user. However, the doorbells had motion or event logging, which were accessible on their companion mobile applications.

Interface Access: Any person with physical access could reset any of the doorbells and access their local interfaces (e.g., micro Universal Serial Bus [USB] port). The method for restricting remote access to all doorbells was requiring a valid username and password for the companion mobile application. There was no way to disable unneeded network interfaces, such as open ports, on any of the doorbells. All doorbells could appear on only one account at a time.

3.5 Smart Plugs

The team reviewed several smart plugs, each from a different manufacturer. All the smart plugs required a companion mobile application that was provided by the manufacturer, which was used to set up and communicate with the device.

3.5.1 Open-Source Research

The open-source research yielded the following information:

Networking: The smart plugs supported Wi-Fi for networking. Once connected to a Wi-Fi network, these devices communicated via IP.

Device Control and Capabilities: The smart plugs could be controlled through manufacturer-provided iOS and Android companion mobile applications. Most of the devices could use voice commands issued by certain other IoT devices (e.g., smart speakers). To set up each device, the end user needed to create an account login and password through the companion mobile application.

Security: Some password length requirements for creating the user account were found. Open-source research described encryption issues with one of the smart plugs. Because the manufacturer used simplistic encryption, a hard-coded encryption key, and no authentication, an attacker could easily send encrypted commands to an open port on the device, allowing control of the device without pairing. A second smart plug contained a vulnerability that allowed anyone to flash custom firmware to the plug, whether they had remote or physical access to the plug or not.

3.5.2 Hands-On Review

The hands-on review identified several characteristics of interest for the plugs:

Wireless Networks: The smart plugs communicated with the router by using WPA2 encryption. Most of the plugs had open Wi-Fi during setup. Once those plugs joined the home Wi-Fi network, they disabled their own Wi-Fi access points. Another plug used an eight-digit code during setup that was provided on a piece of paper in the box. During setup, the smartphone scanned the code, which paired the phone with the plug.

Connections to IP Addresses and Domain Names: Each smart plug connected to numerous IP addresses, but often several IP addresses resolved to the same domain name. The domain names interacting with each plug ranged between five and nine. The types of servers common across all the devices were NTP, user login, application, and firmware/software update servers.

Communications Protection: The smart plugs protected some of their communications with TLS 1.2. Most of the plugs also used HTTP to communicate with certain servers. All plugs used different types of encryption suites. These suites were consistent with best practices.

Most of the smart plugs used certificate pinning [3], which mitigated man-in-the-middle attacks and limited how much of their network communications could be examined during the review. The companion mobile application associated with another plug accepted the proxy certificate and allowed the traffic to be viewed.

Communications Observations: The devices had specific parts of their communications occurring with different domain names, such as login credentials, smartphone information, and software and firmware updates.

Other: The smart plugs could still function properly as plugs without the smart functions. Only one of the plugs could still be controlled by a companion mobile application when internet connectivity was lost (assuming the device running the application was on the same local network as the smart plug). The other smart plugs did not have communications with their companion mobile application.

3.5.3 Security Features Analysis

These are the results of analyzing the information collected during open-source research and hands-on review:

Device Identification: One of the smart plugs had the MAC address displayed on the box. The other plugs did not have a unique physical identifier. Most of the plugs had MAC addresses that could be used as unique logical device identifiers. The other plugs used the serial number as the logical identifier.

Software and Firmware Update: The smart plugs used TLS 1.2 to protect their update communications, but not all their other communications used TLS 1.2. All smart plugs required an authorized user to be logged in to their corresponding companion mobile application to update the software. The applications with notifications of updates were unable to stop the update.

However, security configuration options for updates were limited, and none of the devices offered a rollback capability to restore the previous software version if installing an update caused problems.

Additionally, some vulnerabilities identified through open-source research had not been patched as of August 2019. Examples include a vulnerability publicly known since 2016 that allowed a device to be controlled without being paired, and a vulnerability publicly known since 2018 that allowed custom firmware to be flashed to the device.

Device Configuration: The smart plugs required password-based authentication to log in to their companion mobile applications and change the devices' configuration settings. The password requirements for the plugs were six or eight characters. Note that once logged in to the application on the smartphone or tablet, the user stayed logged on. None of the smart plugs had default passwords. However, one plug had a device personal identification number (PIN) that was used during setup. For all plugs, removing the device from the companion mobile application reset the plug back to factory default, which required initial setup again. None of the smart plugs offered configuration settings for disabling unneeded services and ports.

Device Reset: The smart plugs offered a physical device reset capability with a button on the device. However, it could not be determined if data was wiped cleanly from the devices. Reset could also be completed by deleting the device on the companion mobile application. With resets, initial setup needed to be performed again. Upon loss of power, the device maintained the configuration it had prior to the outage.

Data Protection: Communications were protected using TLS 1.2 for all smart plugs. As for protection of data at rest, all plugs provided no visibility into the state of their data storage, so it could not be analyzed without using invasive review techniques. There were no settings on the companion mobile applications to modify encryption mechanisms.

Security Event Logging: The smart plugs did not have any security event logging capabilities available to the user. All companion mobile applications logged usage statistics from the plugs, which were accessible from the applications.

Interface Access: The devices did not have physical user interfaces. Access to the devices was through their companion mobile applications. There were no configuration settings to disable services or restrict remote access. Once the application was paired with the plug, anyone with a username and password could access the device.

3.6 Smart Thermostats

The team reviewed several smart thermostats, each from a different manufacturer. The thermostats were designed to function in environments without IoT hubs.

3.6.1 Open-Source Research

The open-source research yielded the following information:

Networking: The smart thermostats supported Wi-Fi for networking.

Device Control and Capabilities: The thermostats had a physical user interface to control the settings and functions, and most had a USB port for local access to the device. All the thermostats could be controlled through manufacturer-provided iOS and Android companion mobile applications and manufacturer websites. All devices could use voice commands issued by certain other IoT devices (e.g., smart speakers). To set up each device, the end user needed to create an account login and password through the companion mobile application.

Security: Information was available about the password length requirements when creating the user account and the availability of a PIN to lock the thermostat for all devices. All devices had a USB port, and several research articles stated that one device was susceptible to malicious attack of the firmware if someone had access to the USB port. Another device might have been susceptible to cross-site scripting attacks.

3.6.2 Hands-On Review

The hands-on review identified several characteristics of interest for the smart thermostats:

Wireless Networks: The thermostats used WPA2 to protect their Wi-Fi communications. Unlike other IoT devices observed in this document, which had their own open Wi-Fi for setup, all the thermostats connected to the home wireless network during startup to reach the internet. Once they joined the home Wi-Fi network, they registered with the servers before communicating with the companion mobile application.

Connections to IP Addresses and Domain Names: Each thermostat connected to numerous IP addresses, but often several IP addresses resolved to the same domain name. The domain names interacting with each thermostat ranged between two and five. In all cases, the application servers were supported by cloud services. One thermostat communicated with only one server for most of its functions after communicating with an NTP server for time. Another thermostat communicated with the same domain name, which consisted of three different IP addresses. The types of servers common across all the devices were NTP, user login, application, and firmware/software update servers.

Communications Protection: Most of the thermostats protected their communications with TLS 1.2, while the others used Secure Sockets Layer (SSL), the predecessor to TLS that has been deprecated. The thermostats that used TLS 1.2 used the following suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256. This suite was consistent with best practices.

The thermostats' companion mobile applications used certificate pinning [3], which limited how much of their network communications could be examined during the review.

Communications Observations: Most of the thermostats communicated with one domain name, which means all types of communications were handled through that single domain name.

Other: The thermostats could work without smart functions. All functions of the thermostats could be locally controlled via the API on the device itself. The companion mobile applications communicated with the thermostats via the internet. If the thermostats lost connectivity to the internet, they could not communicate with the applications, but normal functions were not affected.

Any person with physical access to the device could gain access to the thermostats by resetting them. However, all the applications could turn on the PIN lock so that the thermostats' API could be locked from local access. Most of the thermostats used a unique key to set up the device communication with the application. The application for the other thermostats used the MAC address of the device to set up the connection to the device.

3.6.3 Security Features Analysis

These are the results of analyzing the information collected during open-source research and hands-on review:

Device Identification: The devices had unique serial numbers labeled. All devices had MAC addresses that could be used as unique logical device identifiers. The MAC addresses were identified in all of the companion mobile applications. For all thermostats, a PIN was available and could be enabled to lock the API.

Software and Firmware Update: Most of the thermostats used TLS 1.2 to protect their update communications, while the rest did not. While most of the devices required an authorized user to be logged in to their corresponding companion mobile application to update the software, the rest of the devices performed updates automatically. The applications with notifications of updates were unable to stop the update. However, security configuration options for updates were limited, and none of the devices offered a rollback capability to restore the previous software version if installing an update caused problems. All thermostats could trigger an update locally on the device. One manufacturer provided logs of patch updates on its website.

Device Configuration: The smart thermostats required password-based authentication to log in to their companion mobile applications and change the devices' configuration settings. Most of the devices required eight characters minimum with a mix of letters, numbers, and symbols. The other devices required eight characters minimum only (no strength requirement). Configuration settings could also be made on the device, which could be locked by enabling a PIN. Device access was lost after a reset. In that case, initial setup procedures were needed to have the thermostats functioning again and communicating with the application. A new PIN would have to be configured again after the reset.

Device Reset: The devices offered a physical device reset capability. Anyone could perform the reset on the device if a PIN was not configured to lock the device. However, it could not be determined if data was wiped cleanly from the devices. With resets, initial setup needed to be performed again. Upon a power loss, all devices retained the configuration that was stored before the outage.

Data Protection: Communications were protected using TLS 1.2 for most of the thermostats. The other device did not use TLS but instead communicated using HTTPS with SSL, which is a deprecated method no longer considered a best practice. As for protection of data at rest, none of the thermostats provided visibility into the state of their data storage, so it could not be analyzed without using invasive review techniques. However, all devices had a USB port, which could be used to access the devices. None of the devices offered the ability to modify security configurations.

Security Event Logging: Most of the devices offered logging capabilities, while the rest did not. One device logged information in detail, including configuration changes. Another device logged event details such as temperature changes. There was no configuration to modify logging settings or to forward logs. Logs were observed using the device's companion mobile application.

Interface Access: Physical access to the device was possible unless a PIN was configured on the thermostats. The method for restricting remote access to all thermostats was requiring a valid username and password for the companion mobile application. There was no way to disable unneeded network interfaces, such as open ports, on any of the thermostats. Physical access to the thermostats was possible for most of the thermostats, because a USB port was available (likely intended for debugging or manual updates). Even with a PIN that locked the thermostats, someone with physical access to these thermostats could gain access through the USB port.

3.7 Smart Televisions

The team reviewed several smart televisions (TVs), each from a different manufacturer.

3.7.1 Open-Source Research

The open-source research yielded the following information:

Networking: The TVs supported Wi-Fi for networking, Ethernet, Bluetooth, and one or more USB ports for local access to the device.

Device Control and Capabilities: Like traditional TVs, all the smart TVs had a remote control for settings and functions. All the smart TVs could be controlled through manufacturer-provided iOS and Android companion mobile applications, manufacturer websites, and voice commands issued by certain other IoT devices (e.g., smart speakers). One application required a user login and password. Another application required a PIN from the TV. A third application did not require any authentication from the corresponding TV, but that application only had basic TV control functionality. Device setup was completed locally and through the remote control.

Security: Details of several known vulnerabilities in the products were found via open-source research. Some of the TVs could scan for malware.

3.7.2 Hands-On Review

The hands-on review identified several characteristics of interest for the smart TVs:

Wireless Networks: The TVs with Wi-Fi used WPA2 to protect their communications. All also supported using an Ethernet cable to connect the TV directly to the home router instead of using Wi-Fi. All TVs were configured by default to scan for Wi-Fi connections. Once the correct Service Set Identifier (SSID) was identified, the user could manually enter the password into the TV.

Connections to IP Addresses and Domain Names: Each TV connected to numerous IP addresses, but often several IP addresses resolved to the same domain name. The domain names interacting with each TV ranged between three and six. In all cases, the application servers were supported by cloud services. Note that the analysis did not account for different applications that were included in the TV. Most likely, testing those applications would result in more IPs and domain names in the analysis.

Communications Protection: One TV used TLS 1.2 encryption (TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384) for communication with one of the manufacturer's servers. It used HTTP for its update process, but the payload within the HTTP packet was encrypted. Another TV used HTTP for all communications, and application keys for authentication with their servers were in plaintext. While this TV did not use standard encryption to its own servers, it did use encryption to other services such as streaming content. A third TV used TLS 1.2 encryption (TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256) for some of its communications. However, it used HTTP to send its firmware update and used MD5 [7] (which is not considered a best practice) to check the validity of the firmware, so the firmware could be altered. Universal plug and play (UPnP) [5] was used by one companion mobile application to communicate with the TV. There was no authentication mechanism, which meant any user could connect to and control the TV.

Other: The scope of this review did not include applications within the TVs. However, a significant observation was that the first domain name query performed by all the TVs on initial startup was for a streaming service. All other applications needed to be started on the TVs before any communications happened. All the TVs had open ports when nmap was used to perform network analysis of the TVs. All the TVs had open ports that were not used for communications.

The TVs offered core TV functions, such as accessing and viewing local channels, without the need for smart functions. All TV functions could be locally controlled via the remote control or through companion mobile applications, but the user needed to be in front of the TV, because the output of the functions was shown on the TV screen.

3.7.3 Security Features Analysis

These are the results of analyzing the information collected during open-source research and hands-on review:

Device Identification: The devices had unique serial numbers physically labeled. All the devices used the serial number as unique logical device identifiers and were identified in the TV settings.

Software and Firmware Update: While there were no updates to the firmware for the TVs, communication between all TVs and their update servers was through HTTP. All devices could have automatic updates enabled or disabled. Firmware updates for all devices could also be completed by uploading the firmware via the USB port. Most of the TVs had patch information available through their websites. There did not seem to be a way to revert to a previous version of firmware through the settings, although firmware could be loaded through the USB port.

Device Configuration: The TVs could be configured via the remote or the TV locally, including network and feature settings, application setup, and a device reset. Most TVs had companion mobile applications with full functionality as the TV remote controls. The other companion mobile applications had minimal functionality such as power, volume, and channel selection. Most of the TVs required password-based authentication to log in to their applications and change the TVs' configuration settings. Most of the devices required passwords with an eight-character minimum and a mix of letters, numbers, and symbols. The others did not require a password.

Device Reset: The devices offered a physical device reset capability. However, it could not be determined if data was wiped cleanly from the devices. With resets, initial setup needed to be performed again for most of the TVs. The other TVs did not lose their communications with the companion mobile application or paired devices after the reset. Upon a power loss, all the devices retained the configuration that was stored before the outage.

Data Protection: Communications were protected using TLS 1.2 for all TVs except their firmware updates. As for protection of data at rest, none of the TVs provided visibility into the state of their data storage, so it could not be analyzed without using invasive review techniques. However, all devices had a USB port, which could be used to access the devices. None of the devices offered the ability to modify data protection configuration settings.

Security Event Logging: None of the devices offered any logging capabilities to the user.

Interface Access: There was no way to restrict access to physical user interfaces for any of the TVs. Most of the TVs had configuration settings to restrict remote access. Most TVs allowed visibility into which devices were connected and were able to disable those connections. One of those TVs was able to disable the help support feature. The rest of the TVs were unable to restrict remote access. There was no way to disable unneeded network interfaces, such as open ports, on any of the TVs. Each TV had multiple local ports. USB ports can be a source of attacks, because firmware and software can be loaded by someone with physical access to the TV.

4 Summary of Findings and Considerations

The results of the review showed that all reviewed IoT devices implemented at least some cybersecurity features. Common features that devices supported included secure communications among components of the consumer home IoT ecosystem using TLS 1.2, password protection for applications and devices, and secure access to the IoT devices from various user interfaces.

These features were not always implemented, though, or did not all have the same level of maturity across devices in a category. Many devices provided update features, but most categories had some issue with the security of the update process, such as lack of automatic download options; unprotected update communications; or insufficient control provided to the user to schedule or stop automatic updates, including the inability to roll back an update if needed. Regarding insecure communication of updates, some devices received updates over HTTP, and one device provided the location and file name of the update with no verifiable cryptographic means of preserving the integrity of the update file.

Encryption was available on many devices, but some devices used older, deprecated versions of TLS encryption or no encryption at all. Several instances were observed where HTTP was used for communications. In some instances, manufacturers did not use the strongest encryption suites supported and offered by devices and servers to secure their communications. In one case, a device had a hard-coded encryption key, which is not consistent with best practices. Manufacturers should use TLS encryption as recommended by NIST SP 800-52 Revision 2 [4] to protect communications containing updates and other sensitive data.

An update mechanism does not help mitigate vulnerabilities, if software and firmware updates are not provided in a timely manner. Though updates were posted by manufacturers for some of the devices we observed during the study period, there were known vulnerabilities for which updates were not provided. Manufacturers should develop and implement processes to make updates available in a timely manner, consistent with best practices.

Similarly, use of encryption, even following best practices, may be negated if attackers can use open ports to access and manipulate the functionality of the device. By our observations, some devices have open ports that are not used. Devices should close or otherwise prevent access to all unused physical and logical access ports, including physical accesses such as USB.

Outside the devices themselves, many devices had supporting companion mobile applications or web applications that used usernames/passwords to control access (notably, one device did not require a password). In general, despite the mechanism being there, password requirements for application logins were weaker than best practices. To address these concerns, manufacturers should consider requiring the user to establish a new application password, with strength requirements consistent with best practices, upon a device's initial configuration.

Observations identified a number of issues with connections between companion mobile applications and devices, beyond weak password requirements. More than half of the IoT devices allowed someone to view all the data between the companion mobile application and the device by using a man-in-the-middle proxy tool. Manufacturers should consider using certificate pinning [3], a technique that some of the observed devices' companion mobile applications used to secure themselves from man-in-the-middle attacks. Also, UPnP, a plug-and-play communications protocol, was used by some TVs for communications. By default, UPnP does not use authentication. Additional device protections should be used to secure UPnP communications.

Though not all devices were of the same maturity in terms of implementing security features, we did observe many features in devices that would be helpful to users in mitigating threats. Many devices did not log security events (data that home users may be unlikely to use directly), but some did—notably, most of the thermostats examined. The ability to reset and remove the connection between component mobile application and device was available on all smart plugs we looked at for this report. As noted above, updating features and interface access control via username/password were also commonly available. Most devices also used some method to protect their communications, which is a positive trend that can be strengthened through minor tweaks in the methods used, in most cases.

Regarding data protection, security event logging, and logical access to interfaces, striking the right balance in exposing these aspects for the user to configure (e.g., the actual device configuration, interfaces, logging) but keeping such access user-friendly for nontechnical users remains a challenge. Based on this review, it appears most manufacturers decided to make their devices black boxes with few aspects exposed. Some manufacturers may limit features such as extensive, configurable data protection or security event logging to security-focused home-device categories such as security cameras and door locks, and consider these features less critical for devices like smart thermostats and light bulbs. Manufacturers should consider applicability and best implementations for these and all features in their devices, to support strong cybersecurity objectives. For example, although allowing only authorized users to reset a device is generally considered a best practice, for home devices this may not be appropriate, such as wanting to allow a house guest to reset a smart light bulb. Several devices in our reviews had physical buttons that reset devices without checking for user authorization.

Finally, please note that the selected devices were a small sample of consumer home IoT devices that are readily available to consumers. Many more product categories exist, as do more product options within each of these categories, than we were able to realistically review. Due to this wide range, these observations and considerations may not apply to some devices or categories of devices. To recap, here is a summary of the report's considerations for manufacturers of consumer home IoT devices:

- Manufacturers should consider requiring the user to establish a new application password, with strength requirements consistent with best practices, upon a device's initial configuration.
- Manufacturers should consider using certificate pinning, a technique that some of the observed devices' companion mobile applications used to secure themselves from man-in-the-middle attacks.
- Manufacturers should use TLS encryption suites as recommended by NIST SP 800-52 Revision 2 [4], to protect updates and other sensitive data being communicated to and from devices.
- Manufacturers should close or otherwise prevent access to a device's physical and logical access ports that are not used, including physical accesses such as USB.

- Manufacturers should not implement device reset buttons on security-related IoT devices outside the home. It is common for IoT devices to have a physical reset button, which attackers could leverage to gain access. This is problematic for security-related IoT devices placed outside the home.
- Manufacturers should develop and implement processes to make software and firmware updates for devices available in a timely manner, consistent with best practices.
- Manufacturers should implement additional device protections to secure UPnP communications.
- Manufacturers should consider applicability and best implementations for all features in their devices to support strong cybersecurity objectives, such as keeping a device's cybersecurity features user-friendly for nontechnical users.

We intend these results to be a starting point for understanding the security features offered in current devices. Only a larger, broader, and more frequent survey and review of current consumer home IoT devices can truly approach a more comprehensive understanding of the security offered by these devices in general.

References

[1] Fagan M, Megas K, Scarfone KA, Smith M (2019) Core Cybersecurity Feature Baseline for Securable IoT Devices: A Starting Point for IoT Device Manufacturers. (National Institute of Standards and Technology, Gaithersburg, MD), Draft NIST Interagency or Internal Report (IR) 8259. https://doi.org/10.6028/NIST.IR.8259-draft

[2] Grassi PA, Garcia ME, Fenton JL (2017) Digital Identity Guidelines. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) 800-63-3, Includes updates as of December 1, 2017. https://doi.org/10.6028/NIST.SP.800-63-3

[3] Open Web Application Security Project (OWASP) (2018) *Certificate and Public Key Pinning*. Available at https://www.owasp.org/index.php/Certificate_and_Public_Key_Pinning

[4] McKay KA, Cooper DA (2019) Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) 800-52, Rev. 2. https://doi.org/10.6028/NIST.SP.800-52r2

[5] Open Connectivity Foundation (OCF) (2019) *UPnP Standards & Architecture*. Available at https://openconnectivity.org/developer/specifications/upnp-resources/upnp

[6] Rosenberg J, Schulzrinne H, Camarillo G, Johnston A, Peterson J, Sparks R, Handley M, Schooler E (2002) SIP: Session Initiation Protocol. (Internet Engineering Task Force (IETF), Network Working Group), Request for Comments (RFC) 3261, https://doi.org/10.17487/RFC3261

[7] Rivest R (1992) The MD5 Message-Digest Algorithm. (Internet Engineering Task Force (IETF), Network Working Group), Request for Comments (RFC) 1321, https://doi.org/10.17487/RFC1321

[8] Tschofenig H, Arkko J, Thaler D, McPherson D (2015) Architectural Considerations in Smart Object Networking. (Internet Engineering Task Force (IETF), Internet Architecture Board), Request for Comments (RFC) 7452. https://doi.org/10.17487/RFC7452

[9] Federal Communications Commission (FCC) (2019) *FCC ID Search*. Available at https://www.fcc.gov/oet/ea/fccid

Appendix A—Review Methodology

The review methodology included open-source research focused on reviewing publicly accessible documentation and hands-on review in a lab to observe and identify security characteristics of selected consumer home IoT devices.

A.1 Open-Source Research

Before the review began, the project team conducted open-source research on various consumer home IoT device categories and devices to determine what IoT devices would be included in the review. *Open-source research* is the use of public sources of information, such as websites, documents (e.g., user manuals, product reviews), product-user forums, and product packaging to identify characteristics of a product without acquiring, examining, or using the product itself. Because IoT devices are unique in nature, it is difficult to track down books or documents with everything there is to know about a device. Current knowledge of IoT, in particular its components and features, often depends upon researchers willing to share their findings.

For the review, the project team reused the information collected during the pre-review open-source research and conducted additional open-source research to better understand the characteristics of each IoT device to be reviewed. The types of information collected during open-source research included:

- device name, model number, and manufacturer
- target market (types of users)
- functionality provided, including smart, non-smart, and device management functions
- device specifications, such as:
    - processor types and models
    - power capacity (for battery-powered devices)
    - Federal Communications Commission (FCC) identification (ID) (see below for more information)
    - wireless protocols supported (e.g., Wi-Fi, Bluetooth, Zigbee, Z-Wave, near-field communication, proprietary)
    - communications ports exposed (e.g., USB, Ethernet, serial)
    - communication pattern per Request for Comments 7452 (device to device, device to gateway, device to cloud) [8]
- user interface specifications, such as:
    - device inputs (e.g., button, keypad, touchscreen)
    - device outputs (e.g., light-emitting diode, screen, sound, voice)
    - desktop, web, and companion mobile applications
- identities of open-source libraries used to communicate with the device
- security characteristics, such as:
    - security features (e.g., authentication mechanisms, authentication credential forms)
    - manufacturer security claims
    - vulnerabilities or weaknesses with the IoT device, ecosystem, or both
    - history of manufacturer patches and other updates for the device

One item from the list that merits additional explanation is the FCC ID. An FCC ID is a code issued to radio frequency devices certified for use in the United States. Valuable information can be gleaned from an FCC ID lookup [9]. The FCC's Office of Engineering and Technology has product exhibits online from its device certification processes. Two of the more useful types of exhibits are device test reports and photos of device internals. Device test reports provide more details on communications, such as what wireless protocols are being used. The photos show some of the components within the device, such as boards and chips.

A.2 Hands-On Review

The second part of the consumer home IoT-device security review was a hands-on review to discover or identify the functions in the device. Each hands-on review was documented by the team, including:

- date
- tools and tool versions used
- each assessor's name and actions performed
- review vantage point
- data collected
- storage location of review results

Device identifiers were also recorded if applicable, such as for network captures. The team also reviewed the complexity of installing and configuring each device (complexity information is not included in this report, for brevity purposes).

The two primary tools used during hands-on review were utilities for network packet-capture products. These tools were used to capture and decode network traffic between the IoT device and other devices during review and observation. They also calculated statistics and listed the IP addresses, ports, and protocols present in the packet captures. To perform the packet captures, various network configurations were put into place to forward traffic between a laptop's internal network interface card and an Ethernet/USB adapter.

One objective of the packet captures was to identify all communications between an IoT device and other IP addresses in its home IoT ecosystem. For example, a packet capture could identify external IP addresses that a device was contacting. Analysis of the IP addresses and their associated domain names could provide more information on the likely nature of the external host. For example, connecting to UDP port 123 on an external host with "NTP" in its domain name is probably the device using NTP to synchronize its clock with an authoritative external time source.

Another objective of the packet captures was to identify any security protocols or services in use for protecting the communications. For encrypted communications, the packet captures would indicate whether TLS was in use, what version of TLS was in use, and what cryptography suite TLS was using. For Wi-Fi communications, packet captures would indicate which Wi-Fi security protocol was in use (e.g., WEP, WPA, WPA2), if any. IoT devices supporting Bluetooth may send out Bluetooth advertisement packets, which identify the version of the Bluetooth protocol being supported.

In addition to packet captures, other tools were used for hands-on review. One tool performed port scans against IoT devices to identify open network ports.

Appendix B—Acronyms

Selected acronyms used in this report are defined below.

API	Application Programming Interface
FCC	Federal Communications Commission
FOIA	Freedom of Information Act
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
ID	identification
IETF	Internet Engineering Task Force
IoT	Internet of Things
IP	Internet Protocol
IR	Interagency or Internal Report
ITL	Information Technology Laboratory
MAC	Media Access Control
microSD	micro Secure Digital
NCCoE	National Cybersecurity Center of Excellence
NIST	National Institute of Standards and Technology
NISTIR	National Institute of Standards and Technology Internal Report
NTP	Network Time Protocol
OCF	Open Connectivity Foundation
OWASP	Open Web Application Security Project
PIN	Personal Identification Number
RFC	Request for Comments
SD	Secure Digital
SIP	Session Initiation Protocol

SP	Special Publication
SSID	Service Set Identifier
SSL	Secure Sockets Layer
TLS	Transport Layer Security
UDP	User Datagram Protocol
UPnP	Universal Plug and Play
USB	Universal Serial Bus
VPN	Virtual Private Network
WEP	Wired Equivalent Privacy
Wi-Fi	Wireless Fidelity
WPA	Wi-Fi Protected Access
WPA-TKIP	Wi-Fi Protected Access-Temporal Key Integrity Protocol