2015-08-01

Evaluating the Security of Smart Home Hubs

Steven A. Christiaens
Brigham Young University - Provo

Evaluating the Security of Smart Home Hubs

Steven A. Christiaens

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Derek L. Hansen, Chair
Dale C. Rowe
Chia-Chi Teng

School of Technology

Brigham Young University

August 2015

ABSTRACT

Evaluating the Security of Smart Home Hubs

Steven A. Christiaens
School of Technology, BYU
Master of Science

The goal of this research is to improve the security of smart home hubs by developing a standard against which hubs can be evaluated. This was done by first reviewing existing standards, guides, and collections of best practices. I determined that adapting or extending an existing standard was the best way to proceed. Potential candidates were selected, and after thorough comparison, I chose to extend the OWASP Application Security Verification Standard (ASVS).

Extensions were composed of additional security requirements to address smart home hub functionality not covered by the existing requirements of the ASVS. These additional requirements were developed based upon existing best practices and are referred to as the Smart Home Extensions. Where a best practice or guidance did not yet exist for a particular hub functionality, guidance from related fields was adapted. The entire set of Smart Home Extensions were reviewed by industry experts, updated based on feedback, and then sent on for further peer review.

Four smart home hubs – VeraLite, Wink, Connect, and SmartThings – were evaluated using the ASVS with the Smart Home Extensions. The evaluation uncovered security vulnerabilities in all four hubs, some previously disclosed by other researchers, and others new. Analysis of the evaluation data suggests that authentication is a common problem area, among others. Based on the performance of the hubs and the data collected, I suggest that the ASVS and Smart Home Extensions can be an effective tool to provide insight into the security posture of smart home hubs.

Keywords: smart home, hub, internet of things, IoT, home automation, security, OWASP ASVS, application security verification standard, smart home extensions

ACKNOWLEDGEMENTS

I would like to thank my wife, Kathryn, who has been supportive, encouraging, and understanding throughout this journey. I could not have done it without her. To my parents, Steve and Glenda, for teaching me to believe in myself, thank you. Thanks also go to my committee chair, Derek, for the valuable advice and guidance he has given. Finally, thanks to the Lord, for surrounding me with such wonderful people.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

The number of devices connected to the Internet is growing quickly. Cisco Systems estimates that close to 100 things are being added to the Internet every second (Tillman 2013). The increasing popularity of smart home devices is a big contributor to this growth. Whether someone wants to control their lighting system from the couch, check on the oven while working in the garden, or lock up for the night from the comfort of their bed, there is a smart home device that can help them do it. With the increasing ubiquity of smartphones and tablets, finding the remote control is as easy as opening an app.

Many smart home devices offer their own app, but this leads to problems as the number of smart devices continues to grow. A consumer with three brands of connected lightbulbs, two smart locks and a network-connected baby monitor may find they need five or six different apps on their phone to control everything. No one wants to have to open a different app for each brand of light bulb they use, not to mention keeping track of a different password for each device. To address these and numerous other issues, manufacturers have introduced smart home hubs.

Smart home hubs today are similar in form and function to a consumer-grade wireless router. They are generally small form-factor devices which house one or more wireless antennas and often support multiple communications protocols including Wi-Fi, Z-Wave, ZigBee, Insteon, Bluetooth and others (Lodamo and Forsström 2012). They are referred to as hubs because they

act as the central point of contact for a network of smart devices that may include door locks, window and door sensors, lighting controls, thermostats, and a range of other devices.

Most smart home hubs can be thought of as liaisons or butlers. Once a consumer has installed the hub, they are prompted to install an app on their mobile device. From then on all communication with the hub and with any associated smart devices is handled through the mobile app. Any command, such as switching the lights on or off, is issued through the app that then communicates with the hub. The hub takes care of issuing the correct commands to each connected light, even if the lights are different brands and use different protocols. For consumers, this means not having to worry about learning to use disparate apps or remembering multiple passwords. For those with smart locks, it means no more being caught without house keys as long as they have their mobile device. With this convenience, however, comes risk: a malicious party taking control of a hub could control the entire house.

How can we know if these hubs that are so critical a component are secure? Many consumers are oblivious to the security or insecurity of their devices and believe that manufacturers or government regulations will protect them. Unfortunately, many manufacturers entering the market have little to no experience with network or software security and government regulation moves slowly. The Federal Trade Commission recently acknowledged that the potential security risks to consumers includes "unauthorized access and misuse of personal information" and even "risks to personal safety" (FTC Staff Report 2015), but regulation is still many years away. What is needed is a way to systematically assess the security of these hubs.

The purpose of this research is to improve the security of smart home hubs. To do this, I will be addressing the following research objectives:

- Apply and extend an existing application security standard so that it can be used to evaluate the security of smart home hubs.
- Determine what types of security vulnerabilities are most prevalent in existing smart home hubs.
- Develop recommendations for more secure smart home hubs that take into consideration users and manufacturers.

The remainder of this thesis is organized into the following sections:

- Chapter 2: Literature Review
    - An overview of the smart home environment from the perspective of current and past research, focusing on security. A look at existing security frameworks and their applicability to smart homes and the Internet of Things (IoT). This chapter also includes a discussion of where this research fits into the overall discussion.
- Chapter 3: Methodology
    - A detailed description of research questions and objectives along with the methods employed to answer them.
- Chapter 4: Smart Home Extensions for the OWASP ASVS
    - A summary of why the OWASP Application Security Verification Standard (ASVS) was chosen, how extensions were developed, and what was changed over time due to feedback and testing. This chapter also includes the proposed extensions.
- Chapter 5: A Security Evaluation of Four Smart Home Hubs

    - Results of using the ASVS and Smart Home Extensions to test four market-leading smart home hubs. Includes descriptions of where the devices performed well and where they failed to implement security best practices.
- Chapter 6: Recommendations for Smart Home Hub Security
    - Recommendations for improving the security of future smart home hubs, organized by stakeholder. The stakeholders included are device users and device manufacturers.
- Chapter 7: Discussion and Future Work
    - Summary and discussion of findings and recommendations as well as intended applications and limitations of the smart home extensions. Includes a brief discussion of possible future work.
- Appendix A: Supplementary Materials

2 LITERATURE REVIEW

Commercial and academic interest in the smart home arena has picked up in recent years as more and more consumer devices include some sort of network connectivity. Conferences such as Smart Home World, TV Connect, ICOST, Connections, and numerous others organized in just the last ten years continue to be successful. One of the largest showcase venues for commercial devices, the Consumer Electronics Show, expanded in 2015 to include exhibit space specifically for smart home devices (Brown 2014). Some small colleges have even begun to offer certifications in smart home technology, while Duke University engineering students can become "Smart Home Research Fellows" (Martinsburg College 2015; Nicolet College 2015; Pratt School of Engineering 2015).

Along with the increase of popular interest, significant research has gone into defining and expanding the field. Many researchers have studied the problem of smart home network and system designs (Han and Lim 2010; Hussein et al. 2014; Davidoff, Lee, and Yiu 2006), while others have characterized the various wired and wireless standards (Lodamo and Forsström 2012; Gomez and Paradells 2010). Due to the large amount of data generated by these devices, there has also been research into how to process and organize that data (Cook et al. 2003; Zhang, Leung, and Chan 2008).

2.1 **Security**

Policymakers, researchers, and industry groups alike recognize security in the smart home arena as a critical issue. Addressing risks introduced by the Internet of things, of which the smart home is a major part, the Federal Trade Commission (FTC) asserted that it presents "a variety of potential security risks". More specifically, the Commission recognized that these devices could be exploited "to harm consumers by: (1) enabling unauthorized access and misuse of personal information; (2) facilitating attacks on other systems; and (3) creating risks to personal safety." (FTC Staff Report 2015).

Academic and private sector researchers have focused mainly on privacy (Möllers and Seitz 2014; Weber 2010; Jakkula and Cook 2008), though all recognize the link between privacy and security. Numerous privacy issues exist. Weber describes the difficulties of keeping private information truly private and of preventing a smart object from leaking information about its owner (Weber 2010). Möllers and Seitz were able to extrapolate and predict user actions in a home throughout the day based solely on passive wireless packet captures. In one case, they were able to determine the occupant's weekly schedule and accurately predict whether the home was occupied or vacant, making it easy to plan a burglary. All of this was discovered without any prior knowledge of the type or number of sensors in the home (Möllers and Seitz 2014). Similarly, using machine learning and only the data available from smart sensors, Jakkula and Cook developed techniques to automatically build a model of expected behavior for individuals and detect anomalies (Jakkula and Cook 2008). By knowing exactly what an individual was doing inside their home, they hoped to be able to alert emergency responders of any problems. The data generated by this level of surveillance needs to be properly secured, as improper access to it could have significant privacy implications.

In the commercial sector, security firms are beginning to perform penetration testing on smart home devices to identify weaknesses in their design. One firm, Xipiter, published a walkthrough of how they performed hardware and software exploitation against various smart home devices. For one smart home hub, they obtained root shell access, exposed password hashes through a directory traversal vulnerability, and showed that all of these hubs share the same SSH private keys (Xipiter 2014). Another team at Hewlett-Packard looked at 10 different smart home devices and found that "70 percent did not encrypt communications to the Internet and local network" (Smith and Miessler 2014). There is clearly ample room for improvement.

Despite the growing body of research related to smart home security, no material was found that focuses on systematically evaluating the security of hubs. As discussed previously, researchers have performed penetration testing of hubs, but typically it has only been against a single hub (Crowley, Savage, and Bryan 2013; Xipiter 2014). Other researchers have attempted to improve the overall security of these devices by focusing on usability (Kalofonos and Shakhshir 2007). By using metaphors, intuitive design, and a collection of middleware, Kalofonos and Shakhshir hoped to encourage the use of security for non-experts and prevent misconfiguration of security controls. While this could potentially improve overall security, it still does not address the problem of devices or services that are inherently insecure or contain security-related design flaws. Given the increasingly critical role of smart home hubs, a comprehensive approach to evaluating these devices would be valuable.

2.2 Evaluation Standards, Guides, and Frameworks

The goal of this research is to develop a systematic way to evaluate the security of smart home hubs. This has been done in related domains, such as web applications and wireless networking (Meucci and Muller 2014; Council on Cyber Security 2014), however guidance in the

smart home arena is nonexistent. This is likely due to lack of maturity in the field, something noted by FTC Commissioner Joshua D. Wright when he described the "Internet of Things" as "a nascent concept about which the only apparent consensus is that predicting its technological evolution and ultimate impact upon consumers is difficult" (Wright 2015).

When discussing methods to evaluate the security of a device, application, or service, it is helpful to briefly define a few terms. Although there is often overlap in the content of these items, the level of detail and purpose for which they were designed are what set them apart from each other.

Framework: Although many definitions exist, a framework is simply a structured way to think about a topic. In the information technology field, frameworks are often quite detailed and describe processes – ways that an application or system should work – without defining how to implement those processes. Some notable frameworks in information security are Control Objectives for Information and Related Technology, or COBIT (ISACA 2012), the NIST Framework for Improving Critical Infrastructure Cybersecurity (National Institute of Standards and Technology 2014), and the Sender Policy Framework for e-mail (SPF Council 2004).

Hardening Guides: These focus on system hardening, which Berkeley Security describes as "the process of securely configuring computer systems, to eliminate as many security risks as possible". Hardening guides are commonly written by the software publisher or device manufacturer and are aimed at system administrators, the ones most likely to perform the configuration changes. The purpose of most hardening guides is to increase security via built-in configuration settings. Some examples are Apple's OS X Security Configuration Guides (Apple Inc. 2015), VMware's Security Hardening Guides (VMware 2015), and Microsoft's Threats and Countermeasures Guide (Andersen et al. 2011).

Standard: With regard to technical systems, a standard is "an established norm or requirement […] It is usually a formal document that establishes uniform engineering or technical criteria, methods, processes and practices." ("Technical Standard" 2015) If something is going to receive a certification, it is usually compared to or measured against a standard. Notable standards in information security include the ISO 27000 series and the Payment Card Industry Data Security Standard, also known as PCI-DSS or simply PCI (International Organization for Standardization 2015; PCI Security Standards Council 2013).

Testing Guide: A testing guide defines an approach to testing, and may accompany a framework, standard, or hardening guide. While a standard or framework may say "Ensure that the application is not susceptible to SQL injection", the testing guide will say "Test for SQL injection using the following tools and techniques..." and may include specific criteria for what constitutes a success or failure when testing. The most well-known testing guide in the information security arena is probably the OWASP Testing Guide (Meucci and Muller 2014).

As stated earlier, there is often significant overlap in the content of these documents. What one group refers to as a framework may be used as a standard, and some testing guides are more akin to a framework or a hardening guide. Furthermore, other respected resources covering the topic are not referred to by any of the above terms, most notably the Critical Security Controls for Effective Cyber Defense (Council on Cyber Security 2014). When reviewing these documents and the methods contained therein for applicability to a particular topic, the important thing is the information they contain, not their titles.

2.3 Candidate Frameworks

In the arena of smart homes, home automation, and the "Internet of Things", there currently exist no framework or testing guide for evaluating the security of devices or hubs. Best practice

recommendations exist in the form of an FTC staff report, indicating that security is clearly of interest to the FTC (FTC Staff Report 2015). However, the report mentions no methods for performing a security evaluation on an Internet of Things device. Perhaps more concerning, one FTC commissioner dissented from the decision to publish these best practices, noting that the Commission did not "…actually engage in a rigorous cost-benefit analysis prior to disseminating best practices…" (Wright 2015).

Although guidance specific to the Internet of Things does not appear to exist, much less specifically to home automation hubs, there are testing guides, standards, and other projects in related fields that are applicable. For my purposes, I selected six projects as potential candidates for adapting or extending to cover home automation hubs:

2.3.1 The Critical Security Controls for Effective Cyber Defense

The Critical Security Controls for Effective Cyber Defense, also referred to as the Critical Security Controls (CSC), was put together by industry experts in 2008 with the aim to be "the most effective and specific set of technical measures available to detect, prevent, and mitigate damage from the most common and damaging … attacks" (Council on Cyber Security 2014). It is a collection of 20 technical measures along with supporting documentation on how to implement, test, and automate those measures. It has elements of a framework and a generalized hardening guide.

The stated goals of the project are to "protect critical assets, infrastructure, and information by strengthening [an] organization's defensive posture through continuous, automated protection and monitoring of […] sensitive information technology infrastructure to reduce compromises, minimize the need for recovery efforts, and lower associated costs" (SANS Institute 2015). Based

on the longevity of the project and the number of mappings to different regulatory requirements and standards, one may surmise that the project has been successful.

An advantage of the Critical Security Controls is that the implementation guide includes multiple levels of complexity, starting with implementing the control at a very basic level to completely automating the control and associated monitoring. This means that the controls can be beneficial to a wide range of users, from home users and small businesses looking to increase security, to large enterprises with mature security practices and policies.

As mentioned previously, the aim of the Critical Security Controls is to protect networks, devices, and organizations from the most common cyber-attacks. Because smart home hubs share many similarities with existing network devices, such as web servers and wireless access points, the Controls seem a good fit for evaluating the security of hubs. They are, however, a poor fit for this task.

The Controls are not sufficiently specific when it comes to smart home hubs. Although the structure of the Controls is such that they can be useful to a broad range of users, many of the recommendations would only be applicable to medium and large networks and organizations. CSC 3, for example, suggests developing secure configurations for hardware and software on mobile devices, laptops, workstations and servers. It does not offer guidance on how to come up with a secure configuration. A hardening guide could fill that knowledge-gap; however no hardening guides exist for any smart home hubs currently on the market. CSC 3 and other controls would have to be removed or significantly changed to apply to smart home hubs.

Another example of the lack of specificity is the guidance on wireless communication. While the Controls include a section on wireless access control, they consider only WiFi and Bluetooth. The recommendation on Bluetooth is to "Disable wireless peripheral access of devices

(such as Bluetooth), unless such access is required for a documented business need." For smart home hubs, there is often a "business need" for Bluetooth. This introduces another possible attack vector, which the Controls do not adequately address. Furthermore, hubs often use four or more different wireless protocols, many of which are not addressed by the Critical Security Controls.

2.3.2 OWASP Application Security Verification Standard

The Open Web Application Security Project (OWASP) is an online community that started in 2001 with a focus on improving the security of software. The major deliverables of OWASP are its projects, and the Application Security Verification Standard (ASVS) is one of their flagship projects. The focus of the ASVS is "to normalize the range in the coverage and level of rigor available in the market when it comes to performing Web application security verification using a commercially-workable open standard" (OWASP 2014a). Originally released in 2009, the ASVS has received numerous updates, with version 2.0 in August of 2014 and version 3.0 expected in November of 2015.

The ASVS contains verification requirements grouped together by topic. Each verification requirement addresses a single aspect of the overall topic. Under the topic of authentication, for example, one requirement is "Verify all password fields do not echo the user's password when it is entered". Another reads "Verify all pages and resources require authentication except those specifically intended to be public". In addition to authentication, topics include session management, malicious input handling, data protection, communications security, and more. All of these topics and requirements focus on functionality that may be offered by a web application.

Similar to how the Critical Security Controls contain controls of varying complexity, the ASVS is divided into four levels of verification, numbered zero to three. Level zero is intentionally undefined by the standard, while levels one through three are meant to protect an application

against opportunistic, standard, and advanced attackers, respectively. This is useful because not every application has the same business requirements. An online banking application containing sensitive information and functionality may call for scrutiny, while an application made purely for marketing purposes needs only a cursory review.

The limitation of the ASVS when it comes to smart home hubs is that it focuses almost exclusively on web application security. While web applications provide a significant portion of the functionalities in existing hubs, these hubs also include components that are usually handled by an operating system, such as network interfaces and wireless antennas. On its own, the ASVS is not sufficiently comprehensive to evaluate the security of a smart home hub.

2.3.3 CWE/SANS Top 25 Most Dangerous Software Errors

The CWE/SANS Top 25 Most Dangerous Software Errors is a list compiled by SANS Institute and MITRE, two organizations that are well-known in the field of information security for their research and educational materials. The Top 25 is designed to promote education and awareness about "the most widespread and critical errors that can lead to serious vulnerabilities in software" (SANS 2011). SANS and MITRE released the list in 2010 and updated it in 2011, but it is no longer maintained as other projects have taken precedence (MITRE 2014). Because the factors contributing to vulnerable software change little over time, I included the Top 25 as a candidate for adaptation to smart home hubs.

The Common Weakness Enumeration (CWE), from which the Top 25 is taken, consists of a list of common weaknesses in software. These weaknesses are described as flaws, faults, bugs, vulnerabilities, and other errors in the implementation, code, design, or architecture of a piece of software. Some examples are authentication errors, code evaluation and injection, and buffer

overflows. Each list item includes information about the common consequences of these errors as well as examples and a brief discussion about technical impact.

The purposes of the CWE as described by MITRE are as follows: to serve as a common language for describing software security weaknesses; to serve as a measuring stick for software security tools targeting these weaknesses; and to provide a common baseline standard for weakness identification, mitigation, and prevention efforts (2014). This means that the list as a whole was intended to be comprehensive, and that is reflected in the over 1,000 entries it contains (MITRE and SANS Institute 2010). The 25 most common of these weaknesses and errors comprise the Top 25.

As previously mentioned, the Top 25 is meant to promote awareness about the most widespread and critical errors that can lead to serious vulnerabilities in software. Software controls everything in a smart home hub, so the Top 25 should be directly applicable. In this case, however, the widely applicable nature of the Top 25 is a weakness as well as a strength. As stated by MITRE, the list "…contain[s] a mix of weaknesses with some only applicable to specific applications or technologies." (2014). While the CWE was designed to be used as a measuring stick, it was to measure the coverage of security scanning tools, not to measure the security of generic applications. Using the Top 25 to evaluate the security of smart home hubs would require significant changes, not unlike creating an entirely new framework. Existing weaknesses not applicable to smart home hubs would need to be removed and missing weaknesses would have to be added. Such changes would likely impact the effectiveness of the document as a tool to promote awareness. If changes were to be made, other candidates, such as the Application Security Verification Framework and the OWASP Testing Guide, would be better choices given their design as tools for testing web applications.

2.3.4 OWASP Testing Guide

The OWASP Testing Guide is described as a penetration testing framework which users can implement in their own organizations and a penetration testing guide that describes techniques for testing the most common web application and web service security issues (OWASP 2015). The Testing Guide claims to capture the consensus of leading experts on how to perform security testing and is put together completely by volunteers. The first version of the Testing Guide was released in 2003 and the most recent, version 4.0, was released in September 2014.

Similar to the ASVS, the Testing Guide is one of OWASP's flagship projects, which means that it has shown strategic value to OWASP and application security as a whole (OWASP 2015). The guide itself consists of two major parts. The first part contains a framework for penetration testing that includes a description of the Software Development Life Cycle, a discussion about where security testing fits in the life cycle, and why different types of security testing are important. The second part contains a listing of typical vulnerabilities found in web applications along with recommendations for how to test for these vulnerabilities. This listing is grouped by vulnerability type and progresses logically such that a tester could follow it and go from basic to advanced testing of an application.

This ease-of-use is one of the primary strengths of the Testing Guide. A tester with the requisite skill would have little trouble following the guide and, once all tests were complete, could be reasonably confident that they had thoroughly tested the application and knew its weaknesses. The other strength of the guide is it's thorough descriptions of how to test for various weaknesses. Using the guide, a user that is unfamiliar with how to test for a particular vulnerability can quickly learn the basics of the vulnerability as well as a few different methods to test for it.

Similar to the ASVS, the Testing Guide is potentially a great fit for testing the security of smart home hubs because web applications provide much of the functionality of a hub. However, it also shares that same weakness in that it focuses exclusively on web applications. Where it differs a bit more from the ASVS is in the design and purpose of its tests. The tests contained in the guide are designed to detect whether vulnerabilities are present. In practice, this means that the guide is great at helping a tester ascertain that something is done incorrectly, but provides little guidance on how to tell if something is done correctly, or at least according to best practices. For the purposes of the Testing Guide, this is perfectly acceptable; for evaluating the overall security of a smart home hub, this is a significant limitation.

2.3.5 Other Potential Candidates

The final two potential candidates came from the OWASP Top Ten Project and the OWASP Internet of Things Top Ten Project. While the projects have many similarities, the deliverables of the two projects differ in important ways. Neither, however, is a good match for evaluating smart home hubs. We will briefly examine each in turn, starting with the Top Ten Project.

The OWASP Top Ten Project started in 2004 and participants update its deliverables every three years. The output of the Top Ten Project is a list of the top ten most critical security flaws found in web applications based on a broad consensus of industry experts (OWASP 2013). The purpose of this document is to raise awareness about security and it is one of the most recognized resources in the industry. The Top Ten Project provides information about what causes each flaw, how to test for them, and how to prevent them.

With the focus of the Top Ten list being education and awareness, it is a poor fit for evaluating smart home hubs. While the list can be applied to smart home hubs due to its focus on

web applications, it was never meant to be a comprehensive listing of possible flaws. The list authors even say that "Adopting the OWASP Top Ten is perhaps the most effective *first step* towards changing the software development culture within [an] organization" [emphasis added] (OWASP 2013). It is clearly a poor choice for adaptation.

The Internet of Things Top Ten Project is a more recent development, with their first deliverable produced in 2014. That deliverable is a list of the 10 most significant security surface areas presented by IoT systems. Also known as the attack surface of a software environment, the security surface area is the subset of the system's resources that an attacker can use to attack the system (Manadhata 2008). In addition to defining these security surface areas, the list also provides information on threat agents, attack vectors, vulnerabilities, and associated impacts (OWASP 2014b).

The IoT Top Ten stands out from the other candidates mentioned because it focuses specifically on Internet of Things devices, as opposed to web applications or software in general. However, it is meant as a tool for education and guidance, and as such is ill-suited to serving as a tool to evaluate or rate the security of existing products. Furthermore, it was purposely created as a top 10 list to make it easy to understand and present to others. Significantly altering the structure of the existing deliverable, by creating a top 14 or top 16, would likely serve only to confuse users.

While both the Top 10 and IoT Top 10 are respected in the industry, they share many of the same limitations exhibited by previous candidates. This was also the case for other candidates not listed here. Many of the existing frameworks and guides are relevant to IoT and therefore smart home hubs, but none is sufficiently specific or comprehensive in that arena. It is for this reason that I propose to extend an existing framework. Although there is significant overlap among the candidates described herein, for my purposes, the OWASP ASVS is the best candidate given

its format and the breadth of information it already contains. The other candidates will inform the extensions to the ASVS.

3 METHODOLOGY

This thesis focuses on two research objectives and one research question:

Research Objective 1 (RO-1): Develop and test extended requirements to the ASVS for evaluating the security of smart home hubs.

Research Question 1 (RQ-1): What types of security vulnerabilities are most prevalent in existing smart home hubs?

Research Objective 2 (RO-2): Develop recommendations for a more secure smart home hub that take into consideration both users and hub manufacturers.

The methods used to reach these objectives and address the research question are outlined below, in order.

3.1 RO-1: Development and Testing of Extended Requirements

The original purpose of this research was to develop from scratch a framework to evaluate the security of smart home hubs. This framework was to be based, in part, on existing best practices in related domains. However, after reviewing existing frameworks and testing guides for those related domains, it became clear that extending an existing framework would be more feasible and useful. More feasible due to the complexity of the domains involved, including web application security, wireless security, cryptography, and others. More useful because an existing framework would already have a community of users who could comment on the extensions and

provide feedback, and the focus could be on the novel extensions, instead of reinventing yet another framework from scratch, which would unnecessarily compete with existing frameworks.

3.1.1 Compiling Best Practices and Choosing a Framework to Extend

To gain a better understanding of existing best practices, I reviewed frameworks and testing guides in the related domains of web application security and enterprise security and considered how recommended security controls would apply in the context of a smart home hub. Each of the documents detailed in chapter two received a thorough review with an eye toward smart home hubs. Those candidates were chosen because they are well known, contain enough detail to be easily accessible, and the bulk of their content is applicable to smart home hubs. Other potential candidates were discarded because they did not qualify in one or more of these areas. Some of those include the Common Criteria, the ISO 27000 series, COBIT, and ITIL (CCRA 2012; Survey 2006; ISACA 2012; Axelos 2015).

After reviewing the documents, I chose the ASVS to serve as the base that I would extend to include smart home hubs. It was chosen for three primary reasons: the level of detail in its verification requirements, the range of its existing coverage, and its intended purpose. While other candidates include tests or steps to verify that a particular class of vulnerabilities is not present on a target, the verification requirements in the ASVS strike a balance of providing enough detail to communicate what needs to be tested while not going into pages of explanation and examples. This balance is not perfect, as one user will have more or less experience than another and therefore may require more or less information, but the balance chosen by the authors results in a concise document. This also makes for a document that is easy to understand after very little time with it.

The range of security issues covered by existing verification requirements in the ASVS was another strength when compared to most other candidates. As discussed in chapter two,

verification requirements are grouped by topic. The ASVS contains 13 categories and 190 requirements, with an average of 14 requirements for each category. Based on numbers alone, these requirements easily surpass the OWASP Top 10 lists as well as the CWE Top 25. Not having a limit on the number of requirements allows the ASVS to go into more detail. And while the Critical Security Controls covers a similar number of issues, it is focused primarily on enterprise security and network defense. This translates to less overlap with issues that might affect a smart home hub. Finally, the OWASP Testing Guide covers a very similar range of issues, but the level of detail and intended purpose of the ASVS was a better fit for the goals of this work.

As previously mentioned, these documents contain significant overlap in the issues they address, but their purpose is what sets them apart. The ASVS stands out because the authors purposely designed it to be a standard for performing web application security verification (OWASP 2014a). Although the Critical Security Controls and the OWASP Top 10 may be considered de-facto standards, they were not designed with that purpose in mind. Adapting them or otherwise attempting to turn them into an actual standard would require significant effort. Taking such a step does not make sense when there is already an existing standard that covers a similar range of issues. As for the Top 10 and Top 25 lists, they were meant as tools for education and awareness. Turning those into standards would require even more effort. Comparing candidates based on their intended purpose, the ASVS was the most sensible choice.

3.1.2 Iteratively Developing the Smart Home Extensions

Once I had selected the ASVS to act as the base framework, I compared it with other candidates to discover best practices that were obviously missing. Given that the ASVS is designed to evaluate web applications, certain issues, such as wireless communication security, were not covered and would need to be added. Having previously made a list of best practices and

other security issues that could apply to smart home hubs when reviewing potential candidates, this process was straightforward: any item on that list that was not addressed by the existing requirements in the ASVS was added to a list of proposed requirements.

The next step was to gain experience with some smart home hubs to understand the functionality offered and discover potential attack vectors not addressed by the requirements of the ASVS. I selected six hubs as described in section 3.1.3. Each was set up according to the manufacturer's instructions and two smart devices, a Jasco Light and Appliance Control Module and a Schlage Wireless Door and Window Sensor (Jasco Products Company 2015; Amazon Inc. 2014), were connected to each. After setting up the systems for traffic capture, I used as many of the functions of each hub as possible. This included turning individual devices on and off, defining rulesets, assigning groups, adding and removing users, backing up and restoring the hubs, and more. In this manner, I generated a list of functions and capabilities of the different devices and considered how those could be subverted or abused by a malicious party. Controls to prevent such abuse were then added to the list of proposed requirements. Where possible, existing best practice and guidance documents were referenced when defining new controls.

Once I had compiled the list of proposed requirements, I then reviewed and categorized each item so that the structure of the new requirements matched those contained in the ASVS. This spreadsheet of proposed requirements and references was the first draft, which was sent to industry experts for peer review. Members of the ASVS mailing list performed the first review. I posted the first draft to the mailing list accompanied by the following questions:

- Are there any testing categories missing?
- Are there any security requirements missing?
- Are the requirements not specific enough or too specific?

- Would you find these additions to the ASVS useful?
- Are any of these requirements covered by existing ASVS requirements?

I received responses from four subscribers (Christian Heinrich, Jerome Athias, Andrew van der Stock, Aaron Guzman) and added or modified requirements based on their feedback. Next, I contacted a series of individuals via email and asked them to review the updated extensions. As part of this process, I performed a semi-structured interview with each them consisting of the questions listed above. After getting feedback from an individual, I updated the requirements and then sent them on to the next reviewer. In this manner, all modifications or additions to the requirements were reviewed by me, the person suggesting the change, and at a minimum the next reviewer. This process of iteratively incorporating feedback and reviewing changes was repeated until very little new feedback was given. At that point, nine experts in the field of information technology including Andrew van der Stock, Christian Heinrich, Jerome Athias, Rob Ragan, Aaron Guzman, Ryan Speers, Derek Hansen, Dale Rowe, and Daniel Cuthbert had reviewed the extensions. Initial development of the extensions was declared complete.

3.1.3 Selection of Target Hubs for Testing

With initial development of the extensions completed, the next step was to test them in a real-world scenario. I chose four smart home hubs as targets for evaluation. Selection of these hubs was based on perceived popularity and support for wireless protocols, such as Z-Wave and ZigBee. By searching review sites, I was able to generate a list of 13 popular hubs, each of which was discussed by at least two well-known sites. A list of the potential target hubs can be seen in the figure below:

Table 1 - Protocols Supported by Popular Hubs

	Protocols supported									
Hub Name	WiFi	Z-Wave	ZigBee	Bluetooth	Lutron ClearConnect	IrDA	WeMo	INSTEON	Kidde RF	Unspecified RF
Wink HUB	X	X	X	X	X				X	
SmartThings	X	X	X				X			
Lowes Iris	X	X	X							
Vera	X	X								
Staples Connect	X	X								
Piper	X	X								
Logitech Harmony Home Hub (* = with Extender)	X	X*	X*			X				
Nexia Bridge		X								
Control4			X							
Lutron Wireless Smart Bridge					X					
WeMo (App based)							X			
Insteon Hub								X		
iSmart Alarm										X

Once I had determined the 13 most popular hubs, I collected specifications for each of them to determine what wireless protocols were supported. After collecting this information, it became clear that the Z-Wave and ZigBee protocols are the most widely used among existing devices. Although WiFi is also supported by most hubs, this is generally to allow communication with smart phone applications, rather than provide control of smart home devices. The devices with support for the most protocols were selected as testing targets. These devices were: Wink HUB, SmartThings, Lowes Iris, Staples Connect, Vera, and Piper. The Logitech Harmony Home Hub was not selected because the Extender had not been released at the time. Eventually, the Lowes Iris and the Piper hubs were removed from the pool of testing targets due to incompatibility with the Z-Wave- and ZigBee-enabled accessories used.

3.1.4 Setup of the Testing Environments

Setup of the testing environment required significant time and more troubleshooting than anticipated. Some hubs offered the option of connecting to the network using a wired or wireless connection, while others only supported one or the other, so the testing environment required both. Even with complete control of the network, it was not possible to observe all of the traffic

generated by a hub. Traffic between the web browser or mobile application and a hub was captured using a proxy. Some of this traffic was encrypted using TLS. Using a proxy, this was easily bypassed in the web browser. However, the SmartThings and Wink hubs only offered mobile applications and both performed certificate pinning. Decrypting traffic from those applications required software that hooked system API calls to bypass certificate pinning checks (NCC Group 2014). All of the hubs also communicated with manufacturer and other third-party servers. The content of this second communication path over the public Internet could not be decrypted in every case due to the use of TLS and certificate pinning on the hubs themselves.

Along with the difficulties presented by the hubs themselves, accessories essential to the test provided their own set of challenges. Smart sensors and bulbs refused to pair with the hubs, or reported failure when they had successfully paired. A lack of thorough documentation for either hubs or accessories made it difficult to unpair devices once they had been paired. Finally, the only readily-available devices for sniffing Z-Wave and ZigBee traffic required soldering and installation of custom firmware before they could be used. A solution utilizing software-defined radio was suggested, however no one with the proper expertise was available. With so many difficulties, it is easy to see why security research in this arena is sparse.

3.1.5 Testing the Smart Home Extensions Against Hubs

Testing began after finalizing the pool of testing targets and setting up the test environment. I performed all testing activities and tracked the results using the spreadsheet contained in the appendix. This spreadsheet was based on one created by Florent Batard, a French security researcher that uses the ASVS regularly (Batard 2014). I evaluated each hub using the existing level one requirements of the ASVS and all of the smart home extensions. If any hubs passed all of the level one requirements, I would move on to level two requirements. Each hub took an

average of 10 days to complete the verification testing; resulting in an average of 60 hours spent evaluating each of the four hubs.

3.2 RQ-1: Determining the Most Prevalent Security Vulnerabilities in Existing Hubs

Once testing was complete, results were compiled to determine what types of vulnerabilities were most prevalent in the hubs tested. This list was compared to the OWASP IoT Top 10 List to determine whether the results fit with prevailing ideas on the subject.

3.3 RO-2: Developing Recommendations

Using the knowledge gained from RQ-1 as a guide, I developed recommendations for a more secure smart home hub. Separate recommendations are mode for end-users and manufacturers. Existing best practices and applicable standards in this area were taken into account, as well as the performance of the hubs under evaluation.

4 SMART HOME EXTENSIONS FOR THE OWASP ASVS

The following are proposed extensions to the OWASP Application Security Verification Standard (ASVS). They were developed as extensions to ASVS 2014, also referred to as ASVS 2.0. As discussed in the Methodology section, the requirements that make up these extensions are based on best practices from related frameworks, standards, and guidance documents. I incorporated feedback from nine industry experts in the final draft. The following sections explain how the extensions changed from the initial draft to the final draft.

4.1 Initial Draft

As detailed in section 3.1.2, the initial draft was based on my personal experience with six different smart home hubs and an in-depth review of existing best practices from frameworks, standards, and guidance documents. References to these documents were included where applicable. Also included were personal notes on why a requirement was considered important or questions soliciting feedback on how to clearly phrase an item. As can be seen in the below figure, the first draft contained three new categories and eight new requirements.

Table 2 - Initial Draft of the Smart Home Extensions

Category	Verification Requirement	Level 1	Level 2	Level 3	References	Notes
Wireless	Verify that the device supports the WPA2 standard with AES encryption. If the device acts as a wireless access point, verify that WPS is disabled or implements protections against brute-force attacks.	x	x	x	Practical Verification of WPA-TKIP Vulnerabilities (VanHoef 2013)	This could be part of Communications. WPA2 w/TKIP can be broken, as can WEP/WPA.
	Verify that no services are listening on undocumented ports.		x	x	Critical Security Control 11: Limitation and Control of Network Ports, Protocols, and Services	Phrasing? This should cover I2C, JTAG or other protocols going over wireless.
	Verify that Z-Wave uses secure node authentication.			x	Still looking for a Z-Wave security "Best Practices" document. Does one exist outside of Sigma?	This is probably a "Plus" level, as the authentication routine is defined by Sigma and/or libraries are commonly used.
	Verify that the device uses Bluetooth v2.1+ Security Mode 4 with 3 as a fallback. Devices using Bluetooth Smart should use Security Mode 1 Level 3. Verify the device is undiscoverable except as needed for pairing.			x	NIST Guide to BT Security - Security Checklist items 13, 14, 16	This may not be documented and could require sniffing the traffic.
Data Backup and Restore	Verify that a user can backup and retain a copy of the device	x	x	x	Critical Security Control 8: Data Recovery	
	Verify that a user can restore a previously backed up configuration.	x	x	x	Critical Security Control 8: Data Recovery	
	Verify that a user can reset the device to factory defaults, either via a hardware button or through software.	x	x	x	Inferred from Critical Security Control 8: Data Recovery Capability	If a device gets infected or put into a unresolvable state, this may be the only way to fix it.
Updating and Patching	Verify that there is a secure method to update or patch the system. This may be in the form of user-initiated update functionality protected by authentication, or automatic updates over a secure channel such as HTTPS.	x	x	x	Critical Security Control 4: Continuous Vulnerability Assessment and Remediation	

4.2 Revisions and Final Draft

The three verification categories of Wireless, Data Backup and Restore, and Updating and Patching changed little through each round of revisions, undergoing only minor name changes. The Wireless category was renamed to Communication Channels to reflect that wireless standards are not the only way that hubs communicate. Reviewers suggested two additional categories, physical security and privacy. Although physical security was considered as a category in the early development phases, it was decided against for one main reason: an attack scenario against a smart home hub in which the attacker has physical access to the hub implies that they are already in the house and can therefore access sensitive data in other ways. The suggestion to add a category for privacy was compelling, but many of the existing requirements across various categories of the ASVS touch on privacy issues already. The addition of a separate category that contained overlapping requirements would unnecessarily complicate the extensions.

Revisions due to feedback from reviewers resulted in 15 separate requirements in the final draft. For ease of referring to the requirements, I added numbering. In the Communication Channels category, I received conflicting feedback: some reviewers said the requirements were not specific enough because they failed to include certain standards, while others said they were too specific. After some discussion with reviewers, I chose to generalize the requirements. For example, instead of multiple requirements addressing encryption strength separately for WiFi, Bluetooth, Z-Wave and others, there is a single requirement, CC.1, which specifies that the device utilize the currently accepted strongest encryption available for each communication channel. While this puts more responsibility on the individual or organization performing the testing, it makes for a set of extensions that can be applied to new technologies without the need to constantly update the list of requirements.

In both the Data Backup/Restore and Updating/Patching categories, requirements were modified so that they better match the risk-based nature of the ASVS. The first requirement in Data Backup/Restore, for example, originally had the tester "Verify that a user can backup and retain a copy of the device configuration". This was a level one requirement, and the only requirement that touched on backups. After feedback, there are now related requirements for level two and level three testing in the form of DB.2 and DB.3, respectively.

Finally, the terminology used in many requirements was updated for accuracy based on feedback. "Hashed checksums" became "cryptographic checksums" in UP.2. In CC.6, language was added to the requirement to provide context for testers unfamiliar with downgrade attacks. CC.5 was previously included as part of CC.4, until one reviewer noted that these were unrelated requirements and should be split apart.

Although the feedback and revisions described here do not constitute a comprehensive listing of the changes over time, I have included the most significant ones. Further notes can be found under the column titled "Change notes" on the final draft, shown in figures 1 and 2.

Category	Requirement Number	Verification Requirement	Level 1	Level 2	Level 3	References
Communication Channels	CC.1	Verify each communications channel used by the device implements the currently accepted strongest encryption available. For example: Wi-Fi should be secured using WPA2 and AES encryption, Z-Wave should use secure node authentication, Bluetooth 2.1+ should use security mode 4 with 3 as a fallback, Bluetooth Smart should use security mode 1 level 3, and ZigBee should use the encryption security service.		X	X	NIST Guide to BT Security - Security Checklist items 13, 14, 16 / Recommended Practices Guide: Securing ZigBee Wireless Networks in Process Control System Environments - DHS US CERT / NIST Special Publication 800-121: Guide to Bluetooth Security / NIST Special Publication 800-97: Establishing Wireless Robust Security Networks: A Guide to IEEE 802.11i / NIST Special Publication 800-48r1: Guide to Securing Legacy IEEE 802.11 Wireless Networks / NIST Special Publication 800-153: Guidelines for Securing Wireless Local Area Networks (WLANs)
	CC.2	Verify each communications channel used by the device implements some type of encryption.	X	X	X	NIST Guide to BT Security - Security Checklist items 13, 14, 16 / Recommended Practices Guide: Securing ZigBee Wireless Networks in Process Control System Environments - DHS US CERT / NIST Special Publication 800-121: Guide to Bluetooth Security
	CC.3	Verify that previously paired devices authenticate each other upon reconnecting.		X	X	
	CC.4	Verify that no services are listening on undocumented ports.	X	X	X	Critical Security Control 11: Limitation and Control of Network Ports, Protocols, and Services
	CC.5	Verify the device is undiscoverable with regard to wireless protocol pairing except as needed for pairing.	X	X	X	
	CC.6	Verify that fallback settings for any protocol are at least as secure as default settings to prevent downgrade attacks.		X	X	
Data Backup/Restore	DB.1	Verify that a user can backup and retain a copy of the device configuration.	X	X	X	Critical Security Control 8: Data Recovery Capability
	DB.2	Verify that any credentials stored in backups are encrypted.		X	X	
	DB.3	Verify that backups are encrypted and protected with a passphrase or key.			X	
	DB.4	Verify that a user can restore a previously backed up configuration.	X	X	X	Critical Security Control 8: Data Recovery Capability
	DB.5	Verify that a user can reset the device to factory defaults, either via a hardware button or through software.	X	X	X	Inferred from Critical Security Control 8: Data Recovery Capability
Updating/Patching	UP.1	Verify that there is a secure method to update or patch the system. This may be in the form of user-initiated update functionality protected by authentication, or automatic updates over a secure channel such as HTTPS.	X	X	X	Critical Security Control 4: Continuous Vulnerability Assessment and Remediation
	UP.2	Verify that patches or updates are integrity-checked (for example with a cryptographic checksum) and delivered over a secure channel.		X	X	
	UP.3	For patches delivered using HTTPS, verify that the application uses SSL pinning.			X	
	UP.4	Verify that patches or updates are cryptographically signed and verified before being applied. Signature and verification should be done with a public/private key pair to mitigate the risk of key extraction if a symmetric key were used.			X	

Figure 1 - Final Draft of the Smart Home Extensions

Requirement Number	Notes	Change notes
CC.1	This could be part of Communications. WPA2 w/TKIP can be broken, as can WEP/WPA	Prior to comments by Rob Ragan (3/19/2015) and Derek Hansen (3/24/2015), requirements were specific to each protocol. For generalizability and to ensure the extensions do not become outdated quickly, the requirements have been updated.
CC.2		
CC.3	Is this already covered by ASVS 10.6: "Verify that all connections to external systems that involve sensitive information or functions are authenticated"?	Removed "...if the protocol provides a mechanism to do so." per feedback by Ryan Speers
CC.4		
CC.5		Split from previous requirement per discussion w/Derek Hansen
CC.6		Added "...to prevent downgrade attacks" per feedback by Ryan Speers.
DB.1		
DB.2		Updated from "hashed" to "encrypted" per feedback from Ryan Speers. This is an L2 requirement, so encryption makes more sense than simply hashing.
DB.3		
DB.4		
DB.5	If a device gets infected or put into a unresolvable state, this may be the only way to fix it.	
UP.1		Split into 3 separate requirements per Dale Rowe
UP.2		Added per feedback by Dale Rowe. Changed "hashed checksum" to "cryptographic checksum" per feedback by Ryan Speers.
UP.3		Added per feedback by Aaron Guzman.
UP.4	Third party CA used for signing. Better protection in case of compromise and need to revoke/reissue.	Added per feedback by Dale Rowe. Updated to "cryptographically signed" per feedback by Aaron Guzman. Updated to add "public/private key" requirement per feedback by Ryan Speers.

Figure 2 - Final Draft of the Smart Home Extensions Continued

5 A SECURITY EVALUATION OF FOUR SMART HOME HUBS

To determine the effectiveness of the Smart Home Extensions in evaluating the security of a hub and to better understand the state of security in existing hubs, four hubs were selected as described in section 3.1.3. Each hub was evaluated against the level one ASVS requirements and all requirements, levels one through three, of the Smart Home Extensions. This was done to ensure that all of the Smart Home Extensions were tested. Although I had planned to perform level two ASVS verification for any hub that passed level one in all categories, none did so.

The ASVS contains 13 categories of verification requirements, the addition of the Smart Home Extensions bringing that total to 14. Restricting evaluation activities to level one requirements reduces that number to 11, as the Cryptography at Rest, Malicious Controls, and Business Logic categories contain only level two and level three requirements. Furthermore, because some hubs did not implement functionality addressed by a requirement, some requirements were marked "Not Applicable" for a particular hub. For example, requirement V16.3 in the Files and Resources category requires that "files obtained from untrusted sources are scanned by antivirus scanners to prevent upload of known malicious content." The Wink HUB, however, does not provide any file upload functionality. It is for this reason that the overall total requirements tested for each hub varies slightly. For convenience, any level one requirements deemed "Not Applicable" are listed on the first page of the testing results spreadsheet for each hub, included in the appendix.

5.1 **Limitations**

It should be noted that evaluation and testing activities that required monitoring Z-Wave wireless communications was not practical due to hardware problems. Due to the proprietary nature of Z-Wave, just two avenues are available for sniffing Z-Wave traffic: the Sigma Designs Z-Wave Development Kit (Sigma Designs Incorporated 2014) or the open source Z-Force Packet Interception and Injection Tool (Fouladi and Ghanoun 2013). The development kit from Sigma Designs was prohibitively expensive at a cost of close to $3,000 dollars, so the Z-Force tool was used instead. The required hardware for this tool was $75 dollars. After over 40 hours of work preparing the hardware, which included installing custom firmware, soldering on header pins, and testing on multiple operating systems, I was unable to get the tool to work. Errors reported by the software could not be traced back to a root cause because, although the z-force tool claims to be open source, the source code does not appear to be publicly available ("Issue 1 - Z-Force - Project Marked as Open Source but No Source Provided - Z-Wave Packet Interception and Injection Tool - Google Project Hosting" 2014).

Another consideration of note is that attacks on servers or devices not owned by me were specifically out-of-scope. All of the hubs communicated with third-party servers on the public Internet and this communication was often essential for the proper functioning of a hub. Where evaluation activities would have required actions that could disrupt or otherwise interfere with out-of-scope assets, the verification requirement was skipped and marked as Not Applicable. The same was done for requirements that could not be verified due to insufficient access to server-side source code. One example of these is requirement 5.10, which requires testing for SQL injection. The VeraLite hub does not use a SQL-type database on the device itself, but the manufacturer's site with which the hub communicates appears to use one. However, as mentioned previously,

performing SQL injection attacks against the manufacturer's site was out of-scope. Consequently, this verification requirement was marked Not Applicable. All activities falling under this category have been noted in the original testing notes, contained in the appendix.

5.2 VeraLite Smart Home Controller

The VeraLite Smart Home Controller is produced by Vera Control, Ltd. and was released in March of 2012. Vera Control releases updates regularly and at the time of testing, the VeraLite was running the most current version of the firmware, version 1.7.541. The VeraLite supports Wi-Fi and Z-Wave protocols. The hub itself is pictured below:

Figure 3 - The VeraLite Smart Home Controller

5.2.1 Overall Results

The VeraLite Smart Home Controller performed poorly as evaluated by the ASVS with the Smart Home Extensions. It was the lowest performing hub when measured solely by the extensions. The hub met all applicable level one requirements in just three of 11 categories. Looking at the table below, we can see that on average, this hub met less than half of the requirements in a single category. Moreover, considering the total number of requirements, the hub met just 19 out of a possible 54, or 35%.

Table 3 - Overall Evaluation Results for the VeraLite Smart Home Controller

Category	Requirements Passed	Requirements Tested	Percent Passed Requirements
Authentication	2	8	25%
Session Management	3	7	43%
Access Control	1	8	13%
Malicious Input Handling	3	6	50%
Error Handling and Logging	1	1	100%
Data Protection	1	2	50%
Communication Security	1	1	100%
HTTP Security	0	3	0%
Files and Resources	0	5	0%
Mobile	4	4	100%
Smart Home Extensions	3	9	33%
Cryptography at Rest	0	0	N/A
Malicious Controls	0	0	N/A
Business Logic	0	0	N/A
Overall Total	***19***	***54***	***35%***

5.2.2 Highlighted Results

Detailed information concerning how the hub performed against each requirement can be found in the appendix. However, to better illustrate how the hub performed, I have highlighted selected results here by category.

5.2.2.1 Authentication

Authentication was a particularly weak category for the VeraLite hub. Requirement 2.1 states that "all pages and resources require authentication except those specifically intended to be public". However, with knowledge of the IP address of the hub, a user may browse to http://<ip_address>/cmh/ and control connected devices without authenticating. Not all functionality is available, but a user could turn on and off lights and other appliances, lock and unlock doors, and arm or disarm door and window sensors.

If a malicious user wanted further control over the hub, testing for requirements 2.4 and 2.16 showed that communication with the hub was sent over unencrypted links, leaving credentials exposed to any attacker eavesdropping on the network. Furthermore, session variables stored in cookies were not expired after a user logged out, allowing me to reuse old values for MMSAuth and MMSAuthSig to successfully authenticate. In such a case, an attacker does not even need to observe the initial authentication; they simply need to steal a recent cookie.

Other problems with authentication included new passwords sent in plaintext via email (2.17), the ability to enumerate accounts using the account registration functionality (2.18), and the use of a hardcoded, shared username and password for accessing the publicly available log server (2.19).

5.2.2.2 Access Control

The VeraLite passed only one of eight requirements under Access Control. The one it did pass was 4.5, the requirement to disable directory browsing unless deliberately desired. However, this success is partially undercut by the failure to pass 4.5, which requires protection against direct object references. In practice, this means that although an attacker cannot see a directory listing, that attacker can see the contents of any file if they know the name and location of it. Additionally, failure to meet requirements in the Access Control category resulted in vulnerabilities such as local file inclusion (4.3), arbitrary file uploads (4.1), and cross-site request forgery (4.16).

5.2.3 Smart Home Extensions Results

Under the Smart Home Extensions, the VeraLite passed three out of a possible nine requirements. Significantly, all three requirements that were met were in the Data Backup/Restore category, with the hub not meeting any of the requirements in the Communications Channel or

Updating/Patching categories. The requirements that were met, DB.1, DB.4, and DB.5, are all functional requirements, which show that the hub can be backed up, restored, or reset to factory defaults. These functions are important in the case of a hub becoming infected by a virus or taken over by a malicious party. In such a case, the only recourse may be a factory reset and restore.

All but one of the failed requirements concerns encryption, potentially making it easier for an attacker to obtain sensitive information that would allow them to take over a hub. The final failed requirement, CC.4, requires that "no services are listening on undocumented ports". At the time of writing, no documentation was available from Vera Control Ltd. that described what services were listening on which ports. As will be seen, and further discussed in chapter seven, this was the case with all of the hubs.

5.3 **Wink HUB**

Wink was originally developed by Quirky, Inc. as an application for smart phones to control smart home devices. The ecosystem of supported devices grew, and in July of 2014, the Wink HUB was released (Quirky Inc. 2014). Quirky releases regular updates and to both the firmware for the hub and to the companion smart phone application. At the time of testing, the hub firmware was version 0.86.0 and the application version was 3.0.5.2. The Wink HUB supports Wi-Fi, Z-Wave, ZigBee, Bluetooth LE, Lutron ClearConnect, and the Kidde protocol. The hub itself is pictured below:

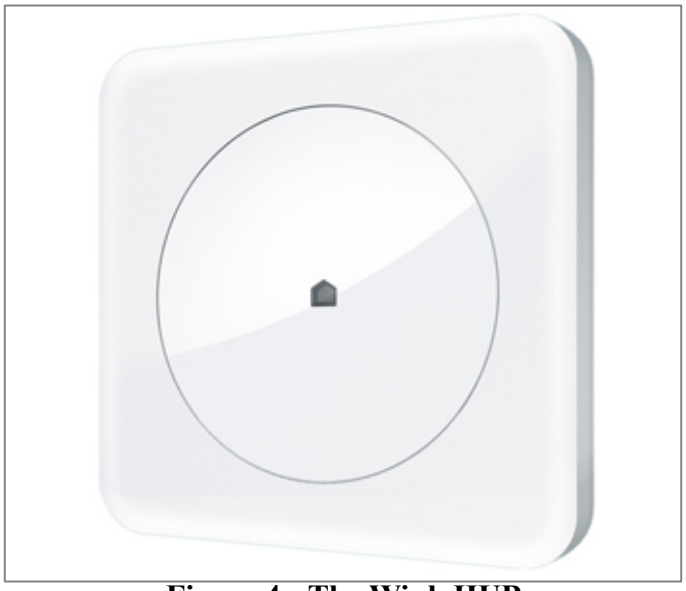

Figure 4 - The Wink HUB

5.3.1 Overall Results

The Wink HUB performed well against the ASVS with the Smart Home Extensions. It was the highest-performing hub as measured solely by the extensions. It was also the only hub to meet 100% of the applicable requirements in seven of the 11 categories, the highest of any hub evaluated. Referring to the table below, we can see that the Wink HUB never scored less than 50% in a category. When looking at the total number of requirements, the hub met 74%, a significant difference from the 35% of the VeraLite, and the second most secure hub as measured by that metric.

Table 4 - Overall Evaluation Results for the Wink HUB

Category	Requirements Passed	Requirements Tested	Percent Passed Requirements
Authentication	5	8	63%
Session Management	2	4	50%
Access Control	4	8	50%
Malicious Input Handling	6	6	100%
Error Handling and Logging	1	1	100%
Data Protection	2	2	100%
Communication Security	1	1	100%
HTTP Security	2	2	100%
Files and Resources	2	2	100%
Mobile	4	4	100%
Smart Home Extensions	6	9	67%
Cryptography at Rest	0	0	N/A
Malicious Controls	0	0	N/A
Business Logic	0	0	N/A
Overall Total	***35***	***47***	***74%***

5.3.2 Highlighted Results

The most notable result from the evaluation of the Wink HUB is that it passed all applicable requirements in so many categories. This may be due, in part, to the fact that Quirky participates in a bug bounty program for the hub (Quirky Inc. 2015a). This is a program whereby independent security researchers may receive payment for discovering and submitting security bugs to Quirky. This type of program can effectively expand the security QA team of a company, and this may be the reason that the Wink HUB is so secure in some areas.

Detailed information concerning how the hub performed against each requirement can be found in the appendix. However, to better illustrate how the hub performed, I have highlighted selected results here by category.

5.3.2.1 Authentication, Session Management, and Access Control

The Wink HUB failed requirements in four categories. One was Smart Home Extensions, which will be addressed presently. Failures in the other three categories, Authentication, Session Management, and Access Control, are related because nearly all of the failures are due to the implementation of the open authentication standard OAUTH 2.0. Quirky's implementation of this standard relies on a bearer token to verify that every action is authorized. Testing showed that if an attacker were to steal a bearer token, they could perform any action on the hub, including deleting the user account, potentially causing a denial of service condition. This would mean that the valid user could not control any of their smart devices until the hub had been restored to factory defaults. As discussed in the following section, such a restoration is difficult to perform.

5.3.3 Smart Home Extensions Results

The Wink HUB performed admirably on many of the Smart Home Extensions, passing 67% of the requirements. While it still supports weak encryption in some areas, which could leave it vulnerable to downgrade attacks, in the Communication Channel subcategory it generally performed well. Where it fell short was in the subcategory of Data Backup/Restore. The findings here were not discovered by any existing requirements in the ASVS, demonstrating the effectiveness of the Smart Home Extensions.

As mentioned previously, one particular attack performed during testing left the hub in an unusable state. Important functionality was inaccessible to the primary user, as the account had been deleted in a simulated attack. Without access to a secondary user, such an attack could prevent the owner from controlling their smart home devices without a factory reset. Unfortunately, the Wink HUB has no factory reset functionality, as required by DB.5. In such a scenario, it is likely that fixing the hub would require a call to Wink HUB support.

A closely related issue is addressed by DB.1 and DB.4. These requirements state that a device should have both a backup and a restore functionality, so that user-defined configurations can be saved. The importance of these is best illustrated by referring to the previously mentioned attack. Let us suppose that a user has installed smart light switches on the main floor, smart bulbs in bedrooms, smart locks on the front and back door, a smart thermostat, and connected cameras covering the front and back doors. This setup could consist of anywhere between 10 and 20 devices. If a user loses access to their account through means malicious or accidental, or the hub undergoes a factory reset, that user will be forced to add each device once again, a process that may take hours. Allowing users to back up and restore their settings would significantly diminish the impact of such an attack.

5.4 **SmartThings Hub**

The SmartThings Hub is made by SmartThings Inc. and has been available since August 2013. Updates to the hub firmware and related smart-phone application are done on a regular basis, with updates to the hub pushed out automatically. During testing, the hub firmware was version 000.011.00705 and the Android application version was 1.7.0. The SmartThings Hub supports Wi-Fi, Z-Wave, ZigBee, and WeMo devices. The hub itself is pictured below:

Figure 5 - The SmartThings Hub

5.4.1 Overall Results

The SmartThings Hub performed fairly well against the ASVS with the Smart Home Extensions, although similar to the VeraLite, it failed all requirements in two categories. It was the second-highest-performing hub when measured solely by the extensions, and third for overall requirements. This hub was also the only one to meet 100% of the applicable requirements in the Authentication category.

5.4.2 Highlighted Results

As mentioned previously, detailed information concerning how the hub performed against each requirement can be found in the appendix. Selected results are highlighted below.

Table 5 - Overall Evaluation Results for the SmartThings Hub

Category	Requirements Passed	Requirements Tested	Percent Passed Requirements
Authentication	8	8	100%
Session Management	5	7	71%
Access Control	5	7	71%
Malicious Input Handling	5	7	71%
Error Handling and Logging	0	1	0%
Data Protection	1	2	50%
Communication Security	1	1	100%
HTTP Security	0	3	0%
Files and Resources	4	4	100%
Mobile	4	4	100%
Smart Home Extensions	5	9	56%
Cryptography at Rest	0	0	N/A
Malicious Controls	0	0	N/A
Business Logic	0	0	N/A
Overall Total	***38***	***53***	***72%***

5.4.2.1 Authentication

The SmartThings hub was the only one of those tested to score 100% in the Authentication category. Given the importance of authentication in the context of a device that can control your home, it is surprising that this was the only hub to pass all of the requirements in this category. It was discovered during testing that SmartThings utilizes the Spring Framework (Pivotal Software 2015). This widely-used Java framework includes an authentication and authorization module, which may explain why the SmartThings Hub did so well in this area.

5.4.2.2 Error Handling and Logging

The failure in Error Handling and Logging, requirement 8.1, states that the application should not output error messages containing sensitive data. In testing, I was able to get the application to output error messages that included information about function names, the framework in use, service versions, and an IP address for an internal server. While this information

is sensitive in the sense that it can help an attacker carry out further attacks, none of it is inherently sensitive or problematic on its own.

5.4.2.3 HTTP Security

The failure to pass requirements 11.2 and 11.3 in HTTP Security is similarly low risk. These requirements concern, respectively, only allowing specific HTTP methods while blocking others, and the inclusion of content-type headers. The failure to comply with 11.2 is little more than incorrect documentation, as the response to an OPTIONS request states that TRACE is allowed, when it is not. 11.3 is a bit more concerning, as some pages do not include a correct content-type header, and could therefore be used in attacks against the browser. This risk is partially mitigated by the fact that only a small number of pages fail to return a correct content-type header.

5.4.3 Smart Home Extensions Results

Evaluating the SmartThings Hub against the Smart Home Extensions turned up results similar to the Wink HUB. There is no backup or restore functionality, although a factory reset is possible through a hardware button. In the case of the SmartThings Hub, the lack of backup or restore functionality is less of a risk because the hub does not have the same account deletion vulnerability. However, if a user wants to replace their hub while keeping all of the same devices associated, or replicate their current setup, the inability to backup and restore a configuration makes these things impossible.

The SmartThings Hub did well in other areas of the Smart Home Extensions, although fallback settings were not nearly as secure as the default settings for many protocols. This was

common among all of the hubs, and suggests that downgrade attacks are less well known, or that compatibility is being put before security.

5.5 Staples Connect Hub

The Staples Connect Hub is a home automation hub marketed by the office supply company Staples. It relies on a home automation platform provided by Zonoff, Inc., a company which specializes in home automation software. The Staples Connect Hub was first released in the Fall of 2013, with a device manufactured by Linksys. One year later, a newer version of the hub was released with a different design, and this time manufactured by D-Link. The D-Link hub, pictured below, was the one used in this evaluation. The D-Link Staples Connect Hub supports Wi-Fi, Z-Wave, ZigBee, Lutron Clear Connect, and Bluetooth LE protocols. The Android application version tested was 1.7. There was not a way to discover the version of the firmware in use.

5.5.1 Overall Results

Based on percent passed requirements, the Connect is the most secure of all hubs reviewed. However, when judged solely by the Smart Home Extensions, the Connect was second overall. It scored 60% or above in all categories. The most significant result is not that the hub performed particularly well or poor in a single category, but that it performed fairly well in all categories.

Figure 6 - The Staples Connect Hub

5.5.2 Highlighted Results

Given that the Connect Hub performed so well in all categories, it is important to understand the impact of those requirements it did not meet. I have highlighted these briefly, below. For detailed information regarding testing procedures and individual requirements, please refer to the appendix.

5.5.2.1 Authentication

In the Authentication category, the Connect failed to meet requirement 2.18 regarding protection against user enumeration. User enumeration is the act of collecting a list of, or enumerating, the valid users of an application. Once an attacker has a list of valid users, it can be used to perform various attacks including mass denial-of-service and horizontal password brute forcing. Depending on the company, user enumeration may be considered a significant risk or a

non-issue. As seen in figure 7, the email addresses of Connect users can be enumerated through the password reset functionality.

Table 6 - Overall Evaluation Results for the Staples Connect Hub

Category	Requirements Passed	Requirements Tested	Percent Passed Requirements
Authentication	7	8	88%
Session Management	6	7	86%
Access Control	6	8	75%
Malicious Input Handling	4	5	80%
Error Handling and Logging	1	1	100%
Data Protection	2	2	100%
Communication Security	1	1	100%
HTTP Security	2	3	67%
Files and Resources	3	3	100%
Mobile	4	4	100%
Smart Home Extensions	5	8	63%
Cryptography at Rest	0	0	N/A
Malicious Controls	0	0	N/A
Business Logic	0	0	N/A
Overall Total	***41***	***50***	***82%***

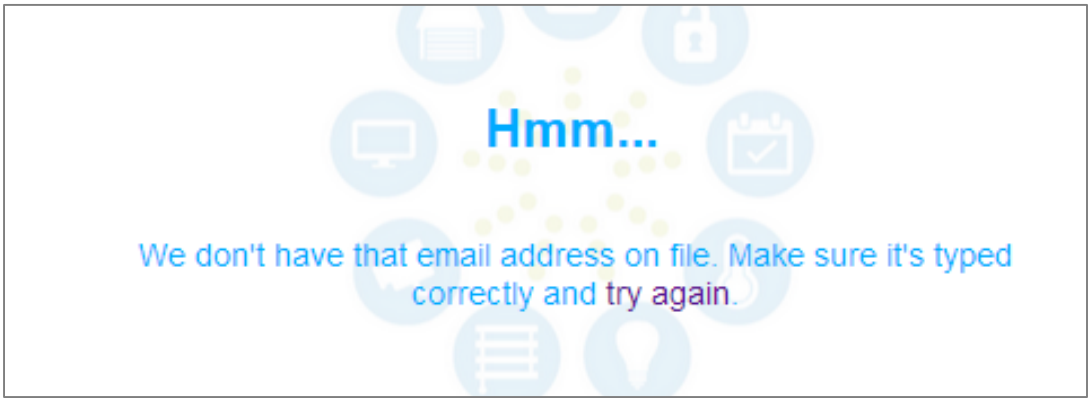

Figure 7- Password Reset Message for the Connect Hub

5.5.2.2 Access Control

The Connect Hub failed two requirements under Access Control: 4.4 and 4.16. The first is a requirement to protect against direct object references. The hub fails this requirement because it is possible, via direct object reference, to access user-interface items that are meant to be hidden. This can be seen in the screenshot below. While this failing does not appear to have any security implications in this instance, it suggests that the application designers did not consider this attack vector. If that is the case, this may be a problem with greater security implications elsewhere in the application.

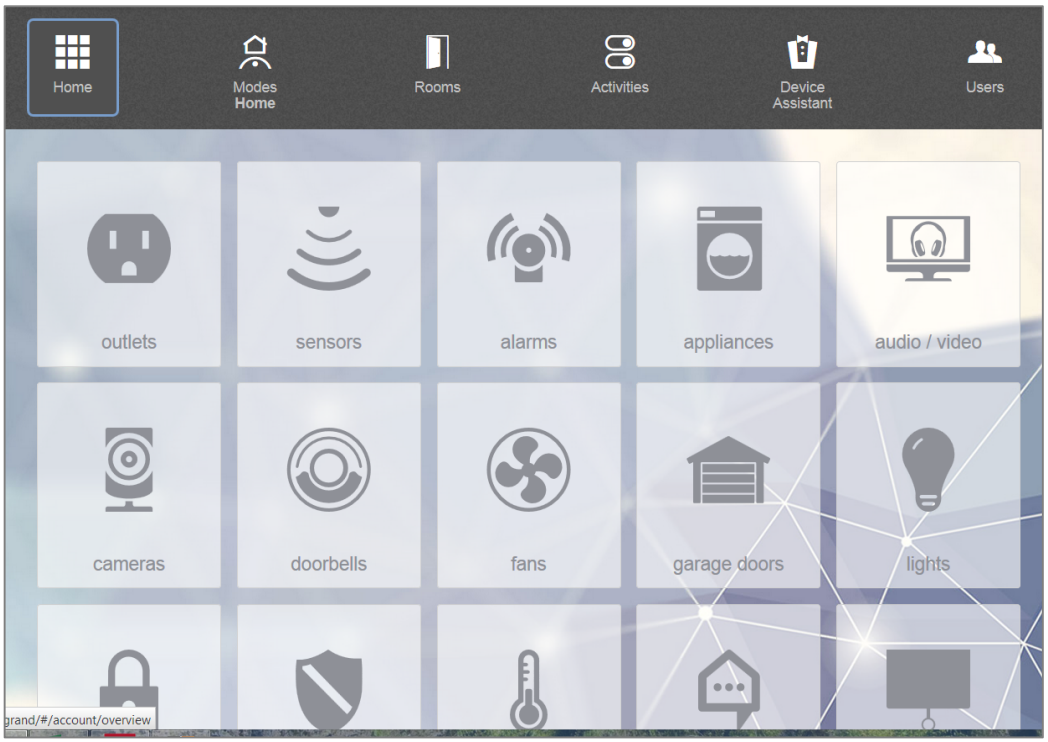

Figure 8 - Interface Items Not Intended for Staples Connect Users were Disclosed via Direct Object Reference

The second requirement, 4.16, requires anti-CSRF tokens for all high value transactions. Zonoff uses websocket requests for communication between their servers and the Connect Hub, and no anti-CSRF tokens are included in these requests. For this reason, the Connect Hub did not meet the requirement. Furthermore, testing indicated that the application may be vulnerable to cross-site websocket hijacking attacks (Schneider 2013), a more concerning problem than the previously mentioned direct object reference.

5.5.2.3 Malicious Input Handling

Under the Malicious Input Handling category, the Connect failed just one requirement. That requirement, 5.5, states that all input validation or encoding routines must be performed and enforced on the server. By intercepting and modifying a websocket request to change the name of a door sensor, I was able to bypass the client-side restrictions on length. While this particular attack had no direct security implications, insufficient server-side input validation can lead to other vulnerabilities such as injection flaws and cross-site scripting.

5.5.3 Smart Home Extensions

In this category, the Connect performed similar to other hubs. A lack of documentation meant that it was impossible to tell if there was anything suspicious among the four open ports (443, 10010, 33791, and 50002) detected in an nmap scan. And fallback settings for SSL still included the cryptographically weak RC4 and MD5 suites, leaving the hub's communications vulnerable to downgrade attacks.

Similar to both the Wink and SmartThings hubs, the Connect does not allow a user to make a backup of their configuration settings. This leaves it open to the same problem suffered by the SmartThings hub if an account were maliciously or accidentally deleted – significant loss of time

for the end user in re-establishing the network of smart devices. Fortunately, the Connect does have a way for a user to perform a factory reset, mitigating the risk of the hub becoming stuck in an unusable state.

5.6 Summary

The below graphs and charts highlight some similarities and differences among the four hubs. Figure 9 shows that all hubs did well in both Mobile and Communication Security, with all hubs passing 100% of applicable requirements. Conversely, all hubs had problems with Session Management, Access Control, and the Smart Home Extensions, with none of the hubs able to meet 100% of the requirements. In addition, in the HTTP Security category neither the VeraLite nor the SmartThings hubs were able to pass a single requirement. Exact percentages are displayed in Table 8 for convenience. Further visual comparison can be made with Figures 10 through 13, which are included below.

Table 7 shows that the Wink scored the highest in the greatest number of categories, followed closely by the Connect. Going by this table, it is clear that the VeraLite performed the worst. Investigating this further by referring to Figure 9, we see that the VeraLite hub failed to outperform another hub in any category.

Table 7 - Highest Rated Hubs

Hub	Highest Rated in a Category
VeraLite	3
Wink	8
SmartThings	4
Connect	7

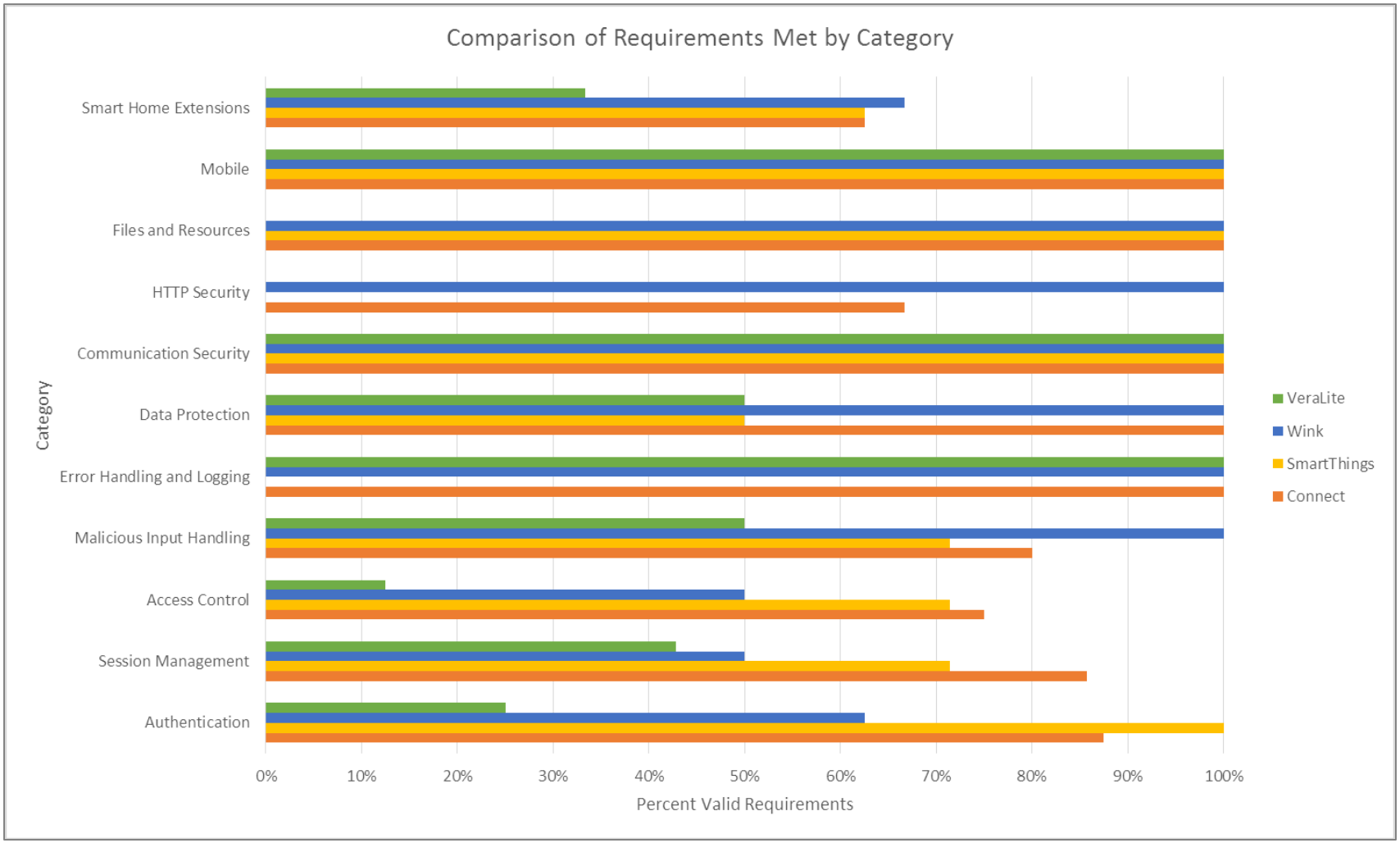

Figure 9 - Percentage of Requirements Met by Category for All Hubs

Table 8 - Percentage of Requirements Met by Category for All Hubs

	Authentication	Session Management	Access Control	Malicious Input Handling	Error Handling and Logging	Data Protection	Communication Security	HTTP Security	Files and Resources	Mobile	Smart Home Extensions
VeraLite	25%	43%	13%	50%	100%	50%	100%	0%	0%	100%	33%
Wink	63%	50%	50%	100%	100%	100%	100%	100%	100%	100%	67%
SmartThings	100%	71%	71%	71%	0%	50%	100%	0%	100%	100%	63%
Connect	88%	86%	75%	80%	100%	100%	100%	67%	100%	100%	63%

Percent Valid Requirements

Table 9 shows the performance of each hub as measured by the Smart Home Extensions, by requirement. As this table makes clear, I was not able to test requirements CC.3, UP.2, and UP.4. This was due to the limitations mentioned previously in Section 5.1. What stands out here is that all of the hubs had trouble with requirement CC.6, which is the requirement that covers fallback settings for encrypted connections. Although the VeraLite did not fail this requirement, that hub did not implement encryption, leaving it possibly more vulnerable than the hubs that failed the requirement.

The other result of note is each hub's result for DB.1, which requires that a user can backup and retain a copy of the device configuration. The VeraLite hub was the only one to provide this functionality. The potential impact of not providing this has been covered previously. This also explains why the other hubs were not evaluated against DB.2, DB.3, or DB.4, as those requirements depend upon DB.1.

Finally, I was only able to verify requirement UP.3 with the Wink HUB. This requirement states that for patches delivered using SSL, the application should utilize certificate pinning. Without access to source code or debug logs on the devices, it was impossible to know whether certificate pinning was being used. However, during the testing window the SSL certificate for the Wink HUB expired, resulting in an unusable hub (Quirky Inc. 2015b). Due to the nature of the failure and the manufacturer's suggested fix, I was able to infer that SSL certificate pinning was used on the Wink.

Table 9 - Hub Performance on Smart Home Extensions

Requirement	VeraLite	Wink	SmartThings	Connect
CC.1	Fail	Pass	Pass	Pass
CC.2	Fail	Pass	Pass	Pass
CC.3	N/A	N/A	N/A	N/A
CC.4	Fail	Pass	Fail	Fail
CC.5	N/A	Pass	Pass	Pass
CC.6	N/A	Fail	Fail	Fail
DB.1	Pass	Fail	Fail	Fail
DB.2	Fail	N/A	N/A	N/A
DB.3	Fail	N/A	N/A	N/A
DB.4	Pass	N/A	N/A	N/A
DB.5	Pass	Fail	Pass	Pass
UP.1	Fail	Pass	Pass	Pass
UP.2	N/A	N/A	N/A	N/A
UP.3	N/A	Pass	N/A	N/A
UP.4	N/A	N/A	N/A	N/A

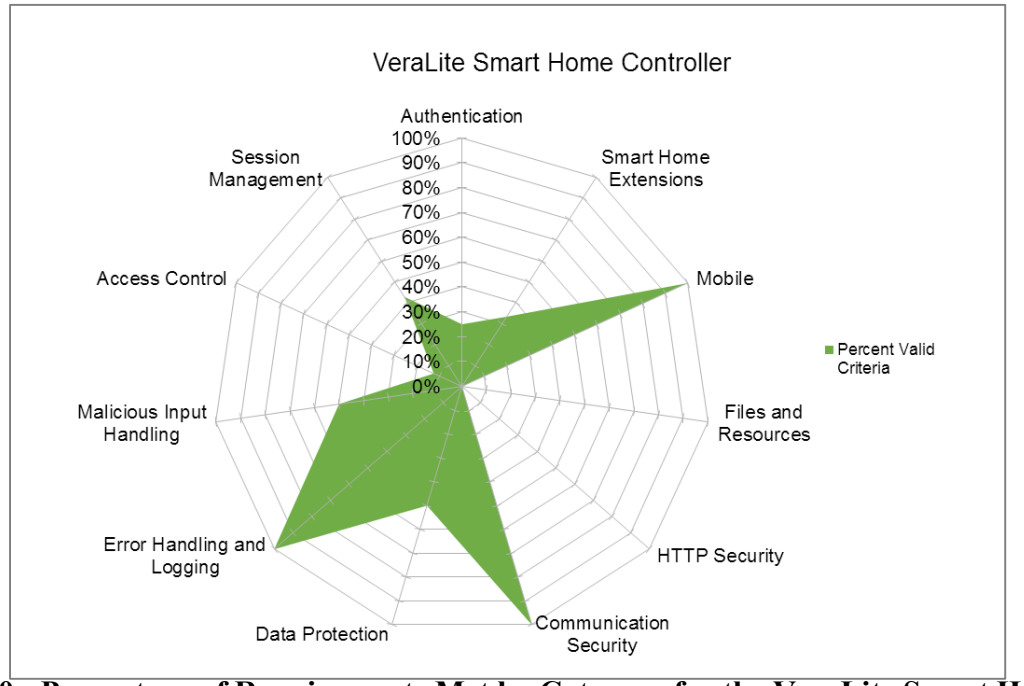

Figure 10 - Percentage of Requirements Met by Category for the VeraLite Smart Home Controller

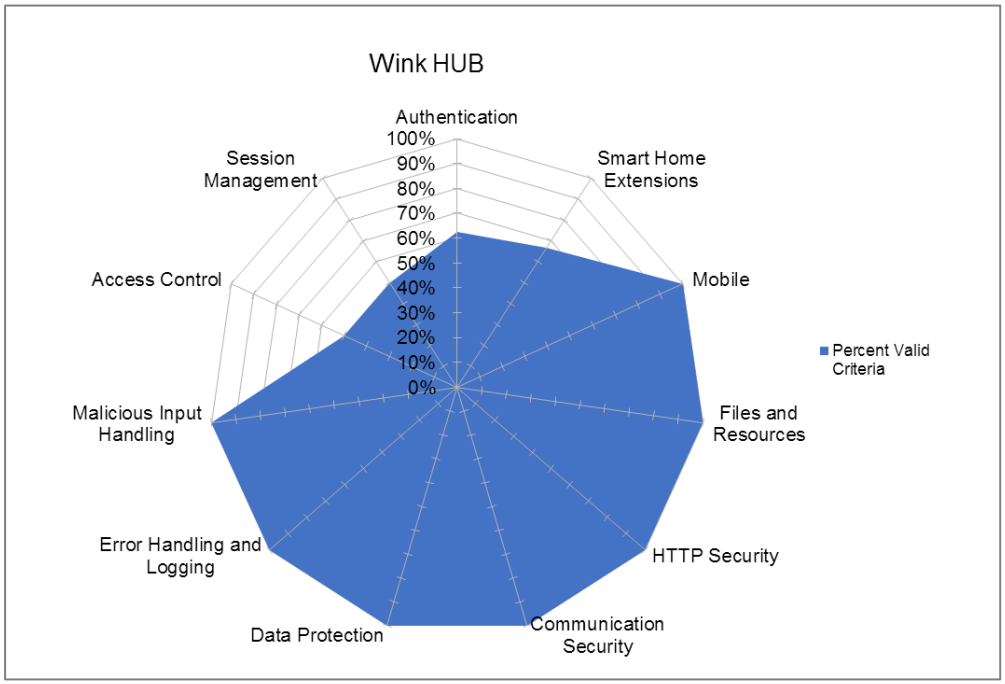

Figure 11 - Percentage of Requirements Met by Category for the Wink HUB

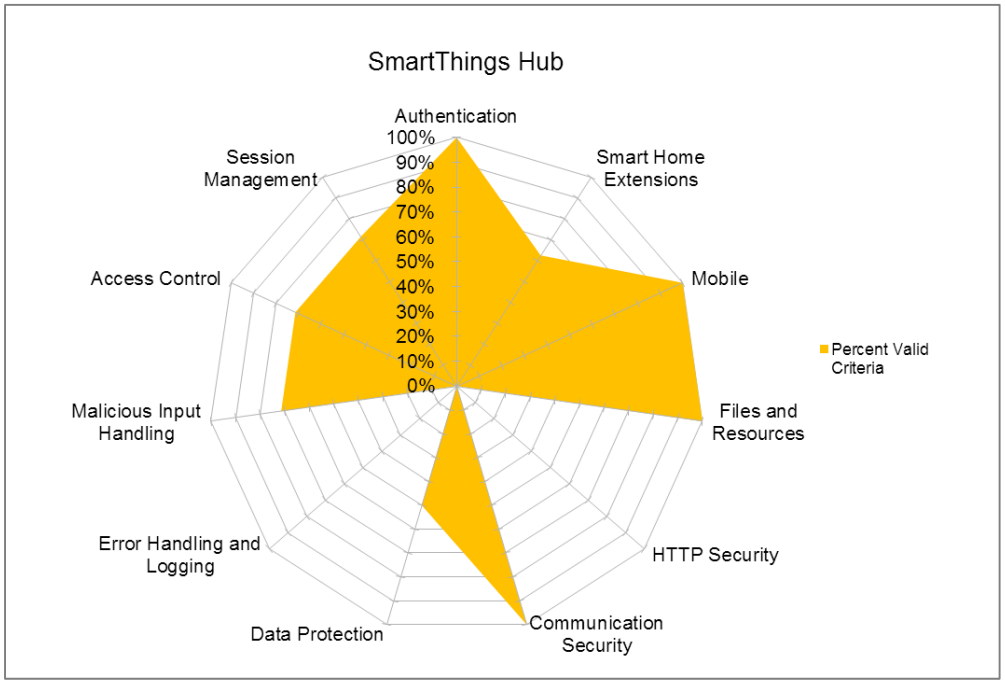

Figure 12 - Percentage of Requirements Met by Category for the SmartThings Hub

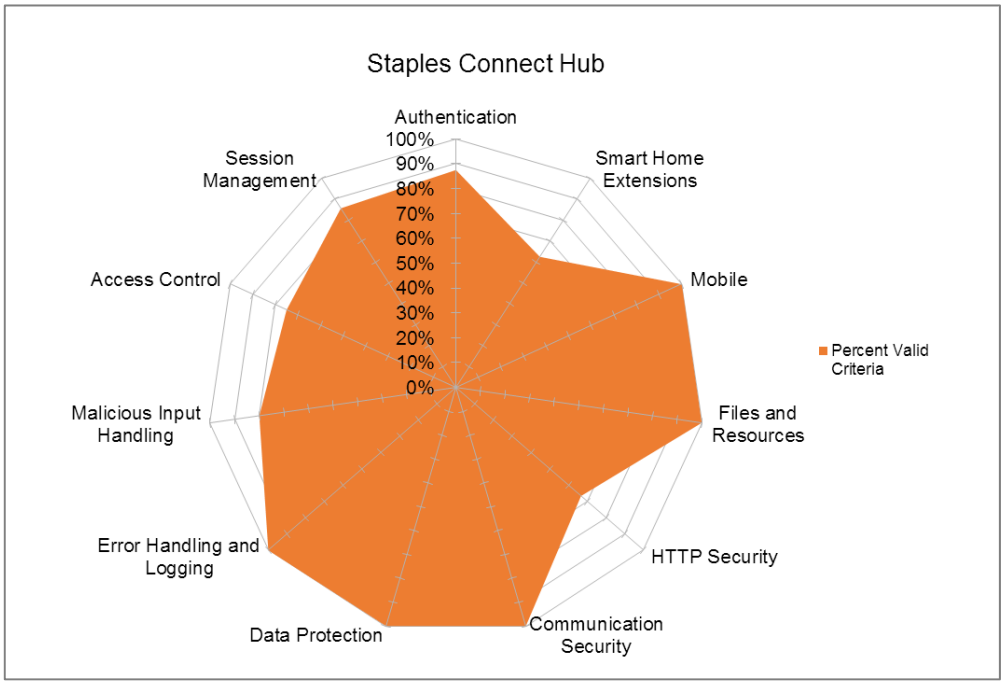

Figure 13 - Percentage of Requirements Met by Category for the Staples Connect Hub

6 RECOMMENDATIONS FOR SMART HOME HUB SECURITY

The following are my recommendations for a more secure smart home hub. Separate recommendations have been developed for end-users and hub manufacturers in the smart home arena. All of the recommendations are based on my research into existing best practices and the results of performing a security evaluation of four smart home hubs, described previously.

6.1 Recommendations for End-Users

1) **Utilize Your Wireless Router's Guest Network Functionality if Available.** Many of the vulnerabilities discovered during the evaluation of the hubs require an attacker to be on the same network as the hub to exploit. If friends, family, or neighbors request access to your wireless network, give them access to the guest network, not the network containing the hub and other smart devices. Use a different password for your regular network and the guest network.
2) **Use the Strongest Available Modern Encryption Scheme to Secure your Wireless Network.** At the time of writing, this is WPA2-PSK with AES or WPA2-Enterprise with a Radius server. All other options at the time or writing have proven weaknesses, making it easier for an attacker to obtain access to your wireless network. If your router offers it, disable WPS PIN-based access.

3) **Use Strong Passwords and Change Default Passwords.** Make sure the passwords on your wireless network and hub are long and complex. Passwords should be a minimum of 12 characters long and contain a mix of upper- and lower-case letters, numbers, and special characters. Change any passwords that shipped with a device, or that you did not create yourself.

6.2 Recommendations for Hub Manufacturers

1) **Perform a Comprehensive Security Review of the Hub.** If the hub is still under development, security requirements should be part of the design and functional specifications. Security should also be integrated into the entire development lifecycle of the product. Perform a penetration test against the hub prior to release. These activities should include a security review of any third-party services which the hub relies upon for operation.

   If the hub has already been released, perform a penetration test against the hub. If security expertise is not available within the organization, hire a third party who is familiar with these types of assessments or consider a bug bounty program.

2) **Consider Using an Existing Standard or Framework for Sensitive Functionality.** This includes authentication, authorization, session management, input validation, cryptography, output encoding/escaping, error handling and logging, data protection, and communication security. Using existing standards and frameworks, you leverage the security knowledge and experience of others to improve the security of your product. It can also help you lower R&D risks and costs, provide transparency to prospective customers, and avoid duplication of work.

3) **Ensure that your Hub has Back-Up, Restore, and Factory Reset Functionality.** In the event that a hub is compromised – whether through malicious hacking, malware, or data

corruption – it is important that a customer be able to reset and, if possible, restore their previous settings.

7 DISCUSSION

7.1 Evaluation of the Extensions

Evaluating the Smart Home Extensions can be done by ensuring they are not flawed, and by asking how well they provided insight into the security posture of the hubs. I tested them for flaws in two ways. First, as described in section 3.1.2, industry experts reviewed the requirements and they were then updated to reflect feedback. Second, by attempting to verify the requirements, I was able to confirm that it was possible to pass or fail a requirement. As mentioned previously, all hubs failed some requirements, and they did not all fail the same ones. Likewise, they did not all pass the same requirements. This can be seen by referring back to Table 9 in section 5.6. These results suggest that the requirements are not flawed, in that they are not impossible to pass or impossible to fail.

As has been mentioned already, three extension requirements went untested: CC.3, UP.2, and UP.4. CC.3 requires that previously paired devices re-authenticate each other upon reconnecting. This requirement could not be tested due to insufficient testing hardware. A RZUSBSTICK (Atmel Corporation 2008) running the killerbee framework (Wright 2011) was used to sniff traffic. However, what did not come out in preliminary research was that ZigBee devices can perform channel hopping among 16 channels, changing channels with each message. An ideal testing environment would contain 16 RZUSBSTICK devices, one to monitor each channel. Without such a setup, I was unable to capture the necessary traffic to confirm this

requirement. However, I was able to capture traffic, indicating that with the right tools CC.3 could be verified.

UP.2 and UP.4 concern cryptographically verifying patches or updates. Because this sort of verification was done on the device itself, testing would have required access to source code, which was not available. Alternatively, a patch could have been intercepted and tampered with, then sent on to the hub. If the patch was applied even after tampering, it could be inferred that the hub does not perform cryptographic verification, or that the implementation is flawed. Unfortunately, no patching or update activity was observed during the testing window. Given access to source code, it is reasonable to believe that these requirements could be verified.

The summary of findings, below, details how well the Smart Home Extensions provided insight into the security posture of hubs.

7.2 Summary of Findings and Significance of Appendix

None of the hubs evaluated passed all of the security requirements, and testing showed that every hub had some security vulnerabilities, some of them previously unknown or undisclosed. The majority of the security vulnerabilities can be considered low to medium risk; however, more severe vulnerabilities included local file inclusion, which resulted in obtaining the root password hash, and cross-site scripting. Furthermore, not a single hub was able to pass all of the requirements in the Smart Home Extensions. This is not surprising given that no hub passed all of the ASVS level one requirements, and the extensions include requirements for levels two and three. The most significant security vulnerabilities discovered in each hub, along with the related ASVS or Smart Home Extension requirement, are summarized in the table below. More information on each vulnerability, as well as a complete log of testing results, is in the appendix.

The ‘Further Details’ column identifies the spreadsheet file, worksheet and requirement number to reference for this information.

Table 10 - Table of References for Hub Vulnerabilities

Hub	Vulnerability Type	Further Details
VeraLite Smart Home Controller	Authentication bypass	Appendix B - VeraLite.xlsx | 2 - Authentication | V2.1
	Session reuse	Appendix B - VeraLite.xlsx | 2 - Authentication | V2.4
	Session reuse	Appendix B - VeraLite.xlsx | 3 - Session Management | V3.2
	Local file include - originally demonstrated by Xipiter (2014)	Appendix B - VeraLite.xlsx | 4 - Access Control | V4.3
	Local file include - originally demonstrated by Xipiter (2014)	Appendix B - VeraLite.xlsx | 16 - Files and Resources | V16.2
	Log disclosure	Appendix B - VeraLite.xlsx | 2 - Authentication | V2.19
	File upload unprotected by authentication	Appendix B - VeraLite.xlsx | 16 - Files and Resources | V16.5
	Account enumeration	Appendix B - VeraLite.xlsx | 2 - Authentication | V2.18
	Open redirect	Appendix B - VeraLite.xlsx | 16 - Files and Resources | V16.1
	Insufficient documentation	Appendix B - VeraLite.xlsx | Smart Home Extensions | CC.4
Wink HUB	Passwords stored in plaintext	Appendix B - Wink.xlsx | 2 - Authentication | V2.16
	Sensitive information disclosure	Appendix B - Wink.xlsx | 4 - Access Control | V4.5
	Insufficient reset/restore functionality	Appendix B - Wink.xlsx | Smart Home Extensions | DB.1
	Insufficient reset/restore functionality	Appendix B - Wink.xlsx | Smart Home Extensions | DB.5
Staples Connect Hub	Account enumeration	Appendix B - Staples Connect.xlsx | 2 - Authentication | V2.18
	Insecure cookies	Appendix B - Staples Connect.xlsx | 3 - Session Management | V3.15
	Direct object reference	Appendix B - Staples Connect.xlsx | 4 - Access Control | V4.4
	Insufficient documentation	Appendix B - Staples Connect.xlsx | Smart Home Extensions | CC.4
SmartThings Hub	Sensitive information disclosure	Appendix B - SmartThings.xlsx | 3 - Session Management | V3.6
	Sensitive information disclosure	Appendix B - SmartThings.xlsx | 8 - Error Handling and Logging |8.1
	Cross-site scripting	Appendix B - SmartThings.xlsx | 5 - Malicious Input Handling | V5.16
	Insufficient documentation	Appendix B - SmartThings.xlsx | Smart Home Extensions | CC.4

These results show that the OWASP Application Security Verification Standard combined with the Smart Home Extensions can be used to discover previously unknown security vulnerabilities, providing critical insight into the security posture of a smart home hub. If hub manufacturers used these tools to assess and mitigate risks, both hub manufacturers and users would be better off.

The results of the security evaluation suggest that the level of security in the current crop of smart home hubs varies widely, and in some cases may put users’ possessions, data, and safety at risk. As detailed in chapter two, a number of security standards and guides exist that could be applied to smart home hubs. While none of them completely addresses the functionality of hubs, the proposed Smart Home Extension to the ASVS may fill that gap in coverage, as shown by this body of research. Whether hub manufacturers select this or another method, there are resources

available to improve the security of smart home hubs. In a market where users are entrusting device manufacturers with the keys to their homes, failing to implement best practices is irresponsible.

7.3 **Hub Rankings**

Although the purpose of this research was not to determine which of the hubs was the most secure, the question deserves to be addressed. Regarding this question, it must be noted that security in this context is application-specific. As explained by security researcher Brenda Larcom, "What one stakeholder thinks is the very nature of security may be unimportant to another." (Larcom 2015). Depending on the context of use, certain vulnerabilities may be considered severe, or of negligible risk. The hub rankings discussed here are based solely on the performance of each device as measured by the ASVS and Smart Home Extensions.

Based on the total number of requirements met, the Staples Connect Hub is the most secure hub, meeting 41 requirements. Because not all hubs offered the same set of features, not all requirements applied to all hubs. For this reason, the percent of passed requirements was also calculated. The Connect also came in first as measured by this metric, with 82% passed.

The Wink HUB was the second most secure by percent passed requirements, at 74%, but was beat out by the SmartThings in total number of requirements met, with 35 versus 38 for the SmartThings. As will be explained shortly, however, this metric is somewhat misleading. When measured solely by the Smart Home Extensions, the Wink performed the best, with the Connect a close second.

While the SmartThings Hub did beat out the Wink in total number of requirements met, it was third in the majority of metrics highlighted here. Where the SmartThings did stand out was

in the Authentication category. As mentioned in chapter five, it was the only hub to pass 100% of the requirements in that category.

The VeraLite performed very poorly overall. While it passed 100% of requirements in three categories, it did not outperform another hub in a single category. Here I would like to highlight again that security in this context is application-specific. The VeraLite offers many customization options, going so far as to allow a user to upload and run their own LUA scripts to define interactions with smart devices that are not yet officially supported. Some users may prioritize that functionality above other security controls. Neither the ASVS nor the Smart Home Extensions account for this. Nevertheless, when it comes to implementing best practices, the VeraLite falls short.

I mentioned previously that the SmartThings hub met more requirements than the Wink HUB, but that this metric could be misleading. Figure 14 shows the distribution of passed requirements across categories for each hub. Comparing the graph for the Wink with that for SmartThings, we see that the greater number of requirements met by the SmartThings were in a smaller number of categories. This figure also illustrates that Authentication, Session Management, Access Control, and Malicious Input were a problem for the majority of hubs, as can be seen by observing the top-left quadrant of each graph. Looking at the top-right, we see that no hub did particularly well when it came to the Smart Home Extensions.

This figure also shows that the results for the Connect and the Wink were very similar. The overall security of each might be considered comparable if one were to go solely off percentages and graphs. However, the type of vulnerabilities found in each hub could have widely differing impacts depending on the environment. This is another case where context matters.

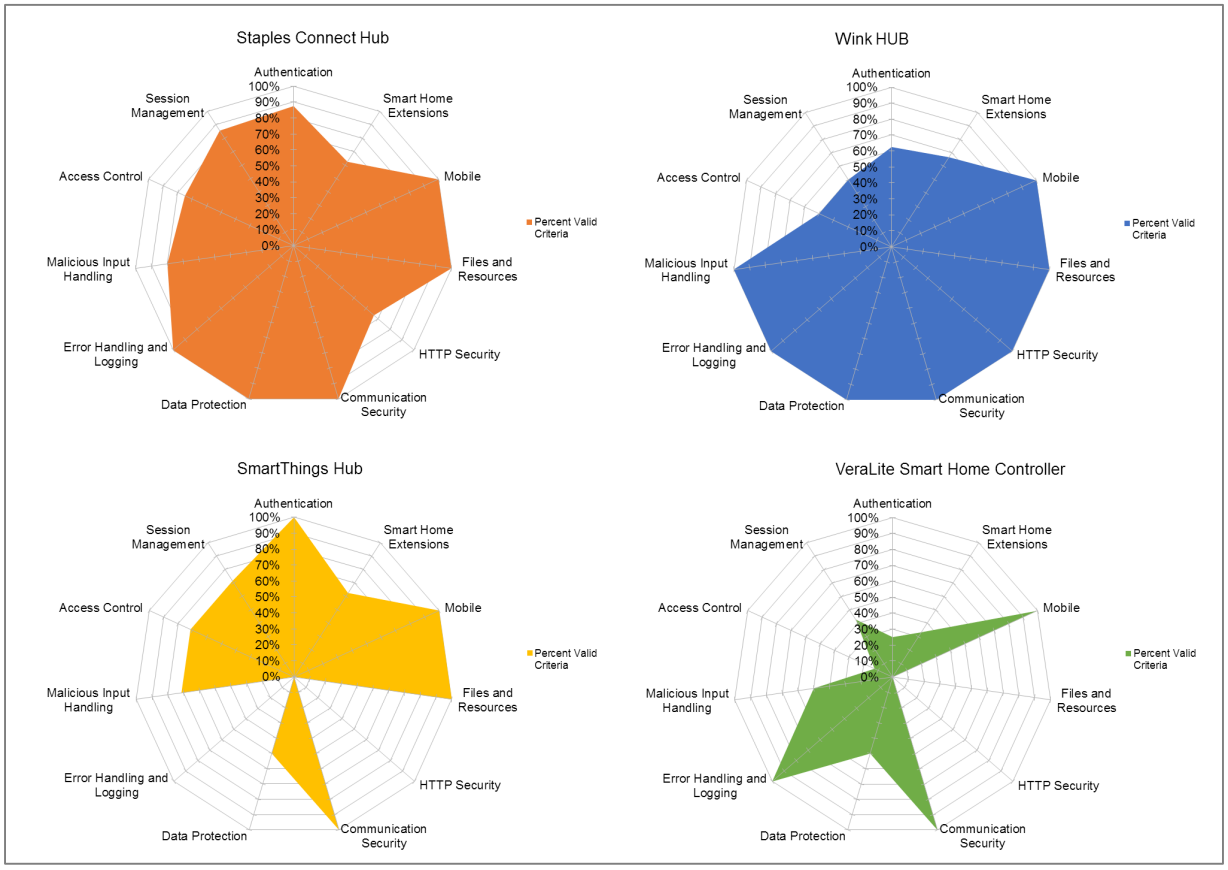

Figure 14 - Distribution of Passed Requirements Across Categories for Each Hub

7.4 Design Issues

Many design issues were encountered during setup and testing of the hubs that have low security impact, but are worth highlighting nonetheless. For example, when associating a Z-Wave or ZigBee device with a hub, the device must be within wireless range of the hub to communicate. Later, when the device is in its permanent location, perhaps in an upstairs light socket, it does not itself need to be within range of the hub, as long as it has a route to the hub through the mesh network. However, some devices cannot easily be relocated. To associate with a mesh networking device that is out of range, some hubs can run off of batteries for a short period of time. Hubs that do not have this ability must be shut down and moved to a location where they can associate with

the new device, then moved back to the original location. It may seem like a small issue, but disabling the entire smart home network for a time is a security risk, not to mention a usability problem.

A design issue mentioned previously is the lack of a factory reset on some devices. After setting up the Wink HUB on a particular wireless network, it was determined that testing would have to be done in a different location, and therefore on a different wireless network. Unfortunately, the Wink provided no way to disassociate from the current wireless network and no way to perform a factory reset. The only way of getting the hub into setup mode was to be out-of-range of the associated wireless network for a period of 5 minutes or more. Due to the extensive coverage of the wireless network to which the Wink had been joined, a faraday cage had to be used to get it out-of-range of the network. A factory reset button would have saved significant time and effort.

One issue for potential hub users is the extensive reliance on third-party servers. Every hub evaluated here requires access to the Internet, not just for convenience, but also as a requirement for proper functioning of the hub. Users are authenticated using the manufacturer's servers, meaning that any outage which prevented the hub from contacting those servers would prevent users from using their smart devices. One can imagine a situation in which thousands of users are locked out of their homes because a malicious group launched a denial-of-service attack against the authentication servers for the Staples Connect.

With the market still in its early stages, these types of issues are mostly minor inconveniences for a handful of users. However, as the market grows, so too will the impact of poor design decisions. Hub manufacturers would do well to consider these issues when designing a hub and users would do well to stay informed of the risks they accept when choosing one.

7.5 Limitations of Findings

The findings discussed here rely on certain assumptions that should be taken into account when considering their impact. Furthermore, testing activities were restricted by time, privacy, and legal concerns.

7.5.1 Network Access

All tests were performed under the assumption that the attacker was on the same network as the target device. For home wireless networks, this assumption is strengthened by the existence of vulnerabilities affecting WEP, WPA, and WPA2 w/TKIP encryption schemes, as well as WPS PIN-based access. Tools available to exploit these vulnerabilities include reaver-wps, wifite, Aircrack-ng, and more.

7.5.2 Black-Box Testing

Testing was performed under black-box conditions. This means that no privileged knowledge of the hubs, associated applications, or source code was available. I assume that an attacker would be working under similar constraints. However, privileged knowledge or insider access to source code and other details of a hub could result in more significant risks than those discovered by this research.

7.5.3 Time

The evaluation of hubs was time-constrained to an estimated 80 hours per hub. This time included setup of tools and understanding how a hub was expected to function. With more time, further vulnerabilities may have been discovered. A motivated attacker would likely not be under

the same time constraints, and could dedicate more time to discovering and exploiting vulnerabilities, perhaps months.

7.5.4 Legal and Privacy Constraints

Some potential attacks were not performed because they were illegal or unethical. For example, all of the hubs communicated with the manufacturer's servers, and many communicated with other third-party servers. Attacks against third-party servers were out-of-scope, and attacks against manufacturer servers were limited to those only likely to affect my own accounts, for privacy and legal reasons. Denial-of-service, social engineering, brute-force, and similar attacks were therefore out-of-scope. A real-world attacker would not operate under the same constraints, and therefore might find more vulnerabilities.

7.6 Limitations of the Smart Home Extensions

The Smart Home Extensions also have certain limitations inherent in their design. Known limitations are documented here.

7.6.1 Physical Security

Evaluating the physical security of the hubs was specifically out-of-scope for the purposes of this research. The assumption driving this is that an attacker that has physical access to a hub is likely already in the home and can access sensitive data in other ways.

7.6.2 Privacy

Multiple reviewers brought up the concern that the Smart Home Extensions do not have a category for privacy. There are two main reasons for this. First, many of the existing requirements

across various categories of the ASVS touch on privacy issues already. This includes requirements regarding account enumeration, verbose error messages, and the use of encryption, to name a few. Attempting to enumerate all potential privacy issues would result in overlap with existing requirements and likely serve to confuse users.

The second reason privacy was left for other researchers is that the ASVS is an application security standard, not a privacy standard. While privacy is a major issue in the field of smart homes and the Internet of Things as a whole, the Smart Home Extensions are designed to extend the *security requirements* of the ASVS to cover smart home hubs. Depending on the context, privacy and security may be opposing goals, a problem which the ASVS is not designed to address.

7.6.3 Expertise and Tools

Evaluating a device or application according to the requirements contained in the ASVS requires no specific tools other than a computer and an understanding of each requirement. Due to the protocols involved, an evaluation involving the requirements of the Smart Home Extensions may require specialized tools. This includes antennas and monitoring tools for Wi-Fi, Bluetooth, Z-Wave, ZigBee, Insteon ClearConnect, and other protocols and technologies. Not all of the tools are widely available, and as I experienced, some of the tools are unreliable. This potential limitation should be resolved prior to beginning testing activities.

7.7 Intended Application of the Smart Home Extensions

The Smart Home Extensions are designed to evaluate the security of smart home hubs in tandem with the ASVS. It is expected that Security or QA personnel who are interested in evaluating a smart home hub will use them. Because they are an extension to the ASVS, they are only meant to be used with it. Considered on their own, they are incomplete.

It is possible that the extensions and the ASVS as a whole could be used to evaluate the security of other smart devices. Before pursuing this course, it would be necessary to determine what functionality the device or class of devices in question offered. By comparing this to what is covered by the ASVS and the Smart Home Extensions, once could start to understand how well the they might work in evaluating the product. More work would have to go into the preparation and adaptation of the requirements, but much of the methodology has been laid out here.

7.8 **Future Work**

The leading smart home hubs on the market today are first- and second-generation devices. To improve on the next generation, more research in the area of smart home security is vital. Given the pace at which smart devices are being introduced, an easy-to-understand security standard is already overdue. Future research could integrate the Smart Home Extensions into the ASVS. With some modification, such a document could be adapted to a wide variety of smart devices, and the ASVS could be applied to any device that hosts an application.

The data set created by this research could also be used to update the OWASP Internet of Things Top 10 List. As it stands, the current list is based on an evaluation of just 10 devices. More data, even from four devices, would improve the accuracy and therefore usefulness of that project.

Along those lines, more research looking at the relative security or insecurity of devices and products that are meant to help manage people's lives is needed. As articulated by Thomas S. Monson, "When performance is measured, performance improves. When performance is measured and reported, the rate of improvement accelerates." (Monson 1970) By measuring and reporting on the security performance of smart devices, that performance will improve.

REFERENCES

Amazon Inc. 2014. “Schlage RS100HC V N N SL Home Door and Window Sensor with Nexia Home Intelligence (Z-Wave) - Household Alarms And Detectors - Amazon.com.” http://www.amazon.com/Schlage-RS100HC-SL-Window-Intelligence/dp/B008Q5CTBE.

Andersen, S., G. Marshall, E. Mitchell, and R. Winkler. 2011. “Threats and Countermeasures Guide : Security Settings in Windows 7 and Windows Server 2008 R2,” no. May.

Apple Inc. 2015. “Security Configuration Guides - Apple Support.” Accessed May 25. https://www.apple.com/support/security/guides/.

Atmel Corporation. 2008. “RZUSBstick.” http://www.atmel.com/tools/RZUSBSTICK.aspx.

Axelos. 2015. “What Is ITIL?” Accessed June 30. https://www.axelos.com/best-practice-solutions/itil/what-is-itil.

Batard, F. 2014. “OWASP ASVS 2.0 CheatSheet - IKangae | IKangae.” http://www.ikangae.net/application-security/owasp-asvs-2-0-cheatsheet/.

Brown, R. 2014. “A New Home for the Smart Home at CES 2015.” *CNET*. http://www.cnet.com/news/ces-2015-appliances-and-smart-home/.

CCRA. 2012. “Common Criteria for Information Technology Security Evaluation Part 3 : Security Assurance Components.” *Security*. NIST.

Cook, D.J., M. Youngblood, E.O. Heierman III, K. Gopalratnam, S. Rao, A. Litvin, and F. Khawaja. 2003. “MavHome: An Agent-Based Smart Home.” *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, 2003. (PerCom 2003).*, 1–4. doi:10.1109/PERCOM.2003.1192783.

Council on Cyber Security. 2014. “The Critical Security Controls for Effective Cyber Defense - v5.0.” SANS.

Crowley, D., J. Savage, and D. Bryan. 2013. “Home Invasion 2.0: Attacking Network-Connected Embedded Devices.” https://media.blackhat.com/us-13/US-13-Crowley-Home-Invasion-2-0-WP.pdf.

Davidoff, S., M.K. Lee, and C. Yiu. 2006. “Principles of Smart Home Control.” *UbiComp 2006: ...*, 19–34.

Fouladi, B., and S. Ghanoun. 2013. “Security Evaluation of the Z-Wave Wireless Protocol.”

FTC Staff Report. 2015. *Privacy & Security in a Connected World*.

Gomez, C., and J. Paradells. 2010. “Wireless Home Automation Networks: A Survey of Architectures and Technologies.” *IEEE Communications Magazine*, no. June: 92–101.

Han, D., and J. Lim. 2010. “Design and Implementation of Smart Home Energy Management Systems Based on Zigbee.” *IEEE Transactions on Consumer Electronics* 56 (3): 1417–25. doi:10.1109/TCE.2010.5606278.

Hussein, A., M Adda, M. Atieh, and W. Fahs. 2014. “Smart Home Design for Disabled People Based on Neural Networks.” *Procedia Computer Science* 37. Elsevier Masson SAS: 117–26. doi:10.1016/j.procs.2014.08.020.

International Organization for Standardization. 2015. “ISO 27000 - ISO 27001 and ISO 27002 Standards.” Accessed May 25. http://www.27000.org/index.htm.

ISACA. 2012. “COBIT 5: A Business Framework for the Governance and Management of Enterprise IT.” http://www.isaca.org/cobit/pages/default.aspx.

“Issue 1 - Z-Force - Project Marked as Open Source but No Source Provided - Z-Wave Packet Interception and Injection Tool - Google Project Hosting.” 2014. https://code.google.com/p/z-force/issues/detail?id=1.

Jakkula, V., and D. J. Cook. 2008. “Anomaly Detection Using Temporal Data Mining in a Smart Home Environment.” *Methods of Information in Medicine* 47: 70–75. doi:10.3414/ME9103.

Jasco Products Company. 2015. “GE Z-Wave Plug-In Smart Switch | Jasco.” http://jascoproducts.com/products/ge-z-wave-plug-smart-switch.

Kalofonos, D. N., and S. Shakhshir. 2007. “IntuiSec: A Framework for Intuitive User Interaction with Smart Home Security Using Mobile Devices.” *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC*. doi:10.1109/PIMRC.2007.4394312.

Larcom, B. 2015. “Security Should Be Application-Specific - Bishop Fox.” http://www.bishopfox.com/blog/2015/04/security-should-be-application-specific/.

Lodamo, A. T., and S. Forsström. 2012. “M2M Protocols, Solutions and Platforms for Smart Home Environments.”

Manadhata, P. 2008. *An Attack Surface Metric*. Pittsburgh: Carnegie Mellon University, School of Computer Science.

Martinsburg College. 2015. “Smart Home Technology Certificate Program - Martinsburg College.” Accessed February 27. http://martinsburgcollege.edu/programs/certificate-programs/digital-technology-integration/smart-home-technology/.

Meucci, M., and A. Muller. 2014. “OWASP Testing Guide 4.0,” no. Cc.

MITRE. 2014. “CWE - Frequently Asked Questions (FAQ).” http://cwe.mitre.org/about/faq.html#C.6.

MITRE, and SANS Institute. 2010. “CWE - Common Weakness Enumeration.” http://cwe.mitre.org/index.html.

Möllers, F., and S. Seitz. 2014. “Short Paper: Extrapolation and Prediction of User Behaviour from Wireless Home Automation Communication.” *Proceedings of the 2014 ….*

Monson, T. S. 1970. “Thou Art a Teacher Come From God.” In *October Conference Report*, 104–8. Salt Lake City.

National Institute of Standards and Technology. 2014. “Framework for Improving Critical Infrastructure Cybersecurity,” 253.

NCC Group. 2014. “Android-SSL-TrustKiller.” https://www.nccgroup.trust/us/about-us/resources/android-ssl-trustkiller/.

Nicolet College. 2015. “Nicolet College - Smart House Technology Training to Start This Fall.” Accessed February 27. http://www.nicoletcollege.edu/news/2009/homeintegrationtech.html.

OWASP. 2013. “Category:OWASP Top Ten Project - OWASP.” https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project.

———. 2014a. “Application Security Verification Standard (2014).” https://www.owasp.org/index.php/Category:OWASP_Application_Security_Verification_Standard_Project.

———. 2014b. “OWASP Internet of Things Top Ten Project - OWASP.” https://www.owasp.org/index.php/OWASP_Internet_of_Things_Top_Ten_Project.

———. 2015. “OWASP Testing Project - OWASP.” https://www.owasp.org/index.php/OWASP_Testing_Project#tab=Project_About.

PCI Security Standards Council. 2013. “Payment Application Data Security Standard Requirements and Security Assessment Procedures,” no. November: 1–115.

Pivotal Software. 2015. “Spring Framework.” http://projects.spring.io/spring-framework/.

Pratt School of Engineering. 2015. “Smart Home Research Fellows Program | Duke Smart Home Program.” Accessed February 27. http://smarthome.duke.edu/students/opportunities/fellows.

Quirky Inc. 2014. “Wink | Wink HUB.” http://www.wink.com/products/wink-hub/.

———. 2015a. “Wink Security.” http://security.wink.com/.

———. 2015b. “Reconnect to Wink.” http://hubfix.wink.com/recovery.html.

SANS Institute. 2015. “SANS Institute - Critical Security Controls: Guidelines.” Accessed May 26. https://www.sans.org/critical-security-controls/guidelines.

SANS, Institute. 2011. “2011 CWE/SANS Top 25 Most Dangerous Software Errors.” *SANS Institute*. http://cwe.mitre.org/top25/#CWE-78.

Schneider, C. 2013. “Cross-Site WebSocket Hijacking (CSWSH).” Christian Schneider.

Sigma Designs Incorporated. 2014. “Dev Kits - Z-Wave - Sigma Designs.” http://z-wave.sigmadesigns.com/dev_kits.

Smith, C., and D. Miessler. 2014. *Internet of Things Research Study*.

SPF Council. 2004. “SPF: Project Overview.” http://www.openspf.org/.

Survey, The I S O. 2006. “The ISO Survey – 2005.” *Analysis*.

“Technical Standard.” 2015. *Wikipedia*. Accessed May 28. https://en.wikipedia.org/wiki/Technical_standard.

Tillman, K. 2013. “How Many Internet Connections Are in the World? Right. Now.” http://blogs.cisco.com/news/cisco-connections-counter.

VMware. 2015. “VMware Security Hardening Guides.” Accessed May 25. https://www.vmware.com/security/hardening-guides.

Weber, R. H. 2010. “Internet of Things – New Security and Privacy Challenges.” *Computer Law & Security Review* 26 (1). Elsevier Ltd: 23–30. doi:10.1016/j.clsr.2009.11.008.

Wright, J. D. 2011. “KillerBee: Practical ZigBee Exploitation Framework.” http://www.willhackforsushi.com/presentations/toorcon11-wright.pdf.

———. 2015. *Dissenting Statement of Commissioner Joshua D. Wright - Issuance of The Internet of Things: Privacy and Security in a Connected World Staff Report*.

Xipiter. 2014. “The Insecurity Of Things.” http://www.xipiter.com/musings/category/the-insecurity-of-things.

Zhang, L., H. Leung, and K. Chan. 2008. “Information Fusion Based Smart Home Control System and Its Application.” *IEEE Transactions on Consumer Electronics* 54 (3): 1157–65. doi:10.1109/TCE.2008.4637601.

APPENDIX A: SUPPLEMENTARY MATERIALS

The Smart Home Extensions, with references and notes, can be seen in the body of the paper as figures 1 and 2. A Microsoft Excel format spreadsheet of the Extensions is also in a compressed file which can be obtained from any of the links at the bottom of this page.

The raw testing data is contained in four Microsoft Excel format spreadsheets. These are also included in the compressed file. There is one spreadsheet for each device, and each spreadsheet contains separate tabs or worksheets for each category of the ASVS. For a quick reference of vulnerabilities discovered organized by hub, please refer to table 10 in section 7.2.

A clean version of the tracking spreadsheet I used is also available in the compressed file. The spreadsheet contains separate tabs or worksheets for each category of the ASVS and the cover page calculates percentage passed and generates the circular graph automatically. As mentioned previously, this spreadsheet is based heavily on one made available by Florent Batard (Batard 2014).

Finally, the 2014 version of the OWASP Application Security Verification Standard, upon which the Smart Home Extensions are based, is also included for in the compressed file for completeness.

- https://cybersecurity.byu.edu/sites/default/files/supplement.zip
- https://cybersecurity.byu.edu/projects/smarthome
- http://www.redsteve.com/smarthome