DOI: 10.3969/j.issn.1673-5188.2015.03.002
http://www.cnki.net/kcms/detail/34.1294.TN.20150908.0928.002.html, published online August 12, 2015

A Solution-Based Analysis of Attack Vectors on Smart Home Systems

Andreas Brauchli and Depeng Li
(Department of Information and Computer Sciences, University of Hawaii at Manoa, HI 96822, USA)

Abstract

The development and wider adoption of smart home technology also created an increased requirement for safe and secure smart home environments with guaranteed privacy constraints. In this paper, a short survey of privacy and security in the more broad smart-world context is first presented. The main contribution is then to analyze and rank attack vectors or entry points into a smart home system and propose solutions to remedy or diminish the risk of compromised security or privacy. Further, the usability impacts resulting from the proposed solutions are evaluated. The smart home system used for the analysis in this paper is a digitalSTROM installation, a home-automation solution that is quickly gaining popularity in central Europe, the findings, however, aim to be as solution independent as possible.

Keywords

digitalSTROM; smart home systems (SHS); digitalSTROM server (dSS)

1 Introduction

As welfare increases and technological gadgets become ubiquitous, we lighten our daily lives by automating trivial and common tasks. A clear trend of automation technology usage within both personal homes and commercial buildings has been shown over the past few years. The increasing adoption of Smart Home Systems (SHS) leads to the need for not only more functionality but also for a safe, secure and function environment. The ongoing battle for smart grid security [1] includes smart homes [2]. When one technology becomes particularly wide spread it automatically creates a high-reward target type. Several companies offer products on the market to automate lighting, shades, heating, cooling etc. Among the many systems which feature different wired or wireless topologies is digitalSTROM (dS) with its powerline based bus and embedded central server. This research is dedicated to finding security and privacy weaknesses in SHS on the example of a dS system. Wherever possible we try to approach the problem in a generic way that can also be applied to other systems [3].

This works is organized as follows: We begin with this introduction and proceed with a review of how smart homes fit into the broader smart world context and present related work. In the fourth section the dS environment is covered before listing possible attack vectors on SHS in the fifth section along with two example attacks on the dS infrastructure. In section six, solutions to prevent or diminish those attack vectors are proposed and discussed. In section seven, we analyze the proposed solutions which are followed by the conclusions.

2 Smart World

With our world growing "smarter" than ever, there are different ways of integrating smart homes into the broader context of smart services, smart grids or even smart cities. Researchers of different fields have amply been studying this ongoing trend and come up with interesting and useful applications. We briefly present some of those to emphasize the security and privacy needs of a modern SHS, especially in the light of a majority of consumers being agnostic of technology and not necessarily trained in computer and network security. They may thus not be fully aware of what privacy invasions must be expected when certain sensory data is leaked or revealed from their smart environment.

In [4], the authors define smart communities as interconnected sets of co-located homes that share certain common processing infrastructure. The authors give an example of a distributed intrusion detection/aversion scheme based on surveilance data from multiple homes that is processed centrally in the community and an example of smart health care where neighbors are alerted when a critical health situation is detected. A call center responsible for multiple smart communities for emergencies or further assistance is introduced in this paper. While the au-

thors propose a centralized processing of data for privacy reasons, it is likely that not every smart community will want to maintain a data center on its premises. In [6], the authors predict a trend towards more artificial intelligence and thus processing power in future smart homes. The paper foresees that with the increasing number of sensors and readings, a single smart home might not be able to process all data and thus processes them in a cloud environment. Privacy is listed as a potential issue. In a similar light, a framework [5] is proposed to integrate smart homes into Platform as a Service clouds. Data privacy is supposedly managed by the user but the decision on which data to use and process seems to take place post transmission in the cloud. The cloud interface provides additional services or virtual smart home devices provided by third parties. Further improvements in the broad context of lifestyle are presented in [7] where Ambient Intelligence is mentioned. In this paper, we predicts how smart devices will carry an individual's preferences who will then experience personalized results in places like museums and other public places.

Overall, the trend is clearly geared towards a highly interconnected smart world where data is processed in a distributed fashion and the line between private data sharing, such as highly sensitive and individual medical records, and beneficial services is at risk of becoming increasingly blurred. In fact, both might not even remain separable due to design, marketing or infrastructural decisions. The problem is further amplified when individuals may not have a choice anymore in what part of the collected bulk sensor data is shared or even transmitted over a potentially insecure network. It is thus crucial to set the bar high for both security and privacy. Even when individuals do explicitly consent to data sharing the actual transmission protocol must always be open and reviewable for potential leaks (Ideally, the protocols are independently audited and published with unredacted raw data to the general public). Accountability is the key to gain the user's trust and, once obtained, can only be beneficial to the product's success. Since smart home installations have a comparatively long life time of several years or decades and process sensitive sensory information, interested parties will likely take their time to evaluate and research their options. Open source approaches are, in general, very favorable towards trustworthiness and, possibly, also towards the longevity of a product when the modifications and extensions can be installed by the owner/user without requiring specific tools or requiring digital signatures. Unfortunately the shift towards more open protocols is slow and customers might not always see the benefits of open solutions. A change is only expected to happen when demanded by a majority of customers or with comparably successful open solutions.

3 Related Work

This section lists related security research in the smart home context and explains the differences to this work. We conclude that there has been no previous security assessment of this kind on smart home environments with a wired power line bus type and, particularly, not for the dS architecture. The journal article [8] surveys available SHS technology but only briefly lists potential attack vectors on the SHS control infrastructure (DDoS). It also details personal security, i.e. not software system related security, automation logic proposals such as notifying emergency services when a fire is detected, unusual user behavior detection using neural networks and a privacy guard to protect against sensitive information leakage. The paper [9] covers the detect and prevent approach to several security issues in Wireless Sensor Networks in the SHS context. Several attack vectors that compromise confidentiality, integrity and availability are shared in this paper. In contrast, we analyze security issues on the example of dS products which uses a wired bus system with non-factory-default and optional wireless connectivity.

A meter reporting system [10] based on public key encryption that doesn't reveal specific power usage to the utility company is proposed. The system is based on signed readings by a trusted reader. The processing then directly applies the matching price tariff to those readings resulting in a fully verifiable bill without specific usage information. In this paper, we create a good solution to verifiably aggregate metering data but requires a trusted meter by the utility company, which the dS environment does not target or provide. A framework [11] for evaluating security risks associated with technologies used at home is proposed. The paper also associates high level attacker goals such as extortion or blackmail to low-level attacks compromising the infrastructure. We focus solely on low-level security issues and leave out inferring the potential consequences. In [3], the authors present a deep literature review of smart homes and provide a prediction of future development going towards integrated health care systems. Due to the amount of time that people spend in their homes, there is a large economic potential for integrated services. Additionally, the paper includes a section of papers dedicated to security. dS does not appear in any of the papers, however, some wired systems such as KNX are listed.

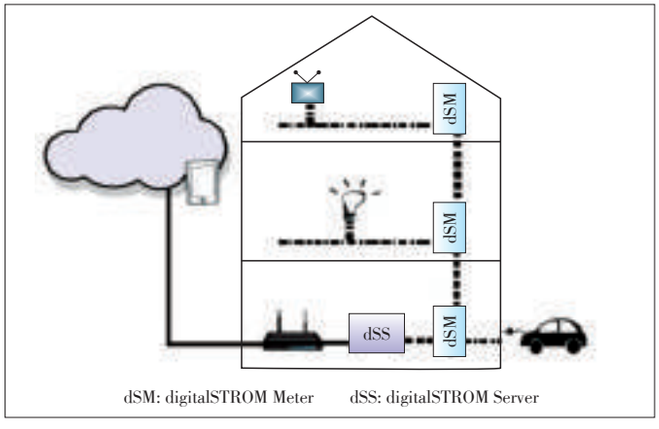
4 The digitalSTROM Environment

The digitalSTROM environment is a SHS designed primarily for personal home use. It can also be simultaneously used in multiple apartments of a building, whereas each apartment has its own installation. The installation consists of one (optional) digitalSTROM Server (dSS), usually one digitalSTROM Meter (dSM) and one digitalSTROM Filter (dSF) per circuit and numerous terminal blocks (small clamps) with a digitalSTROM chip (dSC) for each device. The dSF is responsible for filtering out dS messages on the power bus from and prevent them from reaching the outside world. It is technically required when multiple dS installations are present nearby to prevent crosstalk.

Each dSM can handle up to 128 clamps and communicates with the other dSM and the dSS by use of the ds485 (is an analogy to the serial RS485 bus protocol.) two-wire protocol. The ds485 bus can span up to 100 m but is usually confined within the cabinet (dashed line in **Fig. 1**). DSC are conventionally integrated in a terminal block ("clamp") that, in turn, is connected directly to a power switch or an appliance. The DSC can also be integrated directly into an appliance, into a power socket or a socket list by a licensed manufacturer. The appliances communicate over the power wire by use of a proprietary closed protocol (dash-dotted line in Fig. 1). The bandwidth available to dS devices is very limited with100 bauds (dSM→dSC) / 400 bauds (dSC→dSM) [12]. The reaction time for events is between 250 and 750 ms. Fig. 1 shows a simplified SHS consisting of three separate power circuits (one per floor), two dS appliances (TV, light on the dash-dotted line) and a non-dS charging electric vehicle on an outdoor plug. The dSM are interconnected (dashed lines) with the dSS by the 2 wire bus. The dSS is connected to the home network, symbolized by the wireless router, by a Cat.5 cable or, optionally, by a supported wireless USB dongle. A control device (typically a smart phone or tablet) is connected to the home network with the wireless network. The dSS provides a web interface for configuration and an AJAX/JSON Application Programmable Interface (API) for control.

5 Attack Vectors on SHS

We grouped the possible SHS attack vectors into five vulnerability categories which are detailed in this section: Wired SHS commonly use (1) a server for state management and to provide a control interface or API, (2) a bus for communication with the appliances and (3) a small clamp or control-device for switching individual appliances. This system is ultimately controlled by the user with (4) a control-device such as a smart phone. Additionally, (5) remote third party services may be contracted to extend the system's core functionality. The categories and their communicative interaction are visualized in **Fig. 2**.

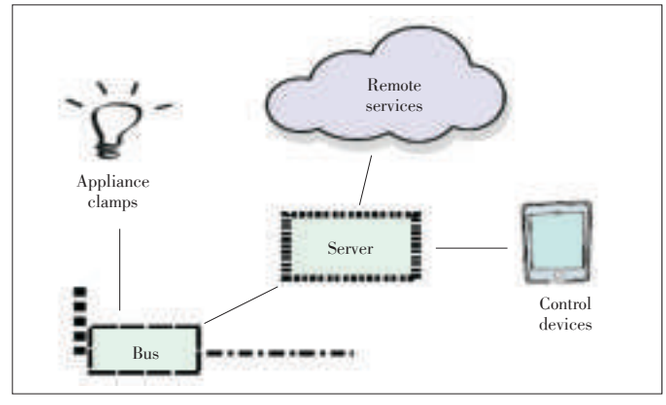


▲**Figure 1. A sample digitalSTROM SHS.**

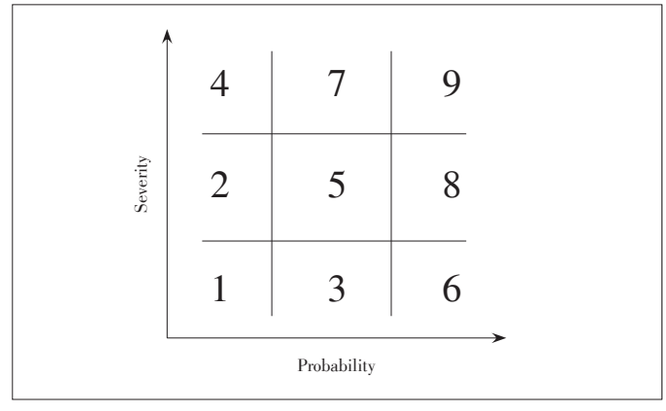
We divided the attacks into the nine relative and perceived risk categories: low, medium and high in each of the two dimensions severity and probability shown in **Fig. 3**. The risk is based on how likely and severe a given attack is. We note that more probable attacks are assigned higher risk ratings than more severe ones.

5.1 Central digitalSTROM Server

This first subsection elaborates on the possibilities to gain access to the central dS server as a mean to compromise the entire SHS. The central server has total access privileges to the SHS: It can switch appliances, read out metering values, manages API connections on the home network and runs virtually permanently. The server is thus the most crucial component to secure within the SHS. Due to the many interfaces it is also the most exposed part. This server role is assumed by the dSS component and is located in the cabinet. In dS systems, the location is dictated by the proximity to the dSM circuit meters. The dSS is an embedded Linux platform with 400 Mhz ARM9 CPU, 64 Mb ram, 1 Gb flash memory, two USB ports and an RJ45 100 Mbit Ethernet port. It features an on-board RS-232 serial port for recovery purposes [13]. The first possibility to attack the dSS is to gain physical access and compromising the root



▲**Figure 2. The risk categories.**



▲**Figure 3. The nine risk categories.**

system password. This can be done using the debug ports to gain access to the serial console and thus the (uBoot) boot-loader. Earlier versions of the dSS featured only 256 Mb flash storage but used an SD card as main storage drive which adds the possibility of maliciously switching SD cards to one with added or modified credentials. Due to the high impact but local constraint (physical access required), this attack is rated at risk level four. The second possibility is to gain access to the local wired or, if available, wireless network and (1) exploit a system vulnerability (e.g. TCP/IP vulnerability in the

Linux IP stack or network driver both LAN or WLAN if a WLAN dongle is plugged to the dSS), or (2) exploit a service vulnerability of a service running with system privileges, e.g. ssh server (Dropbear), if enabled. We note at this point that the dss process handling dS events does not run with elevated privileges. Alternatively an attacker can (3) exploit an API vulnerability within the dSS process. This attack is generally locally bound to the home network and wireless range but weak router/firewall rules may directly expose the dSS to the Internet and thus pose a major potential flaw. Since home automation systems are long-term systems with expected run times of 10–15 years, the software is highly likely to become outdated and unmaintained during its life cycle thus greatly increasing the risk. Due to the high potential severity we assign these two vectors the risk rating seven. Third, an attacker may target the server via the dS485 bus interconnecting the dSM by (1) directly gaining wire access, (2) indirectly by a rogue dSC that injects events that trigger a given message by the dSM on this bus. This attack is judged as having a medium-impact due to the ability to control the whole static SHS, i.e. the functionality of the SHS available when no dSS is installed, with low probability. Besides the impact on the powerline bus, it is questionable whether such an attack would be able to compromise the dSS integrity and would have to be determined by a code analysis of the dS485 bus handler process. We thus assign this attack vector the risk level two. The fourth attack possibility is to redirect or abuse the app store to (1) inject rogue updates with open backdoors, this is possible because updates are not digitally verified, or (2) rogue apps may be installed either by mistake or by misguiding the user into installing them. As dS apps do not have system privileges because they are run from within the dss process and are restricted to a JavaScript sandbox, the main threat is to privacy, as all events can be triggered and registered. Both rogue updates and apps can be installed when the attacker has control over the local network and can intercept and modify the home network traffic from and to the dSS as the updates are served through an unencrypted HTTP connection. Without local access, it is very hard to manipulate network traffic, however, due to the high impact of a compromised update, this attack vector is assigned the risk level four. When considering rogue apps, we increase the risk to level five due to the higher probability of such an attack but lower severity: tricking a user into installing a rogue app is possible but depends highly on the victim.

5.2 Smart Control Devices

This subsection describes how a compromised Smart Control Device (SCD) such as a smart phone or control station leads to a compromised SHS.

Besides the wall switches in rooms, control of the SHS is generally delegated to trusted and/or authenticated control devices such as smart phones or control terminals. In the dS case, the JSON-API is only accessible by a secure HTTPS connection and requires a token that is obtained after successful authentication. However, if a control device such as an Android or iPhone smart phone is compromised, the control of the whole system, as far as API support reaches, is consequently compromised until the token is revoked or expires, in case the device doesn't store the actual credentials. DS does not currently feature specific (usually wall-mounted) control terminals, thus this scenario is omitted. DS published both an iOS and Android app. Since smart phones are mostly connected to the Internet they are exposed to many third party apps and, possibly, viruses or worms. Additionally the device usually has full access to the home network. These facts lead to a high-risk attack vector with risk category nine.

5.3 Smart Home Communication Bus

In this subsection we analyze the risks of a compromised communication bus. The implications of which directly lead to a largely compromised SHS. DS uses an proprietary but unencrypted protocol for its communication on the power wiring (powerline) [12]. As the protocol uses neither encryption nor authentication, any received messages are assumed to be valid. This opens the possibility for (1) injecting control signals to directly control appliances or disrupt the system, or (2) inject invalid power readings to falsify the report system power consumption. When falsifying consumption readings, this only falsifies the reading of individual single devices as the dSM is aware of the total sub-circuit consumption independently of any attached dSC. Having access to the communication bus allows easy jamming of the SHS thus creating a Denial of Service (DoS) type attack. The low bus bandwidth makes this attack particularly effective. The attacker has the choice of jamming only the sub-circuit with the attached rogue sender device or the whole system by continuously sending systemwide events, such as alarms, that are then broadcast by the dSM into the adjacent sub-circuits. As all dS appliances have access to the powerline bus and thus have full control of the bus within its sub-circuit, an attacker may attach a rogue appliance anywhere in the system. If the attacker does not have physical access, he may still trick someone who does into plugging in an appliance for him, for instance by gifting or lending such a prepared appliance. DS appliances can be anything from a lamp to a TV or computer. As dSC clamps are relatively small and only draw minimal power, they are easily hidden inside an ap-

pliance case. An alternative and limited attack consists of connecting unmodified original dS clamps to the system which automatically registers and adds the device, an automatic Plug-n-Play (PnP) procedure which takes less than 10 minutes. Once registered, the device is ready for use: e.g. a clamp with a yellow color code4 switches all room lights in the room it's plugged in. A generic panic button will trigger the panic procedure which defaults to turning on all lights and opening all shades and blinds in the entire installation. With the locally limited exposure of the powerline bus, generally secure premises (except for outdoor plugs) but with high control level, this attack vector is rated a risk category four (private home without outdoor plugs) or seven (with easily reachable outdoor plug or when the SHS is a semi-public environment such as an office space.) An alternative point of entry is the ds485 bus interconnecting the dSM and dSS. The implications are the same as compromising the powerline bus with an additional small but unverified possibility of exploiting the dSS's process by buffer overflow. This attack does not seem very likely or attractive as dSM are usually located next to the more rewarding dSS. We thus assign the risk category four.

5.4 Remote Third Party Services

This subsection analyzes the trust implications of connecting third party services with the SHS. A third party service provides additional functionality to the SHS. Those services can be classified into two categories: (1) monitoring services and (2) control delegation services. A service can also be classified in both categories simultaneously. The monitoring services accept consumption statistics, system events or other collected data and provide a suggestive or analytical service based on the data interpretation. As such this type of service imposes purely a privacy risk as identifying events such as home presence and activities may be leaked [18]. We rank this attack vector at risk level three but the actual danger could highly vary depending on the nature of the leaked information and the danger that such a leak could go unnoticed for a very long time. The second category of services requires control permissions and thus API access by token which may be revoked individually [18]. Such services may for instance provide an alternative Internet-based user interface. By consequence, a compromised third-party service directly implicates a compromised SHS and carries an elevated risk rated at seven or nine, depending on how secure and trustworthy the third party service is. DS offers such a service called "mein.digitalSTROM" [19] using a dS app which allows remotely controlling the installation. It also allows temporary control delegation with a time expiring link and backs up local configuration and metering data. It is inevitable that all third-party services be trusted with private data and system control respectively.

5.5 Two Attack Scenarios

In this sub-section, we elaborate on two theoretical attack scenarios based on our previous analysis. The first attack uses the dS Android smart-phone app [14] as entry vector and switches lights on at night when the home owners are sleeping. The second attack uploads power readings to a remote server, allowing the attacker to know when the home is empty or will likely be. The first attack is created by installing a rogue app on the home owner's Android smart phone. This app poses as totally unrelated app to the SHS. Once the app is installed on the SHS owner's smart-phone, it launches a background service that sends an Android intent [16], a cross app message using the dS app's public interface, to the dS app sometime during the night. The unmodified and unknowing dS app then performs the action using the stored credentials. The malicious apps does not need to know any connection details or the API token. While the attack may sound banal, more frightening scenarios can be envisaged. In the second attack, the dS app is user-installed on the dSS using the official dS app-store. Once installed, the app collects consumption data from all connected dSM and periodically uploads them to a remote location. The attacker uses this collected data to establish when the residence is likely to be empty. We do note that third party apps will likely have to pass a code review before being entered into the dS app-store. There are enough legitimate uses for sending private data and the app should thus pass a code inspection based on different expectations by the reviewer and the app's user, especially if the documentation is ambiguous, suggestive or simply missing.

6 SHS Hardening

This section is modeled after the previous chapter: It is organized into central dS Server, Smart Control devices, Smart Home Communication Bus and Third Party Services. In an effort to harden SHS against the attacks described in the previous section, we recommend adopting proven strategies from other domains. In addition to providing security-enhancing suggestions, we reflect on the usability impact of the proposed solutions.

6.1 Central digitalSTROM Server

This subsection reiterates the crucial role of the central dS Server in the overall system security. Because of its central role and exposure to different interfaces in the SHS, a physical server breach is rated at both the highest severity and highest probability. To protect against physical server breaches, the easiest and, at the same time, most effective method is, arguably, to lock the cabinet if it is located in a (semi-) public space. This should be recommended to every customer through the installation documentation. This solution

has a low usability impact and leaves the choice and risk assessment to the customer. Within private spaces the risk of a physically compromised dSS is rated low. If additional security is desired, one could make use of a tamper-evident case which

may avert certain attackers. This change requires a customer to be aware of how to check the integrity seal, which could possibly be done remotely, but does still require a lock secured cabinet. A tamper-evident case incurs a high usability impact due to needed additional training. To protect against network-based attacks on the dSS it is important to make the user change the default access password, preferably during the initial setup. A default access password together with an open network results in a very high probability and high severity risk. Usability is only minimally impacted by requesting the user to set a password on setup. The initial setup could be streamlined by a setup wizard which would cover this step. To prevent Man-in-the-Middle (MitM) attacks such as modifying system or app updates, dS update servers should default to an encrypted HTTPS connection with a valid SSL certificate. Such a secure connection is transparent to the user and does thus not incur any usability changes. To reduce the risk of a totally compromised SHS the introduction of a permission based access control system for the API is suggested. Possible permissions include reading out meter values, controllable dSMs/rooms such that an application may be restricted to controlling appliances in one sub-circuit or even individual appliances, the events that can be triggered and the events that one can register with. This list is not exhaustive and further permissions may be applicable. There is a certain trade-off between usability and permission-configurability as analyzed by [15], however, the impact could be lessened by allowing full permissions by default and leaving the specific constraining to knowledgeable users within the "advanced settings" menu option.

6.2 Smart Control Devices

SCD have full control over the SHS. Thus it is crucial to educate all users that a compromised SCD implies a compromised SHS. The dS app for Android provides other apps on the smart phone with the possibility to send intents (Android control messages) that the app will then react upon. Thus any app on such a smart-phone can control the SHS. We propose adding a white-list of registered apps, managed by the user, to the Android dS app to verify that a certain app is allowed to control the SHS. The list would be updated on the fly upon first request as to incur usability only minimally. Users may also feel more secure when they know which apps can, or are trying to, control their SHS.

6.3 Smart Home Communication Bus

DS uses a proprietary protocol for communication between dSC and dSM. The technology does not permit inter-dSC communication without going through a dSM first due to separate up and downstream channels. If one were to reverse engineer the communication protocol and implement a device speaking the protocol or a reverse engineering a dSC's interface/firmware, an attacker could easily inject messages or jam the circuit and installation and create a DoS attack. We thus strongly recommend investigating adding an encryption layer such as [17] targeting low power and very low overhead settings. An encryption layer may incur a moderate overhead in usability if keys have to be set up by the user. We further suggest adding an option to disable the PnP functionality for automatically registering new devices. Especially in semiprivate environments such as offices where power plugs are readily available to everyone having physical access. For ease of use we do not suggest disabling PnP by default, but when the auto-registration function has been disabled, we suggest adding a timer-based enable function—analogous to how bluetooth pairing works that allows auto-registering appliances that are plugged in during a short time frame. The usability impact of such a feature is minimal resulting in only one more option which could be placed within the advance configuration mode.

6.4 Remote Third Party Services

Remote services provide additional functionality to the SHS by either providing remote access to the dSS or by analyzing and reporting on collected data. To harden the system against privacy leaks, we suggest implementing configurable time-resolution limit permissions to the already proposed permission system. Such a resolution limit would e.g. not allow access to resolutions below a 15 minute-aggregation in order to maximize privacy. As such a restriction is optional, the usability impact remains small while giving the user a much greater sense of privacy. To harden against compromised third party services, a restricted set of permissions should be applied to remote controlled API accesses, additionally all API accesses and transactions should be logged for a future audit. As the user is responsible for checking the logs, he does incur a great usability impairment unless combined with a method of automatically checking logs for irregularities. A Third party app should only be accepted into the dS appstore when sufficient, clear and unambiguous documentation is available as to what data is being processed, sent off remotely and what control events are raised by the app. The code reviewers are responsible for checking the code paths against the documentation and ask for corrections before accepting it. Before installing an app, a user should have the possibility to accept or reject the requested functionality. There is a minimal usability overhead to display the app documentation which have to be manually accepted or rejected by the user.

7 Solution Analysis

We now look back on the sample attacks in the light of the suggested improvements and find that the attacks would not be possible anymore. We do note that all proposed solutions are

theoretical improvements based on the research and experience in related fields. The physical experimentation of the suggested solutions in this exact context is left as future work item. The first attack scenario uses the dS Android app to

stealthily inject control events into the SHS. With a whitelist of apps that are allowed to send control events through the dS Android app, any app on the smart phone would have to request permission before being granted access, thus thwarting a stealthy attack. A visual clue should make it apparent that

said app, which has nothing to do with SHS, pursues a malicious purpose when it seeks access to the SHS via the exposed Android intent.

The second app that sends consumption events to a remote server would have to declare the intent to send readings to a remote service in the documentation and request those specific permissions during the installation. If this is against the purpose of the app, the user should recognize the threat and choose not to install the app. After an implementation of our proposed solutions, both sample attacks would thus not be possible anymore.

8 Conclusion

We conclude this paper by reiterating that homes are very intimate places where people expect and deserve a high level of privacy and security at a level that is currently not

satisfyingly offered by the feature-driven industry. We have elaborated different attack vectors on a dS SHS which range from physical breaches, over networked attacks all the way to third party remote issues. We demonstrated actual abuse of two of those attack vectors and suggested various improvements to all of the pointed out attack vectors along with possible usability impairments resulting from the solutions. We hope that this research will lead to an increase of openness and security awareness from the early development process on in both generic SHS products and particularly to an improved digitalSTROM system.

References

[1] European Union Agency for Network and Information Security ENISA. *Smart grid security recommendations* [Online]. Available: http://www.enisa.europa.eu 2012

[2] National Institute of Standards and Technology NIST. *NISTIR 7628 guidelines for smart grid cyber security* [Online]. Available: http:www.nist.gov/index.html

[3] M. R. Alam, M. B. I. Reaz, and M. A. M. Ali, "A review of smart homes-past, present, and future," *IEEE Transactions on System, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 42, no. 6, pp. 1190–1203, Nov. 2012. doi: 10.1109/TSMCC.2012.2189204.

[4] X. Li, R. X. Lu, X. H. Liang, X. M. Shen, J. M. Chen, and X. D. Lin, "Smart community: an Internet of Things Application", *IEEE Communications Magazine*, no. 49, pp. 68–75, 2011.

[5] B. Eom , C. Lee, C. Yoon, H. Lee, and W. Ryu, "A platform as a service for smart home", *IJFCC*, vol. 2, no. 3, pp.253–257, 2013.

[6] D. Cook, "How smart is your home?" *Science* , vol. 335, no. 6076, pp.1579-1581, March 2012.

[7] M. O'Grady and G. O'Hare, "How smart is your city?" *Science*, vol. 335, no. 6076, pp. 1581–1582, March 2012.

[8] R. Robles and T. Kim. *A Review on Security in Smart Home Development* [Online]. Available: http://www.sersc.org/journals/IJAST/vol15/2.pdf

[9] K. Islam, W. Sheng, and X. Wang, "Security and privacy considerations for wireless sensor networks in smart home environments," *2012 IEEE 16th International Conference* on *Computer Supported Cooperative Work in Design (CSCWD)*, Wuhan, China, 2012, pp.626-633.doi: 10.1109/CSCWD.2012.6221884.

[10] A. Rial and G. Danezis, "Privacy-preserving smart metering," *Proceedings of the 10th annual ACM Workshop on Privacy in the Electronic Society (ACM WPES11)*, NJ, USA, pp. 49–60. 2011.

[11] T. Denning, H. M. Kohno, and T. Leving, "Computer security and the modern home," *Communications of the ACM*, vol. 56, no. 1, pp. 94–103, 2013.

[12] Aizo AG. *digitalSTROM FAQ* [Online]. Available: http://www.aizo.com/de/support/documents/A0818D044V004 FAQ.pdf

[13] Aizo AG. *dSS 11 Produktinformation* [Online]. Available: http://www.aizo.com/de/support/documents/ digitalSTROMServerdSS11ProduktinformationV1.0.pdf

[14] Google Playstore, Aizo AG. *dS Home Control* [Online]. Available: https://play.google.com/store/apps/details?id=com.aizo.digitalstrom.control

[15] T. H. -J. Kim, L. Bauer, J. Newsome, A. Perrig, and J. Walker. *Challenges in access right assignment for secure home networks* [Online]. Available: https://sparrow.ece.cmu.edu/group/pub/kim_bauer_newsome_perrig_walker_hotsec10.pdf

[16] Google Ltd. *Android API Reference* [Online]. Available: http://developer.android.com/reference/android/content/Intent.html

[17] M. Luk, G. Mezzour, A. Perrig, and V. Gligor, "MiniSec: a secure sensor network communication architecture," *Proceeding in Sensor Network (IPSN07)*, Massachusetts, USA, 2007, pp. 479-488.

[18] I. Rouf, H. Mustafa, M. Xu, W. Xu, R. Miller, and M. Gruteser. *Neighborhood watch: security and privacy analysis of automatic meter reading systems* [Online]. Available: http://www.winlab.rutgers.edu/~gruteser/papers/fp023-roufPS.pdf

[19] Aizo AG. *digitalSTROM Installation Manual* [Online]. Available: http://www.aizo.com/de/support/documents/html/digitalSTROM Installationshandbuch A1121D002V010 EN 2013-11-12/ index.html#page/digitalSTROM% 2520Installationshandbuch/digitalSTROM% 2520Installationshandbuch A1121D002V010 EN 12-11-2013 Final.1.56.html

Manuscript received: 2015-04-26

Biographies

Andreas Brauchli (andreasb@hawaii.edu) received his MS in Computer Sciences from the University of Hawaii at Manoa, USA (UHM) and his B.Sc. C. S. from the Federal Institute of Technology in Zurich, Switzerland (ETHZ). He is employed as Mobile Software Engineer at Sensirion and previously interned at AIZO where he worked on digitalSTROM smart home applications. His research position at the University of Hawaii focused on privacy and security in multiple domains, amongst others, he worked on the AllNet delay tolerant secure networking protocol. His interests lay in the fields of Open Source Software and the vast areas of security and privacy around the digitally connected life.

Depeng Li (depengli@hawaii.edu) obtained his PhD in computer science from Dalhousie University, Canada. He received his BS and Master Degree in Computer Science from Shandong University, Jinan, China. He is currently an Assistant Professor in Department of Information and Computer Sciences (ICS) at University of Hawaii at Manoa (UHM). Before that, he had been worked in RIM (Blackberry) in Ottawa, Canada and Microsoft, Redmond, US to release Blackberry smart phone and Windows 7, respectively. He had also been worked as Post-Doc researcher for MIT and Masdar Institute cyber security project. His research interests are in security, privacy, and applied cryptography. His research projects span across areas such as Internet of Things, smart grids, mobile Health-tech, aerospace safety and physical-human-cyber triad. He had published around 40 papers at some famous journals and conferences.