The Current State of Security in Smart Home Systems

Threats in the Internet of Things

Version: V1.0 | Date: 2016-01-25
Author: | Responsible: Daniel Schwarz, SEC Consult Vulnerability Lab - Vienna
Confidentiality Class: Public

Abstract

This paper considers smart home systems as currently one of the most popular application fields regarding the “Internet of Things”. It examines what a smart home system is, how it is constructed and which protocols are used to communicate between the components themselves and their users. Furthermore, common threats to different sub-areas of smart home systems are discussed and the most popular communication protocols as well as their current state of security are presented.

Contents

1 The Internet of Things and Smart Home Systems

Smart Home, Smart Metering, Smart Grid, Industry 4.0, Connected Cars… or just, the Internet of Things.

All these words are very popular buzzwords nowadays, but what is really behind them? Increasingly devices are equipped with computer chips to enable a connection between them in order to make our lives more comfortable. However, are there no disadvantages? Are there no threats? How is the status of security in this area?

This paper looks at smart home systems as one possible application of the Internet of Things. It briefly discusses what a smart home system is and how its architecture looks like. Then it presents common threats and the security status of popular smart home protocols.

1.1 What is a Smart Home System

When talking about smart home systems, it all comes down to a few basic buzzwords, as shown in Figure 1.

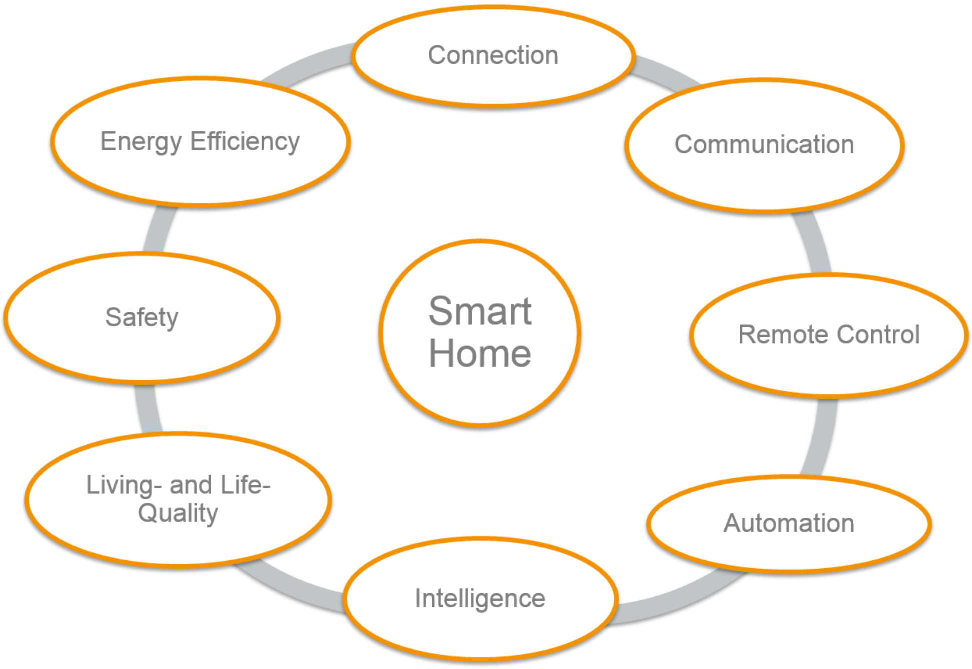

Figure 1: Smart Home Buzzwords

Based on these buzzwords, a smart home system can be defined as a network of connected components in the living area, which communicate with each other and the inhabitants in order to raise their living- and life-quality, their safety and the efficiency of their energy usage. This is done by providing the availability for remote control and automation as well as a certain degree of self-intelligence. [1]

However, this definition is quite theoretical. In practice, a smart home system is a little computer that one puts into his house, connects it with the home router and connects every imaginable device to it. After that it is possible to control the devices e.g. via the smartphone or a web-browser. Sounds great, doesn't it?

Such a setup could be used within many different areas. The most common fields of application are the following:

- **Light control/automation**
  - E.g. based on the presence and preferences of the inhabitants in combination with the current sun exposure.
- **Intelligent white goods**
  - E.g. the washing machine starts exactly when the energy price is at the lowest level of the day or the fridge automatically orders missing products and gives advice, which meals can be prepared with the available products.
- **Entertainment**
  - All media-content is accessible from every device.
- **Alarm System**
  - The complete alarm system with all the necessary sensors could be integrated and controlled centrally.
- **HVAC Systems**
  - Climate control could work highly efficient due to a big basis of information gathered by different sensors.
- **Assisted Living**
  - Everyday living could be made easier for elder, invalid or physically disabled people by automation of different tasks and monitoring of their vital parameters.

While every single one of these areas is interesting for itself, the real advantage of a smart home system is the combination of everything and the resulting synergy effects. E.g. an infrared sensor recognizes the presence of an inhabitant. If everything is connected and centrally managed, this information could be used as an input for the alarm system, the light control and the heating system. In this way, a few sensors could gather valuable information that could be used in many different ways.

1.2 Architecture of a Smart Home System

Although there exist many different smart home components, they can be reduced to a few basic required elements, which define the architecture of a typical smart home system, as shown in Figure 2. First of all there are sensors that measure the status of the environment and the inhabitants and there are actors that preserve or change this status.

These sensors and actors are all connected to a central base station, which aggregates and controls all the information. Furthermore, the base is connected to some sort of remote control client, usually a web browser or a smartphone app. This connection can either be established directly, e.g. within the same network or, as in most cases of today’s smart home systems, via a cloud-interface provided by the vendor in order to enable remote access to the system.

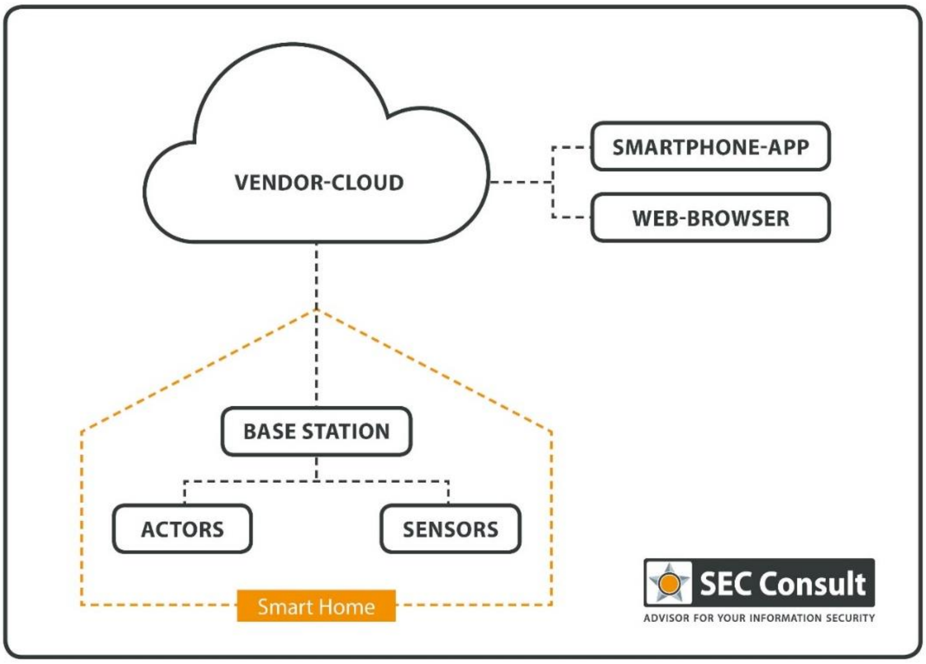

Figure 2: Basic Elements of a Smart Home System

1.3 Protocols in Use

When it comes down to technologies and protocols in use, the architecture mentioned above can clearly be divided into two parts.

The first part is the communication between the central base station and the control client, in most cases realized via the vendor-cloud. Basically, this is the interface between the smart home system and the user. In this area, all traditional web technologies, e.g. HTTP, WebSockets, Web services etc., are used to establish a connection between the base station and a web-browser or smartphone app. In the course of the paper, this area will be referred to as “web area”.

The second part is the communication between the base station and the connected sensors and actors. Many specialized smart home communication protocols exist in this area, which can be radio- or cable-based (except some minor exceptions, which work with both). Some of these protocols are vendor-proprietary, some of them are standardized. In the course of the paper, this area will be referred to as “local area”.

Based on a recent market analysis [1], the following protocols can be seen as the most common, at least in Central Europe:

Medium	Protocol
Radio	Z-Wave
	ZigBee
	EnOcean
	BidCos (HomeMatic)
Cable	KNX

2 Threats to Smart Home Systems

A smart home system directly connects our virtual world with our real life. Therefore, a smart home system has a huge impact on our daily life, and so have the associated threats. Therefore, the following assets should be protected by smart home security:

- Life and health
- Property
- Control of
  - access to the home
  - the connected devices
- Information

In order to protect these assets, both, the local area and the web area must be considered.

2.1 Threats to the Web Area

Most of the currently available information regarding smart home security focuses on the local area, as all the new smart home communication protocols are located there. Nevertheless, it is also very important to consider the web area, because even good local security mechanisms can be circumvented by relatively simple and traditional web vulnerabilities. SEC Consult proved in a previous advisory [2], [3] as well as in a recent blogpost [4], that it is possible to take control over a smart home system by exploiting simple web vulnerabilities. Since the complete threat modeling for the web area of smart home systems would go beyond the scope of this paper, some specific mechanisms of smart home systems will be analyzed.

When talking about the web area, there are at least three security-relevant basic functions, which nearly all smart home systems have in common:

1. Provide the basic possibility for a user to interact with the system.
2. Provide the possibility of controlling the system remotely over the Internet.
3. Provide the possibility of updating the system.

2.1.1 User-Interaction

A smart home system needs to provide some mechanism that enables the user to interact with the system. This can be achieved by simple web interfaces or mobile apps, which can be realized by all kinds of traditional web technologies, e.g. HTTP, WebSockets, Web services (SOAP/REST) etc.

In 2014 OWASP released the OWASP Internet of Things Top Ten Project [5], which ranks the ten most important weaknesses of the Internet of Things. “Insecure Web Interface” is ranked as number one, directly followed by “Insufficient Authentication/Authorization”. And “Insecure Mobile Interface” is also ranked on the seventh place.

These categories do not contain any highly sophisticated Internet of Things specific vulnerabilities, but mainly describe all the traditional and well-known web vulnerabilities, like Cross-Site Scripting, Cross-Site Request Forgery, usage of standard credentials etc. Because of the direct relation between a smart home system and a person’s real life, these vulnerabilities can have an enormous impact. In a previous advisory [2], SEC Consult described an attack to a specific smart home system that enables an attacker to control e.g. the alarm system of the victim by simple Cross-Site Scripting and Cross-Site Request Forgery attacks.

It is very important to reduce the attack surface as much as possible in this area. E.g. the smart home system mentioned at this SEC Consult advisory [2] has two mechanisms for user interaction: WebSockets and plain HTTP-Requests. While no weaknesses have been identified in the WebSockets connection, the plain HTTP-Requests contained the previously mentioned vulnerabilities.

Furthermore, it is very important to conduct regular security audits, performed by independent security professionals, in order to identify existing vulnerabilities.

2.1.2 Remote-Control

The really fancy thing about smart home systems is the possibility to control them via the Internet. In order to realize this functionality, there are four possible ways, which are currently used by different vendors:

1. **Port forwarding**: A port of the home router’s public interface is forwarded to the central base station, which therefore is directly exposed to the Internet.
2. **VPN**: The remote control client has to connect to a VPN-Server of the internal network in order to connect to the central base station. On the one hand, this requires a more complex configuration, which could overwhelm an ordinary home user. On the other hand, this procedure reduces the lines of code exposed to the Internet to a minimum set of source code that is specialized to deal with unauthenticated network packets. The smart home base station itself is therefore protected against direct attacks from the Internet.
3. **Cloud Interface**: The central base station has a permanent connection to an online cloud service provided by the vendor. The remote control just connects to this cloud service, which relays the commands to the central base station. This widespread method requires very low effort for the end user, but the vendor potentially has full control of the system and the user completely depends on the vendor. E.g. if the vendor has technical problems or has to quit business because of economic reasons, the remote control may not work anymore. Furthermore, privacy issues have to be considered if a vendor has full access to all the private information processed by a smart home system. In the OWASP Internet of Things Top Ten Project the categories “Privacy Concerns” and “Insecure Cloud Interface” are ranked on places five and six.
4. **Mobile communication**: In this area, two different mechanisms could be observed. The first mechanism is the connection between the remote control and the central base station that can be established via a mobile internet connection, integrated within the base station. From a security perspective, this option is equal to the port forwarding option, as the base station is directly exposed to the Internet. However, as a pure backup connection, it could also increase the security e.g. by providing access to the alarm system even if the regular internet connection fails. The second mechanism is the possibility to control the base station via text messages. This option is not recommended due to known security issues [6], [7] regarding text messages.

While the remote control via cloud interface is the most common on the current market, the secured connection via a VPN-Server is recommended from a pure security perspective.

2.1.3 Updates

Since smart home systems are operated by software, they have to provide a possibility to receive updates, e.g. functionality or security patches. If an attacker is able to manipulate such an update, he could gain full control of the smart home system. There are three possibilities of how updates can be received, all of them are currently used by different vendors:

1. **Configuration software**: A configuration client with an installed configuration software is used to download the update from the online vendor service (or a new version of the configuration software that includes the update is downloaded). Then the update is transferred from the configuration client to the central base station, e.g. via IP/WLAN, radio protocols, serial, USB cable etc. This setup requires at least two connections, one from the vendor to the client and one from the client to the base station.

2. **USB stick**: Another method used by different vendors is to manually download the update from the vendor's homepage, put it on an USB drive and plug it into the base station. In this scenario, the update could again be manipulated while downloading it from the Internet. By allowing updates via an USB drive, two further threats arise. On the one hand, an attacker with physical access to the base station could be able to plug in an USB drive with a manipulated update. On the other hand, the USB drive could contain malware that may attack the smart home system [8].

3. **Direct internet connection**: The last scenario is the most direct one. The base station downloads the update directly from the vendor and installs it automatically (or at least after asking the user for permission). This scenario just requires one single connection and therefore has the least attack surface. Of course, this connection has to be secured via TLS and an appropriate certificate pinning mechanism. Furthermore, it is necessary to digitally sign and verify the update by using strong cryptography.

2.2 Threats to the Local Area

By looking at the local area and the communication protocols in use, the following basic threats and related requirements to mitigate the threats have been identified [1]:

Threat	Mitigation Requirement
Injection of wrong data	Mutual Authentication
Unauthorized information gathering	
Redirection of messages	
Replay of messages	Freshness
Manipulation of messages	Integrity
Communication sniffing	Confidentiality
Gathering metadata	
Denial of Service	Availability

Since jamming of radio communication is a common problem for all radio based protocols and there are publicly known possibilities for protocol based denial of service attacks for most of the protocols mentioned below, these threats will not be discussed further in the following chapter.

3 Current State of Security

This chapter shows the current state of security of the five most common smart home local area communication protocols, which were mentioned in section 1.3.

3.1 Z-Wave

Z-Wave is a radio based communication protocol developed by ZenSys, which is now a part of Sigma Designs, Inc. Currently the protocol is managed and improved by the Z-Wave Alliance. It uses the frequency area between 868.4 MHz and 926.3 MHz with data rates of 9.6 kb/s, 40 kb/s (most common) or 100 kb/s. Since the Z-Wave 400 Series also 2,4 GHz, 200 kb/s and an 128-Bit AES encryption are provided. The specification of the protocol isn’t publicly available. It is just provided for vendors after they sign a non-disclosure agreement.

Z-Wave offers three different security mechanisms [9] in order to prevent the above-mentioned threats - starting with the 400 series chips. Older chips do not incorporate any security features.

Requirement	Mechanism
Confidentiality	AES-128 OFB-Mode
Authenticity & Integrity	CBC-MAC
Freshness	64 Bit Nonce

While this does not sound that bad, there is a problem in the way encryption keys are transferred to new devices.

If a new device joins a Z-Wave network, a hardware based pseudo random generator generates a symmetric key. Then, this key is encrypted with a temporal default key, which is hardcoded in the Z-Wave chip and always consists of 16 bytes with the value 0.

By knowing this, an attacker could sniff the initial device pairing, steal the generated encrypted key and decrypt it with a default key.

The second publicly known problem is an implementation fault of a Z-Wave door lock [9], rather than a general protocol vulnerability. An attacker could pretend to be a Z-Wave controller (central node that coordinates all client nodes) and start the initial key exchange mechanism with the door lock. The door lock automatically accepts this and establishes a connection with the attacker’s controller, even if it is already connected to a real controller. Afterwards, the attacker is e.g. able to open the door lock. This is a typical case of missing mutual authentication.

It is mandatory to bear in mind that all the mentioned security mechanisms are optional and are just provided by Z-Wave chips starting with series 400. Older generations of the protocol are completely unprotected.

3.2 ZigBee

ZigBee is a publicly available, radio based communication standard, developed by the ZigBee Alliance. While the first version of ZigBee was released in 2004, the current version of ZigBee is known as ZigBee2012. The next version, ZigBee 3.0, is still a draft, but according to the vendor it should be released soon [10].

ZigBee consists of four protocol layers. While the application- and network-layer are defined by the ZigBee standard itself, the MAC-layer as well as the physical layer are defined by the publicly available IEEE 802.15.4 standard. ZigBee operates on the frequency bands 868 MHz, 915 MHz and 2,4 GHz, with data rates of 20 kb/s, 40 kb/s and 250 kb/s.

Furthermore, ZigBee defines different application profiles, which describe specific commands, attributes, device-descriptions etc. for different types of applications, e.g. ZigBee Home Automation or ZigBee Smart Energy.

Regarding security, ZigBee implements an algorithm called AES CCM*, which combines an AES-128 encryption in CTR-mode and a CBC-MAC:

Requirement	Mechanism
Confidentiality	AES CCM* (AES-128 CTR-Mode)
Authenticity	AES CCM*
Integrity	AES CCM* (CBC-MAC)
Freshness	Counter

Furthermore, ZigBee has two different security modes, the standard security mode and the high security mode. The biggest difference between these modes is the kind of keys they use. In the standard security mode, just one network key secures the whole communication. This key could be preconfigured at the device by either the vendor or the user, or it is transmitted unencrypted across the network. At the high security mode, network- as well as link keys are used. These keys could also either be preconfigured or they could be established via a challenge-response mechanism, based on a master key. This master key could again be preconfigured or transmitted via the network in plaintext.

As mentioned before, ZigBee defines different application profiles. Among many other things, some of them also define default keys. E.g., the Home Automation profile defines a default key that should be used for securing the transmission of the actual network key, if no other key is preconfigured. Although this should just be a fallback-mechanism, many manufacturers implement it as default behavior. [11]

Another interesting application profile is the ZigBee Light Link profile, which addresses easy-to-use light control for end consumers. In order to propagate the current network key of such a Light Link network, a pre-installed default key is used. This default key is the same for all certified ZigBee Light Link devices and of course, it has already been leaked to the Internet. Furthermore, if this default key should not work, the same fallback default key, as described in the Home Automation profile, is defined for Light Link devices. [11]

While there are some more minor issues and also some theoretical attack scenarios to the security mechanisms themselves, they can be considered as quite good, as long as the key management is implemented in a safe way by the manufacturer. If the keys get stolen, the whole security is compromised.

3.3 BidCos

BidCos is a radio based communication protocol developed by eQ-3. It was designed for eQ-3's popular HomeMatic system and is now supported by different other vendors. It works within the 868 MHz frequency band and its security is based on an optional AES-128 security mechanism.

This AES-128 security mechanism is not about communication-encryption. An AES-based handshake is used to authenticate the sending device. E.g., when the central base station sends a command to a device, the device responds with a challenge. The base station encrypts the challenge with the shared AES-key and sends it back to the device, which could now authenticate the base station. Therefore, the security is solely based on the shared AES-key. [12]

Generally every HomeMatic device has the same default AES-key preinstalled. According to eQ-3, this default key has not been hacked yet [13], but the internet community does not share this opinion since the key itself [14] and an analysis of the handshake algorithm [15] were leaked via pastebin.com.

Of course, the user has the possibility to change the used AES-key on the base station, which transmits it to the connected devices afterwards. However, due to the fact that the AES security mechanism of BidCos is just an authentication handshake, but provides no encryption, it has to be assumed, that the new AES-key is transmitted in plaintext. Therefore, it is not protected against sniffing attacks.

3.4 EnOcean

EnOcean is a radio based communication protocol developed by the EnOcean GmbH, which formed the EnOcean Alliance Inc. with several other companies. Based on this technology the EnOcean Wireless Standard ISO/IEC 14543-3-10 was created. It is optimized for very low power consumption and energy harvesting, which means that most of the devices do not need an external power supply or batteries. They just gather the needed energy from their surroundings, e.g. light, movement or temperature differences. EnOcean can use the frequency bands 868 MHz, 315 MHz, 902 MHz, 928 MHz and 2,4 GHz, with data rates of 125 kb/s.

The EnOcean standard specifies many different optional security mechanisms, which can be combined flexibly [16]:

Requirement	Mechanism
Confidentiality	AES-128 CBC-Mode or VAES[1]
Authenticity	CMAC
Integrity	CMAC
Freshness	Rolling Code

While there are no publicly known vulnerabilities in the EnOcean protocol, the specification isn't very strict at some important security parts and therefore provides some possibilities to make implementation flaws, e.g.:

[1] *Combination of AES-Encryption and Rolling Code. Same plaintext results in different ciphertext.*

- **All security mechanisms are optional**. This means, that manufacturers have to understand the threats and choose the needed measures on their own.
- **Resynchronization of the rolling code** is not specified clearly. If devices lose their synchronization of the rolling code for some reason, it has to be resynchronized in a secure way. The security of this resynchronization process is not clearly defined.
- **Optional blocking of devices.** If a wrong rolling code is used, the sender can be blocked in order to prevent misusage. This behavior is optional.
- **Optional usage of a pre-shared key.** The usage of a pre-shared key to secure the initial transmission of security related information is optional.
- etc.

3.5 KNX

KNX is an international standard for home and building control. The KNX Association founded it in 2002, based on the three predecessors European Installation Bus (EIB), the European Home Systems Protocol (EHS) and BatiBUS. While KNX is traditionally a cable-based protocol, it could now actually be operated based on the following communication media:

- Dedicated Twisted-Pair Cable (KNX TP-1)
- Powerline (KNX PL110)
- Radio (KNX RF)
- Ethernet (KNX IP or KNXnet/IP)

Basically KNX does not provide any security mechanisms in order to prevent the protocol against the threats mentioned above. The only installed security mechanism is the password-based access control for the management interface of KNX devices. The user has to provide a four-byte long password, which is transmitted in plaintext, so it could be sniffed by a man-in-the-middle. If the password is correct, a specific privilege level is assigned to the source address of the sender. Every packet that originates from this source address is authorized from now on. Therefore, an attacker just has to set the right source address of the management packets he wants to send to the device, in order to bypass the access control mechanism. In consequence, not even this mechanism can be considered as secure.

The security of KNX is completely dependent on the physical isolation of the communication medium, which is quite hard to provide for radio communication, but possible for e.g. dedicated twisted pair cables. However, if it comes to hotels or automated apartment buildings, it becomes very difficult to protect the data-transmission completely from unauthorized access [17].

Besides various different scientific proposals, there is also a specification draft of a KNX security extension. However, currently there are no products available on the market, which support this extension.

4 Conclusion and Prospect

Smart home systems and the Internet of Things are highly interesting technologies, which connect our virtual world with our real life. They introduce many new communication protocols and combine them with all the traditional web technologies. Therefore, one has to face all the well-known web vulnerabilities, as well as new, sophisticated low-level communication protocol flaws. Currently, the main problem is to provide a secure management of cryptographic keys, which is still easy to use for end consumers.

New products, protocols and technologies are released nearly every single day. Therefore, a lot of interesting research is waiting for security professionals. At least the previously mentioned ZigBee 3.0 specification, which should be released soon, and the brand new protocol “Thread”, developed by Google’s Nest, Samsung, Silicon Labs and much more will be of particular interest.

5 References

If no other reference is quoted for a specific statement within this paper, the content originates from the diploma thesis [1] of the paper’s author. Original references can be looked up directly in the diploma thesis.

[1] D. Schwarz, “Smart Home Security”, University of Applied Sciences St. Pölten, 2015

[2] D. Schwarz, A. Inführ, R. Pölzelbauer and M. Deticek, "Loxone Multiple Vulnerabilities", 2015, https://www.sec-consult.com/fxdata/seccons/prod/temedia/advisories_txt/20150227-0_Loxone_Smart_Home_Multiple_Vulnerabilities_v10.txt

[3] J. Greil, "Multiple vulnerabilities in Loxone Smart Home (part 2)", 2015, https://www.sec-consult.com/fxdata/seccons/prod/temedia/advisories_txt/20150514-0_Loxone_Smart_Home_Multiple_Vulnerabilities_part2_v10.txt

[4] D. Schwarz, "SEC Consult Study on Smart Home Security in Germany - a first silver lining on the horizon of IoT?", 2016, http://blog.sec-consult.com/2016/04/smart-home-security.html

[5] OWASP, "Internet of Things Top Ten Project", 2014, https://www.owasp.org/index.php/OWASP_Internet_of_Things_Project#tab=Top_10_IoT_Vulnerabilities__282014_29

[6] J. Boie, "So lässt sich das UMTS-Netz knacken", Süddeutsche Zeitung, 2014, http://www.sueddeutsche.de/digital/abhoeren-von-handys-so-laesst-sich-das-umts-netz-knacken-1.2273436

[7] M. Knasmüller, "Zur Echtheit und Manipulation von SMS", 2013, http://www.gewaltinfo.at/uploads/pdf/news/knasmueller_02-2013.pdf

[8] E. Blakemore, "Can digital picture frames and thumb drives spread viruses?", 2012, http://blogs.microsoft.com/cybertrust/2012/04/20/can-digital-picture-frames-and-thumb-drives-spread-viruses/

[9] B. Fouladi and S. Ghanoun, "Security Evaluation of the Z-Wave Wireless Protocol", Black Hat USA, Las Vegas, 2013

[10] ZigBee Alliance, "ZigBee 3.0", http://www.zigbee.org/zigbee-for-developers/zigbee3-0/

[11] T. Zillner and S. Strobl, "ZigBee Exploited - The good, the bad and the ugly", Black Hat USA, Las Vegas, 2015, https://www.blackhat.com/docs/us-15/materials/us-15-Zillner-ZigBee-Exploited-The-Good-The-Bad-And-The-Ugly.pdf

[12] S. Laufer and C. Mallas, "Attacking HomeMatic," 2013, http://media.ccc.de/browse/congress/2013/30C3_-_5444_-_en_-_saal_g_-_201312301600_-_attacking_homematic_-_sathya_-_malli.html

[13] eQ-3, "FAQ", 2014, http://www.eq-3.de/faq.html

[14] pastebin, "homematic bidcos default aes key", 2014, http://pastebin.com/eiDnuS8N

[15] pastebin, "Dissecting HomeMatic AES", 2015, http://pastebin.com/bas7Lrk7

[16] EnOcean GmbH, "Security of EnOcean Radio Networks, V1.9", 2013

[17] J. Molina, "Learn how to control every room at a luxury hotel remotely: The dangers of insecure home automation deployment", Black Hat USA, Las Vegas, 2014, https://www.blackhat.com/docs/us-14/materials/us-14-Molina-Learn-How-To-Control-Every-Room-At-A-Luxury-Hotel-Remotely-The-Dangers-Of-Insecure-Home-Automation-Deployment.pdf

About the Vulnerability Lab

Members of the SEC Consult Vulnerability Lab perform security research in various topics of technical information security. Projects include vulnerability research and the development of cutting edge security tools and methodologies and are supported by partners like the Technical University of Vienna. The lab has published security vulnerabilities in many high profile software products and selected work has been presented at top security conferences like Blackhat and DeepSec.

For more information, see http://www.sec-consult.com/