ERNW Newsletter 49 / August 2015

Security of Home Automation Systems

ERNW Enno Rey Netzwerke GmbH
Carl-Bosch-Str. 4

69115 Heidelberg

Tel. +49 6221 480390
Fax +49 6221 419008
www.ernw.de

Version: 1.0

Date: 7/31/2015

Author(s): Dominik Schneider, Wojtek Przybilla

ERNW
providing security.

Table of Content

ERNW
providing security.

List of Figures

1 Abstract

Home Automation Systems are used more and more in new and modern buildings. They provide many comfortable functions, which make our daily life easier. Nearly every functionality in a building can be controlled with such a system, also security-relevant mechanisms like alarm systems. Therefore the security of the home automation itself should be as secure as possible. This fact should also apply to extensions like web interfaces for controlling smart homes via a web browser. This document examines different security aspects of the KNX technology as well as extensions, for example web interfaces which can also be part of an installation.

2 Introduction

Computers are an integral part of both daily business, and private life and are widely used. The amount of communicating devices increases from day to day. The Internet of Things and Industry 4.0 are imminent, as well as the next step in the evolution of the communication area. Nearly every device has some kind of technology to communicate via the Internet with other participants, be it the vendor or some central agency. In the same extent, the amount of transferred data increases as well. The exchanged information can be completely different, but with the entry of information systems or any kind of smart device into our daily life, the security aspects of the exchanged data are mostly unconsidered. These days even medical records are exchanged through the Internet, which represent highly sensitive information about a person.

More and more things are connected to each other and nearly every device becomes some kind of "smart". After smart phones and smart watches, the new technique which will be part of our life will be smart homes. A building will be equipped with devices that allow for control of nearly every functionality, be it the light or the blinds. Besides these devices security relevant devices like an alarm system or smoke detectors can also be controlled in a smart home. For that reason, particular attention should be placed on this technology.

Of course there are some benefits which come with home automation systems like comfort and reduced operational costs, but security and safety aspects should also be valued. The intention of this document is to analyze the security of KNX in a variety of sectors. First, a web application for a comfortable controlling of an installed KNX network, provided by a KNX extension device, is examined. This will be followed by a practical attack against a building equipped with KNX will be demonstrated.

3 WHAT IS EIB / KNX?

The European Installation Bus (EIB) is a technology for building home automation systems. In 1999, three different members founded the KNX Association. The members were the European Installation Bus Association (EIBA), the European Home Systems Association (EHSA) and the BatiBUS Club International (BCI). The goal of the newly founded KNX Association was to provide a technology which can be widely used, and to become the single standard in the field of building home automation systems. These days there are about 315000 smart homes in Germany; three quarters of them are supplied with the KNX technology. According to a survey of BITKOM, the federal Association for Information Technology, Telecommunications and New Media, by 2020 there will be one million smart homes in Germany.[1]

[1] *Andreas Streim, Tobias Arns: Connected Home, (2014)*

4 Components of a KNX Network

The assembled components of a KNX installation are dependent on the desired functionality. For turning a light on and off, different components are required than for checking the temperature of a specific room. Certainly every KNX installation needs some basic components, for example a power supply or an interface for the initial programming of the installation.

4.1 Required Components

The basic components which are required are:

- Power supply
- Interface for the initial programming (e.g. KNX/IP – Interface)
- Switch actuator
- Sensor
- Bus line (for the interconnection)

If more features are required, the installation has to be expanded by the specific devices, which provide the desired functionality. There are devices for nearly every scenario. If somebody wants that the light in the kitchen turns on at a specific time or that this light has a dim feature, buying a device that provides this feature is the only requirement. You have just to select a switch-actuator by a vendor of your choice and program it. Furthermore, an alarm system together with a motion sensor could be installed and connected to the KNX installation. The capabilities, which are provided by a home automation system with the KNX technology, are versatile. More and more devices are developed for this system, and at the moment, this is just the beginning.

4.2 Testing Environment

For testing KNX devices and learning how all the comfortable things work, a minimal testing environment was set up. The set up consists of devices from different vendors, for example Berker, Busch-Jäger and EIBMarkt. Due to the excellent standardization and the coalition of vendors in the Konnex Group, using different devices in one set up leads to no issues. Five devices form the testing environment, which are shown in the figure below.

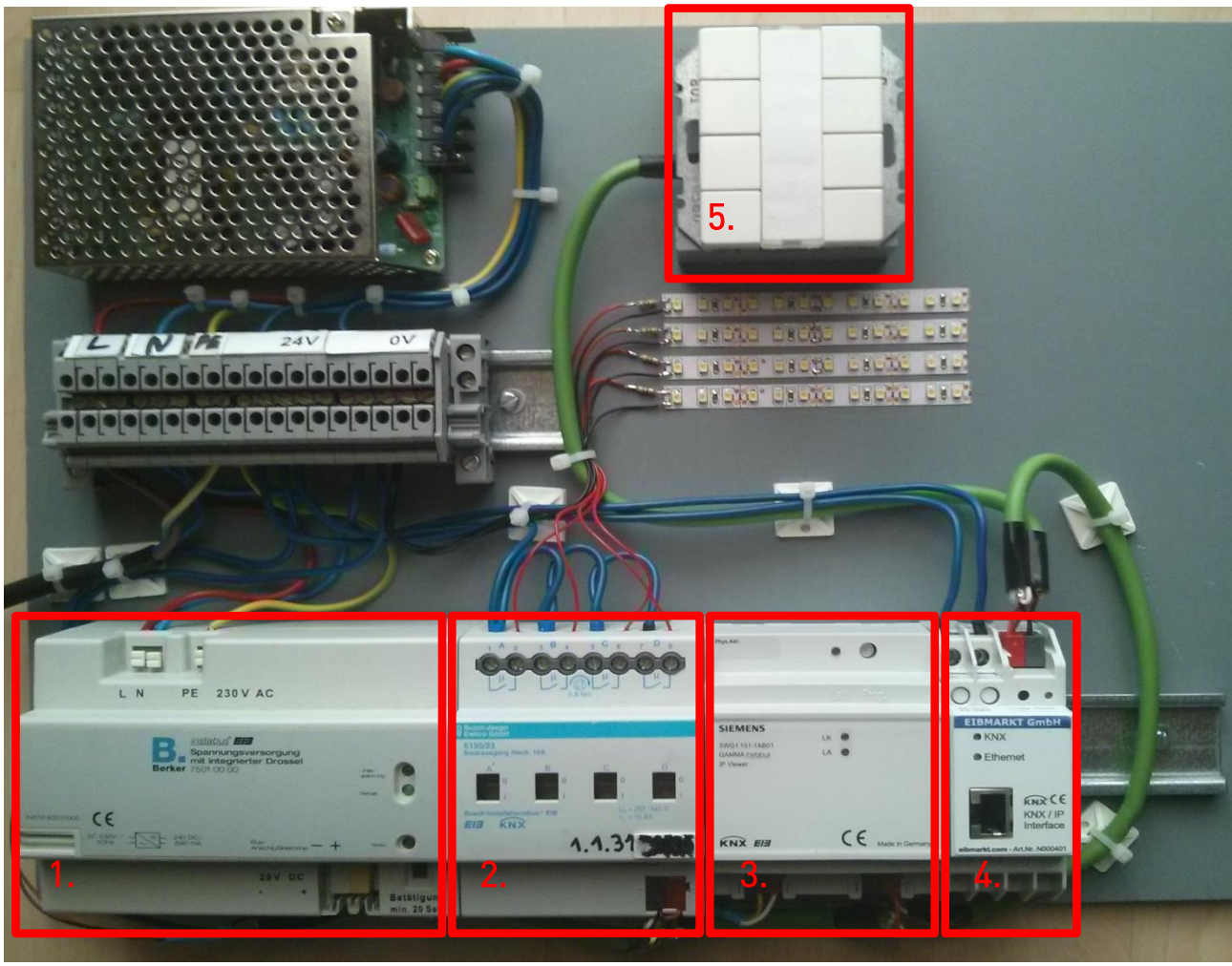


Figure 1: Testing Environment

1. Berker bus power supply

   The power supply provides the electricity which is required by the different KNX components. The electricity is supplied via the bus wire and delivers a 28 volt direct current. A further function of the power supply is the reset. The reset is used for restoring the KNX devices to the original system state. This means, that all programmed functionality of the set up will be deleted. Once a reset is performed, the KNX devices have to be programmed again.

2. Busch-Jäger switch actuator

   This device is required for controlling the electric load. In the testing environment it is used to control the LED stripes. Four separated LED stripes are mounted; all of them can be controlled individually. For turning these LED stripes on or of a button has to be pushed.

3. Siemens IP Viewer

   This device can be used for the comfortable control via the web browser or an app on the mobile phone. The Siemens IP Viewer comes with an integrated web server. This KNX device is designed for people who want to control their smart home from all over the world. The handling of the embedded web application is very easy and the most of the functions are intuitive.

4. EIBMarkt IP Interface

   Like the device before, the IP Interface is also needed to program the KNX system. In contrast to the IP Viewer this device doesn't comes with an embedded web application. The device has some basic network protocols functionality. These are ARP, ICMP, IGMP, UDP/IP and DHCP. The primary function of the IP Interface is to transform KNX telegrams, which are packed in IP packets, into "real" KNX telegrams. After transformation, the IP Interface sends the KNX telegrams to the bus.

5. Berker switch sensor

   The Busch Jäger switch actuator turns the light on or off but needs to receive a command to do so. For this purpose, the Berker switch sensor is needed. If a button gets pushed on it, the Busch Jäger switch actuator receives the signal and turns the corresponding light on or off, depending on the state of the light, before the button gets pushed.

4.3 ETS – Engineering Tool Software

In contrast to the standard and the KNXnet/IP protocol, which is open source, the software which is required for the initial programming task is proprietary. This software can be downloaded from the KNX website and installed on a Windows-based computer, it's called ETS – Engineering Tool Software[2]. Software for other operating systems is not provided by the KNX group. At the moment, there is no other software available besides ETS to program KNX devices. If someone wants to use the KNX technology, to transfer their home to a smart home or install it in a new building, there isn't the possibility to use other software.

The software is available in three versions; these are Demo, Lite and Pro. The difference between these 3 versions is the number of devices which can be programmed at the same time. With the Demo version only 3 devices can be programmed at the same time. For a testing environment with just a few devices this could be enough, but for a real installation in a building the demo version wouldn't be adequate. With the Lite version it's possible to program up to 20 devices at the same time, while the Professional version has no such restrictions. The figure below shows the software's interface.

[2] *KNX Association: ETS5, (2014)*

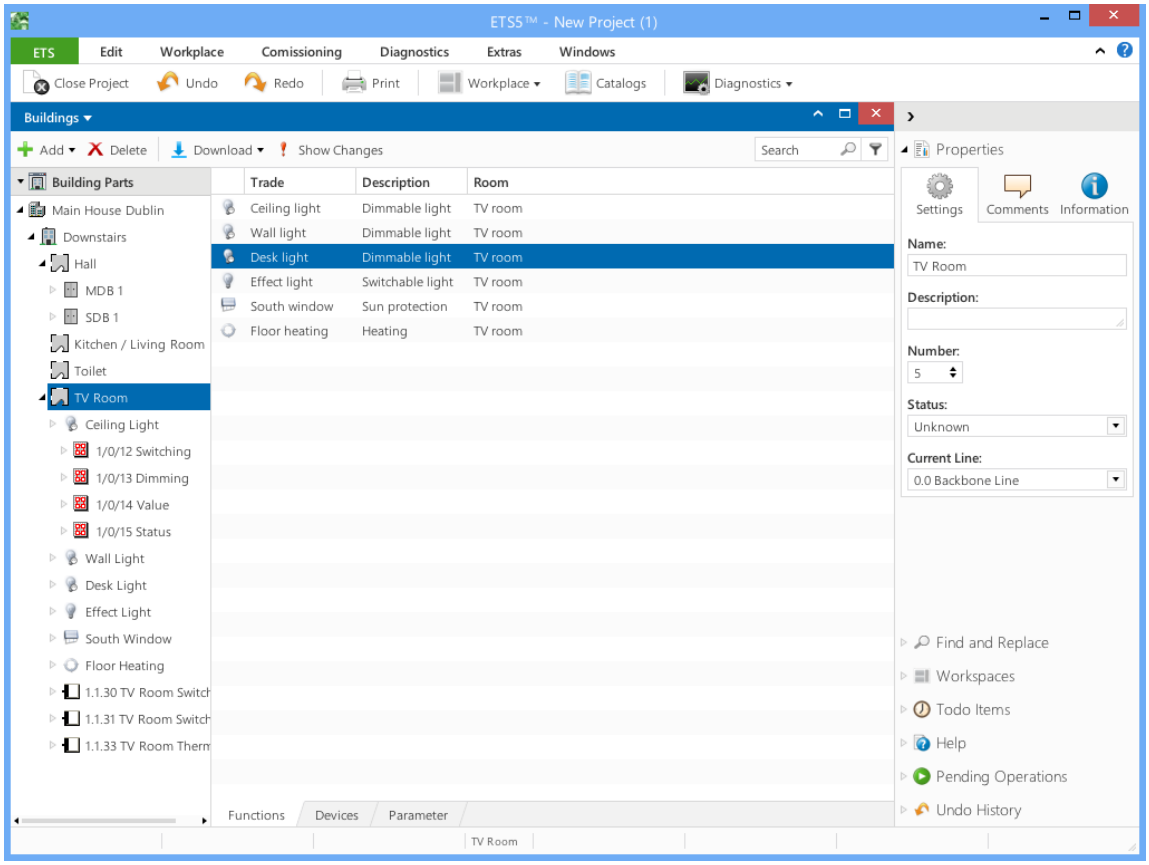


Figure 2: Engineering Tool Software

ETS5 has an extensive amount of functionalities and nearly every scenario can be programmed with it. Whether a light should turn on at specific time or the air conditioner should keep the room temperature at a specific level. For all these features a device is needed which supports them. If an actuator just supports turning lights on or off, dimming actions cannot be programmed. For this reason it is very important to specify all functionalities which should be available in the planning phase. Otherwise, wrong devices would be installed in the building and have to be replaced by devices which support the desired functionalities. The KNX association provides a document for this approach; the checklist "Step-by-step project management Part 1: Start of project"[3]. Within this document different questions have to be answered. These questions provide a better understanding to the electrician who will install the KNX network, which requirements the customer has. Depending on the result of this document, the installation can be adjusted accordingly. Making use of this document during the planning stages prevents more time being spent fixing mistakes in the implementation phase.

If all devices are installed the next step is to program them. For this task special software is required which provides the specific functionality depending on the device. This software, called a "catalogue", can be downloaded from the vendor's website. After importing it into the ETS software, the device, for which the software is designed, can be programmed.

[3] *KNX Association: Checklist, (2015)*

5 Security of Smart Home Systems

Controlling a house with just one click on the computer or an app on the smart phone, presents a very easy way to manage one's home. From everywhere in the house or all over the world via the Internet the components can be controlled. Therefore the security aspects should be considered as thoroughly as possible.

The standard EN 13321-2:2012 suggests that security played a minor role in the development of this technology which can be observed in chapter 5.1.3.1 of the standard.

> *"Für KNX war und ist das Thema Sicherheit von keiner großen Bedeutung, da man für eine Verletzung der Sicherheit lokal Zugriff auf das Netzwerk haben muss. Im Fall von KNX TP (EIB) und KNX PL bedeutet das, dass man dafür sogar den physikalischen Zugriff auf die Netzwerkkabel benötigt, was in fast allen Fällen unmöglich ist, da die Kabel innerhalb des Gebäudes oder unter der Erde verlegt sind. Aus diesem Grund spielen Sicherheitsaspekte für KNX-Medien auf der Feldebene eine untergeordnete Rolle."*[4]

In chapter 5.1.3 some attack scenarios are discussed and possible defensive measures proposed which lead to the following conclusion:

> *"Es ist eher unwahrscheinlich, dass legitimierte Benutzer eines Netzwerks über Mittel zum Abfangen und Entschlüsseln verfügen, um KNXnet/IP anschließend zu verfälschen, ohne die KNX-Normen intensiv studiert zu haben.*
> *Daher wird die verbleibende Sicherheitsbedrohung als sehr gering eingeschätzt und rechtfertigt nicht die Verwendung einer Verschlüsselung, die beachtliche Computerressourcen erfordern würde."*[5]

[4] *EN 13321-2:2012, Chapter 5.1.3.1*
[5] *EN 13321-2:2012, Chapter 5.1.3.4*

The conclusion of this chapter is that a potential attacker doesn't have the knowledge to analyze intercepted packets and to tamper them without knowing the standard EN 13321-2:2012. Whether this assumption represents a valid attitude nowadays is highly questionable.

5.1 Ethernet / KNX - Gateway

Besides KNX there are also other technologies available to control a building. But the fact that the protocol has existed since the early 90's, and is well standardizedation ,makes it unique. Nowadays, just equipping the building with the actuators and sensors, which are needed doesn't fit the expectation on a smart home. The people want to control their blinds, the heating, the fridge and also their washing machine when they aren't at home. For this purpose an additional device is required to establish a connection via the Internet. This means that with the installation of such a device, a further point of failure becomes part of the whole system. This is similar to an example in the area of computers and viruses. Antivirus software will be installed to protect the computer but if the antivirus software itself contains vulnerabilities this additional software can make the whole system insecure.

This could also happen with smart home devices. You equip your building with such devices, for example an alarm system, motion detectors and cctv. If you want to control your smart home via the Internet a gateway, which connects the smart home network with the Internet is required. If this device is affected by just one vulnerability, this vulnerability can make the whole system insecure and the actual goal to make the building more secure fails.

There are many different gateway devices available. Some of them come with an easy to use web application as user interface, while others just receive the packets from the Internet and forwards them to the smart home network. We tested one of these devices, which comes with a web application as user interface. The following chapter isn't related to the IP Viewer included in our testing environment.

5.2 The tested device

Normally a user authentication is required to get access to the visualization. After a successful login the lights, the blinds as well as other devices included into the smart home network can be controlled via mouse clicks.

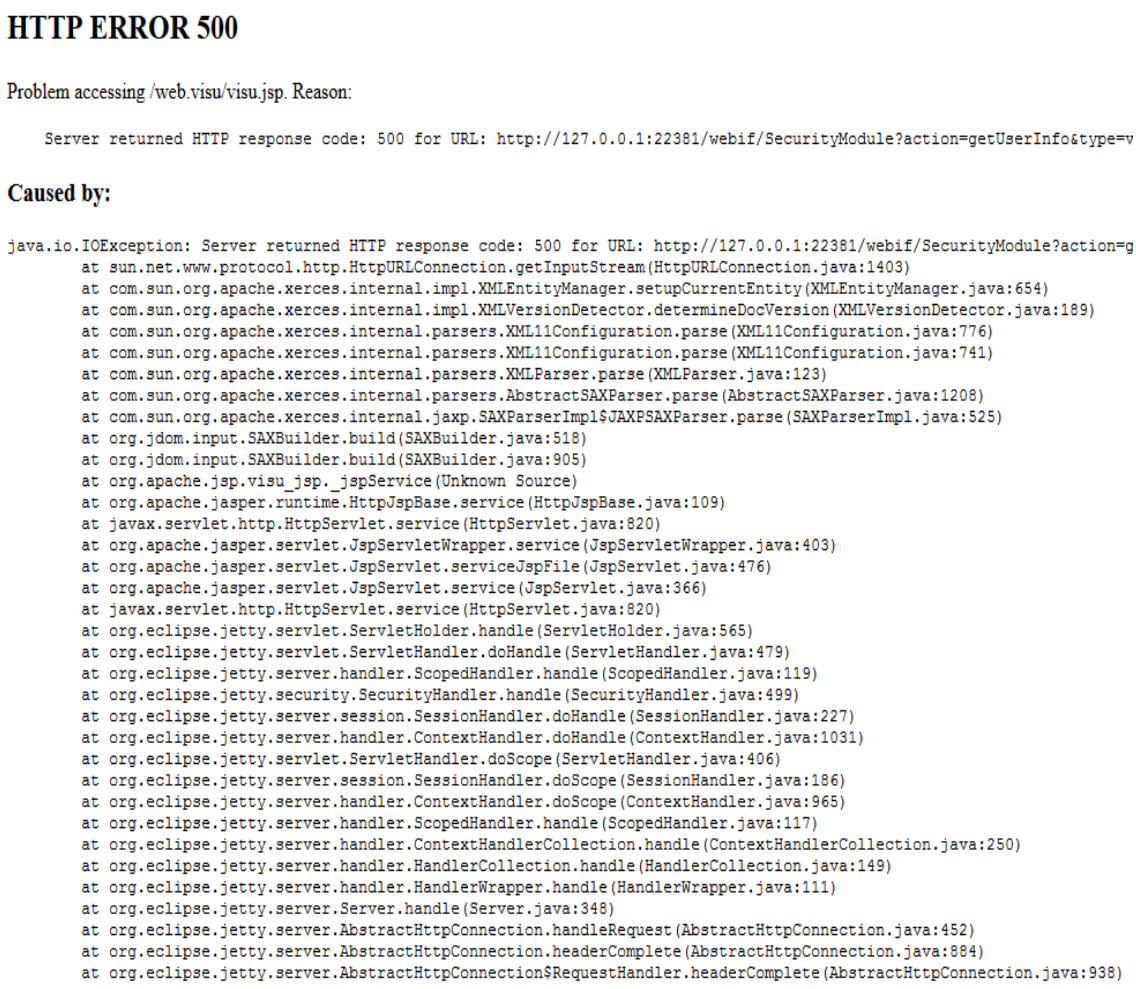
To get a feeling of how many of these gateways are available through the Internet, we wrote a short script, which crawls for a special URL pattern. This path is available even when a user authentication is enabled and only authorized people should be able to get the content.

The results are terrifying. There are so many devices out there that are running with disabled user authentication. Therefore everybody who knows the host can control the smart home installation. The following screenshots are a subset of our results.

Figure 3: Ethernet / KNX - Gateway1

Figure 4: Ethernet / KNX - Gateway2

During the test we also detected multiple cross-site scripting vulnerabilities, which can be used to take over a valid session. Further, application errors like stack traces can be generated by supplying uncommon values for parameters or even no input. In case of the tested gateway, invalid input into the parameter projectID led to the stack trace below. As input the character ' was used, but the stack trace occurs as well if no value was supplied via the projectID parameter. The stack trace displays a lot of information about the used libraries and software. Analyzing it shows that a Jetty server is also running on the device and the XML parser Apache Xerces is used. With this information a potential attacker can do some further research, which can lead to the identification of possible vulnerabilities.



HTTP ERROR 500

Problem accessing /web.visu/visu.jsp. Reason:

```
    Server returned HTTP response code: 500 for URL: http://127.0.0.1:22381/webif/SecurityModule?action=getUserInfo&type=v
```

Caused by:

```
java.io.IOException: Server returned HTTP response code: 500 for URL: http://127.0.0.1:22381/webif/SecurityModule?action=g
        at sun.net.www.protocol.http.HttpURLConnection.getInputStream(HttpURLConnection.java:1403)
        at com.sun.org.apache.xerces.internal.impl.XMLEntityManager.setupCurrentEntity(XMLEntityManager.java:654)
        at com.sun.org.apache.xerces.internal.impl.XMLVersionDetector.determineDocVersion(XMLVersionDetector.java:189)
        at com.sun.org.apache.xerces.internal.parsers.XML11Configuration.parse(XML11Configuration.java:776)
        at com.sun.org.apache.xerces.internal.parsers.XML11Configuration.parse(XML11Configuration.java:741)
        at com.sun.org.apache.xerces.internal.parsers.XMLParser.parse(XMLParser.java:123)
        at com.sun.org.apache.xerces.internal.parsers.AbstractSAXParser.parse(AbstractSAXParser.java:1208)
        at com.sun.org.apache.xerces.internal.jaxp.SAXParserImpl$JAXPSAXParser.parse(SAXParserImpl.java:525)
        at org.jdom.input.SAXBuilder.build(SAXBuilder.java:518)
        at org.jdom.input.SAXBuilder.build(SAXBuilder.java:905)
        at org.apache.jsp.visu_jsp._jspService(Unknown Source)
        at org.apache.jasper.runtime.HttpJspBase.service(HttpJspBase.java:109)
        at javax.servlet.http.HttpServlet.service(HttpServlet.java:820)
        at org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.java:403)
        at org.apache.jasper.servlet.JspServlet.serviceJspFile(JspServlet.java:476)
        at org.apache.jasper.servlet.JspServlet.service(JspServlet.java:366)
        at javax.servlet.http.HttpServlet.service(HttpServlet.java:820)
        at org.eclipse.jetty.servlet.ServletHolder.handle(ServletHolder.java:565)
        at org.eclipse.jetty.servlet.ServletHandler.doHandle(ServletHandler.java:479)
        at org.eclipse.jetty.server.handler.ScopedHandler.handle(ScopedHandler.java:119)
        at org.eclipse.jetty.security.SecurityHandler.handle(SecurityHandler.java:499)
        at org.eclipse.jetty.server.session.SessionHandler.doHandle(SessionHandler.java:227)
        at org.eclipse.jetty.server.handler.ContextHandler.doHandle(ContextHandler.java:1031)
        at org.eclipse.jetty.servlet.ServletHandler.doScope(ServletHandler.java:406)
        at org.eclipse.jetty.server.session.SessionHandler.doScope(SessionHandler.java:186)
        at org.eclipse.jetty.server.handler.ContextHandler.doScope(ContextHandler.java:965)
        at org.eclipse.jetty.server.handler.ScopedHandler.handle(ScopedHandler.java:117)
        at org.eclipse.jetty.server.handler.ContextHandlerCollection.handle(ContextHandlerCollection.java:250)
        at org.eclipse.jetty.server.handler.HandlerCollection.handle(HandlerCollection.java:149)
        at org.eclipse.jetty.server.handler.HandlerWrapper.handle(HandlerWrapper.java:111)
        at org.eclipse.jetty.server.Server.handle(Server.java:348)
        at org.eclipse.jetty.server.AbstractHttpConnection.handleRequest(AbstractHttpConnection.java:452)
        at org.eclipse.jetty.server.AbstractHttpConnection.headerComplete(AbstractHttpConnection.java:884)
        at org.eclipse.jetty.server.AbstractHttpConnection$RequestHandler.headerComplete(AbstractHttpConnection.java:938)
```

Figure 5: Stack Trace Output

5.3 Special Case

A quite special case is the following installation we found.

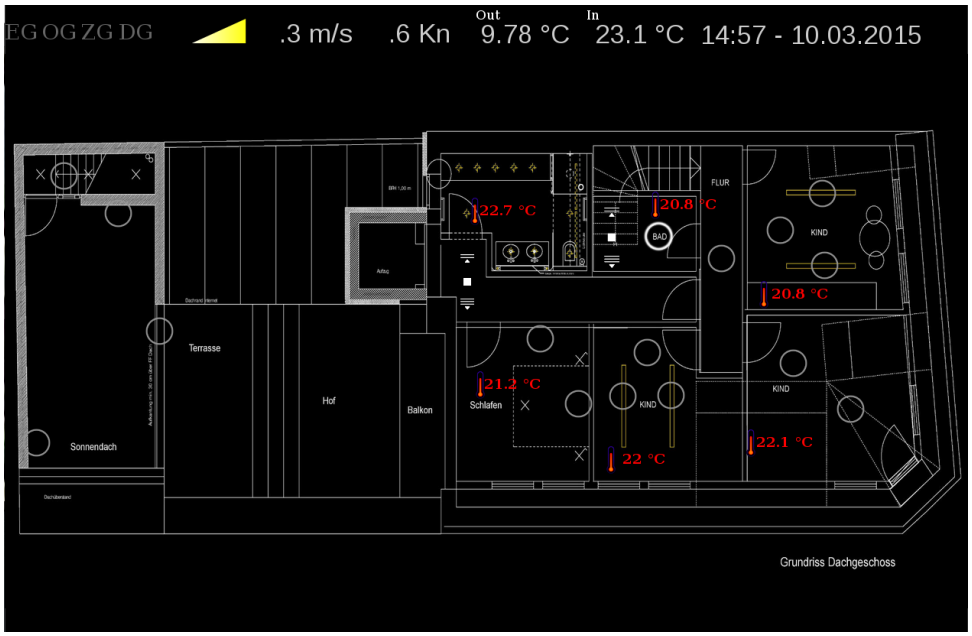


Figure 6: Special Case

It is possible to provide weather information to the gateway. For using this feature some localization information must be provided to the gateway, for example a zip code. Amongst the zip code, a project name was also provided to the device. The project name was something like that "Fam. ..." which is basically the family name of the persons living in the house. We used the gathered information, the zip code as well as the family name, for a search via google maps. The result is shown below:

Figure 7: Special Case - Google Maps

5.4 Result Ethernet / KNX – Gateway

Related to the described results it can also be summarized that this new smart device contains multiple security issues, which are known in the area of web applications. These security issues can lead to a collapse of the whole system and the actual goal, of making the facility more energy-efficient or secure in case of included alarm systems, fails. Ethernet / KNX – Gateways could represent the weakest link of the chain. Therefore it is highly recommended to evaluate besides the correct functioning also the security of these devices on every level.

6 A Real World Scenario – Facility Hacking

The University of Applied Sciences in Offenburg built a new building on the campus. All other buildings before got a letter as a label. A to D are already assigned to other buildings, so the new building received the label E. As written in a press release, the building got an energetical concept, which is forward-looking[6]. This new concept needs a technology that provides the system with information for an automated controlling. The used technology is KNX.

6.1 Getting Physical Access to the Bus

The first part was to get access to the bus. Otherwise, no communication could be established. The first time the building was open and courses were held the building was still under construction. Because of this some doors within the rooms at the wall weren't closed. Behind these doors different devices were built in, for example KNX switch actuators for controlling the light or the blinds.

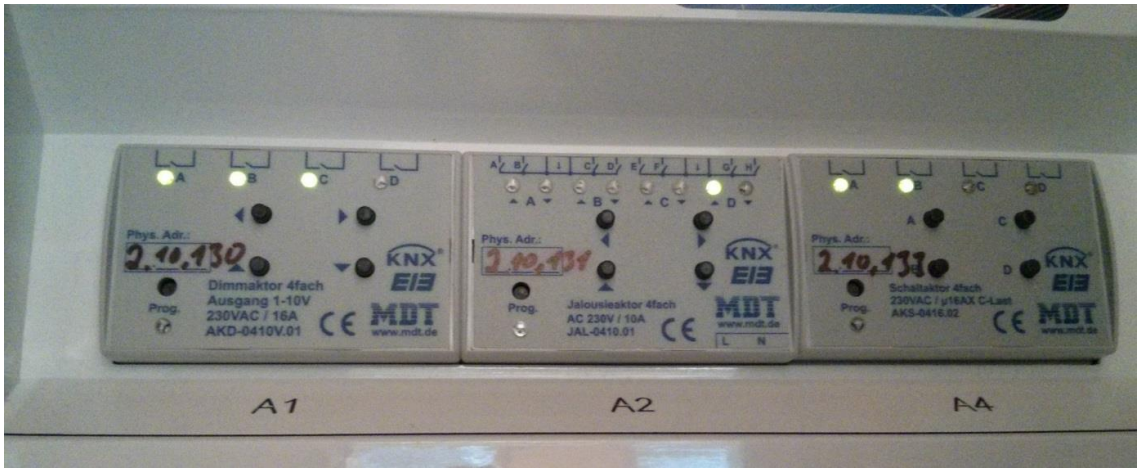


Figure 8: Actuators

[6] *Susanne Gilg: Neues E-Gebäude eingeweiht, (2014)*

The picture above shows the installed switch actuators by the vendor MDT. For example, the switch actuator "A1" is used for dimming the light. This can be discerned by the label on it saying "Dimmaktor". Switch actuator "A2" is used for controlling the blinds and "A3" for controlling the light. On the different switch actuators a physical address is labeled. This presents helpful information, but there was no easy possibility of getting access to the bus without any kind of action like removing a device and replacing it with another one.

After an inspection of potential access points a recess in the ground was found. Through this recess different cables, like network cables or power cables, protruded and were connect to a computer. In the recess there was an unconnected KNX bus cable, ready to be used (the green one seen in the figure below).

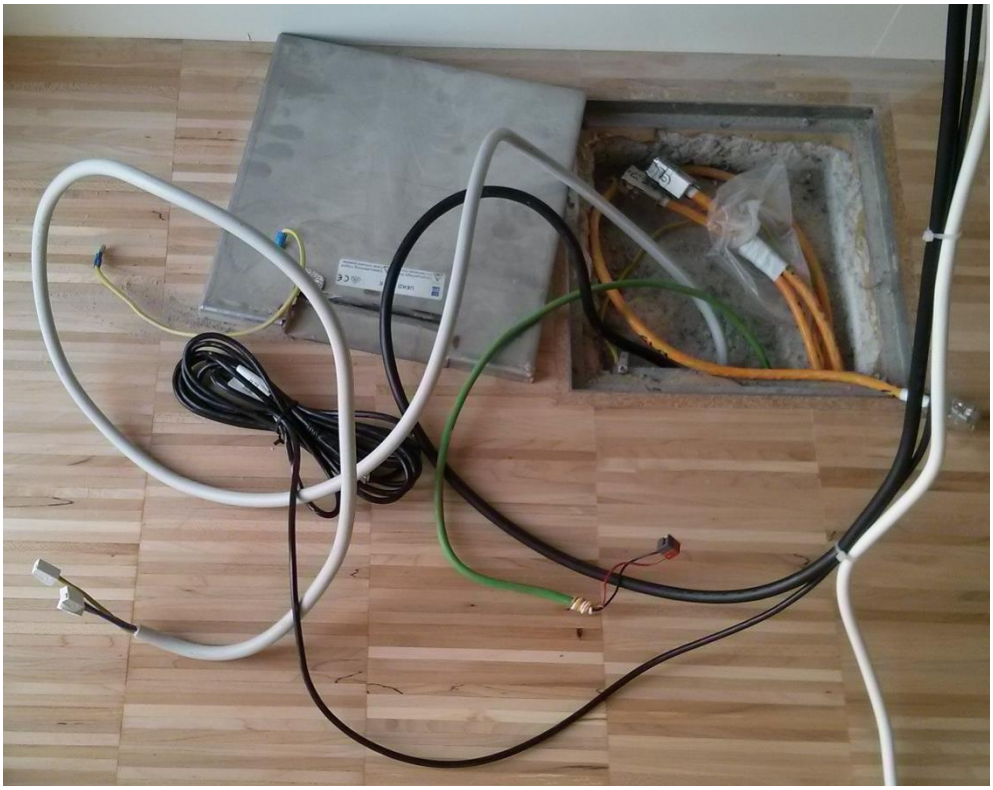


Figure 9: Recess with bus cable

The reason for the unconnected bus cable is questionable Maybe it exists for extending the existing installation. In every room there is a recess like the one shown in the picture above. This makes it very easy for a potential attacker, as there are a large number of available recesses. The building has 5 floors and on every floor there are at least 4 rooms, besides offices of professors or other staff, which are accessible for everyone. Getting physical access to the bus would be no challenge.

6.2 Getting Access to the Communication

After a physical access is established, the next step would be to get access to the communication on the bus. For this purpose some kind of hardware and software are required, which can interpret the communication and receive or send packets, too. The testing environment in chapter 4.2 already contains devices for establishing a connection to the bus,

such as the IP Viewer or the KNXnet/IP-Interface. The IP Viewer by Siemens was chosen, because this device only needs the bus for getting the needed power. The KNXnet/IP-Interface needs a separate power adapter, so there are more devices and cables to be handled. As an impression for the complete attack setup serves the picture below.

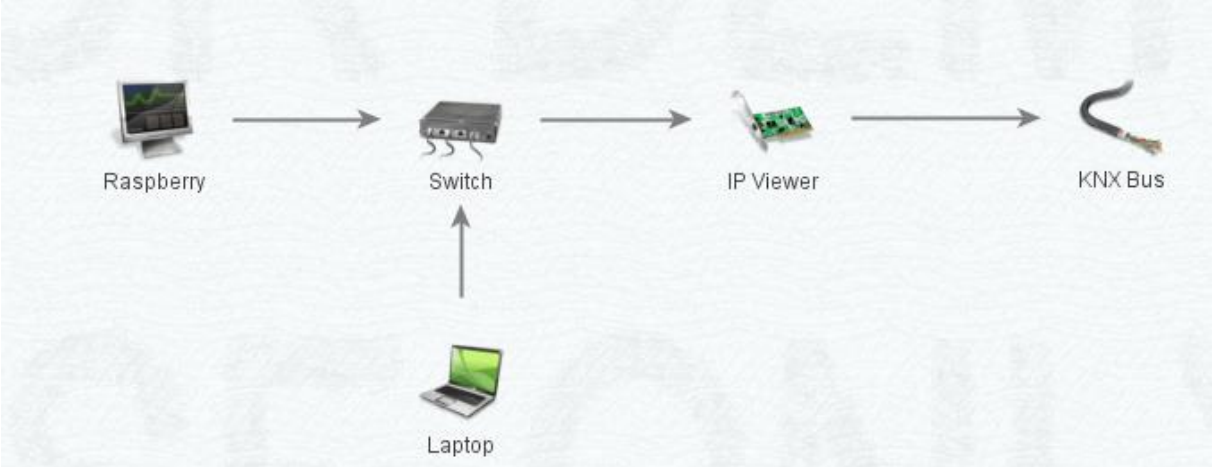


Figure 101: Attack setup

As described before the IP Viewer is used for the connection to the bus. Furthermore, a switch is required for the connection to the IP Viewer, as well as a computer for running specific software. A Raspberry Pi b+[7] takes the part of the computer. It is a very handy device and all kind of software could be deployed on it. The Raspberry Pi is controlled via a laptop, which is also connected to the switch.

6.3 Reading / Sending Packets

Via the bus all kind of devices exchange data, for example which switch was pushed or what state the light on a specific floor has. Because of the 5 different floors in the "E" building there is a big amount of devices, thus the resulting traffic is higher than at a KNX installation in a single-family house. The first step would be to analyze which devices are talking to each other. For this approach some kind of software is required, a sniffer for example.

After some research the software eibd[8] was found. This software contains a server for reading or sending packets, which are transferred via the bus. The developer of the software, Martin Kögler, describes it like follows:

> *"It provides an interface to the EIB / KNX bus." […] "Eibd provides over a TCP/IP and/or unix domain sockets access to the EIB bus using a simple protocol. It provides access at layer 4 as well as to high level mangement function. Multiple concurrent users are supported. A special bus monitor mode*

[7] *Raspberry Pi Foundation: Raspberry Pi*
[8] *Martin Kögler: EIBD*

call vBusmonitor is implemented, which delivers all traffic, which eibd passes, but not disturbs send activities." Additionally, eibd can acts as limited EIBnet/IP Tunneling and/or Routing Server. The limitation is, that only one KNX address can be used as source address of EIBnet/IP Clients. This address is shared between multiple possible routing or tunneling connections. To the EIBnet/IP client, the address 0.0.0 is used, to the EIB/KNX bus the address of the interface device is used. eibd maps between these addresses (like NAT for TCP/IP)."Enabling the EIBnet/IP Server disables the normal bus monitor function (vBusmonitor still works). It supports all interfaces supported by eibd." [9]

Martin Kögler describes eibd as a tool for basic communication to the bus through different interfaces. Therefore, the goal of the software is to provide basic functionalities, which can be used by people, who want to program their own software for their personal smart home.

The purpose for using this useful software depends on the user. An attacker would use this software for sniffing on the bus and sending malicious packets to the KNX devices. On the other hand a normal user would just write his personal KNX software and extend the existing installation. In the following paragraphs it is assumed that an attacker would use this software.

First, the software has to be installed. Which requirements are given and which additional packages are needed can be looked up in chapter "10.1. Installation" of the document "Free Development Environment for Bus Coupling Units of the European Installation Bus" written by Martin Kögler[10]. There are also some scripts available, which install eibd and also all the required packages at a stroke.[11]

6.3.1 Establishing a Connection

After setting up eibd, a DHCP server is required for providing the devices connected to the switch with an IP address. For this purpose the standard Debian DHCP server, available under the package isc-dhcp-server[12], was also installed on the raspberry. The basic setup, eibd for the communication with the devices connected to the bus and a DHCP server for the communication of the network devices was done.

To determine if the installed software is working try to run eibd would be the first task. To do so some information is required, for example the IP of the KNXnet/IP – Interface. The DHCP server was configured to deploy IP addresses in the range of 192.168.178.10 to 192.168.178.20, a quick network scan of this range should find the desired information. For this purpose nmap, a network scanner[13], was used with the following call:

---

[9] *Martin Kögler: EIBD - Quotation*
[10] *Martin Kögler: Free Development Environment for Bus Coupling Units of the European Installation Bus, (2008)*
[11] *Michael Albert: Raspberry Pi: eibd with a KNX USB Interface, (2014)*
[12] *Internet Systems Consortium, Inc.: ISC DHCP, (2015)*
[13] *Gordon Lyon: Nmap, (2015)*

nmap –sn 192.168.178.10-20

As result the following output was generated.

```
[root@localhost ds]# nmap -sn 192.168.178.10-15

Starting Nmap 6.45 ( http://nmap.org ) at 2015-03-26 16:20 CET
Nmap scan report for 192.168.178.11
Host is up (0.00081s latency).
MAC Address: 00:05:26:50:06:5C (Ipas Gmbh)
Nmap done: 6 IP addresses (1 host up) scanned in 0.25 seconds
[root@localhost ds]#
```

Figure 112: Nmap scan

A quick comparison between the MAC address labeled on the device (00:05:26:50:06:5C) and displayed in the output led to the fact, that the KNXnet/IP – Interface received the following IP: 192.168.178.11. Now all information is present to run eibd.

6.3.2 Running EIBD

At this point all devices should be connected and their addresses known. For running eibd these information are required otherwise the KNXnet/IP server couldn't start.

eibd –t1023 –i ipt:(ip_KNXnet/IP_Interface_device)

Immediately after starting eibd, telegrams, which are sent via the bus, are displayed on the screen, shown in the picture below.

```
Layer 1(018D1988,54E92F80) Recv L_Data low from 2.3.8 to 1/0/105 hops: 04 T_DATA_XXX_REQ A_GroupValue_Write (small) 01
Layer 2(018D1988,54E92F80) Recv L_Data low from 2.3.8 to 1/0/105 hops: 04 T_DATA_XXX_REQ A_GroupValue_Write (small) 01
Layer 3(018F28B8,54E92F80) Recv L_Data low from 2.3.8 to 1/0/105 hops: 04 T_DATA_XXX_REQ A_GroupValue_Write (small) 01
Layer 0(018D1E50,54E92F81) Recv(021): 06 10 04 20 00 15 04 01 02 00 29 00 BC C0 23 04 08 65 01 00 81
Layer 1(018D1E50,54E92F81) Recv(015): 04 01 02 00 29 00 BC C0 23 04 08 65 01 00 81
Layer 1(018D1E50,54E92F81) Send(004): 04 01 02 00
Layer 0(018D1E50,54E92F81) Send(010): 06 10 04 21 00 0A 04 01 02 00
Layer 1(018D1988,54E92F81) Recv L_Data low from 2.3.4 to 1/0/101 hops: 04 T_DATA_XXX_REQ A_GroupValue_Write (small) 01
Layer 2(018D1988,54E92F81) Recv L_Data low from 2.3.4 to 1/0/101 hops: 04 T_DATA_XXX_REQ A_GroupValue_Write (small) 01
Layer 3(018F28B8,54E92F81) Recv L_Data low from 2.3.4 to 1/0/101 hops: 04 T_DATA_XXX_REQ A_GroupValue_Write (small) 01
Layer 0(018D1E50,54E92F84) Recv(021): 06 10 04 20 00 15 04 01 03 00 29 00 BC C0 2B 16 0F 85 01 00 81
Layer 1(018D1E50,54E92F84) Recv(015): 04 01 03 00 29 00 BC C0 2B 16 0F 85 01 00 81
Layer 1(018D1E50,54E92F84) Send(004): 04 01 03 00
Layer 0(018D1E50,54E92F84) Send(010): 06 10 04 21 00 0A 04 01 03 00
Layer 1(018D1988,54E92F84) Recv L_Data low from 2.11.22 to 1/7/133 hops: 04 T_DATA_XXX_REQ A_GroupValue_Write (small) 01
Layer 2(018D1988,54E92F84) Recv L_Data low from 2.11.22 to 1/7/133 hops: 04 T_DATA_XXX_REQ A_GroupValue_Write (small) 01
Layer 3(018F28B8,54E92F84) Recv L_Data low from 2.11.22 to 1/7/133 hops: 04 T_DATA_XXX_REQ A_GroupValue_Write (small) 01
Layer 0(018D1E50,54E92F85) Recv(021): 06 10 04 20 00 15 04 01 04 00 29 00 BC C0 23 08 08 69 01 00 81
Layer 1(018D1E50,54E92F85) Recv(015): 04 01 04 00 29 00 BC C0 23 08 08 69 01 00 81
Layer 1(018D1E50,54E92F85) Send(004): 04 01 04 00
Layer 0(018D1E50,54E92F85) Send(010): 06 10 04 21 00 0A 04 01 04 00
Layer 1(018D1988,54E92F85) Recv L_Data low from 2.3.8 to 1/0/105 hops: 04 T_DATA_XXX_REQ A_GroupValue_Write (small) 01
Layer 2(018D1988,54E92F85) Recv L_Data low from 2.3.8 to 1/0/105 hops: 04 T_DATA_XXX_REQ A_GroupValue_Write (small) 01
Layer 3(018F28B8,54E92F85) Recv L_Data low from 2.3.8 to 1/0/105 hops: 04 T_DATA_XXX_REQ A_GroupValue_Write (small) 01
Layer 0(018D1E50,54E92F86) Recv(021): 06 10 04 20 00 15 04 01 05 00 29 00 BC C0 23 04 08 65 01 00 81
Layer 1(018D1E50,54E92F86) Recv(015): 04 01 05 00 29 00 BC C0 23 04 08 65 01 00 81
Layer 1(018D1E50,54E92F86) Send(004): 04 01 05 00
Layer 0(018D1E50,54E92F86) Send(010): 06 10 04 21 00 0A 04 01 05 00
Layer 1(018D1988,54E92F86) Recv L_Data low from 2.3.4 to 1/0/101 hops: 04 T_DATA_XXX_REQ A_GroupValue_Write (small) 01
Layer 2(018D1988,54E92F86) Recv L_Data low from 2.3.4 to 1/0/101 hops: 04 T_DATA_XXX_REQ A_GroupValue_Write (small) 01
Layer 3(018F28B8,54E92F86) Recv L_Data low from 2.3.4 to 1/0/101 hops: 04 T_DATA_XXX_REQ A_GroupValue_Write (small) 01
```

Figure 123: Output of EIBD

It can be seen that a large number of devices are installed in the building. Nearly every few seconds, a new telegram is displayed on the screen. From the perspective of a potential attacker that's a partial success, because every device could be a target. Furthermore the huge number of devices led to the fact, that many functionalities are integrated into the KNX network.

Over the time, if eibd is started as described above, the output becomes quiet much and it's like to lose overview. Therefore eibd should be started with an "&" at the end of the start command. This ensures that eibd is running in the background and the received data won't be displayed on the screen.

In the first step the received data is too much. The first goal should be to identify which addresses are used and between which devices telegrams are exchanged. For this purpose eibd provides a command called "groupsocketlisten". Groupsocketlisten "displays all received frames for all (destination) group address"[14]. This command displays the source address as well as the destination address and which data was send. An example output is shown in the picture below.

[14] *Martin Kögler: Free Development Environment for Bus Coupling Units of the European Installation Bus, (2008), P. 165*

```
Write from 2.11.17 to 1/7/1: 01
Write from 2.11.100 to 5/7/1: 01
Write from 2.4.23 to 1/5/129: 01
Write from 2.10.16 to 1/7/151: 01
Write from 2.11.16 to 1/7/2: 01
Write from 2.11.100 to 5/7/2: 01
Write from 2.3.8 to 1/0/105: 01
Write from 2.1.12 to 1/0/206: 01
Write from 2.10.4 to 1/7/121: 01
Write from 2.10.26 to 1/7/144: 01
Write from 2.9.19 to 1/7/203: 01
Write from 2.4.23 to 1/5/129: 01
Write from 2.10.4 to 1/7/121: 01
Write from 2.9.19 to 1/7/203: 01
Write from 2.4.21 to 1/5/129: 01
Write from 2.10.4 to 1/7/121: 01
Write from 2.6.20 to 1/6/129: 01
Write from 2.9.19 to 1/7/203: 01
Write from 2.4.24 to 1/5/136: 01
Write from 2.2.19 to 1/0/128: 01
Write from 2.0.1 to 10/4/4: 87 92
Write from 2.0.1 to 10/4/5: 01 72
Write from 2.0.1 to 10/4/11: 01
Write from 2.0.1 to 10/4/18: 00
Write from 2.0.1 to 10/4/19: 00
Write from 2.0.1 to 10/4/20: 01
Write from 2.4.25 to 1/5/136: 01
Write from 2.0.1 to 10/4/12: 00
Write from 2.0.1 to 10/4/13: 01
Write from 2.10.4 to 1/7/121: 01
Write from 2.4.24 to 1/5/136: 01
Write from 2.6.10 to 1/6/143: 01
```

Figure 134: groupsocketlisten

Starting the command could be done via the following line:

groupsocketlisten ip:localhost >> result.txt

- IP: Specifies the device used for recording the traffic. Because of the reason that eibd is running on the Raspberry Pi and the command will be started on the same device, the value localhost will be committed.
- >> results.txt: Ensures that the output data will be stored into a file called "results.txt". This file can be used to analyze the gained data at a later time.

6.4 Discovering all Devices

In chapter 6.3.2 the command "groupsocketlisten" sniffs for traffic, which is exchanged on the bus. Therefore not all devices, which are connected to the bus, could be discovered. If some devices aren't sending packets, the sniffer will not recognize them. For discovering all devices, another technique has to be used. Besides the groupsocketlisten command and several others "progmodestatus", exist. This command "returns the state of the programming mode of a device"[15].

The programming mode is used for telling the devices which functionalities they have and to which group addresses they are belonging. After the devices are installed in the building and connected to the bus line, this task will be done.

[15] *Martin Kögler: Free Development Environment for Bus Coupling Units of the European Installation Bus, (2008), P. 165*

To activate the programming mode a button on the device has to be pressed. If the programming mode was successfully enabled a small LED will light up.

A few tests revealed that at least traffic would be generated with this command. If the programming mode of a device would be activated, the device would respond with the message "in programming mode", if not "not in programming mode". Nevertheless, if a device has the checked address, it would definitely respond with a message. It doesn't matter if the programming mode is really activated or not. Both kinds of information are enough to determine that a device has the checked address. In case that a tested address wouldn't be owned by a device, the software recognizes a timeout and responds with the message "Set failed: Connection reset by peer".

6.5 Analyzing the Traffic

The received data gives some indication of the communication partner, but not how often they communicated. How often devices communicate with each other represents quite important information. From this information the type of the device could be distilled, a motion detector sends more often a telegram to an actuator than a normal switch for example. To extract this kind of information the following Python script was used.

```
1.  #!/usr/bin/python
2.  import sys
3.  import pprint
4.  from collections import defaultdict
5.  import collections
6.
7.  pp = pprint.PrettyPrinter(depth=6)
8.  f = open('all.txt', 'r')
9.  hosts ={}
10. sender = []
11. receiver = []
12. for line in f:
13.     send = line.split()[2]
14.     if send not in sender:
15.         sender.append(send)
16.     recv = line.split()[4]
17.     recv = recv.rstrip(":")
18.     if recv not in receiver:
19.         receiver.append(recv)
20.     if send in hosts:
21.         hosts[send][recv] += 1
22.     else:
23.         d={}
24.         d = defaultdict(lambda: 0, d)
25.         hosts[send] = d
26.         hosts[send][recv] += 1
27. f.close()
28. hosts2 = collections.OrderedDict(sorted(hosts.items()))
29. for key in hosts2:
30.     print (key)
31.     for key2 in hosts2[key]:
32.         print ("\t"+key2,hosts2[key][key2])
33. sender.sort()
34. receiver.sort()
```

The approach of the script looks like follows. First, all source addresses will be identified and saved once. At this point it doesn't matter if senders send packets 1 or 20 times to a groupaddress. In the second step the receivers will be identified and are also saved just once in relation to the sender. The last step is counting how often a source address communicates with a destination address. This information will be saved together with the communication partners. After running the script, the following output is given:

```
2.0.1
        10/4/20 36
        10/4/3 31
        10/4/9 1
        10/4/18 36
        10/4/19 36
        10/4/4 10
        10/4/5 148
        10/4/12 88
        10/4/13 119
        10/4/10 1
        10/4/11 54
2.1.100
        5/0/60 1
2.1.103
        5/0/60 2
        5/0/53 5
2.1.11
        1/0/206 31
2.1.12
        1/0/206 46
2.1.6
        1/0/53 5
2.1.8
        1/0/206 7
2.1.9
        1/0/206 18
2.10.1
        1/7/120 2
2.10.10
        1/7/17 2
2.10.100
```

Figure 145: Result

The left column shows the source address, the column in the middle the destination groupaddress and the right column the amount how often the source address has sent a telegram to the specific destination group address. Already, the picture above shows that the amount of the packets to a specific group address varies. It ranges from only 1 exchanged packet up to 148 packets.

6.6 Controlling the Installation from Everywhere

From the perspective of an attacker, staying undiscovered as long as possible represents an important goal. The longer an attacker is undiscovered the more information can be gained. This goal represents a very complicated challenge.. First, a physical connection has to be established. This means that the attacker has to enter the building and find some access point within a building or find some access point outside of a building. In the case of the new E-building described in chapter 6 this task wasn't very complicated, but related to other building this could be a tough challenge. Furthermore, there are some requirements the physical access point has to meet. The access point should be a place

where additional connected devices can't be discovered after a few minutes. If an attacker removes a switch, which is connected to the bus line and connects an additional device to it, the switch wouldn't fit in their fixture anymore and both devices would hang outside the wall. An access point inside of a storeroom would be much better.

In the case of the new E-building every lecture room contains a desk with different controlling devices and also a computer. These desks are placed next to the recesses in the ground. Devices, which are part of the attack setup, could be placed under them and easily connected to the bus line within the recess. No one would discover the devices unless looking under the desk, which would not happen very often.

An attacker needs a physical access to the bus once and connect additional devices. If this happens he can control the installation through the Internet.

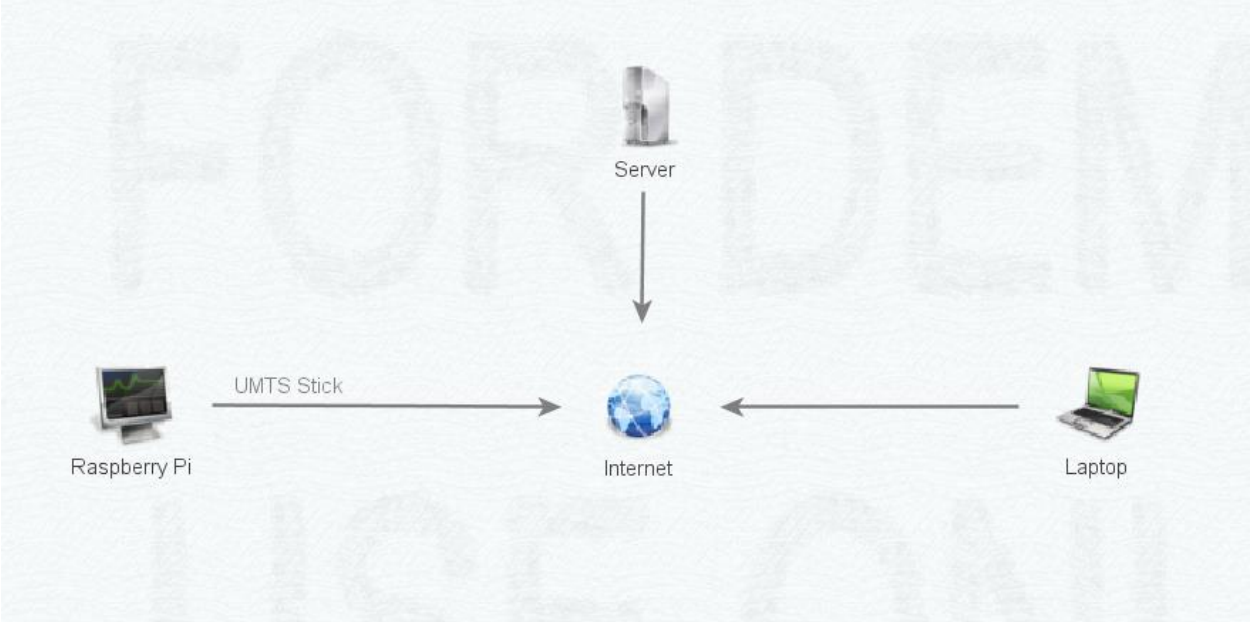


Figure 156: Attack setup UMTS

6.6.1 Attack Setup

The attack setup contains different devices. These are listed below:

- Siemens IP Viewer
- Raspberry Pi with an Huawei UMTS dongle
- Server available via the Internet
- Control Station e.g. Laptop

For establishing a connection to the bus the Siemens IP Viewer is used. The bus line is directly connected to the device as well as the Raspberry Pi. On the Raspberry Pi the EIBD software is running for sending and reading packets that are exchanged on the bus line. As a connection to the Internet the UMTS dongle Huawei E355 is connected to the Raspberry Pi.

The problem, which occurred with this setup, was that UMTS providers supply an internal IP address to the devices. If you are using an UMTS dongle to connect to the Internet the device doesn't get a public IP. Furthermore, all connections

to the Internet are routed via a central gateway. Connections from the Internet to the device with the UMTS dongle, in this case the Raspberry Pi, can't be established. All connections have to be initiated by the Raspberry Pi. The problem was to make the device available via the Internet. Some further research led to the possibility to use a reverse shell. For that reason some kind of server available on the Internet is required, which serves as SSH handler.

The solution for this problem looks like follows. First the tool autossh has to be installed on the Raspberry Pi. Autossh provides various features, like reconnecting to a SSH session when the connection was lost. If this task is done autossh has to be started with the following command:

```
autossh –M 9999 –R 3001:localhost:22
user@server_on_the_internet
```

The option –M indicates a port which is used to monitor the connection. As the second option –R is supplied and indicates that all further parameters are directed to the normal SSH command. Now a SSH connection to the server on the Internet is established.

```
ssh user@server_on_the_internet
```

The next part will be to connect from the control station to the server on the Internet. For this purpose ssh will be called in the standard way with the user and server combination. In this state the Raspberry Pi as well as the control station are connected to the server. To map the SSH tunnel established from the control station to the server, with the SSH tunnel from the Raspberry Pi to the server the following line has to be executed.

```
ssh –p 3001 user@localhost
```

At this point the Raspberry Pi can be controlled via the control station, for example a laptop. Since the software on the Raspberry Pi for receiving and sending packets is running, the KNX installation is now fully under an attackers control.

6.7 Result

Besides the missing restriction of physical access to the bus the decisive mistake, which was done in the case of the E-building, all devices are connected to one single bus line. This fact results in a compromise of the whole installation if physical access is established.

7 Further Attack Scenario

Like in any other situation or used technology there are different possibilities for executing an attack. Many possibilities are limited to the given conditions but there are often attacks, which work nearly every time. The next chapter describes some possible attacks and should give some view into which possibilities exists to attack a KNX installation.

7.1 Close Down the Bus

A very simple attack against the KNX bus is to close down the bus. This represents a very simple attack but it can have a heavy impact. As described in chapter 6.1, getting physical access to the bus itself could be an easy requirement. If the bus is also installed at the outside, for example a motion detector at the courtyard entrance, a potential attacker doesn't need to get inside a building. To identify if the device uses the KNX technology, demounting it is the best way. After the device is demounted you can see which cables are used to supply electricity to it. In Europe a wiring for a normal device requires two or three pathways for providing it with electricity depending on the fact whether the device needs grounding or not. These are a blue wire (neutral), a brown wire (phases) and the third green/yellow wire (protective earth/ground)[16]. Instead of these 2 or 3 lines KNX installations use four lines. A KNX line contains 4 strands, a red, a black, a yellow and a white one. If the demounted device is connected to a red and a black one, the probability that these are bus cables is very high. Now, if the attacker disconnects both strands from the device and closes them down, the whole bus line is jammed. This means that all other devices are jammed. If someone wants to turn on the light, this functionality doesn't work like every other functionality that is connected via this line. The probability that the whole KNX installation in a house is affected by this attack is quite high. A bus line could contain up to 64 devices. In a normal house 64 devices can suffice a big amount of functionality, for that reason the probability that all devices are connected via one bus line is very high. The impact of such an attack could affect also security-related devices, which are connected to the bus line, alarm systems for example.

7.2 A Practical Attack

Alessio Antonini, Federico Maggi and Stefano Zanero described in 2014 in their paper "A practical Attack Against a KNX-based Building Automation System" an approach for attacking a KNX installation[17]. The overall goal of their attack is „to be able to reset an arbitrary actuator to its default settings, take control of it, and in general disable the KNX-managed devices"[18]. The following part describes the steps that are required for a successfully executed attack and originated in their paper.

At the initial programming of the KNX installation, a four-character password can be set to protect the installation. Instead of securing the whole installation with this password, like sending packets to actuators, which will be executed by them, only a new unauthorized programming of a KNX device is prevented by this feature. Therefore, a packet crafted by an attacker and send to an actuator, which has the goal to deactivate an alarm system, will be executed normally. Nevertheless, the authors also included the condition that this password is set. Their attack looks like follows:

First they created a kind of malware, which has to be passed to a victim who is a part of the IP network within a building. This task could be done via an email attachment, drive-by downloads or portable USB devices. Once the malware is delivered to any machine in the IP network, it scan's for KNXnet/IP – Interfaces. If a KNXnet/IP – Interface is located in the network, it will respond. The first goal, getting access via the IP network to the bus, is accomplished.

---

[16] *Wikipedia contributors: Electrical wiring, (2015, at 13:10.)*
[17] *Alessio Antonini, Federico Maggi, Stefano Zanero: A Practical Attack Against a KNX-based Building Automation System, (2014)*
[18] *Alessio Antonini, Federico Maggi, Stefano Zanero: A Practical Attack Against a KNX-based Building Automation System, (2014), P. 58 Line 11-13*

As soon as the KNXnet/IP device is identified KNX telegrams to the bus can be sent. The further steps differ depending on whether the KNX installation is secured by a password or not. If a password is set to protect the reprogramming of the installation, the malware could find it out too.

7.2.1 Enabled Password Protection

After determination that a password is set, the malware will send KNX telegrams to the bus randomly. This behavior results in random actions by the devices which are connected to each other. For example, lights will be turned on and off or blinds will shut down and up. This procreated behavior will result in some maintenance action by an administrator of the installation, because of the malfunction. The probability that the administrator will reprogram the installation is quite high. To get this done he has to enter the password, which will be eavesdropped by the malware. Because of the eavesdropped password the attacker has also the possibility to program the devices arbitrarily. Now the installation can be completely controlled by the attacker.

7.2.2 Disabled Password Protection

If no password is set to protect the installation, the attacker has full access to every device that is connected to the line. He can directly send telegrams to actuators or reprogram the installation. Further he can set a password to lock out the legitimate administrator.

8 Securing the Implementation

The secure operation of a KNX installation is tied to some conditions. This chapter gives some basic recommendations for using KNX as a secure medium for home automation.

8.1 Network Level

On the network level different mechanisms can be implemented to ensure that only authorized entities can communicate with the KNX installation. These mechanisms are described in the following part.

8.1.1 VPN

The circumstances, that KNXnet/IP provides no security at all lead to the possibility to install security on a lower layer, for example on the IP layer. Especially when the KNX installation should be available from the Internet, this technique should definitely be used. On the third layer IPsec could be used to secure the communication and to restrict access to the KNX installation for valid users. IPsec ensures the following security properties:

- Confidentiality
- Integrity
- Authentication

IPsec was designed for providing a secure communication channel via an insecure network. This advantage should be used for KNX installations, if the installation should be reachable via the Internet. Especially when the KNXnet/IP-Interface comes with an embedded webserver to control the KNX installation through a web interface.

The reviewed device provides the possibility to reach the device via a VPN through the Internet. Nevertheless, most of the devices are reachable publicly without being part of a VPN.

8.1.2 VLAN

In addition to an external attacker an internal attacker could be a potential threat, too. To ensure access to a KNXnet/IP-Interface only for administration tasks or re-programming of the KNX installation, the KNXnet/IP-interface could be transferred into a separate network. Setting up different VLANs, which are assigned to different tasks, could accomplish this. With this approach the clients who are located in VLAN 1 can't communicate with clients in another VLAN, because of the logical network separation.

8.2 Physical Level

Besides ensuring security on the network level it's also possible on the physical level. How this can be realized is described in the following chapter.

8.2.1 No Bus to the Outside

In chapter 7.1 the threat of the bus line being reachable from the outside was described. The first task, which an attacker has to do, is to gain physical access to the bus line. In the case that the bus line is reachable from the outside, the attacker doesn't have to enter a building. This fact decreases the possibility for getting detected significantly. An important point when using a bus line at the outside is that this bus line isn't easily accessible. In case of a motion

detector, the device should be attached to the building at a high level. This requirement is listed in the "KNX Security Checklist" as the second point [19].

Another recommendation is to separate the bus line inside from the bus line that is used outside with a line coupler. Should the attacker have gained physical access to the bus line outside, they can control the devices, which are connected to the affected bus line. All other bus lines wouldn't be affected and can't be controlled by the attacker.

8.2.2 Line Coupler

At the initial programming of the installation it will be analyzed which devices are communicating with each other. Based on this information, filter tables are generated and set up on the line coupler. Therefore line couplers can be used to filter KNX telegrams on the bus. If someone wants to install KNX devices outside the house, for example a motion detector, an additional line should be used for these devices. This line can be connected via a line coupler to the bus line where all devices from the inside are connected. A possible attacker can't gain access to the inside bus line anymore, because of the filtered traffic by the line coupler. The attacker can only control the devices connected to the bus line outside, while, all further devices aren't affected. This approach decreases the attack surface and should be used if a bus line at the outside is desired.

8.3 System Level

As a further step the following precautions, which are located at the system level, should also take place.

8.3.1 Latest Firmware

Nowadays the rate of discovered vulnerabilities in software products increases from day to day. As seen in chapter 5 KNX devices are also affected by vulnerabilities in this case cross-site scripting.

Using the latest firmware ensures that possible or already discovered vulnerabilities are not present in the own KNXnet/IP-Interface which comes with a web application and other kind of software. A high patch level minimizes the attack surface significantly.

8.3.2 Strong Authentication

If any kind of authentication mechanism is implemented in a KNX installation, like described in chapter 5 use strong authentication methods. If a password is required the following requirements should be complied with:

- At least 10 characters
- A periodical change
- Upper case together with lower case characters
- Numbers
- Special characters

These best practices ensure that potential brute force attacks are impossible, because of the required time to guess the correct password.

In relation to the tested device, the option that usernames will not be displayed should be activated. Because of this a potential attacker has to brute force two strings, which increases the required time to guessing both strings.

[19] *KNX Association: KNX Security Checklist*

8.3.3 Check Logs

Log files could contain very useful information. Besides the normal traffic, which occurs in the daily operation, failures and errors are recorded as well. If a potential attacker sends manipulated telegrams via the bus to control devices like lights he needs to state a source address. Normally the addresses, which are communicating to each other, are always the same. In this specific case the source address isn't valid related to the destination address. Another aspect would be that this source address was not used before. All this very important data is recorded in the log files and helps to discover manipulation. Because of this reason, log files should be checked on a regular basis. This ensures that the installation is running without errors and suspicious traffic could be detected.

8.4 KNX Security Checklist

As already seen the KNX Association provides amongst other documents, the "KNX Security Checklist"[20]. Within the document basic security requirements are surveyed, for example, whether cables in- or outside of the building are easily accessible. In the case of the E-building this check would fail, because in nearly every room cables are accessible via a recess as described in chapter 6.1. The checklist should be a very important part of the acceptance test as soon as a KNX installation is put into operation. Common mistakes can be avoided and a secure installation is ensured.

8.5 Additional Security Mechanisms

Besides the devices which have an authentication feature factory-adjusted, there are devices which haven't. In this case it would be the best to deploy an own authentication mechanism. One possibility was described in the magazine "c't wissen – Smarthome"[21].

For this purpose the authors used the software lighttpd[22].

The developer, Jan Kneschke, describes lighttpd as follows:

> *"lighttpd is a secure, fast, compliant, and very flexible web-server that has been optimized for high-performance environments. It has a very low memory footprint compared to other webservers and takes care of cpu-load. Its advanced feature-set (FastCGI, CGI, Auth, Output-Compression, URL-Rewriting and many more) make lighttpd the perfect webserver-software for every server that suffers load problems"[23].*

Lighttpd doesn't require much power. In this example a Raspberry Pi will be used as device because the performance would be sufficient. For adding the additional security features described in the following parts, lighttp will be deployed

[20] *KNX Association: KNX Security Checklist*
[21] *Christian Heise, Ansgar Heise, Christian Persson: Smart Home, (2014)*
[22] *Jan Kneschke: lighttpd, (2014)*
[23] *Jan Kneschke: Lighttpd*

as a proxy. A proxy is some kind of software, which serves as an intermediate service and can be used to filter and/or inspect requests. The picture below shows an example setup.

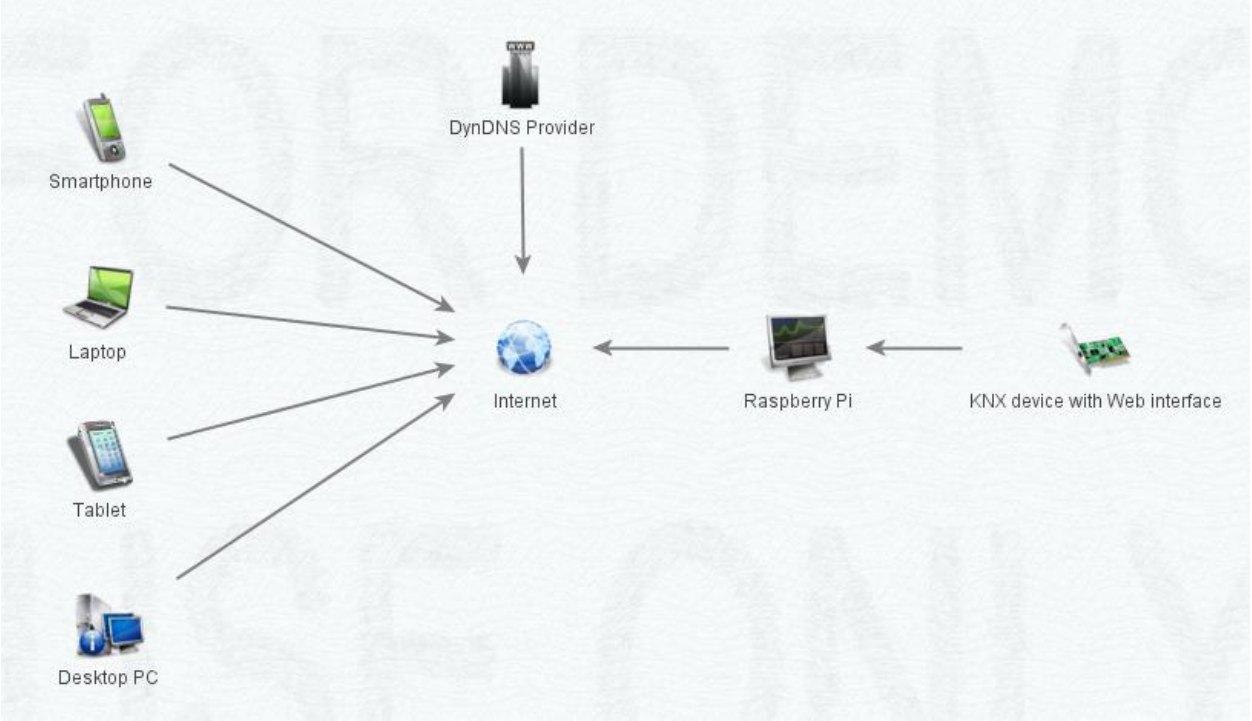


Figure 167: Example Implementation

If a user gets a dynamic IP, which changes from time to time, he can make use of a dynamic DNS provider. Therefore the dynamic DNS provider has to be indicated to the router. On the side of the dynamic DNS provider, a DNS name has to be indicated which would be mapped to the connection of the user's router, the dynamic IP. The mapping is done at regular intervals thereby the router is permanently available through the given DNS name. At this point the Raspberry, which provides the proxy services, isn't reachable from the Internet because the router will block the port, in the case of a web interface 80 or 443. To ensure that a connection could be established to the Raspberry Pi behind the router, port forwarding hast to be set up. This option has to be set up on the router; the description differs between different router vendors.

The next step is to install and set up lighttpd on the Raspberry. Depending on the operating system this task can be done in different ways. In this example, we installed through the shell. For installing lighttpd the following command has to be used:

"sudo apt-get install lighttpd"

8.5.1 Enable Proxy Mode

If lighttpd was installed, it's running like a normal webserver. The desired functionality, proxy mode, has to be activated in the configuration file, which can be found at "/etc/lighttpd/lighttpd.conf". Uncommenting the line "mod_proxy"[24] will

[24] *Jan Kneschke und Lighttpd Community: The Proxy Interface*

activate proxy mode. Adding the following lines will initiate the proxy to forward all connections to the specified port to the host with the corresponding destination IP address.

```
proxy.server = ( "" =>
                        ( "localhost" =>
                          (
                           "host" => "ip_knx_web_interface",
                           "port" => port_the_web_application_is_reachable_through_e.g._80
_or_443
                          )
                        )
                )
```

Now the KNX web interface is reachable via the Internet through a DNS name, which forwards all packets via the proxy to the KNX web interface.

8.5.2 Setup Authentication

At this point no security features are activated, just the basic functionality, which is required for establishing a connection to the web interface, is set up. For activating the authentication mode, the line "mod_auth"[25] has to be uncommented. Authentication can be realized by adding the following lines to the configuration file. These lines ensure that requests are only processed if a successful login has taken place previously.

```
auth.require = ( "" =>
                        (
                         "method" => "basic",
                         "realm" => "Proxy",
                         "require" => "user=admin"
                        ),
                         "/server-config" =>
                         (
                          "method" => "basic",
                          "realm" => "Server Config",
                          "require" => "user=admin"
                         )
                )
```

The method to authenticate to the proxy is basic and the information which will be displayed is Proxy, and the user who has the rights to login is admin. Users who are allowed to login have to be indicated in a separate file in the following style: username:password. As a last step, the lighttpd proxy has to be restarted with the following command, after the restart indicated users can login successfully:

"/etc/init.d/lighttpd restart"

[25] *Jan Kneschke und Lighttpd Community: Module mod_auth - Using Authentication*

8.5.3 Using Encryption

As before, to make use of encryption, additional software is required. Installing OpenSSL[26], an open source toolkit that provides amongst other things encryption features, is the first task. For the goal of encrypting exchanged traffic between the proxy and the user who wants to authenticate, a certificate is required. A certificate is a data structure, which provides properties about, in this case the proxy, like authenticity or integrity information. The following command generates a self-signed certificate that can be used for the proxy. In the generation process different questions are asked. One of them asks for the "Common Name", which has to be answered with the dynamic DNS name.

"openssl req –new –x509 –keyout cert/server.pem –nodes –out cert/server.pem –days 365 –config ./openssl.conf"

[26] *Mark J. Cox, Dr. Stephen Henson, Ben Laurie, Andy Polyakov: OpenSSL,*

9 Conclusion

At the beginning of the document the core advantages, as well as the field of application of home automation systems, were explained. After a short introduction into the area of KNX, the used testing environment was presented. The testing environment includes different devices, which are explained in detail. In a further step a case study of the tested device, which provides visualization for comfortable controlling of an installation, was done. On this device various vulnerabilities were revealed including cross-site scripting flaws.

Because of the fact that these devices can be available via the Internet to control the smart home independently of your current position, a script to search for these devices was developed. Some tests revealed that many devices could be found through the Internet. Many of them are operating without authentication mechanisms.

As a further step a building equipped with the KNX technology was attacked. Packets exchanged on the bus were recorded and analyzed. With the information gained blinds as well as lights could be controlled. At the end of the document different approaches to secure a home automation system are given, related to different layers.

The tests revealed that security isn't implemented in the way it should be. Therefore the assumption made that there are security weaknesses in home automation systems was confirmed. Especially if devices, like the device we tested, are part of the installation. In case of using such devices, which come with an embedded webserver, the requirement of physical access to the bus becomes unnecessary for an attacker.

Further, the statements in the standard don't represent the reality. There are tools for investigating packets and analyzing them in a further step, for example Wireshark.

It is highly recommended to follow the KNX security checklist to secure an installation as well as possible.

10 APPENDIX

10.1 References

[1] Andreas Streim, Tobias Arns. (2014, 23. Oktober). Connected Home. Eine Million Smart Homes bis 2020, BITKOM. Available at *http://www.bitkom.org/de/presse/81149_80552.aspx. Accessed 16.02.2015*.

[2] KNX Association (KNX Association, Hrsg.). (2014). ETS5. Engineering Tool Software, KNX Association. Available at http://www.knx.org/knx-de/software/ets/ETS5/herunterladen/index.php. Accessed 05.02.2015.

[3] KNX Association. (2015). Checklist. Step-by-step project management Part 1: Start of project. Checklist for implementing an electrical installation with KNX, KNX Association. Available at http://knx.org/media/docs/downloads/KNX-Flyers/Checklist%20Step-By-Step%20Project%20Management/Checklist-Part-1_en.pdf. Accessed 05.02.2015.

[4,5] EN 13321-2:2012 (03.2013). Offene Datenkommunikation für die Gebäudeautomation und Gebäudemanagement - Elektrische Systemtechnik für Heim und Gebäude: Beuth Verlag GmbH.

[6] Hochschule Offenburg. (2014). Neues E-Gebäude eingeweiht. Offenburg. Available at *http://www.hs-offenburg.de/news-detail/archive/2014/december/article/neues-e-gebaeude-eingeweiht/*.

[7] Raspberry Pi Foundation. Raspberry Pi. Available at http://www.raspberrypi.org/. Accessed 02.02.2015.

[8] Martin Kögler: EIBD [Computer software]: Martin Kögler / TU Wien. Available at https://www.auto.tuwien.ac.at/~mkoegler/index.php/eibd.

[9] Martin Kögler. EIBD - Zitat. Available at https://www.auto.tuwien.ac.at/~mkoegler/index.php/eibd. Accessed 04.02.2015.

[10,14,15] Martin Kögler. (2008, 18. Dezember). Free Development Environment for Bus Coupling Units of the European Installation Bus. BCU SDK Edition. Available at http://www.auto.tuwien.ac.at/~mkoegler/eib/sdkdoc-0.0.5.pdf. Accessed 04.02.

[11] Michael Albert. (2014). Raspberry Pi: eibd with a KNX USB Interface. Available at http://michlstechblog.info/blog/download/shell_scripts/install_eibd_usb.sh. Accessed 03.04.2015.

[12] (Internet Systems Consortium, Inc., Hrsg.). (2015, 20. März). ISC DHCP. Enterprise Grade Solution for Configuration Needs. Available at https://www.isc.org/downloads/dhcp/. Accessed 27.03.15.

[13] Gordon Lyon. (2015). Nmap. Nmap Security Scanner. Available at http://nmap.org/. Accessed 04.02.2015.

[16] Wikipedia Contributors. Electrical Wiring. Available at https://en.wikipedia.org/w/index.php?title=Electrical_wiring&oldid=672022626. Accessed 22.07.2015.

[17, 18] Alessio Antonini, Federico Maggi, Stefano Zanero. (2014). A Practical Attack Against a KNX-based Building Automation System. DOI: http://dx.doi.org/10.14236/ewic/ics-csr2014.7. Available at http://ewic.bcs.org/upload/pdf/ewic_icscsr14_paper7.pdf. Accessed 31.01.2015.

[19, 20] KNX Association. KNX Security Checklist. Checklist for increased security in KNX installations. Available at http://knx.org/media/docs/downloads/KNX-Flyers/KNX-Security-Checklist/KNX-Security-Checklist_en.pdf. Accessed 04.02.2015.

[21] Christian Heise, Ansgar Heise, Christian Persson (Hrsg.). (2014): Smart Home [Themenheft]. c't Wissen. Hannover: Heise Zeitschriften Verlag GmbH & Co. KG (IT-Haustechnik sinnvoll einsetzen).

[22] Jan Kneschke. (2014, 12. März). lighttpd. Available at http://www.lighttpd.net/. Accessed 08.02.2015.

[23] Jan Kneschke. Lighttpd. Available at http://redmine.lighttpd.net/projects/lighttpd. Accessed 08.02.2015.

[24] Jan Kneschke & Lighttpd Community. The Proxy Interface. mod_proxy. Available at http://redmine.lighttpd.net/projects/1/wiki/Docs_ModProxy. Accessed 08.02.2015.

[25] Jan Kneschke & Lighttpd Community. Module mod_auth - Using Authentication. mod_auth. Available at http://redmine.lighttpd.net/projects/lighttpd. Accessed 08.02.2015.

[26] Mark J. Cox, Dr. Stephen Henson, Ben Laurie, Andy Polyakov: OpenSSL [Computer software]: OpenSSL Project. Available at https://www.openssl.org/.

10.2 Disclaimer