1 2 9 0

UNIVERSIDADE D
COIMBRA

Vitor Manuel Garcia do Nascimento Graveto

# SECURITY AND SAFETY FOR BUILDING AUTOMATION AND CONTROL SYSTEMS

Tese no âmbito do Doutoramento em Engenharia Informática, Especialidade em Arquiteturas, Redes e Cibersegurança, orientada pelo Professor Doutor Paulo Simões, e pelo Professor Doutor Tiago Cruz, e apresentada ao Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

Dezembro de 2022

Department of Informatics Engineering
Faculty of Sciences and Technology
University of Coimbra

# Security and Safety for Building Automation and Control Systems

Vitor Manuel Garcia do Nascimento Graveto

PhD in Informatics Engineering
PhD Thesis submitted to the University of Coimbra

Advised by Prof. Dr. Paulo Simões
and Prof. Dr. Tiago Cruz

December, 2022

Departamento de Engenharia Informática
Faculdade de Ciências e Tecnologia
Universidade de Coimbra

# Segurança e Proteção para Sistemas de Controle e Automação de Edifícios

Vitor Manuel Garcia do Nascimento Graveto

Doutoramento em Engenharia Informática
Tese de Doutoramento apresentada à Universidade de Coimbra

Orientado pelo Prof. Dr. Paulo Simões
e pelo Prof. Dr. Tiago Cruz

Dezembro, 2022

This work was partially supported by the following projects:

**ATENA** – Advanced Tools to assEss and mitigate the criticality of ICT compoNents and their dependencies over Critical InfrAstructures; financed by H2020 – Innovation action, within the scope of the project (H2020-DS-2015-1 Project 700581). ATENA aims at achieving the desired level of Security and Resilience of Critical Infrastructures with Industrial and Automation Control Systems, while preserving their efficient and flexible management.

**Smart5Grid** – Automated 5G Networks and Services for Smart Grids; co-financed by European Regional Development Fund (ERDF) of the European Union (EU) through the POCI – Competitiveness and Internationalization Operational Program (COMPETE 2020), of Portugal 2020 (POCI-01-0247-FEDER-047226); Smart5Grid aims to profit from the advantages of using 5G networks by energy operators (TSOs and DSOs).

Cofinanciado por:

COMPETE 2020 PORTUGAL 2020 UNIÃO EUROPEIA Fundo Europeu de Desenvolvimento Regional

# Acknowledgments

My journey towards realising this doctoral course began with the migration from civil engineering to informatics engineering by completing a second degree. This work was only possible due to the excellent reception and follow-up in the Department of Informatics Engineering from Coimbra Univerity and the wonderful teachers, researchers and staff. Thank you, everyone.

I want to thank my supervisors Prof. Paulo Simões and Prof. Tiago Cruz, for their tireless and unconditional support. Their valuable contributions, incentives and advice are at the heart of the success of this doctoral work. I also emphasise their human abilities and the friendship developed over these years, which will undoubtedly last forever.

I would also like to thank my laboratory colleagues Rui and Jorge, who always had a word of encouragement and support in the most challenging moments. My gratitude to other friends and researchers that helped to create a group. In addition to research, that also extended to social and human relationships. I am thinking of David, Karima, Ricardo, Marcelo, Paulo, Ngongo and Joca.

My thanks to my many friends, especially Reinaldo, Marco, Pedro, Javier and Luís, who accompanies me in moments of relaxation and leisure, namely in the practice of Golf.

Finally, I would like to thank my family, consisting of my wife Cristina and my children Catarina and Manel. In this transitional phase of my life, they have always supported and encouraged me. This journey would never have been possible without their unconditional support.

# Abstract

Building Automation and Control Systems (BACS) are traditionally based on specialized communications protocols, such as KNX or BACnet, and dedicated sensing and actuating devices. Despite the increased awareness about the security risks associated with BACS, there is generally a need for more security tools for protecting this particular breed of cyber-physical systems. These threats are further aggravated as general-purpose security tools typically cannot cope with specific BACS requirements and technologies, thus calling for the development of domain-specific approaches, as is the case for the KNX Secure initiative, led by the KNX Association.

Nevertheless, despite the advances brought by KNX Secure and similar initiatives, there is still a considerable gap between the security needs of BACS and the solutions available. This thesis addresses this gap by proposing a Network Intrusion Detection System (NIDS) designed specifically for BACS. This NIDS is protocol-agnostic and can potentially support different BACS protocols and technologies, such as KNX, BACnet, Modbus or mixed ecosystems, without loss of generality. We also present a specific proof-of-concept implementation of this NIDS concept for KNX – one of the more widespread BACS protocols. To this purpose, a real-world KNX deployment was used to showcase and evaluate the proposed approach.

The present work also explored the vulnerability of using a BACS network to exfiltrate sensitive data from an air-gapped space. There are presented and validated two different methodologies and discussed possible countermeasures to mitigate the identified problem.

**Keywords:** Home Automation, Building Automation and Control Systems, BACS, NIDS, Smart Building, Security, Safety, KNX.

# Resumo

Os sistemas de automação e controle de edifícios são tradicionalmente baseados em protocolos de comunicação especializados, como KNX ou BACnet, e dispositivos específicos de detecção e atuação. Apesar da maior conscientização sobre os riscos de segurança associados a estes sistemas, genericamente existe uma falta de ferramentas de segurança para proteger esse tipo específico de sistemas ciberfísicos. Essas ameaças são agravadas ainda mais, pois as ferramentas de segurança de uso geral normalmente não conseguem lidar com os requisitos e tecnologias específicas da automação de edifícios. Assim, tem-se verificado a necessidade de conceber abordagens específicas para este domínio – a iniciativa KNX Secure desenvolvida pela Associação KNX é um desses exemplos.

No entanto, apesar dos avanços trazidos pelo KNX Secure e iniciativas semelhantes, ainda existe uma lacuna considerável entre as necessidades de segurança e as soluções disponíveis na automação de edifícios. Esta tese aborda esta lacuna, propondo um Sistema de Detecção de Intrusão de Rede, projetado explicitamente para a domótica de edifícios. Este sistema é agnóstico ao protocolo e pode potencialmente suportar, sem perda de generalidade, diferentes protocolos e tecnologias de domótica, como KNX, BACnet, Modbus ou ecossistemas mistos. Também apresentamos, como prova de conceito, uma implementação específica do sistema proposto, para KNX – um dos protocolos mais difundidos na automação de edifícios. A implementação e validação recorreu a uma casa de habitação real na qual existe uma instalação de domótica baseada em KNX.

O presente trabalho também identificou e explorou a vulnerabilidade de recorrer à rede de automação do edifício para exfiltrar dados confidenciais de um espaço potencialmente seguro devido ao seu isolado das redes de dados. São apresentadas e validadas duas metodologias distintas e discutidas possíveis contramedidas para mitigar o problema identificado.

**Palavras-chave:** Automação de Edifícios, Sistemas de Automação e Control de Edifícios, Domótica, Segurança, KNX

# Foreword

THE work detailed in this thesis was undertaken at the Laboratory of Communications and Telematics (LCT) of the Center for Informatics and Systems of the University of Coimbra (CISUC), within the context of the following projects and grants:

**ATENA** - Advanced Tools to assEss and mitigate the criticality of ICT compoNents and their dependencies over Critical InfrAstructures; financed by H2020 - Innovation action, within the scope of the project (H2020-DS-2015-1 Project 700581). ATENA project, aims at achieving the desired level of Security and Resilience of Critical Infrastructures with Industrial and Automation Control Systems, while preserving their efficient and flexible management.

The outcome of the design, experiments, and assessments of several mechanisms on the course of this thesis resulted in the following publications:

**Journal papers:**

- **(j1)** Graveto, V., Rosa, L., Cruz, T., and Simões, P. (2019). A stealth monitoring mechanism for cyber-physical systems. *International Journal of Critical Infrastructure Protection*, 24:126–143;
- **(j2)** Graveto, V., Cruz, T., and Simöes, P. (2022a). Security of building automation and control systems: Survey and future research directions. *Computers & Security*, 112:102527;
- **(j3)** Graveto, V., Cruz, T., and Simões, P. (2022). A Network Intrusion Detection System for Building Automation and Control. Submitted to IEEE Access (under review) ; and
- **(j4)** Graveto, V., Cruz, T., and Simões, P. Using KNX-based Building Automation and Control Systems for Data Exfiltration. Submitted to IEEE Internet of Things (under review).

**Co-advisory of MSc Thesis:**

- **(t1)** Silva, A. (2016). *Desenvolvimento de uma Infraestrutura Eletrónica de Comunicação para o Controlo Remoto de ”Casas Inteligentes” Usando KNX*. Msc thesis, University of Coimbra, DEEC. Within the scope of this master's thesis, the first prototype used for the connection to the KNX bus was developed. In the now presented PhD work it evolved into the KNX Bus Coupling Unit.

**Joint papers:**

- **(j5)** Rosa, L., Proença, J., Henriques, J., Graveto, V., Cruz, T., Simões, P., Caldeira, F., and Monteiro, E. (2017). An evolved security architecture for distributed industrial automation and control systems. In *European Con-*

*ference on Cyber Warfare and Security*, pages 380–390. Academic Conferences International Limited;

**Other contributions:**

- **(c1)** Graveto, V., Simões, P., and Cruz, T. (2022b). A dataset bundle for building automation and control systems security analysis. `https://dx.doi.org/10.21227/16a5-m134`. Last visited: 2022-10-06;
- **(c2)** Graveto, V., Development of Deep Packet Inspector (DPI) for KNX protocol in open source project Google Inc. (2022). GoPacket project repository. `https://github.com/google/gopacket`. Last visited: 2022-10-06;

# Table of Contents

# List of Figures

# List of Algorithms

# List of Tables

# Acronyms

| | |
|---|---|
| **AI** | Artificial Intelligence |
| **AL** | Automated Learning |
| **ANSI** | American National Standard Institute |
| **APCI** | Application Layer Protocol Control Information |
| **ASHRAE** | American Society of Heating, Refrigerating and Air Conditioning Engineers |
| **AUC** | Area Under Curve |
| **BACS** | Building Automation and Control Systems |
| **BACnet** | Building Automation and Control NETworks |
| **BAF** | Bandwidth Amplification Factor |
| **BC** | Backbone Coupler |
| **BCU** | Bus Coupling Unit |
| **BPIE** | Building Performance Institute Europe |
| **CA** | Accuracy |
| **CAN** | Controller Area Network |
| **COTS** | commercial off-the-shelf |
| **CSV** | Comma Separated Values |
| **D** | Device |
| **DA** | Destination Address |
| **DB** | Data Base |
| **DDoS** | Distributed Denial of Service |
| **DIY** | do it yourself |
| **DNS** | Domain Name System |
| **DoS** | Denial of Service |
| **DPI** | Deep Packet Inspector |
| **D-PPM** | Differential Pulse Position Modulation |
| **DWT** | Discrete Wavelets Transform |

| | |
|---|---|
| **ECC** | Elliptic Curve Cryptography |
| **EEPROM** | Electrically-Erasable Programmable Read-Only Memory |
| **EHS** | European Home Systems Protocol |
| **EIB** | European Installation Bus |
| **EIBA** | European Installation Bus Association |
| **ETS** | Engineering Tool Software |
| **EU** | European Union |
| **FMEA** | Failure Mode Effects and Analysis |
| **FN** | False Negaative |
| **FP** | False Positiver |
| **FPR** | False Positive Rate |
| **GA** | Group Address |
| **GPIO** | General Purpose Input/Output |
| **HA** | Home Automation |
| **HEMS** | Home Energy Management System |
| **HetIoT** | Heterogeneous Internet of Things |
| **HIBE** | Hierarchical Identity Based Encryption |
| **HVAC** | Heating, Ventilation and Air Conditioning |
| **IMEI** | International Mobile Equipment Identity |
| **IA** | Individual Address |
| **IACS** | Industrial Automation and Control Systems |
| **ICT** | Information and Communication Technologies |
| **IDS** | Intrusion Detection System |
| **IMECA** | Intervention Mode Effects and Criticality Analysis |
| **InstaBus** | Installation Bus |
| **IP** | Internet Protocol |
| **IT** | Information Technology |
| **IoT** | Internet of Things |
| **IR** | Infrared |
| **IWT** | Integrated Wireless Technology |
| **LAN** | Local Area Network |
| **LC** | Line Coupler |

| | |
|---|---|
| **LCD** | Liquid Crystal Display |
| **LonWorks** | Local Operating NetWorks |
| **LPDU** | Logical Protocol Data Unit |
| **LR** | Logistic Regression |
| **M2M** | Machine-to-Machine |
| **MQTT** | Message Queuing Telemetry Transport |
| **MSRP** | Manufacturer's Suggested Retail Price |
| **NIDS** | Network Intrusion Detection System |
| **NN** | Neural Network |
| **OS** | Operating System |
| **OSI** | Open System Interconnection |
| **PCAP** | Packet Capture |
| **PDR** | Packet Delivery Rate |
| **PL** | Powerline |
| **PLC** | Programmable Logic Controller |
| **PoC** | Proof of Concept |
| **PPM** | Pulse Position Modulation |
| **PS** | Power Supply |
| **PWM** | Pulse Width Modulation |
| **Recall** | True Positive Rate |
| **RF** | Radio Frequency |
| **RFID** | Radio-Frequency IDentification |
| **ROC** | Receiver Operation Characteristic |
| **ROI** | Region of Interest |
| **SA** | Source Address |
| **SBC** | Single Board Computer |
| **SCADA** | Supervisory Control and Data Acquisition |
| **SDN** | Software-Defined Networking |
| **SGAM** | Smart Grid Architecture Model |
| **SHMC** | Smart Home micro-computers |
| **SIEM** | Security Information and Event Management |
| **SoA** | State of the Art |

| | |
|---|---|
| **SSL** | Secure Sockets Layer |
| **SSU** | Shadow Security Unit |
| **SVM** | Support Vector Machine |
| **TLS** | Transport Layer Security |
| **TN** | True Negative |
| **TP** | twisted pair |
| **TPo** | True Positive |
| **TP/UART** | twisted pair/Universal Asynchronous Receive Transmit |
| **TCP** | Transmission Control Protocol |
| **TCP/IP** | Transmission Control Protocol over Internet Protocol |
| **TPCI** | Transport Layer Protocol Control Information |
| **UART** | Universal Asynchronous Receive Transmit |
| **UI** | User Interface |
| **WAN** | Wide Area Network |

# Chapter 1

# Introduction

Contents

This chapter constitutes the introductory section of this thesis, which is focused on presenting and contextualising the work that was undertaken during its development. First, Section 1.1 presents the motivation and problem statement, focused on the main research question. Secondly, Section 1.2 presents the objectives that needed to be fulfilled to answer the main research question, also being key to plan and steer the thesis development process. The main outcomes of this dissertation are presented in Section 1.3. Finally, Section 1.4 outlines the remaining content of this thesis

## 1.1 Motivation and Problem Statement

Similarly to many automation-related domains, Building Automation and Control Systems (BACS) security is frequently handled superficially, often foregoing even the most basic protection mechanisms and policies. However, considering this situation as the outcome of negligent practices may be somehow hasty, as it is more the result of a natural evolution process than necessarily arising from carelessness. This is due to the fact that, right from its inception, traditional automation security (and also BACS) relied on a mix of air gapping and lack of component documentation, as well as the need for domain-specific knowledge, to deter potential attackers, something that was reinforced by the general perception that such infrastructures lacked appeal.

BACS are supported by industry standards and protocols like Building Automation and Control NETworks (BACnet) (see ASHRAE [2020]), KNX (KNX Association [2020a]) and/or Supervisory Control and Data Acquisition (SCADA) systems. The latter represent autonomous services or devices used to automate a single task normally included in IoT. BACS represent, in the broadest sense, all the existing automated mechanisms used on any building. As most popular BACS and IoT systems used in building automation become interconnected with general-purpose Local Area Network (LAN) and even to the Internet, there is an increased security risk by means of the exposure of potentially vulnerable vectors.

Thus, the fundamental question that must be addressed when it comes to BACS security can be stated in the form of two basic research questions:

- Which are the most pressing issues when it comes to BACS security, and what can be done to address them?
- How to identify the abnormal behaviour of BACS, supported by field-level monitoring?

These questions enclose the need to adopt an end-to-end take on the problem, from technology to solution, starting by acquiring familiarity with the associated technology landscape, proceeding to the analysis of its attack surface and vulnerable vectors, and proposing ways of addressing these risks, while safeguarding user/tenant security, safety and privacy.

Although many of the security issues tackled in the context of this thesis may also have an impact on safety, this latter aspect is not specifically addressed in a formal way, being deemed to be out of scope regarding the main research topic. Nevertheless, it should be acknowledged that any improvement in terms of the BACS tenant and integrator security posture, as well as the development of the required supporting infrastructure, does provide an added value in terms of safety.

## 1.2 Objectives

Starting from the initial formulation of the aforementioned research question there was the need to identify the specific associated objectives that had to be fulfilled to properly answer it, namely:

- To familiarize with the BACS technology landscape and identify related security problems/issues;
- To address the lack of tools for monitoring the traffic of building control and automation networks, a situation that needs to be addressed before tackling security issues;
- To research and develop *black-box* detection approaches solely based on the monitoring and observation of physical and device inputs/outputs, with no previous knowledge of the controlled system/process;
- To research and develop *white-box* detection approaches based on partial knowledge of the BACS processes (to predetermine the expected behaviour of the system);
- And to develop and publish datasets to support BACS security research and methodologies – in this scope, this thesis intends to contribute back to the research community by providing a dataset which will also be used to support its own research efforts for developing new BACS security approaches and methodologies.

Overall, identifying these objectives helped define a natural progression path that steered the thesis development efforts towards its conclusion.

## 1.3 Contributions

Taking into consideration the goals described above, this thesis has produced the following contributions:

- **Contribution 1, BACS dataset**. The literature review and the search on known dataset sources reveal the lack of datasets for the analysis and research of Building Automation and Control Systems. Furthermore, the need for datasets with possible threats and attacks on those systems was assessed, which led to the creation of a dataset bundle, published as **(c1)**, which was made publicly available. This dataset also supported some of the research efforts undertaken in the scope of this thesis.

- **Contribution 2, Proposal and architecture design of a monitoring framework for enhanced security and safety of BACS**. The identified absence of tools capable of dealing with BACS security and safety problems led to the proposal of a new framework to address this issue, published in **(j3)**. The proposed architecture enables different detection techniques based on rules and heuristics or supported by artificial intelligence algorithms. The developed framework also allows monitoring and visualising traffic in BACS networks. This work was also supported by **(j2)**, **(c1)** and **(c2)**.
- **Contribution 3, KNX DPI implementation for GoPacket**. The intention to use the GoPacket open source library (Google Inc. [2022]) for packet decoding in this thesis, led us to identify the absence of DPI for KNX protocol. As part of this work, DPI for KNX was developed and incorporates now that publicly available GoPacket library **(c2)**.
- **Contribution 4, Development of a PoC for KNX BACS**. The proposed architecture and respective framework were implemented in the form of a proof-of-concept targeting KNX-based BACS, which was later validated in a real environment. This work was published as **(j3)**, that was enriched with previous IACS experience as published in **(j1)** and **(j5)**.
- **Contribution 5, Data Exfiltration using a KNX BACS**. Secure physical spaces in institutions that handle critical, private or confidential information are common. The co-existence of these spaces with the management and automation system of the building enables the possibility of creating a covert channel for data exfiltration. When that BACS is supported by the KNX protocol, the exfiltration of data with malicious purposes is possible. This contribution investigates the required techniques and also some mitigation actions, and was published as **(j4)**.

Having discussed the goals and contributions of this research, the following section shows the outline of this thesis.

## 1.4 Outline of the Thesis

The remaining of this thesis is organized into six chapters. Chapter 2 introduces and presents the Building Automation and Control Systems landscape, encompassing an extensive literature review and identification of open issues and research opportunities, also providing the discussion of their main differences with regards to the IoT domain. This chapter will pay special attention to KNX BACS, which were targeted by the specific research and development efforts undertaken in the scope of this thesis, as well as by the Proof of Concept solution that was developed for this specific technology.

Chapter 3 introduces the motivation for this work reviewing BACS issues and known intrusion detection solutions in this and other similar scopes. This approach enabled the development of a new concept and requirements that guided the proposed reference architecture to handle the security and safety issues identified on those BACS.

Chapter 4 describes the Proof of Concept that was developed to showcase the proposed security and safety solution for a KNX BACS. The PoC objectives and architecture are detailed. A laboratory testbed was used to support the development and some initial tests (with further validation being presented in Chapter 5). Finally, the existing knowledge from previous work on Industrial Automation and Control Systems (IACS) security and safety that contributed to this solution is summarized.

Chapter 5 discusses the evaluation processes that was undertaken to demonstrate the pertinence and effectiveness of the developed solution. The techniques and methodologies deployed to monitor and detect the attacks on a BACS are presented along with the obtained results. The evaluation scenarios and the developed datasets creation process are also described. Finally, the chapter concludes with a discussion of the obtained results.

Chapter 6 presents two different techniques that enable the exfiltration of data from an air-gapped secure space using a KNX BACS network. This chapter also discusses how the previously proposed framework could be used to detect and mitigate these threats.

Chapter 7 concludes this document, offering a synthesis of the thesis and a reflection on possible research paths for extending this work.

# Chapter 2

# Background and State of the Art

## Contents

Building Automation and Control Systems (BACS) designate the mechanisms used to automate buildings' operations, such as climate control, lighting and access control. As such, traditional BACS encompass extensively automated buildings managed in an integrated manner, with the support of Supervisory Control and Data Acquisition (SCADA) systems and technical industry standards such as BACnet and KNX. More recently, the increasing adoption of Internet Protocol (IP)-connected, *IoT-like* devices for automating single tasks led to a substantial increase in the number of automated building functions (especially for the smart home domain), although rarely with extensive or integrated automation levels. The interconnection with building LAN and even the Internet comes with the cost of a broader exposition to attacks. Those can either begin inside the building or be initiated from anywhere outside of it.

In contrast with other domains that recently received substantial attention (e.g. IACS), the security of BACS has been addressed in a somehow more superficial and less structured manner. Nevertheless, recent security concerns are raising security concerns, combined with the fact that these systems are increasing interconnection with the building networks and the Internet.

This chapter, also partially published in Graveto et al. [2022a], provides a systematic survey of recent research and industry developments related to the security, safety and privacy of BACS. It also presents an overview of the existing threats and known attacks against them, as well as open issues and future research directions.

## 2.1 Building Automation and Control Systems

Technological evolution, as well as the search for increasing energy efficiency and occupancy comfort, have pushed for the introduction of BACS. Early classic BACS, introduced in the 1970s, were designed to be autonomous and isolated by nature. Their security was supposedly based on such isolation and on the use of proprietary technologies, both in the communication channels and in the operation of the microcontrollers involved in related control processes.

Meanwhile, the BACS community has joined efforts in standardizing and evolving related technologies. These efforts eventually led to the creation of protocols such as BACnet (ASHRAE [2020]), in the early 1980s, or European Installation Bus (EIB) in the late 1980s (Goossens [1998]). EIB, which was developed by the European Installation Bus Association (EIBA [2020]), later became the basis for the KNX specification, maintained and developed under the scope of the KNX Association [2020a], which was established in 1990s (with EIBA as one of its founders). In parallel, general SCADA protocols such as Modbus (MODICON [1996]) were also used to control Heating, Ventilation and Air Conditioning (HVAC) systems.

Since the 1990s, personal computers and the Internet evolved rapidly, becoming widely accessible. Information Technology (IT) has developed and remote management has become a reality. Ethernet and IP communications became widespread and, due to practical and economical reasons, they were gradually adopted in BACS environments, encapsulating the legacy protocols over Ethernet and/or IP communications. The interconnection between control and IT networks became a reality, enabling reduced costs and added convenience.

More recently, a noteworthy evolution of BACS is the increasing adoption of wireless communications (using both BACS-specific solutions such as wireless KNX and general purpose technologies such as ZigBee from Connectivity Standars Alliance [2021]), for convenience and cost reduction. In parallel, we have witnessed the increased adoption of consumer-grade commercial off-the-shelf (COTS) IoT devices for functions such as energy measurement, lighting, remotely controlled power outlets and blind control. While these IoT devices are often used in a less structured and integrated manner (when compared with classic traditional BACS), they have significantly lowered the entry barriers for the consumer market. More recently, cloud-based smart home solutions such as digital voice assistants (e.g. Alexa – Amazon [2014] – and Home Assistant – Google [2016]) have brought some sort of integration to the consumer-focused IoT landscape, although still far from the sophistication of the best professional-grade BACS solutions. Nevertheless, as those are sometimes viewed as building automation systems, this chapter will address them as part of the BACS landscape, although with less detail.

A common factor among all building automation solutions available nowadays is the lack of satisfactory security mechanisms. On the side of conventional BACS, this has mostly to do with the reliance on isolation and the lack of widespread knowledge about related protocols and technologies. Despite the recent introduction of security-oriented features such as encrypted communications, it is still relatively easy to maliciously interfere with the communication channels and bypass existing encryption and authentication mechanisms. Additionally, BACS sensors and actuators are prone to physical tampering, and the remote management features are often outdated and vulnerable to more sophisticated attacks. Moreover, there is also a general lack of security monitoring and management tools for BACS.

Regarding consumer-grade IoT equipment, there is also a considerable number of known issues and vulnerabilities, which have been at the source of recent security incidents (such as the Mirai botnet – Peterson [2019]). Moreover, the increasingly narrow frontier between building automation and personal user space introduced by these IoT-based scenarios (e.g. always-on microphones for voice assistants; widespread adoption of video-cameras inside the home) also raises substantial privacy concerns.

These security concerns are not exclusive to the BACS domain. Looking at areas with some similarities, such as IACS, the existence of legacy and/or highly specialized systems and their interconnection with the IT networks substantially increased the exposure to various threats. However, while for IACS such security issues have been the subject of intensive study, research and industry developments, the same does not apply to BACS. The security community has been paying much less attention to BACS ecosystems, which is often considered as a niche of IACS. This lack of attention reflects not only in noticeable less research efforts, but also in the absence of structured analysis of such research and open research issues.

Despite the attention it has received in the last years, there is a general lack of systematic literature reviews covering this topic. An extensive report sponsored by the ASIS Foundation Brooks et al. [2017] includes an analysis of BACS vulnerabilities and security management best practices (among more general aspects, such as a general BACS industry and market analysis and also BACS standardization), but focuses more in the mainstream industry landscape than on recent industry and research advances. An introduction to smart buildings security has been provided by Wendzel et al. [2017]. However, it is more a tutorial-style overview than a systematic literature review. Finally, a preprint from Ciholas et al. [2019] does provide a literature review of security for smart buildings, but it is not exhaustive enough, probably due to the authors' ambition of covering a broader spectrum of topics around the concept of smart buildings. In this chapter we bridge this gap by providing a comprehensive survey of research and industry developments specifically addressing the security of BACS.

The rest of the chapter is organized as follows. First, an introductory overview of BACS and related topics (Section 2.1.1) is provided, followed by the introdution of KNX systems (Section 2.1.2) and an analysis of the differences between IoT and BACS (Section 2.1.3). Next, a literature review is provided (Section 2.2), with a description of the methodology adopted for the systematic review (Section 2.2.1). Next, the State of the Art (SoA) is presented (Section 2.2.2), followed by the discussion of open issues and research opportunities (Section 2.3). Finally, Section 2.4 summarizes the chapter.

### 2.1.1 Overview

Smart buildings are automated buildings designed to increase safety and comfort, save costs and be environmentally friendly, while being able to interact with other smart buildings and service grids. These buildings are supported by control systems designated as BACS.

ISO 16484 EN [2014] specifies the phases required for BACS projects and the hardware needed to perform the tasks within a BACS, as well as the requirements for overall functionality and communication. According to these specifications, the building automation and communication is organized in three distinct layers: *Management*, *Automation* and *Field* (Figure 2.1).

The *Management Level* corresponds to the Information and Communication Technologies (ICT) network. This level entails the operation stations, monitoring and programming units, that process data and support the monitoring and management of the automation system.

The *Automation Level* normally represents a dedicated communication network

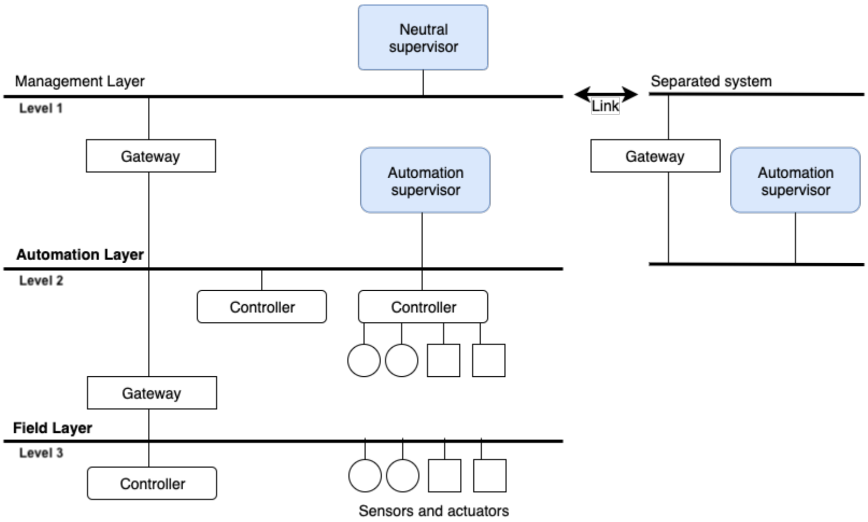


Figure 2.1: Three-layer BACS Architecture (adapted from Brooks et al. [2017])

used to interconnect the devices that have as main purpose the control (automation) of the building. This layer groups global building controllers such as chillers, energy production systems and air handling units.

The *Field Level* groups all the devices that are connected to the physical systems under control. These devices are generally self-contained physical units like sensors and actuators. In some situations they are connected to controllers in the *Automation Level*, communicating using specific protocols. In other situations they have their own processing and decision capabilities, to control local processes.

The Join Research Centre of European Commission recently published a report with a good SoA (see Serrenho and Bertoldi [2019]) that provides an introduction to the whole smart home ecosystems, with a focus on their energy implications. Several recent challenges are identified, with the *do it yourself (DIY)* mindset being one of the most important, since it enlarges the number of buildings with some sort of automation but eventually hampers the introduction of professional-grade, integrated BACS solutions.

In 2017, the Building Performance Institute Europe (BPIE) evaluated how ready was Europe for the Smart Building Revolution (see Groote et al. [2017]). It also associated the word *smart* with the concern of optimizing energy consumption and the use of clean renewable energy sources (see Figure 2.2). It created a function with several parameters for that evaluation, designated as Smart Build Environment Indicator.

Smart buildings include mostly two kinds of solutions: those that integrate the existing building automation systems (that we will generally refer to as BACS); and those that only have mostly independent assets that automate a specific task or device on the building (that we will designate, in the scope of this thesis, as IoT). This last one is mainly out of scope in the present thesis and only briefly reviewed in Section 2.1.3.

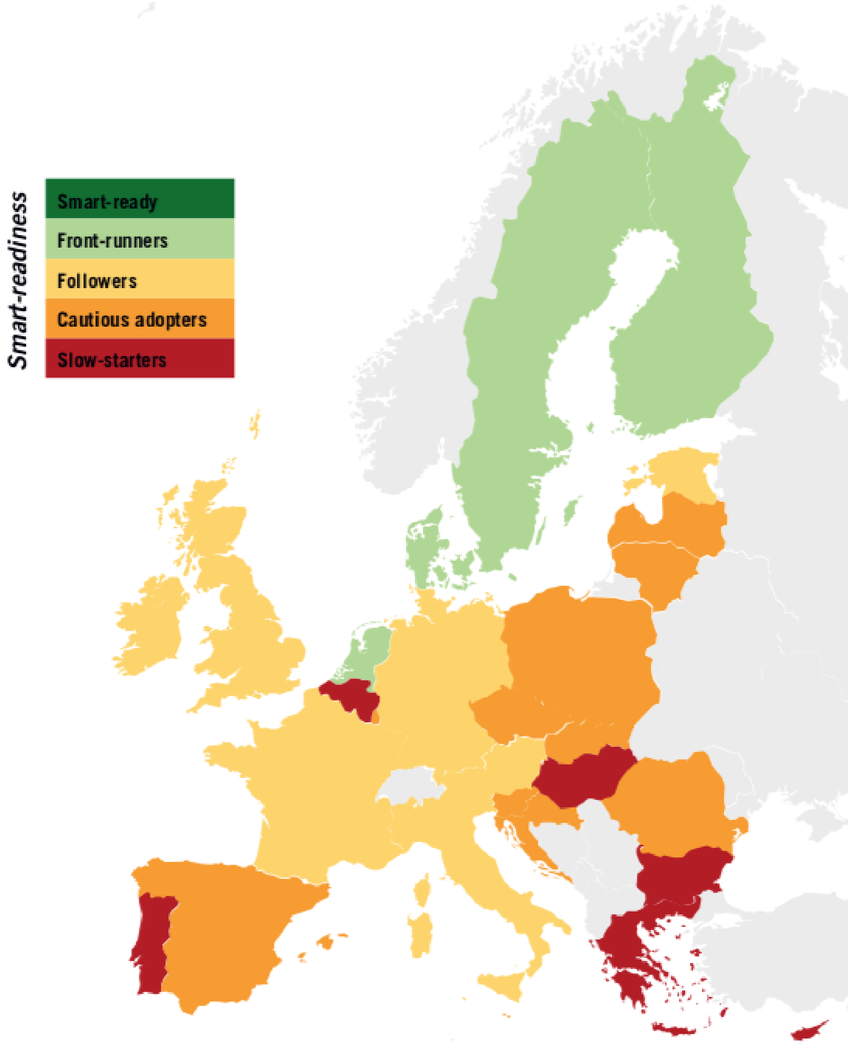


Figure 2.2: Smart-readiness across Europe – Groote et al. [2017]

The most commonly used standards and protocols in BACS are BACnet (see ASHRAE [2020]), Local Operating NetWorks (LonWorks) (see ANSI [2010]), KNX and Modbus (see MODICON [1996]).

*BACnet* was created in 1987 at Cornell University, to address the needs of building automation and control systems. It uses the Open System Interconnection (OSI) model and it became an American National Standard Institute (ANSI) standard under the auspices of American Society of Heating, Refrigerating and Air Conditioning Engineers (ASHRAE).

*LonWorks* was created in 1989 by the Echelon Corporation, and was accepted in 1999 as a standard for control networking (ANSI/CEA 709.1-B) by ANSI/CEA [2010].

*KNX* resulted from the association of the European Home Systems Protocol (EHS), BatiBUS and Installation Bus (InstaBus) (successor of EIB), and has been standardized through EN50090 (CENELEC [2012b]), ISO/IEC 14543 (ISO IEC [2006]) and EN13321 (CENELEC [2012a]), then extended to Chinese standard GB/T 20965 (China Machinery Industry Federation [2020]) and ANSI/ASHRAE 135 (ASHRAE [2016]). It is also based on the OSI model and extends the communication protocol to incorporate system commissioning and parameterization to allow interaction between devices from different manufacturers.

*Modbus* was developed in 1979 by Modicon (now Schneider Electric), as a serial communication protocol for Programmable Logic Controllers (PLCs), and was released as an open protocol in 2004. It is based on a master/slave architecture, using simple function codes, together with a plain data model. It is widely used in industrial automation for SCADA systems. In building automation it is mostly used to control equipment such as chillers, boilers and fans.

EEBus Initiative [2019] is also worth mentioning. It is a relatively recent effort with the prerequisite of exchanging information to coordinate and shift the energy between an intelligent power grid and the individual components in the households and buildings (e.g. photovoltaic system, battery storage, heating and electric vehicle) with the aim of creating a standardized language for energy. Its main objective is helping to achieve the climate goals by enabling transparency of energy demand; avoidance of load peaks and grid bottlenecks; use of flexibility on the supply and demand side; and use of decentralized energy generation. EEBus architecture is based on the Smart Grid Architecture Model (SGAM) (see CEN/CENELEC/ETSI [2012]) and represents a data communication standard forming the interface between in-house communication and energy supplier.

Besides these standards, there are many other standards and protocols with some relevance in the scope of BACS (some of them originally intended for different purposes), as illustrated in Figure 2.3. Nevertheless, for sake of conciseness, we will not address them in this work.

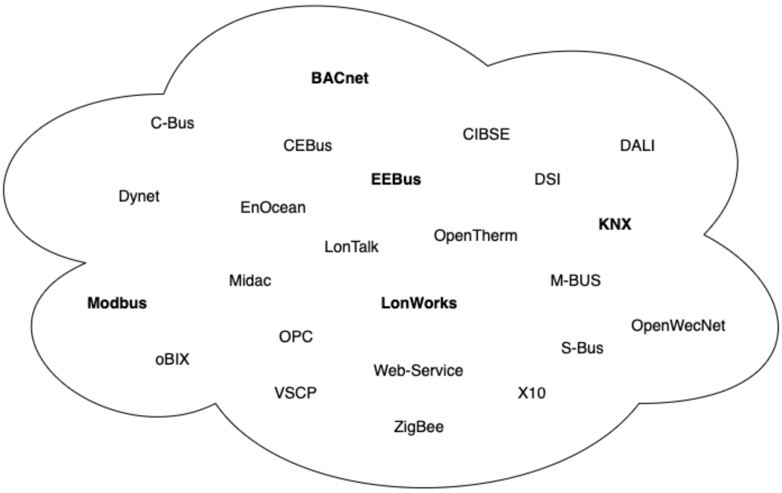


Figure 2.3: BACS Architecture Industry Standards and Protocols

Figure 2.4 is a diagram, proposed by Siemens Brooks et al. [2017], that represents the distribution of the most used protocols and standard and their relation with the building automation and communication layers. The bar *Web* represents all the different web services that exist either for the Automation and Management layers.

Domotics systems were initially designed to function autonomously and isolated from other systems. This is also true in BACS. However, the paradigm has changed with the constant integration of different services and functionalities associated with the use of ICT to exchange information between different protocols. These systems can no longer rely on isolation and obscurity for ensure proper security. This carries the cost of threats and potential attacks, not just from interconnected networks but also from the Internet in general.

When compared with ICT systems, the lifespan of BACS devices and systems are considerably longer, Such components are expected to reliably operate in a continuous and 24/7 basis during decades, often regardless of any security issues that may be eventually found. Due to the increasing reliance on those systems to ensure critical building functions, customers often have no other choice than to keep using them despite knowing about existing security problems. This situation has been somewhat worsened by the encapsulation of BACS protocols in IP, which has resulted in the inheritance of known security weaknesses from the ICT domain.

### 2.1.1.1 BACS-related literature

The BACS market is undergoing rapid expansion (see Khedekar et al. [2016]), with smart buildings being considered one of the main driving forces behind this trend. Conceptually, smart buildings are perfectly aligned with the scope of BACS, encompassing a series of requirements outlined in Hui et al. [2017], namely:

- heterogeneity;

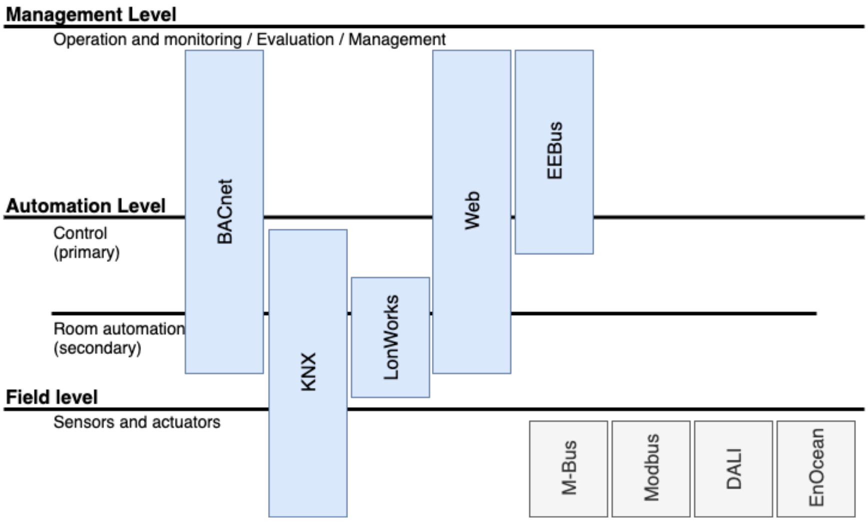


Figure 2.4: BACS Software Architecture (adapted from Brooks et al. [2017])

- self-configuration;
- extensibility;
- context awareness;
- usability;
- security and privacy protection;
- and intelligence.

While these requirements provide the groundwork for an encompassing definition (and, to a certain extent, a taxonomy) of what a smart building is, several other perspectives can also be found in the literature, some of which are going to be presented and discussed in the following paragraphs.

For instance, Lobaccaro et al. [2016] provides a systematic review of smart home technologies, grouping them into four categories: Integrated Wireless Technology (IWT); Home Energy Management System (HEMS); Smart Home microcomputers (SHMC) and Home Automation (HA). Toschi et al. [2017] provides a survey about network elements, definitions and standards used in Machine-to-Machine (M2M) communications for different BACS environments, with Domingues et al. [2016] providing an overview about concepts and technologies used in this domain. Also, a survey on ontologies in building automation was performed by Butzin et al. [2017].

Other works are more focused on BACS communications, from the physical medium to protocol-level aspects. For instance, Hallak and Bumiller [2016] provides an overview of powerline communication technologies used in home and industrial automation, also providing some application examples.

Experimental results were obtained by Zhibo et al. [2017] for the validation of IP Wireless protocols used for intelligent grid and smart homes. The study was mostly concerned with latency, Packet Delivery Rate (PDR), coverage and power consumption metrics, having concluded that a PDR between 80-90 percent with a maximum 150 ms deadline can only be achieved with a 3-hop boundary.

A good review of the BACnet protocol is addressed by Hersent et al. [2012]. This protocol is focused on the network layer and above, being used to orchestrate several other technologies (KNX, ZigBee, Webservices, etc) as it specifies internetworking interfaces for each of them. Also regarding KNX, one of the most popular BACS standards for which only IPv4 interoperability is provided, Seifried and Kastner [2017] proposes a possible KNX IPv6 architecture, and also compares the recent KNX IP Secure initiative with IPSec network layer security.

The integration of BACS with the cloud and IoT devices is also addressed by Li [2018], which proposed the development of a smart home cloud server where the communication is established through a Message Queuing Telemetry Transport (MQTT) broker.

More experimental aspects, such as the integration of Software-Defined Networking (SDN) into smart buildings was considered by Usman et al. [2019]. The study considered the adoption of SDN to be generally beneficial having also identified several SDN-related gaps/challenges in terms of network management, maintenance, east-west/southbound interface integration, traffic management, energy and automation.

Other works are more focused on architectural or development aspects. For instance, Fatehah [2018] proposes the use of a software engineering approach for the design of BACS, while Bugeja et al. [2018] has an overview of smart connected homes architectures (centralized or distributed) and with different communication models (device-to-device, device to cloud or device to gateway).

Regarding security aspects, a comprehensive industry study by Brooks et al. [2017] about vulnerabilities, current industry practices and security management best practices was undertaken in 2017, with support of the ASIS Foundation, Security Industry Association and Building Owners and Managers Association. It covered several different aspects, including a survey involving practitioners from 38 different nations and diverse areas (72 percent from security and the remaining from facilities), a survey review undertaken by a focus group of 14 experts, and the draft of BACS security guidelines for the industry. The report also provides an overview of BACS, its fundamental concepts, the BACS market and its industry landscape.

### 2.1.2 KNX Systems

This section provides a basic description of the KNX protocol and related technology ecosystem. More detailed information can be found on the KNX Association site (see KNX Association [2020a]).

The KNX standard appeared in the early 1990s, driven by the European Installation Bus Association (EIBA) (see KNX Association [2020d]) as a way to enable the connection, configuration and communication between multiple building automation devices (e.g. sensors, actuators, buttons and other user interfaces), using a common language and a standard communications protocol. It is widely used for home and building automation, for instance to control lighting, shutters, security systems, energy management, heating, ventilation, air-conditioning systems, signalling and monitoring systems, remote control, and audio/video control. All these functions are managed via the KNX protocol set.

Opposite to traditional electric installations, in KNX installations there are no dedicated hard-wired connections between control devices and actuating devices. For example, a light switch is not directly connected with the controlled lights. Instead, all devices are connected via a shared bus that runs on 29 Volts. All bus devices can be programmed with a common tool – Engineering Tool Software (ETS), enabling easy and flexible deployment. Moreover, subsequent changes require no rewiring.

A KNX system requires the following components (cf. Figure 2.5):

- Power Supply (PS), that feeds the bus and KNX devices.
- Sensors (push buttons, thermostats, air velocity meters, etc.) that generate commands as *telegrams*.
- Actuators (switch relays for lights, blinds, etc.) that receive the *telegrams* and perform predefined actions.
- The bus that connects all sensors and actuators.

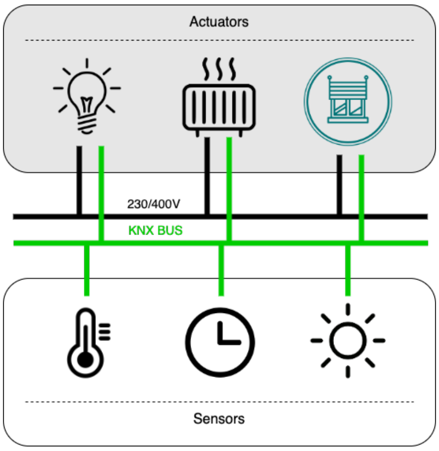


Figure 2.5: Basic KNX elements

KNX is designed to be independent of any particular hardware platform – simple control functionalities are often implemented using basic 8-bit micro controllers, while more complex functionalities may require more powerful hardware platforms. The most common transmission medium in KNX is twisted pair (TP), but KNX also supports other medium, such as Powerline (PL) networking, Radio Frequency (RF), infrared, and Ethernet/IP – even though some of these medium are rarely used in production scenarios.

The smallest entity within the KNX network topology is a line. A line contains a maximum of 64 Devices (Ds), which is enough for most small scale projects. Larger projects may use up to 15 lines, combined within one area, all connected via a main line. Different lines may be connected to the main line with Line Couplers (LCs). Furthermore, it is also possible to connect up to 15 areas to a backbone. Single areas are connected to the backbone line via Backbone Couplers (BCs) (cf. Figure 2.6).

KNX *end devices* may be connected anywhere in this topology. Up to 255 end devices may be addressed in any sub-network. Those devices may be numbered from 1 to 255. Each device (backbone coupler, line coupler, end device) must have an Individual Address (IA), which is unique throughout the complete topology.

To standardize the configuration and commissioning of a KNX installation, the ETS, created by KNX Association [2022], should be used to define, program, configure and commission an entire KNX network. The most recent versions of ETS support the use of device catalogues, available online, to streamline the solution design. The catalogues are supported by the KNX Association with

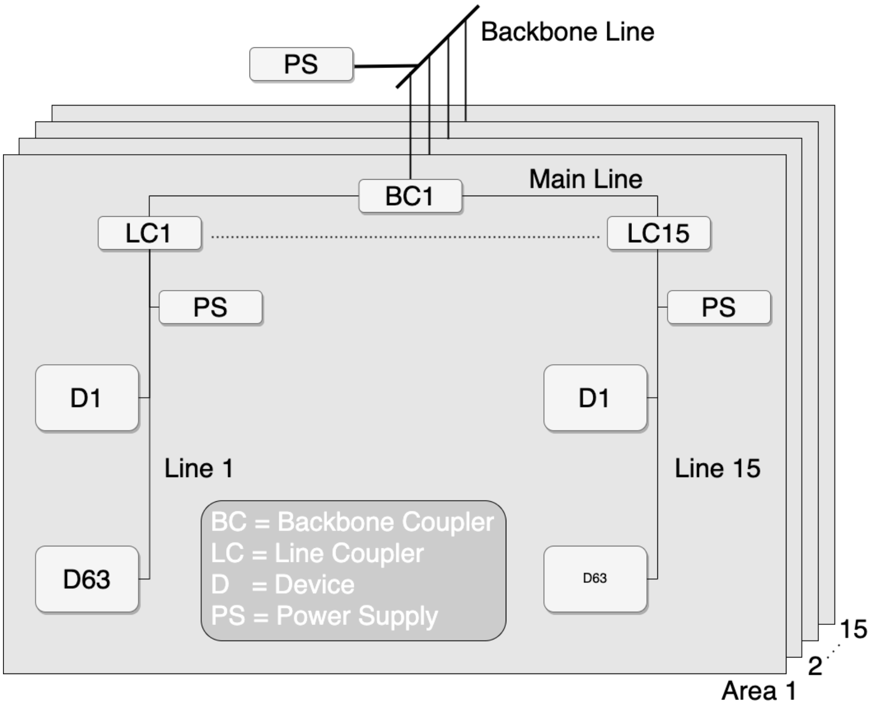


Figure 2.6: KNX logical topology (adapted from KNX Association [2020b])

contributions from their manufacturer members (around 500 at the present date, with around 8,000 products).

In ETS terminology each installation is known as a project. These projects incorporate a local copy of all the installation details – like devices, topology and commissioning state. The application allows exporting the project or some of its parts in a well-documented XML format, specified by the KNX Association (KNX Association [2004]).

The development of KNX-certified devices can follow three different models: partial, OEM or full development. *Partial* development devices are based on available and already certified system components, communication stacks and modules, including the (KNX-certified) Application Program. *OEM devices* are straightforward relabels of already certified KNX devices (typically developed by other KNX members). With this option the development effort is nearly reduced to zero, and only the Application Program(s) need to be registered in the name of the reselling manufacturer. Lastly, *fully developed* devices require several steps: definition of the characteristics, selection of the profile, selection of the communication medium and implementation of its stack and finally the development of the *Application Program.* The *Application Program* will be signed by the KNX association and certified using tests by certified entities.

KNX is a fully distributed network, which accommodates up to 65,536 devices in a 16 bit Individual Address (IA) space (see Figure 2.6). The IA of each device is composed by its area, line and device numbers, in the format:

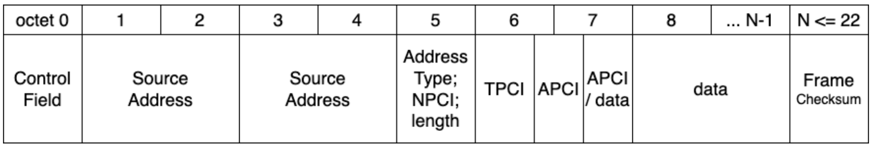
$$area.line.device$$

There are two addressing modes, corresponding to specific communication profiles and purposes:

- **Broadcast and Unicast**, used for network and resource management. As a first step, broadcast (optionally using a device's unique serial number) is used to assign a unique IA to the device. Then, point-to-point communications are used to upload the *applet* binary image of the device *Aplication Program* (firmware) or when directly communicating with a specific device is need (e.g. Device Information request).
- **Multicast** Group Addresss (GAs) are used for full multicast (group) addressing, for runtime efficiency. The various devices have the ability to expose multiple Datapoints, which can be independently grouped as network-shared variables. These shared variables, with read and write capability, use a 16-bit address space that will allow the system a total of 64K GA space.

The Logical Protocol Data Unit (LPDU) of the KNX standard frame, as specified in the KNX standard (KNX Association [2009]), is presented in Figure 2.7. This KNX message format is serially encoded in the frames or telegrams which are sent on the bus. It corresponds to the interface above Layer 2, and its messages include the following fields:

- Control Field – determines the frame priority and distinguishes between Standard and Extended frames (where N $<$ 255).
- Source Address – the IA of the source.
- Destination Address – either an IA for point to point communication or a GA for multicast (group) communication.
- Address type – specifies the type of the destination address (IA or GA).
- Hop Count – is decremented by routers to avoid looping messages; when it becomes zero, the frame is discarded from the network.
- Length – the frame length.
- Transport Layer Protocol Control Information (TPCI) – controls the Transport Layer communication relationships (for instance, build up and maintain a point-to-point connection).
- Application Layer Protocol Control Information (APCI) – can tap into the full toolkit of Application Layer services (Read, Write, Response, ...) which are available for the relevant addressing scheme and communication relationship.
- Data – the message payload. Depending on the addressing scheme and APCI, the standard frame can carry up to 14 octets of data (even more on extended frames). Segmentation for bulk transfer, like the download of an entire application program, is handled by the management client (e.g. ETS tool). The standard frame ensures direct upward compatibility with EIB.
- Checksum.



| octet 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | ... N-1 | N <= 22 |
|---|---|---|---|---|---|---|---|---|---|---|
| Control Field | Source Address | | Source Address | | Address Type; NPCI; length | TPCI<br>APCI | APCI / data | data | | Frame Checksum |

Figure 2.7: KNX LPDU standard frame structure

Considering the nature of the information travelling on the KNX bus, it becomes obvious to which extent its protection is a relevant matter, moreover if one considers the implications in terms of BACS security risks. The potential impact of such risks scales according with the dimension of the application scenario, which can range from a single house to a multi-story smart building supported by a complex automation infrastructure encompassing functionalities such as access/gate control, alarms, lightning or climate control.

If left unprotected, BACS can be exploited for malicious purposes, such as unlocking access to building premises or deactivating alarms, as well as allowing for intruders to eavesdrop unprotected data from presence detectors, energy consumers and administration programs. The manipulation of lighting control systems, heating control systems and other processes in building technology is also a potential risk. In response to these potential risks, KNX Secure (KNX Association [2020c]) has been developed to improve cyber security in building automation, enabling protection at the message data level (*KNX Data Secure*) and at the level of communication over IP (*KNX IP Secure*). However, the inertia derived from the large number of existing legacy systems is a serious obstacle to widespread deployment of KNX Secure. Existing devices would require upgrading and that is not possible by simple upgrades of their *Application programs*, due to insufficient memory and computing resources.

### 2.1.3 IoT and BACS

The usage of IoT for home automation has received considerable attention, both from a commercial point of view and from a research perspective.

A review of system architecture, software, communications, privacy and security of IoT-based smart homes can be found in Mocrii et al. [2018]. Another survey of the adoption of IoT for the development of smart buildings, within academic and industry contexts, is provided in Jia et al. [2019]. The authors argue that a mature adoption of IoT technologies in building industry is not yet realized and still requires intensive research.

Some authors have proposed specialized Intrusion Detection System (IDS) for IoT. A good summary on this subject, that includes mobile ad hoc networks, wireless sensor networks, cloud computing and cyber-physical systems, can be found in Santos [2018]. It covers works from 2009 to 2017, concluding that IDSs for IoT are still in their infancy, cover just a few of the existing technologies, and are not able to detect a large range of attacks.

Darabseh and Freris [2019] proposed a cyber-physical architecture for IoT applications, which leverages software-defined principles intending to decentralize decision-making within IoT networks. This architecture entails three main domains: the physical space, the cyberspace and the structured control space, all described as software-defined systems. Alternative examples of low-cost DIY solutions used for home automation systems are provided by Asadullah and Raza [2016], as well as by Vikram et al. [2017], which proposed a low-cost home automation system based on Wi-Fi wireless sensor networks.

A discussion of security in existing IoT communication protocols (e.g. Bluetooth, BLE, ZigBee, NFC, Wi-Fi, Thread, LoraWAN) is presented by Ray [2017], supported by a previous survey from Granjal et al. [2015]. Within the same scope, Dutta and Wang [2018] proposed an IoT-based security system for smart buildings using Radio-Frequency IDentification (RFID) and International Mobile Equipment Identity (IMEI) numbers for two-step authentication. An investigation of security requirements and solutions for an IoT-based smart home architecture is provided by Ali et al. [2017]. Finally, Fischer et al. [2017a] proposed a security demonstrator for experimental evaluation, testing it with two attack scenarios using the Z-Wave protocol.

As for mediated or middleware-based IoT developments, the smartFW framework from Ilieva et al. [2016] was proposed for integrating short-range devices in smart home buildings. It acts as a mediator between IoT integration platforms, allowing end-users to control their smart homes. Also, the use of Blockchain technology is proposed by Abunaser and Abunaser and Alkhatib [2019] to solve the centralized cloud drawback of IoT in smart homes. Blockchain may eventually help securing data and transactions, but more research is needed until such promises are materialized.

Figure 2.8 represents the typical architecture of current implementations of IoT for building automation. It shows the segregation that exists between the components locally deployed. In this particular use case scenario, integration between sensors, appliances and actuators takes place in the cloud service. Quite often, system integration services from different providers do not communicate with each other, requiring another layer for interconnecting different systems from different providers. This clearly differs from classic BACS, which are locally deployed with full operation support and were designed to work in closed environments, though frequently supporting interconnection to the ICT layer and to the Web – a natural evolution introduced mostly for maintenance and support purposes.

Lilis et al. [2017] provides a good discussion of the opportunities and side-effects of fully IoT enabled and controllable intelligent buildings, when compared with the well-established classic BACS. One of the main points against IoT is that it is not possible to expect continued product development and support, indefinitely, from a single manufacturer. The only possible way to reassure the market is the existence of compatible products from multiple manufacturers. This is a key point in favour of BACS, with their standards. BACnet claims more than 800 vendors, LonWorks claims a range of more than 4,000 products, and KNX claims more that 8,000 compliant devices from more than 590 members (most of them manufacturers).

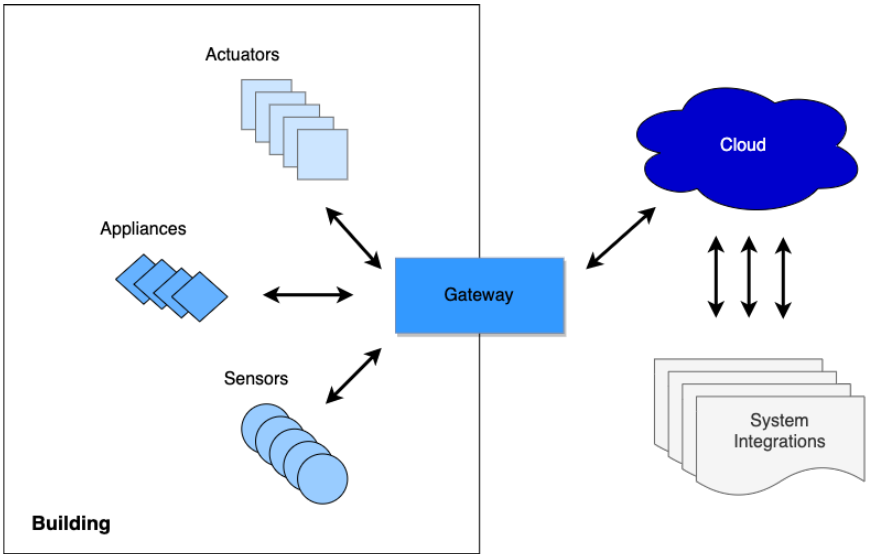


Figure 2.8: IoT Cloud-based architecture for smart home (adapted from Mocrii et al. [2018])

Qiu et al. [2018] introduced the concept of Heterogeneous Internet of Things (HetIoT), supported by the intrinsically heterogeneous architecture which is characteristic of IoT solutions. The authors propose a four-layer HetIoT architecture consisting of sensing, networking, cloud computing and applications. They also present and discuss a SoA in HetIoT research and applications.

Vanus [2018] focuses on the functional interconnection of a KNX-based BACS system and IBM Watson cloud services, in order to enrich the system with a natural language interface.

A number of deployment-limiting issues currently impact the scope of IoT utilization, including: the lack of comprehensive end-to-end standards, fragmented cybersecurity solutions, and a relative dearth of fully-developed vertical applications – as stated by Minoli et al. [2017], which review some of the technical challenges and opportunities related with the adoption of IoT for building automation. The authors concluded that, from a technological perspective, the development of appropriate reference architectures and supporting standards is fundamental, fostering interoperability and equipment cost-effectiveness. It is also critical to develop and deploy strong system-wide IoT security capabilities, as it expected that the ongoing network softwarization trend, as well as the introduction of 5G communications will improve the support for IoT traffic. From this perspective, it is expected that the development of cloud-based analytics will become an enabler for efficient optimization, data mining, trending and forecasting capabilities.

The above arguments lead to the conclusion that the easier deployment and the lower cost of IoT devices will turn them into an extension of existing BACS systems. Their integration with the Cloud is one of their greatest assets, though at the cost of additional security concerns and challenges as Bajer [2018].

## 2.2 Literature Review

### 2.2.1 Methodology

The main objective of this literature review is to gather and organize information about research and industry developments in the field of BACS security, in order to characterize the current SoA. A wide systematic search was conducted as source of information, based on five databases: IEEE Xplore, Science Direct, Springer, ACM and Wiley, complemented with other sources such as search engines and specialized conferences.

The query pattern used for search was: *(((smart AND building) OR (Building AND Automation) OR (home AND automation) OR (Domotics) OR (building AND management)) AND (Safety OR Security OR Attack OR Threat) AND NOT(energy)).* This pattern was adapted to the different database engines in order to get the best results. For some databases, additional filters such as *computer science* or *communication networks* were also used, to refine the search. The adopted inclusion criteria were:

- Publication in the last five years.
- Studies published in English.
- Inclusion of the relevant papers referred by included studies.

In order to complete this search, we've added an extra query to retrieve privacy-related studies in BACS.

Table 2.1: Documents processed in this study

| Database | Total retrieved | After applying inclusion criteria | After title selection | Used after abstract selection |
|---|---|---|---|---|
| IEEE | 4896 | 1966 | 53 | 33 |
| Science Direct | 10604 | 2089 | 24 | 9 |
| Springer | 36665 | 1450 | 23 | 9 |
| ACM | 785 | 340 | 12 | 7 |
| Wiley | 2821 | 1335 | 12 | 8 |
| Other Sources | | | 138 | 50 |
| Total | | | | 116 |

As exclusion criteria, we chose to eliminate all documents whose full text was not available and those that dealt mainly with energy issues, as our focus is domotics and building or residential automation, in a broader sense.

The selection of records was then made through the analysis of documents whose titles and/or abstracts were retrieved through the search strategy and that met the inclusion criteria's mentioned above (see Table 2.1).

## 2.2.2 Security and Privacy Concerns in BACS Scenarios

In this Section we discuss the impact of security in typical BACS scenarios. First, we briefly overview the relevance of security (and safety) in such scenarios, identifying general risks associated with intentional or accidental failures of the controlled home automation processes and/or with loss of privacy. Next, we discuss previous works that analyze potential safety and security risks directly or indirectly related with BACS. Afterwards, we approach some studies related with privacy in BACS. Finally, we present a set of known attacks to BACS, both in laboratory testbeds and in real work systems.

The BACS facilities control and are controlled by devices which are often physically accessible to the users of the buildings. This way, malicious users can easily hamper sensors and controllers. More over, since many of those devices allow bidirectional access to the automation and management platform, they may provide an access path to BACS platforms. In parallel, BACS platforms may also be reached via the IT systems they are interconnected with, providing a remote attack path.

The unauthorised access to the data that circulates in the BACS systems opens the possibility of inferring knowledge about the usage and occupation of spaces, in a clear violation of the privacy of their users. The manipulation of these control networks makes it possible to block or confine users to certain spaces, or to change environmental conditions (e.g. by manipulating the HVAC, ventilation and lighting systems).

Intrusion into BACS systems creates a privacy issue. Building occupants' data and their habits can potentially be exposed. This potential exposure may lead to various forms of misuse.

The failure or malfunction of certain BACS equipment is also a safety problem, since it may cause improper functioning of the rest of the system. In this sense, monitoring and anomaly detection should also be a concern when analysing BACS safety and security. Moreover, malicious access and manipulation of BACS platforms may lead to the excessive deterioration or even failure of building equipment, through forced operation outside the normal thresholds. Ultimately, this situation may even put the whole building at risk (e.g. fire, intrusion).

### 2.2.2.1 Security Risks

In this subsection, we discuss some of the most relevant previous works focused mostly on identifying and analysing security risks somehow related with BACS.

BACS security issues were already a concern in 2010, especially in the anticipation that insecure protocols would soon be opened to ICT networks. A few approaches to BACS safety and security have been proposed by Granzer et al. [2010] and Novak and Gerstinger [2010], but had no impact in the real world. As of 2015, a study from the Gartner Group prediceted that, by 2018, 20% of smart buildings would suffer from digital vandalism in some way (Levy [2015]).

Similarly to IACS, BACS security breaches are often considered to be a consequence of using systems, protocols and standards that were originally conceived to operate in isolated environments, without any connection to ICT networks or the Internet. This is aggravated by the fact that many legacy devices cannot be patched, often meaning that only isolation or complete replacement might ensure adequate security (see Wendzel et al. [2017]). In general, most attack categories that are characteristic of IACS proposed by Macaulay and Singer [2011] may be somehow transposed to BACS scenarios. However, even though some the protection strategies used in IACS might somehow provide hints on how to keep BACS secure, there are considerable context differences that, eventually, require specific approaches to the problem of BACS security.

An overview of the most used BACS protocols, security issues and recent security research trends is presented by Wendzel et al. [2017]. Authors summarize and compare some of the most used BACS communication protocols (e.g. KNX/EIB, BACnet, ZigBee and EnOcean GmbH [2020]) and identify attacks as belonging to two different levels: network level (management and automation levels of BACS architecture) and device level (field level of BACS architecture). At network level, attacks are split into four different categories: traffic interception (network sniffing); malicious packet creation; network packet change (man-in-the-middle attacks); and outage or reduction of network service quality (Denial of Service (DoS)). On device level, the attacks are grouped into three patterns: physical tamper; side-channel analysis (e.g. usage of monitoring to obtain cryptographic keys); and software attacks (such as code injection).

A review specifically focused on the intersection of smart grid and smart homes (in the sense that information is exchanged between them to optimize energy management) is provided by Komninos et al. [2014]. Several scenarios are presented, accompanied by potential security countermeasures, based on a review of contemporary literature.

Lei et al. [2018] address the vulnerabilities of home digital voice assistants, which often rely on single factor authentication – a voice password like just some words (eg. ”Alexa”, ”Hi, Google”). Authors provide a set of proof-of-concept attacks that send fake commands to the voice assistant, using both hacked Bluetooth speakers and smart TVs. Then, they implement and test the introduction of a second authenticated factor (only allowing commands if any person is detected nearby), using WiFi technology to detect indoor human motions.

Liu et al. [2018] propose a taxonomy for security assessment of IP-based BACS (see Figure 2.9) and apply it to Thread (an IP-based protocol for IoT in building automation by Thread Group [2019]).

Heartfield et al. [2018] propose a different taxonomy approach defining a causal relationship (see Figure 2.10) between three different root criteria’s (attack vector, impact on domestic life and impact on systems) of the home cyber-threat taxonomy. Then a classification is provided for each of those root criteria (the diagrams are omitted from the provided figure for lack of space), considering

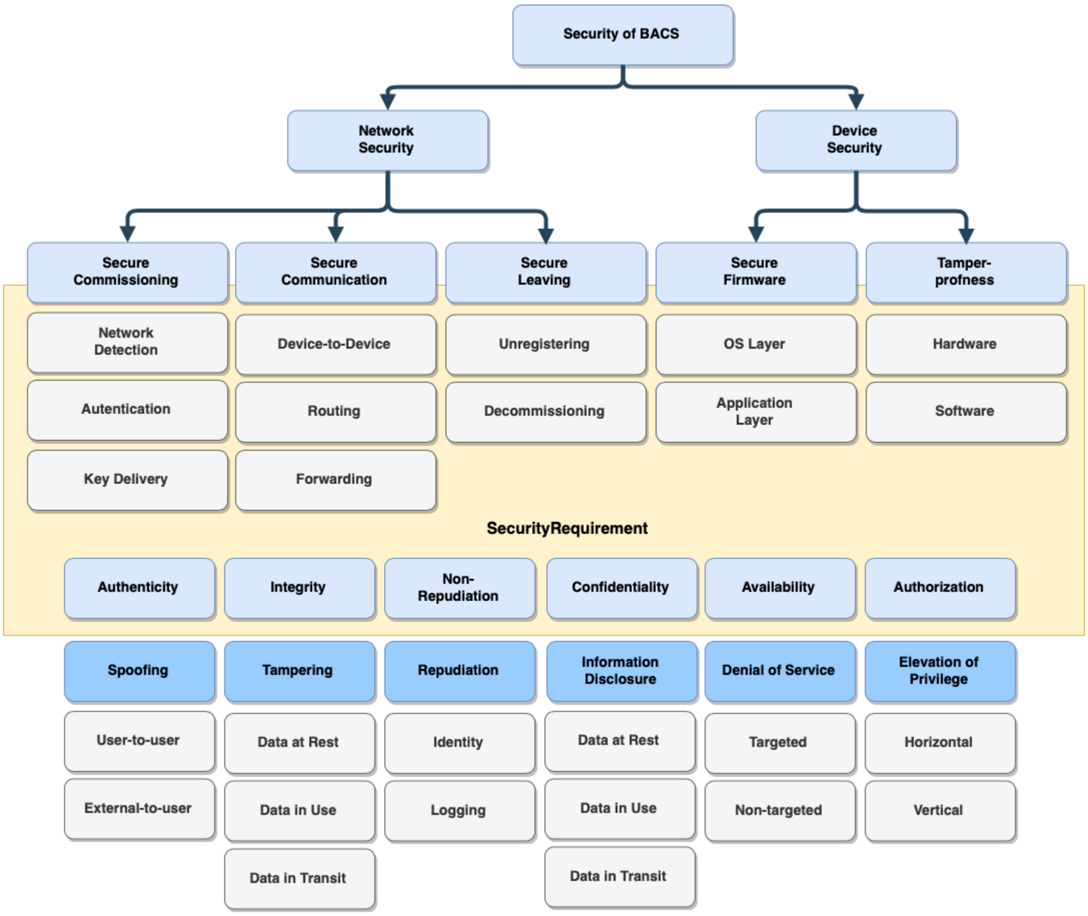


Figure 2.9: Security analysis taxonomy for BACS (adapted from Liu et al. [2018])

the attack vector as well as the impact on systems and, consequently, on the occupants of a smart home.

A very simple taxonomy for classifying security threats is also proposed by Anwar et al. [2017], with three main groups of threats: unintentional, intentional/-abuse and malfunctions.

Graveto et al. [2019] propose a taxonomy that, despite being originally developed for the IACS domain, can also be used to classify network attacks in BACS, as shown in Table 2.2.

The BACnet protocol and its vulnerabilities are presented by Valli et al. [2017]. DoS, halt or buffer overflow of legacy network interfaces by the relative brute force represented by a 10/100 Mbit/s or a 1 GBit/s connection are reviewed. BACnet specifies AES 128 bits encryption and end-to-end authentication, but only the more recent devices with security-based objects and properties apply these specifications. They are optional in the standard due to the need of supporting legacy devices. The protocol has minimal session protections and, therefore, it is vulnerable to replay attacks and spoofing. Finally, the payloads are binary or even clear text, allowing trivial decoding and subsequent tampering.

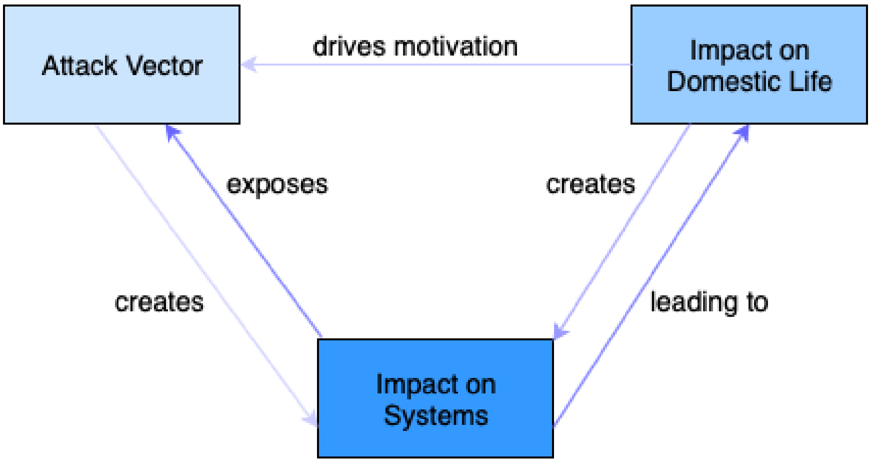


Figure 2.10: Causal relationship betweeen root criteria in smart home cyber-threat taxonomy (adapted from Heartfield et al. [2018])

A description, simulation and testing of proof-of-concept protocol attacks on a BACnet system are provided by Matthew Peacock [2018], which also presents a classification of known attacks according to the STRIDE matrix (**S**oopfing, **T**ampering, **R**epudiation, **I**nformation disclosure, **D**enial of Service, **E**levation of privilege) developed by Garg and Kohnfelder [1999].

Gai et al. [2018] tested home appliances (e.g. SmartTVs, smart home theatre, smart kettle, smart refrigerator, smart thermostat, smart lights or smart security cameras) and categorised vulnerabilities and attack surfaces.

An analysis of two use cases based on the field level on LON and KNX, using BACnet at the automation level, is provided by Mundt and Wickboldt [2016].

Coppolino et al. [2015] overview the risks resulting from the introduction of Internet-enabled devices (e.g. smart home gateways) on BACS for supporting remote access and control. In the same line, Meyer et al. [2017] identify three new attack vectors in BACS related with Internet connections: acquisition of local network access through a provider-supplied device; access to other existing user devices; and data access at remote storage providers.

A set of network scan results for open, real world BACnet and KNX BACS installations was published in Praus and Kastner [2014]. A summary of network attacks that may threaten BACS has been provided by Saxena et al. [2017].

A survey of software security requirements and software protection methods for distributed control applications is provided by Praus et al. [2016].

Looking specifically at BACS platforms that communicate over powerline, such as digitalSTROM AG [2019], Brauchli and Li [2015] provide an analysis of potential risks and mitigation strategies.

An overview of the Building Energy Management Open Source [2019] Software (BEMOOS), developed for energy load-balancing, is provided by Rathinavel et al. [2017]. Security threats and their countermeasures in this context are also

Table 2.2: Simplified Taxonomy of BACS attacks

| Level | Class | Impact | Attack examples |
|---|---|---|---|
| Layer 2/3 | Scanning/ Scouting | Getting information about network topology and devices | On KNX/IP and BACnet/IP, ARP or LLDP queries can be used to track devices; Probe for available services and protocols using a FIN or SYN scan. Simple sniff of KNX/TP messages (2nd and 3rd bytes represent the sender Individual Address) |
| | Attack on data integrity | Unstable and/or unpredictable behaviour | Corrupt inflight data through packet manipulation |
| | Denial-of-Service and/or service degradation | Loss of visibility and/or control | Overwhelm or crash device, via SYN or ICMP flooding; Employment of CAM table overflow to disrupt communications |
| Protocol/ service level | Scanning/ Scouting | Getting information about service and device capabilities | Brute force use of KNX *T_Connect_PDU* to discover existing devices, subsequent scan attacks for device profiling; Use of MITM to analyse used services and protocols |
| | Integrity | Unstable and/or unpredictable behaviour | Abuse of protocol specifications and features, such as the BACnet *ReadProperty* and *Whoami* or KNX *A_Memory_Write_PDU* attacks |
| | Denial-of-Service and/or service degradation | Loss of visibility and/or control | Exploit vulnerability to crash or disable service or device (such as a FTP buffer overflow); Introduce latency or communications failures through MITM attack; Use of managment commands to influence device operation |
| Process level/ semantic | Scanning/ Scouting | Reveal details about the nature of the process | MITM attack for scouting purposes or preparation of replay attack; Use of KNX instructions to download parameters and/or group address tables; Structural analysis of memory map thorugh probing using KNX *A_Memory_Read_PDU* |
| | Direct manipulation | Manipulation of process variables | Manipulation of process variables to alter behaviour, through direct device access (KNX *A_GroupValue_Write_PDU* or *A_GroupValue_Read_PDU*) |
| | Interception and fuzzing | Interception and manipulation of process values | Manipulation of process variables to alter behaviour, through command injection or protocol fuzzing, using a MITM (via ARP poisoning or CAM table) attack to intercept communications and conceal the intruder; Process-aware response injection or replay attacks |
| | Reprogramming | Process behaviour is modified and/or hijacked | Use of of KNX instructions to upload firmware, parameters and/or group address tables |

analysed.

Jia et al. [2017] discuss the vulnerabilities in a reference smart home architectures, proposing a semi-automatic vulnerability detection system for detecting vulnerabilities prior to factory shipment of BACS devices.

#### 2.2.2.2 Privacy

One of the first associations between privacy and buildings probably took place in 1964, when a couple rented an apartment and the owner placed an audio recording device in the bedroom (Hamberger, C.; Eastman, C. [1964]). This situation and subsequent legal actions led to the a legislation change focused on intrusion of privacy of personal quarters. The timeline of the privacy problem in residential buildings (and other scenarios) is discussed by George et al. [2020], addressing the system dynamics of data collection by building automation devices and IoT, as well as their technical and social integration, challenges and significance. As most users are not aware of the information that is collected and the risk to their privacy, this paper suggests a solution with two steps. First, the implementation of a packet tracer that displays the collected data, increasing people awareness and encouraging them to better preserve their privacy. This awareness will lead to a second phase in which new legislation could emerge, requiring manufacturers to implement algorithms that guarantee that devices and services are compliant with privacy regulations.

We only found a couple of papers addressing privacy in the scope of BACS, which forced us to further extend the search towards papers on privacy for so-called ”smart buildings” (mostly linked with IoT and smart metering privacy concerns) that, somehow, are also relevant in BACS scenarios.

Kraemer and Flechais [2018] enumerate five steps to address the challenges of privacy research in smart homes, that could be also applied to BACS: data collection and processing; in-depth analysis of the context; longitudinal panel studies to gather empirical data and privacy behaviours; addressing the perspective of policy makers; and, finally, addressing the criticism that existing frameworks for product design are too vague. However, this vague and generalist approach is also demonstrative that almost everything remains to be done regarding research in the field of privacy in BACS.

Next, we group the surveyed works into five groups: studies based on users' feedback and perception of privacy; case studies on privacy in buildings with BACS; the usage of mathematical algorithms at the service of privacy; some IoT implementations that, by analogy, could be adopted in BACS; and, finally, the issue of smart energy meters and some solutions to enhance privacy in this context.

### User's Feedback and Perception

Zeng et al. [2017] conducted a set of semi-structured interviews with fifteen people residing in smart homes (twelve of them being administrators of these systems) to understand how they use their smart homes, their actions related with security and privacy, and their expectations. The authors found out, as expected, that users are little concerned with their privacy. The natural tendency of users is to trust device and service providers, even claiming that they have nothing to hide, or simply thinking that the existence of a password is enough to guarantee their privacy. When asked about mitigation methodologies, the answers were limited to the usage of independent Wi-Fi networks and the usage of secure passwords as problem mitigation techniques. Finally, they also verified that the existence of users with different levels of access may even lead to privacy issues between the various users of the same home.

A set of interviews to 97 UK-based users of smart assistant devices (Alexa or Google assistant), to gauge their perception of these smart assistant devices when compared to other more familiar devices such as smartphones and computers, is presented by Lin and Parkin [2020]. About half those users were unsure of how to address the privacy issues and settings, and 20 of them, when using shared devices, used sensitive information that should be kept private from other users. The reported transfer of privacy-related behaviours between previous used computing devices and newer smart home devices was low in the adoption of available privacy controls.

Kaaz et al. [2017] conducted a study on the installation and perception of privacy of users of IoT devices, having concluded that understanding how these devices operate is not trivial, making it difficult to perceive threats and the risks associated with their use.

Pathmabandu et al. [2020] propose an informed consent model to address the balance between privacy and convenience. This model is implemented using five steps: apply textual patterns to privacy policies; list privacy permissions; identify privacy infringements; track and log events; and recommend preventive actions that allow the user to control and mitigate emergent privacy issues that have occurred and/or my happen in the future. The proposed model enhances the user awareness, helps in the detection of privacy compliance and infringement by devices, and improves the user's privacy-protecting behaviours in small steps.

### Use Cases on BACS

Across Europe, seniors want to live their old age in their homes, instead of retirement homes. Instead of providing care on scheduled appointments, there is the possibility of providing event-based services, improving costs and effectiveness. A case study is presented by Franke et al. [2016], analysing a house that uses the KNX standard as the basis of its BACS infrastructure. However, to guarantee the privacy of the occupants, all the information is processed on-premises, and only part of it is transmitted to remote care providers. Residents and their families can define the information to pass on to external entities (privacy by design), such as "the resident did not use the bathroom within the last 24 hours" or, for example, "the resident is not moving for more than 2 hours". These events allow the action without violating the privacy of residents.

The case study presented by Mundt et al. [2012], opposed to the previous example, demonstrates the possibility of violating the privacy of users of an office building to find out "who refuses to wash hands". The office building holds a BACS infrastructure, based on KNX, with motion sensors every 8 meters, lighting control in all offices, laboratories and other divisions, and blind control in all convenient locations. The authors demonstrate that the sampling of KNX traffic, based on the collection of previous tests (asking some users to make their way from their office to the bathroom, with and without hand washing) allowed a posteriori, in an extended data collection, to infer the desired information. Accessing the information was easy by simply removing any switch with access to the KNX twisted-pair bus and then connecting there the collection system.

### Privacy-focused Analysis of BACS Data

Xu and Agung Julius [2019] present the construction of a map of observations in the form of metric temporal logic formulas, which can be formally proved to allow the detection of faults in a switched system, while preserving certain privacy conditions. Two scenarios are considered: in the first, all room occupancy possibilities are private (unoccupied, one occupant or two occupants) and, in the second, only the room occupation by one person is private, considering it public when there are two or even no occupants. The entire mathematical formulation is presented and the inclusion of systems with both external and internal events, or even hybrid systems, are indicated as possible future works.

The usage of Gaussian noise in the corruption of measurements in a BACS system if presented by Alisic et al. [2020], as a way to mitigate unauthorized access to sensors data. This corruption of information aims at concealing the state of occupation in the apartment.

### Issues in IoT implementations

A system that uses infrared retro-reflection is presented by Santo et al. [2017], as an indoor positioning system that preserves the users' privacy. The device does not capture any details of the persons' appearance, despite using infrared images (if due care is guaranteed, such as placing the device avoiding to capture occupants near windows and avoiding their capture less than one meter from the places where residents spend most of their time).

Gao et al. [2020] use a Home Brain with a processing model, computing model and database to preserve the voice authentication for each IoT device, enabling privacy-preserving speaker verification. In an initial registration phase the features of the valid user voices/IoT pair are extracted and preserved in the database for future use.

As with BACS, most IoT devices have limited processing capabilities and patching to add security features is not allways possible. Thus, Iqbal et al. [2021] proposes to use SDN in smart homes, by means of installing an Openflow switch, between the domestic gateway and the automation devices, as well as an SDN controller. This way, all requests from home users and even remote ones could be validated and even subject to authentication. The protocols necessary for authentication and privacy preservation are presented and discussed, as well as an evaluation and comparative analysis. The authors conclude that the protocol can be implemented in any smart system as it is based on lightweight nature of symmetric cryptography.

A framework based on spatio-temporal mining for efficient recognition of human activities in smart homes, accompanied by a technique to enhance privacy using micro-aggregation, is proposed by Samarah et al. [2017].

### Privacy and Energy

The intelligent control and measurement of energy consumption in buildings is a fundamental part of the smart grids vision. However, continuous submetering or sampling at tight intervals poses serious privacy risks to the users. The survey by Finster and Baumgart [2015] focuses precisely on these issues, starting by dividing the problem into two approaches: metering for billing and metering for operations. In the first situation, the continuous measurement is not important, but rather the accumulated consumption, sampling at longer intervals (in the limit extended up to the billing period) will improve privacy. In this case, the invoicing value being important, the problem can be reduced to a problem of trust addressed by: delegating the calculations to a third-party trusted by both (consumer and supplier); using a trusted platform; or having the smart meter itself calculating the amounts due. However, in the second situation, regarding smart grid management, instantaneous measurements or at least at short time intervals are necessary, and four possible approaches for preserving privacy are analyzed: anonymization or pseudo-anonymization without aggregation; aggregation using trusted third party; aggregation without recourse to a trusted third party; and, finally, the submission of inaccurate information. In this last approach, the submission of imprecise information implies some coordination between the smart meters, so that the global accuracy is not too affected. The alternative to privacy issues will be to avoid generating information that creates privacy risks. For this purpose, two concepts are used: to use batteries; and to determine the sampling rates of smart meters as a design parameter.

Pham and Mansson [2019] discuss in detail the use of energy storage systems as a technique for mitigating privacy problems. Different types of storage technologies are analyzed, and the minimum storage/cost capacities are determined in cases of one or multiple users of the housing.

Sarbhai et al. [2019] also use batteries to obscure the data collected by smart meters, presenting three distinct algorithms as a solution for peak load reduction: random charging; random charging with linear response; and random charging with quick response (to avoid the risk of peak loads leading to outages, in case a large number of homes start charging their batteries at the same time).

Wu et al. [2016] provide a mathematical formulation of optimization for online privacy-aware cost-effective appliance scheduling. It should however be noted that the time needed for the calculations will grow according to the number of appliances.

Dasari et al. [2021] apply federated learning for energy load prediction approaches that enhance users' privacy. Each building uses local data to train its local model and compute gradients, then the masked gradients are sent to a trusted third-party server, which in turn performs the aggregation (without capturing information from any participant), and the aggregated model is sent to the model owner (e.g. energy supplier or grid manager). The final model is finally sent back to building users, allowing them to update their local models.

#### 2.2.2.3 Possible attacks

The scientific community has analysed and showcased several attacks in controlled or laboratory environments, exploiting known BACS vulnerabilities and security issues. In this subsection we identify some of the most relevant works in this specific line of research, which we complement in the next subsection with an overview of the more well-known attacks to real systems.

Ling et al. [2017] demonstrate four attacks to a popular smartplug model (the EDIMAX SP-2101W): device scanning; brute force attack; spoofing and a firmware attack.

The vulnerability of BACnet to amplification attacks has been assessed by Gasser et al. [2017]. These DoS attacks where the response payload is larger than the request payload (by the Bandwidth Amplification Factor (BAF)). An identification of the BACnet properties that provide responses larger than the requests (i.e., BAF>1) is presented, leading to the conclusion that around 90% of the BACnet requests lead to responses at least 5 times larger (i.e. BAF>5), in some cases up to 19.8 larger responses.

Potential attacks in wireless communications potentially used in BACS (near field communication (RFID), ZigBee and WiFi) are identified by Krishnan et al. [2017]. Potential threats to these systems include eavesdropping, physical attacks, DoS, spoofing, replay attacks, data manipulation or injection, man-in-the-middle and packet rerouting.

#### 2.2.2.4 Publicly known attacks in real systems

In this subsection we overview five known attacks to real BACS systems: the attack to the St. Regis ShenZhen Hotel: the Mirai Malware; the attack to the Google Australia Office; the attack to the Target Corporation; and the attack to the Fragrance Hotel Singapore.

The **St. Regis ShenZhen Hotel**, that occupies the top 28 floors of a 100 story skyscraper, allows guests to use an iPad to control all the facilities of their room: music, blinds, lights, TV, temperature, do-not-disturb lights, etc. The hotel BACS system had several flaws that allowed Jesus Molina to create a remote control that allowed access to all the hotel rooms. The attacker stated that he could even be located in another country (Molina [2015]).

The BACS system existing at this hotel uses devices with the KNX standard, and the KNX twisted-pair network was interconnected to the WiFi local network in order to communicate with the iPad app, using a KNX/IP router. By using a network sniffing such as Wireshark, and just pressing every button on the iPad, the researcher was able to create a dictionary of actions. The packed decoding provided the KNX Group Address of each action, and also disclosed each device's IA.

First, the *eibd* open source tool (BCU SDK [2006]) was used to perform the handshake with the target IP and to keep the connection alive. Then, by using a simple write, the hacker could send any KNX command to the KNX network (e.g. *groupswrite local:/tmp/eib 2/0/3 80* will switch on the lights).

The performed network sniffing also showed the existence of ”ghost” addresses, not used by the iPads – pointing to several other devices available at the KNX network, besides those from guests rooms.

The only possible solution to solve this vulnerability while maintaining the existing architecture, according to Molina [2015], would be to implement a secure tunnel between the iPad and a network device preceding the KNX/IP router. The tunnel should provide mutual authentication (such as Secure Sockets Layer (SSL)) to avoid the certificate steal from the iPad. Before each guest checks-in, the certificate should be reinstalled and the integrity of the app should also be verified.

The **Mirai Malware** is a very relevant example of an attack to real world systems. Even though it did not specifically target BACS platforms, the generic profile of the target devices is very similar to the profile of typical BACS devices.

In 2016 Dyn, a high-profile provider of Domain Name System (DNS) services, was the victim of a Distributed Denial of Service (DDoS) attack that was clocked at 1.2 TBps by Hallman et al. [2017]. Less than a month before, the KrebsOnSecurity cyber security blog was also targeted with a similar attack, with about half the power (around 620 GBps). A detailed analysis of all the preparation and evolving steps of this attack, based on the now well-known Mirai botnet, is provided by Peterson [2019].

A bot network is composed of a Botmaster that controls the all system, a set of command and control servers, and finally an army of infected and conscripted bots. A botnet can be used either to perform a distributed task like distributed computation (e.g. mining) or to empower an action and concentrate efforts against a specific target (e.g. DDoS).

The Mirai botnet was conducted through Internet-connected unsecure IoT devices (e.g. CCTV cameras, home routers). As stated by Elliot Peterson (Wright [2019]) the evolution of the Mirai army was the result of a ”war” between competitors like Lizard Squad and others, that started back in August 2016. Both groups launched a botnet in an effort to gain advantage in the booter black market.

The first high-profile Mirai attack targeted the Krebs website (taking it down for several days and forcing Akamai Technologies to drop the site from its DDoS protection service). Following this attack, several other Mirai-based attacks took place against other targets, such as DYN – a large DNS service provider.

The building management system of **Google offices** located at Wharf 7, Sydney, was hacked by two security researchers in 2013. This system was built using the Tridium Niagara AX platform and Tridium SoftJACE controllers (basically Windows systems with a Java virtual machine and the Tridium client running on it).

After hacking the system, the security researchers opted for reporting the issue to Google (Zetter [2013]). Nevertheless, malicious hackers could have used the same vulnerabilities to gain full control of the building management system.

The accessed data included a control panel showing blueprints of the floor and roof plans, as well as a clear view of water pipes snaked throughout the building and notations indicating the temperature of water in the pipes and the location of a kitchen leak. Moreover, due to unpatched vulnerabilities, researchers were able to remotely access and get the config.bog file (which holds the system configuration data, usernames and passwords) by means of privilege escalation, also allowing to overwrite other files.

Tridium has meanwhile released a patch for the vulnerability that was exploited on this attack. The involved security researchers stated that a good fraction of the 25,000 other Tridium systems they have found connected to the Internet are still unpatched and just as vulnerable as the Google's system they hacked. Such systems were in use, for instance, at a British Army training facility, at Boeing's manufacturing facilities in Renton, at the Changi airport in Singapore and at the Four Points Sheraton Hotel in Sydney.

**The Target Corporation**, a large retailer in United States, saw its network hacked and broke into in November 2013, by means of credentials stolen from a vendor of refrigeration, heating and air conditioning equipment (Fazio Mechanical Services), a subcontractor that worked at several Target locations (Krebs [2014]).

An unidentified source stated that in order to monitor heating and energy management systems, access to outside suppliers to control systems and production costs was guaranteed. This created a gateway to the internal networks to which these systems were connected. First, the attackers uploaded their card-stealing malicious software to a small number of cash registers within Target stores, for testing all the functions. Then, before Black Friday, the intruders pushed their malware to a majority of Target point-of-sales. Finally, the stolen credit card data from Target's customers was uploaded to compromised computers in the United States and Brazil, accessed from the Eastern Europe and Russia.

This incident shows that outsourced BACS services may lead to the creation of external backdoors to the systems, either due to lack of security updates or improper use of access credentials. Similarly, the simple installation of IoT devices (such as basic DIY solutions) may support malicious actions without the owners' knowledge. Both legacy BACS systems and IoT devices are prone to exploitation by hackers outside their normal scope or purpose.

### 2.2.3 Proposals for Improving BACS Security

This section summarizes the most relevant proposals for improving security in BACS systems found in the literature. According to their scope, they are organized into five different groups: security monitoring; anomaly detection; Intrusion Detection System; and contributions to the improvement of BACS.

#### 2.2.3.1 Security Monitoring

The works discussed in this subsection focus on improving the monitoring of BACS systems, namely with the addition of specialized devices (able to read and process the messages exchanged between the different BACS nodes) and/or with specialized analysis techniques able to detect potential attacks.

Jones et al. [2018] propose an automated device-level solution to monitor BACnet networks. Deployed in a Single Board Computer (SBC), this device intercepts communications between BACS devices at field-level. It supports deep packet inspection and is able to produce a few simple active responses, by using unsupervised artificial neural networks. When an attack is detected, malicious traffic is blocked until the affected node is brought back to its normal working state. The open source time series database *influxDB* is used, with a retention time period of one hour. Data collection is performed using Phyton scripts (*pcapy* library in network sensors and VOLTTRON for physical censoring system – Katipamula et al. [2016]). Artificial Neural Networks (ANN) based on the unsupervised Adaptative Resonance Theory are used for the recognition of normal and abnormal behaviour.

Abdulmunem et al. [2016] analyse a scenario of cyber-attacks on a BACS test-bed, as a case study of how they might affect the system performance, using Intervention Mode Effects and Criticality Analysis (IMECA) and Failure Mode Effects and Analysis (FMEA). Markov models are used to calculate BACS availability considering the possibility of recovery and different kinds of faults.

Chowdhury proposed a framework named Expat Chowdhury [2019], which aims at protecting smart-home platforms from malicious automation apps. For this purpose, a platform-agnostic formal specification language is used to encode the users' expectation of the building automation behaviour, thus defining a set of policies which are later used to verify actions and validate app behaviour. This proposal was tested on OpenHAB, a representative platform used in home automation, as stated by the authors.

A multi-agent system named JMonA was proposed in Vasyutynskyy et al. [2006]. It spreads agents across the various nodes of the BACS system, for enlarged monitoring. This framework was first tested in a LonWorks laboratory setup, later using a network simulator and several control systems as a mockup of larger BACS. Moreover, the authors also identified a set of fundamental requirements for monitoring BACS systems, such as: independence from specific low-level data formats; support for heterogeneous hardware and software; and ability to meet the different real-time requirements of different diagnosis tasks; ability to filter collected data.

Xu et al. [2016] proposed a bloom-filter based analytic framework, which they used for to an extended analysis (over 18 months) of real-world home network traffic.

Liu et al. [2015] analysed the impact of net metering technology on detection of cyberattacks targeting smart home energy pricing. More specifically, the authors developed a smart home energy pricing cyberattack detection framework which integrates the net metering technology with short/long term detection (based on support vector regression).

The approach proposed by Pedro and Silva [2007] enables the development of generic monitoring and generic command of home automation facilities, independently of the underlying BACS technologies. This approach is based on

DomoBus technology (Renato Nunes [2016]), which through its device abstraction model and communications service allows the development of easily configurable applications from XML files. This enables monitoring and controlling device networks based on heterogeneous technologies. The main tests and results presented by the author were obtained in a testbed based on standard KNX components.

#### 2.2.3.2 Anomaly Detection

Zheng and Reddy [2017] describes *The Driven*, an anomaly detector for BACnet that is able to detect suspicious traffic in BACS networks with a small rate of false alarms. A dataset of BACnet traffic was also created, using Wireshark to capture traffic traces with detailed data: timestamp, source and destination IP, port number, packet length, and data payload. *The Driven* uses different mechanisms, according to three different types of traffic (data):

- Time-driven Traffic – used to determine if a flow-service stream presents time regularity behaviour at different time scales, and which regularity patterns it follows.
- Human-driven Traffic – generated by operators from the server or workstation. It constitutes around 5 percent of the total BACnet traffic and does not present time regularity.
- Event-driven Traffic – triggered by other service messages or changes in the system. Similarly to human-driven traffic, it also presents no regular/periodic behaviour, and represents a small volume of overall traffic.

The authors concluded, from their analysis, that (i) aggregated BACnet traffic does not exhibit diurnal patterns nor look strictly periodic because it consists of time-driven messages with different periodic behaviour as well as non-periodic streams; and (ii) the non-periodic traffic includes human-driven and event-driven traffic.

Pan et al. [2014] also presented an anomaly detection system for BACnet. This is a rule-based system which is trained with data flows that are dynamically captured from a Fire Alarm System testbed. Rules are generated by applying an inductive-rule learning algorithm (RIPPER Cohen [1995]). Authors tested their system with a number of well-known attacks, and concluded their platform can detect attacks against the BACnet protocol with a low rate of false positives, but the used testbed is rather simplistic and the injected attacks are also straightforward, making it difficult to extrapolate achieved results to larger buildings or more sophisticated attacks.

Pan et al. [2016] presents an anomaly based IDS that monitors BACnet traffic to extract its features (e.g. packet flow amount, header, payload) in order to describe the behaviour of BACS assets. More specifically, collected features are modeled into two types of data structures. Behaviour analysis methods including Discrete Wavelets Transform (DWT) and rule based anomaly behaviour analysis are implemented for their detection. Finally, a rule based attack classification is performed to trigger proper counter measures.

An autoencoder neural network was used by Legrand et al. [2018] for anomaly detection in BACS. The key point of an autoencoder is the dimension reduction taking place in it. Over training, an autoencoder neural network learns to approximate two functions: the encoding function that execute the dimension reduction and compresses the data; and the decoding function that recreates an approximation of the original input (the output). In this paper, autoencoders are used to measure the distance between a set of input and output vectors, establishing a threshold for anomaly classification. The authors used the REFIT dataset (Firth et al. [2017]) of smart home measurements to test several recurrent and convolutional models, having concluded that recurrent autoencoders appear to be the best candidates in the field of neural networks applied to the detection of anomalies in connected buildings. While results are interesting in the scope of anomaly detection in general, the nature of the REFIT dataset makes it difficult to extrapolate conclusions to the scope of cybersecurity.

#### 2.2.3.3 Intrusion Detection System

Fauri et al. [2018] present an IDS for BACS that detects known and unknown attacks, as well as anomalous behaviour. It does so by leveraging BACnet protocol knowledge and semantics. A BACnet parser is used to extract the relevant message fields from each message, in order to create a white-box model of the nominal system behaviour. Additionally, a human domain expert manually refined a collection of known BACnet threats into attack patterns. Once an attack is detected, the system generates enriched alerts that include semantic information helpful to the operators.

The use of active model discrimination with application to fraud detection in BACS is proposed by Harirchi et al. [2017]. The active model discrimination problem aims to find optimal separating inputs that guarantee that the outputs of all the affine models cannot be identical over a finite horizon. This will enable a system operator to detect and uniquely identify potential faults or attacks, despite the presence of process and measurement noise.

Context aware and anomaly behaviour analysis IDS for BACS were discussed and presented by Pan et al. [2019]. This paper describes an implementation of such an IDS, for a BACnet system, that involves five phases:

- Feature acquisition;
- Context modelling, based on BACS Context Aware Data Structure;
- Behaviour analysis;
- Threat assessment;
- And actions management.

In the first phase, features are selected and acquired from various BACS sources. During the second phase, the collected features are grouped and mapped into a well-defined behaviour context model named Protocol Context-Aware Data

Structure. In the third phase, the runtime models are generated and compared with those that are associated with normal BACS operations, in order to detect any malicious behaviours that might have been triggered by attacks against the BACS network and its services. The model comparison is performed with respect to both security and functionality. In the last phase, the detected attacks are classified according to their mechanisms and asset targets. In addition, a threat level is calculated in order to quantify the attack severity and, consequently, determine the appropriate defensive actions.

A fully automated approach to deploy specification-based IDS at network level was implemented for BACnet by Esquivel-vargas et al. [2017]. The creation of specifications often require human intervention, but this works proposes an automated approach supported by BACnet protocol where properly certified devices are demanded to have technical documentation stating their capabilities. The authors leverage on those documents to create specifications that represent the expected behaviour of each device in the network.

Rehman and Gruhn [2018] proposed a solution that has a firewall between the net/LAN and the Internet Service Provider (ISP), for protecting smart home and IoT environments. That firewall acts like a filter between the home appliances' interfaces and the Internet.

#### 2.2.3.4 BACS Improvements

Shuai et al. [2019] propose an efficient and anonymous authentication schema for smart home environments, using Elliptic Curve Cryptography (ECC). Computational costs, communication overhead and energy consumption costs are evaluated in this paper.

Still in the field of improved authentication solutions for Smart Home and IoT environments, Li et al. [2018] proposed SecHome, a large-scale home system using the Hierarchical Identity Based Encryption (HIBE) protocol. When a homeowner begins defining a smart home, he/she issues a secret key to house members based on the house hierarchy. Then, when any house member buys a smart device, he or she issues a private key to connect that device to the private network. This private network communicates with the public cloud using encryption, making data confidential, and allowing remote control. To enable users to control and access smart home devices, proper hierarchy and authentication are required in addition to said encryption. The root of the hierarchy can control all devices. The lower levels only see and control the ones below them and the devices on the leafs, corresponding to their branches of the hierarchy.

Werner et al. [2018] discuss suitable access control mechanisms specifically tailored to Web-connected smart home platforms. Then, they present their experiences from implementing access control solutions meeting the identified requirements in OpenHAB.

A lightweight symmetric keychain encryption and authentication for BACS, to distribute and manage session keys between Human Machine Interfaces (HMI) and Programmable Logic Controllers (PLC), is proposed by Ng and Keoh [2018]. A prototype was implemented using the BACnet/IP communication protocol. The schema facilitates automatic renewal of session keys, periodically, based on the use of a reversed hash-chain.

A pen testing approach for the assessment of a distributed Modbus-based BACS is proposed by Tenkanen and Hamalainen [2017]. This approach is applied to data flow recognition and environment analysis. Methods for risk mitigation are also suggested by the authors.

The creation of an additional level of security to control authentication violation cases, beyond the traditional authentication method and based on the user's behaviour, is proposed by Rath [2017].

The addition of hardware-based node authentication, over Transport Layer Security (TLS) connections, was proposed by Fischer et al. [2017b]. The use of identity-based signcryption for smart homes was addressed by Ashibani and Mahmoud [2017].

An alternative approach to BACS security is presented by Bondarev and Prokhorov [2017]. Instead of focusing on communication patterns or specific intrusion vectors, the proposed approach is concerned about the robustness of process-level data (e.g., sensor feeds). For this purpose, parameter filtering techniques are applied, in order to safeguard systems from taking wrong actions based on faulty or maliciously injected data.

## 2.3 Open Issues and Research Opportunities

A single BACS may have hundreds or even thousands of devices to monitor. Most of the available research works focus on exploring and adapting the existing knowledge from ICT and IACS areas (cf. section 2.2), often without addressing the specific requirements of BACS. In general, the proposals reviewed in this chapter reveal that the approach to BACS security is still in its infancy, especially when compared to more generalist ICT applications fields.

In general, a suitable BACS monitoring solution should include devices capable of collecting data and performing DPI of the BACS messages, at local level. Eventually, the design of an encompassing security solution for BACS may cover aspects ranging from specialized probes, such as domain-specific honeypots or traffic analysis devices to the creation of Security Information and Event Management (SIEM) solutions to acquire, aggregate and process collected evidence. There is also space for forensic capabilities, in order to create knowledge and enable the analysis of past events.

Regarding the detection of anomalies, BACS have a particularity when compared to other automation systems: the need to distinguish between traffic resulting from automated actions and events and traffic resulting from asynchronous human actions (e.g. a user enters a room). This increases the complexity of anomaly detection, especially for approaches based on the establishment of nominal reference operation models, something which some authors tried to address by

Table 2.3: Mapping of referred research works

| | | | | | |
|---|---|---|---|---|---|
| Building automation | IoT | Security | | | Santos [2018]; Dutta and Wang [2018]; Ali et al. [2017]; Fischer et al. [2017a]; Abunaser and Alkhatib [2019] |
| | | Architectural Solutions | | | Mocrii et al. [2018]; Lilis et al. [2017]; Qiu et al. [2018]; Minoli et al. [2017]; Bajer [2018]; Jia et al. [2019]; Darabseh and Freris [2019]; Asadullah and Raza [2016]; Ray [2017]; Ilieva et al. [2016] |
| | BACS | Standards | | | EN [2014]; ASHRAE [2020]; ANSI [2010]; KNX Association [2020a]; MODICON [1996]; Initiative [2019]; Toschi et al. [2017]; Usman et al. [2019]; Seifried and Kastner [2017]; Hersent et al. [2012]; Zhibo et al. [2017]; Wendzel et al. [2017] |
| | | Energy | | | Serrenho and Bertoldi [2019]; Groote et al. [2017]; Initiative [2019]; Komninos et al. [2014]; Rathinavel et al. [2017] |
| | | Architectural Solutions | | | Butzin et al. [2017]; Fatehah [2018]; Bugeja et al. [2018]; Li [2018]; Zhibo et al. [2017]; Vanus [2018] |
| | | Security Analysis | Vulnerabilities | | Brooks et al. [2017]; Lei et al. [2018]; Valli et al. [2017]; Gai et al. [2018]; Meyer et al. [2017]; Brauchli and Li [2015] |
| | | | Attacks | Management Automation | Zetter [2013] |
| | | | | Network | Ling et al. [2017]; Krishnan et al. [2017]; Hallman et al. [2017]; Peterson [2019]; Wright [2019]; Deng [2018] |
| | | | | Protocol | Matthew Peacock [2018]; Gasser et al. [2017]; Krishnan et al. [2017] |
| | | | | Field Level | Mundt and Wickboldt [2016]; Molina [2015] |
| | | | | Other | Macaulay and Singer [2011]; Levy [2015]; Lei et al. [2018]; Gai et al. [2018]; Saxena et al. [2017]; Krebs [2014] |
| | | | Taxonomies | | Liu et al. [2018]; Heartfield et al. [2018]; Anwar et al. [2017]; Graveto et al. [2019] |
| | | Safety | | | Brooks et al. [2017]; Nicklas et al. [2016]; Chhetri and Motti [2019]; Sutherland et al. [2015]; Han et al. [2018] |
| | | Privacy | Users Feedback | | Zeng et al. [2017]; Lin and Parkin [2020]; Kaaz et al. [2017]; Pathmabandu et al. [2020] |
| | | | Use cases | | Franke et al. [2016]; Mundt et al. [2012] |
| | | | Data Analysis | | Xu and Agung Julius [2019]; Alisic et al. [2020] |
| | | | IoT Implementations | | Santo et al. [2017]; Gao et al. [2020]; Iqbal et al. [2021]; Samarah et al. [2017] |
| | | | Energy | | Finster and Baumgart [2015]; Pham and Mansson [2019]; Sarbhai et al. [2019]; Wu et al. [2016]; Dasari et al. [2021] |
| | | | Other | | Hamberger, C.; Eastman, C. [1964]; George et al. [2020]; Kraemer and Flechais [2018] |
| | | Contributions | Monitoring | | Minoli et al. [2017]; Abdulmunem et al. [2016]; Chowdhury [2019]; Vasyutynskyy et al. [2006]; Xu et al. [2016]; Liu et al. [2015]; Pedro and Silva [2007] |
| | | | Anomaly Detection | | Zheng and Reddy [2017]; Pan et al. [2014, 2016]; Legrand et al. [2018] |
| | | | IDS | | Fauri et al. [2018]; Harirchi et al. [2017]; Pan et al. [2019]; Esquivel-vargas et al. [2017]; Rehman and Gruhn [2018] |
| | | | BACS Improvements | | Seifried and Kastner [2017]; Shuai et al. [2019]; Li et al. [2018]; Werner et al. [2018]; Ng and Keoh [2018]; Tenkanen and Hamalainen [2017]; Rath [2017]; Fischer et al. [2017b]; Ashibani and Mahmoud [2017]; Bondarev and Prokhorov [2017] |
| | | | Other | | Demeure et al. [2016]; Wang et al. [2015, 2017]; Handa et al. [2019] |
| | | Market | | | Brooks et al. [2017]; Khedekar et al. [2016]; Groote et al. [2017] |

using systems based on rules, auto-encoders, support vector machines and/or discrete wavelet transforms.

The intrusion detection systems found in the literature are mostly based on rule-based approaches allowing for the identification of attacks or abnormal functioning, such as deviations from the expected operational behaviour. The majority of presented examples are mostly based on small testbeds, not representative of real world scenarios.

Many of the analysed proposals address BACS security mainly by means of evolving the BACS protocols, which is not an acceptable solution for legacy equipment already existing facilities. A noteworthy exception is the work developed by Bondarev and Prokhorov [2017], which proposes a different approach to the problem, based on data and not on protocols, as a possible methodology to increase the robustness, security and effectiveness of BACS.

Most studies focus on management and automation levels, thus creating space for new directions of research focused on the field level. Presented examples deal with IP communications, leaving direct messages between devices to be explored. Those communications use local and specific networks that may vary from protocol to protocol.

At field level, where the interaction with the physical systems takes place, it should be possible to identify threats and anomalies. From this perspective, Single Board Computer, connected to the field level for monitoring purposes could act as Network Intrusion Detection System (NIDS) devices. Additionally, these devices could also be used to sniff the IP network, where the Management and Automation actions take place, to enrich the obtained information and add value to the overall security system.

Another general gap in this field relates with the absence of useful datasets, based on real testbeds and capable of supporting validation work. This translates in two needs:

- Obtaining datasets and making them available to the scientific community. These must contain communication captures at the various levels, but especially at the field level (since at the management and automation level some of the already existing network capture datasets can be used).
- Documentation and characterization of real environments and on-site data collection, including the various existing devices and implemented home automation functions as well as labeled datasets.

The absence of these elements is hampering and limiting the scope of research in this area. In order to address these limitations, it makes sense to develop appropriate capture mechanisms to enable extraction of field-level datasets.

The amount of data obtained with a probe directly connected to the field bus, and the packets collected through the network port, represent a large amount of valuable data. This points to the potential of using low-cost SBCs connected to the field bus to act as specialized probes able to capture and analyse field network traffic, for security purposes. While this approach may sound interesting from a cost/practicality perspective, one must take into account the limited computing capabilities of the hardware platforms, which may impose some design choices and/or compromises, namely:

- The construction of analysis models should happen during an initial learning phase, or the information might be sent to an external processing unit, with more capacity, to build the model and then import it back in the SBC-based probe;
- Data stream processing should be handled with a throughput compatible with a buffer at the scale of the SBC;
- The data lifecycle should be handled using tight rules, concerning local storage of data (due to the limited capacity of the probes) and longer-term storage in central locations, for deeper analysis or forensics.

With the identified challenges, a non-restrictive list of available anomaly detection techniques includes, for instance:

- Classification-based techniques, such as static neural networks, some of the support vector machine variants or rule-based methodologies, used in two steps to create a model and test during the evaluation phase;
- Clustering-based techniques, with the assumption that the clusters are computed on the initial learning phase;
- Statistical-based techniques, on which the stochastic model is pre-processed;
- Also, the use of Finite State Automatons and Markov chains could provide good results, keeping the model definition off-path of the testing process.

## 2.4 Summary

This chapter provides a comprehensive perspective on the BACS security and privacy landscape. From this analysis, it becomes apparent that the majority of the published research works are focused on the automation and management level of the BACS architecture, often considering the use of IP-based protocols at such levels. For such reasons, existing knowledge from ICT systems is frequently adapted and enhanced to overcome the differences, between BACS and ICT.

Due to the aforementioned reasons, the specific nature of field-level protocols and technologies is often overlooked. For BACS this also means that datasets are scarce, especially the ones containing BACS-specific protocol traces – something that constitutes a crucial limitation when it comes to fostering further research and developments regarding BACS security. This finding opened the opportunity to create a new specific dataset presented in Chapter 5.

Local tampering is a reality and lots of threats exist at the field level. Thus, safety and security measures should encompass this level, which opens up a wide area of future research. In addition, all information collected at local level, at several points of the field network, can be sent to centralized and more robust systems for detecting anomalies or attacks, thus increasing the detection probability in complex BACS, using more powerful systems.

Existing threats, as well as gaps detected in mitigation techniques and mechanisms, are at the foundation of the development of the concept and architecture presented in the next Chapter.

# Chapter 3

# Proposed framework

## Contents

Building Automation and Control Systems (BACS), as detailed in the previous chapter, are traditionally based on specialized communications protocols, such as KNX or BACnet, and dedicated sensing and actuating devices.

Despite the increased awareness about the security risks associated with BACS, there is generally a lack of security tools for protecting this particular breed of cyber-physical systems. General-purpose security tools aggravate this and typically cannot cope with the specific requirements and technologies associated with BACS. This makes it necessary to devise domain-specific approaches – as shown, for instance, by the KNX Secure initiative led by the KNX Association. Nevertheless, despite the advances KNX Secure and similar initiatives brought, there is still a considerable gap between the security needs of BACS and the solutions available.

In this thesis, this gap is addressed by proposing a Network Intrusion Detection System (NIDS)) designed specifically for BACS. This NIDS is protocol-agnostic and can potentially support different BACS protocols and technologies, such as KNX, BACnet, Modbus or mixed ecosystems, without loss of generality. A specific proof-of-concept implementation of this NIDS concept for KNX – one of the more widespread BACS protocols – is presented in Chapter 4. To this purpose, a real-world KNX deployment was used to showcase and evaluate the proposed approach (cf. Chapter 5).

## 3.1 Scope and Motivation

Over the last few years, there has been a progressive mindset shift in the automation domain towards considering security as much of a critical requirement as reliability or safety. From this perspective, Building Automation and Control Systems (BACS) constitute no exception, as both the need for monitoring the proper operation of physical devices and the security of the whole building operation should be considered crucial requirements. Security and safety are vital if one considers the increasing permeability between building automation and traditional IT systems. This interaction increases the security challenges faced by BACS, as most of the existing implementations were initially designed with isolation as an acquired safety guarantee.

The growing awareness about existing problems led to various improvements to the standards used in building automation incorporating authentication and encryption mechanisms (e.g., KNX, BACnet, and ZigBee). However, in most existing installations, it will not be easy or even possible to retrofit these improvements – existing devices' memory lack and/or computational power to implement the required security features. Moreover, even buildings where these improvements are retrofitted remain vulnerable to a wide range of attacks.

In this thesis, the concept of a domain-specific NIDS for BACS is proposed to mitigate this situation. Such NIDS is a monitoring probe that can intercept all the Fieldbus traffic of the building automation network (control network) and also observe the LAN. In this way, all the messages can be processed for threat detection (e.g. firmware updates, command messages, sensor outputs, actuator inputs and status reports).

The feasibility of this BACS-specific NIDS approach depends on the implementation of two key capabilities. First, it must incorporate anomaly detection mechanisms to ingest the data obtained from the legacy BACS control network and the traditional LAN, to detect system anomalies and potential cyberattacks. Second, this domain-specific NIDS must target an appropriate cost-efficiency balance, ideally within the magnitude of a single device Manufacturer's Suggested Retail Price (MSRP).

This proposition provides a viable (albeit not perfect) alternative to implementing a security layer for existing deployments by adding a new device instead of undergoing mass replacement of existing equipment. Moreover, this approach is complementary even for modern deployments incorporating security-oriented features. In this proposal, both challenges are addressed.

The current section (3.1) introduces the context and motivation for the current chapter, also providing two background subsections on BACS-specific security and intrusion detection, for the sake of consistency and completeness. Next, the proposed concept (Section 3.2) is detailed along with the associated requirements (Section 3.3) in the scope of BACS. The proposed architecture is presented in Section 3.4. Finally, Section 3.5 concludes this chapter. Later Chapter 4 (Section 4.3) will further detail the proposed architecture, in the scope of the PoC that was developed for KNX.

### 3.1.1 Cybersecurity and BACS

As already pointed out, BACS security breaches are often considered to be a consequence of using systems, protocols and standards that were originally conceived to operate in isolated environments, without any connection to ICT networks or the Internet. This is aggravated by the fact that many legacy devices cannot be patched, often meaning that only isolation or complete replacement might ensure adequate security (see Wendzel et al. [2017]). In general, most attack categories that are characteristic of IACS may somehow be transposed to BACS scenarios (see Macaulay and Singer [2011]). However, even though some the protection strategies used in IACS might somehow provide hints on how to keep BACS secure, there are considerable context differences that require domain-specific approaches. Similarly, the protections typically used in IACS at network level can also be adapted to BACS.

For BACS-specific studies, Wendzel et al. [2017] performed a comparative analysis of the security issues for some of the most widespread BACS communication protocols (KNX/EIB,BACnet, ZigBee and EnOcean). Also, Graveto et al. [2022a] analysed several attacks to real-world BACS, such as the attack to the St. Regis Shenzhen luxury hotel (Molina [2014]), the attack on Google's Wharf 7 building workspace (Zetter [2013]), the attack on Target Corporation which was used to access the company's BACS systems (Krebs [2014]), and the attack on the Singapore Fragrance Hotel (Deng [2018]). Overall, most of those attacks could have been easily detected and/or prevented with adequate monitoring and intrusion detection mechanisms.

There are also reports about an incident in Germany (Higgins [2021]), where a building automation engineering company lost contact with roughly three quarters of the BACS devices in an office building system network, after being locked out of the system by a cyberattack. The company was forced to revert to manual operation of central circuit breakers to control the lights in the building, until a security consultant was able to retrieve the Bus Coupling Unit (BCU) key from a bricked device's memory, in order to regain control. Once again, adequate monitoring mechanisms could have mitigated the impact of this attack – even basic system logs were unavailable to support forensics analysis, making system recovery much more difficult.

### 3.1.2 Intrusion Detection for BACS Scenarios

As already discussed, several successful attacks have occurred and existing mechanisms were unable to detect them, pointing to the need of better intrusion detection systems for BACS. Actually, there is already extensive work in anomaly and intrusion detection in related areas, such as cyber-physical systems and IoT (see Mitchell and Chen [2013]; Buczak and Guven [2015]; Ahmed et al. [2016]; Zarpelão et al. [2017]), but in the specific topic of building automation there are much less proposals.

The solution proposed by Pedro and Silva [2007] addresses the development of generic monitoring and actuation of home automation facilities for use with different technologies. This solution is based on the DomoBus technology, which through its device abstraction model and its communications service is supposed to enable the development of configurable applications from XML files – allowing the monitoring and control of device networks of several technologies.

Jones et al. [2018] used a SBC to deploy an unsupervised artificial neural network to monitor building automation systems and improve their resilience. The proof-of-concept used BACnet and all the network packets were stored and analyzed using an on-board Adaptive Resonance Theory neural network. When anomalies are found, the source and destination addresses are added to an access control list and those communications are blocked. In our opinion, this type of automated reactions may become a problem for the overall performance of the building system and even constitute a new attack vector to be exploited by malware, as already known in similar domains such as IACS. This could be solved by means of human intervention in the reaction process, for improved safety.

A multi-modeling based approach, using a mix of modeling, simulation and analysis tools, was used by the CPS Association to design the INTO-CPS cyber-physical platform (INTO-CPS Association [2020]), which has been used to assess the security of smart buildings, using a *man in the middle attack* attack for validation purposes (see Mace et al. [2018]).

A Hidden Markov model is proposed by Ramapatruni et al. [2019] to identify anomalous activities. Sensor data from a smart home environment was used to train this model.

Chavis et al. [2020] proposed a system that helps reducing the cognitive load on a user in keeping his smart home network secure. Machine Learning (ML) is used to achieve that goal, with data collected and stored in Packet Capture (PCAP) format (documented by The Tcpdump Group [2020]).

Since the amount of previous work related with intrusion detection in BACS is so scarce, one might also look at works addressing intrusion detection in other domains.

The open source projects Snort (Cisco Systems, Inc. [2020]), Suricata (Open Information Security Foundation [2020]) and Zeek (Paxson [2020]) are among the most widely used NIDS in Transmission Control Protocol over Internet Protocol (TCP/IP) networks in general. Bhosale and Mane [2015] review these three solutions, concluding that Snort and Suricata are mostly signature-based IDS, while Zeek describes itself as a passive open source network traffic analyser for security monitoring purposes – e.g. capturing traffic and forwarding it to SIEM systems.

Dupont et al. [2019] surveyed NIDS for Controller Area Network (CAN) systems. The CAN bus, mostly used for in-vehicle control systems, is an automation technology that also poses a challenge to intrusion detection systems – with a large number of messages that, without context, carry very little information. In some way, this is similar to BACS, where the majority of messages carry only binary information such as ON/OFF or UP/DOWN.

Considering the foreseen requirements for BACS NIDS in terms of embedded security probe capabilities and deployment, other works were also considered in this analysis. Al-Maksousy et al. [2018] presented a real time monitoring system with very low CPU usage that is capable of detecting and classifying malware based on network behavior using split deep neural networks. The effectiveness of flow-based vs packet based NIDSs was evaluated by Ficke et al. [2019], that analyse the gap between flow-based and packet-based NIDS in terms of detection effectiveness. Hinting towards possible optimization strategies, Gouveia and Correia [2016] evaluated the performance of NIDSs with feature set tuning and reduction.

Robinson and Kim [2017] validate their intrusion detection framework by using a ModbusTCP control system, based on a SBC that enables simulation of cyber-attacks and illustrates a mitigation measure with the added feature of Modbus monitoring using Snort. Also in this line, Graveto et al. [2019] proposed the Shadow Security Unit (briefelly presented in Section 4.4), a monitoring probe designed to be attached in parallel to IACS control devices, being able to passively monitor the network communication flows and the physical process interfaces, in order to detect anomalies with potential impact on system safety and security.

## 3.2 Concept

The discussion presented in the previous section clearly identified an existing gap regarding BACS security mechanisms. This constitutes an opportunity for the development of the BACS NIDS, an intrusion and anomaly detector that operates mainly at the control network level (fieldbus). The concept and main requirements are outlined in this section and following Section 3.3, with the architecture being presented in Section 3.4.

The proposed NIDS fits into the BACS security architecture, as shown in Figure 3.1, supporting local monitoring of:

- The building fieldbus where all the control devices are connected (e.g. actuators, blind control, sensors, light dimming, heating control systems).
- And the local area network, where commodity devices such as computers, phones and other devices (e.g. IoT devices) may be present.

The BACS NIDS provides a Wide Area Network (WAN) connection that enables remote management for building owners and BACS integrators. This connection uses a side channel and prevents direct access to the local control network, thus guaranteeing security and safety.

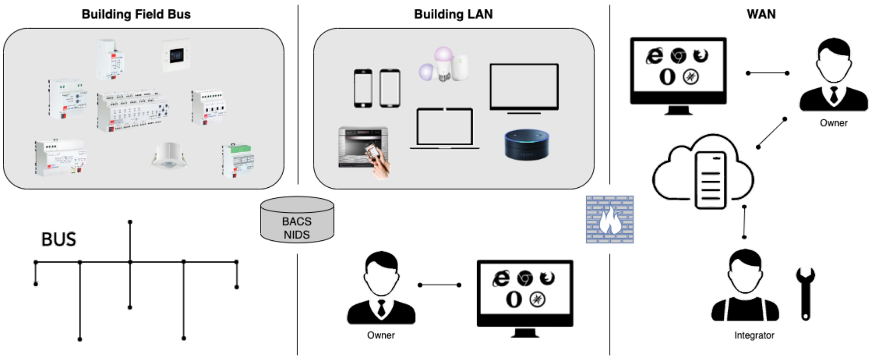


Figure 3.1: BACS Security Architecture

The BACS NIDS also features a Web-based dashboard for (optional) local management, together with a secure interface for management and event feeds. The later is intended to provide integration into larger security frameworks, as well as support for off-site management by owners and building automation integrators.

The proposed integration model for the BACS NIDS is illustrated in Figure 3.2. It is built around an SBC with two interfaces, one connected to the building fieldbus (twisted pair or other support medium) and an Ethernet interface connected to the local building LAN. The proposed approach is also compatible with multiple building automation standards (e.g. KNX, BACnet, ZigBee, EnOcean), despite KNX being the target protocol for the PoC implementation presented in Chapter 4.

The BACS Security Management Platform constitutes a web application that can be either locally served by the BACS NIDS host or remotely exposed by a

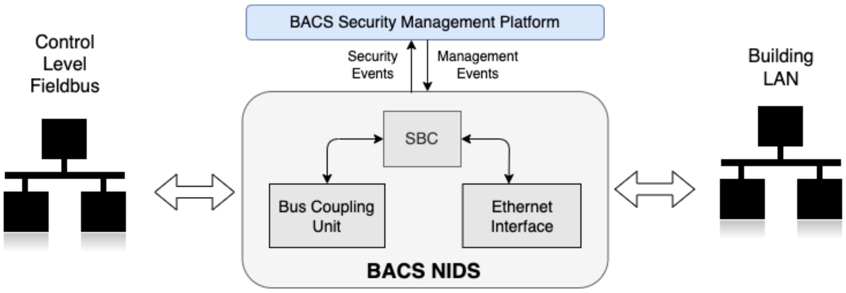


Figure 3.2: BACS NIDS Integration Model

web server, using an out-of-band Ethernet connection. The *security events* generated by the system are forwarded to the platform, processed and presented in a friendly User Interface (UI). This same UI may also provide management and configuration capabilities, supported by BACS NIDS *management events*.

## 3.3 Requirements

The design of the BACS NIDS considers the following key principles:

- Seamless and transparent operation – by design, most of the required evidence for processing should be obtained by passively monitoring the protocol message flow, without any interference in the normal BACS operation.
- Cost-effectiveness – the NIDS must be cost-effective when compared to regular automation devices, ideally being within the same price range.
- Protocol readiness – The device should be fully compatible with the target BACS standards (e.g. KNX), for stealth operation and perfect compatibility with existing BACS devices.

In addition to those design principles, the NIDS is expected to address the following key requirements:

- Semantic command stream processing – the NIDS should be able to transparently capture and decode inflight protocol messages (e.g. KNX TP or KNX/IP).
- Reliability – when evaluated under stress testing, the NIDS should obtain consistent results for the same given use cases and should clearly identify ongoing anomalies or attacks.
- Trustworthiness – the NIDS must provide trusted results under different test use cases, encompassing validation of accuracy and false positive rates for the system.
- Stealthiness – the system should be invisible to outside attackers, which in addition to the aforementioned passive monitoring abilities, implies the support of (optional) out-of-band communication channels with the BACS security management platforms, to avoid exposure in the LAN or in the field network (e.g. KNX TP).

- Auditing support – information and event records (both for alerts and metrics) must be preserved for auditing purposes.

The following Section, presents a reference architecture for the proposed BACS NIDS.

## 3.4 Reference Architecture

The architecture of the proposed BACS NIDS is presented in Figure 3.3. It is based on a neutral concept that is compatible with the majority of BACS protocols (e.g. KNX, BACnet, Modbus), communications technologies (e.g. KNX TP, KNX RF, Ethernet, Zigbee, RS 485) and deployment scenarios. Nevertheless, for sake of readability, some of the technical details presented and discussed next, in Chapter 4, directly relate with the PoC prototype, which is based on a KNX TP bus.

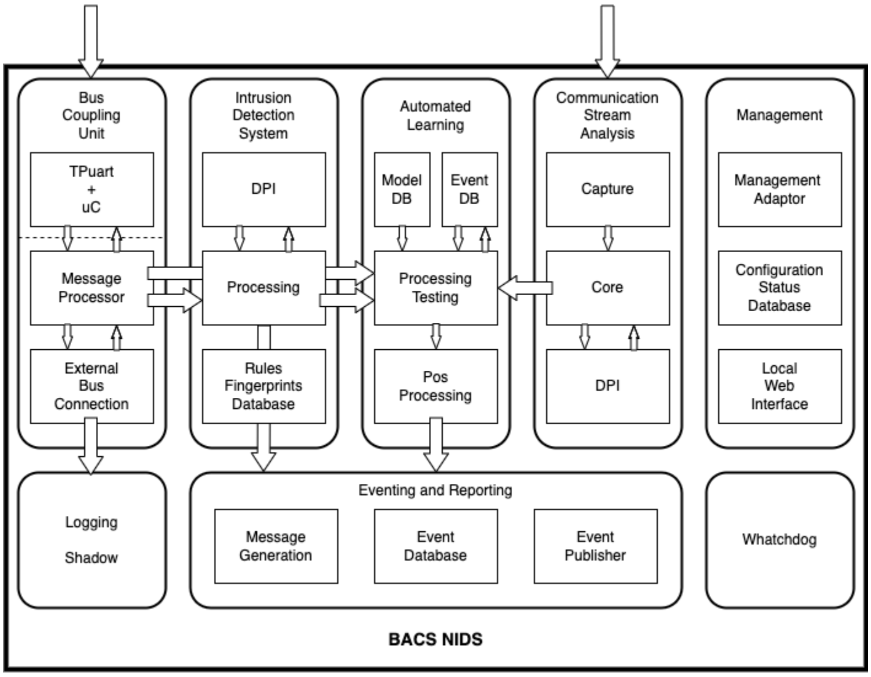


Figure 3.3: BACS NIDS Architecture

The main building blocks of the reference BACS NIDS architecture, further detailed and discussed in Section 4.3, are:

- Bus Coupling Unit – provides the connection to the fieldbus network along with some pre processing and messages time stamping when needed;
- Intrusion Detection System – enables the use of rules and heuristics in detection;
- Automated Learning – enables the use of artificial intelligence techniques for detection;
- Communications Stream Analysis – provides the connection to the building LAN network;

- Management – allows either locally or remotelly to manage the BACS NIDS, and also enables data visualization tools;
- Shadow Logging Module – empowers the detection of altered messages;
- Eventing and Reporting – provides local storage of produced logs and the connection to external BACS security management platform; their;
- Watchdog – enables the keep alive testing/rebooting of the BACS NIDS.

## 3.5 Summary

This chapter provides the scope and motivation supported by the existing cybersecurity problems in BACS. Overviews intrusion and/or anomaly detection scenarios in BACS as evidence for the underlying motivation for the development of this thesis, supported by the existing security gaps and the lack of equipment and techniques that improve security conditions in BACS.

The work carried out allowed the design of a new concept of cybersecurity approach in BACS, which consists of developing a stealth device for monitoring and processing messages on the Fieldbus and LAN level. The proposed architecture enables the use of rules, heuristics and artificial intelligence to detect attacks and anomalies in BACS.

Next, Chapter 4 embodies an implementation of a Proof of Concept BACS NIDS that specifically supports the KNX protocol.

# Chapter 4

# Proof of Concept

## Contents

In this chapter, we present a Proof of Concept (PoC) developed for the particular case of KNX-based BACS. Taking into account the concepts, requirements and architecture of the framework proposed in Chapter 3, Section 4.1 presents the PoC. Section 4.2 describes a laboratory testbed that was used in the development process and for the initial testing phases. The implementation and detail of the reference building blocks are presented in Section 4.3. Finally, Section 4.4 briefly describes some complementary work in the field of cybersecurity of industrial control networks. Section 4.5 concludes this chapter.

## 4.1 PoC Implementation

Figure 4.1 presents the external view of the KNX NIDS that has been built for demonstration and testing purposes, addressing the specific scenario of KNX TP BACS systems.

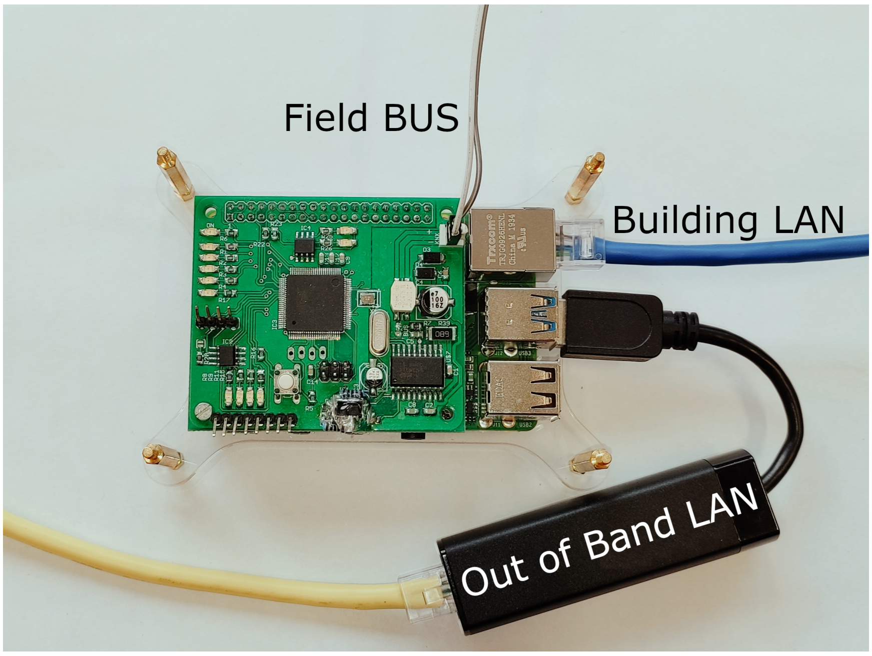


Figure 4.1: External view of the KNX NIDS

A Raspberry Pi 4 was adopted as host SBC due to its ease of use, availability, expandability and price/performance ratio, as well as the considerable amount of available documentation and related information sources. Moreover, a hardware watchdog is already supported, thanks to the native CPU watchdog driver for the RPi (Raspberry PI Foundation [2020]). Overall, the PoC cost was kept below 200€, which is deemed acceptable for a prototype, moreover considering that a mass produced system could easily cost a fraction of this value (probably less than 100€ for the most common types of fieldbus scenarios), due to savings obtained via bulk component acquisition and increased component integration.

The connection to the KNX bus was achieved with the development of a shield (see Figure 4.2), connected to SBC General Purpose Input/Output (GPIO) interface. This shield contains a twisted pair/Universal Asynchronous Receive Transmit (TP/UART) (optically isolated from the SBC host using a ILD213T optocoupler) that provides fieldbus connectivity, as well as an ATmega 2560 micro controller that establishes the interface between one of the SBC serial ports and the TP/UART serial line. An interface to a PCD8544 Liquid Crystal Display (LCD) was added to locally display device information, and a 24C02C Electrically-Erasable Programmable Read-Only Memory (EEPROM) was used to persist data.

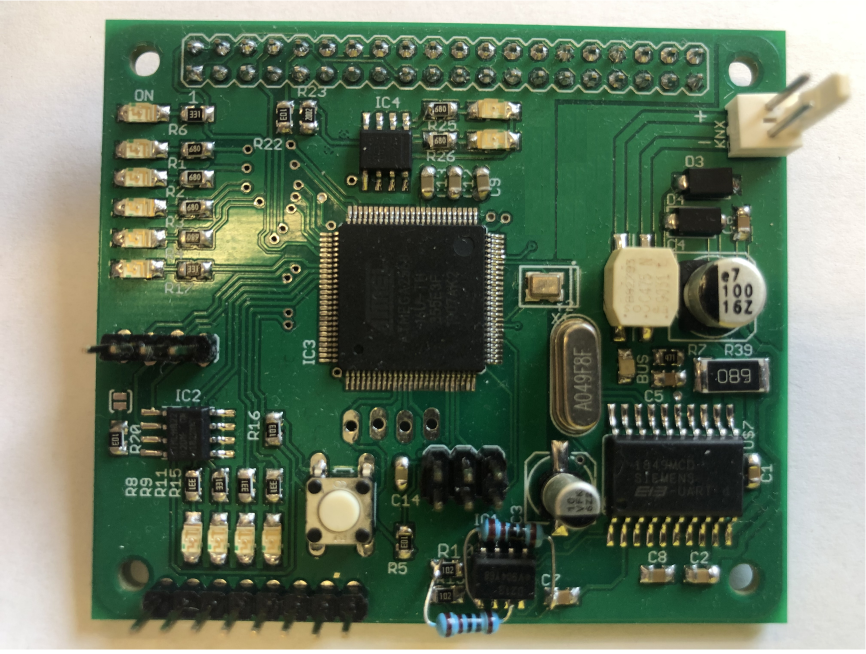


Figure 4.2: KNX TP shield for KNX NIDS

The KNX NIDS Operating System (OS) is based on a Debian Linux distribution (Raspbian), with Docker being used to deploy each functional module in a separate container, for sake of improved isolation, stability and reliability for the overall system.

Most of the modules were developed using the GO language, using the open source GoPacket library (Google Inc. [2022]) for decoding and encoding messages. A contribution to this open source project was produced and archived, extending this library to support the manipulation of KNX protocol flows. Communication between modules was implemented using the ZeroMQ library (ZeroMQ [2020]).

Apart from the KNX-TP shield (which was integrated into a single board), the whole prototype was built using COTS hardware components, with no special optimization. The ATmega micro-controller could implement the required KNX stack for commercial certification as a KNX device, and the SBC could be stripped from the unnecessary components, eventually being replaced by a Raspberry Pi *Compute module*. This means that a mass-produced KNX NIDS would take a fraction of the prototype footprint.

Overall, the presented PoC fulfils all the requirements deemed crucial for the implementation of a BACS NIDS (cf. Section 3.3). Section 4.3 details the building block modules and the capabilities of this current implementation.

## 4.2 The Laboratory Testbed

Figure 4.3 represents the KNX testbed that was developed to allow the replication of communications at the field bus level in the laboratory and to simulate KNX communications, for the purpose of testing the implemented framework.

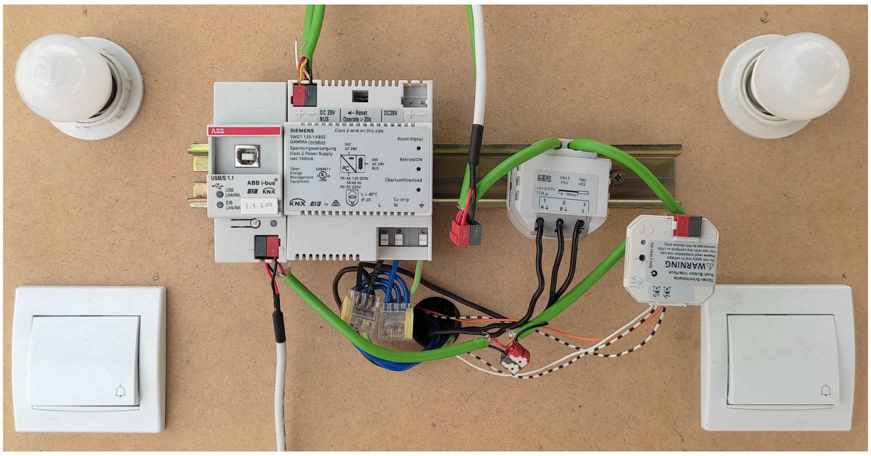


Figure 4.3: KNX Testbed

The KNX devices used in the testbed (see Figure 4.4), are:

- a USB / KNX TP gateway that allows connection to the KNX twisted pair fieldbus (green cable);
- a power supply that guarantees the fieldbus power supply to the intended 29V;
- an actuator that allows the connection of two normal switches to the system, allowing its operation;
- a relay module that allows the reception of information from the KNX bus and the supply/absence of power to the lamps.

Figure 4.5 represents the power schematic and bus lines of the developed testbed. As can be seen, in this architecture, there is total independence between the actuators/sensors and the relays that allow the equipment (e.g. lamps) operation. The data bus (e.g. KNX TP) transports messages between the various devices, thus guaranteeing the system operation. When a switch is activated, a GA message is sent to the network and the relay, which is configured to re-

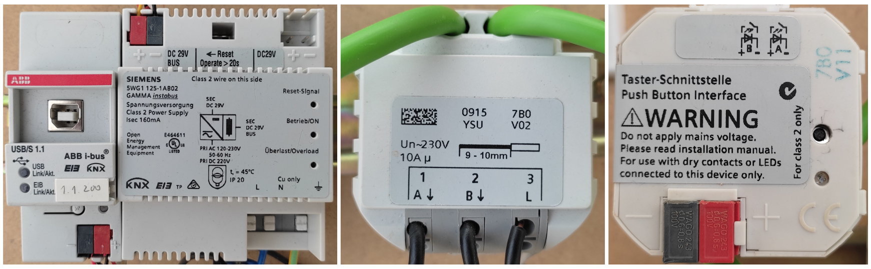


Figure 4.4: KNX Devices – USB Gateway / Power Supply / Relay / Actuator

ceive this message, will activate in order to turn the light on/off in the required channel.

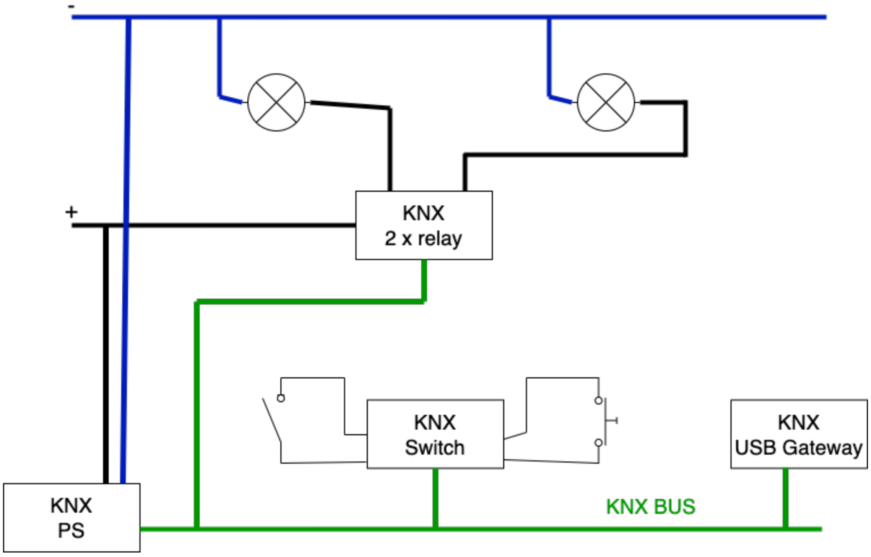


Figure 4.5: KNX Testbed – Wire Schematic

This installation is simple but yet manages to simulate most of the operations existing in a BACS. It can be configured, commissioned and debugged using ETS through the USB gateway. It also allows the connection of one or multiple KNX TP BACS NIDS. Its use was crucial for the development and test phases that preceded the evaluation process carried out in a real BACS, described in Chapter 5.

## 4.3 Building Blocks

The reference architecture materialized in a BACS scenario that uses KNX, without loss of generality, is presented and designated as KNX NIDS. The various modules that constitute it are described and detailed next. The sub-modules specifically designed for connection to the KNX TP bus and deep packet inspection in KNX are the only protocol-dependent ones. The rest of the implementation is generic and can be used in any BACS installation.

### 4.3.1 Bus Coupling Unit

The Bus Coupling Unit (BCU) is responsible for the connection with the fieldbus (control network), passively capturing all in-flight traffic. While this module could also be able to actively interact with the control network (e.g. for active device fingerprinting), such capability is not leveraged in the PoC implementation. The BCU encompasses three main blocks:

- The bus coupler is an hardware device that connects to the bus (e.g. KNX TP, Zigbee, Ethernet). In the case of KNX TP, a shield was developed using a TP/UART and a microcontroller (ATmega 2560) that handles real time needs of the protocol and message flows to/from the host SBC.
- a message processor that processes the telegrams collected from/to the fieldbus, using the well-known PCAP format. This block also enriches those messages with a timestamp, unifying the communication with other system modules.
- The external Bus Connection, that connects to the log system and online communication between external systems and the control network bus.

Naturally, the adaptation of this generic architecture for different BACS scenarios will impose the use of a specialized BCU modules for each type of system (e.g. KNX, BACnet, ZigBee). In some cases, when Ethernet transport is used, the Bus coupler may consist of a conventional network interface card.

### 4.3.2 Intrusion Detection System

The Intrusion Detection System (IDS) module provides the anomaly and/or attack detection capabilities. Depending on the capacity of the host SBC, the IDS may implement several detection techniques, such as signature-based (recognising bad patterns, such as malware), anomaly-based (detecting deviations from known *good* working model), or reputation-based detection (scoring the reputation of potential threats and raising an alert when a predefined threshold is reached). This module includes the following building blocks:

- Deep Packet Inspector – this block is used, when needed, to code/decode the byte array of the messages from/to the building automation protocol. This block also depends on the underlying BACS technology (KNX, BACnet, ZigBee).
- Processing – this block constitutes the core processing unit of the IDS, using raw messages from the BCU *message processor* and *automated learning* blocks as input, supported by a pattern database, for decision making. It can also use the DPI module, if needed.
- Database – the database used to persist and store the information about the patterns used to process and identify the anomalies or attacks (e.g. fingerprints, rules).

The aforementioned modules provide the basic protocol data acquisition and threat detection capabilities, which are to be complemented by an *automated learning* module described next.

### 4.3.3 Automated Learning

The information from the BCU, the IDS and the communication stream analysis feeds this module that locally implements a correlator and a classifier based on Artificial Intelligence (AI) techniques. The correlator may perform event reduction and aggregation within preferred time windows, as well as checking if commands arrive from a legitimate source, if they are coherent with the expected control interface flow, and if I/O information is in line with expected values. The classifier is supported by models that, due to potential limitations of computational resources of the NIDS device, may be built outside the NIDS device and later imported to the *Model DB*. To build the required models, a learning phase may be executed during an initial monitoring period before entering detection mode. Alternatively, a previously created dataset may be used to skip this stage. The correlation rules, module database and the learning processes are controlled via the *Management* module (cf. Section 4.3.5).

The *Automated Learning* module provides a generic framework where different models can be uploaded and classification algorithms can be deployed. It is supported by four functional blocks:

- Model Data Base (DB) – stores the preloaded modules to be used by the AI processor, as well as minor updates resulting from model improvements in real time.
- Event DB – this database stores the inputs or aggregated raw events, allowing for time window-based and and some minor batch processing.
- Processing/Testing – this is the core block where correlation and classification take place, producing the output of the *Automated Learning* module.
- Post-Processing – this block formats the module outputs accordingly with an established data model, for subsequent handling of user interfaces and interconnection with other, more comprehensive security systems.

This module is able to further extend the NIDS capabilities to detect anomalies and/or attacks.

### 4.3.4 Communications Stream Analysis

This module connects to the building LAN via an Ethernet interface. It is responsible for capturing network traffic and forwarding it to the corresponding modules, for processing and analysis. The use of a passive TAP allows to intercept traffic in a seamless way, thus hiding its presence from potential attackers.

The *Communications Stream Analysis* is functionally similar to the bus coupler module, but is connected to a LAN instead of a fieldbus. It is composed by three functional blocks:

- Capture – this block gets the byte array messages from the link, queues them and feeds the remaining blocks, also performing time-stamping.
- Core – this is the processing block that controls encoding, integrity checking and the connection to the IDS and Automated Learning (AL) modules.
- DPI – this block is invoked as needed for deep pack inspection and decoding.

### 4.3.5 Management

The *Management* module provides the means to configure and persist settings for the device, thus providing the main interface to the KNX NIDS, both for local and remote operations. It encompasses three key functional blocks:

- A management adapter that exposes a secure TLS connection for interaction with broader security management platforms and/or for remote access by integrators and building owners.
- A database service that provides access to the configurations of the various BACS NIDS modules, that are persisted in local storage – thus allowing for real-time device reconfiguration.
- A lightweight local web server that allows an operator to query basic functional indicators, data visualization and/or to configure the system.

### 4.3.6 Logging Shadow

This module provides a database to log incoming fieldbus protocol messages. It can be used either for auditing purposes or to assist debugging (possible) operational faults, by comparing the differences between intercepted messages and the ones actually sent.

### 4.3.7 Eventing and Reporting

The *Eventing and Reporting* module processes all the output events from the KNX NIDS, from both the IDS and the AL subsystems. This module takes care of the BACS NIDS security event stream, also persisting messages/events to provide tolerance to disconnected operation, in case of communication interruptions. It is supported by the following functional blocks:

- Message Generation – aggregates and processes all the events from other modules and generates the output events, aligned with an established data model.
- Event Database – persists the output events for retrieval by local or remote consumers.
- Event Publisher – implements the interface between the system and the security event consumers.

All the KNX NIDS outputs are processed via this module, thus ensuring integrity and consistent temporal sequencing for future analysis.

### 4.3.8 Watchdog

A watchdog module takes care of self-monitoring, for both component and service operation. This watchdog leverages the Docker framework to provide isolation, implementing a series of software routines that periodically check component operation and attempt recovery or restart in case of stalled operation. Moreover, it also provides system-level checks through a Linux kernel module working together with a watchdog service that provides regular feedback to the hardware watchdog timer of the SBC. Using the hardware watchdog makes it possible to reboot the entire BACS NIDS platform in case of a critical failure, after a predefined number of missed timer events.

## 4.4 Complementary Work

Within the scope of the research laboratory, complementary work was carried out within IACS that use SCADA systems. Like BACS, these systems are also supported by control devices. In this case, they are primarily PLC, Remote Terminal Units and Intelligent Electronic Devices. Such devices are deployed at the edge of the SCADA infrastructure and directly interface with the physical processes under control. They are often based on embedded systems with limited capabilities and exposed to significant security and safety-related risks, as demonstrated by past incidents such as Stuxnet (see Langner [2013]). However, despite the recognized relevance of those edge devices, they usually lack monitoring mechanisms to detect device anomalies and/or cyber-physical threats.

The main contribution by Graveto et al. [2019] was the proposition of a new approach for stealth monitoring of those control devices, useful for security and safety management. This approach builds on cost-effective probes, designated as Shadow Security Unit (SSU), directly attached to the monitored control devices. This privileged positioning enables the direct and fine-grained observation of both physical inputs/outputs (i.e. the processes under control) and network communication flows – allowing the exploitation of various novel monitoring approaches able to address sophisticated security threats not noticeable otherwise. Moreover, the SSU approach is not limited to SCADA scenarios, being also adapted to similar domains such as the IoT, BACS, Avionics and Self-Driving systems.

Figure 4.6 and Figure 4.7 represent, respectively, the deployment and architecture of the proposed SSU. There is some analogy between this equipment used in SCADA systems and the one developed within the scope of this work since, in both situations, the Fieldbus and the local network are monitored to infer useful information to address security and safety issues.

This work enabled some valuable insights that were substantially deepened in this thesis and that contributed to the solution obtained and detailed in the

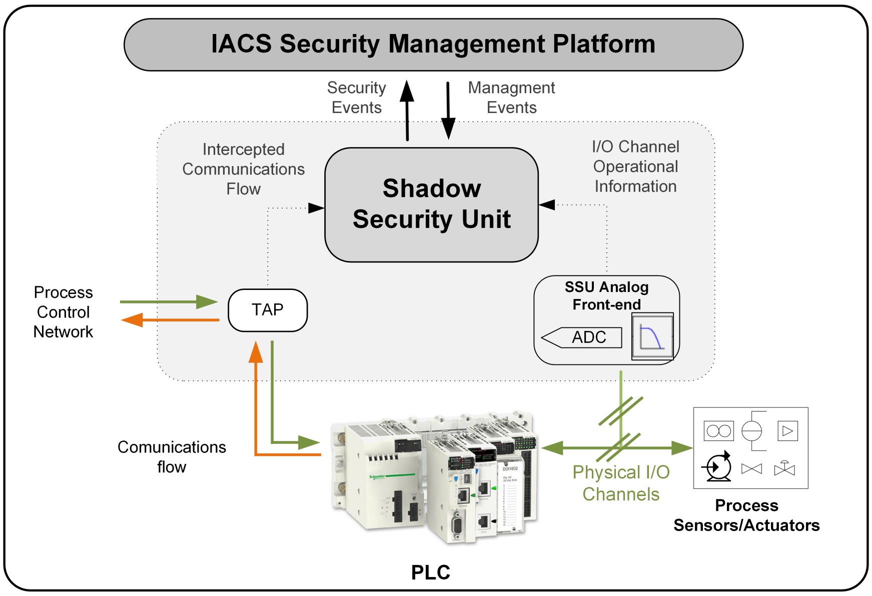


Figure 4.6: Deployment of the shadow security unit

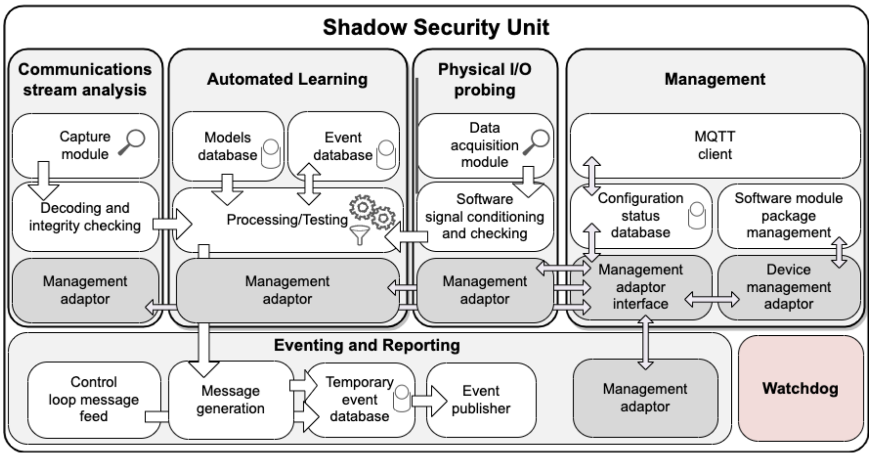


Figure 4.7: Shadow security unit architecture

previous sections.

## 4.5 Summary

The existing pressure on security in BACS inspired the focus of this thesis on finding tools to help withstand the threats, namely in monitoring the field level. This work proposed a NIDS concept which can monitor and analyse Fieldbus traffic, designed explicitly for BACS environments considering its typical scenarios in terms of physical deployment, costs and management interfaces.

After introducing the proposed concept, requirements and architecture on Chapter 3, this Chapter 4 describes a specific PoC implementation for KNX TP Fieldbus, as an example. The software components, hardware details and generic detection and management capabilities were presented. A laboratory testbed was created and used to develop and test all the hardware and software tools required for the system. In Chapter 5 we demonstrate and evaluate these capabilities, in the scope of a real BACS scenario and the developed testbed.

# Chapter 5

# Evaluation of the BACS NIDS

## Contents

The evaluation of the proposed framework is detailed in this chapter that is organized as follows: Section 5.2 describes the two types of scenarios used to build the attacks; Section 5.3 describes a bundle of datasets that were created for the validation (Graveto et al. [2022b,c]); Section 5.4 presents the techniques and methodologies that were used in the validation process (which represent, nevertheless, a small subset of the available possibilities enabled by the proposed tool); and Section 5.5 presents the obtained results; and Section 5.6 provides a final discussion of the validation process.

## 5.1 Objectives

The validation process hereby described addresses two key objectives:

- To demonstrate the effectiveness of the proposed framework in detecting anomalies and attacks on a BACS;
- To highlight the versatility of the BACS NIDS and the open possibilities for future development. Either by integrating and applying different algorithms or through the tuning and combination of different approaches.

The proposed BACS NIDS aims to detect and alert whenever a security/safety risk or malfunction occurs. No way of blocking potential attacks is available, as humans in the loop are considered necessary in control systems for safety reasons: the automatic reaction in a control system could be exploited as a target for attacks, further enhancing the adverse effects on the system.

The total absence of BACS datasets concerning the regular operation of these systems or even more regarding possible attacks or malfunctions identified the prior need for their creation. Thus, the creation of a datasets bundle was established as a secondary objective before the validation process (see Section 5.3.2).

## 5.2 Reference Security Use Case Scenarios

A generic BACS is comprised of devices interconnected by a communication network. Thus, threats to such a system may result from attacks on these devices and/or their interconnection network. In this work, two scenarios were created for study and research purposes: Scenario 1, in which a compromised device is simulated, and Scenario 2, which illustrates unauthorised network access. Combined, these two scenarios enable validating a large group of possible attack situations against a broad spectrum of BACS.

### 5.2.1 Scenario 1 – Compromised Device

The first approach for validating the tool is to assume that somehow the attacker managed to access at least one of the BACS devices and compromised its behavior. The establishment of such a bridgehead is typically one of the first steps in the cyber kill chain.

The attacker's objective is to understand the building's control network and, later, to collect detailed information about the various devices and their programming. Executing the necessary commands from a valid device will allow these tasks to be somehow hidden and carried out with little interference in the normal system operation.

The KNX *Line Scan* operation consists in the repeated use of an attempt to establish a one-to-one communication link with all possible destination addresses of a given line. This procedure allows identifying valid and existing IAs in the system.

After obtaining those valid IA's, the simple use of the *Device info* function of KNX allows querying these devices, obtaining information such as manufacturer, data, configuration parameters. This extraction of information is possible due to the KNX specifications, since all devices must report their data and characteristics whenever queried.

These commands are not typical during normal BACS, since they are used mostly for debugging and system development. Therefore, its detection outside of this scope should trigger an alert from a security point of view.

### 5.2.2 Scenario 2 – Network Access

The second approach used in validation is to assume the attacker somehow got access to the BACS field network, being able to read and inject KNX messages.

On a KNX system, malformed messages are ignored by devices, therefore our tests use valid messages with malicious purposes – even though the tool could also be used to identify or remove invalid KNX messages, as mentioned in Section 5.4.2.

In this scenario several attacks can be performed, such as:

- creating malicious messages from an unknown sender, with the purpose of acting over devices;
- injecting a large number of messages in a short period of time, with the aim of causing a denial of service;
- simulating human action by sending messages that are valid but that are not appropriate for the context (such as the faked activation of a physical switch in rooms where presence detectors report no human presence).

All these cases were used as scenarios creating the datasets presented in next.

## 5.3 Generation of a Dataset Bundle

Due to the lack of publicly available datasets or data libraries containing BACS system traces, a decision was made to create a dataset bundle to support the analysis and research efforts undertaken in the scope of this thesis, also to be made available to the research community. As such, the publication of the generated dataset constitutes an effort towards complying with research transparency guidelines, also contributing towards the development, study and validation of detection and protection tools for BACS.

This section describes the dataset bundle, including a diverse variety of attacks, as well as traces obtained during normal operation, for the reference use cases presented in Section 5.2. The acquisition process, which was undertaken in a real house with a full-scale BACS, is equally described, encompassing an explanation of how this data can be employed for the security analysis of KNX BACS. The attack categories that are covered include four types: line scan, device info, message injection and invalid context actions. These represent a realistic diversity of possible attack types used to develop, fine-tune and validate the proposed framework. The data has been maid available online (Graveto et al. [2022b,c]).

The purpose of this dataset bundle goes far beyond this PhD thesis, as it contributes to the research and development of solutions aimed at increasing security and safety in BACS. In addition to security issues, the original dataset can be used to study the regular operation and characterization/optimization of BACS systems of this type. Algorithms developed using KNX systems will be extensible and reusable in other building automation systems.

### 5.3.1 Collection Process

The data gathering was carried out in a single-family house with three floors. The building has devices for monitoring the environmental conditions outside and movement and lighting control in the house's backyard and its annexes. This house is equipped with a KNX home automation system that controls the lighting, blinds, and heating system that uses radiators and an alarm system. Some local environments are also controlled with light and motion detectors.

#### 5.3.1.1 Description of the Scenario

The validation of the proposed architecture was carried out using a single family house as reference scenario. This house has a global home automation system, based on KNX technology, and includes three floors, four bedrooms, kitchen, common living and dining room, office, laundry, TV and games room, as well as a machine room.

The developed KNX NIDS was deployed for validation using that house as a testbed. The BCU was connected to the KNX bus, and the *Communication Stream Analysis* module was connected to the house LAN.

The BACS consists of around eighty control devices, automating the operation of lighting systems, shutters, central heating, security, interior and exterior monitoring, etc. The system is based on KNX and has devices such as actuators, switches, motion and luminance sensors, temperature sensors, roller shutter actuators, solenoid controllers, power supplies, a touch panel for information and action, a weather station, and security and alarm devices.

The KNX network uses a twisted pair line 1.1.X for the connection of all devices. A wide range of three-tier group addresses is used for the automation system operation. The main group distinguishes the various floors, the middle group sectorizes the various services (lighting, scenarios, blinds, air conditioning, alarm...) and the address specifies each of the spaces or zones (for instance, the group address 2/0/8 corresponds to the second floor – 2, light on/off – 0, and the kitchen room – 8).

#### 5.3.1.2 Overall Architecture

The KNX topology can be made up of areas that group together several lines (cf. Figure 2.6). A Backbone line is used to connect all the existing areas. However, the installation in the house used for this data collection consists of a single area with just two lines, in which all devices are connected in a star topology.

Data collection was performed using the previously described BCU, connected to the field bus (KNX TP bus), sampling all the messages that flow through the bus. As KNX frames have no time stamp, these messages are archived and enriched with a local time stamp from the device's operating system.

#### 5.3.1.3 Used Formats

The collected data was stored in PCAP format (The Tcpdump Group [2020]). Each packet contains the raw message as a byte array and the timestamp representing the instant that message was received from the bus.

Then, a Comma Separated Values (CSV) file is created during the treatment and enrichment process. The collected dataset is augmented using the exported project file of the house (XML file as specified by KNX Association [2004]), obtained from ETS (KNX Association [2022]). The added information is represented on fields ETS Function, Location, Location Type, and Location Name. The CSV file contains the following fields:

- Pkt – the packet number;
- TimeStamp – the time-stamp representing the instant the packet was received;
- InterMessageTime – the time between two consecutive messages;
- NumMessagesSec – the number of messages per second;

- AvgMessagesSec – the moving average of messages per second;
- MessageSize – the message size;
- MessageChecksum – the KNX message checksum;
- SA_text – the Source Address (SA) as text;
- SA_int – the SA value;
- DA_text – the Destination Address (DA) as text, using dots in IA or slashes in GA;
- DA_int – the DA value;
- KNX_Info – the KNX operation in text;
- KNX_info_int – the KNX operation value;
- KNX_Code – the KNX code;
- KNX_Value – the value send as the operation argument;
- Msg – the byte array of the KNX message;
- EtsFunction – the known function;
- Location – the device location;
- LocationType – the location type;
- LocationName – the location name;
- Target – the classification field that is *zero* for messages of the normal operation and *one* for non-normal messages (either attack and the respective response from the system).

#### 5.3.1.4 Sampling Interval

The BACS system was monitored non-stop during thirty-seven days, to create the initial dataset. The sampling took place uninterruptedly between 3:26 pm 7/March/2020 and 5:00 pm 13/April/2020, leading to the storage of 381 337 packets which, after processing and eliminating some invalid data, allowed the creation of a sample with 379 875 packets.

### 5.3.2 Generated Datasets

The dataset bundle contains five distinct datasets: the first corresponds to regular house operation, and the remaining four to attack situations that are injected into that original dataset. To obtain each of the attack scenarios, the attack took place in the actual installation, having collected the messages corresponding to the attack and the existing system responses to it. Thus, it is guaranteed that reactions and behaviours provoked in the system by that same attack are also obtained.

The set of messages was collected in PCAP files. Then the datasets were built with their injection in the original dataset, which implied some manipulation of time stamps to obtain the desired datasets. The datasets are unbalanced as the attacks represent a small number of messages compared with the total number in the dataset.

The temporal windows used for the attacks are different, thus allowing their overlapping, which could be helpful and enable the creation of multiclass classifiers.

#### 5.3.2.1 Normal Operation Dataset

This dataset corresponds to the regular use of the home automation system of a single-family house inhabited by a couple with two older children for slightly longer than two months. Its use will help calibrate situations and operate models devoid of intrusion or attack.

The following datasets use, as a base, this same dataset, in which the poison files are merged in the specified conditions.

#### 5.3.2.2 Line Scan

The *line scan attack* consists of a sequence of messages at the KNX transport layer of type *TConnect*, that is a connection request for one-to-one communication. In this attack the requests are sent in sequence to all possible individual addresses (IA). The used sender address (SA) was a valid address IA 1.1.102 (a switch located in one of the rooms, which was previously hacked for the purpose of this attack). The characteristics were the following ones:

- attack started at 2:30:00 am of 20/March/2020
- attack ended at 2:30:32 am of the same day
- duration of 32 seconds
- 1 649 poison messages

Whenever a valid device exists on the destination IA, it replies with a *TDisconnect*. Then the the attacker sends a new *TConnect* and a *DeviceDescription* request, followed by *TACK* and *DeviceDescription* messages from the destination. Finally the attacker sends a *TACK* and *TDisconnect* messages ending that conversation.

#### 5.3.2.3 Device Info

The *device Info* attack consists of an information request, in a sequence, to two known devices IA 1.1.142 (a light and motion sensor on the leaving room of basement floor) and IA 1.1.136 (the leaving room switch) in a sequence. Again the send was a valid buit previously compromised switch with IA 1.1.102. This same attack was injected three times in the original dataset, resulting in a total of 8’820 poison messages, with the following characteristics:

- Situation 1 – Attack started at 1:30:00 am of 16/March/2020 and was repeated 10 times in a row. The attack stopped at 1:39:07 am;
- Situation 2 – Attack started at 5:30:00 am of 27/March/2020 and was repeated 20 times in a row. The attack stopped at 5:48:13 am;
- Situation 3 – Attack started at 2:00:00 pm of 6/April/2020 and was repeated 15 times in a row. The attack stopped at 2:13:40 am;

Each *Device info* request is composed of the attacker sending *TConnect* and *DeviceDescription* messages. The device replies with *TACK* and *DeviceDescription* messages. The attacker sends *TACK* and *MemoryRead* messages which are replied with *TACK* and *MemoryResponse* messages. The attacker sends *TACK* and *ADCRead* replied with *TACK* and *ADCResponse*. The last two exchanges are repeated a few times to gather the device information and finally the attacker ends the conversation sending a last *TACK* and *TDisconnectin*.

#### 5.3.2.4 Message Injection Attacks

The *message injection* attacks consist of sending valid KNX messages with different rates and contents to disturb the BACS. Two variants were considered according to the message rate.

**Slow rate attack**

This *slow rate message injection attack* uses KNX messages with valid fields but with invalid SA or DA, injected at random intervals. Three datasets are provided with different poison densities: 1%, 5% or 10 % of the total bus messages during the attack period, respectively.

**High rate attack – DoS**

The *high rate message injection attack* consists of a Denial of Service attack where a large number os messages (15 000) is injected into the bus with a high-density rate (one thousand messages per second). A valid KNX message is injected with valid SA and DA with a command payload to open the dining room blinds (0xBC116E1403E100804A) from switch IA 1.1.110. A message of type GroupValue_Write is sent to GA 2/4/3 with value 0x00. The attack started at 9:55:10 pm on 17/March/2020 and lasted 15 seconds.

#### 5.3.2.5 Invalid Context Attack

Invalid context attacks explore the possibility of an apparently valid message, with apparently valid addresses, being issued out of context. For instance, pressing a switch will turn on the lights in the bedroom. However, it will not be possible for a human to physically access the switch without being flagged by the bedroom presence detector. Thus, a light activation on behalf of the switch without presence detection will likely be a message improperly inserted in the system and out of context. In real-world scenarios, a similar approach can be used to create more complex context rules.

The motion detector (with IA 1.1.120) emits a message of type *GroupValueWrite* to turn on the room light, whenever movement is detected in room. When pressed, the light switch emits also a *GroupValueWrite* message with a value of 0 or 1 (requesting a suite bedroom light to turn off or on).

The analysis of the original network trace with a relatively short time window allows, in normal situations, to verify that whenever there is a pressure on the switch, there is also a motion detection. Thus, the existence of switch messages not associated with motion detection messages is abnormal and interpreted as a potential sign of attacks.

The present dataset was built by injecting messages from the switch at random intervals without any corresponding detection messages. Three datasets were created with different densities of poisonous messages with 1%, 5% and 10% of the total messages in the dataset.

### 5.3.3 Structure and Files

The dataset bundle is available as a zipped file in which there are five folders:

- Original – The dateset that represents the normal opetation of the BACS, without any attack.
- LS – The files of the Line Scan Attack.
- DI – The files corresponding to the device info attack.
- MI – The message injection attacks with two folders.
  - SR – The slow rate attack.
  - HR – The high rate attack.
- IC – the invalid context attack.

In all situations, the raw capture files are provided, in PCAP format, along with the files in CSV format with the data augmentation, which in addition to the enrichment information also contains a classification field name *target*. There is a one-to-one relationship between packets in PCAP format and CSV format.

### 5.3.4 Other Scenarios

The validation of the proposed architecture was also improved by monitoring and creating data sets in two other types of spaces: a hospital and an office building. In both situations, the BACS was based on KNX technology. In the first case, it was used fundamentally for lighting control. In the second, in addition to lighting control, there was facade shading control and presence and/or movement detection.

The data and results obtained also contributed to the validation of the proposed framework. However, they have been omitted in this document due to privacy reasons and even security of the public or semi-public places used.

## 5.4 Techniques and Methodologies

The architecture proposed in Chapter 3 and materialized in Chapter 4 has modules that empower the detection or signalling of suspected security problems or abnormal functioning in three approaches. The module called *Management* includes the *Local Web Interface* that provides data visualization tools that allow the operator to analyze network data. The *Intrusion Detection System* and *Communication Stream Analysis* modules enable detection with the use of rules and heuristics. In contrast, the *Automated Learning* module allows the use of artificial intelligence techniques in a generic way with the same purpose.

In this section, the techniques and methods used in each approach are presented, which led to the global validation of the proposed architecture and made it possible to demonstrate its usefulness in achieving the objectives of this work.

### 5.4.1 Data Visualization

Data visualization enables users to process data graphically, allowing the identification of possible existing patterns and the real-time identification of abnormal patterns. A simple visualization tool was developed for the proposed architecture, which is able to depict a KNX network in a graph form (see Figure 5.1), where nodes represent devices and edges correspond to communication flows.

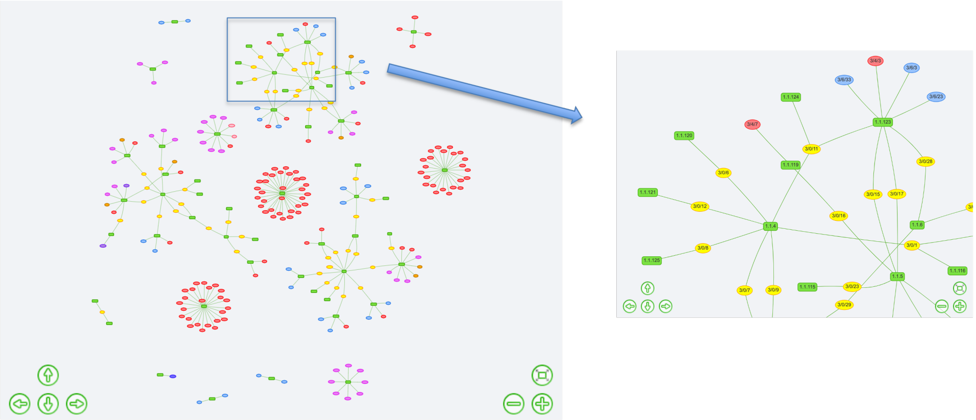


Figure 5.1: NIDS Field Network Graph – message flow

The information leveraged to build this representation is obtained through KNX field network monitoring, allowing the identification of both sender and destination addresses (the nodes) and the message flow (the edges). The relationship between several devices can also be visualized in this *black box* approach. The IA and GA are displayed inside the nodes contributing to the design of new rules for the system as specified in Section 5.4.2.

In a *white box* approach, the presented figure incorporates the known information of the housing project, using the XML file from the ETS by KNX Associ-

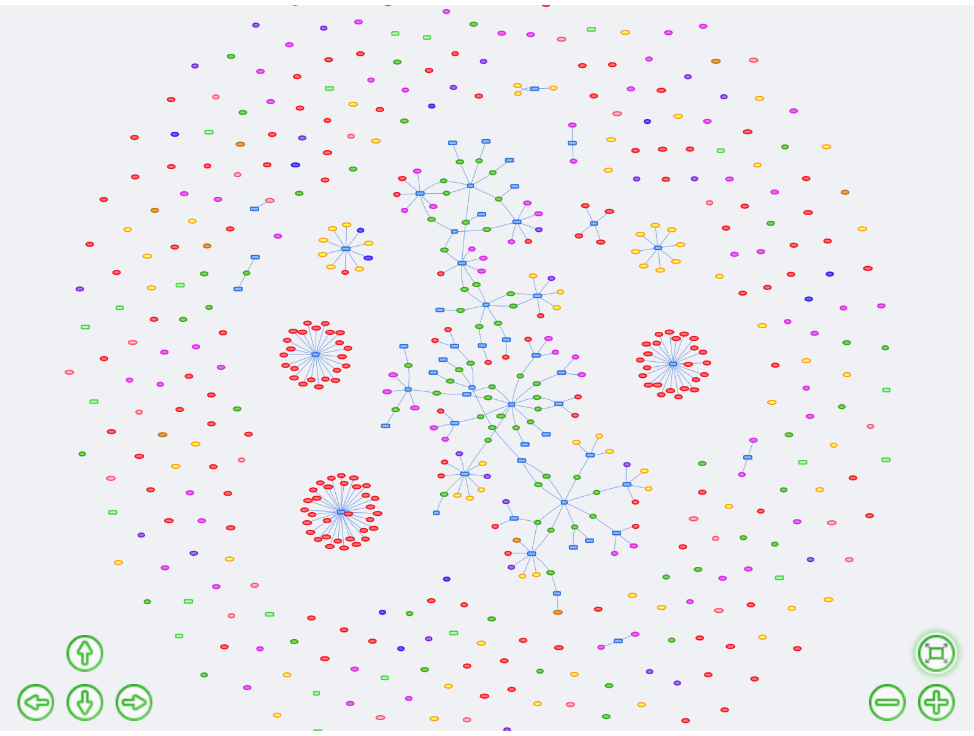


Figure 5.2: NIDS Field Network Graph – message flow and existing devices

ation [2022], which allowed the colouring of nodes, thus identifying the system functions to which they relate:

- Green – lighting on/off
- Orange – light dimming
- Violet – blinds short press
- Red – blinds long press
- Lavender – air conditioning / temperature control
- Pink – alarm
- Yellow – other functions
- Blue – the devices

Also, using the current knowledge of the project, all existing devices (see Figure 5.2) can be added to the representation, as well as all valid message paths (dashed lines), as shown on Figure 5.3.

The device located in the center, with IA 1.1.100, is a touch panel that allows home users to carry out most of the available functions, hence the great convergence of arcs towards this node. However, most are represented with dashed lines, because in the capture performed for the dataset construction, few interactions with this panel were recorded.

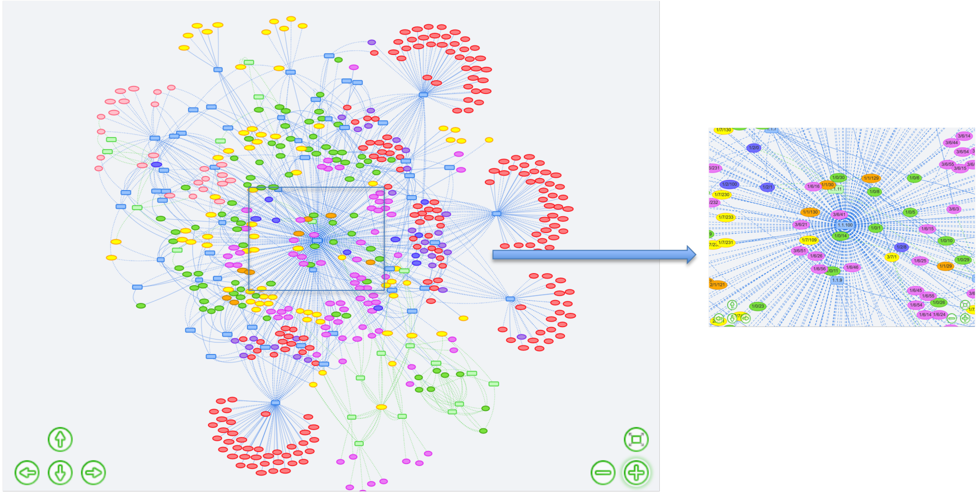


Figure 5.3: NIDS Field Network Graph – all existing devices and valid messages

### 5.4.2 Rules and Heuristics

As described in Section 4.3.2 the IDS module provides the anomaly and/or attack detection supported by patterns or fingerprints stored as rules on the existing local database. To the best of my knowledge, there is no previous work in defining rule-based detection for KNX. For this reason, we created a syntax which resembles (with specific adaptations to KNX) the well-known SNORT Chris Green; Martin Roesch [2020], with the following structure:

```
action protocol SA SPort operator DA DPort ( options )
```

The valid values for each field are those provided on Table 5.1

Table 5.1: Field values of IDS Rules

| action | protocol | source address (SA) | source port (SPort) | operator | destination address (DA) | destination port (DPort) |
|---|---|---|---|---|---|---|
| `pass` | any | any | any | -> | any | any |
| `alert` | knxtp | IA | | <> | IA | |
| `log` | KNXTP | | | | GA | |

The operator is used to apply the rule in one or both directions. The keyword *any* specifies that all the messages (packets) satisfy the condition. The *pass* action ignores the message that meets the rule, *alert* action issues an *alert* when it identifies any message that satisfies the rule, while the *log* action exports that message to a PCAP file. The *protocol* filters the search to the specified *protocol*, on the implemented PoC (see section 4.1) represents KNX.

Table 5.2 presents the available options, which can be combined in the same rule (*content* can actually be used multiple times to build the intended pattern or fingerprint). All option values can use the *!* (not) operator, thus facilitating the construction of rules. In the following example, all contents that do not include the word *light* will trigger an *alert*:

```
alert any any any -> any any ( msg:"exclude light msg";  content:
   !"light"; rid:200; rev:1; )
```

Table 5.2: Available options on IDS Rules

| Option Keyword | Mandatory | Usage | Example |
|---|---|---|---|
| `msg` | No | simple text to identify the rule | msg: ”this is a rule”; |
| `tpci` | No | used to search a transport layer function | tpci: ”T_Connect-PDU”; |
| `apci` | No | used to search an aplication layer function | apci: ”A_Group-Value_Write_PDU”; |
| `content` | No | text or hexadecimal bytes (allowed multiple times on same rule) | content: ”light”; or content: ”lBC\|”; |
| `pcre` | No | use of regular expression to search the message | pcre: ”.+”; |
| `rid` | No | integer to identify the trigged rule | rid: 5; |
| `rev` | No | integer that represents the revision of the rule | rev:3; |

The rule engine goes through the rules sequentially for each packet until there is a match. After this trigger, the engine will: (i) jump to the following message in case of a `pass` action; (ii) write to the output file in the case of a `log`; and (iii) trigger an *alert* in case of an `alert` rule. These types of Rules can be combined for more complex cases.

This process allows a multitude of uses for the tool. For instance, to create a file containing only valid KNX messages (excluding all others), the following rule could be used:

```
log knxtp any any -> any any ( msg:"valid KNX msg";  pcre:".+";
 rid:300; rev:1; )
```

The DPI KNX functional block of the IDS is used to decode KNX messages whenever the protocol is specified or when the source or destination addresses are used on any rule. Moreover, this module also implements some feature extraction, and computing statistics, such as:

- Time between two consecutive messages.
- Number of messages per second.
- Moving average of messages per second.
- Message size.

The combination of several rules allows more complex cases to be equally detected by the system. And yet this definition of multiple rules enables the system at runtime to be equipped with the knowledge necessary to identify many attacks or anomalies whose patterns or fingerprints are known.

### 5.4.3 An Overview of Candidate Artificial Intelligence Techniques for BACS Anomaly Detection

Since rule-based detection can only handle previously known attacks, AI-based anomaly detection was also explored. Anomaly detection using AI usually resorts to classifiers, which can require resources outside devices’ limited hardware capabilities, such as the proposed BACS IDS. For this reason, in the proposed architecture, AI models are built and trained in external equipment (based on previously collected traffic captures) and then imported to the IDS to allow real-time traffic analysis and anomaly detection.

The use of AI is already widespread in many cyber-security application domains. However, a search for its application in the specific scope of BACS cyber-security revealed few examples. Patil et al. [2019] present a machine learning (ML) algorithm to distinguish between normal operation, malfunctions and attacks on a BACS. The scenario consists of a building with HVAC control, fire alarm, access control and lighting. Using various previously trained bi-linear classifiers, a bi-linear classifier is used to distinguish between the three situations. Chavis et al. [2020] developed the Connected Home Automation Security Monitor (CHASM). It uses a Multiclass Decision Forest classifier to identify present IoT devices. It intends to use Multiclass Neural Networks to leverage the time series nature of IoT data to characterize normal and abnormal behaviour of an IoT-based BACS. Finally, Ramapatruni et al. Ramapatruni et al. [2019] use Hidden Markov Models to detect operating anomalies in a smart home.

Looking at the somehow related domain of industrial automation and control systems (IACS), it is possible to find a broader range of works (Iturbe et al. [2017]; Ding et al. [2018]; Nazir et al. [2017]). For instance, Rosa et al. [2021] propose a flexible intrusion detection platform able to support multiple AI-based anomaly detection tools. Anton et al. [2019] used Support Vector Machines and Random Forests for intrusion detection. Phillips et al. [2020] used a SCADA system based on the Modbus protocol to evaluate the use of SVM, Decision Trees, K-Nearest Neighbors and K-Means Clustering techniques for anomaly detection. Keliris et al. [2016] also use SVM-based algorithms as a defence for process-aware attacks in IACS.

To assess the suitability of AI-based anomaly detection in the scope of the proposed BACS IDS, we explored three techniques: neural networks; support vector machines; and logistic regression.

#### 5.4.3.1 Neural Network Models

The Neural Network (NN) implementation uses a sklearn's multi-layer perceptron (Pedregosa et al. [2011]), which produces, through a supervised learning algorithm with backpropagation, to obtain a non-linear function $f(X) : R^m - > R^o$ that represents the data. The training process was accomplished using the datasets collected in our reference scenario. From a known set of features $X = x_1.x_2, ..., x_m$ and a specific objective, previously known as the dataset, is classified, and a non-linear function is obtained – the model. This model was saved and imported into the KNX NIDS, allowing regression and/or classification of data subsequently submitted to that device.

Figure 5.4 represents the used neural network, with three hidden layers (a, b, c) and, respectively, 100, 20 and 100 neurons. Each neuron in the hidden layers transforms the values received from the previous layer through a weighted

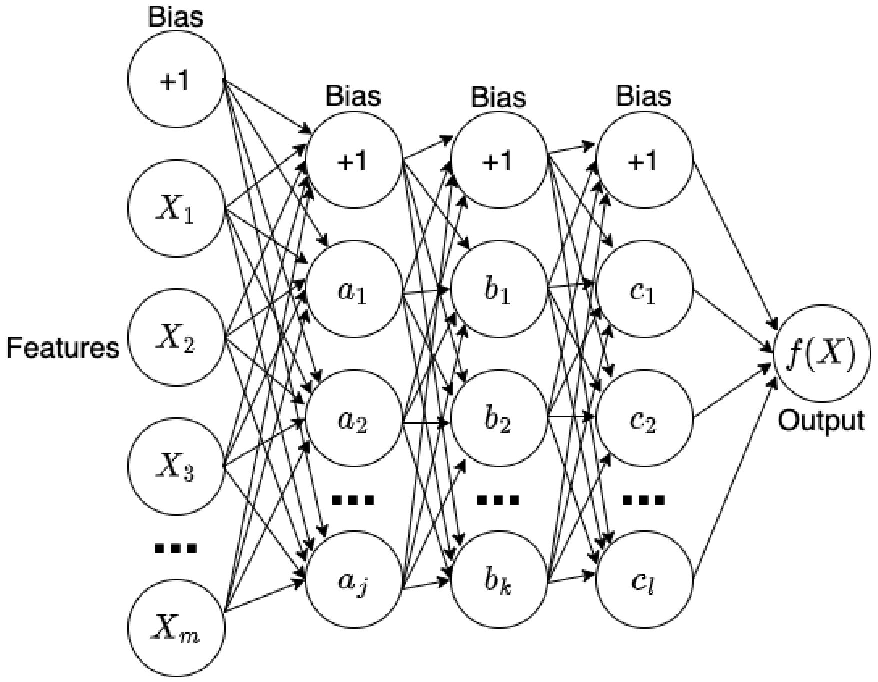


Figure 5.4: Multi-layer perceptron network

summation $(w_1x_1 + w_2x_2 + ... + w_mx_m)$, followed by an activation function. In the following validation examples, the activation function called *ReLu* was used. It implements a rectified linear unit function. The *Adam* solver, a stochastic gradient-based optimizer, was used for weight optimization. Finally, the output layer receives the values from the last hidden layer and computes them into the output values.

Fine-tuning this network is beyond the scope of the present work. Nevertheless, as this model type is quite sensitive to feature scaling, the data values were previously standardized to a mean of zero and a variance of one.

#### 5.4.3.2 Support Vector Machines

The Support Vector Machine (SVM) proposed by Cortes and Vapnik [1995] was adopted as the second AI approach in our IDS. For binary classification purposes, the created dataset entries were labelled as 1 (true) or 0 (false) to signal if they belonged to an attack or normal operation.

The input vectors are nonlinearly mapped into a high-order multidimensional space, where a linear decision surface is constructed to allow the separation of input data into two distinct groups – by maximizing the margin between instances of different classes.

The implementation used cost and regression loss epsilon with values of 1.00 and 0.10, respectively, where cost represents the penalty in terms of loss and epsilon is the distance between true values within which no penalty is associated with the predicted values. The kernel function $exp(-g|x-y|^2)$, named *RBF*, transforms the attribute space into a new feature space, where $g = 1/k$, with $k$

being the number of attributes.

Before creating the model, the following preprocessing tasks are sequentially undertaken:

- removal of instances with unknown target values;
- turning of categorical variables into continuous;
- removal of empty columns;
- imputation of missing values with the mean of surrounding values.

As with the neural network, the model was built in an external computer and later imported into the KNX NIDS. The IDS supports the coexistence of several models, allowing them to be used separately or in parallel in ensemble approaches.

#### 5.4.3.3 Logistic Regression

The Logistic Regression (LR)) consists of estimating parameters for a logistic model, which are the coefficients necessary for the linear combination of one or more variables. It uses the natural logarithm to create a continuous criterion that maps probability to log odds using a linear combination of one or more independent variables.

The implementation of logistic regression uses Ridge regression Hilt and Seegrist [1977] as the regularisation type. This method estimates the parameters in scenarios where linearly independent variables are highly correlated. Thus, using a dataset with $N$ points, each one represented by a set $m$ of variables $(x_1, x_2, ..., x_m)$, called independent variables and a dependent variable, the output $y$ (in our dataset named target), the logistic regression allows creating a predictive model for that output.

Again, the models were produced in an external computer and later imported to the KNX NIDS to enable real-time classification of received messages.

#### 5.4.3.4 Approach followed for he training and testing phases

The datasets were split using 70 per cent for training and 30 per cent for testing. The following features were considered:

- InterMessageTime.
- NumMessagesSec.
- AvgMessagesSec.
- MessageSize.
- MessageChecksum.
- SA_int.
- DA_int.

- KNX_info_int.
- KNX_Code.

First, the three methods (*neural network*, *support vector machine* and *logistic regression*) were explored, using ten-fold cross-validation, to identify which ones best suited each attack scenario. Next, the models were trained using the entire training subset. Finally, the resulting models were saved for later use.

Next, the produced models were loaded and tested against each attack scenario (using the testing subset).

Observed results are presented using confusion matrices. Moreover, we used the following metrics:

- area under the curve (AUC) – area under receiver operation characteristic (ROC) curve, which is create plotting the True Positive Rate (Recall) against the False Positive Rate (FPR)

$$\frac{FP}{FP + TN} \tag{5.1}$$

- accuracy (CA)

$$\frac{TP + TN}{TP + TN + FP + FN} \tag{5.2}$$

- F1

$$2 * \frac{Recall * Precision}{Recall + Precision} \tag{5.3}$$

- Precision

$$\frac{TP}{TP + FP} \tag{5.4}$$

- Recall

$$\frac{TP}{TP + FN} \tag{5.5}$$

## 5.5 Obtained Results

This section analyses the results obtained for each type of attack previously presented. For rule-based detection we present mostly the rules that have been defined and discuss achieved detection capabilities. Regarding AI techniques, we resort to the aforementioned confusion matrices and metrics.

The dataset bundle detailed on Section 5.3 was used for the following validation process. It was obtained on a private house, with a KNX BACS system monitored for thirty-seven consecutive days without interruption to create the initial normal dataset. This dataset was archived using the PCAP format.

The dataset was preprocessed to extract features such as:

- arrival time between messages;

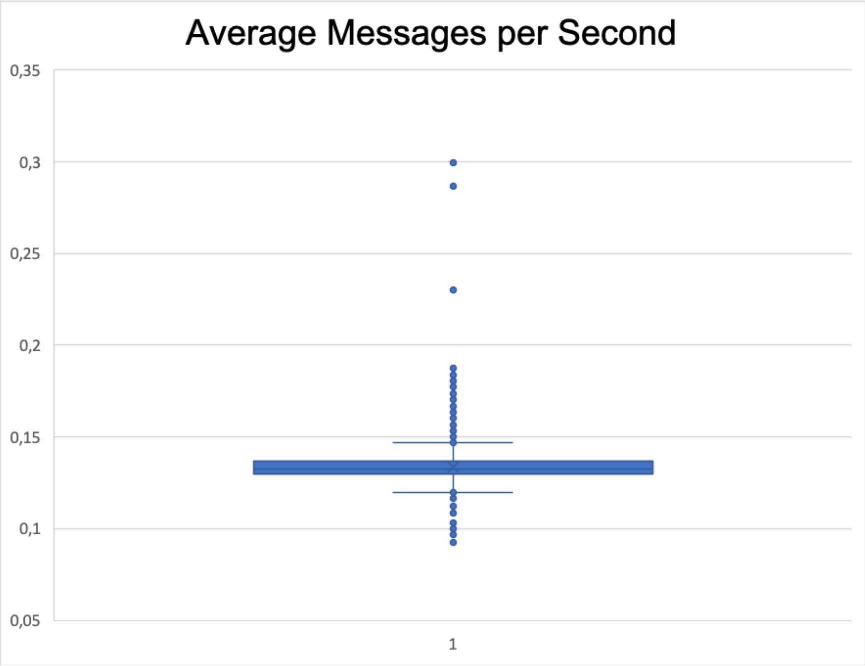


Figure 5.5: Dataset boxplot – outlier identification

- number of messages per second;
- cumulative average of messages per second;
- message size;
- and message checksum.

Obtained features were used to remove outliers (as shown in Figure 5.5), with the dataset also being characterised (cf. Table 5.3). The number of outliers removed from the dataset represented approximately 6.3 per cent of the total data.

Table 5.3: Dataset stats

| Dataset | Inter Message time | Average Messages per second | Message size |
|---|---|---|---|
| Average | 8,421 | 0.117 | 9,258 |
| Std | 7,556 | 0,005 | 0,661 |
| 1st Quartile | 1,331 | 0,134 | 9 |
| 3rd Quartile | 16,138 | 0,117 | 9 |

## 5.5.1 Message Injection Attack

In order to assess the IDS capabilities, we performed an attack consisting of sending a thousand valid messages per second during fifteen seconds, which corresponds to an abuse of the system.

The statistical features were extracted from the corresponding capture and compared to those of the initial data (without the attack). This attack was easily detected by both AI techniques and also by a basic rule defining a threshold for network traffic rates. Patient attackers could try to reduce the rate of the attack, to bypass detection, but at some point the attack would become useless since it would put no stress on the system.

### 5.5.2 Line Scan Attack

*Line Scan* operations provide the means for an attacker to undertake scouting operations in a KNX environment. However, this is an operation which is not commonly used in production environments, representing a sign of security issues. The compromised device (with IA 1.1.102) was used to send a sequence of messages of type *TConnect*, representing a request to establish a one-to-one connection. To detect this situation, a specific rule was added into the IDS signature testing module database:

```
  alert knxtp any any -> any any ( msg:
  "Transport Layer Connect - Line Scan detection";
  tpci:"T_Connect-PDU"; rid:20; rev:1; )
```

This rule was written using the format detailed at Section 5.4.2, searching for *knxtp* protocol messages from *any* source to *any* destination, and a Transport Layer Control Field (tpci) equal to *T_Connect-PDU*.

The rule-based mechanisms detected connection attempts to all possible Individual Addresses (DA 0 to 255), coming from 1.1.102, as shown in Listing 5.1.

```
2020/04/29 16:00:22 detection: Engine: [**] [20:1]
Alert - "Transport Layer Connect - Line Scan detection"
on packet 15403[**] - SA: 1.1.102 DA:0.0.0 - T_Connect-PDU

2020/04/29 16:00:22 detection: Engine: [**] [20:1]
Alert - "Transport Layer Connect - Line Scan detection"
on packet 15404[**] - SA: 1.1.102 DA:0.0.1 - T_Connect-PDU

2020/04/29 16:00:22 detection: Engine: [**] [20:1]
Alert - "Transport Layer Connect - Line Scan detection"
on packet 15405[**] - SA: 1.1.102 DA:0.0.2 - T_Connect-PDU

(...)

2020/04/29 16:00:22 detection: Engine: [**] [20:1]
Alert - "Transport Layer Connect - Line Scan detection"
on packet 15657[**] - SA: 1.1.102 DA:0.0.253 - T_Connect-PDU

2020/04/29 16:00:22 detection: Engine: [**] [20:1]
Alert - "Transport Layer Connect - Line Scan detection"
on packet 15658[**] - SA: 1.1.102 DA:0.0.254 - T_Connect-PDU

2020/04/29 16:00:22 detection: Engine: [**] [20:1]
Alert - "Transport Layer Connect - Line Scan detection"
```

```
on packet 15659[**] - SA: 1.1.102 DA:0.0.255 - T_Connect-PDU
```

Listing 5.1: KNX NIDS Line Scan output

The produced *Alert*, which uses a format similar to SNORT alerts, reports the occurrence of messages that are in accordance with the signature defined by the detection rule.

## 5.5.3 Device Info Attack

In the *device info* (DI) attack scenario (cf. 5.3.2.3), the compromised switch (IA 1.1.102) unduly questions a luminosity/presence detector (IA 1.1.142) and a switch (IA 1.1.136), to gather information about their settings and capabilities – using the KNX functions *Device Description* and *Memory Read*, besides the necessary handshakes to establish one-to-one communication with the two target devices.

In this attack, we evaluated both AI and rule-based detection, as discussed next.

### 5.5.3.1 Automated Learning Approach

This module uses the enhanced dataset, provided in CSV format when training the models and computed in runtime when using the model for the detection phase. The three methods described in Section 5.4.3 : neural network, support vector machine and logistic regression were used.

Regarding AI techniques, all methods achieved almost perfect scores, as expected (due to the nature of the attack). SVM had one false negative (cf. Table 5.4). In contrast, the neural network had one false positive (cf. Table 5.5). As for logistic regression, all the 2 595 fraudulent messages were detected, and no regular messages were wrongly marked as fraudulent (cf. Table 5.6).

Table 5.4: DI Attack – SVM Confusion Matrix

| | | Predicted | | |
|---|---|---|---|---|
| | | 0 | 1 | Total |
| Actual | 0 | 114013 | 0 | 114013 |
| | 1 | 1 | 2594 | 2595 |
| | Total | 114014 | 2594 | 116608 |

Table 5.5: DI Attack – NN Confusion Matrix

| | | Predicted | | |
|---|---|---|---|---|
| | | 0 | 1 | Total |
| Actual | 0 | 114012 | 1 | 114013 |
| | 1 | 0 | 2595 | 2595 |
| | Total | 114012 | 2596 | 116608 |

Table 5.6: DI Attack – LR Confusion Matrix

|  |  | Predicted |  |  |
|---|---|---|---|---|
|  |  | 0 | 1 | Total |
| Actual | 0 | 114013 | 0 | 114013 |
|  | 1 | 0 | 2595 | 2595 |
|  | Total | 114013 | 2595 | 116608 |

#### 5.5.3.2 Rules-based Approach

Regarding **rule-based detection**, it is known in advance that the attack will need to use the following specific sequence of messages:

```
A_DeviceDescriptor_Read_PDU
A_DeviceDescriptor_Response_PDU
A_Memory_Read_PDU
A_Memory_Response_PDU
A_ADC_Read_PDU
A_ADC_Response_PDU
```

Based on that, a set of rules was created for detecting this type of attacks, as depicted in Listing 5.2.

```
alert knxtp any any -> any any ( msg:"Application Layer - Device
  Info 1"; apci:"A_DeviceDescriptor_Read_PDU"; rid:31; rev:1; )

alert knxtp any any -> any any ( msg:"Application Layer - Device
  Info 2"; apci:"A_Memory_Read_PDU"; rid:32; rev:1; )

alert knxtp any any -> any any ( msg:"Application Layer - Device
  Info 3"; apci:"A_ADC_Read_PDU"; rid:33; rev:1; )

alert knxtp any any -> any any ( msg:"Application Layer - Device
  Info 4"; apci:"A_DeviceDescriptor_Response_PDU"; rid:31; rev:1;
  )

alert knxtp any any -> any any ( msg:"Application Layer - Device
  Info 5"; apci:"A_Memory_Response_PDU"; rid:32; rev:1; )

alert knxtp any any -> any any ( msg:"Application Layer - Device
  Info 6"; apci:"A_ADC_Response_PDU"; rid:33; rev:1; )
```

Listing 5.2: Device Info Attack – Detection Ruleset

When using this ruleset, attacks were successfully detected, as shown in Listing 5.3 (in the presented example, the attack started from the node with IA 1.1.102 and targeted both IA1.1.142 and 1.1.136).

```
----- Situation 1

----- Scouting of Device IA 1.1.142
2022/04/29 16:44:56 detection: Engine: [**] [31:1] Alert - "
    Application Layer - Device Info 1" on packet 83292 [**] - 0
    xB01166118E514300B5 - SA: 1.1.102 DA:1.1.142 - T_Data_Connected-
    PDU - A_DeviceDescriptor_Read_PDU
```

```
(...)
2022/04/29 16:44:56 detection: Engine: [**] [31:1] Alert - "
    Application Layer - Device Info 4" on packet 83297 [**] - 0
    xB0118E11665343400013E4 - SA: 1.1.142 DA:1.1.102 -
    T_Data_Connected-PDU - A_DeviceDescriptor_Response_PDU
2022/04/29 16:44:56 detection: Engine: [**] [32:1] Alert - "
    Application Layer - Device Info 2" on packet 83302 [**] - 0
    xB01166118E5346010104B6 - SA: 1.1.102 DA:1.1.142 -
    T_Data_Connected-PDU - A_Memory_Read_PDU
(...)
2022/04/29 16:44:56 detection: Engine: [**] [32:1] Alert - "
    Application Layer - Device Info 5" on packet 83307 [**] - 0
    xB0118E1166544641010402F3 - SA: 1.1.142 DA:1.1.102 -
    T_Data_Connected-PDU - A_Memory_Response_PDU
2022/04/29 16:44:56 detection: Engine: [**] [33:1] Alert - "
    Application Layer - Device Info 3" on packet 83312 [**] - 0
    xB01166118E5249810835 - SA: 1.1.102 DA:1.1.142 -
    T_Data_Connected-PDU - A_ADC_Read_PDU
(...)
2022/04/29 16:44:56 detection: Engine: [**] [33:1] Alert - "
    Application Layer - Device Info 6" on packet 83317 [**] - 0
    xB0118E11665449C10805A9DF - SA: 1.1.142 DA:1.1.102 -
    T_Data_Connected-PDU - A_ADC_Response_PDU
(...)
2022/04/29 16:44:56 detection: Engine: [**] [32:1] Alert - "
    Application Layer - Device Info 5" on packet 83377 [**] - 0
    xB0118E1166546241010900D8 - SA: 1.1.142 DA:1.1.102 -
    T_Data_Connected-PDU - A_Memory_Response_PDU

----- Scouting of Device IA 1.1.136

2022/04/29 16:44:56 detection: Engine: [**] [31:1] Alert - "
    Application Layer - Device Info 1" on packet 83395 [**] - 0
    xB011661188514300B3 - SA: 1.1.102 DA:1.1.136 - T_Data_Connected-
    PDU - A_DeviceDescriptor_Read_PDU
(...)
2022/04/29 16:44:56 detection: Engine: [**] [31:1] Alert - "
    Application Layer - Device Info 4" on packet 83400 [**] - 0
    xB0118811665343400013E2 - SA: 1.1.136 DA:1.1.102 -
    T_Data_Connected-PDU - A_DeviceDescriptor_Response_PDU
2022/04/29 16:44:56 detection: Engine: [**] [32:1] Alert - "
    Application Layer - Device Info 2" on packet 83405 [**] - 0
    xB0116611885346010104B0 - SA: 1.1.102 DA:1.1.136 -
    T_Data_Connected-PDU - A_Memory_Read_PDU
(...)
2022/04/29 16:44:56 detection: Engine: [**] [32:1] Alert - "
    Application Layer - Device Info 5" on packet 83410 [**] - 0
    xB011881166544641010402F5 - SA: 1.1.136 DA:1.1.102 -
    T_Data_Connected-PDU - A_Memory_Response_PDU
2022/04/29 16:44:56 detection: Engine: [**] [33:1] Alert - "
    Application Layer - Device Info 3" on packet 83415 [**] - 0
    xB01166118852498108333 - SA: 1.1.102 DA:1.1.136 -
    T_Data_Connected-PDU - A_ADC_Read_PDU
(...)
2022/04/29 16:44:56 detection: Engine: [**] [33:1] Alert - "
    Application Layer - Device Info 6" on packet 83420 [**] - 0
    xB0118811665449C10800A0D5 - SA: 1.1.136 DA:1.1.102 -
    T_Data_Connected-PDU - A_ADC_Response_PDU
(...)
2022/04/29 16:44:56 detection: Engine: [**] [32:1] Alert - "
    Application Layer - Device Info 5" on packet 83480 [**] - 0
    xB01188116654624101O904DA - SA: 1.1.136 DA:1.1.102 -
    T_Data_Connected-PDU - A_Memory_Response_PDU

----- Previous listing repeated below more 9 times

2022/04/29 16:44:56 detection: Engine: [**] [31:1] Alert - "
    Application Layer - Device Info 1" on packet 83493 [**] - 0
    xB01166118E514300B5 - SA: 1.1.102 DA:1.1.142 - T_Data_Connected-
    PDU - A_DeviceDescriptor_Read_PDU
(...)
2022/04/29 16:44:56 detection: Engine: [**] [32:1] Alert - "
    Application Layer - Device Info 5" on packet 85291 [**] - 0
    xB01188116654624101O904DA - SA: 1.1.136 DA:1.1.102 -
    T_Data_Connected-PDU - A_Memory_Response_PDU

----- Situation 2

2022/04/29 16:44:57 detection: Engine: [**] [31:1] Alert - "
    Application Layer - Device Info 1" on packet 195235 [**] - 0
    xB01166118E514300B5 - SA: 1.1.102 DA:1.1.142 - T_Data_Connected-
    PDU - A_DeviceDescriptor_Read_PDU
(...)
2022/04/29 16:44:58 detection: Engine: [**] [32:1] Alert - "
    Application Layer - Device Info 5" on packet 199261 [**] - 0
    xB01188116654624101O904DA - SA: 1.1.136 DA:1.1.102 -
    T_Data_Connected-PDU - A_Memory_Response_PDU

----- Previous souting messages repeated above more 20 times

----- Situation 3

2022/04/29 16:44:59 detection: Engine: [**] [31:1] Alert - "
    Application Layer - Device Info 1" on packet 303313 [**] - 0
    xB01166118E514300B5 - SA: 1.1.102 DA:1.1.142 - T_Data_Connected-
    PDU - A_DeviceDescriptor_Read_PDU
(...)
2022/04/29 16:44:59 detection: Engine: [**] [32:1] Alert - "
    Application Layer - Device Info 5" on packet 306371 [**] - 0
    xB01188116654624101O904DA - SA: 1.1.136 DA:1.1.102 -
    T_Data_Connected-PDU - A_Memory_Response_PDU

----- Previous souting messages repeated above more 15 times
```

Listing 5.3: KNX NIDS Device Info output

According to the output obtained, which can be found in the Listing 5.3, some repeated messages were subtracted, indicating the existing repetitions. However, we can verify the detection performed. Situation 1 (described in section 5.3.2) occurs between messages 83'292 and 85'291, an interval in which alternating contact with the sensor and switch is verified ten times. Situation 2 occurs between messages 195'235 and 199'261, in which the same action is repeated twenty times and between messages 303'313 and 306'371 in which requests to obtain information are repeated fifteen times, as described in the dataset creation process.

With the knowledge that normal system operation is based on application-level messages such as:

```
A_GroupValue_Write_PDU
```

Using the negation operator `!`, we could simply use a rule like:

```
alert knxtp any any -> any any ( msg:"Application Layer - Device
  Info"; apci:!"A_GroupValue_Write_PDU"; rid:34; rev:1; )
```

Methodology with which identical results were obtained. However, a reduced number of verified messages were flagged with false positives throughout the global dataset.

It should also be noted that in both situations, communications at the transport layer level are not captured in this way, although they are part of the set of messages involved in the attack.

The two approaches presented in Section 5.5.3 demonstrate that the proposed architecture allows the detection of the same type of attack or anomaly by several approaches that may be complementary or alternative.

### 5.5.4 Invalid Context Attack

The adopted *invalid context* (IC) attack consisted of a false message containing a valid source address from a real switch device being injected into the network. The attack is out of context as it is physically impossible for a human to use this switch without being signalled by the on-site presence detector. Implementing rule-based detection for this case can be easily accomplished by implementing temporal sliding windows to process multiple messages aggregated in narrow intervals and find infringing patterns. Nevertheless, this approach implies the overhead of having to manually identify patterns characteristic of normal operation for a specific deployment, later to be encoded as rules.

As an alternative, the use of AI-based techniques can potentially detect such attacks without requiring the definition, beforehand, off all possible invalid context situations.

Table 5.7 presents the obtained results for each technique. *Logistic Regression* and *neural networks* obtained excellent results, when compared with *SVM*, and therefore only those techniques were explored in more detail.

Table 5.8 shows that *Neural Network* had 50 false positives and no false negatives. *Logistic regression* (see Table 5.9) presents worse results, missing the identification of 86 of the total 1154 messages involved in the attack and, even worse, generating 192 false alerts.

Thus, in the universe of more than three hundred thousand messages, the results obtained with the Neural Network are very good. It should also be noted that

Table 5.7: IC Attack – Method Analysis

| Model | AUC | CA | F1 | Precision | Recall |
|---|---|---|---|---|---|
| SVM | 0.954 | 0.893 | 0.935 | 0.990 | 0.893 |
| Neural Network | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| Logistic Regression | 0.997 | 0.998 | 0.998 | 0.998 | 0.998 |

Table 5.8: IC – NN Confusion Matrix

| | | Predicted | | |
|---|---|---|---|---|
| | | 0 | 1 | Total |
| Actual | 0 | 113897 | 50 | 113947 |
| | 1 | 0 | 1154 | 1154 |
| | Total | 113897 | 1204 | 115101 |

Table 5.9: IC – LR Confusion Matrix

| | | Predicted | | |
|---|---|---|---|---|
| | | 0 | 1 | Total |
| Actual | 0 | 113755 | 192 | 113947 |
| | 1 | 86 | 1068 | 1154 |
| | Total | 113841 | 1260 | 115101 |

despite the architecture conceptually allowing the use of any method, their choice must be adapted to the type of attack that is intended to be recognized.

## 5.6 Discussion

This chapter filled the gap in the need for datasets from buildings with BACS. We also described the scenarios used for collecting information in a production environment. The dataset creation process included the injection of several attacks that enabled the validation process.

The validation process was supported by two methodologies: analysis by rules or heuristics, and AI-based solutions or artificial intelligence. Various AI techniques were described and used to demonstrate the framework's versatility. Also, the graphic representation capabilities are an added value, allowing an experienced user to elaborate on the rules and the approach to detect anomalies and/or threats to the BACS.

The obtained results validate the BACS NIDS and the underlying concept and architecture. The variety of attacks and scenarios used sustains this validation. And the redundancy of presented tests also supports the universality claimed for this framework in detecting anomalies and threats to BACS.

The PoC presented in Chapter 4, and the entire validation process presented, despite the exemplification with a KNX network, allowed to confirm that the proposed architecture can be applied to any BACS platform (BACnet, LonWorks, etc.). For that purpose, the BCU and DPI should be changed to the ones suitable for those system.

Finally, combining the two methodologies is possible, providing the framework with an even greater detection capability.

# Chapter 6

# Exfiltrating Data Using KNX

## Contents

When it comes to protecting confidential and/or sensitive information, organizations have a plethora of recommendations, standards, policies and security controls at their disposal, conceived to deal with a wide variety of threats. However, most of them share the same fundamental premise: that weaknesses are inline by nature, as a consequence of infrastructure, social and/or technological gaps that can be controlled, mitigated or constrained.

Side-channel threats are a different matter, though. Stemming from unconventional intrusion or attack vectors whose existence was inconceivable, deemed unfeasible, or even completely unexpected, their successful exploitation may provide attackers with the means to bypass and render most security controls ineffective or useless. This chapter addresses one such case: using a KNX-based building automation and control system to exfiltrate data from an air-gapped infrastructure. The basic premise for this scenario assumes that an attacker managed to introduce a rogue device in an existing KNX building Fieldbus, thus becoming able to send data through it or even control other devices with no interference in the operation of the BACS network. The feasibility of this approach was validated through an experimental setup, which was used to successfully evaluate two different techniques: inline bus exfiltration and optical transmission via dimmer control. Finally, some measures for detecting and mitigating these attacks are proposed.

## 6.1 Introduction

Isolation is often regarded as a sound strategy to increase security, preserve sensitive data and safeguard critical information. Thus, when an organisation or company needs to handle critical and sensitive information, it frequently resorts to physically isolated rooms and computers without any external network connection, as it is the case for Sensitive Compartmented Information Facilities (Office of the Director of National Intelligence, USA [2010]) or Special Access Program Facilities (National Counterintelligence and Security Center [2021]). Whenever confidential and highly sensitive information is present, the security of these air-gaped systems (i.e., systems without any internet or even local network connection) can be increased using techniques such as electromagnetic radiation isolation (through Faraday cages), frequency jammers (to prevent any wireless communications with the exterior) or acoustic insulation.

Quite often, organizations with those special security and confidentiality requirements are physically hosted in buildings equipped with automation systems, often called Building Automation and Control Systems (BACS). Historically, these systems base their security on infrastructure isolation and on the use of specific protocols, leading to several vulnerabilities, weaknesses and security gaps – as discussed in Chapter 2 and by Antonini et al. [2014]. Due to their vital role in managing the operational needs, from lightning to climate control, BACS are capillary by nature, spreading over the entire building in an almost pervasive fashion.

This chapter demonstrates to which extent BACS can be leveraged to exfiltrate sensitive data outside secure/isolated spaces designed to protect confidential information or systems. This is achieved by using the BACS fieldbus network as a covert channel for the attacker to transmit information. Two ways of exfiltrating data were developed and will be presented: the sending or streaming of data with potentially sensitive information (cf. Section 6.3.1); and the usage of the lighting system for transmission of data in real time (cf. Section 6.3.2). These proof-of-concept attacks also challenge the perspective that Operation Technology (OT) network security, whether industrial or for building control, should be based on the implicit premise that the main threats often arise from Information Network interconnections and not within the BACS itself.

The rest of this chapter is organized as follows. First, a review previous works on exfiltration in general is presented (Section 6.2). Next, the proposed BACS exfiltration techniques are described (Section 6.3), along with a PoC implementation (Section 6.4). Section 6.5 addresses validation of the proposed techniques, and possible countermeasures are discussed in Section 6.6. Finally, Section 6.7 summarizes the chapter.

Also recap Section 2.1.2 from Chapter 2, where an overview of the KNX framework for building automation and control systems is presented. The technical details of the proposed approach can be grasped within that section.

## 6.2 A Review of Side Channel Exfiltration Techniques

The need to safeguard access to data and protected and/or confidential information has always existed and will most likely persist in the future. To cope with this challenge, the academic and research communities have been long working on the identification of potential security gaps, in order to develop prevention, mitigation and defense mechanisms against possible exploits.

One such case are exfiltration attempts, which allow a malicious actor to stealthily extract sensitive information from specific systems or places. In this context, the term *Bridgeware* was originally coined by Guri and Elovici [2018] to designate malware that is designed to bridge the air-gap between an isolated environment and an attacker. After breaching the isolated infrastructure, bridging is vital to make it possible to exfiltrate data, often by resorting to covert channels, which may also be useful to bypass existing protection mechanisms in non-contained environments. Covert channels identified and implemented in the past encompass several mediums, such as electromagnetic radiation, acoustic waves, thermal radiation or optical transmission (e.g. LED, visible light, Infrared), among others. This section presents a brief chronological review of several works that have been developed in this area.

In 2002, Loughry and Umphress [2002] reviewed existing knowledge and studied the potential for information leakage through optical LED emanation. This work complemented the scarce information available at the time about the topic, mostly available from R. Anderson's book (Anderson [2020]) (whose first edition dates from 2001) and seminal paper (Anderson and Kuhn [1999]) – the lack of publicly available information at the time was also due to the classification of all related information as sensitive by the US National Security Agency.

Loughry and al. concluded that it is possible to read information from modulated optical signals that carry data injected by an eavesdropper into LEDs of various devices, such as modems, storage, routers and other miscellaneous devices. They also demonstrated that it was possible to reconstruct error-free data at speeds of up to 56 kbit/s, being theoretically possible to go up to speeds of about 10 Mbit/s using the same fundamental principles. Moreover, a taxonomy for classifying optical emanations according to their risk level was proposed and used in tests performed on various market equipment, to assess their potential risk, with some possible countermeasures also being analysed.

Hanspach and Goetz [2013] developed a covert acoustic mesh using commonly available speakers and microphones operating in the ultrasonic frequency range. This mesh was based on an underwater communications stack, whose central frequency was changed from 4.2 KHz to 21 KHz to guarantee its operation as a covert and stealthy communication channel. The use of multiple hops allows information to be sent over considerable distances – tests using conventional laptops (Lenovo T400) were successful with distances of up to 25 meters. The authors also proposed some countermeasures, such as using an audio filter guard connected to the audio input and output devices, implementing a trusted bandpass or lowpass filter. This approach was also studied later by Carrara and Adams [2014], extending it to several commercial equipment and analysing the relationship between distances and data transmission speeds, both in open spaces and closed environments, also resorting to error correction algorithms. Some tests were also carried out at audible frequencies, thus allowing the increase of transfer rates (which become available for overnight attacks) by widening the spectrum of usable frequencies.

Matyunin et al. [2016] demonstrated the use of electromagnetic emissions emanating from a laptop, together with the ability for a mobile phone to sense magnetic fields, to create a covert channel for data exfiltration. Using this technique did not require any specific hardware. Writes between CPU and RAM, and from it to storage, have been found to produce electromagnetic fluctuations which can be modulated to convey information. The distance between devices has to be small (up to 4 cm) in order to allow for their successful use with a rate of up to 2 bits/s. Proposed countermeasures include shielding electronic laptop components, performing I/O operations randomly to mask electromagnetic emissions and, on the mobile phone side, to make it mandatory for applications to request permissions to access magnetic sensors (which has meanwhile been implemented in most recent mobile operating systems).

Robles-Durazno et al. [2019] used two lamps connected to a Programmable Logic Controller (PLC) for data exfiltration from an Industrial Automation and Control System. The inability for humans to detect flickering in the order of

60Hz was taken into account, in order to avoid detection and make it a stealthy process. Transmitted information was received by a camera placed with line of sight to the transmitter, and then decoded.

Naz and Zeki [2020] performed a review of various attack methods on air-gapped systems, summarising used methods and devices, the attack effectiveness, and potential countermeasures.

In the last years the Ben-Gurion Univerity of the Negev has been producing a significant set of contributions in this field, developing several exfiltration techniques, such as those summarised below:

- MOSQUITO (Guri et al. [2019b]) inverts the operation of the computer speakers, turning them into input devices. This technique is used to exchange ultrasonic waves with the purpose of creating a covert channel between two air-gapped computers, setting up a speaker-to-speaker link. As receivers, headphones, earphones and earbuds were also tested, obtaining a transmission capacity between 300 and 600 bps at distances between 1 and 8 meters. Even though some spaces do not allow audio recording as a security measure, this proved to be an insufficient restriction for the proposed technique, which leads to the need of alternative countermeasures, such as ultrasonic jammers. Overcoming countermeasures such as removing speakers, by resorting to computer fans and other sound sources, was also researched by Guri et al. [2016b].

- CD-LEAK (Guri [2020]) proposes the usage of CD/DVD drives for the emission of acoustic signals in air-gapped computers without any audio equipment, for data exfiltration purposes. The analysis of acoustic signals emitted by the three mechanical components (drive motor, tray loading mechanism, the motor used to open/close the tray and drive motor) existing in an optical device (CD, DVD or Blu-ray) allowed the creation of algorithms for the controlled emission of these acoustic signals. As a receiver, another computer or mobile device with a network connection can be used, allowing a second hop in the exfiltration of information to the outside.

- AirHopper also (Guri et al. [2014]) allows to exfiltrate data from air-gaped computers by leveraging the FM radio reception capability built into many mobile phones, without the owners' knowledge. Assuming that the attacker was able to previously deploy appropriate malware on the target computer, it becomes possible to emit radio signals by modulating the electromagnetic radiation generated by the video display adapter. These signals are then picked by a compromised employee's mobile phone (whose use is normally allowed) once it comes close to the compromised computer. A similar approach, based on radiation from USB cables, is proposed in Guri et al. [2016a].

- BeatCoin (Guri [2018a]) details how private keys could be leaked from an air-gapped computer where a cryptocurrency wallet is managed in offline mode. Several known techniques were explored, such as physical (removable media); electromagnetic; electric; magnetic; optical; and acoustic.

- ODINI (Guri et al. [2020]) allows the exfiltration of data from a computer that is both air-gapped and placed in a Faraday-cage, exploiting the electromagnetic fields generated by the computer's CPU. Increasing the number of calculations to be performed by the CPU consumes more current and, consequently, generates a stronger magnetic field. The workload of each core is managed independently, which allows greater control over the generated magnetic field. Also, magnetic field control using virtual machines was explored with good results. Moreover, the Faraday-cage can be "by-passed" due to the fact that low frequency magnetic fields penetrate metals.

- BitWhisper (Guri et al. [2015]) makes it possible to establish a communication channel between two adjacent computers without any physical or radio frequency connection, using a covert channel that exploits the thermal radiation emitted by the devices. On infected computers, the BitWhisper malware uses the heat emanating from the CPU and other components (like the GPU, I/O controllers or mechanical systems such as hard disks or optical drives), as well as the multiple temperature sensors that normally exist, to emit and receive heat, modulating commands over it to create a new covert channel. The handshaking process using thermal pings was also implemented in the developed prototype. Several tests were conducted, using a mix of distances, times and temperatures. A similar cover channel implementation, using HVAC equipment, is proposed by Mirsky et al. [2017].

- Nassi et al. [2017] showcased how to create a covert channel between a C&C server and a malware installed in a computer by resorting to a document scanner and using it as a means of interaction.

- The possibility of creating a covert channel using the activity LEDs of a compromised network-connected equipment is discussed by Guri [2018b]. A compromised host on the network generates a blinking LED pattern by controlling the traffic flow on the corresponding LAN port, encoding the required binary data; if root/admin privileges are obtained on the compromised device, the host colour of the status LED can also be controlled, by manipulating the link speed. This technique does not require firmware changes on involved network devices.

- BRIGHTNESS (Guri et al. [2019a]) explores the use of a slight variation in the brightness of a display as a way of encoding data for exfiltration purposes. An adjustment of 3% in the red color component was used to encode a binary sequence on a 19" screen at a rate of 5 bit/s, making it possible to capture it at a distance of 6 meters from that screen. The evaluation process proved that 5 to 10 bps data rates can be obtained at distances of 1 to 9 meters, depending on the capture device that is used.

- CTRL-ALT-LED (Guri et al. [2019c]) uses the keyboard LEDs to exfiltrate data from air-gapped computers. Three keyboard LEDs (CAPS, SCROLL and NUM LOCK) are used to encode 3 bits of data (light represents 1 or 0 when it is either on or off), with the capture component using a video camera hidden on a smart watch of an employee or a compromised security camera. In certain cases, such as in Linux, the keyboard LEDs are accessible from user space, so no special permission is required. Transfer rates of 15 to 45 bps can be achieved, depending on the encoding technique, with tests confirming a transfer rate of 30 bps at a distance of 9.5 meters.

- Finally aIR-Jumper (Guri et al. [2017]) demonstrates how near infrared LEDs used to provide light vision illumination for surveillance cameras can be used as a covert channel, both for exfiltration or infiltration purposes, as the cameras can also be used to read data modulated in Infrared (IR) pulses.

The use of wireless network covert channels is addressed by Carpentier et al. [2019] and Ogen et al. [2018], which provide two takes on the subject: the first resorted to Bluetooth controlled light bulbs to implement both light and stenography-encoded payloads to create covert channels, while the second resorts to micro-jamming and backscatter modulation to exfiltrate data over Zigbee or Wifi channels.

Closer to our approach, which is focused on using the building automation infrastructure as a covert channel, there are fewer examples. Tienteu et al. [2017] demonstrated an exfiltration scenario using 802.15.4/ZigBee, which is a consumer IoT solution and not a large scale, structured automation protocol such as KNX or BACnet. A specific BACnet-based exfiltration scenario is demonstrated by Wendzel et al. [2012], which also describe a purpose-built testbed for testing and development purposes.

While this section is by no means exhaustive, the aforementioned examples provide an encompassing perspective on the topic of data exfiltration from air-gapped devices and/or infrastructures, as well as the nature of the side channels already identified and successfully exploited in the past, as summarized in Table 6.1, organized by method, medium, distance and throughput. To the best of our knowledge, this work, also documented in **(j4)**, is the first reported example of using the KNX field bus as a covert channel.

## 6.3 Data Exfiltration Using KNX

Most of the work carried out in the field of data exfiltration is focused on the identification of potential side-channels and analysing to which extend they may be successfully exploited. This work is no exception, as it started as an exercise to identify which systems and active devices could be present on air-gapped spaces, isolated from any type of communication and/or emissions (magnetic, optical, radio frequency, electromagnetic). Quite often, and due to specific confidentiality and security needs or even compliance with specific certifications (which are often mandatory for subcontractors), those isolated spaces are created within the premises of organisational building sites, which are often equipped with BACS. The basic premise for this work is to use the BACS infrastructure

Table 6.1: Surveyed exfiltration techniques

| Method | Medium | Source | Distance [m] | Speed [bps] | Refs. |
|---|---|---|---|---|---|
| Acoustic | Sonic | Speaker | ∼30 | 20 | Carrara and Adams [2014] |
| | | Computer Fan | 8 | 0.3 | Guri et al. [2016b] |
| | | HDD or Optical drive actuators | 8 | 0.3 | Guri [2020] |
| | Ultrasonic | Speaker | ∼20 | 3 | Hanspach and Goetz [2013]<br>Carrara and Adams [2014]<br>Guri et al. [2019b] |
| Electromagnetic | | Video card/cabling | ∼8-30 | 480 | Guri et al. [2014] |
| | | CPU/RAM bus | 5+ | 1-2 | Matyunin et al. [2016] |
| | | CPU | 0.2/0.4/1 | 10/1/1 (loss=0) | Guri et al. [2020] |
| | | USB bus | 9+ | 640 | Guri et al. [2016a] |
| Thermal | | CPU/GPU | ∼0.4 | 0.13 | Guri et al. [2015] |
| | | HVAC | room | 0.83 | Mirsky et al. [2017] |
| Optical | Visible light | Keyboard, HDD, LEDs, Lamps | line of sight | 150 | Loughry and Umphress [2002] Robles-Durazno et al. [2019] Guri [2018b] Guri et al. [2019c] |
| | | LCD Screen | 8 | 20 | Guri et al. [2019a] |
| | Laser | Laser, Scanners, Drones | 1200 | 20 | Nassi et al. [2017] |
| | Infrared | Cameras | 10-100s | 20 | Guri et al. [2017] |
| IoT, BACS, Backscatter sideband | Zigbee, Wifi | Implanted device | 15 | 40 | Ogen et al. [2018] |
| | Bluetooth | Light Bulbs | 25 | 120 | Carpentier et al. [2019] |
| | BACNet | Compromised device | n/a | n/a | Wendzel et al. [2012] |

as a covert channel to exfiltrate data outside isolated spaces and, in some cases, of the building itself.

When compared to most of the exfiltration techniques presented in Section 6.2, using the BACS infrastructure provides several advantages: this infrastructure is usually spread all over the building, making it easier to collect exfiltrated data (when compared with close proximity or line-of-sight requirements of other techniques); it may support bi-directional communications; it may support comparatively good throughput rates; and it can be combined sequentially with some of those techniques for multi-channel/multi-hop exfiltration.

Section 2.1.2 presented the KNX reference architecture that is generally used in BACS/KNX deployments. In these deployments, sensors, switches and actuators are interconnected by a fieldbus which consists of a twisted pair cable. As the BACS permeates the entire building, typically there are several devices in each room, such as switches, fire detectors, presence detectors or BACS-integrated climate control equipment. At each device installation point there will be a bus coupling unit connected to the fieldbus that can be easily accessed, allowing the attacker to access the building's BACS network. The device used by the attacker to access the BACS network can be placed temporarily (e.g. during the time the attacker is alone in a room) or, more likely, can replace an existing device (e.g. replacing a regular light switch with a compromised light switch, non-discernible to the human eye).

Following this rationale, two possible approaches to exploit the BACS field bus for data exfiltration are presented next: sending data to another point of the BACS network, likely located in a less secure room where the attacker can collect the data; and activation of one or more luminaries (possibly visible from some point outside the building), for transmission of sensible data directly to the outside.

### 6.3.1 Exfiltrating Data Through the Fieldbus Network

The main idea of this approach is to send data through the BACS network, using one of the existing automation device connections as an entry point within the air-gapped room, and another automation device located in another space (typically with easier access) as an exit point, therefore allowing to receive the exfiltrated data (see Figure 6.1).

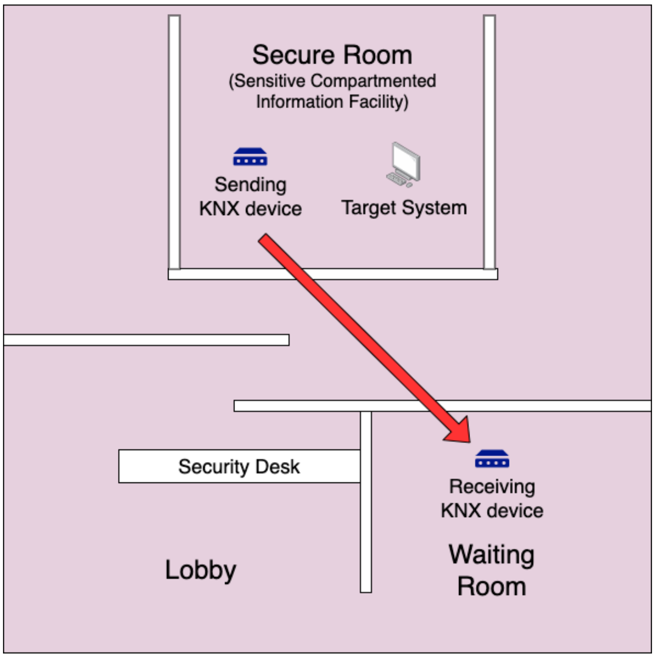


Figure 6.1: Office Plan

How the data originally reaches the sending KNX device is outside the scope of this work. Nevertheless, several options can be considered:

- The attacker is in the protected room, and uses for instance an USB port in the KNX device to directly connect it to the computer or storage device where the sensitive data is located.
- The sending KNX device has for instance concealed microphones, which record private conversations in the protected room and send them, encoded, to the KNX receiving device.
- The computer where the sensitive data is located has been previously compromised, and one of the short-range methods discussed in Section 2 (for instance Guri et al. [2019b], Guri [2020],Guri et al. [2014]) is used to move data from the (previously compromised) target system to the sending KNX device. This sequential use of two different side channels combines the benefits of short range channels (for extracting data from compromised systems) and longer range channels such as KNX, to make the location of the final receiving device much less restricted.

To send significant amounts of data across the network, the attacker must slice it into sets of bytes small enough to be encapsulated within valid KNX messages, to be later reconstructed at the reception point. These messages must also flow over the network in a stealthy way without interfering with normal building operation and without being detectable. This implies understanding how to embed them in normal operation flows. For instance, the peer-to-peer communication messages used mostly in installation and commissioning actions should not be used, since they could trigger an alert as they could be associated to scouting operations.

The standard structure of KNX messages was already discussed in the previous section (cf. Figure 2.7). The majority of messages used in production environments are sent from any device to a Group Address (GA), using broadcast over the serial line. All devices, when commissioned, are programmed to listen for group messages related to their tasks. This architecture allows one-to-many communications, and assumes that when a given device hears a message that is not in its object database it will simply ignore it. Thus, if the messages used to exfiltrate data are addressed to any GA not used in the production environment, they will be ignored by all existing devices except the one used by the attacker as receiver.

Sending a message of this type requires the APCI field to be set for a *Group Value Write* message, that is, the binary value **0010**. The sliced data will then be encapsulated in octets 8 to 21 (a maximum of 14 bytes per frame) – with an empty payload being considered the end of file or stream transmission, for the sake of simplicity, in the case of our proof of concept implementation.

The proof of concept implementation is backwards compatible with any KNX device, as it uses the *Standard Frame* data format. The use of *Extended Frames* could allow to send a larger payload per frame (up to a maximum of 254 bytes), increasing throughput. The KNX design would still allow these frames to be ignored by previous equipment, as they would soon be considered invalid in the control field analysis.

### 6.3.2 Using the Lighting System to Exfiltrate Data

The approach described in the previous section works well when it is feasible to install two compromised KNX devices (sending/receiving), and when the attacker has some sort of access to the receiving device, within the building, to eventually collect exfiltrated data.

However, in some scenarios, later access to the building (to install or access a receiving KNX-device) can be not possible. For those scenarios, an alternative solution is to use already existing and non-hacked building equipment, such as luminaries, as a replacement to the receiving device. For instance, the sending KNX device encodes sensitive data as commands to luminaries in other rooms (possibly visible from outside the building) using binary encoding such as basic Morse code or more sophisticated schema. The induced blinking of these luminaries can be captured by cameras placed outside the building, allowing to reconstruct the transmitted data by decoding the blinking patterns.

Turning a luminary on and off via the KNX protocol consists of sending a *Group Value Write* message, with a value of **1** or **0** respectively, addressed to the GA that is associated with an actuator object for that luminary. Hence, sending a batch of messages in a controlled manner allows sending data according to certain encoding rules. Moreover, the use of multiple luminaries in a synchronised way may allow for parallel transmission, increasing the throughput and/or enhancing stealthiness.

The KNX messages that are sent have a size of 8 bytes: one byte corresponds to the desired state (0 or 1), while the remaining bytes carry control information. It should be noted that the APCI will also be the binary value **0010**, corresponding to the *Group Value Write* function. Such messages are similar to those used to control luminaries in regular building operation, thus hampering detection by stateless mechanisms (since the information is not encoded in the message payload, but rather in the on/off cadence, or in the conjunction between actions on different luminaries). Moreover, the use of an existing source address in the system will also allow for increased message obfuscation and overall stealthiness.

While the previous approach requires accessing the KNX fieldbus in two different locations (to send and to receive sensitive data), this approach only requires accessing the fieldbus to send the sensitive data, since no changes are necessary at the luminaries. Moreover, this makes it possible for the receiver to be outside the building – which in some cases compensates for the lower data rates achievable with this alternative.

To improve stealthiness, several strategies can be adopted. The most basic is to resort to light fixtures placed in spaces where the blinking would not call the attention of building occupants, or to adjust the blinking frequency to make it imperceptible to the human eye (as explored by Robles-Durazno et al. [2019], in a different scope).

A more stealthy approach could be implemented if a dimmer is present. Dimming luminaries in small steps (instead of using on/off commands) may be imperceptible to humans (as demonstrated in Guri et al. [2019a], for LCD screens). Curiously, several KNX dimmers also support a specific load mode that allows them to control solenoid valves or single phase motors, the latter often used to drive ceiling fans or active ventilators, for instance. This could eventually pave the way for the implementation of exfiltration methods akin to those proposed in Guri et al. [2016b].

Specifcally for KNX, one must take into account that dimming control disciplines are pre-established via the ETS tool, which allows to configure parameters related to dimming control (for instance, whether there is a looped single button or two buttons for control), the delays between switching and dimming or the dimming percentages per step. Even if we discard more complicated strategies, such as compromising engineering stations or taking possession of BCU keys (when used to protect devices), which would provide an attacker with the means to fine tune dimming control via provisioning mechanisms, attacks may be implemented via inline control mechanisms. In fact, if enough dimming levels are supported (via small steps), an attacker may generate write messages (akin to the ones produced by a two-button up/down controller) to slightly increase and decrease brightness to modulate data transfers, in a more stealthy way. Finally, this concept could eventually be improved if we consider that many KNX dimmable LED drivers also support RGB led strips, meaning that colour-based encoding could be possible (as shown in Carpentier et al. [2019]).

## 6.4 Proof-of-Concept Implementation

This section describes our proof-of-concept implementations for two of the proposed exfiltration techniques: inline (via KNX bus), and via light dimming.

### 6.4.1 Inline KNX Bus Exfiltration

To this purpose, a Raspberry Pi 4 was adopted as base SBC platform for the KNX devices, due to its availability and low cost, even though real-world attacks would likely resort to devices with smaller footprints and resembling typical KNX components.

The connection to the KNX bus was achieved by the previously presented bus coupling unit in the form of a shield (see Figure 4.2) connected to the SBC GPIO interface. This shield contains a TP/UART (optically isolated from the SBC host using a ILD213T optocoupler) that provides fieldbus connectivity, as well as an ATmega 2560 micro controller that establishes the interface between one of the SBC serial ports and the TP/UART serial line (see Figure 6.2).

Similarly to what happened with the PoC NIDS described in Chapter 4, the SBC Operating System is based on a Debian Linux distribution (Raspbian), and the toolset was developed using the GO language, together with the open source GoPacket library by Google Inc. [2022] for decoding and encoding messages.

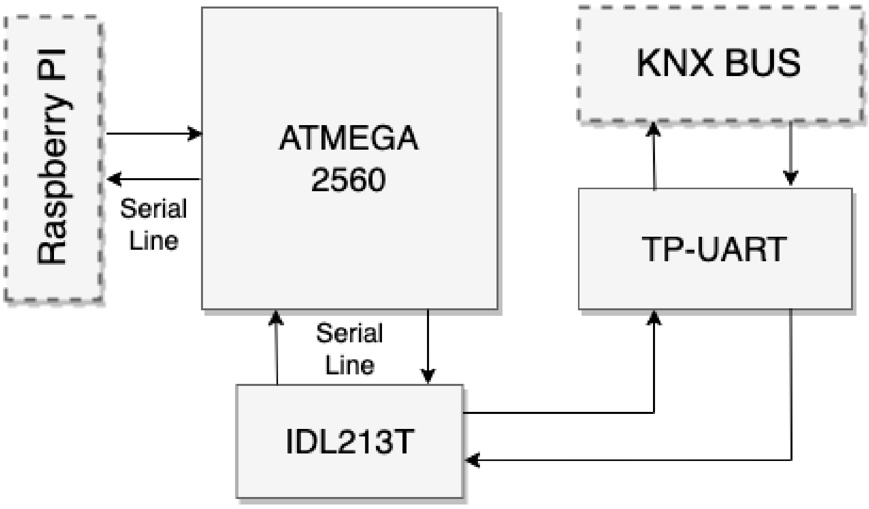


Figure 6.2: Shield diagram

Apart from the KNX-TP shield (which was integrated into a single printed circuit board), the whole prototype was built using commercial of-the-shelf hardware components, with no special optimisation. The SBC could be stripped from unnecessary components, eventually being replaced by a Raspberry Pi *Compute module*. Also, an increased component integration on the shied could be achieved, along with the removal of components used for debugging, like LED's, push buttons and LCD connections. This means that the physical device form factor could take a fraction of the prototype footprint and could certainly be resized to the thickness of a credit card.

In the scope of this work, and regarding the proposed techniques, two command line applications were developed (*KNXsend* and *KNXreceive*) that, when used in devices connected to different points of the KNX bus, allow sending and receiving a file, thereby creating a covert channel for data exfiltration.

As already mentioned, it should be noted that this chapter is mostly focused on demonstrating how the KNX network could be exploited to exfiltrate data out of an isolated/air-gapped space, not considering the physical intrusion/access process or how the target data was obtained in the first place, as such matters are outside the scope of the present work.

The first thing to do after gaining access to the restricted/secure physical space is to locate a KNX component, replacing it by the prototype device, which will be connected to the fieldbus, to enable data transmission. A counterpart device must also be discreetly deployed in a place with unrestricted access, in order to receive the transmitted data.

First, the *KNXreceive* application is configured to receive messages from the sender with IA address 1.1.4, and destined to the GA address 5/5/5. The device connected to the bus will (stealthily) listen to any messages travelling through the bus using the above identified source/destination pairs. Second, the data file to be transmitted is provided to the *KNXsend* application, which will slice it into byte blocks, for encapsulation in KNX messages using IA 1.1.4 as source and GA 5/5/5 as destination.

The developed code uses an internal buffer with a maximum of 14 bytes, in order to guarantee that the data sent in each message can use standard KNX frames. Depending on the building, the payload size could be increased with the use of extended KNX frames, therefore enabling to use up to 254 byte payloads.

### 6.4.2 Dimmer-based Exfiltration

The implementation of the exfiltration method described in Section 6.3.2 only requires one device, connected in a similar way at the location from where the data is accessed and exfiltrated from. Prior scouting work would be carried out to identify the KNX group addresses associated with the intended (already existing) luminaries. Finally, the sending application would generate dimming commands (or on/off messages, in alternative) at controlled time intervals to activate the luminaries, whose dimming (or blinking) patterns would be captured by video cameras.

For this specific case, one possible approach may be based on a low-rate transmission strategy (compatible with a KNX control flow rate), for instance, sending bits by means of Differential Pulse Position Modulation (D-PPM). D-PPM encodes bits as sequences of pulses with a variable gap (see Figure 6.3). This way, the pulse position is encoded in such a way that the receiver must only measure the difference in the arrival time of successive pulses to distinguish between zeros and ones.

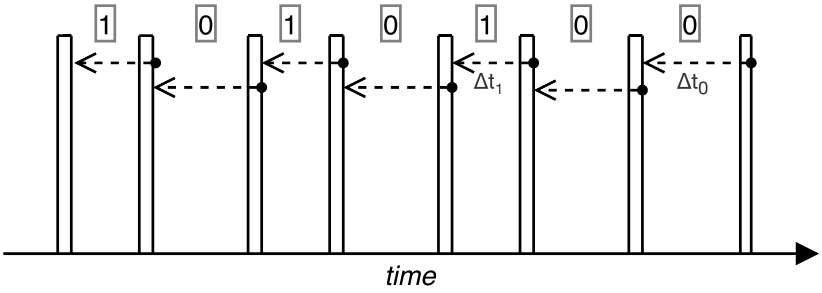


Figure 6.3: D-PPM modulation

In order to undertake a feasibility evaluation study, a proof-of-concept device was built using an Arduino Uno microcontroller, which communicates with the SBC via I2C. This Arduino is coupled to a LED driver circuit using 3 TIP120 transistors, connected to a 12V WS2812 LED strip. The implementation for the modulation side follows Algorithm 6.1. For brightness control, we used the Pulse Width Modulation (PWM) capabilitiy of the ATMega microcontroller, which was deemed suitable for this application since most LED strip dimmers use PWM, with variable duty cycles to control LED brightness levels.

In the specific case of this PoC implementation, two brightness levels were configured: $level_{low}$ and $level_{high}$, corresponding to 90% and 100% duty cycle modes.

The difference between them is practically indistinguishable to the human eye, but can be easily detected by a smartphone camera.

**Algorithm 6.1:** D-PPM modulation

```
Pre-established: $bitdelay$, the impulse duration; $zerodelay$, the
  duration for a binary "0"; $onedelay$, the duration for a binary "1".
  Brightness levels adjusted for two-step dimming ($high$ and $low$).
  procedure BITCODE(boolean $bit$)
      brightness($level_{high}$)
      delay($bitDelay$)
      brightness($level_{low}$)
      if $bit == 0$ then
          delay($zerodelay$)
      else
          delay($onedelay$)
  end procedure
```

The demodulation component (which could be located outside the building, as long as there is line of sight to the LED strip) may run in a smartphone or a dedicated device. In the specific case of the proof-of-concept implementation, Python was used, along with the *opencv*, *pandas*, *scipy* and *numpy* libraries.

Algorithm 6.2 showcases the demodulation process, which is performed from a captured video file. First, the user needs to define a Region of Interest (ROI) in the video stream, in order to speed the decoding process and avoid unwanted noise sources – for this reason, the video capture must be performed from a stable device in order to make sure the ROI is not shifted over time. Subsequently, all the video frames are preprocessed via cropping (to ROI boundaries), converted to grayscale and subject to a average blur filter with a 6x6 kernel before being averaged. The brightness averages for each frame are stored on the *avgvalues* list.

Once all frames are processed, the first derivative of the resulting time series is calculated. The gradient eases the task of locating the rising edges of the Pulse Position Modulation (PPM) pulses (detection thresholds are calculated accordingly with the algorithm). From this point on, it is just a question of locating the timestamps for each edge and computing the time deltas between consecutive edges. The results obtained with our proof-of-concept implementation of the D-PPM dimming schema will be discussed next.

## 6.5 Validation

The validation of the proposed techniques was carried out in two different scenarios. The first consisted of a laboratory testbed that allowed the definition of a performance baseline for inline bus exfiltration, also providing the integration context for the dimming scenarios. A single-family dwelling provided the second validation environment, geared towards validation of inline bus exfiltration in

**Algorithm 6.2:** D-PPM demodulation

```
INPUT: Video capture file
    $avgvalues \leftarrow ()$
    $bitstream \leftarrow ()$
    $blur_{ksize} \leftarrow (7, 7)$
    $sample \leftarrow \textsc{loadframe}()$
    $ROI \leftarrow \textsc{select}\mathrm{ROI}(sample)$
    foreach $f \in frames$ do
        $cropped_f \leftarrow \textsc{crop}(f, ROI)$
        $gray_f \leftarrow \textsc{grayscale}(cropped_f)$
        $blurred_f \leftarrow \textsc{averageblur}(gray_f, blur_{ksize})$
        $avg \leftarrow \textsc{Average}(blurred_f)$
        $avgvalues.\textsc{append}(avg)$
    $grad \leftarrow \textsc{gradient}(avgvalues)$
    $max_g \leftarrow \textsc{max}(\textsc{abs}(grad))$
    $min_g \leftarrow \textsc{min}(\textsc{abs}(grad))$
    $threshold \leftarrow -(min_g + (max_g - min_g)/3)$
    $timestamps \leftarrow grad.\textsc{where}(gradient > threshold)$
    foreach $t \in t_{frames}$ do
        $deltas \leftarrow timestamps.\textsc{diff}(t, t+1)$
    foreach $d \in deltas$ do
        if $d > 400ms$ then
            $bitstream.\textsc{append}(0)$
        else
            $bitsteam.\textsc{append}(1)$
OUTPUT: $bitstream$, Demodulated bit stream
```

a production environment, to assess potential mutual interference between the exfiltration process and the regular operation of the building.

### 6.5.1 Laboratory Testbed

Section 4.2 describes the created testbed, shown on Figure 4.3, that was also used to test and assess the PoC implementation of the BACS NIDS.

This laboratory testbed also hosts the custom LED strip driver, depicted in figure 6.4. This driver will be used for the dimming exfiltration tests.

This simple installation provides the means to replicate most of the operations existing in a BACS. It can be configured, commissioned and debugged through the KNX ETS, using the USB gateway. It also provides connection for one or multiple KNX devices like those built to host the *KNXsend* and *KNXreceive* applications.

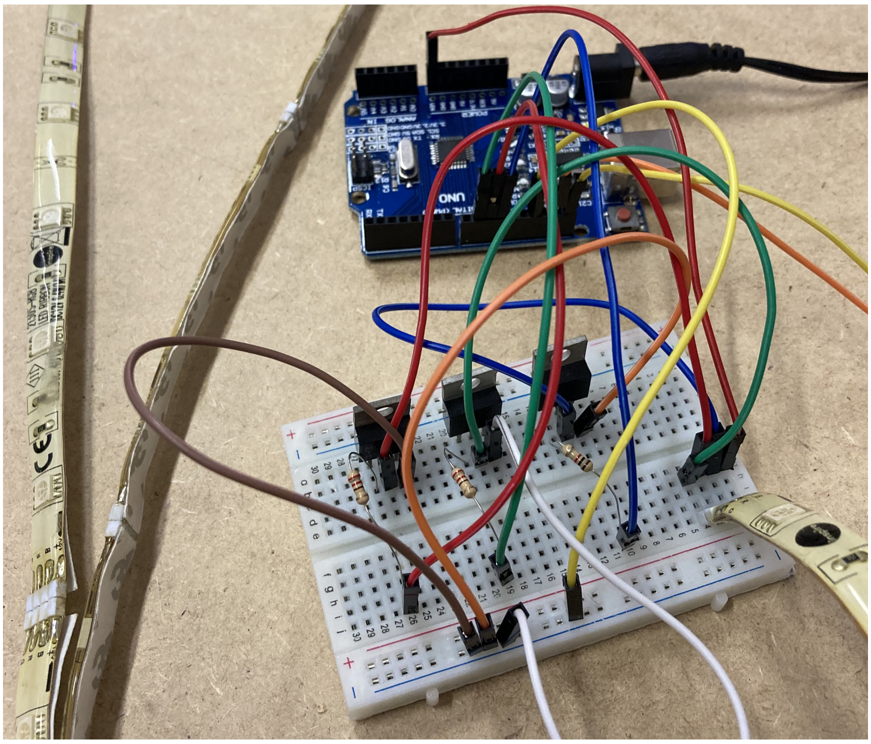


Figure 6.4: LED Strip Driver

### 6.5.2 Production Environment

Additionally to the laboratory environment, tests were carried out in a single-family house with three floors, as described in Section 5.3.

This house can be compared to an office building in terms of the extension of its automation network and the diversity of existing equipment, so it was used to validate the described attack scenario. Thus, it was assumed that the existing storage in the basement would be the room where we had the data to be exfiltrated. This room does not have any fenestration, however it has lighting and the respective KNX switch. A room located next to the main entrance, on the ground floor, would be the waiting room where the attacker's accomplice would be receiving the exfiltrated data.

### 6.5.3 Inline Exfiltration Tests

The inline exfiltration tests performed in both scenarios (testbed, production environment) consisted of sending files with content randomly generated from */dev/random*, with the following sizes: 1K, 10K, 100K and 1M bytes. The tests were repeated ten times and the total transmission and reception times, as well as the bytes transmitted and received, were recorded. It was confirmed that the received content matched the sent content.

Table 6.2 shows the measurements obtained in the laboratory testbed. A peak rate of 712 bps was obtained in this controlled environment, with no packet losses. It should be noted that, during the execution of these tests, the only messages transmitted on the bus were those related to the ongoing data exfiltration. The standard deviation for the measurements is very small, which is explained by the stability of the field bus. Variations were proportional to the size of transmitted files and, consequently to the number of transmitted packets.

Table 6.2: Data exfiltration performance – Lab testbed

| File Size | Direction | Mean latency [s] | Std [s] | Throughput [bps] | Packet loss [%] |
|---|---|---|---|---|---|
| 1K | send | 11,489 | 0,0003 | 713 | 0 |
| | receive | 11,581 | 0,0003 | 707 | |
| 10K | send | 115,046 | 0,0031 | 712 | 0 |
| | receive | 115,149 | 0,0033 | 711 | |
| 100K | send | 1151,065 | 0,0261 | 712 | 0 |
| | receive | 1151,179 | 0,0255 | 712 | |
| 1M | send | 11787,245 | 0,1632 | 712 | 0 |
| | receive | 11787,511 | 0,1688 | 712 | |

The measurements obtained in the production environment (house scenario) are provided in Table 6.3, and denote a higher standard deviation, that results from the impact of the regular KNX traffic that is present in the field bus. Nevertheless, measured throughput is identical, suggesting that for small office networks the impact on such traffic does not pose a significant impact. The measured packet loss is very low but not zero, which suggests real world attacks definitively need to support recovery from packet losses (as was the case of our PoC). Building occupants did not report any strange behaviour from the building's automation system, which is a good indicator of the stealthiness of the proposed technique – moreover if we consider that some of the tests lasted longer than three hours.

Table 6.3: Data exfiltration performance – Production environment

| File Size | Direction | Mean latency [s] | Std [s] | Throughput [bps] | Packet loss [%] |
|---|---|---|---|---|---|
| 1K | send | 11,492 | 0,0057 | 713 | 0 |
| | receive | 11,563 | 0,0055 | 708 | |
| 10K | send | 115,086 | 0,0312 | 712 | 0,014 |
| | receive | 115,170 | 0,0316 | 711 | |
| 100K | send | 1151,361 | 0,0832 | 712 | 0,017 |
| | receive | 1151,456 | 0,0847 | 711 | |
| 1M | send | 11791,693 | 1,9070 | 711 | 0,019 |
| | receive | 11791,947 | 1,9082 | 711 | |

### 6.5.4 Visible Light Dimming Exfiltration Scenarios

For evaluation purposes, the dimming driver controller was deployed in two different setups (see Figure 6.5): a close range setup, within the laboratory; and an outdoors scenario with straight line distance of approximately 4 meters and 15 meters, respectively. In both cases, the ASCII-encoded string ”ABC” was continuously sent, in a loop. Video capture was performed using a second generation Apple iPhone SE, in 1080p HD mode, at 30fps.

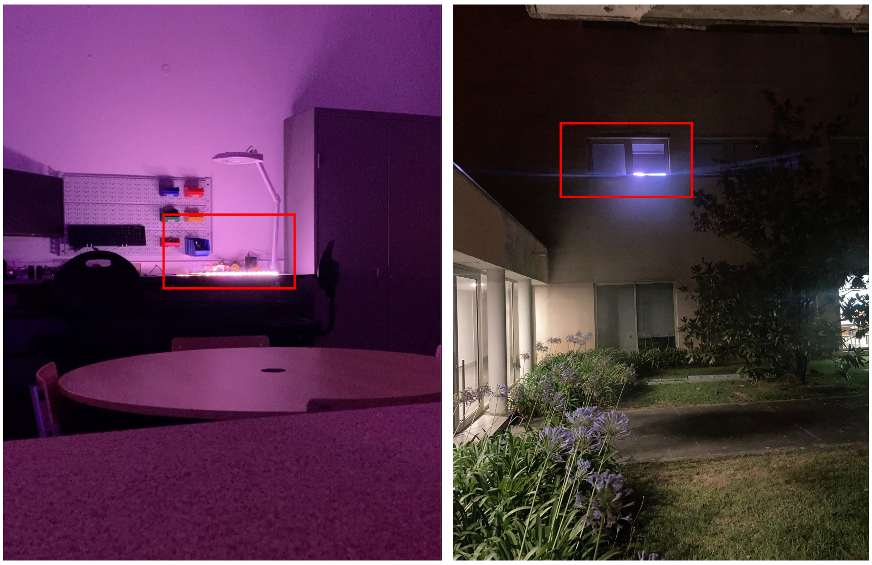


Figure 6.5: Dimming Test Setups (red rectangles locate the light sources)

Figures 6.6, 6.7 and 6.8 present the results obtained with the decoding algorithm presented in Section 6.4.2. The trade-off between ROI dimension and demodulation accuracy was studied for several boundary sizes, starting with a 250x100 rectangle centred in the light source, increasing the width and height in steps of 50 pixels. Results are shown in Figure 6.6, demonstrating that in our case the optimum ROI (no observed errors) was between 300x150 and 500x350.

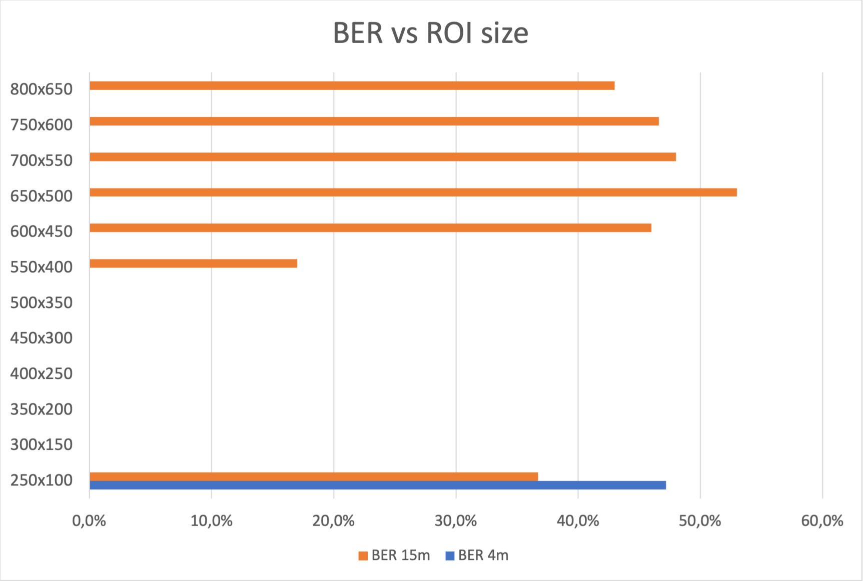


Figure 6.6: Bit Error Rate (BER) vs. ROI size

As expected, the correct choice of the ROI area seems to be particularly relevant for the outdoors scenario. Most likely this is due to the fact that a balanced

ROI boundary size may filter some noise while, at the same time, allowing for optical phenomena (such as light scattering, which may vary accordingly with room and window characteristics) to enhance the detection rate. Camera setup is also crucial for a good capture: optical artefacts introduced by auto-focus and automatic sensitivity adjustment greatly affect the demodulation process, as they induce brightness distortion.

Figures 6.7 and 6.8 provide a visualisation of the time series gradients which are used for D-PPM demodulation purposes. Red lines depict the inter-pulse widths corresponding to the time deltas between consecutive bits. While threshold detection is undertaken using a simple heuristic (as shown in Algorithm 6.2), in many cases a manual adjustment may be required to improve sensitivity.

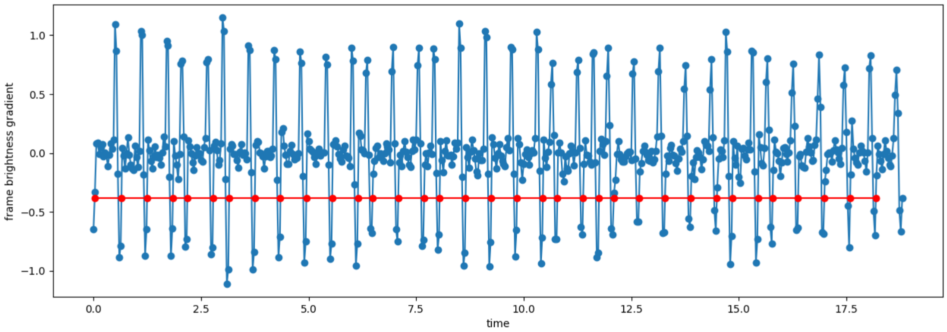


Figure 6.7: Results for 4m indoor test, 350x200 ROI area

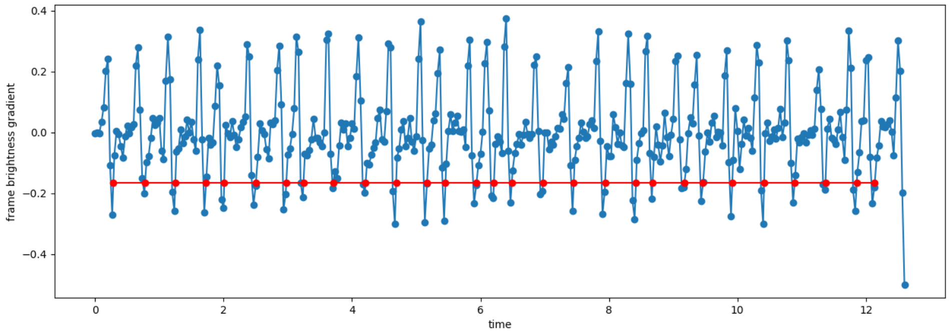


Figure 6.8: Results for 15m outdoor test, 350x200 ROI area

For this proof of concept implementation, transmission speeds around 2 bps were measured. Despite being much lower than the first exfiltration method, these rates are still within typical ranges observed for more exotic side channels. Moreover, there is a good margin for improvement by means of finely tuning delay, as well as the introduction of more evolved modulation techniques and better signal processing, for noise filtering and thresholding.

## 6.6 Possible Countermeasures

In this section we discuss possible countermeasures for the proposed exfiltration techniques.

While these techniques are relatively simple and easy to implement, they take advantage of the fact that the KNX fieldbus is typically overlooked when analysing IT security but is still widespread across the building, with relatively easy physical access. This fieldbus is often air-gapped and not connected to the local network or the Internet, which keeps it outside the radar of IT staff. Moreover, monitoring the KNX fieldbus traffic is not common, increasing its potential for stealthy data exfiltration. To the best of our knowledge, there are no commercially available security monitoring tools for KNX field bus, and even at the academic level, there are very few such proposals – which motivated our proposal of the BACS NIDS, discussed in the previous chapters.

Apart from simply excluding KNX devices from sensitive locations, all other countermeasures require monitoring the field bus traffic, in order to detect anomalous behaviours. However, the most basic detection approaches (e.g. based on traffic volumes or stateless packet inspection) might render ineffective, since attackers may hide data exfiltration in packets which resemble normal building control operations packets, and may also adjust the exfiltration throughput to keep it bellow traffic volume thresholds.

This problem is well known in local IP networks, and some lessons may be potentially translated to and reused in KNX scenarios (e.g. machine-learning techniques for detecting anomalous traffic). Nevertheless, it is always necessary to adapt them to the specific nature of the KNX field bus and the messages it conveys.

In the situation where the exfiltration involves sending a batch of messages destined to a specific GA outside the range of those being used in the system, detection may be enhanced by using traffic whitelisting, together with a ruleset for detecting messages whose destination address is unknown or not used in normal operation.

With regard to the situation of abusive lightning control, in cases where it is common for the activation of lights to be driven by motion sensors, it is possible to detect invalid context situations. In other words, commands sent to luminaries in rooms were there is no movement detection (i.e. no occupants) should be flagged as suspicious. However, applying these heuristics to medium and large-sized buildings quickly becomes unpractical, at least without the support of artificial intelligence techniques such as decision trees.

Alternatively, in both situations, the computation of statistical values related to the existing traffic on the network (for instance, by using time series of timed Markov chains) could also be a trigger for detecting and identifying abnormal traffic patterns which may correspond to ongoing exfiltration activities. Screening this traffic would allow the isolation of messages corresponding to abnormal frames for further analysis. These techniques can be further enhanced by resorting to modelling of protocol interactions, using for instance Discrete Time Markov Chains or Finite State Automata (see Graveto et al. [2019] and Graveto et al. [2022a]).

## 6.7 Summary

This chapter demonstrated how BACS can be abused for exfiltrating sensitive data, constituting a potential security risk for organisations, moreover if we consider that such data theft can be carried out stealthily (as it was hereby demonstrated).

Evaluation results for the proposed proof-of-concept prototype have shown a measured transfer rate of 711 bps over the KNX field bus, with no packet loss for a controlled environment and negligible packet loss for a production BACS. This is more than enough for the timely transmission of relevant information. Knowing that, for instance, a Bitcoins private key takes only 256 bits or that a SSL private certificate usually occupies 2048 bits, it would take only 3 or 24 seconds, respectively, for its exfiltration. These figures have room for improvement, as the validated prototype didn't implement compression, error correction or any other possible optimisation methods, paving the way for future research and development directions. Moreover, this exfiltration technique can be bi-directional. They allow attackers to insert malicious code/data in the target systems or to control them from a remote Command & Control Server.

A second method involved KNX commands to building luminaries to access exfiltrated data from outside the building. While achieved data rates were considerably lower, as expected, they are still within the usable range for many practical use cases (e.g. exfiltration of passwords or certificates), and there is also considerable room for improvement.

Finally, it should be pointed out that this work provides a perspective on a side channel that has not yet been explored. Contributing to advise the attention of the industrial and research communities to the importance of researching and developing adequate protection measures, also calling for the introduction of proper regulation and policies to protect BACS installations – both existing and new – from potential abuse.

# Chapter 7

# Conclusions and Future Work

Contents

DESPITE the first building automation solutions being found in the 1970s, and their continuous evolution, their security is still embryonic, and much still needs to be done. The security of these systems was based on isolation and the use of proprietary and non-documented systems, supporting (hypothetical) security mainly by obscurity. Nowadays, the search for autonomous and automation solutions is growing, meeting the human desire to use machines for their benefit. Smart buildings, or as we call them in this thesis, those that have a Building Automation and Control Systems, are an example of this historic human will. At the same time, the evolution of IT networks, exploring remote management facilities and their interconnection with BACS legacy systems, exposes them to a new paradigm and a wide range of threats. In the scope of this thesis, these security and safety problems were addressed, and a framework that contributes to significantly increase the reliability, safety and security of the BACS was proposed.

This chapter synthesizes the most relevant thesis outcomes, results and achievements, also pointing out future research directions.

## 7.1 Synthesis of the Thesis

The research work developed within the scope of this thesis had as its primary objective the detection of potential cyber attacks on BACS, focusing this detection on local monitoring of the existing building infrastructure. In addition, the first chapter identifies some secondary goals and the motivations needed to achieve that objectives. The chapter concludes with a presentation of this PhD work's outcomes and an overview of this thesis.

Chapter 2 presents a systematic literature review that provides a comprehensive perspective on the BACS domain, allowing to identify its most pressing needs and weaknesses regarding safety and security. This made it possible to define the scope of the work more clearly. This chapter also presents the concepts associated with BACS and compares classic BACS with partial building automation based on IoT. Finally, a brief description of the BACS systems powered by KNX was included as support for describing the developed PoC.

Chapter 3 proposes the concept of having domain-specific NIDS for BACS. This chapter also provides some background on BACS-specific security and intrusion detection for consistency and completeness. This new framework enables detection using rules/heuristics and/or artificial intelligence algorithms. New horizons are also opened for anomaly and threat detection in the immense possibilities available when these techniques are used autonomously or combined.

A Proof of Concept for KNX-based BACS was presented in Chapter 4. This BACS NIDS is used for cyber attack detection and anomaly identification. Its harware and software architecture was extensively detailed. The use of this framework in other systems (BACnet, LonWorks, etc.) requires some minor changes, that were also discussed. Finally, a materialised laboratory testbed was built for the initial development and test phases.

Chapter 5 addressed the validation of the entire developed system. Different use cases supported this validation process and the obtained results. The use of rules, heuristics, and AI-based detection was showcased, demonstrating the power of the BACS-specific NIDS. This chapter also showed the potential for future development using specific research to improve detection algorithms.

Finally, in Chapter 6, some threats to privacy and data security arising from the coexistence of BACS and secure air-gapped spaces were presented and demonstrated. This chapter presents a new approach, creating a covert channel for data exfiltration using BACS. The chapter concludes with a discussion of some possible mitigation approaches.

Overall, the goals of this work were achieved, starting with identifying needs, existing threats and available solutions, followed by the definition of objectives, their implementation and finally, a validation process that resulted in several contributions detailed in the next section.

## 7.2 Contributions

The research work carried out within the scope of this thesis was governed by the objectives defined in Chapter 1, leading to the following contributions:

- **Literature Review.** The first objective was archived with the State of the Art analysis of Building Automation and Control Systems identifying leading solutions, existing problems and open issues and was published in **(j2)**;
- **Definition of an Architecture for BACS Security and Safety.** A new architecture was developed to process collected information and identify and alert possible anomalies and/or attacks on BACS. The proposed architecture allows the analysis of all messages through rules, heuristics and using artificial intelligence techniques. The deployment is technology independent from the automation systems **(j3)**;
- **Development of a Proof of Concept Using a KNX BACS.** The proposed architecture was developed as a Network Intrusion Detection System specific for a KNX-based BACS, and the main results were published and discussed in **(j3)**;
- **Creation of BACS Datasets.** A dataset bundle was created from traffic acquired on a residential BACS infrastructure based on KNX. All the field bus messages were logged and timestamped, and the injection of several attacks was carried out. The created datasets are publicly available in **(c1)**;
- **Development of a KNX Deep Packet Inspector Module.** According to the protocol specifications, a new module for coding and decoding KNX

messages was developed and made available to the widely used open-source project *GOPacket* **(c2)**; and

- **Exploration and Analysis of KNX-based Side Channel Attacks.** New ways for potential abuse vectors were identified that could be leveraged to steal data from secure facilities using BACS covertly. Thus, some forms of data exfiltration were demonstrated and validated, and some possible mitigation actions were also discussed **(j4)**.

These contributions and the performed literature review were published or submitted in journal articles as detailed on initial *Foreword* section.

## 7.3 Future Work

The work developed and the contributions of this PhD thesis open an opportunity for the future development of several lines of research. For instance, regarding the proposed architecture, the development of prototypes for different BACS environments with other protocols, such as BACnet and LonWorks, will contribute to further evolving the proposed security solution. Validation in these other environments will allow the architecture to extend to multi-protocol environments resulting from the phased implementation of home automation in buildings by different installers/suppliers. The improvement of building safety conditions could also be empowered with the incorporation of IoT devices in the proposed architecture, using the *Communication Stream Analysis* module, since most of these use the LAN for their communication.

Another potential line of research is the exploration of AI potentialities, namely the use of diversified techniques to enhance and expand system and security anomaly detection capabilities by taking advantage of the versatility of the proposed solution. The possibility of creating ensemble approaches based on both rules/heuristics and AI also constitutes a potential development path. It should be noted that, despite the validation effort covering a series of potential anomaly and attack use case scenarios, there is space for further research regarding this topic, enabling the development of detection capabilities for other technologies and evolved threat profiles.

Finally, using the proposed monitoring capabilities associated with the development of Bus Coupling Units for different protocols used in BACS will enable the creation of datasets of BACS communications using those protocols. These datasets will enrich the range of tools available to study and investigate the BACS, opening the path for future research in optimizing their operation, safety and security. Research to improve living conditions and sustainable buildings with automation will be unquestionable.

# References

Abdulmunem, A.-s. M. Q., Al-khafaji, A. W., and Kharchenko, V. S. (2016). The Method of IMECA-based Security Assessment : case study for buildind automation system. *EU ERASMUS+ Project: Internet of Things: Emerging Curriculum for Industry and Human Applications (ALIOT)*, 1(138).

Abunaser, M. and Alkhatib, A. A. A. (2019). Advanced survey of Blockchain for the Internet of Things Smart Home. *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, pages 58–62.

Ahmed, M., Naser Mahmood, A., and Hu, J. (2016). A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, 60:19–31.

Al-Maksousy, H. H., Weigle, M. C., and Wang, C. (2018). NIDS: Neural Network based Intrusion Detection System. *2018 IEEE International Symposium on Technologies for Homeland Security, HST 2018*, pages 14–19.

Ali, W., Dustgeer, G., Awais, M., and Shah, M. A. (2017). IoT based Smart Home : Security Challenges, Security Requirements and Solutions. *2017 23rd International Conference on Automation and Computing (ICAC)*, (September):7–8.

Alisic, R., Molinari, M., Pare, P. E., and Sandberg, H. (2020). Ensuring privacy of occupancy changes in smart buildings. *CCTA 2020 - 4th IEEE Conference on Control Technology and Applications*, pages 871–876.

Amazon (2014). Amazon Alexa. `https://developer.amazon.com/en-GB/alexa`. Last visited: 2022-10-06.

Anderson, R. (2020). *Security engineering: a guide to building dependable distributed systems*. John Wiley & Sons.

Anderson, R. J. and Kuhn, M. G. (1999). Soft tempest–an opportunity for NATO. *Protecting NATO Information Systems in the 21st Century.*

ANSI (2010). Smart Grid Standards Information Section I : Use and Application of the Standard Section I : Use and Application of the Standard.

ANSI/CEA (2010). Smart Grid Standards Information Section I : Use and Application of the Standard. *Power Engineering*, pages 1–12.

Anton, S. D. D., Sinha, S., and Schotten, H. D. (2019). Anomaly-based intrusion detection in industrial data with svm and random forests. In *International conference on software, telecommunications and computer networks (SoftCOM)*, pages 1–6. IEEE.

Antonini, A., Maggi, F., and Zanero, S. (2014). A practical attack against a knx-based building automation system. In *2nd International Symposium on ICS & SCADA Cyber Security Research 2014*, ICS-CSR 2014, page 53–60, Swindon, GBR. BCS.

Anwar, M. N., Nazir, M., and Mustafa, K. (2017). Security Threats Taxonomy : Smart-Home Perspective. *2017 3rd International Conference on Advances in Computing,Communication & Automation (ICACCA) (Fall).*

Asadullah, M. and Raza, A. (2016). An Overview of Home Automation Systems. *2016 2nd International Conference on Robotics and Artificial Intelligence (ICRAI)*, pages 27–31.

Ashibani, Y. and Mahmoud, Q. H. (2017). An Efficient and Secure Scheme for Smart Home Communication using Identity-Based Signcryption. *2017 IEEE 36th International Performance Computing and Communications Conference (IPCCC).*

ASHRAE (2016). ANSI/ASHRAE 135 - A Data Communication Protocol for Building Automation and Control Networks. `https://t.ly/btsf`. Last visited: 2022-10-06.

ASHRAE (2020). Bacnet website. `http://www.bacnet.org`. Last visited: 2022-10-06.

Bajer, M. (2018). IoT for smart buildings - long awaited revolution or lean evolution. *2018 IEEE 6th International Conference on Future Internet of Things and Cloud (FiCloud)*, pages 149–154.

BCU SDK (2006). EIBD. `https://www.auto.tuwien.ac.at/~mkoegler/index.php/eibd`. Last visited: 2022-10-06.

Bhosale, D. A. and Mane, V. M. (2015). Comparative study and analysis of network intrusion detection tools. *International Conference on Applied and Theoretical Computing and Communication Technology, iCATccT 2015*, pages 312–315.

Bondarev, S. E. and Prokhorov, A. S. (2017). Analysis of Internal Threats of the System "Smart Home" and Assessment of Ways to Prevent Them. *2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, pages 788–790.

Brauchli, A. and Li, D. (2015). A Solution Based Analysis of Attack Vectors on Smart Home Systems. *2015 International Conference on Cyber Security of Smart Cities, Industrial Control System and Communications (SSIC).*

Brooks, D. J., Coole, M., Haskell-Dowland, P., Griffiths, M., and Lockhart, N. (2017). Building Automation & Control Systems: An Investigation into Vulnerabilities, Current Practice & Security Management Best Practice. Technical report, ASIS Foundation. Last visited: 2022-10-06.

Buczak, A. and Guven, E. (2015). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials*, PP(99):1.

Bugeja, J., Jacobsson, A., and Davidsson, P. (2018). Smart Connected Homes. In *Internet ofThings A to Z: Technologies and Applications*, chapter 13, pages 359–384. Wiley-IEEE Press, 1 edition.

Building Energy Management Open Source (2019). BEMOSS™ Features. `https://www.bemoss.org/overview/`. Last visited: 2022-10-06.

Butzin, B., Golatowski, F., and Timmermann, P. D. (2017). A Survey on Information Modeling and Ontologies in Building Automation. *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society.*

Carpentier, E., Thomasset, C., and Briffaut, J. (2019). Bridging the gap: Data exfiltration in highly secured environments using bluetooth IoTs. *Proceedings - 2019 IEEE International Conference on Computer Design, ICCD 2019*, (Iccd):297–300.

Carrara, B. and Adams, C. (2014). On acoustic covert channels between air-gapped systems. In *International Symposium on Foundations and Practice of Security*, pages 3–16. Springer.

CEN/CENELEC/ETSI (2012). Smart Grid Coordination Group: Smart Grid Information Security. `https://t.ly/L2nB`. Last visited: 2022-10-06.

CENELEC (2012a). EN 13321 - Open Data Communication in Building Automation, Controls and Building Management - Home and Building Electronic Systems solution. `https://t.ly/WOiF`. Last visited: 2022-10-06.

CENELEC (2012b). EN50090 - Home and Building Electronic Systems (HBES). `https://t.ly/_1k0`. Last visited: 2022-10-06.

Chavis, J. S., Buczak, A., Rubin, A., and Watkins, L. A. (2020). Connected Home Automated Security Monitor (CHASM): Protecting IoT Through Application of Machine Learning. *Proceedings - 2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 0684–0690.

Chhetri, C. and Motti, V. G. (2019). *Eliciting Privacy Concerns for Smart Home Devices from a User Centered Perspective.* Springer International Publishing.

China Machinery Industry Federation (2020). GB/T 20965 - Control network HBES technical specification. Home and building control system. `https://www.chinesestandard.net/PDF/English.aspx/GBT20965-2013`. Last visited: 2022-10-06.

Chowdhury, O. (2019). Expat : Expectation-based Policy Analysis and Enforcement for Appified Smart-Home Platforms. *24th ACM Symposium on Access Control Models and Technologies*, pages 61–72.

Chris Green; Martin Roesch (2020). SNORT Users Manual 2.9.16. `https://www.snort.org/documents`. Last visited: 2022-10-06.

Ciholas, P., Lennie, A., Sadigova, P., and Such, J. M. (2019). The Security of Smart Buildings: a Systematic Literature Review. `http://arxiv.org/abs/1901.05837`. Last visited: 2022-10-06.

Cisco Systems, Inc. (2020). SNORT - Network Intrusion Detection & Prevention System. `https://www.snort.org`. Last visited: 2022-10-06.

Cohen, W. H. (1995). Fast effective rule induction. *Proceedings of the Twelfth International Conference on International Conference on Machine Learning (ICML'95)*, pages 115–123.

Connectivity Standars Alliance (2021). Zigbee. `https://zigbeealliance.org/solution/zigbee/`. Last visited: 2022-10-06.

Coppolino, L., Alessandro, V. D., Antonio, S. D., Lev, L., and Romano, L. (2015). My Smart Home is Under Attack. *2015 IEEE 18th International Conference on Computational Science and Engineering*, pages 145–151.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3):273–297.

Darabseh, A. and Freris, N. M. (2019). A software-defined architecture for control of IoT cyberphysical systems Prominent applications enlist intelligent transportation. *Cluster Computing*, 8.

Dasari, S. V., Mittal, K., Sasirekha, G. V., Bapat, J., and Das, D. (2021). Privacy enhanced energy prediction in smart building using federated learning. *IEEE International IOT, Electronics and Mechatronics Conference, IEMTRONICS 2021*, pages 0–5.

Demeure, A., Caffiau, S., Elias, E., Roux, C., Demeure, A., Caffiau, S., Elias, E., Building, C. R., Home, U., Systems, A., and Study, A. F. (2016). Building and Using Home Automation Systems : A Field Study. *ISEUD 2015*.

Deng, I. (2018). Tencent engineer slapped with fine for hacking hotel Wi-fi in Singapore.

digitalSTROM AG (2019). Smart Home by digitalSTROM: A home of unlimited possibilities. `https://www.digitalstrom.com/en/technology/`. Last visited: 2022-10-06.

Ding, D., Han, Q.-L., Xiang, Y., Ge, X., and Zhang, X.-M. (2018). A survey on security control and attack detection for industrial cyber-physical systems. *Neurocomputing*, 275:1674–1683.

Domingues, P., Carreira, P., Vieira, R., and Kastner, W. (2016). Computer Standards & Interfaces Building automation systems : Concepts and technology review. *Computer Standards & Interfaces*, 45:1–12.

Dupont, G., Den Hartog, J., Etalle, S., and Lekidis, A. (2019). A survey of network intrusion detection systems for controller area network. *2019 IEEE International Conference on Vehicular Electronics and Safety, ICVES 2019.*

Dutta, J. and Wang, Y. (2018). ES3B : Enhanced Security System for Smart Building using IoT. *2018 IEEE International Conference on Smart Cloud (SmartCloud).*

EIBA (2020). European installation Bus Association. `https://www.itwissen.info/en/European-installation-bus-association-EIBA.html`. Last visited: 2022-10-06.

EN, I. S. O. (2014). EN ISO 16484 - Building Automation and Control Systems. `https://www.iso.org/standard/63753.html`. Last visited: 2022-10-06.

EnOcean GmbH (2020). EnOcean. `https://www.enocean.com/en/technology/`. Last visited: 2022-10-06.

Esquivel-vargas, H., Caselli, M., and Peter, A. (2017). Automatic Deployment of Specification-based Intrusion Detection in the BACnet Protocol. *CPS-SPC@CCS*, pages 25–36.

Fatehah, M. (2018). Design and Process Metamodels for Modelling and Verification of Safety-Related Software Applications in Smart Building Systems. *ICIT 2018 Proceedings of the 6th International Conference on Information Technology: IoT and Smart City*, pages 60–64.

Fauri, D., Kapsalakis, M., Ricardo, D., Costante, E., Hartog, J. D., and Etalle, S. (2018). Leveraging Semantics for Actionable Intrusion Detection in Building Automation Systems. *13th International Conference on Critical Information Infrastructures Security (CRITIS)*, 1(700665):113–125.

Ficke, E., Schweitzer, K. M., Bateman, R. M., and Xu, S. (2019). Characterizing the Effectiveness of Network-Based Intrusion Detection Systems. *Proceedings - IEEE Military Communications Conference MILCOM*, 2019-October:76–81.

Finster, S. and Baumgart, I. (2015). Privacy-aware smart metering: A survey. *IEEE Communications Surveys and Tutorials*, 17(2):1088–1101.

Firth, S., Kane, T., Dimitriou, V., Hassan, T., Fouchal, F., Coleman, M., and Webb, L. (2017). REFIT Smart Home dataset. `https://repository.lboro.ac.uk/articles/dataset/REFIT_Smart_Home_dataset/2070091`. Last visited: 2022-10-06.

Fischer, R., Lamshöft, K., Dittmann, J., and Vielhauer, C. (2017a). Advanced Issues in Wireless Communication Security: Towards a Security-Demonstrator for Smart-Home Environments. *2017 International Carnahan Conference on Security Technology (ICCST).*

Fischer, T., Lesjak, C., Hoeller, A., and Steger, C. (2017b). Security for Building Automation with Hardware-Based Node Authentication. *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA).*

Franke, S., Hermann, A., Junghans, S., Leonhardt, S., Neumann, T., Teich, T., and Trommer, M. (2016). Event-Driven and District-Related Home Care.

Gai, A., Azam, S., Shanmugam, B., Jonkman, M., and Boer, F. D. (2018). Categorisation of security threats for smart home appliances. *2018 International Conference on Computer Communication and Informatics (ICCCI).*

Gao, X., Li, K., Chen, W., Hu, W., Zhang, Z., and Li, Q. (2020). Efficient and Privacy-Preserving Speaker Verification Scheme for Home Automation Devices. *Proceedings - 3rd International Conference on Multimedia Information Processing and Retrieval, MIPR 2020*, (1):237–240.

Garg, P. and Kohnfelder, L. (1999). STRIDE (security). `https://en.wikipedia.org/wiki/STRIDE_(security)`. Last visited: 2022-10-06.

Gasser, O., Scheitle, Q., Denis, C., Schricker, N., and Carle, G. (2017). Security Implications of Publicly Reachable Building Automation Systems. *2017 IEEE Security and Privacy Workshops (SPW).*

George, C. G., Tyranski, D. R., Simons, D. P., O'Quinn, J. D., York, E. R., and Salman, A. A. (2020). Integrating Social and Technical Solutions to Address Privacy in Smart Homes. *2020 Systems and Information Engineering Design Symposium, SIEDS 2020.*

Google (2016). Google Home Assistent. `https://assistant.google.com`. Last visited: 2022-10-06.

Google Inc. (2022). GoPacket project repository. `https://github.com/google/gopacket`. Last visited: 2022-10-06.

Goossens, M. (1998). The EIB System for Home & Building Electronics.

Gouveia, A. and Correia, M. (2016). Feature set tuning in statistical learning network intrusion detection. *Proceedings - 2016 IEEE 15th International Symposium on Network Computing and Applications, NCA 2016*, (1):68–75.

Granjal, J., Monteiro, E., and Sá Silva, J. (2015). Security for the Internet of Things: A Survey of Existing Protocols and Open Research Issues. *IEEE Communications Surveys Tutorials*, 17(3):1294–1312.

Granzer, W., Praus, F., and Kastner, W. (2010). Security in Building Automation Systems. *IEEE Transactions on Industrial Electronics*, 57(11):3622–3630.

Graveto, V., Cruz, T., and Simöes, P. (2022a). Security of building automation and control systems: Survey and future research directions. *Computers & Security*, 112:102527.

Graveto, V., Rosa, L., Cruz, T., and Simões, P. (2019). A stealth monitoring mechanism for cyber-physical systems. *International Journal of Critical Infrastructure Protection*, 24:126–143.

Graveto, V., Simões, P., and Cruz, T. (2022b). A dataset bundle for building automation and control systems security analysis. `https://dx.doi.org/10.21227/16a5-m134`. Last visited: 2022-10-06.

Graveto, V., Simões, P., and Cruz, T. (2022c). A dataset bundle for building automation and control systems security analysis. `https://github.com/vgraveto/knx-datasets`. GitHub repository. Last visited: 2022-10-06.

Groote, M. D., Volt, J., and Bean, F. (2017). *Is Europe Ready for the Smart Buildings Revolution ?* Buildings Performance Institute Europe.

Guri, M. (2018a). Beatcoin: Leaking private keys from air-gapped cryptocurrency wallets. *Proceedings - IEEE 2018 International Congress on Cybermatics: 2018 IEEE Conferences on Internet of Things, Green Computing and Communications, Cyber, Physical and Social Computing, Smart Data, Blockchain, Computer and Information Technology, iThings/GreenCom/CPSCom/SmartData/Blockchain/CIT 2018*, (June):1308–1316.

Guri, M. (2018b). Optical covert channel from air-gapped networks via remote orchestration of router/switch LEDs. *Proceedings - 2018 European Intelligence and Security Informatics Conference, EISIC 2018*, pages 54–60.

Guri, M. (2020). CD-LEAK: Leaking Secrets from Audioless Air-Gapped Computers Using Covert Acoustic Signals from CD/DVD Drives. *Proceedings - 2020 IEEE 44th Annual Computers, Software, and Applications Conference, COMPSAC 2020*, pages 808–816.

Guri, M., Bykhovsky, D., and Elovici, Y. (2017). aIR-Jumper: Covert Air-Gap Exfiltration/Infiltration via Security Cameras & Infrared (IR). `https://arxiv.org/abs/1709.05742`. Last visited: 2022-10-06.

Guri, M., Bykhovsky, D., and Elovici, Y. (2019a). Brightness: Leaking Sensitive Data from Air-Gapped Workstations via Screen Brightness. *2019 12th CMI Conference on Cybersecurity and Privacy, CMI 2019.*

Guri, M. and Elovici, Y. (2018). Bridgeware: The air-gap malware. *Commun. ACM*, 61(4):74–82.

Guri, M., Kedma, G., Kachlon, A., and Elovici, Y. (2014). Air hopper: Bridging the air-gap between isolated networks and mobile phones using radio frequencies. *9th IEEE International Conference on Malicious and Unwanted Software, MALCON 2014*, pages 58–67.

Guri, M., Monitz, M., and Elovici, Y. (2016a). Usbee: Air-gap covert-channel via electromagnetic emission from usb. *2016 14th Annual Conference on Privacy, Security and Trust (PST)*, pages 264–268.

Guri, M., Monitz, M., Mirski, Y., and Elovici, Y. (2015). BitWhisper: Covert Signaling Channel between Air-Gapped Computers Using Thermal Manipulations. *Proceedings of the Computer Security Foundations Workshop*, 2015-September:276–289.

Guri, M., Solewicz, Y., Daidakulov, A., and Elovici, Y. (2016b). Fansmitter: Acoustic data exfiltration from (speakerless) air-gapped computers. `https://arxiv.org/abs/1606.05915`. Last visited: 2022-10-06.

Guri, M., Solewicz, Y., and Elovici, Y. (2019b). MOSQUITO: Covert Ultrasonic Transmissions between Two Air-Gapped Computers Using Speaker-To-Speaker Communication. *DSC 2018 - 2018 IEEE Conference on Dependable and Secure Computing*.

Guri, M., Zadov, B., Bykhovsky, D., and Elovici, Y. (2019c). CTRL-ALT-LED: Leaking data from air-gapped computers via keyboard LEDs. *Proceedings - International Computer Software and Applications Conference*, 1:801–810.

Guri, M., Zadov, B., and Elovici, Y. (2020). ODINI: Escaping Sensitive Data from Faraday-Caged, Air-Gapped Computers via Magnetic Fields. *IEEE Transactions on Information Forensics and Security*, 15:1190–1203.

Hallak, G. and Bumiller, G. (2016). PLC for Home and Indystry Automation. In Lampe, L., Tonello, A. M., and Swart, T. G., editors, *Power Line Communications: Principles, Standards and Applications from Multimedia to Smart Grid*, chapter 7. John Wiley & Sons, Ltd.

Hallman, R., Bryan, J., Palavicini, G., Divita, J., and Romero-mariona, J. (2017). IoDDoS — The Internet of Distributed Denial of Service Attacks : A Case Study of the Mirai Malware and IoT-Based Botnets. *2nd International Conference on Internet of Things, Big Data and Security*, (November 2018).

Hamberger, C.; Eastman, C. (1964). Carl H. Hamberger & a. v. Clifford C. Eastman. `https://law.justia.com/cases/new-hampshire/supreme-court/1964/5258-0.html`. Last visited: 2022-10-06.

Han, S. H. I., Zhang, D., Lin, S., and Li, X. (2018). Systematically Ensuring the Confidence of Real-Time Home Automation IoT Systems. *ACM Transactions on Cyber-Physical Systems*, 2(3).

Handa, A., Sharma, A., and Shukla, S. K. (2019). Machine learning in cybersecurity : A review. *WIREs Data Mining and Knowledge Discovery*, (December 2018):1–7.

Hanspach, M. and Goetz, M. (2013). On covert acoustical mesh networks in air. *Journal of Communications*, 8(11):758–767.

Harirchi, F., Yong, S. Z., Arbor, A., and Royal, K. T. H. (2017). Active Model Discrimination with Active Model Applications to Fraud Detection in Smart Buildings. *IFAC-PapersOnLine*, 50(1):9527–9534.

Heartfield, R., Loukas, G., Budimir, S., Bezemskij, A., Fontaine, J. R. J., Filippoupolitis, A., and Roesch, E. (2018). A taxonomy of cyber-physical threats and impact in the smart home. *Computers & Security*, 78:398–428.

Hersent, O., Boswarthick, D., and Elloumi, O. (2012). Legacy M2M Protocols for Sensor Networks , Building Automation and Home Automation - The BACnet Protocol. In *The Internet of Things: Key Applications and Protocols*. Wiley.

Higgins, K. (2021). Lights Out: Cyberattacks Shut Down Building Automation Systems. `https://www.darkreading.com/attacks-breaches/lights-out-cyberattacks-shut-down-building-automation-systems`. Last visited: 2022-10-06.

Hilt, D. E. and Seegrist, D. W. (1977). *Ridge, a computer program for calculating ridge regression estimates*, volume 236. Dept. of Agriculture, Forest Service, Northeastern Forest Experiment Station,.

Hui, T. K. L., Sherratt, R. S., and Sánchez, D. D. (2017). Major requirements for building Smart Homes in Smart Cities based on Internet of Things technologies. *Future Generation Computer Systems*, 76:358–369.

Ilieva, S., Penchev, A., and Petrova-antonova, D. (2016). Internet of Things Framework for Smart Home Building. *International Conference on Digital Transformation and Global Society*, pages 450–462.

Initiative, E. (2019). EEBUS. `https://www.eebus.org/en/`. Last visited: 2022-10-06.

INTO-CPS Association (2020). INTO-CPS - Integrated Tool Chain for Model-based Design of Cyber-Physical Systems. `https://into-cps.org`. Last visited: 2022-10-06.

Iqbal, W., Abbas, H., Rauf, B., Abbas, Y., Amjad, F., and Hemani, A. (2021). PCSS: Privacy Preserving Communication Scheme for SDN Enabled Smart Homes. *IEEE Sensors Journal*, (c):1–13.

ISO IEC (2006). ISO/IEC 14543 - technology - Home electronic system (HES) architecture. `https://www.iso.org/standard/80934.html`. Last visited: 2022-10-06.

Iturbe, M., Garitano, I., Zurutuza, U., and Uribeetxeberria, R. (2017). Towards large-scale, heterogeneous anomaly detection systems in industrial networks: A survey of current trends. *Security and Communication Networks*, 2017.

Jia, M., Komeily, A., Wang, Y., and Srinivasan, R. S. (2019). Adopting Internet of Things for the development of smart buildings : A review of enabling technologies and applications. *Automation in Construction*, 101(February):111–126.

Jia, X., Li, X., and Gao, Y. (2017). A Novel Semi-Automatic Vulnerability Detection System for Smart Home. *the International Conference*, pages 195–199.

Jones, C. B., Carter, C., and Thomas, Z. (2018). Intrusion Detection & Response using an Unsupervised Artificial Neural Network on a Single Board Computer for Building Control Resilience. *2018 Resilience Week (RWS)*, (Section II):31–37.

Kaaz, K. J., Hoffer, A., Saeidi, M., Sarma, A., and Bobba, R. B. (2017). Understanding user perceptions of privacy, and configuration challenges in home automation. *Proceedings of IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC*, 2017-October:297–301.

Katipamula, S., Haack, J., Hernandez, G., Akyol, B., and Hagerman, J. (2016). Volttron: An open-source software platform of the future. *IEEE Electrification Magazine*, 4:15–22.

Keliris, A., Salehghaffari, H., Cairl, B., Krishnamurthy, P., Maniatakos, M., and Khorrami, F. (2016). Machine learning-based defense against process-aware attacks on industrial control systems. In *2016 IEEE International Test Conference (ITC)*, pages 1–10. IEEE.

Khedekar, D. C., Oteyza, D. A., Truco, A. C., and Huertas, G. F. (2016). Home Automation — A Fast - Expanding Market.

KNX Association (2004). XML Data Encoding. Technical report, KNX Association.

KNX Association (2009). KNX Architecture. Technical report, KNX Association.

KNX Association (2020a). KNX. `https://www.knx.org`. Last visited: 2022-10-06.

KNX Association (2020b). KNX ETS5 eCampus. `https://wbt5.knx.org/?lang=en`. Last visited: 2022-10-06.

KNX Association (2020c). KNX Secure. `https://www.knx.org/knx-en/for-professionals/benefits/knx-secure/`. Last visited: 2022-10-06.

KNX Association (2020d). The legacy of KNX. `https://www.knx.org/knx-en/for-professionals/What-is-KNX/KNX-History/index.php`. Last visited: 2022-10-06.

KNX Association (2022). What is ETS professional? `https://www.knx.org/knx-en/for-professionals/software/ets-professional/`. Last visited: 2022-10-06.

Komninos, N., Philippou, E., Pitsillides, A., and Member, S. (2014). Survey in Smart Grid and Smart Home Security : Issues , Challenges and Countermeasures. *IEEE Communications Surveys & Tutorials*, 16(4):1933–1954.

Kraemer, M. J. and Flechais, I. (2018). Researching privacy in smart homes: A roadmap of future directions and research methods. *IET Conference Publications*, 2018(CP740):1–10.

Krebs, B. (2014). Target Hackers Broke in Via HVAC Company. `https://krebsonsecurity.com/2014/02/target-hackers-broke-in-via-hvac-company/`. Last visited: 2022-10-06.

Krishnan, S., Anjana, M. S., and Rao, S. N. (2017). Security Considerations for IoT in Smart Buildings. *2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC).*

Langner, R. (2013). To kill a centrifuge: A technical analysis of what stuxnet's creators tried to achieve. `https://www.langner.com/wp-content/uploads/2017/03/to-kill-a-centrifuge.pdf`. Last visited: 2022-10-06.

Legrand, A., Niepceron, B., Cournier, A., and Trannois, H. (2018). Study of Autoencoder Neural Networks for Anomaly Detection in Connected Buildings. *2018 IEEE Global Conference on Internet of Things (GCIoT).*

Lei, X., Tu, G.-h., Liu, A. X., Li, C.-y., and Xie, T. (2018). The Insecurity of Home Digital Voice Assistants – Vulnerabilities , Attacks and Countermeasures. *2018 IEEE Conference on Communications and Network Security (CNS).*

Levy, H. P. (2015). Gartner Predicts Our Digital Future. `http://goo.gl/3AyTvo`. Last visited: 2022-10-06.

Li, Y. (2018). Design of Smart Home Cloud Server. *2018 IEEE International Conference of Safety Produce Informatization (IICSPI)*, pages 200–203.

Li, Y., B, Y. W., and Zhang, Y. (2018). SecHome : A Secure Large-Scale Smart Home System Using Hierarchical Identity. *International Conference on Information and Communications Security*, 1:339–351.

Lilis, G., Conus, G., Asadi, N., and Kayal, M. (2017). Towards the next generation of intelligent building : An assessment study of current automation and future IoT based systems with a proposal for transitional design. *Sustainable Cities and Society*, 28:473–481.

Lin, V. Z. and Parkin, S. (2020). Transferability of privacy-related behaviours to shared smart home assistant devices. *2020 7th International Conference on Internet of Things: Systems, Management and Security, IOTSMS 2020.*

Ling, Z., Luo, J., Xu, Y., Gao, C., Wu, K., Member, S., and Fu, X. (2017). Security Vulnerabilities of Internet of Things : A Case Study of the Smart Plug System. *IEEE Internet of Things Journal*, 4(6):1899–1909.

Liu, Y., Hu, S., Wu, J., Shi, Y., Jin, Y., Hu, Y., and Li, X. (2015). Impact Assessment of Net Metering on Smart Home Cyberattack Detection. *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC).*

Liu, Y., Pang, Z., Lan, D., and Gong, S. (2018). A Taxonomy for the Security Assessment of IP-Based Building Automation Systems : The Case of Thread. *IEEE Transactions on Industrial Informatics*, 14(9):4113–4123.

Lobaccaro, G., Carlucci, S., and Lofstrom, E. (2016). A Review of Systems and Technologies for Smart Homes and Smart Grids.

Loughry, J. and Umphress, D. A. (2002). Information Leakage from Optical Emanations. *ACM Transactions on Information and System Security*, 5(3):262–289.

Macaulay, T. and Singer, B. (2011). *Cybersecurity for Industrial Control Systems*. CRC Press.

Mace, J. C., Morisset, C., Pierce, K., Gamble, C., Maple, C., and Fitzgerald, J. (2018). A multi-modelling based approach to assessing the security of smart buildings. *IET Conference Publications*, 2018(CP740):1–10.

Matthew Peacock, Michael N. Johnstone, C. V. (2018). An Exploration of Some Security Issues Within the BACnet Protocol. *International Conference on Information Systems Security and Privacy*, pages 252–272.

Matyunin, N., Szefer, J., Biedermann, S., and Katzenbeisser, S. (2016). Covert channels using mobile device's magnetic field sensors. *Proceedings of the Asia and South Pacific Design Automation Conference, ASP-DAC*, 25-28-January-2016:525–532.

Meyer, D., Haase, J., Eckert, M., and Klauer, B. (2017). New attack vectors for building automation and IoT. *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society.*

Minoli, D., Sohraby, K., and Occhiogrosso, B. (2017). IoT Considerations , Requirements , and Architectures for Smart Buildings — Energy Building Management Systems. *IEEE Internet of Things Journal*, 4(1):269–283.

Mirsky, Y., Guri, M., and Elovici, Y. (2017). Hvacker: Bridging the air-gap by attacking the air conditioning system. https://arxiv.org/abs/1703.10454. Last visited: 2022-10-06.

Mitchell, R. and Chen, I.-R. (2013). A Survey of Intrusion Detection Techniques for Cyber Physical Systems. *ACM Computing Surveys*, 46(4):55.

Mocrii, D., Chen, Y., and Musilek, P. (2018). Internet of Things IoT-based smart homes : A review of system architecture , software , communications , privacy and security. *Internet of Things*, 1-2:81–98.

MODICON (1996). Modicon Modbus Protocol Reference Guide. http://www.modbus.org/docs/PI_MBUS_300.pdf. Last visited: 2022-10-06.

Molina, J. (2014). Learn How to Control Every Room at a Luxury Hotel Remotly: The Dangers of Insecure Home. https://pdfs.semanticscholar.org/047f/7d8626e1e2183c1aed2417b498330c7b033a.pdf. Last visited: 2022-10-06.

Molina, J. (2015). VIDEO: Learn how to control every room at a luxury hotel remotely. https://www.youtube.com/watch?v=RX-O4XuCW1Y. Last visited: 2022-10-06.

Mundt, T., Kruger, F., and Wollenberg, T. (2012). Who refuses to wash hands? privacy issues in modern house installation networks. *Proceedings - 2012 7th International Conference on Broadband, Wireless Computing, Communication and Applications, BWCCA 2012*, pages 271–277.

Mundt, T. and Wickboldt, P. (2016). Security in building automation systems - A first analysis. *2016 International Conference On Cyber Security And Protection Of Digital Services (Cyber Security)*, pages 1–8.

Nassi, B., Shamir, A., and Elovici, Y. (2017). Oops!...i think i scanned a malware. https://arxiv.org/abs/1703.07751. Last visited: 2022-10-06.

National Counterintelligence and Security Center (2021). Technical Specifications for Construction and Management of Sensitive Compartmented Information Facilities - IC Tech Spec - for ICD/ICS 705 - Version 1.5.1. https://www.dni.gov/files/NCSC/documents/Regulations/IC_Technical_Specifications_for_Construction_and_Management_of_Sensitive_Compartmented_Information_Facilities_v151_PDF.pdf. Last visited: 2022-10-06.

Naz, M. T. and Zeki, A. M. (2020). A Review of Various Attack Methods on Air-Gapped Systems. *2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies, 3ICT 2020*.

Nazir, S., Patel, S., and Patel, D. (2017). Assessing and augmenting scada cyber security: A survey of techniques. *Computers & Security*, 70:436–454.

Ng, J. and Keoh, S. L. (2018). SEABASS : Symmetric-keychain Encryption and Authentication for Building Automation Systems. *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, pages 219–224.

Nicklas, J.-p., Mamrot, M., Winzer, P., Lichte, D., Marchlewitz, S., and Wolf, K.-d. (2016). Use Case based Approach for an Integrated Consideration of Safety and Security Aspects for Smart Home Applications. *2016 11th System of Systems Engineering Conference (SoSE)*.

Novak, T. and Gerstinger, A. (2010). Safety- and Security-Critical Services in Building Automation and Control Systems. *IEEE Transactions on Industrial Electronics*, 57(11):3614–3621.

Office of the Director of National Intelligence, USA (2010). Intelligence Community Directive 705 - Physical and Technical Security Standards for Sensitive Compartmented Information Facilities. https://www.dni.gov/files/NCSC/documents/Regulations/ICS-705-1.pdf. Last visited: 2022-10-06.

Ogen, R., Shwartz, O., Zvi, K., and Oren, Y. (2018). Sensorless, permissionless information exfiltration with wi-fi micro-jamming. In *Proceedings of the 12th*

*USENIX Conference on Offensive Technologies*, WOOT'18, page 7, USA. USENIX Association.

Open Information Security Foundation (2020). Suricata IDS project home page. `https://suricata.io/`. Last visited: 2022-10-06.

Pan, Z., Hariri, S., and Hall, K. (2014). Anomaly Based Intrusion Detection for Building Automation and Control Networks Youssif Al-Nashif. *2014 IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA)*, pages 72–77.

Pan, Z., Hariri, S., and Pacheco, J. (2019). Context aware intrusion detection for building automation systems. *Computers & Security*, 85:181–201.

Pan, Z., Pacheco, J., and Hariri, S. (2016). Anomaly Behavior Analysis for Building Automation Systems. *2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA)*.

Pathmabandu, C., Grundy, J., Chhetri, M. B., and Baig, Z. (2020). An Informed Consent Model for Managing the Privacy Paradox in Smart Buildings. *Proceedings - 2020 35th IEEE/ACM International Conference on Automated Software Engineering Workshops, ASEW 2020*, pages 19–26.

Patil, A., Kamuni, V., Sheikh, A., Wagh, S., and Singh, N. (2019). A machine learning approach to distinguish faults and cyberattacks in smart buildings. In *2019 9th International Conference on Power and Energy Systems (ICPES)*, pages 1–6. IEEE.

Paxson, V. (2020). The Zeek Network Security Monitor. `https://zeek.org`. Last visited: 2022-10-06.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Pedro, J. and Silva, S. (2007). Aplicação de Interface Com Sistema Domótico EIB Engenharia Informática e de Computadores. *Master tesis.*

Peterson, E. (2019). Mirai Nikki: The Future of DDoS.

Pham, C. T. and Mansson, D. (2019). A study on realistic energy storage systems for the privacy of smart meter readings of residential users. *IEEE Access*, 7:150262–150270.

Phillips, B., Gamess, E., and Krishnaprasad, S. (2020). An evaluation of machine learning-based anomaly detection in a scada system using the modbus protocol. In *Proceedings of the 2020 ACM Southeast Conference*, pages 188–196.

Praus, F. and Kastner, W. (2014). Identifying unsecured building automation installations. *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*, pages 1–4.

Praus, F., Kastner, W., and Palensky, P. (2016). Software Security Requirements in Building Automation. *2010IEEE Transactions on Industrial Electronics.*

Qiu, T., Member, S., Chen, N., Li, K., Member, S., Atiquzzaman, M., Member, S., Zhao, W., and Member, S. (2018). How Can Heterogeneous Internet of Things Build Our Future : A Survey. *IEEE Communications Surveys & Tutorials*, 20(3):2011–2027.

Ramapatruni, S., Narayanan, S. N., Mittal, S., Joshi, A., and Joshi, K. (2019). Anomaly Detection Models for Smart Home Security. *Proceedings - 2019 IEEE 5th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*, pages 19–24.

Raspberry PI Foundation (2020). Raspberry Pi. `https://www.raspberrypi.org`. Last visited: 2022-10-06.

Rath, A. T. (2017). Strengthening Access Control in case of Compromised Accounts in Smart Home. *2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 1–8.

Rathinavel, K., Pipattanasomporn, M., Kuzlu, M., and Rahman, S. (2017). Security Concers and Countermeasures in IoT-Integrated Smart Buildings. *2017 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT).*

Ray, A. K. (2017). Study of Smart Home Communication Protocol ' s and security & privacy Aspects. *2017 7th International Conference on Communication Systems and Network Technologies (CSNT).*

Rehman, S. and Gruhn, V. (2018). An Approach to Secure Smart Homes in Cyber- Physical Systems / Internet-of-Things. *2018 Fifth International Conference on Software Defined Systems (SDS)*, pages 126–129.

Renato Nunes (2016). DomoBus. `https://fenix.tecnico.ulisboa.pt/downloadFile/1126295043835997/extended.pdf`. Last visited: 2022-10-06.

Robinson, D. and Kim, C. (2017). A cyber-defensive industrial control system with redundancy and intrusion detection. *2017 North American Power Symposium, NAPS 2017.*

Robles-Durazno, A., Moradpoor, N., Mcwhinnie, J., and Russell, G. (2019). WaterLeakage: A Stealthy Malware for Data Exfiltration on Industrial Control Systems Using Visual Channels. *IEEE International Conference on Control and Automation, ICCA*, 2019-July:724–731.

Rosa, L., Cruz, T., de Freitas, M. B., Quitério, P., Henriques, J., Caldeira, F., Monteiro, E., and Simões, P. (2021). Intrusion and anomaly detection for the next-generation of industrial automation and control systems. *Future Generation Computer Systems*, 119:50–67.

Rosa, L., Proença, J., Henriques, J., Graveto, V., Cruz, T., Simões, P., Caldeira, F., and Monteiro, E. (2017). An evolved security architecture for distributed industrial automation and control systems. In *European Conference on Cyber Warfare and Security*, pages 380–390. Academic Conferences International Limited.

Samarah, S., Al Zamil, M. G., Aleroud, A. F., Rawashdeh, M., Alhamid, M. F., and Alamri, A. (2017). An Efficient Activity Recognition Framework: Toward Privacy-Sensitive Health Data Sensing. *IEEE Access*, 5:3848–3859.

Santo, H., Maekawa, T., and Matsushita, Y. (2017). Device-free and privacy preserving indoor positioning using infrared retro-reflection imaging. *2017 IEEE International Conference on Pervasive Computing and Communications, PerCom 2017*, pages 141–152.

Santos, L. (2018). Intrusion Detection Systems in Internet of Things A literature review. *2018 13th Iberian Conference on Information Systems and Technologies (CISTI).*

Sarbhai, A., Merwe, J. V. D., and Kasera, S. (2019). Privacy-Aware Peak Load Reduction in Smart Homes. *2019 11th International Conference on Communication Systems and Networks, COMSNETS 2019*, 2061:312–319.

Saxena, U., Sidhi, J. S., and Singh, Y. (2017). Analysis of security attacks in a smart home networks. *2017 7th International Conference on Cloud Computing, Data Science & Engineering - Confluence*, pages 431–436.

Seifried, S. and Kastner, W. (2017). KNX IPv6 : Design Issues and Proposed Architecture. *2017 IEEE 13th International Workshop on Factory Communication Systems (WFCS).*

Serrenho, T. and Bertoldi, P. (2019). *Smart home and appliances : State of the art.* Publications Office of the European Union, Luxembourg.

Shuai, M., Yu, N., Wang, H., and Xiong, L. (2019). Anonymous authentication scheme for smart home environment with provable security. *Computers & Security*, 86:132–146.

Silva, A. (2016). *Desenvolvimento de uma Infraestrutura Eletrónica de Comunicação para o Controlo Remoto de "Casas Inteligentes" Usando KNX.* Msc thesis, University of Coimbra, DEEC.

Sutherland, I., Spyridopoulos, T., Read, H., and Jones, A. (2015). Applying the ACPO Guidelines to Building Automation Systems. *Proceedings of the Third International Conference on Human Aspects of Information Security, Privacy, and Trust*, 1:684–692.

Tenkanen, T. and Hamalainen, T. (2017). Security Assessment of a Distributed , Modbus-based Building Automation System. *2017 IEEE International Conference on Computer and Information Technology (CIT)*.

The Tcpdump Group (2020). Tcpdump & Libpcap. `https://www.tcpdump.org`. Last visited: 2022-10-06.

Thread Group (2019). Thread Certified Products. `https://www.threadgroup.org/what-Is-thread`. Last visited: 2022-10-06.

Tienteu, M., Mason, A., Talley, M., White, T., Ahovi, E., Hamilton, D., Kornegay, K., Reece, M., and Thompsonl, W. (2017). Data Exfiltration using Building Automation to Bridge Air Gapped System. *IEEE International Symposium on Hardware Oriented Security and Trust (HOST) 2017*.

Toschi, G. M., Campos, L. B., and Cugnasca, C. E. (2017). Home automation networks : A survey. *Computer Standards & Interfaces*, 50(September 2016):42–54.

Usman, M., Ali, I., Khan, S., and Khurram, M. (2019). Journal of Network and Computer Applications A survey on software defined networking enabled smart buildings : Architecture , challenges and use cases. *Journal of Network and Computer Applications*, 137(November 2018):62–77.

Valli, C., Johnstone, M. N., Peacock, M., and Jones, A. (2017). BACnet - Bridging the Cyber Physical Divide One HVAC at a Time. *2017 9th IEEE-GCC Conference and Exhibition (GCCCE)*.

Vanus, J. (2018). ScienceDirect of Home Implementation Design of Home Implementation Design of Home Implementation Design of Home Implementation within IoT with Natural Language within IoT with Natural Language Design of Home Implementation within IoT with Natural Language. *IFAC-PapersOnLine*, 51(6):174–179.

Vasyutynskyy, V., Ploennigs, J., and Kabitzsch, K. (2006). *Multi-Agent System for Monitoring of Building Automation Systems*, volume 40. IFAC.

Vikram, N., Harish, K. S., Nihaal, M. S., Umesh, R., Aashik, S., and Kumar, A. (2017). A Low Cost Home Automation System Using Wi-Fi Based Wireless Sensor Network Incorporating Internet of Things ( IoT ). *2017 IEEE 7th International Advance Computing Conference (IACC)*, 100.

Wang, X., Habeeb, R., Ou, X., Amaravadi, S., Hatcliff, J., Mizuno, M., Neilsen, M., Rajagopalan, S. R., and Varadarajan, S. (2017). Enhanced Security of Building Automation Systems Through Microkernel-Based Controller Platforms. *2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW)*, pages 37–44.

Wang, X., Neilsen, M., Rajagopalan, S. R., Baldwin, W. G., and Phillips, B. (2015). Secure RTOS Architecture for Building Automation Categories and Subject Descriptors. *CPS-SPC@CCS*, pages 79–90.

Wendzel, S., Kahler, B., and Rist, T. (2012). Covert channels and their prevention in building automation protocols: A prototype exemplified using BACnet. In *2012 IEEE International Conference on Green Computing and Communications*, pages 731–736.

Wendzel, S., Tonejc, J., Kaur, J., and Kobekova, A. (2017). *Cyber Security of Smart Buildings*, chapter 16, pages 327–351. John Wiley & Sons, Ltd.

Werner, S., Pallas, F., and Bermbach, D. (2018). Designing Suitable Acess Control for Web-Connected Smart Home Platforms. *International Conference on Service-Oriented Computing*, pages 240–251.

Wright, R. (2019). FBI: How we stopped the Mirai botnet attacks. `https://searchsecurity.techtarget.com/news/252459016/FBI-How-we-stopped-the-Mirai-botnet-attacks`. Last visited: 2022-10-06.

Wu, J., Liu, J., Hu, X. S., and Shi, Y. (2016). Privacy protection via appliance scheduling in smart homes. *IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers, ICCAD*, 07-10-November-2016.

Xu, K., Wang, F., and Jia, X. (2016). Secure the Internet , one home at a time. *Security and Communication Networks*, (July):3821–3832.

Xu, Z. and Agung Julius, A. (2019). Robust Temporal Logic Inference for Provably Correct Fault Detection and Privacy Preservation of Switched Systems. *IEEE Systems Journal*, 13(3):3010–3021.

Zarpelão, B. B., Miani, R. S., Kawakani, C. T., and De, S. C. (2017). A survey of intrusion detection in Internet of Things. *Journal of Network and Computer Applications*, 84(September 2016):25–37.

Zeng, E., Mare, S., Roesner, F., Clara, S., Zeng, E., Mare, S., and Roesner, F. (2017). End User Security and Privacy Concerns with Smart Homes This paper is included in the Proceedings of the End User Security & Privacy Concerns with Smart Homes. *Thirteenth Symposium on Usable Privacy and Security (SOUPS)*, (Soups):65–80.

ZeroMQ (2020). ZeroMQ- An open-source universal messaging library. `https://zeromq.org`. Last visited: 2022-10-06.

Zetter, K. (2013). Researchers Hack Building Control System at Google Australia Office.

Zheng, Z. and Reddy, A. L. N. (2017). Safeguarding Building Automation Networks : THE-Driven Anomaly Detector Based on Traffic Analysis. *2017 26th International Conference on Computer Communication and Networks (ICCCN)*.

Zhibo, P., Bag, G., Ngai, E., and Leung, V. (2017). [ Invited Paper ] Native IP Connectivity for Sensors and Actuators in Home Area Network. *Smart Grid Inspired Future Technologies*, 2:222–231.