MASTER

Cyber-physical Reconnaissance in Smart Buildings

Szakállas, Gergely

Award date:
2022

Department of Mathematics and Computer Science
Security Research Group

Cyber-physical Reconnaissance in Smart Buildings

Master's Thesis

Gergely Szakállas

Supervisors:
Jerry den Hartog
Martin Rosso
Emmanuele Zambon

Eindhoven, October 2022

Abstract

Building automation systems facilitate the operation and management of buildings, increase their energy efficiency, and ensure the comfort of their occupants, by constantly detecting and responding to changes in the physical environment caused by natural phenomena and human activities. The wealth of data collected by building automation systems may therefore be sensitive and could leak information about buildings and their occupants, which poses a threat to the security of buildings and the privacy of individuals.

Our multi-stage approach seeks to gather this type of information about a smart building, through observations made in the cyber and physical domains focusing on its building automation system and interior, respectively, in an effort to discover rooms in the building, and identify the devices of the building automation system located in them. As a result of relating devices to rooms, it becomes possible to determine when and where certain human activities occur in the building, which has an impact on both security and privacy.

In our experiments, we have managed to learn the physical locations and dimensions of rooms in Atlas, a state-of-the-art smart building at Eindhoven University of Technology, housing hundreds of people, including university staff and students. By observing and correlating the measurements of occupancy sensors in the cyber domain with changes in illumination in the physical domain, we were able to associate occupancy sensors with physical locations in the building, and thereby infer the movement patterns of the building's occupants from occupancy measurements.

The results of our experiments show that the proposed approach could allow an attacker to gain knowledge about the building automation process of a smart building, and use the data collected by its building automation system to infer possibly confidential information about its interior, as well as carry out privacy attacks against its occupants. On that account, we discuss a number of prevention and mitigation strategies, and highlight future directions to improve our methods and address the limitations of the approach.

Contents

List of Figures

List of Tables

Listings

Chapter 1

Introduction

1.1 Background

As the world is slowly transitioning into a greener and more ecological future, individuals and organisations have been trying to reduce greenhouse emissions by reducing energy usage and moving away from fossil fuels in favour of renewable energy sources, such as solar and wind. This trend has been coinciding with the advent of the Internet of Things (IoT), a paradigm that aims to make commodity electronics *smart* by connecting them to the Internet, or some other communication network, which in turn allows for control, integration, automation, and data collection at an unprecedented speed, scope, and scale. It has also given rise to building automation systems (BAS), which utilise networked devices at scale to increase the energy efficiency, reduce the operating costs, enhance the security, and facilitate the management and operation of buildings, as well as improve the comfort of their occupants.

A typical building automation system consists of myriads of devices (e.g. lights, door locks, occupancy sensors, temperature sensors, etc.) and management components (e.g. controllers, gateways, switches, etc.) that constitute and communicate over the building automation system network, using a variety of technologies and communication protocols. Devices work in a coordinated manner to optimise the operation of the building and improve the comfort of its occupants, by detecting and responding to changes in the physical environment (e.g. drop in temperature) and physical events (e.g. someone entering a room), with respect to the configuration of the building automation system. By contrast, management components are responsible for integrating devices into the building automation system, and coordinating their behaviour and communication across its different parts and layers. In an example scenario, a person walks into an empty room, which is detected by an occupancy sensor, causing it to notify the room controller, which in response turns the lights on and activates the ventilation. In terms of control and communication flows, components typically form hierarchies, in which devices are situated at the bottom, directly interacting with and influencing the physical environment, whereas controllers take place above them, managing them and ensuring the flow of information through the building automation system, with the help of gateways and switches that hold the different parts and layers of the network together. At the top, there may also be operator workstations that provide an overview of the entire building automation system and let the operators of the building monitor, manage and control its various parts and components. Similarly to the architecture of the network, in the physical environment, devices are deployed throughout the interior of the building, typically in plain sight, while management components tend to be hidden in the walls and ceilings of the building. If there are any operator workstations, they are usually located in designated areas that are only accessible to the operators.

Modern building automation systems are built on top of robust proven protocols, such as BACnet, LonWorks, and KNX, which can easily be integrated into the internet protocol suite (TCP/IP) for better connectivity and improved interoperability[19][28]. For that reason, building automation system networks are often IP-based, using the same underlying protocols and technologies as general purpose computer networks. That said, IP protocols are typically only used in the backbone infrastructure, connecting the management components and operator workstations together[18], as devices are usually connected to the building automation system network through controllers, which manage and control them using low-level protocols (e.g. FOX) that are often proprietary and vendor-specific[30]. Regarding communication, most protocols implement the request-response and publish-subscribe patterns for information exchange, allowing components to read and write the information of other components synchronously, or be notified of the information of other components asynchronously. As such, these interactions typically occur between operator workstations and controllers, and between controllers and devices, where operator workstations read and write the information of controllers, which read and write the information of devices, realising the flow of information through the building automation system.

Most building automation protocols represent components as stateful resources that are uniquely identified, addressable, and discoverable, and have a collection of properties, whose values we refer to as *state*. These properties store primitive data values (e.g. integers, booleans, strings, etc.) that typically relate to sensor measurements, status information, or configuration options, which can be read and written by other resources. In BACnet, which is the most widely used communication protocol in building automation systems[9], components are known as *devices*, whose information is modeled in terms of *objects*, which have a collection of *properties*. Devices are uniquely identified within the BACnet network and can have an arbitrary number of objects, which are uniquely identified within the device and can have an arbitrary number of properties, which are uniquely identified within the object and store data directly, however, the types of objects and properties are confined, and certain object types may have required properties that must always be defined[25]. The standard defines 65 object types (e.g. `analog-value`, `binary-value`, etc.) and 13 property types (e.g. `boolean`, `double`, etc.), which can be used to represent virtually any kind of resource or information that may be of interest to other devices in the network[11]. When it comes to the exchange of information, a device may read and write the values of object properties on another device using `readProperty` and `writeProperty` requests, or subscribe to the value changes of object properties using `subscribeCOV` requests, upon which it will receive notifications of the value changes of the respective object properties[25].

1.2 Motivation

Building automation systems affect the physical environment by controlling lighting, temperature, air quality, and other physical properties. In general terms, they share this ability with industrial control systems (ICS) and other cyber-physical systems (CPS), which are frequently used in industrial domains to control and manage facilities, such as factories and power plants. Cyber-physical systems are often targeted by cybercriminals, especially statesponsored attack groups, who aim at disabling, damaging, or hijacking them in order to sabotage economic and societal processes, or make money[15][8]. Such attacks can cause an enormous amount of damage to the target, with wide-ranging ramifications in both the cyber and the physical environment. Similarly, attacks on building automation systems can also impact the physical environment, albeit to a lesser extent, with less severe consequences, which might be one of the reasons why they have not been so prevalent[31]. That said, as building automation systems continue to evolve and become more widespread, they provide new attack surfaces and opportunities to malicious actors, which is expected to lead to a rise in the frequency and sophistication of attacks, driving the need to secure them against potential threats. Even though an attack carried out against a building does not have as much of an impact as an attack on a power plant, building automation systems are much easier to come by, more standardised, and typically less secured, which make them both easier and more appealing to attack, especially at scale, which in turn drastically increases the risks and the overall impact of a potential attack[31].

Depending on the comprehensiveness of the building automation system and the capabilities of the attacker in question, potential threats to the building and its occupants can range from annoying and petty, to obstructive, dangerous, and paramount. In the simplest scenario, the building automation system only supervises lights, so the attacker can only hack the lighting infrastructure, in which case the worst thing that can happen is that the occupants cannot see due to the lack of illumination, or the building wastes energy by keeping the lights on at all times. By contrast, in a more advanced scenario, the building automation system manages most of the building's infrastructure, in which case an attack could have serious ramifications for individuals and assets alike. For instance, access control mechanisms, like door locks, could go out of commission, allowing the occupants to enter restricted areas, or on the contrary, restricting their movement and impeding their productivity. Additionally, the HVAC (heating, ventilation, and air conditioning) system might stop working, failing to regulate temperatures in specific areas, like server rooms or cold rooms. In most cases, the biggest concern is financial damage, but in more sensitive environments, like hospitals, tampering with the building automation process can put the occupants in danger, for instance, by keeping chemical substances at the wrong temperatures, or not disinfecting personnel and equipment properly. Nevertheless, disruption and sabotage are not the only ways in which a cyberattack can cause harm to a building and its occupants.

Building automation systems typically operate by following a set of predefined rules and routines, which describe what state the building should be in at all times. Since the physical environment is constantly changing as a result of natural phenomena (e.g. weather) and human presence and activities (e.g. walking), the building automation system has to continuously monitor and adapt to it, so that it can maintain the desired state of the building. As such, the building automation system is constantly collecting data about the physical environment, which may relate to human activities as well. Consequently, having access to the data of the building automation system, and knowing the physical locations of its devices could theoretically allow an attacker to determine what human activities occur when and where in the building, which could pose a threat to the privacy of the occupants and undermine the security of the building. For example, there could be a security guard, whose responsibilities include opening the building for the day, closing it for the night, and performing inspections during the day. If the attacker could learn when the security guard performs these tasks and what their movement patterns are, that would expose and weaken the security policies of the building, and make it more vulnerable to intrusion, theft, etc.

Nonetheless, gaining knowledge about a building automation system and locating its devices is neither easy nor straightforward, which may prompt us to look for alternative, potentially faster and easier approaches to obtain this type of information. On a fundamental level, we can distinguish between three types of approaches.

The easiest and fastest way to learn more about a smart building is to ask its operators for the resources and documentation they have on its building automation system, however, they are unlikely to hand them over, as they are typically considered confidential or otherwise sensitive information. By means of social engineering, we could mask our identity and pretend to be a trusted party, asking for these documents for legal or auditing purposes, which would be more likely to succeed, however, it would also be illegitimate, as it would constitute fraud in most jurisdictions of the world[37]. On the one hand, the information obtained this way may contain knowledge that cannot be extracted or inferred, but on the other hand, it could be outdated, incomplete, or inaccurate, and might not yield as much information about the physical environment as our approach.

A more offensive and sophisticated approach would be to obtain this information from the building automation system by means of hacking, in particular, by exploiting security vulnerabilities in its components. Depending on the target, a successful attack may take hours, days, weeks, or months to execute, or may not succeed at all, making it not only illegal, but also highly unreliable[22]. Additionally, the obtained information might not exceed in quality or quantity what we could capture using a packet sniffer, or acquire using the previous approach, therefore there is not much added benefit compared to less sophisticated approaches.

The third alternative is based on open-source intelligence (OSINT), which concerns the collection of data from various publicly available sources. Carrying out open-source intelligence is about as difficult and time consuming as social engineering, yet it is completely legal, and very likely to return useful information[26]. Nevertheless, information in public sources is unlikely to be as detailed as the information possessed by the operators, and it will not be as empirical or extensive as the information obtained by our approach, therefore it cannot be exclusively used for the same intents and purposes, however, it can complement any of the other approaches[4].

Our approach, which we refer to as cyber-physical reconnaissance, draws inspiration from anomaly and intrusion detection methods developed for cyber-physical systems, as discussed in Chapter 2, and uses a combination of the aforementioned approaches, while also providing two key benefits. Firstly, it gains knowledge from empirical data and real-world observations rather than ad-hoc information collected and published by other parties, ensuring its actuality and recency, and secondly, it considers a building automation system not as an abstract networked system, but as a distributed cyber-physical system that is an integral part of its host building, which puts its devices and the data collected by them in the context of the real-world, giving them additional meaning that can be extracted and utilised in an effort to learn more about the building and its occupants.

Chapter 2

Related work

The security and privacy aspects of IoT devices, building automation systems, industrial control systems, and cyber-physical systems in general, have continuously been studied in the past two decades, however, with the spread and commoditisation of these systems, there has been an increasing amount of research into their drawbacks and potential vulnerabilities, as well as the attacks targeting them, from a variety of angles.

In their works, Alisic, Charyyev, Acar, and Qin showcase various methods of inferring valuable information about humans and devices, by means of capturing and analysing data. Alisic et al. used air quality measurements to infer occupancy, whereas the other authors utilised network fingerprinting techniques for the purpose of identifying devices and individuals in smart home environments, or extracting sensitive information about critical infrastructure. More specifically, Alisic et al. alleged that an attacker can use temperature or humidity measurements in a room to infer its state of occupancy, and they verified their theory on a testbed. To mitigate the privacy leak, they recommended adding Gaussian noise to the measurements in order to ensure the privacy of the occupants, but warned that adding too much noise can hinder the essential functions of the respective sensors[2]. By contrast, Charyyev et al. captured network traffic to detect misactivations in smart speakers and identify their users in a smart home environment. They generated fingerprints from the network traffic and fed them to various machine learning models based on locality-sensitive hashing, which attained over 90% accuracy on misactivation detection and over 70% accuracy on user identification. The authors also considered limitations with respect to the number of users, computational complexity, and variations in wake words, but concluded that these would not be relevant or significant in most home environments[10]. A similar strategy was followed by Acar et al., who developed a multi-stage privacy attack capable of identifying devices and their states, as well as user activities in a smart home environment, based solely on network traffic. Their machine learning based approach achieved over 90% accuracy on a dataset containing real measurements from popular off-the-shelf IoT devices, while it is mostly automated, device type and protocol agnostic, impervious to encryption, and does not require expert knowledge, or details about the environment. That said, the proposed attack is not immune to countermeasures and can easily be disrupted by injecting noise into the network traffic[1]. Lastly, Qin et al. investigated the exposure of sensitive information in the management and diagnostic network traffic of a SCADA system responsible for the distribution of natural gas. They employed network fingerprinting techniques to reconstruct the commands and interactions of the operators over the local network, which allowed them to learn the physical layout of the distribution system, and the properties of the gas flowing through it, without any existing background knowledge. The authors considered encryption as the primary mitigation strategy, but cautioned against its use due to its adverse effects on compatibility, monitorability, complexity, and latency[27].

The works of Wüstrich and Bernieri highlight the importance of considering both the cyber and the physical domain when dealing with cyber-physical systems, by demonstrating the benefits of relying on features from both domains, as well as the risks of observing only one domain. The approach of cyber-physical reconnaissance takes inspiration from anomaly and intrusion detection systems, like the one developed by Wüstrich et al., which monitor both domains in order to accurately and effectively identify malfunctions and potential attacks. Wüstrich et al. proposed an anomaly detection method for industrial control systems, that uses sound measurements in addition to network traffic to spot anomalies faster and more accurately, than methods relying on observations from only one domain. They also discussed a number of challenges around processing and correlating measurements, and highlighted difficulties around observing the physical environment, which complicate the viability of their approach in the real world[36]. Meanwhile, the study of Bernieri et al. attests the vulnerabilities of a fault diagnosis system that does not monitor the physical domain, allowing an attacker to carry out availability and integrity attacks over the network, triggering false alarms, and misleading the operator into following up with unnecessary and erroneous countermeasures. Since there were no actual faults in the physical environment, the system would have been able to detect these attacks, had it also monitored the physical domain[5].

The combination of domains and information sources is useful not only for anomaly and intrusion detection, but also for reconnaissance, as shown by Lee et al., who associated occupancy sensors with individuals by combining occupancy data with auxiliary information, and Keliris et al., who leveraged information from various publicly available resources to plan a cyberattack on critical infrastructure. Fundamentally, both approaches rely on open source intelligence (OSINT), but to different degrees. Lee et al. used auxiliary information obtained through OSINT to derive a correspondence between the occupancy sensors and occupants of a building, enabling them to surveil specific individuals, and demonstrating the possibility of turning sensor measurements into personally identifiable information (PII). The authors considered three mitigation strategies against the proposed attack, none of which were found to fully protect against it without also significantly impairing the utility of the sensors[23]. At the same time, Keliris et al. examined the impact of publicly available information on critical infrastructure security by using exclusively OSINT techniques to plan a cyberattack. They used publicly available resources, such as power system databases and public reports, to construct the model of an electric power system, identify its critical operation points, and find attack vectors. Possible prevention methods include security practices and measures on the infrastructure side, while anonymisation and randomisation techniques can help mitigate risks on the information side[20]. In addition, privacy issues around data linkage have been investigated more broadly by Zheng at al., who looked at various data sources in IoT systems, how data from these sources can be linked, and how linked data may reveal the private information of individuals. They also considered how linked data may be accessed and utilised, and what potential threats, questions, and challenges there might be with respect to them[38].

Privacy attacks targeting the occupants of a building have been taken a step further by Kessler and Wang, who found various methods to track the movements of individuals using sensor measurements. Kessler et al. employed machine learning on vibration data for this purpose, while Wang et al. fed occupancy measurements to a stochastic model to achieve the same goal, however, both approaches were found to struggle with multiple targets and noisy measurements, which caused their performance to plummet. Kessler et al. provided an overview of methods capable of tracking people inside a building, and demonstrated that vibrations can reveal much more information about a building's occupants, than some other methods. Their results show, that given multiple vibration sensors deployed at different locations in a building, it is possible to not only track the movement of an individual, but also predict their gender, based on the attenuation of the vibration signals and the signature of their footsteps, respectively. The authors also discussed a privacy-preserving strategy for gender classification, which aims at obscuring the footstep signatures of the occupants by aggregating measurements using a time window[21]. Wang et al. found that one can use room-level occupancy measurements in a smart building to reconstruct the location traces of its occupants, at an accuracy comparable to that of video surveillance, given a sufficiently high sensor density and measurement frequency with respect to the number of occupants. The authors also demonstrated that the accuracy and reliability of the tracking, and thereby the viability of the approach, are heavily dependent on these factors, and as such, the sensing interval can be adjusted in accordance with the number of occupants and the desired trade-off between utility and privacy in any smart building[35].

The approaches and techniques mentioned above explore how sensor data in a smart building can be used to derive information about its processes and occupants, given some existing knowledge, such as the locations of the sensors, but it is not discussed how that knowledge can be learned or acquired. Similarly, the literature does not consider the aspects of combining features from multiple domains, especially in non-critical environments, and the opportunities it creates for attackers and defenders alike. In buildings in particular, this will be of increasing importance in the future, caused by the emergence of IoT and the spread of building automation systems, affecting the work and lives of more people in more industries than ever before. For example, physical observations could help an attacker identify devices, spot glitches, find attack vectors, or gather information about a building and its occupants, that a building automation system cannot provide, and there are a lot of possibilities enabled by the combination of features from multiple domains and the combination of information from various sources. The approach of cyber-physical reconnaissance takes advantage of these techniques, giving it an upper hand over approaches that do not observe multiple domains, or only do so for redundancy or validation purposes.

Chapter 3

Methodology

Our multi-stage approach, which we call cyber-physical reconnaissance, aims to learn as much about a smart building and its occupants as possible, from observations made in the cyber and physical domains, and information obtained through open-source intelligence. It consists of three stages, each of which is explained in further detail below, alongside the prerequisites that have to be met and the assumptions that have to hold. In the first stage, the building's automation system is examined in the cyber and physical domains, in an effort to discover its components, and understand its concepts, behaviour, and characteristics. In the second stage, the combination of observations from the cyber and physical domains allows one to relate the representations of devices to physical locations, while in the third and final stage, this knowledge is utilised to infer when and where certain physical events occur in the building, without having to observe the physical environment.

3.1 Prerequisites and assumptions

There are a few prerequisites that must be met by any smart building and its building automation system to become a target, in both the cyber and the physical domain.

In the cyber domain, the state changes of the devices of the building automation system must be communicated over the network, such that they are observable in the network traffic, which must be accessible to the attacker. The packets in the captured network traffic that are relevant to these state changes must contain a timestamp, source and destination addresses, and protocol-specific information that identifies the device and conveys its state or state change.

In the physical domain, the devices of the building automation system must have a detectable or measurable effect on the environment that the attacker can capture by observing the physical domain. The captured observations must contain a timestamp, and information about the environment that identifies its location and conveys its properties.

3.1.1 Attacker model

The attacker model considers somebody who seeks to gather information about a smart building and its occupants, and is able to observe the building, as well as read the network traffic of its building automation system.

In the cyber domain, the attacker must be able to gain access to the network of the building automation system and read the traffic going between its components. In a typical smart building environment, neither of these expectations is unreasonable. When it comes to accessing the network, the building automation system may share the network with other devices, like smartphones and laptops, or it may be completely isolated from other networks, either physically or logically. General purpose networks often employ weak or outdated security measures that are easy to bypass, allowing the attacker to directly access the network, however, even if the network is adequately secured or not directly accessible, the attacker could still access it indirectly by hijacking a device that is already connected to it. This is not unlikely, considering the inadequate security protections typically found in IoT devices, which make them more vulnerable than other networked devices[34][33]. Moreover, building automation systems typically comprise myriads of such devices, which all serve as attack vectors, and thereby increase the likelihood of a successful intrusion[31]. As for reading the network traffic, building automation protocols, such as BACnet, often forgo encryption and openly communicate the properties of resources over the network, while protocols used in industrial control systems, like Modbus, typically do not, due to their more specialised and rudimentary nature.

In the physical domain, the attacker must be able to detect or measure the properties of the physical environment, such as its brightness or temperature, and observe various human activities, such as people walking around. For the purpose of making observations, the attacker could use their own senses, or a variety of instruments (e.g. cameras, sound recorders, etc.), which could be fixed at specific locations inside or outside the building, depending on the level of physical accessibility. If the building is publicly accessible, with no authorisation or credentials required, then the attacker is free to observe the building from the inside, otherwise it has to be observed from the outside, for instance, through the windows.

3.2 Understanding the cyber-physical process

The first stage of the approach concerns the technical aspects of the building automation system and its ability to perceive the physical environment. Before we can take advantage of a building automation system, we have to understand how it works, to which end we need to figure out what types of components it comprises, how they are represented, and how they interact with each other.

We learn these details by capturing the network traffic of the building automation system, inspecting its traffic flows, and analysing its packets using a network analyser. Optionally, information obtained through open-source intelligence can supplement the information extracted from the network traffic, and help us learn more about the components of the building automation system.

Once we have established the concepts and characteristics of the building automation system at hand, we need to learn how its devices perceive physical events in the building. When devices detect changes in the physical environment, they change their state, which is observable in the cyber domain, allowing us to correlate physical events with state changes.

We find these correlations by observing the inside of the building while capturing the network traffic of the building automation system, and correlating the observed physical events with the state changes that have been captured around the same time.

Following these actions, we expect to know what types of components the building automation system comprises, how these components are represented, how they interact and communicate with each other, and how they perceive certain physical events in the building.

3.3 Constructing the cyber-physical mapping

The devices of the building automation system not only detect changes in the physical environment, but also induce them, in an effort to maintain the desired state of the building. When devices induce changes in the physical environment, they change its properties, typically in response to a change that has been detected, which are observable in the cyber and physical domains, allowing us to associate state changes, and thereby the representations of devices, with physical locations.

We obtain these associations by capturing the network traffic of the building automation system and observing the inside of the building simultaneously, correlating the observed changes in physical properties with the state changes that have been captured around the same time, and associating the representations of the respective devices with the respective physical locations. For each association, we also indicate the confidence $c \in [0, 1]$, denoting the proportion of correlated changes in physical properties, i.e. the number of changes in physical properties that have been correlated with state changes, divided by the number of changes in physical properties.

From these associations, we can construct a cyber-physical mapping, more specifically a *quasi-surjection*, that associates the representations of devices with physical locations, such that each representation relates to one or more locations, with varying degrees of confidence.

3.4 Using the cyber-physical mapping

Once we have established how the devices of the building automation system perceive certain physical events, and what physical locations their representations are associated with, we can use that knowledge to infer when and where certain physical events occur in the building, without having to observe the physical environment.

We make these inferences by capturing the network traffic of the building automation system, and relating the captured state changes to the physical events and locations, that have been correlated with them in the first stage and associated with the representations of their respective devices in the second stage, respectively.

From these inferences, we can reconstruct a timeline of events that describes when and where certain physical events occurred in the building, such that each event relates to one or more locations, with varying degrees of confidence.

Chapter 4

Implementation

In order to demonstrate the feasibility of the approach, discover its limitations, and determine possible mitigation strategies, we need to implement it in a real-world environment, as a proof of concept. We have chosen this environment to be the Atlas building of Eindhoven University of Technology, located in Eindhoven, The Netherlands, for three key reasons. Firstly, the Atlas building has recently been outfitted with a state-of-the-art building automation system[14][29], by means of which it has become one of the world's most sustainable buildings[17], as well as an excellent testing ground for our research. Secondly, as the main building of the institution, Atlas is used extensively all year round, seeing hundreds of visitors each day, which makes it not only a realistic but also a challenging and consequential target environment. Lastly, our research is conducted at Eindhoven University of Technology, which provides simple means to get in contact with the operators of the building, which facilitates validation and evaluation, and promotes discussion.

In the following sections, we describe one of the many possible implementations of our approach, and evaluate it through four experiments that seek to achieve the goals and objectives laid out in Chapter 3. In each section, we also motivate our strategy, mention the assumptions we have made, and discuss the problems and challenges we have encountered while executing the experiments.

4.1 Understanding the cyber-physical process

The first stage of cyber-physical reconnaissance focuses on exploring the inner workings of the building automation system, which involves discovering and identifying its components, and understanding its concepts, behaviour, and characteristics. We view the Atlas building as a collection of physically separated areas, which we refer to as rooms, and we are interested in learning what types of devices there are in a typical room, and how they detect and respond to physical events.

For that reason, we capture and analyse the network traffic of the building automation system, focusing on the traffic over BACnet. Following the extraction of MAC addresses and BACnet devices, objects, and properties, we group the packets by the components involved, which gives us insight into their communication patterns, and analyse the packets to extract as much useful information about the components as possible.

In order to find correlations between the behaviour of the building automation system and the real world, we observe the rooms of the building while capturing the network traffic of the building automation system, and examine how the devices of the building automation system detect and respond to typical human activities (e.g. walking around, standing up, etc.) in both the cyber and the physical domain.

4.1.1 Experiments

The first objective of the stage requires us to capture and interpret the network traffic of the building automation system, as a means to discover its components and understand their behaviour, whereas the second objective of the stage requires us to observe the rooms of the building, with the goal of perceiving physical events independently from the building automation system. According to the attacker model in Section 3.1, the attacker must be able to gain access to the network of the building automation system, and read the traffic going between its components, as well as observe various human activities occurring in the building.

Given these abilities, we describe the steps of an experiment aiming to achieve these objectives and demonstrate the feasibility of the approach:

1. When the building is open, start manually observing the rooms of the building, and capturing the network traffic of the building automation system using a packet analyser
2. While observing the rooms, note down and timestamp any physical events of interest occurring in them
3. Before the building closes, stop observing the rooms and capturing the network traffic
4. Analyse the network traffic around the times the physical events occurred
5. Go over the network traffic, and extract as much information about the components of the building automation system as possible

We carried out the experiment in the evening hours of a business day, when most rooms of the building are typically unoccupied, allowing us to use them for the purposes of our research.

In the first and second steps of the experiment, we went inside the building and had one of the occupants perform typical human activities (e.g. walking, sitting, standing, etc.) in an arbitrary subset of the rooms, which we observed, timestamped, and noted down in a text file, while capturing the network traffic of the building automation system.

For the purpose of data collection in the physical domain, we used an Android smartphone to note down and timestamp the events of interest that occurred in the rooms, which we observed using our own senses. Due to time constraints, we did not observe every room in the building, instead, we decided to focus on an arbitrary subset of the rooms, in order to get familiar with the common patterns of behaviour and communication.

In the cyber domain, we used a computer to capture the network traffic of the building automation system. This was done in collaboration with the operators, who mirrored the IP-based network traffic of the building automation system to a remote SPAN port, where we could capture it. Using this method, we have captured all packets on the building automation system network that were transmitted over UDP, but focused on those packets that were communicated over BACnet, since that was the building automation protocol used by the building automation system. We captured the traffic and wrote it to PCAP files using tcpdump, which we filtered and exported to various text formats for further processing and analysis using tshark. The commands responsible for capturing, filtering, and exporting the network traffic are given in Listings 4.1, 4.2, and 4.3, respectively.

```
tcpdump -i cisco_erspan -n -w atlas-%Y-%m-%d_%H.%M.%S_UTC+2.pcap not tcp
```

Listing 4.1: tcpdump command capturing network traffic

```
tshark -r ./traffic.pcap -Y 'bacapp.string_character_set' -T json --no-duplicate-
    keys
```

Listing 4.2: tshark command filtering and exporting network traffic to JSON

```
tshark -r ./traffic.pcap -Y 'bacapp.type == 0 or bacapp.type == 1' -T fields -e
    frame.time_epoch -e eth.src_resolved -e eth.dst_resolved -e bacapp.
    confirmed_service -e bacapp.unconfirmed_service -e bacapp.instance_number -e
    bacapp.objectType -e bacapp.present_value.enum_index -e bacapp.present_value.
    uint -e bacapp.present_value.real -Eseparator=, -Equote=d
```

Listing 4.3: tshark command filtering and exporting network traffic to CSV

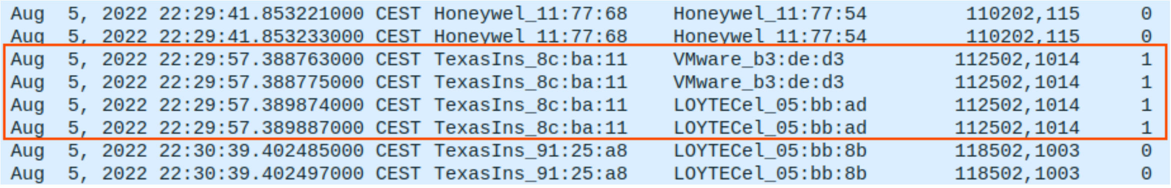
While capturing and subsequently processing the network traffic, we only faced two challenges that we had not expected, which were not inherent to the approach, but likely specific to our environment, thus they may or may not be present elsewhere.

The first challenge was noise, which was twofold and only partially expected. There was a lot of traffic in the network traces that was not relevant to our research, either because the packets were transmitted over protocols different from BACnet, or because they carried information that was not relevant to the task. Owing to our capturing strategy, packets of the former kind were expected to be included in the network traces and could easily be filtered out, while packets of the latter kind required more sophistication to deal with, as we could only filter them out gradually, in an iterative process, using information we have learned along the way as we explored the inner workings of the building automation system.

The second challenge concerned inconsistencies in the behaviour of the building automation system. Every so often, BACnet devices communicated the same information multiple times, while on other occasions, they communicated incompatible information about their objects or properties. For example, they sometimes reported state changes even when there had been no change of state, or returned different types of values for the same properties, respectively. The first kind is essentially noise that can simply be ignored or filtered out, but the second kind is harder to counter. Naively, we could consider whatever information returned most recently or frequently to be the truth, but in general, we could not decide what the truth was.

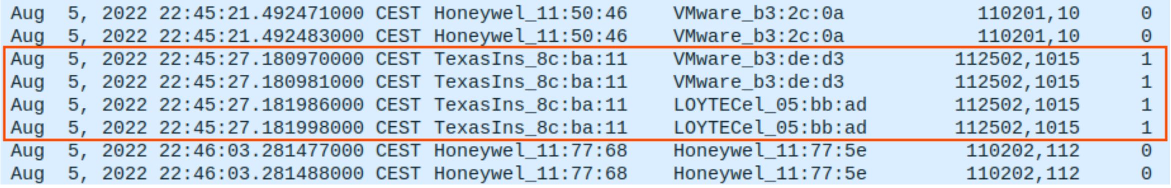
In the fourth step of the experiment, we analysed the network traffic in Wireshark while referencing the data collected in the second step to explore how the building automation system detected and responded to the observed physical events. Based on our findings, the rooms of the building are equipped with occupancy sensors, which are represented by objects on some BACnet devices. They switch between two distinct states, 0 and 1, depending on the presence or absence of motion nearby. More explicitly, the state is 0, indicating that the room is unoccupied, if there has recently been no motion detected, while the state is 1, indicating that the room is occupied, if there has recently been motion detected. Additionally, each sensor has its own timer, and does not report any state changes occurring within 15 minutes of the detection of any motion, meaning that once the state changes from 0 to 1, no further changes are reported for at least 15 minutes, and any motion detected within this duration resets the timer, i.e. the time from which this duration is measured. As such, if no motion is detected for 15 minutes, the state changes from 1 to 0, and remains 0 until motion is detected again.

No BACnet device or object, other than the sensor itself, seemed to respond to changes in occupancy, as seen in Figures 4.1 and 4.2, yet lights always acted in accordance with occupancy measurements, which we could only observe in the physical domain, since the state changes of lights were not communicated over the network, or at least, over BACnet. Whenever the state of an occupancy sensor changed from 0 to 1, lights immediately turned on nearby, and whenever it changed from 1 to 0, the same lights turned off, albeit with a delay of 20 to 50 seconds. The delay was constant for each room, but varied across the rooms of different floors, so depending on the room, it could take up to a minute for the lights to turn off, following a reported change in the occupancy of the room.



```
Aug  5, 2022 22:29:41.853221000 CEST Honeywel_11:77:68    Honeywel_11:77:54        110202,115     0
Aug  5, 2022 22:29:41.853233000 CEST Honeywel_11:77:68    Honeywel_11:77:54        110202,115     0
Aug  5, 2022 22:29:57.388763000 CEST TexasIns_8c:ba:11    VMware_b3:de:d3         112502,1014     1
Aug  5, 2022 22:29:57.388775000 CEST TexasIns_8c:ba:11    VMware_b3:de:d3         112502,1014     1
Aug  5, 2022 22:29:57.389874000 CEST TexasIns_8c:ba:11    LOYTECel_05:bb:ad       112502,1014     1
Aug  5, 2022 22:29:57.389887000 CEST TexasIns_8c:ba:11    LOYTECel_05:bb:ad       112502,1014     1
Aug  5, 2022 22:30:39.402485000 CEST TexasIns_91:25:a8    LOYTECel_05:bb:8b       118502,1003     0
Aug  5, 2022 22:30:39.402497000 CEST TexasIns_91:25:a8    LOYTECel_05:bb:8b       118502,1003     0
```

Figure 4.1: Example of an occupancy sensor reporting a change in occupancy



```
Aug  5, 2022 22:45:21.492471000 CEST Honeywel_11:50:46    VMware_b3:2c:0a           110201,10     0
Aug  5, 2022 22:45:21.492483000 CEST Honeywel_11:50:46    VMware_b3:2c:0a           110201,10     0
Aug  5, 2022 22:45:27.180970000 CEST TexasIns_8c:ba:11    VMware_b3:de:d3         112502,1015     1
Aug  5, 2022 22:45:27.180981000 CEST TexasIns_8c:ba:11    VMware_b3:de:d3         112502,1015     1
Aug  5, 2022 22:45:27.181986000 CEST TexasIns_8c:ba:11    LOYTECel_05:bb:ad       112502,1015     1
Aug  5, 2022 22:45:27.181998000 CEST TexasIns_8c:ba:11    LOYTECel_05:bb:ad       112502,1015     1
Aug  5, 2022 22:46:03.281477000 CEST Honeywel_11:77:68    Honeywel_11:77:5e        110202,112     0
Aug  5, 2022 22:46:03.281488000 CEST Honeywel_11:77:68    Honeywel_11:77:5e        110202,112     0
```

Figure 4.2: Another example of an occupancy sensor reporting a change in occupancy

In the fifth step of the experiment, we exported the captured network traffic to JSON and CSV using the commands given above, from which we extracted information about the components using a Python script. Resolving the MAC addresses of the packets that were communicated between the components of the building automation system allowed us to analyse their relationships and communication patterns, whereas from BACnet messages of types `readProperty` and `readPropertyMultiple`, we could gather information on a multitude of BACnet devices and objects, including what properties they had, what their values were, and what these values meant, from which we were able to infer their respective vendors, names, types, purposes, and capabilities. Unfortunately, it was not possible to derive meaningful information about all components, therefore we could only guess the types, roles, and responsibilities of some of them.

In Chapter 5, we discuss the results and findings of the experiment further, while the source code of the processing algorithm and the data we have extracted using it are available in the project's repository on GitLab.

4.2 Constructing the cyber-physical mapping

The second stage of the approach concerns the physical locations of the devices of the building automation system. Since we view the Atlas building as a collection of rooms, we are interested in learning which devices of the building automation system are located in which rooms, however, we have no existing knowledge about the rooms, so we have to discover them first. As such, the stage can be split into two parts, where the first part focuses on discovering rooms in the building and determining their physical locations and boundaries, while the second part aims at finding the BACnet objects that correspond to devices in these rooms.

In the first part, we utilise the window grids on the sides of the building to denote physical locations, and rely on changes in illumination observed in the physical domain to discover rooms and learn their locations and boundaries. In the second part, we correlate the changes in illumination observed in the physical domain with the state changes of BACnet objects captured in the cyber domain to associate BACnet objects with the discovered rooms, and thereby determine their physical locations.

4.2.1 Experiments

The Atlas building has the shape of a rectangular cuboid, whose sides (lateral faces) are made up of grids of glass panels, serving as windows to the rooms inside. The high degree of transparency provided by these windows allows us to gain knowledge about the interior of the building by looking through them from the outside. Theoretically, this lets us see where people and walls are, observe changes in physical properties, and perceive other visual cues relating to the behaviour of the building automation system. In practice, it is very difficult, if not impossible, to observe the entire building at such a high resolution, especially in a scalable manner, so we will make a number of assumptions, omissions, and simplifications.

For the sake of simplicity, we will assume that all rooms are rectangle or rectangular cuboid shaped and separated by walls, meaning that all walls are either parallel or perpendicular to each side of the building, and all ceilings are parallel to the bases of the building. These are reasonable assumptions for office buildings, however, they may not always hold in the case of university buildings, as there are likely areas, such as lecture halls or laboratories, which are shaped differently. Furthermore, since the sides of the building are made up of grids of glass panels, we will also assume that all walls and ceilings run along grid lines, i.e. along the edges of these glass panels. Unless it is very bright inside the building, the windows are uncovered, and there are no obstacles to visibility (e.g. bad weather), it is not feasible to make out walls inside the building, especially if they are transparent, and there is no way to look inside rooms that have no windows to the outside. As a result, we will omit depth as a dimension of a room, and ignore all rooms that have no windows on any side of the building.

As a consequence of these decisions, every glass panel acts as a window to exactly one room, and every room has at least one window on any side of the building, thus every room can be denoted by a set of (adjacent) windows on each side of the building. A room may have windows on adjacent sides, if it is located in the corner of a floor, or on opposite sides, if it has no walls parallel to those sides. Regarding geometric considerations, this allows a building with a length of L, width of W, and height of H (as measured in glass panels), to have at most H floors, with at most $L \times W$ rooms on each floor, however, $(L-2) \times (W-2)$ rooms would not be located around the edges and therefore would have no windows to the outside, leaving us with at most $L \times W - (L-2) \times (W-2)$ rooms on each floor.

The northern and southern sides of the Atlas building are fully covered by glass panels of uniform size and shape, arranged in a grid of size 10 × 16. The western and eastern sides are covered by the same glass panels, but what would be a single grid on each side, is instead split into two grids of sizes 10 × 33 and 10 × 93 by visually distinct staircases that let the occupants move between floors, which need not be observed. Still, they split both sides into two parts along the north-south axis, which needs to be considered when denoting windows, and thereby rooms. Figure 4.3 shows the southern side of the building, whereas the eastern side can be seen on Figure 4.4, depicting the grid layouts, as well as the visually distinct staircase splitting the grid into two parts.

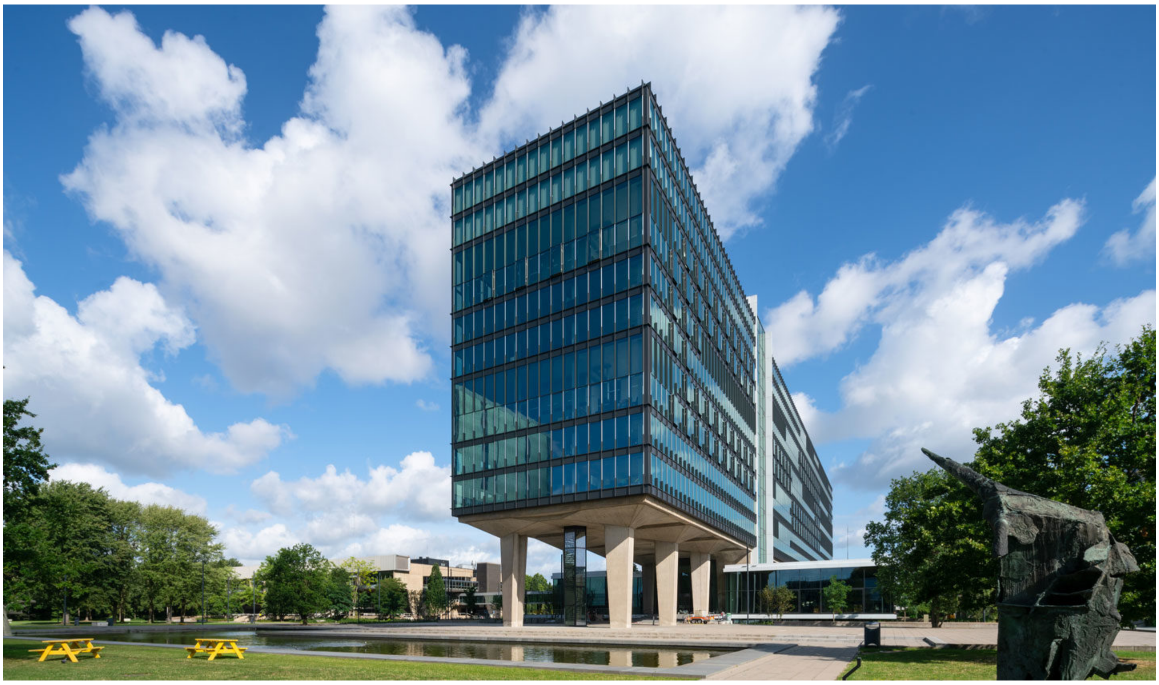


Figure 4.3: The south-eastern side of the Atlas building[3]

Since the majority of rooms face either west or east, we have ignored the rooms facing north or south, meaning that we have only considered those rooms which were visible on the western or the eastern side of the building. In order to identify rooms by their windows, we had to come up with a notation that allowed us to refer to windows clearly and unambiguously on either side of the building. The building has two staircases, which split the western and eastern sides of the building into two parts along the north-south axis, such that each part is fully covered by a grid. Because of this, we have decided to consider windows from the perspective of these grids, and denote their coordinates with respect to them. To that end, we have denoted the two sides of the building by W (western) and E (eastern), and the two grids on each side by N (northern) and S (southern), with the coordinates $x \in [0..9]$ for rows and $y \in [0..32]$ or $y \in [0..92]$ for columns, depending on the side and grid. On the western side, for instance, windows in the northern grid could go from $N_{0,0}$ to $N_{9,32}$, while in the southern grid, they could go from $S_{0,0}$ to $S_{9,92}$.

Regarding the coordinates, as rows likely correspond to floors, it seemed intuitive to have 0 denote the bottom row and 9 the top row, but it was less evident how to enumerate the columns. The intervals could start from either the northernmost or the southernmost column

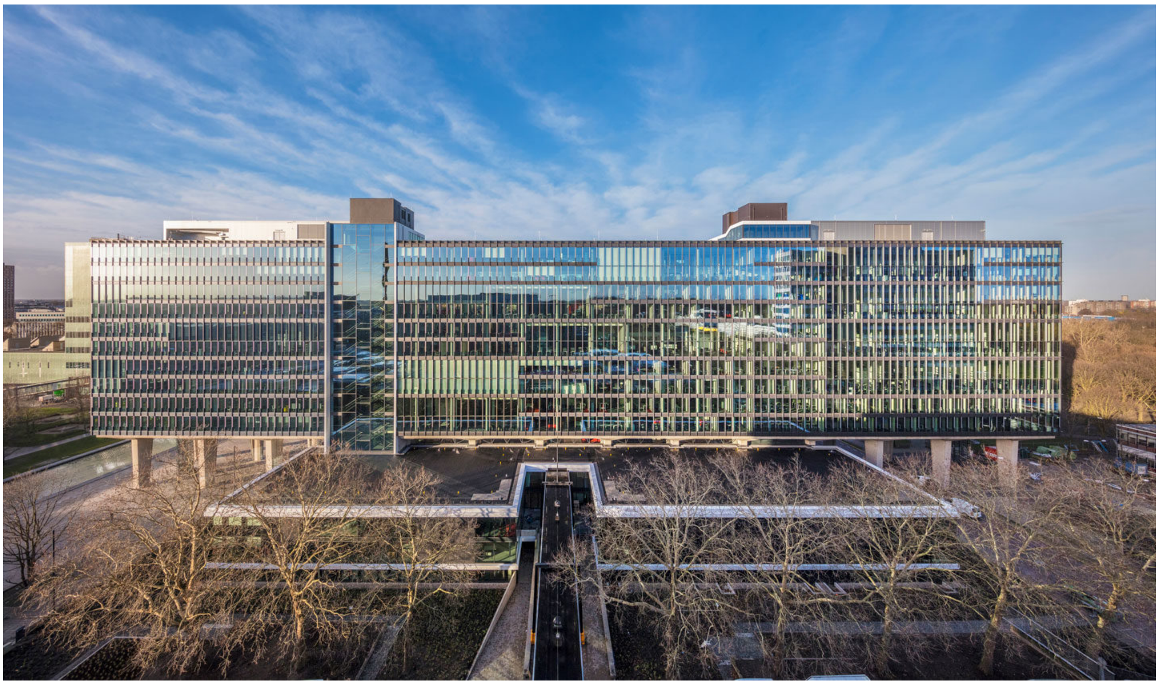


Figure 4.4: The eastern side of the Atlas building[3]

in each grid, however, in order to avoid any (jump) discontinuities in the y coordinates across the staircases, i.e. along the north-south axis, we have decided to regard the staircases as axes, from which column intervals would start in each grid, with 0 denoting the column closest to the staircase. For reference, Tables 4.1 and 4.2 show how this notation works on the western and eastern sides of the building, respectively.

$N_{9,32}$	$\cdots$	$N_{9,0}$	Staircase	$S_{9,0}$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$S_{9,92}$
$\vdots$		$\vdots$		$\vdots$								$\vdots$
$N_{0,32}$	$\cdots$	$N_{0,0}$		$S_{0,0}$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$S_{0,92}$

Table 4.1: Notation of windows on the western side of Atlas

$S_{9,32}$	$\cdots$	$S_{9,0}$	Staircase	$N_{9,0}$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$N_{9,92}$
$\vdots$		$\vdots$		$\vdots$								$\vdots$
$S_{0,32}$	$\cdots$	$S_{0,0}$		$N_{0,0}$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$N_{0,92}$

Table 4.2: Notation of windows on the eastern side of Atlas

Rooms can be denoted by sets of windows, which constitute subgrids of the window grids on each side of the building, assuming that all rooms are rectangular cuboid shaped. This means that we can refer to rooms by their windows with the lowest and highest coordinates in each window grid. For example, the 2×3 subgrid shown in Table 4.3 could be referred to as $X_{2,1}^{3,3}$, if the 5×5 grid was denoted by X, and analogously, a room in the northern part of the western side of the building could be referred to as $WN_{5,7}^{6,10}$. Additionally, if a room has windows on multiple sides of the building, then it can have multiple notations, and if the context makes it clear which side of the building is being considered, then the notations W and E can be omitted.

	■	■	■	
	■	■	■	

Table 4.3: Example of a subgrid

The two parts of the stage are implemented in the form of two experiments, with each experiment focusing on each part. As such, in the first experiment, we discover some rooms in the building, then in the second experiment, we find the BACnet objects that correspond to physical devices in the discovered rooms.

In the first experiment, we need to devise a method to learn the locations and boundaries of rooms, which is analogous to deciding which adjacent windows belong to different rooms, given the assumptions and simplifications established above. Intuitively, one way to go about this is to consider lighting. Humans typically turn the lights on when they enter a room, and turn them off once they leave, in case there is not enough light coming in through the windows, skylights, and other openings[16]. As a consequence, lights are well suited to indicate occupancy, but they can also help tell rooms apart. Due to how light works, if an artificial light source turns on or off, it affects the brightness of its physical environment. In practice, this means that rooms will get noticeably brighter or darker when a light source turns on or off in them. From the perspective of windows, this means that the windows, through which the brightness of the physical environment appears to change to a similar extent at the same time, should belong to the same room.

Rooms that have no windows are not directly discoverable this way, however, if they are encompassed by rooms that are discoverable, then they may be indirectly discoverable, and if any such rooms have been purposefully built to be hidden, then the approach also has an impact on physical security and confidentiality. As such, it is possible that not all rooms can be discovered, however, given enough time and a sufficiently large number of observations, it is possible to determine the locations and boundaries of all rooms that are directly or indirectly discoverable.

In order to leverage lights for the purpose of room discovery, there are three additional assumptions that must hold in the target environment.

Firstly, there is an assumption that lights turn on and off. If there is sufficient light coming in from the outside, then artificial light sources are unlikely to be on, while in some environments, they are constantly on (e.g. for security reasons), meaning that there will be no changes in brightness to observe or gain knowledge from. Additionally, if the building has no occupants (e.g. because it is closed), then lights are also not expected to turn on or off, except throughout multiple areas of the building, in response to natural phenomena (e.g. sunset).

The second assumption is that changes in brightness are perceptible when lights turn on or off. If the windows are covered, or visibility is reduced by natural phenomena (e.g. bad weather), then it might not be possible to perceive changes in brightness caused by artificial light sources. Additionally, not all rooms may be uniformly lit, and some rooms may have fewer or more lights, or different types of lights, than other rooms, so changes in brightness might not always be perceptible.

Finally, the third assumption is that lights do not turn on or off in multiple rooms at the same time. If they always do, then the building automation system is unlikely to make a distinction between the rooms, meaning that we can treat them as one room, and if they only sometimes do, then we can detect these instances and resolve them to the constituent rooms.

Taking these into consideration, we describe the steps of the first experiment aiming to discover rooms in the building:

1. When the building is open, start manually observing the windows of the building from the outside
2. While observing the windows, note down and timestamp any changes in illumination, as well as the sets of windows through which they are perceived
3. After the building closes, stop observing the windows
4. Go over the window sets, and identify then resolve any conflicts among them
5. Regard each remaining window set as a room of the building

It would be difficult for a human to painstakingly observe the windows of the building for a long period of time, and take note of changes in illumination. Thankfully, this process can be made far more efficient and accurate by observing the windows through photos or videos taken by cameras, which can later be inspected and processed. As such, we update the first three steps of the experiment as follows:

1. When the building is open, start observing the windows of the building from the outside using cameras
2. After the building closes, stop observing the windows
3. Go over the images, find those that recorded changes in illumination, and note down their timestamps, as well as the sets of windows through which the changes were perceived

Once we have managed to discover as many rooms as possible, we try to find the BACnet objects that correspond to physical devices in these rooms.

In the simplest and most straightforward situation, the state changes of lights are observable in the network traffic of the building automation system, allowing us to observe the windows of the building and capture the network traffic of the building automation system at the same, correlate the state changes of lights with the perceived changes in illumination, and associate the respective BACnet objects with the respective rooms.

In more complex situations, the state changes of lights are not observable in the network traffic, only through the windows of the building, however, we can use lights as indicators of occupancy[16], and assume that other devices, whose state changes are observable in the network traffic, also respond to changes in occupancy, allowing us to observe the windows of the building and capture the network traffic of the building automation system at the same, correlate the state changes of those devices with the inferred changes in occupancy, and associate the respective BACnet objects with the respective rooms.

Bearing that in mind, we describe the steps of the second experiment aiming to associate BACnet objects with the discovered rooms:

1. When the building is open, start observing the windows of the building from the outside using cameras and capturing the network traffic of the building automation system using a packet analyser
2. After the building closes, stop observing the windows and capturing the network traffic
3. Go over the images, find those that recorded changes in illumination, and note down their timestamps, as well as the identifiers of the respective rooms
4. Go over the network traffic, find the BACnet objects whose state has changed every time lights turned on or off in a discovered room, and associate them with the related rooms

The two experiments are quite similar, in fact, their approaches to data collection are nearly identical, such that the data collected as part of either experiment is also suitable for the other experiment, allowing us to combine the first three steps of the two experiments, and carry them out only once.

In the first and second steps of the experiments, we observed the windows of the building and captured the network traffic of the building automation system on three consecutive business days, between 18:00 and 00:00. We collected data on three different days to increase the quantity and improve the quality of our results, and chose these times, because in the evening hours the occupants can no longer rely on natural illumination, therefore there are more changes in illumination to observe and gain knowledge from than during the day.

For the purpose of data collection in the cyber domain, we used the same method to capture the network traffic of the building automation system as in the previous stage, however, we had to filter and export the captured network traffic differently, using the command given in Listing 4.4.

```
tshark -r ./traffic.pcap -Y '(bacapp.type == 0 or bacapp.type == 1) and (bacapp.
    confirmed_service == 1 or bacapp.confirmed_service == 31 or bacapp.
    unconfirmed_service == 2 or bacapp.unconfirmed_service == 11) and bacapp.
    instance_number' -T fields -e frame.time_epoch -e eth.src_resolved -e eth.
    dst_resolved -e bacapp.instance_number -e bacapp.objectType -e bacapp.
    present_value.enum_index -e bacapp.present_value.uint -e bacapp.present_value.
    real -Eseparator=, -Equote=d
```

Listing 4.4: `tshark` command filtering and exporting network traffic to CSV

In the physical domain, we used an Android smartphone, deployed in the windows of a building next to the Atlas building, to record video footage of the windows of the building. Due to time constraints, we did not observe the entirety of the building, instead, we wanted to show a proof-of-concept focusing on the middle part of its eastern side. We could do this without loss of generality, as observing more of the building would have only affected the quality and quantity of the results. We recorded the videos using the stock camera application, transferred them as MP4 files to a computer, and converted their frames into timestamped images using the `ffmpeg` command given in Listing 4.5.

```
ffmpeg -i video.mp4 -frame_pts true %d.jpg
```

Listing 4.5: `ffmpeg` command converting videos into timestamped images

We recorded the videos in 1080p (1920x1080) resolution at 0.1 frames per second. We had chosen these parameters such that the images would be manageable by a single human, yet sufficiently detailed and frequent, where adjacent windows and consecutive events are still distinguishable from one another. Higher values, especially a higher frame rate, would have yielded more accurate results, as the timestamps of the frames would have gotten closer to the actual times of the observed events, and the windows would have been easier to discern, but they would have also increased complexity, as there would have been more frames and pixels to process, so there is a trade-off between accuracy and complexity, and different values may work better in different situations. Regarding the frame rate, no value can be considered too low, as long as all events are observed, asynchronous events do not appear to coincide, and the timestamps allow us to find the relevant state changes in the network traffic. Our chosen frame rate of 0.1 FPS put all events in time frames of 10 seconds, meaning that the difference between the measured and the actual time of an event was always at most 10 seconds.

In the third step of the experiments, we processed the images by viewing and comparing them manually with the intention of detecting changes in illumination and determining the coordinates of the windows in question. Considering the manageable amount of data, this was faster, easier, and more accurate, than developing a more sophisticated, automated approach for the purpose. We noted down our findings in a spreadsheet, containing for each image its timestamp, the sets of windows through which the brightness of the environment has increased, and the sets of windows through which the brightness of the environment has decreased.

While observing the windows of the building and subsequently processing the images, we encountered a number of challenges that we had not considered, which are inherent to our implementation and would likely be present elsewhere.

The first challenge was the positioning of the camera, which has by far the biggest impact on the quality of the images. If the camera is too far from the building, then some windows might be indistinguishable, and if it is too close, then parts of the building may be out of frame, and even the parts that are in frame could be difficult to see, due to bad viewing angles. Similar issues arise if the camera is too off-centre relative to the building, or if it is observing the building at an angle. Furthermore, if one camera cannot be deployed such that none of these issues is a concern, then multiple cameras need to be deployed at various locations and their clocks also need to be synchronised, so that all parts of the building are adequately observed and observations of the same phenomena can easily be combined.

The second challenge was the amount of daylight, whenever there was too much or too little of it, such as in the middle of the day or at night. In the case of the former, we could clearly see the borders of the windows, but it was virtually impossible to perceive changes in illumination, while in the case of the latter, it was the other way around. If all images are taken from the same positions, then we can retrieve the positions of the windows from the images where they are visible, and make use of them in the images where they cannot be discerned, but there is no similar solution to the problem of perceiving changes in illumination, which remains intractable.

The third challenge concerned contradictions in the physical observations. On some occasions, we have observed changes in illumination through a set of windows, whose superset or subset we had already regarded as a room, which raised concerns about the assumptions that we had previously made, however, there are explanations for these phenomena. If the brightness of the environment seems to change, but only through some of the windows of a room, then the windows, through which the brightness did not change, either had blinds on, or were too far from the lights, or did not actually belong to the room, and if the same is observed through the windows of different rooms, then either the lights in those rooms turned on or off roughly at the same time, or the windows actually belonged to the same room. Given a sufficient number of observations, we can reason about most, if not all, dubious cases with a high degree of confidence, based on the amount of evidence for and against each possibility.

The fourth and last challenge was dealing with different levels of brightness in a room. In some instances, lights appeared to be on in a room, but then the room became even brighter, while in other instances, lights appeared to turn off in a room, yet the room remained bright, albeit to a lesser extent. This indicates the presence of multiple (groups of) lights, which are controlled separately and independently, complicating the cyber-physical mapping, because multiple BACnet objects need to be mapped to a single room, but some of those objects may not change their state when there is a perceived change in illumination, as the fourth step of the second experiment suggests. To that end, we have decided to relax our condition on associating BACnet objects with rooms, and update the step in question as follows:

4. Go over the network traffic, find the BACnet objects whose state has changed when lights turned on or off in a discovered room, and associate them with the related rooms

In the fourth and fifth steps of the first experiment, we looked for conflicts, more explicitly intersections, among the window sets, which we resolved manually based on their respective occurrences and possible causes.

In the fourth step of the second experiment, we exported the captured network traffic and the processed physical observations to CSV, from which we constructed the cyber-physical mapping using a Python script. In order to relate BACnet objects to rooms, we looked for state changes occurring at most 60 seconds before the perceived changes in illumination, and associated a BACnet object with a room, if its state has changed from 0 to 1 when lights turned on in the room, or if its state has changed from 1 to 0 when lights turned off in the room, and remained 0 until the time of the correlated change in illumination. For each associated object and room, we have also indicated the confidence c, denoting the proportion of changes in illumination that have been correlated with state changes, with respect to the room and object.

In Chapter 5, we discuss the results and findings of the experiments, while the source code of the mapping algorithm, the list of discovered rooms, and the generated cyber-physical mapping can be found in the project's repository on GitLab.

4.3 Using the cyber-physical mapping

The third and last stage of cyber-physical reconnaissance concerns the application of the knowledge gained in the previous two stages for the purpose of identifying and locating physical events in the building. Since the devices of the building automation system detect and respond to changes in occupancy, the cyber-physical mapping allows us to relate occupancy measurements to rooms, and thereby track the movements of the building's occupants.

For that reason, we capture the network traffic of the building automation system, and use the correlations found in the first stage and the cyber-physical mapping constructed in the second stage to relate the state changes of BACnet objects to physical events and locations corresponding to changes in the occupancy of the discovered rooms. With the aim of reducing ambiguity in the locations of the reconstructed events, we also trim the cyber-physical mapping, in an effort to transform it into a true surjection.

4.3.1 Experiments

The goal of the third stage is to demonstrate the threats posed by the approach, which allows us to weaponise the knowledge gained in the previous two stages as a means to undermine the security of the building and the privacy of its occupants. To that end, we capture the network traffic of the building automation system, and use the cyber-physical mapping that we have procured in the second stage to enrich the state changes of BACnet objects with physical features, that would normally have to be obtained from physical observations. We discussed the specifics of the former in Section 4.1, while the latter depends entirely on the quality of the derived cyber-physical mapping, and the capabilities and characteristics of the building automation system.

More specifically, the precision of the reconstruction is bound by that of the cyber-physical mapping, and the density, frequency, and granularity of the measurements made by the devices of the building automation system. In particular, if the cyber-physical mapping has no association for a BACnet object, then there might be gaps in the reconstructed events, while if it has multiple, potentially incorrect associations for a BACnet object, then there might be contradictions in the reconstructed events, in addition to ambiguity and confusion. Because of this, it would be beneficial to trim the cyber-physical mapping, and discard all suboptimal associations, before using it for the purpose of event reconstruction. As for the limitations imposed by the building automation system, if there is only one device deployed in a room, then no distinction will be made between different parts of that room, and if a device makes a measurement once a minute, then events occurring within one minute may not be detected by it at all.

In the following, we describe the steps of an experiment aiming to demonstrate the utility provided by the cyber-physical mapping:

1. When the building is open, start manually observing the rooms of the building, and capturing the network traffic of the building automation system using a packet analyser

2. While observing the rooms, note down and timestamp the path of an occupant with respect to the discovered rooms

3. Before the building closes, stop observing the rooms and capturing the network traffic

4. Go over the network traffic, and relate the state changes of as many BACnet objects to rooms as possible using the cyber-physical mapping

5. Go over the related state changes, and infer the paths of the occupants with respect to the discovered rooms

Similarly to the first stage, we carried out the experiment in the evening hours of a business day, when most rooms of the building are typically unoccupied, including the ones we had discovered in the previous stage, allowing us to use them for the purpose of data generation and validation. With the intention of validating our results in Chapter 5, we have also obtained the ground truth by observing an occupant in the physical domain.

In the first and second steps of the experiment, we went inside the building and inspected some of the discovered rooms, then had one of the occupants walk in and out of them at arbitrary times in an arbitrary order, which we observed, timestamped, and noted down in a spreadsheet, containing for each event the time, room, and type, while capturing the network traffic of the building automation system.

For the purpose of data collection in the physical domain, we used an Android smartphone to note down and timestamp the path of the occupant with respect to the discovered rooms, which we observed using our own senses, while in the cyber domain, we used the same method to capture the network traffic of the building automation system as in the previous two stages, however, we had to filter and export the captured network traffic differently, using the command given in Listing 4.6.

```
tshark -r ./traffic.pcap -Y '(bacapp.type == 0 or bacapp.type == 1) and (bacapp.
    confirmed_service == 1 or bacapp.confirmed_service == 31 or bacapp.
    unconfirmed_service == 2 or bacapp.unconfirmed_service == 11) and bacapp.
    instance_number' -T fields -e frame.time_epoch -e bacapp.instance_number -e
    bacapp.objectType -e bacapp.present_value.enum_index -e bacapp.present_value.
    uint -e bacapp.present_value.real -Eseparator=, -Equote=d
```

Listing 4.6: `tshark` command filtering and exporting network traffic to CSV

While capturing and subsequently processing the network traffic, we did not face any challenges that we had not encountered before or addressed in Section 4.1, however, the execution was still not flawless, as we had to trim the cyber-physical mapping, and perform additional processing on the reconstructed events in order to mitigate noise and uncertainty, which were caused by three different, but closely related factors.

One of the three causes concerned the nature of the cyber-physical mapping. In the second stage, we associated a BACnet object with a room, if its state has changed when lights turned on or off in the room, which yielded a myriad of objects associated with multiple rooms, with varying degrees of confidence. While most of these associations may be informative or useful for evaluation, they do not provide much benefit in a scenario where we are interested in determining the physical locations of devices as accurately as possible, because they cannot be located in multiple rooms, thus they are expected to be in the room to which they have been related with the highest confidence c. As such, we have discarded all associations in the cyber-physical mapping that did not have the highest confidence c with respect to their BACnet objects, and used the remaining associations to relate the state changes of BACnet objects to the discovered rooms.

Even after discarding associations in the aforementioned manner, there were still rooms associated with multiple BACnet objects, as well as BACnet objects associated with multiple rooms, which acted as the other two causes of noise and ambiguity in the reconstructed events. The former was expected, as we have already established in Section 4.2 that there may be multiple (groups of) devices in a room, however, the latter was a direct consequence of too few observations, which had not allowed us to distinguish between the devices of some adjacent rooms, therefore they had been associated with multiple rooms with the same confidence c.

The issue of rooms associated with multiple BACnet objects can be solved by grouping consecutive state changes together that are similar, if not the same, and taking the earliest state change in each group. This way, we can focus on the state changes that recorded a change in the physical environment first, and discard the additional state changes, or use them for confirmation purposes. Intuitively, this approach makes sense, if we consider the typical scenario of an occupant entering a room with multiple occupancy sensors, which observe the change in occupancy in an arbitrary order, or perhaps in the order that the occupant approaches them, with the sensor closest to the entrance observing it first, whereas in other scenarios, it might be better to take the latest state change, or some aggregate of the state changes. For the sake of simplicity, we have taken the earliest state change in all instances, which has eliminated nearly all noise in the reconstructed events.

By contrast, the issue of BACnet objects associated with multiple rooms is more difficult to deal with, as it is unclear which rooms such objects belong to, which is caused by a lack of observations that cannot be resolved without uncertainty. In most cases, we can make the right choice between the rooms based on contextual information, such as the state changes of other BACnet objects, while in other cases, any choice between the rooms is as good as a coin toss. We have followed the former approach by discarding the state changes of BACnet objects associated with multiple rooms that contradicted the state changes of BACnet objects associated with a single room, so that we do not rely on observations that might have been made in other rooms to reconstruct events that are different from the ones that have certainly occurred around the same time. This technique has helped us eliminate uncertainty in the reconstructed events with regard to all rooms associated with such BACnet objects. Having said that, other techniques relying on contextual information can also work, potentially to a greater or lesser extent.

In the fourth and fifth steps of the experiment, we exported the captured network traffic to CSV using the command given above, trimmed the cyber-physical mapping as described above, and fed them to a Python script. The script used the trimmed cyber-physical mapping to relate the state changes of BACnet objects to the discovered rooms with almost no ambiguity, then reconstructed physical events from these state changes that exposed the movement patterns of the building's occupants.

In Chapter 5, we discuss our findings and validate the results of the experiment based on the ground truth, which are provided in the project's repository on GitLab, alongside the source code of the reconstruction algorithm.

Chapter 5

Evaluation and discussion

In this chapter, we discuss our findings and validate the results of our experiments, as well as consider potential prevention and mitigation strategies, and explore possible future directions of our work. In Section 5.1, we analyse the results of our experiments from Chapter 4, which we interpret, evaluate, and validate in Section 5.2, based on the ground truth. In Section 5.3, we propose prevention methods and mitigation strategies aiming to alleviate the threats posed by the approach, while in Section 5.4, we reflect on the limitations of our methodology and implementation, and explore how they can be improved as part of future work.

The source code of the processing algorithms, the results of the experiments, and the validation of the results can be found in the project's repository on GitLab. The data for the experiments has been collected in cooperation with Eindhoven University of Technology, in compliance with GDPR, and under the terms of a Non-Disclosure Agreement (NDA), which had been signed by the authors.

5.1 Results of the experiments

5.1.1 Understanding the cyber-physical process

In Section 4.1, we have managed to discover and gather information about the components of the building automation system by capturing and analysing its network traffic. The extracted information includes MAC addresses and BACnet devices, objects, and properties, based on which the discovered components can be divided into four groups.

The first group includes floor managers, whose names are prefixed with `TU_E` or `TUE`. In fact, the names not only have the same prefix, but also follow the same naming scheme, which identifies specific areas in the Atlas building. In particular, each component is responsible for overseeing a specific section of a floor, with each floor divided into 2 or 4 sections. According to the properties of their BACnet objects, most of them represent occupancy sensors with the binary states `UNOCCUPIED` and `OCCUPIED`, while the rest of them have analog states, which typically hover around 15.0, often in the interval $[15, 16]$. It is unclear what these states denote, however, the timeouts we have measured in relation to occupancy sensors and lights are exactly in that range, so it is possible that these objects are related to timeouts.

The second group consists of more than 100 room controllers, bearing the name [redacted] and MAC addresses affiliated with [redacted], which indicate that they were made by [redacted]. Looking at the properties of their BACnet objects, some of them clearly represent occupancy sensors, whereas many others, that have analog states, remain obscure. Interestingly, these occupancy sensors are different from those managed by the floor managers, since these differentiate between five states rather than two, namely `NULL`, `BYPASS`, `STANDBY`, `UNOCCUPIED`, and `OCCUPIED`, and there are only a few of these objects on each component. As for the analog states, we have not managed to make sense of most of them, however, the intervals of some are very close to $[20, 24]$, which strongly suggest room temperature measurements. Considering that some of these components also have objects with the binary states `INACTIVE` and `ACTIVE`, these objects could all be related to the HVAC system. Based on the characteristics of these components, they are most likely room controllers responsible for the physical devices of rooms, which would explain why there are so many of them, as well as why they have relatively few BACnet objects.

The third group concerns controllers which are uniformly named [redacted]. There are nine such components in the network, and they all wield MAC addresses associated with [redacted] which, combined with publicly available information[14], suggests that they are made by [redacted]. Based on the properties of their BACnet objects, they represent a variety of devices, many of which have binary states, such as `DICHT` and `OPEN` (meaning *closed* and *open* in Dutch), or multiple states, like `AUTO`, `OMHOOG`, and `OMLAAG` (meaning *auto*, *up*, and *down* in Dutch), while others have analog states scattered across countless different intervals. Most windows of the building are equipped with blinds, and many of them can be opened, so it is plausible that some of these objects are related to the windows, but they could also be related to the doors. Regarding the analog states, they are so diverse and volatile that it is difficult to determine what they are related to, however, for the same reasons, they are likely not relevant to our research.

The fourth group is the smallest, comprising four chillers, with MAC addresses attributed to [redacted], a subsidiary of [redacted]. The names of the components explicitly reference the model numbers [redacted] and [redacted], which are chiller units made by [redacted], therefore they are almost definitely part of the HVAC or the thermal energy storage system.

There are also two outlier components which do not have any BACnet objects or properties, but often communicate with other components over BACnet. Considering that their MAC addresses come from [redacted], they are likely virtual machines representing the operator workstations.

On the subject of communication, there are numerous patterns that we can observe in the network traffic, which provide insight into the relationships between the components. Our findings show that, aside from a few remarkable exceptions, components within the same group play similar roles in the building automation system, and have similar communication patterns, based on which they can be arranged in a hierarchy.

The floor managers and room controllers communicate with each other the most, typically the room controllers send `readProperty`, `readPropertyMultiple`, and `subscribeCOV` requests to the floor managers, to which they respond with the values of their properties, and send updates of the state changes of their objects, respectively. Since room controllers typically communicate with one floor manager, and similarly, floor managers typically communicate with multiple room controllers, as depicted in [redacted], there is clearly a subordinate relationship between these components, which attests our speculations about their types and roles in the building automation system. The components in these two groups do not only communicate with each other, however, most of them also communicate with one specific controller (in the third group) and one of the operator workstations. The controller typically sends `readProperty`, `writeProperty`, and `subscribeCOV` requests to the components of both groups, while the operator workstation tends to send `readPropertyMultiple` and `subscribeCOV` requests to the floor managers, and `readProperty`, `readPropertyMultiple`, and `subscribeCOV` requests to the room controllers. These interactions suggest that both components play central roles in the building automation system, which require them to be aware of the states of all devices in all rooms.

The controllers in the third group, aside from the one we have just mentioned, communicate with each other the most, in the form of `readProperty` and `subscribeCOV` requests, to which they respond with the values of their properties, and send updates of the state changes of their objects, respectively. The other operator workstation also communicates with them, as well as with the chillers, who do not communicate with each other, or with any other component, as illustrated in [redacted]. The operator workstation sends `readPropertyMultiple`, `writePropertyMultiple`, `subscribeCOV`, and `subscribeCOVProperty` requests to the controllers, and `readPropertyMultiple` and `subscribeCOV` requests to the chillers, suggesting that it also plays a central role in the building automation system, which requires it to be aware of the states of all chillers and other devices.

From these observations, it seems like the two outlier components are indeed operator workstations that supervise different parts of the building automation system. The first one oversees the devices related to the rooms, while the second one oversees other devices and components, such as the chillers. There are also controllers, one of which is responsible for the floor managers and their respective room controllers, while the others are responsible for devices and components that we have not managed to discover or learn about. Based on publicly available information[14], however, they could be related to the other systems of the building's infrastructure, such as the thermal energy storage system or the security system.

5.1.2 Constructing the cyber-physical mapping

In Section 4.2, we have managed to discover rooms and determine their locations and boundaries in two-dimensional space by relying on changes in illumination observed in the physical domain, then associate BACnet objects with the discovered rooms by correlating the changes in illumination observed in the physical domain with the state changes of BACnet objects captured in the cyber domain.

We have discovered rooms on almost all floors of the building, 69 in total. Some of these rooms spanned two floors, suggesting that adjacent floors are not always fully separated, and that some floors are not as extensive or spacious as others. We have also found similarities in the floor layouts, as attested by many rooms on different floors having the same relative locations. Examples of that include $N^{4,17}_{4,12}$, $N^{5,17}_{5,12}$, and $N^{9,17}_{9,12}$, as well as $N^{4,52}_{4,50}$ and $N^{6,52}_{6,50}$, $N^{4,56}_{4,53}$ and $N^{6,56}_{6,53}$, and $N^{4,59}_{4,57}$ and $N^{6,59}_{6,57}$. Lastly, some uncertainty remains about the boundaries of some rooms, as there were conflicting observations and not enough information to resolve these conflicts. These are $S^{6,11}_{6,9}$, $S^{6,18}_{6,13}$ and $S^{6,18}_{6,9}$, and $S^{3,11}_{3,7}$ and $S^{3,18}_{3,7}$.

We have managed to associate at least one BACnet object with 55 of the 69 discovered rooms, and multiple BACnet objects with 28 of the 69 discovered rooms, with at least 50% confidence. Since our goal was to associate BACnet objects with physical locations in the building, this was not a problem, as long as BACnet objects were not related to multiple rooms with the same level of confidence. There were several rooms whose devices we have not managed to set apart, because we have made too few physical observations involving them, which made it impossible to unequivocally decide which devices belonged to which room. Examples of this include $N^{3,2}_{2,0}$ and $N^{3,16}_{2,3}$, as well as $N^{8,11}_{8,8}$, $N^{8,14}_{8,12}$, and $N^{8,17}_{8,15}$.

Furthermore, if we look closely at the identifiers of the associated BACnet objects in the context of their respective rooms, some clear tendencies emerge. More specifically, the identifiers of the BACnet objects associated with the same room, as well as those associated with adjacent rooms, tend to be very close to each other. For example, on BACnet device 119507, we have mapped objects 1020 and 1021 to room $N_{9,7}^{9,10}$, whereas on BACnet device 118503, we have mapped objects 1007, 1008, 1009, and 1010 to rooms $N_{6,42}^{6,49}$, $N_{6,50}^{6,52}$, $N_{6,53}^{6,56}$, and $N_{6,57}^{6,59}$, respectively.

Looking at the associated BACnet objects corresponding to occupancy sensors, we noticed that all of them belonged to floor managers, and none of them belonged to room controllers, even though both types of components managed occupancy sensors. Interestingly, the occupancy sensors managed by room controllers rarely reported state changes, indicating that most of them were out of commission, as a result of which we have not managed to relate them to the discovered rooms. Additionally, in the cases of the BACnet objects that belonged to floor managers, we used the location identifiers in their names to discard any mismatches. Based on our findings in the first stage, in addition to the associations with 100% confidence, the building's floors go from -1 to 11, with the bottom row of windows corresponding to the second floor and the top row of windows corresponding to the eleventh floor. With that in mind, we could determine the true floor numbers of the discovered rooms, and discard any associations between the rooms and the BACnet objects of the floor managers of other floors.

5.1.3 Using the cyber-physical mapping

In Section 4.3, we used the cyber-physical mapping constructed in the second stage to associate the state changes of occupancy sensors with the discovered rooms, from which we have managed to reconstruct physical events corresponding to changes in the occupancy of the discovered rooms and exposing the movement patterns of the building's occupants.

We have managed to observe and reconstruct physical events in 17 rooms. In the majority of them, events have been reconstructed from the state changes of a single occupancy sensor, possibly leading a group of occupancy sensors as explained in Section 4.3, but in rooms $N_{4,18}^{5,31}$ and $N_{6,38}^{6,36}$, there were multiple occupancy sensors whose measurements were neither affirmed nor contradicted by other sensors, therefore events have been reconstructed from each of their state changes. There were also two adjacent rooms, $N_{2,0}^{3,2}$ and $N_{2,3}^{3,16}$, whose devices we have not managed to set apart in the second stage, therefore the same events have been reconstructed for both rooms, from the state changes of their occupancy sensors. This has effectively merged the events of the two rooms together, which introduced a lot of confusion into the timeline of events.

5.2 Evaluation and validation

5.2.1 Understanding the cyber-physical process

In the first stage, we have managed to discover nearly 200 components of the building automation system by capturing and analysing its network traffic. Based on their MAC addresses and BACnet devices, objects, and properties, we have divided them into four groups, with each group containing components of the same type, typically from the same vendor. We have combined these details with information obtained through open-source intelligence to figure out what kinds of components they were, and what roles and responsibilities they had in the building automation system. With the help of the operators, we were able to confirm some of our findings.

The components in the first group are [redacted] controllers acting as floor managers that supervise the floors of the building, with 2-4 controllers responsible for the sections of each floor. Their BACnet objects with binary states represent the occupancy sensors of the rooms in their respective sections, while their objects with analog states represent configuration interfaces for these occupancy sensors.

The second group of components comprises [redacted] room controllers made by [redacted], each of which controls 1-6 rooms on the same floor. Depending on these rooms, their BACnet objects correspond to a variety of sensors, measuring occupancy, temperature, CO2 concentration, humidity, and light intensity, and they may also relate to the state variables of the HVAC system, the windows, or the window blinds. Regarding communication, they use BACnet to interoperate with other components of the building automation system, while the operators use proprietary protocols to configure and manage them from the operator workstations.

The third component group consists of nine [redacted] controllers responsible for providing web-based interfaces to different parts of the building automation system, which is why other components notify them of the state changes of their BACnet objects at all times. In addition to this, some of them also serve other purposes, such as aggregating data from multiple components, or integrating other systems into the building automation system.

The four components making up the fourth group are chillers made by [redacted], which are located on the rooftop of the building, and provide the HVAC system with cooling.

Finally, the two outlier components are operator workstations running proprietary software [redacted] used for configuring and managing the controllers of the building automation system. One of the workstations is responsible for the electrical system, while the other is responsible for the HVAC and security systems.

As shown, the network traffic of the building automation system, combined with publicly available information, has provided us with adequate information to infer the types, roles, and responsibilities of most of its components, while the information on their BACnet objects and properties made it possible to relate their values to physical properties, such as temperature.

5.2.2 Constructing the cyber-physical mapping

As part of the second stage, we have managed to discover rooms in the building by observing changes in illumination in the physical domain, then associate BACnet objects with these rooms by correlating the changes in illumination observed in the physical domain with the changes in occupancy captured in the cyber domain. As a result, we have constructed a cyber-physical mapping that associates BACnet objects with rooms, and can therefore be used to enrich the state changes of BACnet objects with physical features. Eindhoven University of Technology provided us with the floor plans of the Atlas building, which we could use to validate the list of discovered rooms, however, we were unable to validate the associations that we have derived between BACnet objects and rooms.

Based on the floor plans, we have found 60 of the 69 discovered rooms to correspond exactly to actual rooms of the building, as 9 rooms were either more extensive in reality than the observations had indicated, or simply non-existent. Interestingly, many of the discovered rooms corresponded to several adjacent rooms, because in Atlas, rooms are not always physically separated by walls, however, in Section 4.2, we considered rooms to be areas separated by walls, thus this is a difference in terminology that we simply omit.

The rooms that we have not identified correctly can be divided into three groups based on the cause of misidentification. Four rooms were identified incorrectly, because they had windows distant from lights, through which we had not perceived the changes in illumination we had perceived through other windows. Four other rooms were misidentified, because they had windows that had remained dark or blank throughout our observations, likely due to window blinds. Lastly, there was one room that did not seem to exist at all, i.e. it was detected at a location where there were supposed to be no rooms, and it is unclear where the perceived light was coming from.

5.2.3 Using the cyber-physical mapping

While we were capturing the network traffic of the building automation system for the purpose of event reconstruction, we followed one of the occupants in the building, and manually observed, timestamped, and noted down their path with respect to the discovered rooms, as mentioned in Section 4.3. We have collected this information to validate some of the results of our experiment, in order to see how much the reconstructed events deviate from the ground truth, and thereby reason about the viability of the approach.

The observed occupant had visited eight of the discovered rooms, some of them multiple times, and our approach has managed to reconstruct events in almost all of them, with varying degrees of success. In particular, the occupant had entered and left room $N_{6,0}^{7,12}$ twice, however, our approach has not managed to reconstruct any of these events, because the cyber-physical mapping did not have associations for the occupancy sensor(s) of the room. In the other seven rooms, most events have been reconstructed either perfectly, or with a time offset, while some events have not been reconstructed at all. Events involving the occupant entering the rooms have been reconstructed not only correctly but also precisely, with less than a second between the inferred and the actual times, however, there has been a constant offset in the inferred times of the events involving the occupant leaving the rooms. Our algorithm has reconstructed these events correctly, but placed them 20 seconds later than they had actually occurred, which indicates an error in our findings concerning the timeouts of the occupancy sensors in Section 4.1. Lastly, the events that have not been reconstructed had all occurred in rooms which the occupants had entered or left in the preceding or the following 15 minutes, so there were no corresponding occupancy measurements from which the events could have been reconstructed.

It is clear from these results, that we have not reconstructed events as successfully as we might have wanted, however, if we learned more about the behaviour of the building automation system and located more of its devices, we would be able to track the movements of the occupants not only more accurately, but also more extensively. Nevertheless, these aspects are directly influenced and limited by the rate at which the devices of the building automation system report measurements, meaning that the extent of the threat is highly dependent on the deployment and configuration of the building automation system. As such, we believe that our approach can pose a risk to privacy, as well as to security, since the tracking of security personnel may reveal and weaken the security policies of the building, which would make it more vulnerable to physical threats, such as burglary.

5.3 Prevention and mitigation strategies

There is not much the occupants of the building can do against attacks targeting its building automation system, since they can hardly avoid its surveillance, or change their behaviour such that it reduces the risks. Nevertheless, even if they could, the building automation system would stop functioning as expected, as changes in human behaviour would also affect its rules and workflows, which would then need to be reconfigured. As such, adjustments would have to be made on the side of the building automation system, and there are numerous steps that the operators or other stakeholders could take in order to reduce the risks posed by the approach.

There are measures one could take with the aim of improving the security of the building automation system network, namely reducing the attack surfaces, and enhancing the security of communications. Concerning the former, the network of the building automation system should be isolated from all other computer networks, and it could also be split into multiple subnetworks, in an effort to prevent the whole network from becoming compromised in the case of an intrusion. As for the devices on the network, they should run up-to-date software that employs modern security protocols and algorithms, does not use default passwords, uses verified boot, and so on, which make them harder to exploit. Regarding the latter, components should encrypt their communications as much as possible, and only forgo encryption for diagnostic and debugging purposes, such that man-in-the-middle attacks become infeasible. Although encryption can mitigate the adverse effects of our approach, it may also impede the functions of the building automation system, curtail its compatibility, and increase its complexity and latency[13].

There are also adjustments that could be made to the deployment and configuration of the building automation system, with the intention of disrupting the processes of the approach. Since it relies heavily on the observation and correlation of events in the cyber and physical domains, the most effective strategy against it would be to weaken the relationship between these two domains. In the case of the building automation system at hand, one could do that by increasing the timeouts of the occupancy sensors and lights, or adding randomness to them. In general, the goal is to limit the frequency and accuracy of the measurements, either by configuring devices in such a way, or deploying hardware that is incapable of making measurements more frequently or accurately than needed. While such measures would not make the approach infeasible, they would hinder its processes, as well as limit the reliability of the cyber-physical mapping and the accuracy of event reconstruction. For that reason, a more powerful and effective strategy would be to add noise to the data, which would cripple the algorithms used for cyber-physical association and event reconstruction, as they would no longer be able to identify the relevant state changes in the network traffic.

5.4 Limitations and future work

Atlas has a typical building automation system comprising off-the-shelf components, primarily from Honeywell[14] and Philips[29]. Due to its standardised components, protocols, deployment, and configuration, we can expect other building automation systems to behave very similarly, in some, or all of these aspects. This makes our approach applicable in many environments, either as is, or with minor adjustments to accommodate different types of components and protocols, and potentially also different types of cyber-physical systems.

Nonetheless, the approach comes with several limitations, which could be remedied in the future, with the aim of increasing its accuracy, improving its flexibility, and extending its applicability.

Our methodology is based on the fact that the components of the building automation system form a cohesive network, however, this may not always be the case, so we should consider situations where the observation of a single network does not suffice, because the architecture of the building automation system comprises several independent networks, whose components do not have a complete view of the system.

Another assumption was that the attacker could access and read the network traffic of the building automation system, but this would be severely undermined by the use of secure channels, i.e. encryption, which would prevent the attacker from reading the contents of the packets communicated between the components of the building automation system. To that end, we should investigate techniques to bypass or overcome encryption, or rely on other methods to obtain the state changes of devices.

State changes and changes in physical properties were used to associate the representations of devices with the rooms of the building, based on the fact that the cause and effect of their actions could be observed in the cyber and physical domains roughly at the same time. Unfortunately, it could be that state changes are not communicated over the observed network channels, or that they are communicated at a delay. Consequently, we should consider alternative approaches that rely on other types of correlations.

One limitation that we cannot really influence is the quality of the measurements, more explicitly their density, frequency, and granularity, which are set by the building automation system, put an upper bound on the accuracy and reliability of the approach, and limit the amount of knowledge that can be extracted, as explained in Section 4.3. In the Atlas building, we were able to infer the movement patterns of the building's occupants, but in other smart buildings, we might be able to infer more or less information about human activities.

In our implementation, we were unable to distinguish between (groups of) lights within a single room, because the physical observations were not detailed enough to enable that. If we could distinguish between them by improving the quality of our physical observations, we could discriminate between parts of rooms, and thereby improve the accuracy of our approach. We could also use the improved physical observations to gain more knowledge about the behaviour of the building automation system by observing more nuanced physical events in the rooms.

The physical attributes of the Atlas building, particularly its uniform shape, combined with the window grids on all sides, helped us devise a method that allowed us to refer to locations in the building, discover its rooms, and determine their locations and boundaries. Atlas, however, is unique in this regard, as other buildings are unlikely to be so uniformly shaped, or have their sides almost entirely covered by glass panels, therefore we need to develop a better, more generic approach to denote locations in buildings and discover their rooms. Additionally, we need to think of better ways to observe them in the physical domain, as looking through windows and perceiving changes in illumination may not always be feasible, either because the building does not have (many) windows, or because the windows are reflective or otherwise opaque. Visibility is also strongly influenced by environmental factors, such as daylight and weather, due to which we might never discover rooms that are used only at certain times of the day. Consequently, we should diversify our approach, and explore what other aspects of the physical environment could be observed that allow us to discover rooms and construct a cyber-physical mapping.

On the technical side, we have focused on the communications of the building automation system that occurred over UDP, more specifically over BACnet, and as a result, we have not managed to discover or learn about components that used different protocols. In order to gain more knowledge about the building automation system, and enhance the capabilities of our approach, we should consider additional protocols which may be used by the components of the building automation system.

On the topic of gaining knowledge about the building automation system and the occupants of the building, our current processes and techniques could be improved in a multitude of ways, and there are a lot of additional use cases that we have not touched upon in our research.

First of all, if we collected more data, especially in the physical domain, our existing approach would produce more accurate and substantial results, since it could learn about other parts of the building, as well as increase the confidence and reduce the ambiguity in the derived cyber-physical mapping. This would require improving the way images are processed, as they are currently manually reviewed by a human, which is highly error-prone and does not scale. To that end, we should develop methods based on computer vision that can automatically process and extract data from images. These could be implemented using machine learning in a relatively short amount of time, and would make the approach not only more scalable, but also far more efficient and reliable, assuming that they are sufficiently capable and performant. They would also enable us to increase the frame rate of the image/video capture, in an effort to produce more accurate observations with fewer potential co-occurrences.

When it comes to the cyber domain, we correlated changes in physical properties with state changes, but we have not considered the strength of the correlations, which could provide vital information to our confidence metric c. In Section 5.1, we have highlighted two rooms, $N_{2,0}^{3,2}$ and $N_{2,3}^{3,16}$, whose devices we have not managed to set apart, because we have made the same physical observations involving them. As a consequence, their cyber-physical associations received the same confidence c, even though there were time differences between the relevant state changes, which could have indicated which devices belonged to which room. If we considered these differences, it would likely mitigate the ambiguity in the cyber-physical mapping, and improve the accuracy of event reconstruction.

There are also aspects to the state changes that we have not explored, which could provide additional information about the components of the building automation system and the occupants of the building. In Section 5.1, we discussed that some BACnet objects had analog states that we could not interpret. If we collected the states of these objects, and analysed their distributions over a period of time, we could likely infer their correspondences, and maybe even learn from their changes. For instance, we could discover which BACnet objects corresponded to air quality sensors, and use their state changes to infer occupancy, which could allow us to determine the number of occupants in a room, not just whether the room is occupied or not. We could also employ machine learning techniques to derive the movement patterns of the occupants at scale, even when the building is busy. Lastly, we could analyse the measurements of various kinds of sensors throughout the building to learn about various trends, such as which rooms are the most popular, which floors are the busiest, when most people enter and leave the building, and so on, which would reveal additional information about the occupants.

Chapter 6

Conclusions

In this paper, we presented a multi-stage approach that employs techniques proposed for anomaly and intrusion detection in industrial control systems to gather information about smart buildings and their occupants, by considering building automation systems as cyber-physical systems, whose activities can be observed in both the cyber and the physical domain. In particular, cyber-physical reconnaissance relies on observations made in the cyber and physical domains to associate the representations of devices with physical locations, and enrich the data collected by them with physical features, which could turn them into personally identifiable information that may have an impact not only on the privacy of individuals but also on the security of buildings.

In our case study, we have devised an elaborate method to refer to physical locations in a smart office building, and shown that it is possible to derive the locations and boundaries of its rooms by perceiving changes in illumination, as well as learn which devices of the building automation system are located in these rooms by also capturing the network traffic of the building automation system, and correlating the changes in illumination observed in the physical domain with the changes in occupancy captured in the cyber domain. Following this approach, we have discovered and located nearly 70 rooms in the smart building at hand, 80% of which we have managed to associate occupancy sensors with, allowing us to monitor their state of occupancy, and thereby track the movements of the building's occupants.

Our proof of concept clearly demonstrates that building automation systems can pose a risk to the security of buildings and the privacy of their occupants by collecting data about the physical environment at all times, and highlights the importance of deployment and configuration in these aspects, which have a significant impact on the extent of these risks. More specifically, devices that collect detailed data about the physical environment may expose sensitive or confidential information about buildings and their occupants, which can be used to carry out targeted attacks with wide-ranging ramifications. By taking these aspects into consideration and ensuring that building automation systems collect only as much data as necessary, it is possible to mitigate these risks, while maintaining the utility and flexibility of the building automation processes. That said, our work has explored only one of the many possible implementations of cyber-physical reconnaissance in only one real-world environment, and has only scratched the surface of the possibilities our methodology opened up for reconnaissance in general, therefore in the future, we would like to realise its full potential by improving our existing methods, as well as conduct experiments in more challenging environments, and explore various strategies that address its limitations.

Bibliography

[1] Abbas Acar, Hossein Fereidooni, Tigist Abera, Amit Kumar Sikder, Markus Miettinen, Hidayet Aksu, Mauro Conti, Ahmad-Reza Sadeghi, and Selcuk Uluagac. Peek-a-boo: I see your smart home activities, even encrypted! In *Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, pages 207–218, 2020. 5

[2] Rijad Alisic, Marco Molinari, Philip E Paré, and Henrik Sandberg. Bounding privacy leakage in smart buildings. *arXiv preprint arXiv:2003.13187*, 2020. 5

[3] Team V Architectuur. Atlas tu/e. `https://teamv.nl/en/projects/mainbuilding-tu-e/`. vii, 18, 19

[4] Leslie Ball, Gavin Ewan, and Natalie Coull. Undermining: social engineering using open source intelligence gathering. In *4th International Conference on Knowledge Discovery and Information Retrieval*, pages 275–280. Scitepress Digital Library, 2012. 4

[5] Giuseppe Bernieri, Estefania Etcheves Miciolino, Federica Pascucci, and Roberto Setola. Monitoring system reaction in cyber-physical testbed under cyber-attacks. *Computers & Electrical Engineering*, 59:86–98, 2017. 6

[6] Carrier. 30rbm - luchtgekoelde koelmachine | carrier. `https://www.carrier.com/commercial/nl/nl/producten-en-regelingen/koeling/luchtgekoelde-koelmachines/30rbm-30rbp/index.html`.

[7] Carrier. 30xw-v - watergekoelde koelmachine | carrier. `https://www.carrier.com/commercial/nl/nl/producten-en-regelingen/koeling/watergekoelde-koelmachines/30xw-v/index.html`.

[8] Defense Use Case. Analysis of the cyber attack on the ukrainian power grid. *Electricity Information Sharing and Analysis Center (E-ISAC)*, 388:1–29, 2016. 3

[9] Michael Cash, Shan Wang, Bryan Pearson, Qun Zhou, and Xinwen Fu. On automating bacnet device discovery and property identification. In *ICC 2021-IEEE International Conference on Communications*, pages 1–6. IEEE, 2021. 2

[10] Batyr Charyyev and Mehmet Hadi Gunes. Misactivation detection and user identification in smart home speakers using traffic flow features. In *Proceedings of the 14th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, pages 135–146, 2021. 5

[11] Wireshark contributors. wireshark/packet-bacapp.c at master · wireshark/wireshark. https://github.com/wireshark/wireshark/blob/master/epan/dissectors/packet-bacapp.c. 2

[12] Saia Burgess Controls. Sbc and honeywell - sbc. https://www.saia-pcd.com/en-gb/about-sbc/the-company/sbc-and-honeywell/.

[13] Davide Fauri, Bart de Wijs, Jerry den Hartog, Elisa Costante, Emmanuele Zambon, and Sandro Etalle. Encryption in ics networks: A blessing or a curse? In *2017 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 289–294. IEEE, 2017. 35

[14] Honeywell. Case studies - eindhoven university of technology. https://buildings.honeywell.com/us/en/solutions/case-studies/eindhoven-university-of-technology. 13, 30, 31, 35

[15] Abdulmalik Humayed, Jingqiang Lin, Fengjun Li, and Bo Luo. Cyber-physical systems security—a survey. *IEEE Internet of Things Journal*, 4(6):1802–1831, 2017. 3

[16] DRG Hunt. The use of artificial lighting in relation to daylight levels and occupancy. *Building and environment*, 14(1):21–33, 1979. 20, 21

[17] Ivo Jongsma. Breeam hails atlas the world's most sustainable education building. https://www.tue.nl/en/news/news-overview/15-04-2019-breeam-hails-atlas-the-worlds-most-sustainable-education-building/, April 2019. 13

[18] Markus Jung, Christian Reinisch, and Wolfgang Kastner. Integrating building automation systems and ipv6 in the internet of things. In *2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, pages 683–688. IEEE, 2012. 2

[19] Wolfgang Kastner, Georg Neugschwandtner, Stefan Soucek, and H Michael Newman. Communication systems for building automation and control. *Proceedings of the IEEE*, 93(6):1178–1203, 2005. 2

[20] Anastasis Keliris, Charalambos Konstantinou, Marios Sazos, and Michail Maniatakos. Open source intelligence for energy sector cyberattacks. In *Critical infrastructure security and resilience*, pages 261–281. Springer, 2019. 6

[21] Ellis Kessler, Moeti Masiane, and Awad Abdelhalim. Privacy concerns regarding occupant tracking in smart buildings. *arXiv preprint arXiv:2010.07028*, 2020. 7

[22] Ritika Raj Krishna, Aanchal Priyadarshini, Amitkumar V Jha, Bhargav Appasani, Avireni Srinivasulu, and Nicu Bizon. State-of-the-art review on iot threats and attacks: Taxonomy, challenges and solutions. *Sustainability*, 13(16):9463, 2021. 4

[23] Phillip Lee, Eun-Jeong Shin, Valerie Guralnik, Sharad Mehrotra, Nalini Venkatasubramanian, and Kevin T Smith. Exploring privacy breaches and mitigation strategies of occupancy sensors in smart buildings. In *Proceedings of the 1st ACM International Workshop on Technology Enablers and Innovative Applications for Smart Cities and Communities*, pages 18–21, 2019. 6

[24] Loytec. L-roc room automation. https://www.loytec.com/products/lroc.

[25] H Michael Newman. *BACnet: The global standard for building automation and control networks.* Momentum Press, 2013. 2

[26] Javier Pastor-Galindo, Pantaleone Nespoli, Félix Gómez Mármol, and Gregorio Martínez Pérez. The not yet exploited goldmine of osint: Opportunities, open challenges and future trends. *IEEE Access*, 8:10282–10304, 2020. 4

[27] Xi Qin, Martin Rosso, Alvaro A Cardenas, Sandro Etalle, Jerry den Hartog, and Emmanuele Zambon. You can't protect what you don't understand: Characterizing an operational gas scada network. In *2022 IEEE Security and Privacy Workshops (SPW)*, pages 243–250. IEEE, 2022. 6

[28] Wentao Shang, Yingdi Yu, Ralph Droms, and Lixia Zhang. Challenges in iot networking via tcp/ip architecture. *NDN Project*, 2016. 2

[29] Signify. Smart sustainable educational building – tu/e. https://www.interact-lighting.com/global/customer-stories/tue. 13, 35

[30] Muhammad Tariq, Z Zhou, J Wu, M Macuha, and T Sato. Smart grid standards for home and building automation. In *2012 IEEE International Conference on Power System Technology (POWERCON)*, pages 1–6. IEEE, 2012. 2

[31] Martino Tommasini, Martin Rosso, Emmanuele Zambon, Luca Allodi, and Jerry den Hartog. Characterizing building automation system attacks and attackers. In *2022 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 139–149. IEEE, 2022. 3, 10

[32] Tridium. Jace 8000 controller | niagara | tridium. https://www.tridium.com/us/en/Products/niagara/jace-8000.

[33] Pierre-Antoine Vervier and Yun Shen. Before toasters rise up: A view into the emerging iot threat landscape. In *International Symposium on Research in Attacks, Intrusions, and Defenses*, pages 556–576. Springer, 2018. 10

[34] Benjamin Vignau, Raphaël Khoury, and Sylvain Hallé. 10 years of iot malware: A feature-based taxonomy. In *2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, pages 458–465. IEEE, 2019. 10

[35] Xiao Wang and Patrick Tague. Non-invasive user tracking via passive sensing: Privacy risks of time-series occupancy measurement. In *Proceedings of the 2014 Workshop on Artificial Intelligent and Security Workshop*, pages 113–124, 2014. 7

[36] Lars Wüstrich, Lukas Schröder, and Marc-Oliver Pahl. Cyber-physical anomaly detection for ics. In *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pages 950–955. IEEE, 2021. 6

[37] Surendra Yadav, Brahmdutt Bohra, et al. A review on recent phishing attacks in internet. In *2015 International Conference on Green Computing and Internet of Things (ICGCIoT)*, pages 1312–1315. IEEE, 2015. 4

[38] Xu Zheng, Zhipeng Cai, and Yingshu Li. Data linkage in smart internet of things systems: a consideration from a privacy perspective. *IEEE Communications Magazine*, 56(9):55–61, 2018. 6