NETWORK TRAFFIC CHARACTERIZATION AND INTRUSION DETECTION IN BUILDING AUTOMATION SYSTEMS

A Dissertation
Presented to
The Academic Faculty

By

Celine Irvene

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology

August 2021

NETWORK TRAFFIC CHARACTERIZATION AND INTRUSION DETECTION IN BUILDING AUTOMATION SYSTEMS

Thesis committee:

Dr. Raheem Beyah, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. John Copeland
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Dennis Shelden
School of Architecture
Rensselaer Polytechnic Institute

Dr. Alvaro Cardenas
School of Computer Science and Engineering
University of California, Santa Cruz

Dr. Lee Lerner
CIPHER Lab
Georgia Tech Research Institute

Date approved: May 14, 2021

Whether you think you can, or you think you can't – you're right.

Henry Ford

ACKNOWLEDGMENTS

All glory be to God. Won't He do it! Will He won't! Won't He Will!

This one is for my mother - my biggest fan, my biggest critic, but most importantly my greatest supporter. Without her I would not be the person I am today or the best version of myself that I am striving to grow into everyday. Her sacrifices, efforts, tears, and prayers built me and I am so grateful.

I would be remiss to overlook the *village* that assisted me during this process, so here's my attempt. A very special shout out to my advisor Dr. Raheem Beyah, who must have seen something in me that I did not see in myself for him to have chosen to guide and support me through the (occasionally) tumultuous journey from undergraduate to PhD. Thank you to the wonderful women of the Georgia Tech College of Engineering Center for Engineering Education and Diversity (CEED) whose thoughtfulness, encouragement, and generally uplifting demeanor always snapped me back into reality when I was fading.

Thanks also to my father and the rest of my family (inclusive of my family and friends in Christ at GHWC) who never let me forget why I put myself through this, kept me humble, and always reminded me of where I came from.

Last, but not least thank you to all my friends and labmates in the CAP Group. Many times you all helped me and did not even realize it. One person in particular only just came into my life during this last stretch, but has made an incredible impact (you know who you are). The motivation I garnered from each of you propelled me to the present and I am truly appreciative.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The goal of this research was threefold: (1) to learn the operational trends and behaviors of a real- world building automation system (BAS) network for creating building device models to detect anomalous behaviors and attacks, (2) to design a framework for evaluating BA device security from both the device and network perspectives, and (3) to leverage new sources of building automation device documentation for developing robust network security rules for BAS intrusion detection systems (IDSs). These goals were achieved in three phases, first through the detailed longitudinal study and characterization of a real university campus building automation network (BAN) and with the application of machine learning techniques on field level traffic for anomaly detection. Next, through the systematization of literature in the BAS security domain to analyze cross protocol device vulnerabilities, attacks, and defenses for uncovering research gaps as the foundational basis of our proposed BA device security evaluation framework. Then, to evaluate our proposed framework the largest multiprotocol BAS testbed discussed in the literature was built and several side-channel vulnerabilities and software/firmware shortcomings were exposed. Finally, through the development of a semi-automated specification gathering, device documentation extracting, IDS rule generating framework that leveraged PICS files and BIM models.

CHAPTER 1
INTRODUCTION

Now, more than ever before, the concept of connected everything is becoming reality. In this new digital age networked refrigerators, dishwashers, thermostats, and various other home appliances are commonplace. The connectivity has allowed for the convenience of ”being there” without requiring one to actually ”be there”. Examples of this include using a smart watch to unlock a door for letting in a repair woman or using a phone to remotely monitor cameras when on vacation. These technological advancements are not contained to the home automation industry. Building automation has also seen major shifts in the last few decades [1] from only offering a few basic isolated services for heating, ventilation, and air conditioning (HVAC) and lighting [2] to managing the full control of a building with the touch of a button.

The emergence of smart or intelligent buildings has been driven by the desire to optimize building efficiency and sustainability, reduce building operating cost, and increase user comfort and ease of building management. The already massive industry ($46.37 billion in 2015) is anticipated to expand by a compound annual growth rate (CAGR) of 11.3% to 98.6 billion by 2022 [3]. The rapid development of the domain combined with the inherent lack of security from connecting legacy and devices running insecure protocols to the Internet has unearthed a host of vulnerabilities.

In 2013, researchers were able to hack the Google Australia building control system due to Google’s failure to patch their system [4]. The researchers, who had previously found numerous vulnerabilities on the Tridium Niagara AX platform, were able to obtain the admin password for the building management system (BMS). In 2018, a Google employee successfully hacked into the Google Sunnyvale access control system when he noticed the devices were all using the same hard-coded encryption key [5]. Not to mention the host

of indirect vulnerabilities that surface from the interconnection of these systems to information technology (IT) networks. After thorough investigation of the Target hack, which exposed about 110 million customers, on November/December of 2013 it was revealed that the attackers made the initial intrusion through network credentials stolen from an HVAC vendor [6].

The goal of this research was threefold: (1) to learn operational trends and behaviors of a real-world building automation system (BAS) network for creating building device models to detect anomalous behaviors and attacks, (2) to design a framework for evaluating BA device security from both the device and network perspectives, and (3) to leverage new sources of building automation device documentation for developing robust network security rules for BAS intrusion detection systems (IDSs). These goals were achieved in three phases, first through the detailed longitudinal study and characterization of a real university campus building automation network (BAN) and with the application of machine learning techniques on field level traffic for anomaly detection. Next, through the systematization of literature in the BAS security domain to analyze cross protocol device vulnerabilities, attacks, and defenses for uncovering research gaps as the foundational basis of our proposed BA device security evaluation framework. Then, to evaluate our proposed framework the largest multiprotocol BAS testbed discussed in the literature was built and several side-channel vulnerabilities and software/firmware shortcomings were exposed. Finally, through the development of a semi-automated specification gathering, device documentation extracting, IDS rule generating framework that leveraged PICS files and BIM models.

CHAPTER 2
LITERATURE REVIEW

2.1 Building Automation System Security

In recent years with the number of cyber attacks plaguing industrial control systems (ICSs) on the rise, the field of building automation system security has become a very hot topic. This for good reason as the demand for smart buildings and resulting growth of BASs has seen tremendous strides and with it the proliferation of vulnerabilities that make them attractive targets for attack. The key features of the modern day integrated building automation systems tend to be their main sources of insecurity, namely external facing connections, largely centralized systems, and connections to local enterprise networks. As much as these networks resemble traditional IT networks, existing IT security mechanisms cannot be seamlessly integrated [7]. Some reasons for this are the encapsulation of BAS protocols in IP which can be problematic for some detection systems to unpack, strict BAS bandwidth and processing requirements which hinder the use of some confidentiality processes, and the high availability requirement of BASs which interfere with IT patching philosophies [8]. Though research is being performed to address some of aforementioned issues, security gaps still exist and thus dedicated solutions must be developed.

Building automation system security, at a high level, addresses three areas of concern: secure protocol design, BAS security awareness, and threat detection/prevention. Many works in protocol design security have surfaced proposing secure alternatives and improvements to current protocols. In [9] the researchers proposed a secure BAN protocol based on time slotting and periodically refreshed ephemeral secret sharing, but this proved to be unsurprisingly slow by building automation standards. In [10], the authors proposed a centralized controller, which would serve as a proxy for all commands sent on the network,

be added to the KNX BAN. Transmitting nodes would then be required to go through the proxy and use Diffie-Hellman to establish a new key. In addition to the large overhead, this proposed solution offers no guarantees against attackers who could perform a man-in-the-middle attack during the key setup phase. Concerning security awareness in building automation the authors of [11] found that many industrial control system operators believe controller data is of no value to an attacker, and thus security need not be implemented. Raising security understanding and awareness is one of the first steps in securing BASs, and according [12] this should start at the organizational level for maximum effectiveness and be easy to comprehend to by all (technical and non-technical people).

The bulk of BAS security research lies in detection and prevention, which work hand in hand because without the ability to detect an attack or threat it is hard to prevent them. To address this [13] calls for layered network defenses like identity validation, firewalls, and encryption. This has led to solutions like BACnet specific firewalls [14] and rule based intrusion techniques which automatically drop malicious packets [15]. One must be cognizant of the application use case when considering each of these approaches for example, the latter technique could result in disastrous consequences for life-safety critical systems, as noted in [16]. Attack trees have also been used as tools for visualizing network and host system vulnerabilities [17, 18, 19], as well as building graphical security models. With the proper weighting of variables, attack trees could be leveraged to find problems in new systems, avoid vulnerabilities during system development, and evaluate security controls for known points of weakness. The main drawback of these approaches is that as the system/model complexity grows, so does the difficulty of attack tree formulation and evaluation.

2.2 Network Traffic Characterization

Network traffic characterization is extremely helpful in optimization and modeling applications. It can effectively highlight the difference between network operation in theory and

in practice. In optimization, knowledge of real-life traffic patterns are critical for designing more efficient future networks which is useful in modeling for creating accurate simulations. Models are necessary when the resources to build full production systems are too expensive or otherwise unavailable (as is typically the case in ICS/BAS). For these systems the foundation of knowledge depends largely on the correctness of the simulation/model and network intelligence is key. Other applications that rely on the precise knowledge of the underlying network include the development of intrusion detection algorithms and the detection of networked device fingerprints.

While there has been much research in characterizing pure Internet traffic, there has been less research published related to the characterization of dedicated control network (ICS/BAS) traffic. One of the first and most widely cited Internet characterization papers was on end-to-end behavior of bulk TCP transfers across nodes published in 1999 by Vern Paxson et. al. [20]. This paper revealed abnormal behavior seen on the network including details about packet loss, bottleneck bandwidth, and out of order deliveries which opened the floodgates for similar work. Control networks however, face different challenges than those in traditional IT. For example, the bandwidth consumption is typically low and it is uncommon for the network layout to frequently change, once connected devices tend to stay constant. In addition to this, control networks contain embedded devices and controllers which, in terms of their network behavioral footprint, are not very well studied. However, some limited research has taken place in the SCADA domain, in 2012 it was found that the traffic of a water distribution facility was unlike that of a typical IT network [21]. Research was also published in 2014 and 2017 that provided insight into power substation traffic [22, 23].

Concerning network traffic characterization in building automation networks (BANs), while the majority have Human Machine Interfaces (HMIs) for accessing and controlling devices via graphical interfaces, most maintain minimal to no network visibility for monitoring device communication [24]. With the upswing in the trend of smart buildings, data

and network visibility is more crucial than ever for ensuring the optimal performance and operation of buildings systems. Still little research has been performed to measure and analyze the long term disposition of BASs. In [25] a week long measurement study was performed on a university campus to collect BACnet/IP traffic, but provided no insight into the inner workings of the field level devices and (perhaps due to the short study length) offered no novelties regarding the operation of the IP network. Similarly in [26], the authors obtained three short ($\leq$ 9 days each) BACnet/IP traffic traces from a university campus and observed that their traffic was a combination of multiple flow-services. In this work they also built an anomaly detector to categorize traffic flows, but the false alarm rate is questionable and there is no support for field level devices. While presenting some interesting insights, this research only scratches the surface of anomaly detection in building automation. There is a whole sub-domain of intrusion detection solely dedicated to the identification of abnormal network behavior in control systems.

2.3 Intrusion Detection in BAS

Since the 1980s when the concept of Intrusion Detection Systems (IDSs) were first delineated, the domain has greatly evolved. There are many ways to classify IDSs, but one of the most commonly accepted methods is based on scope. An IDS can be host based or network based. When discussing intrusion detection in the context of control networks like ICSs and BASs, network based approaches are the most commonly utilized and can be separated into anomaly based, signature based, or specification based [27].

One of the first IDSs specifically designed for BASs was developed by Celeda et. al. [28]. Like the previously discussed traffic measurement works [25, 26], this work analyzed BACnet/IP and expanded the BACnet Flow tool to extract features from the BACnet Application Layer. The authors key hypothesis was that cyber attacks have higher randomness than normal traffic and that they could detect anomalous flows to compute data randomness. This technique can identify flooding/scanning attacks and potential botnet activity,

but falls short in detecting single flow attacks where flows have small rates and are drawn out. In [15, 29] rule based and anomaly based IDSs are proposed, respectively. The former approach first builds a normal model of BACnet traffic characterized by Network Protocol Data Unit (NPDU) and Application Protocol Data Unit (APDU) data. The researchers generate abnormal traffic and test their approach on a testbed fire alarm system where they obtain good classification accuracy, but false positive rates were too high for a real implementation. In the latter, network data is collected and modeled into Protocol Context Aware Data Structure (PCADS) and Sensor-DNA (s-DNA) then analyzed with Discrete Wavelets Transform (DWT) and rule based anomaly behavior analysis methods are performed to classify the data and perform countermeasures. The detection rate on each of the six attack categories defined by the researchers exceeded 90%, but like the former approach the false positive rates were too substantial to deploy in practice. Additionally, both approaches extend intrusion detection into intrusion prevention with ”protective actions”, but given their false positive rates this could lead to the dropping of legitimate network traffic. The work proposed in [16] succumbs to the same issues using specification-based methods to identify modifications to and violations of the BACnet protocol structure before packet dropping. In [30], the authors propose a context aware anomaly behavior analysis approach which pulls information from disparate BAS assets to extract features for creating a device data structure. These data structures are then used to form a baseline models which are queried to decipher normal and abnormal BAS behaviors. This work however has major flaws, namely it requires manual input of BAS asset information to be robust, it does not take into account field level sensor data, and the small scale evaluation results do not generalize outside of their testbed.

Other techniques employ machine learning for intrusion detection, like in [31] where the authors train an Artificial Neural Network (ANN) to detect BACnet timing attacks. The approach, however is very limited and only detects flooding attacks where commands are sent in rapid succession to cause system failure. Furthermore the use of obscure machine

learning models adds a black box into the equation, rendering the tracing of system alerts very difficult. The work of [32] suffers from the same black box obscurity of the last approach, though they leverage unsupervised learning techniques such as Principal Component Analysis (PCA) and clustering for identifying BACnet anomalies. This approach in addition to being preprocessing heavy, leads to a loss in semantics which help in understanding system decisions and comes with a non trivial false positive rate.

Caselli et. al. [33], the closest work by contribution to stage 3 of the proposed research, presents a specification-based BACnet IDS which scans for network devices then performs online searches to find documentation for each device. The data source used were BACnet Performance Implementation and Conformance Documents (PICS) and system configuration files (though these were rarely available). From these documents the authors automatically generated rules based on the permissible services, objects, and properties of each device. Then once rule generation was complete, monitored the network for rule violations, and alerted upon discovery of traffic infractions. This research was limited by the fact that it could only retrieve device documentation to characterize and generate rules once the device in question transmitted data packets with its vendor name, model number, etc. Furthermore, the system could only process PICS within a certain predefined format often requiring manual intervention and even then failing to extract key device details. Some of the latter limitations are overcome in [34], with automatic specification extraction, but the previously mentioned device identification shortcomings still exist and the researchers were unable to fully parse device PICS. Related to these, the researchers in [35, 36] create network IDSs to discover, classify, and characterize BAS devices through heuristic and threshold based methods for anomaly detection. The success of these approaches rely heavily on the accuracy of the hand picked thresholds and white box models. Furthermore the datasets used for analysis are small (only a few days of network traffic) and so the resulting evaluations are not generalizable.

CHAPTER 3

CAMPUS BUILDING AUTOMATION NETWORK CHARACTERIZATION

3.1 Introduction

The recent emergence of smart buildings has led to the connection of many formerly siloed and isolated BASs to the Internet. This has largely been driven by the desire to optimize building efficiency and sustainability, reduce operating costs, and increase user comfort and ease of building management. The connection to IP networks has offered a window into measuring BASs from the network layer [37, 25, 28]. To the best of our knowledge there is no published research which explores the BAS application layer or considers field level traffic (network data from sensors and actuators). Our analysis looks at BAS network traffic from the field layer, where raw payloads that shape the operation of the building are transmitted. This is arguably the most critical layer, without which there would be nothing to command and control for ensuring building function. In our efforts to better understand BACnet traffic as it exists in real world BAS deployments we captured building traffic through one of the most tumultuous events in recent world history, a global pandemic caused by the Coronavirus [38].

Our data captures uncovered unique aspects of building behavior that were identifiable due to the lockdowns imposed by COVID-19. From approximately March to June 2020 on-campus operations at several universities nationwide were closed to the public. Students were asked to leave on-campus housing and go back home, faculty and staff were instructed to work from home if possible, and quarantine began. The once bustling campuses that teemed with activity day and night were abruptly deserted at the hand of the Coronavirus. With the bulk of the student populations gone, campus infrastructures sat under utilized and primarily vacant for weeks on end.

In this chapter, we analyze the traffic across 9 buildings spanning a large urban university campus building automation network (BAN) and evaluate how the imposed shutdowns affected network activity. We also perform a case study on a 10th campus building to investigate the use of traditional IT network outlier detection methods in the building automation system (BAS) traffic collected from a select week of post-shut down data. While the popularity of network measurement on industrial control systems (ICSs) continues to grow, our analysis covers a largely untouched space in the community. The main contributions of this chapter are -

- We are the first to perform a field layer analysis of a real building automation network (to the best of our knowledge)
- We are the first collect a large scale longitudinal dataset from the selected university's BAS
- We study raw BACnet MS/TP traffic. BACnet is the most prominently deployed BAS protocol and our analysis gives us insight to traffic trends directly from field devices
- We show various changes in the campus BAN traffic due to operational shutdown which contradict the expected seasonal shift
- We build an outlier detection model from the traffic of a campus building and show it can be used to identify anomalous flows

These observations serve as a critical first peek into formerly obscure BANs and offer better insight towards understanding them. They also open the door for more exploration into the application layer of BASs. The remainder of this chapter is organized as follows, section 3.2 gives a general overview of building automation systems and the BACnet protocol. In section 3.3 we detail the university BAN and our dataset. Then section 3.4 presents our campus measurement study findings, followed by section 3.5 which focuses on our case study analysis. Lastly, section 3.6 summarizes and concludes the work.

3.2 Background

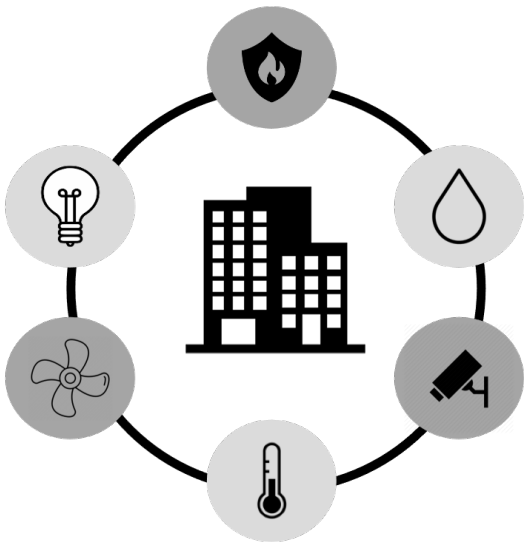


Figure 3.1: Example Building Automation System

A general description of a building automation system is a system used to monitor and control electrical and mechanical building equipment using a computer based platform. As shown in Figure 3.1, present day BASs may consist of several interconnected components. These components could consist of, but are not limited to, fire and life safety systems, HVAC systems, lighting control systems, and access control systems. Building automation systems scale from small and simple deployments to robust large customizable environments. While maintaining operational and functional requirements are the primary goals of BAS operators balancing operational cost, energy management, and maintaining user comfort must also be accounted for. These efforts should be simultaneously accomplished with the integration of a myriad of subsystems and the steady increase of intelligence and autonomy in the system. Field devices, controllers, and human machine interfaces (HMIs) communicate with the building automation network at all times to facilitate the BAS performance achievement.

The management level is where a remote operator receives relevant system details and input from the HMIs [39]. The automation level processes data received from the field

level as parameters for its automation logic and control statements. Sensors and actuators live at the field level, they operate and monitor control equipment which interacts with the physical environment. Together these levels make up the basic structure of every BAN, an illustration is given in Figure 3.2.

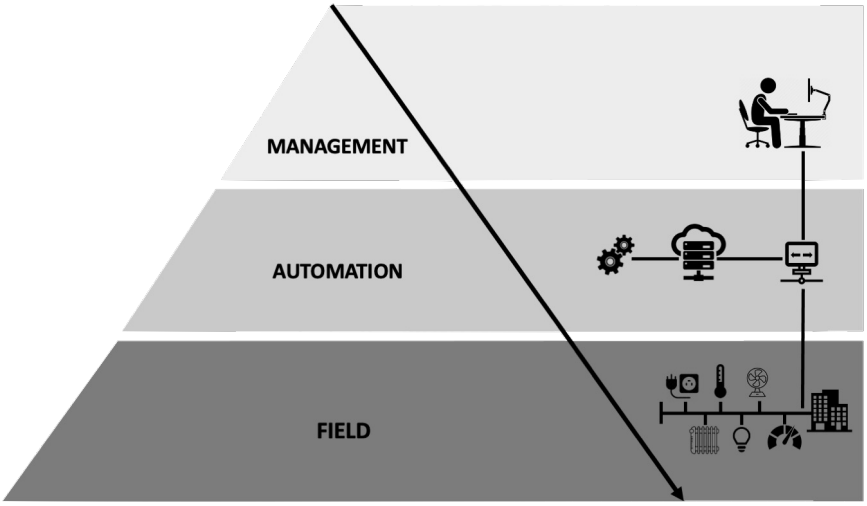


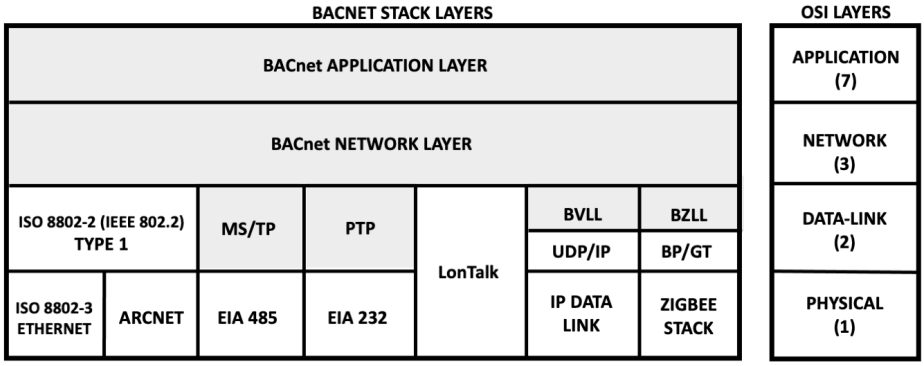
Figure 3.2: Network Architecture Pyramid for Building Automation Systems

3.2.1 BACnet

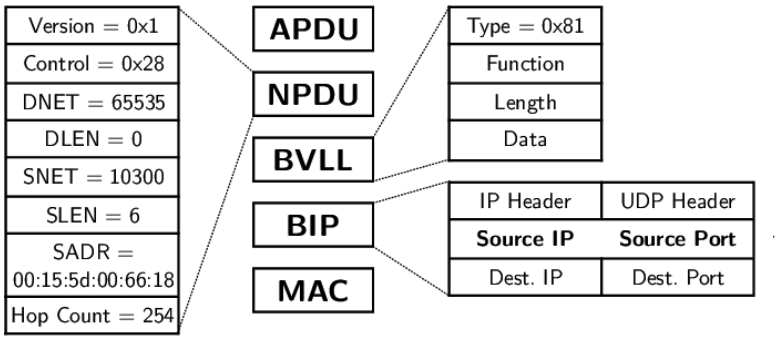
The Building Automation Control Network, BACnet, protocol was developed in 1987 by the American Society of Heating, Refrigerating, and Air Conditioning Engineer's (ASHRAE). BACnet utilizes a model consisting of objects, properties, services, and devices to facilitate *server/client* type of communication.

BACnet objects are collections of information related to functions that can be uniquely identified and accessed over a network. Every BACnet object is characterized by a set of properties which describe and command its behavior. A BACnet device is a cluster of objects that represent the functions of a physical BAS device. Services are the means through which BACnet devices share data and perform discovery of the networked entities. Figure 3.3a shows the BACnet protocol stack, consisting of an application, network, data link, and physical layer. In Figure 3.3b a sample BACnet/IP frame is presented, the *MAC*,

BIP, and *BVLL* form the data link layer, the *NPDU* makes up the network later, and the *APDU* is the application layer.



(a) BACnet Protocol Stack



(b) BACnet/IP Frame

Figure 3.3: BACnet Protocol Diagrams

Physical Layer

As shown in Figure 3.3a, the physical layer is the means by which devices are connected through the transmission of electrical signals for conveying data. In BACnet rules are defined for addressing, error checking, network access, flow control, message formatting, and signaling [40]. Each of these is dependent upon the physical media used for transmission.

Data-Link Layer

The BACnet data link layer is responsible for organizing data into frames/packets, regulating access to the physical medium, providing addressing, and to an extent, handling some

level of error recovery and flow control. Figure 3.3a, shows multiple data-link types supported by the BACnet protocol, with MS/TP and UDP/IP being the most commonly used [41].

Network Layer

The BACnet network layer is the mechanism through which messages are relayed from one network to another indifferent of the data-link technology used [42]. To do this it must provide functions like translating global addresses to the local address space, accounting for differences in network types and maximum message sizes, sequencing, flow and error control, as well as multiplexing.

Application Layer

The BACnet application layer is the level at which interfacing with user application programs occur. Figure 3.4 shows the structure of a BACnet MS/TP and an BACnet/IP frame. The inclusion of network protocol data units (NPDUs) and application protocol data units (APDUs) is dictated by the content of a control octet, which is the first octet of the NPDU (shown in Figure 3.3b).

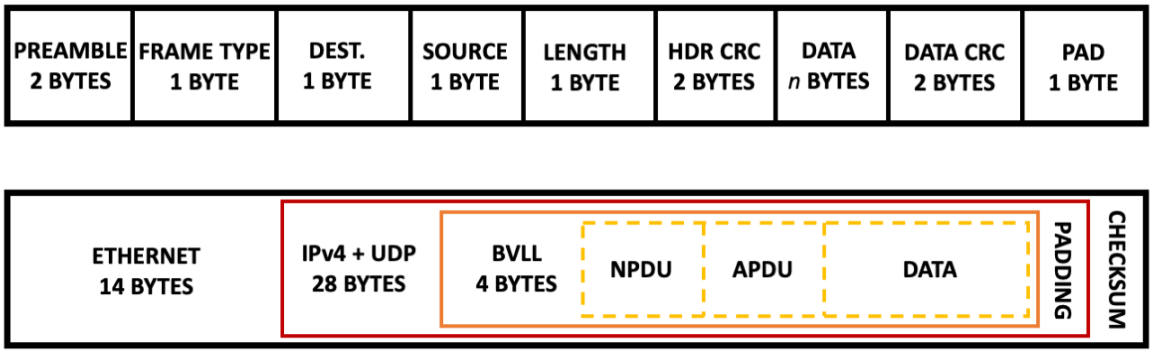


Figure 3.4: Detailed Structure of a BACnet MS/TP Frame (top) and BACnet/IP Frame (bottom)

For every BACnet packet the first byte of the APDU is split into two nibbles. The first identifies the APDU type, this could be one of eight options - Confirmed Request,

Unconfirmed Request, Simple ACK, Complex ACK, Segment ACK, Error, Reject, and Abort. To guarantee packet delivery the Confirmed Service is used with unique Invoke IDs to confirm each request. Confirmed Services require acknowledgement and thus signal the receiver to send either a Simple, Complex, or Segment ACK in response. Segment ACKs are used to acknowledge the receipt of one or more APDUs containing a segmented message. Invoke IDs are not used with Unconfirmed Services, as there is no expected response from the receiver. Errors are used to explain why previous Confirmed Services failed. Rejects indicate one of two things, syntactical flaws and other protocol issues that prevent interpretation or that the requested service is not available. Lastly, Aborts are used to terminate the connection between two peers. The second nibble typically contains flag values relating to segmentation, negative acknowledgements, or the origin of an APDU.

3.3 University Network and Dataset Details

The main campus of the university building automation network we studied is home to over 50 buildings with a myriad of purposes including academic buildings, sport venues, research buildings, residence and dining halls, as well as religious buildings, and miscellaneous multipurpose structures. Table 3.1 gives a breakdown of the different types of buildings studied for this characterization. For the purpose of this research an academic building is defined as a building used primarily for teaching and instruction with less than 50% square footage dedicated to other activities.

Our studied datasets come from a 2 year non-consecutive campus data collection effort dating from about 2019 to early 2021. The subsets explored for this work contain raw BACnet MS/TP network traffic captures divided into three time frames. The first time frame is Pre-COVID, from January 1, 2020 to March 12, 2020. The next time range encompasses the campus Shutdown and runs from March 13, 2020 to June 17, 2020. The university began ramping up and reopening limited on campus operation from June 18, 2020 and so our final time range (Ramp-Up) starts then and runs until the right before the commencement

Table 3.1: Campus Building BACnet MS/TP Details

Building	Primary Purpose	MS/TP Buses	BACnet Devices	Unique Device Types
One	Academic	1	13	4
Two	Research	4	22	12
Three	Research	1	3	1
Four	Academic	5	214	9
Five	Research	3	34	11
Six	Academic	2	135	9
Seven	Research	3	91	12
Eight	Academic	1	1	1
Nine	Academic	2	25	7
Ten	Research	11	220	15

of the Fall semester on August 7, 2020. Network traffic from every building (One - Ten) studied is present to some extent in each of the three time ranges, but gaps may exist. This is due to the isolated nature of BASs, many capture locations are electrical and mechanical rooms with poor Internet connectivity that posed a challenge to 24/7 data capture and transmission.

3.3.1 Building Details

The buildings studied come from a subset of the total university campus buildings. Through a partnership with the university facilities team and Johnson Controls Inc. (JCI), we were able to gain access to all buildings within central campus, 14 in total. Of the 14 buildings on central campus only 12 contain BACnet infrastructure (the others use a proprietary JCI communication protocol) and of those 12, 10 were selected for their interesting disposition. Buildings One through Ten not only vary in size and use, but also in diversity of equipment. The university BACnet networks studied are primarily used for facilitating communication to and from HVAC equipment, but lighting devices also have a small presence on the network.

As denoted in Table 3.1 the primary purpose of the central campus buildings is either Academic or Research with most buildings serving additional functions. For this work

Academic buildings are defined as buildings that are used more for teaching than any other activity. These buildings will largely contain classrooms and other instructional spaces. Research buildings are defined as buildings with more labs, offices, and collaborative spaces than classrooms. Other building types exist on the larger university campus, but the aforementioned are the only types on central campus. A high level overview of each building is given below -

- Building One - This is one of the older buildings on the campus, opened in 1952. It houses a variety of facilities including a workshop for laser cutting/3D printing and prototyping in wood, plastics, and metal. There is a gallery for exhibiting design work, an open reconfigurable review space for showing and critiquing work, multiple open plan studios for collaborating, and also several traditional classrooms.

- Building Two - Built in 1964, this building is home to much of the microsystems, optics, and photonics research that takes place on campus. It houses a number of chemical labs so a lot of the HVAC resources are dedicated to exhaust fans, gas monitors, and fan filtration.

- Building Three - Originally dating back to the 1950s, but recently renovated this building has over 10,000 sqft of teaching area, collaborative space, and faculty offices. Comparatively it is one of the smaller buildings at the university, but utilizes a newer high-efficiency small footprint HVAC system. As referenced in Table 3.1, the building leverages a single energy recovery unit (ERU) for exchanging the energy contained in exhaust air to precondition the incoming outdoor ventilated air.

- Building Four - This academic building houses multiple administrative offices for a university department as well as 9 classrooms, and 3 computer labs. Additionally it contains a 2,000 sqft data center which provides resources for both research and instructional servers. Due to the variety of zones and multipurpose spaces in this building there are a multitude of HVAC terminal units for individually controlling

small zones and other equipment for connecting the larger centrally controlled devices.

- Building Five - This building is versatile in use and made up of approximately 26,000 sqft of instructional space, 156,000 sqft of research space, and a multitude of office spaces for students and professors. Built in 1967, the structure most frequently houses physics and calculus classes, as well as labs. Standard HVAC equipment is used in this building such as air handler units (AHUs), fan coils, variable frequency drives (VFDs) for monitoring and controlling AHU motors.

- Building Six - Originally constructed in 1969 and located in the center of central campus, this building is about 90,000 sqft and was most recently updated between 2012 and 2013. This facility houses an academic department's main office, administrative and faculty offices, classrooms, and instructional and research labs. Housing the 3rd largest number of BACnet devices in our dataset, this building allows for some of the finest granularity temperature control and monitoring of all the buildings we studied.

- Building Seven - A lot of interdisciplinary research in microelectronics, microsystems, integrated optoelectronics, and microsensors/actuators takes place in this building. It is most commonly used by researchers who have offices, labs, and even clean room spaces on site. Given the vast nature of research interests and capabilities explored in the facility the networked HVAC system is just as advanced. It is made up of several lab exhausts, equipment for monitoring the intricate ductwork, and AHUs.

- Building Eight - Standing at 30,000 sqft and built in the late 1990s this building is one of the smallest studied. The facility is a living lab of sorts and largely used for teaching sustainable technology applications. It consists of a multimedia theater, research labs, computer centers and several faculty offices. From the BACnet side there is only 1 monitored roof top unit (RTU) for the building. RTUs are self-contained

units that provide both heat and air conditioning all in one box. These devices use less energy, take up less space, and tend to be easier to install.

- Building Nine - This structure is home to a departmental main office and contains a number of administrative offices, classrooms, and research labs. Originally built in 1962, the building is uniquely used for a lot of radio communications and thus has multiple antennas on the roof. The HVAC system of this building contains equipment for managing the hot water pumps (HWPs), chill water pumps (CWPs), and heat exchange (HX) between the two.

- Building Ten - The last building is unique from all the others we studied. It is the only building on central campus that has an entirely BACnet BAS system. This is due largely to its age, built in 2009 it is by far the newest building studied and one of the newer buildings on the university campus. Another interesting aspect of this building is that it houses one of the largest cleanroom labs for the fabrication, characterization, and assembly of biomedical semiconductors devices in the United States. Naturally, this has led to a very large and complex HVAC system consisting of exhaust fans, chillers, room pressure monitors, and makeup air units (MAUs). MAUs are incredibly prevalent on the building network because they pull in fresh air from outside to replace existing air that cannot be recirculated.

The operation of the HVAC systems in each building can be described as a controller driving an actuator based on a pre-programmed schedule or feedback from a sensor. The pre-programmed schedules are similar to what is commonly used in the residential sector for setting the heating and cooling of a home. The feedback from sensors are motivated by human or environmental changes, such as the triggering of motion sensors when people enter a room or outdoor temperature increasing past 90 degrees and making the indoor temperature rise above a cooling setpoint. With commercial buildings there are far more HVAC components than in the average residence, but the goal of optimizing user comfort

and energy efficiency remain. In Figure 3.5 an example controls sub-network is shown. This is representative of the campus controls sub-networks studied in this work containing several HVAC devices, including vents, fume hoods, air handlers, and thermostats all connected to one main controller.

We observed 10 unique types of system controllers on the network. These field level controllers included general purpose application controllers, vendor device controllers, humidity/temperature/critical environment controllers, and even controllers to map the wired bus communications to the wireless ZigBee protocol. Each of these controllers was connected to at least 1 of 26 sensor/actuator devices. These sensor/actuators ranged from duct sensors, to gas/temperature/humidity monitors, to ERUs, to hot/chilled water pumps, and all kinds of air handling/recirculating/supply/return units.

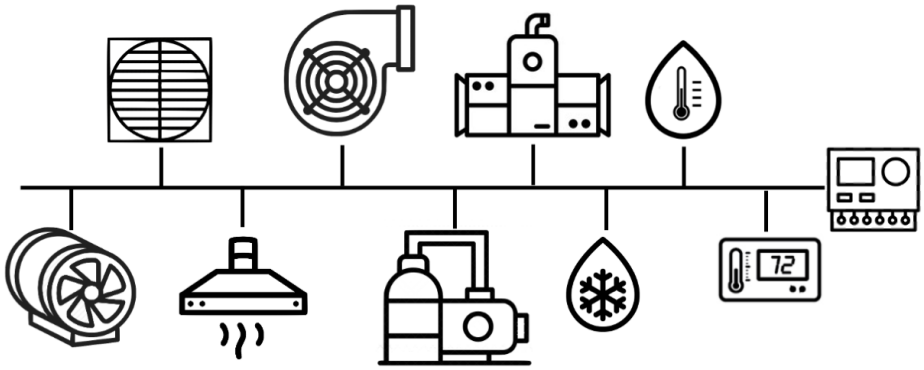


Figure 3.5: Types of HVAC devices in campus buildings

3.3.2 Data Collection

For data collection we designed custom capture tools (called Data Capture Tools or DCTs) built on top of the Raspberry Pi 4 (RPi4) Raspbian operating system. Each RPi4 capture tool had a quad core ARM processor, 4GB RAM, 64GB SD card, as well as Bluetooth and WiFi capabilities. A dual interfaced software module was developed and used to capture and process BACnet traffic. The software module ran locally on the capture tool and securely communicated via WiFi to a cloud server for further data processing. The capture tool node sat on field bus as indicated by the red nodes in Figure 3.6.

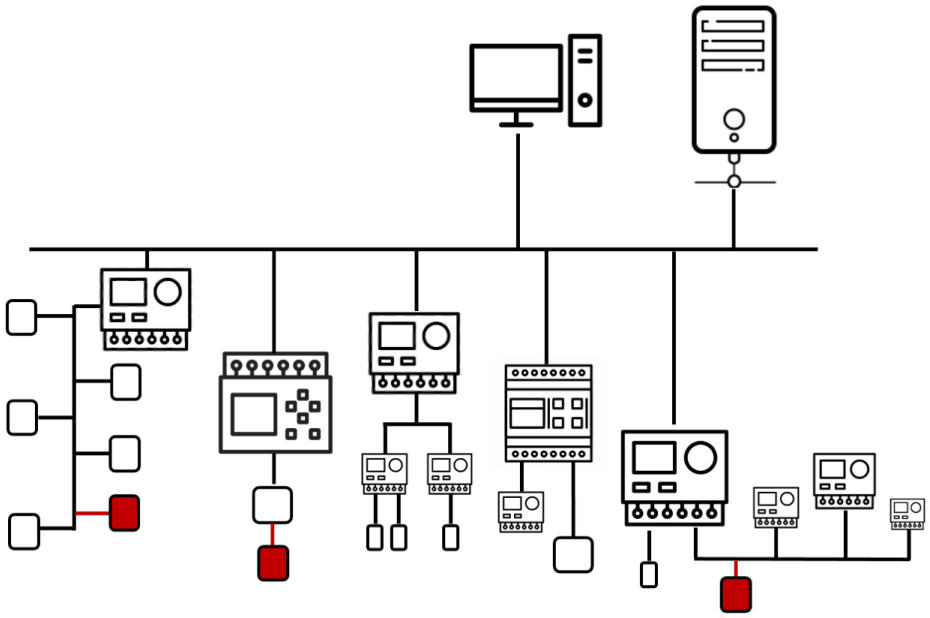


Figure 3.6: Logical network structure of each building's BAN

Each of the campus buildings studied has a BAN which logically resembles that of Figure 3.6. They each have 1 to n network control engines with 1 to n (typically 2) field buses running from them. Each field bus can have up to 127 field devices (controls equipment) connected at a time per protocol specification. For our buildings of study, HVAC devices are the primary used networked controls devices on each bus. Figure 3.5 provides a high level view of the type of controls devices we observed. Some examples of devices include thermostats, vents, fume hoods, hot and chill water systems, air handler units, and many others. Additionally, Figure 3.7 shows a data cabinet which served as a testbed and model for what most of the data collection sites in each building looked like. The black box in the bottom center is the custom DCT, built from a RPi4 as previously described (and highlighted in red in Figure 3.6).

3.3.3 Data Storage and Processing

After the data collection step occurred, as described in subsection 3.3.2, data from each building was transmitted over a secure WiFi connection to an S3 cloud server storage hosted on Amazon AWS [43]. The upload of a raw PCAP file would trigger an AWS Lambda

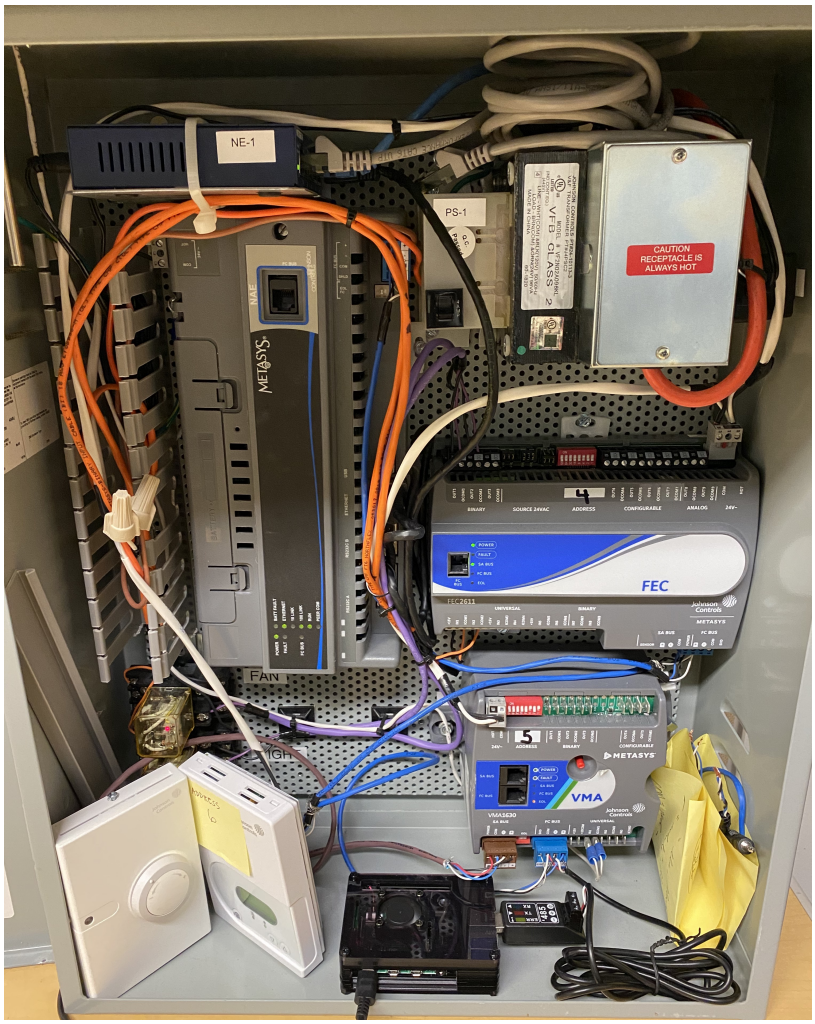


Figure 3.7: JCI Testbed and Data Collection Site Model

function which would then push the file details onto an AWS Queue. A local server, hosted in the Communications Assurance and Performance (CAP) Lab, would constantly consume messages from the AWS Queue. On the local server, several multiprocessing programs ran in parallel to repeatedly check the AWS Queue, pull individual files listed in the queue message from AWS S3, extract relevant PCAP details, and store them in an InfluxDB database.

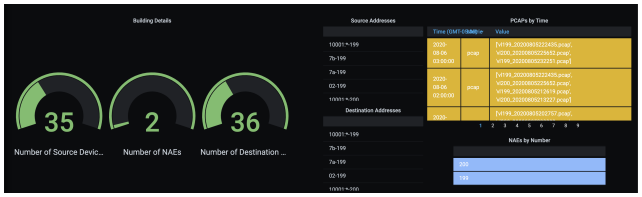
The information extracted from each PCAP covered about 95% of every packet. The only information not pulled was packet data payloads. This is because the payloads varied greatly from packet to packet and would have required deep packet inspection which is

very expensive for little reward. The packet details that were gleaned are as follows -

1. Source Address - Packet sender
2. Destination Address - Packet receiver
3. Network Engine/Controller Data - Primarily controller of the sender
4. Packet Length - Size of the packet
5. APDU Type - BACnet specific application layer packet data category
6. APDU Service - BACnet specific application layer packet operation being performed or requested
7. Packet Details - Extra data given if the packet APDU was in error, rejected, aborted, or malformed/unknown

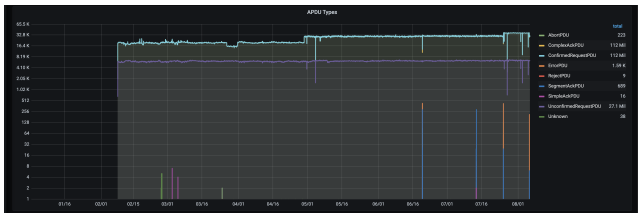
3.3.4 Data Analysis and Real Time Web BAN Dashboard

For data analysis once the network traffic was captured, and key network statistics extracted, they were displayed in a real time dashboard as visualizations. The dashboards are divided by building so that one can easily pinpoint and identify network issues on a more granular level. Among the metrics that have built out dedicated dashboard panels are packet lengths, shown in Figure 3.8e and Figure 3.8f. This is useful for tracking which network devices are transmitting uncharacteristic amounts of data. Packet lengths can also be an effective way for tracking the data bandwidth of devices over time. There are panels for source and destination addresses of packets (shown in Figure 3.8a and Figure 3.8b), these are critical for understanding the flow of traffic and identifying illicit device communication. Another metric that was extracted and given its own dashboard panel is the BACnet APDU type, shown in Figure 3.8c. This tells what category of data is being sent, for instance an Unconfirmed or Confirmed Request, an Acknowledgement packet, or even



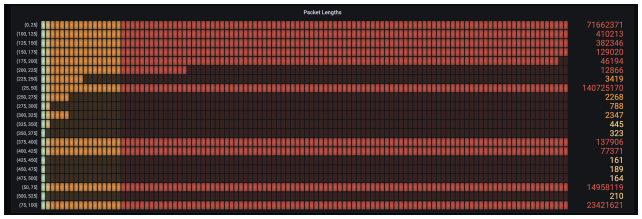
(a) Source and Destination Detail

(b) Source Address Traffic over Time

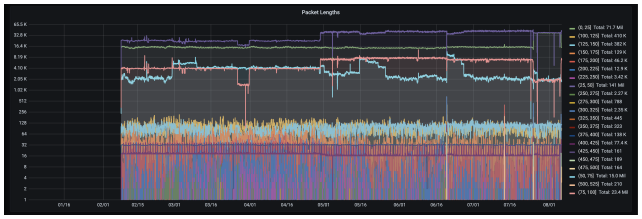


(c) APDU Type over Time

(d) APDU Service over Time



(e) Full Building Packet Length



(f) Packet Length over Time

Figure 3.8: Real Time Web Dashboard Panels

an Error or Reject packet and can be monitored over time for unusual change. Lastly, the BACnet APDU service choice was given its own panel and was analyzed to offer greater insight to the specific service or action being performed given a particular APDU type, shown in Figure 3.8d. Some examples of service choices associated with Confirmed Request APDU types are Reads, Writes, and Changes of Value.

The end to end pipeline from data collection to data analysis is shown in Figure 3.9 and a full screen capture from the Real Time Web BAN Dashboard is given in Appendix A.

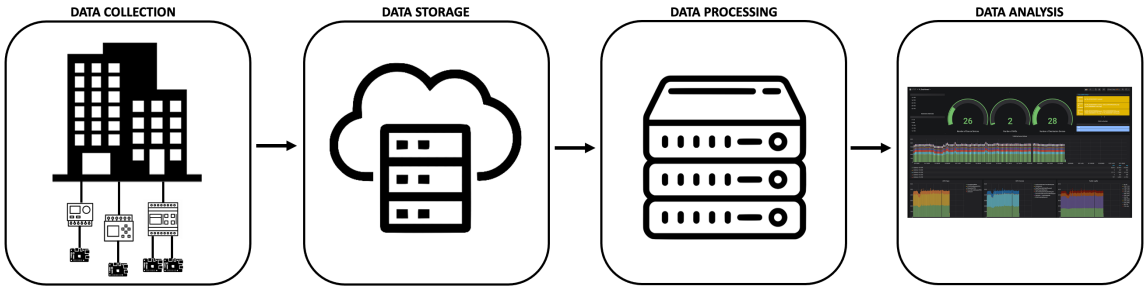


Figure 3.9: Campus Characterization Pipeline from Data Collection to Data Analysis

3.4 Network Measurement Through COVID Lens

In the three time ranges of interest, Pre-COVID, Shutdown, and Ramp-Up, we explore five aspects of the BAN traffic in the following subsections.

3.4.1 Traffic Volume Trends

The individual normalized traffic flow for each of the 9 campus buildings of interest is shown in Figure 3.10. The grey shaded region in each plot serves to highlight the campus shutdown time window from 3/14 to 6/17. An area filled version of this graph is shown in Figure B.1.

During the approximately 8 months of data collection 62.7 million packets were captured from Building 1. Being one of the smaller buildings within the dataset the number of packets transmitted was almost 10 times smaller than the largest observed. As seen in

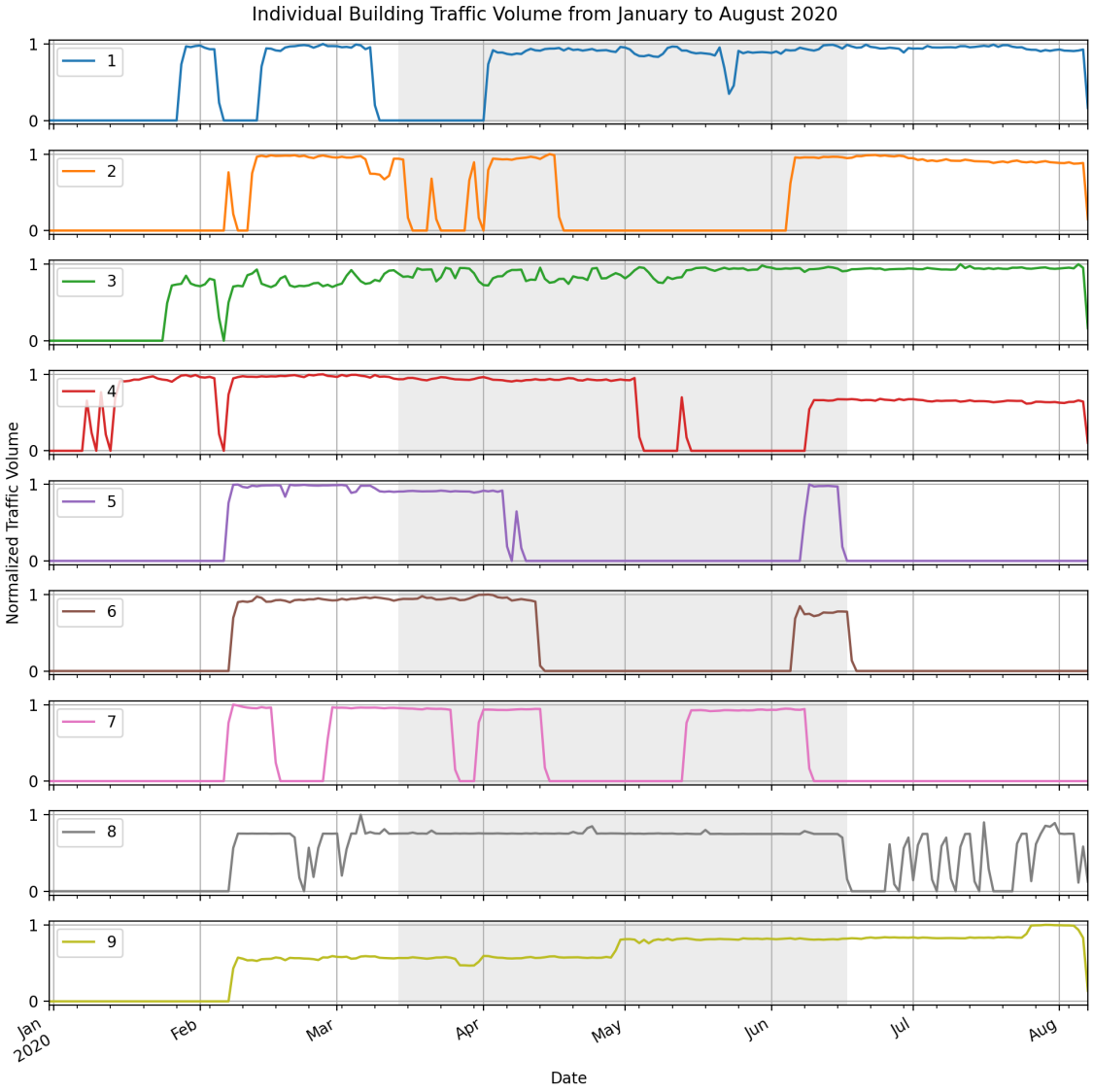


Figure 3.10: Individually Normalized Traffic Volume for Buildings 1-9

Figure 3.10 there are 2 gaps or time chunks with no data for Building 1. These represented periods of time (2/5 - 2/13 and 3/9 - 4/1) when the DCT went offline and was thus unable to capture any usable data. One other notable dip in traffic occurred between 5/22 and 5/24. For those 3 days the number of Confirmed Requests suddenly and without reason dropped, this was followed by an equally sharp decline in the number of Complex Acks and slight uptick in the number of Errors on the network. After digging deeper into the phenomena it was discovered that the *master* controller device or network engine reduced the number of transmitted heartbeat messages and this led to a ripple effect throughout the network. Heartbeat messages are messages sent as *keep alives* from the *master* controller to the *slave* or field devices. In the case of this university BAN the heartbeat messages are APDU service choice Confirmed Private Transfer and in Building 1 make up over 70% of the total network traffic. On a larger scale this down tick can also be attributed to the overall traffic decrease of the entire building in week 5/24, shown in Figure 3.11. Figure B.2 shows a line plot visualization of all the building volume traffic from January to August.

Data capturing from Building 2 began in early February and consisted of 5 data dropout periods (2/8 - 2/11, 3/16 - 3/20, 3/22 - 3/28, 3/31 - 4/2, and 4/18 - 6/4). Building 2 contains data from 4 separate field bus networks and the resulting dropout periods represent times when any 1 of the connected DCTs was unable to capture traffic. Commensurately the 3 spikes in the traffic following the dropouts do not represent anything unusual, but rather illustrate the rapid increase in traffic between zero points. In total, approximately 99.3 million packets were gathered over the 8 month data collection period. Figure 3.11 shows how the drop outs affected the overall network volume from March to June. One interesting observation noted in Figure 3.10 is the dip in traffic between 3/7 and 3/12. Surprisingly this was caused by a sudden drop in the number of Errors seen on the network. For almost 1 week the usual number of about 1,000 Errors fell to zero and this lead to a slight reduction in building traffic.

Building 3 is probably the most consistent building in terms of network traffic of all

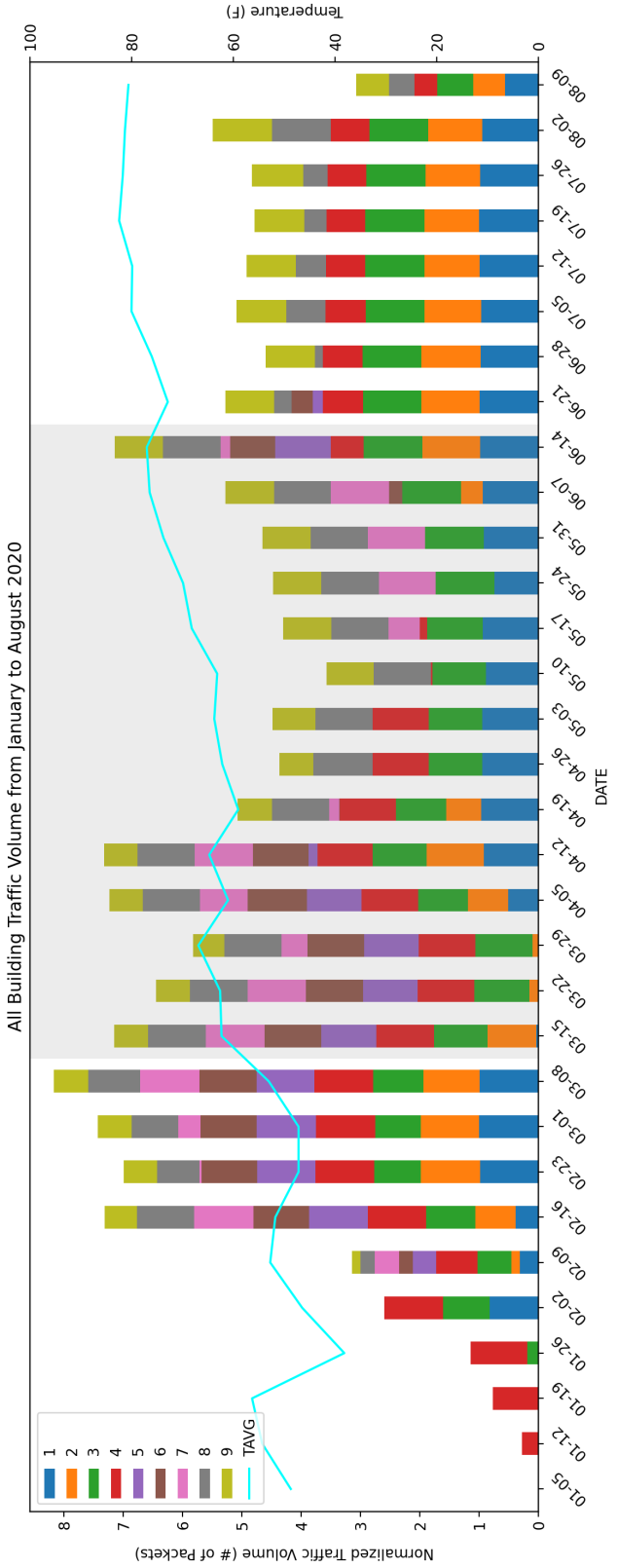


Figure 3.11: Building 1-9 Normalized Weekly Traffic Volume

those studies on the university's central campus. Except for one 2 day drop out between 2/5 and 2/7 the single DCT stayed online and captured data throughout the entire collection period. That stated, there were still only 37.6 million packets amassed throughout the time, which is 16 times less than Building 4. Given the small size of the building this is not surprising, there was only 1 BACnet field bus network for the whole building with 1 controller, 3 senders, and 5 receivers (where senders/receivers are not mutually exclusive).

Nearly 600 million packets were collected from Building 4 over the 8 month collection period in spite of the 5 dropout periods experienced between (1/9 - 1/10, 2/5 - 2/6, 5/4 - 5/12, and 5/14 - 6/8). Building 4 traffic consisted of packets from 5 field bus networks with 5 distinct controllers, 200 senders and 214 receivers. Depicted in red in Figure 3.11, Building 4 maintained a consistent presence on the campus network, many times having the most traffic.

Throughout the entire summer of 2020 Building 5 was under construction in preparation to reopen in the Fall 2020 semester. Due to this construction there was no power in the building for nearly 3 months from early April to early June. This is the cause of the long dropout in Figure 3.10. By mid June the power in the building was restored, but the construction continued and the DCTs eventually fell offline unable to be accessed for the remainder of the collection period. There were 153 million packets transmitted between 3 controllers, 30 senders, and 34 receivers on the network.

Building 6's traffic is similar in dimension to Building 5's, but there are some subtle differences. Building 6 had the same number of field buses and controllers on the network, but nearly 4 times as many BACnet devices. As a result, 305 million packets were transmitted over the 8 month collection period. Traffic volume levels were consistently maintained until 1 DCT fell offline and the capturing was discontinued.

With 224 million packets transmitted over the data collection period, Building 7 is the 4th largest building in the campus dataset by traffic volume. Due to the poor signal strength locations of these building network controllers, there were several dropout points.

The earliest of which goes from 2/17 - 2/27, then 3/26 - 3/30 and 4/13 - 5/13, before finally going offline permanently in mid June. When the DCTs were online they recorded fairly consistent rates of traffic from this building.

The smallest building of the dataset with less than 10 million packets captured over the collection period was Building 8. Unlike most of the other buildings previously discussed, the spikes in subplot 9 of Figure 3.10 are not due to DCT capture issues. The traffic volume was so small in this building that minor delays or drops in communication lead to inconsistency in the traffic volume. For instance, on 2/21 the number of Confirmed Requests dropped from approximately 1000 to 50 for an hour, then shot back up. This is the reason for the first dip in traffic, but is not attributable to any significant change in the network, just a lag in response from the controller. Another interesting occurrence that happened in early March and lead to a dramatic spike in traffic was a sudden increase in the number of Read Property Requests. The abruptness of the traffic flow indicates there may have been operator intervention at that time polling for additional data. Though only 1 controller was present on the network throughout the capture, traffic was transmitted between 2 unique senders and 4 unique receivers.

The final building of the dataset, Building 9, is the only building where the traffic volume increased considerably from the start of data capturing (Pre-COVID) through the end of the capture (Ramp-Up). As shown on the last subplot of Figure 3.10, in February the building comes online at the rate of 130,000 - 140,000 packets/3 hours, then suddenly at the start of May the rate jumps to about 200,000. Upon deeper inspection it was found that 3 new devices were added to 1 of the 2 field bus networks on the building. Another upward trend occurs at the end of July, this is not due to new devices joining the network. This phenomena was discovered to be caused by one particular BA device which doubled its normally transmitted 11,000 packets/3 hours and began sending 24,000 packets. The adjustment largely consisted of the device sending several Read Property Requests on the network. This could have been triggered by a changed operator setting that affected how

often the device reported data. These commensurately caused increases in the number of packets transmitted and led to shifts in the building's network traffic volume.

Through the investigation of the traffic volume from each campus building a pattern was observed. The initial hypothesis of this work was that due to COVID-19 and the subsequent university campus operation closures there would be noticeable changes reflected in the BAN traffic. Our expectations were that the Pre-COVID window for each building would have higher traffic flows than the Shutdown window because the buildings should have been heavily under utilized during those times. Additionally we hypothesized that the Ramp-Up window would have less traffic than the Pre-COVID window, but more traffic than the Shutdown window because campus was reopened (at 50% capacity). Our findings depicted in Figure 3.12 shows the traffic volume of each building's Pre-COVID, Shutdown, and Ramp-Up periods as a function of three 6-day time frames. The solid *Pre* data is from 3/3 - 3/8 (Pre-COVID), the dashed *Shut* data is from 4/2 - 4/7 (Shutdown), and the dotted *Ramp* data is from 7/23 - 7/28 (Ramp-Up). Buildings 5 - 7 have no *Ramp* line because there was no data from those buildings during that time.

What we discovered from reviewing this data is that there are 2 classes of buildings in our dataset, static and dynamic. The static buildings are driven more by their pre-programmed schedule than any other factor. The dynamic buildings are also influenced by their pre-programmed schedule, but also by feedback from the environment. The dynamic class can be distinguished from the static class by their APDU Service Choices. Particularly for this university campus dataset the Read Property Request and Read Property Multiple Request are highly prevalent and indicative that *environmental* BA device changes led to new data being generated and requested from the controller.

Buildings that fall into the dynamic class can be seen visually in Figure 3.12 to have lower *Shut* traffic volume than *Pre*. This trend is observed in Buildings 1, 2, 4, 5, 7, 8 and 9. Of the dynamic buildings only Building 1 exactly portrays the hypothesized behavior with the *Pre* having greater traffic volume than the *Shut*, and the *Ramp* higher than the *Shut*. The

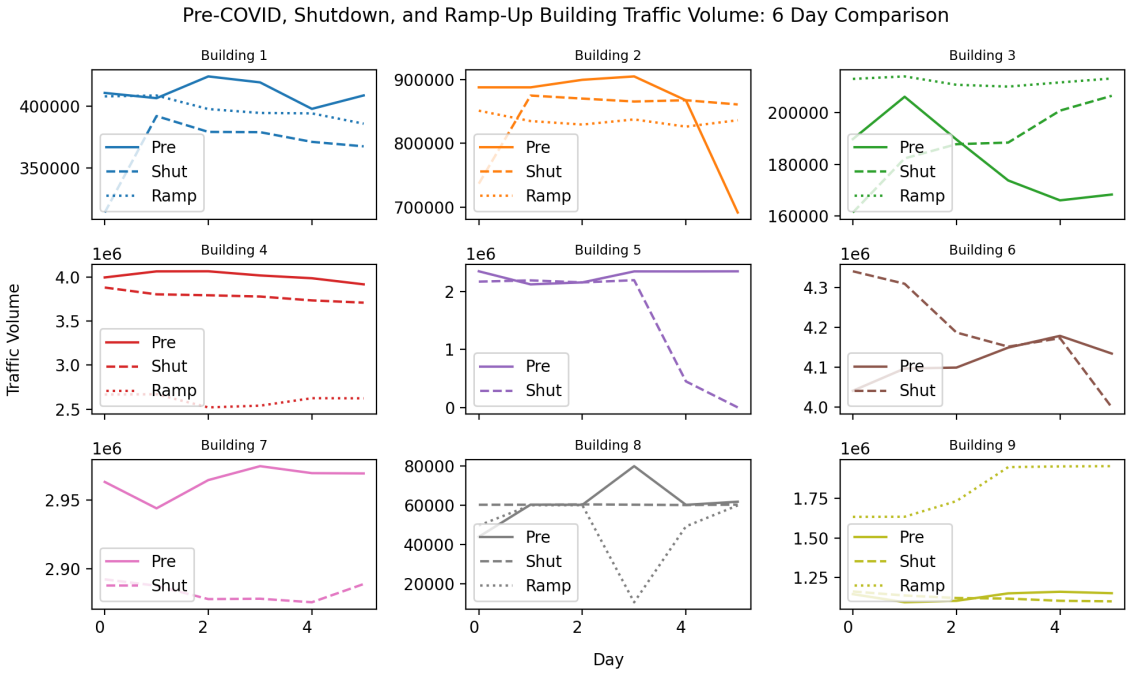


Figure 3.12: March 3-8 (Pre-COVID), April 2-7 (Shutdown), and July 23-28 (Ramp-Up) Traffic Comparison

Pre and *Shut* pattern can be seen in the subplots for each of the dynamic buildings. Upon analyzing the raw traffic to understand why the *Ramp* data of the other dynamic buildings (2, 4, 5, 7, 8, and 9) does not follow suit, a series of irregular occurrences were discovered. Buildings 2 and 4 lost BA devices from their respective networks during the *Ramp* week and thus their overall traffic volume decreased. No *Ramp* data exists for Buildings 5 and 7, but it is expected that the data would have been in accordance with Building 1 if it did. Building 8 saw bursts of unresponsiveness to heartbeat messages during the *Ramp* week and thus trended downwards, but began picking back up after the referenced time window. Lastly, Building 9 experienced what appeared to be the manual reconfiguration of a BA device to increase polling from late July and this led to much higher traffic flows than previously seen. Buildings 3 and 6 are classified as static because their traffic flow is much more heavily skewed towards heartbeat messages and other Confirmed Requests that are transmitted periodically rather than sent because of a physical occurrence in a space.

Another factor that must be considered when studying the effect of campus closures on

this university buildings dataset is the weather. As stated earlier, the BAN of the university primarily contains HVAC devices and by design these devices react to outdoor conditions as they affect the indoor climate. Figure 3.13 shows the average temperature in the city of the university during the *Pre*, *Shut*, and *Ramp* weeks. Given that the university is located in a subtropical climate zone and the study took place from winter to summer, it is necessary to consider the potential effect of the weather on our findings.

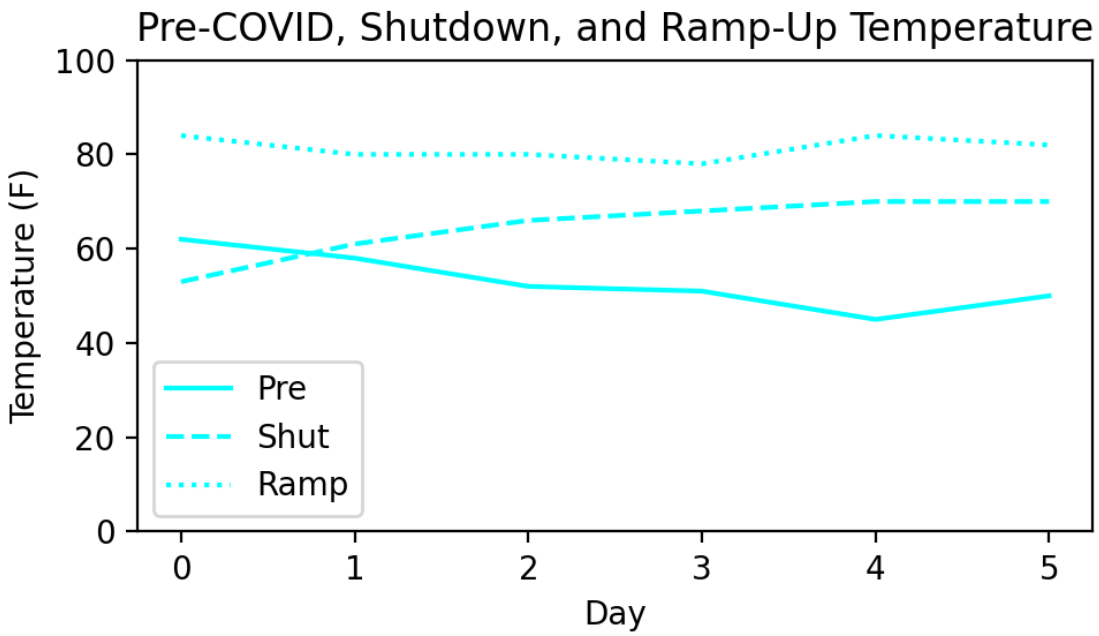


Figure 3.13: Pre-COVID vs Shutdown Week Temperature Comparison

In [44], five standard HVAC system types are designed in the EnergyPlus modeling software and the general consensus (among all the models) is that energy consumption is at peak in the summer months. July and August typically have the highest energy draw and April the lowest. This trend can be seen in Figure 3.14 and means that generally speaking HVAC equipment work hardest (using more energy) during the summer and under normal circumstances we would expect this to lead to increased BAN traffic. More devices should theoretically be talking to one another, exchanging data, sending notices of changing variables, and in general there should be more network activity. We would not expect to see

the trends we observed in our data Figure 3.12 where during July, the hottest month of the year, the traffic volume is lower than in March during a peak point of the semester under normal conditions. From this we draw the assumption that the campus closures reduced load the on the campus buildings so much that the effect of higher temperatures (as shown in Figure 3.13) were reduced. We explore this assumption further in the next subsection.

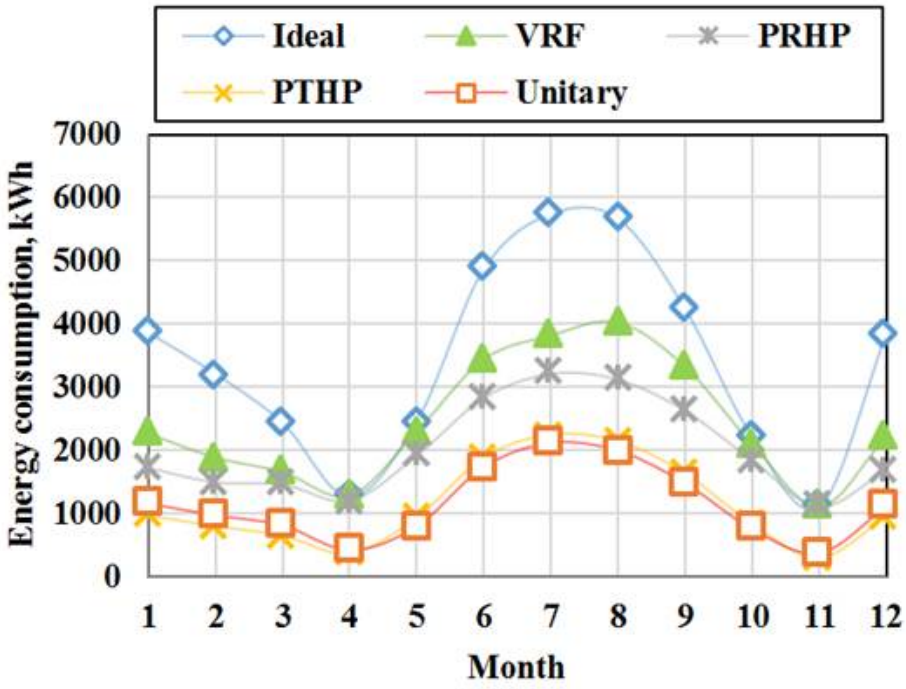


Figure 3.14: HVAC Energy Consumption vs Month of Year for 5 System Models [44]

3.4.2 2019 vs 2020 Building 3 Comparison

For Building 3, a historical analysis was performed to explore the affect of temperature on the traffic volume. Data was pulled from July 18 - August 7 of both 2019 and 2020 and is compared in Figure 3.15. No network changes were observed on the BAN between capture year 2019 and capture year 2020. Additionally as seen in Figure 3.10, Building 3 has a fairly consistent and stable traffic flow due to its small field bus network size. From Figure 3.15 we can see that the normalized traffic volume during 2019, prior to all the madness of the pandemic, is actually greater than the Ramp-Up time frame data

from the same building at approximately the same time of year in 2020. At this time during the year (July/August) the summer semester at the university would have still been underway in 2019 with students in and out, but in 2020 the university was only at 50% capacity. Furthermore, we can see from the plotted temperatures that there were minimal temperature differences in the city between the two years. Seeing this we can conclusively rule out weather for causing a difference in building traffic volume, and since there we no changes in the number of BA devices on the network, the next logical conclusion is that there was likely a small effect caused by the usage of the building (or lack thereof) during the 21 day interval.

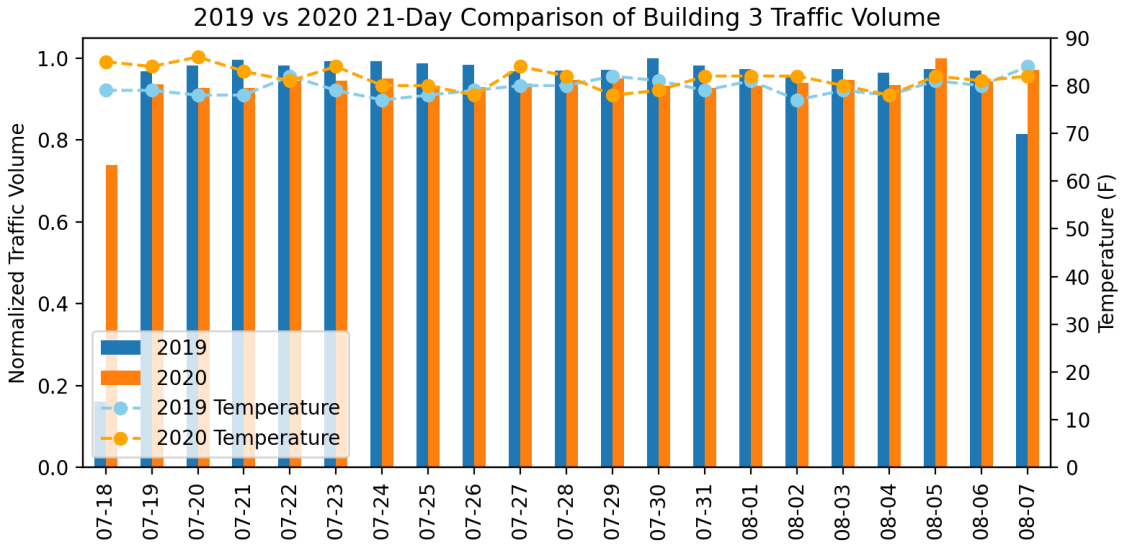


Figure 3.15: Building 3 Two Year Traffic Volume Comparison

3.4.3 Building APDU Type Trends

Next we will explore the distribution of the university campus building APDU Types. Figure 3.16 shows the full distribution of the weekly aggregated APDU Types for all the buildings studied. Additionally Figure B.3 in the appendix shows the data as a line plot from January to August. As expected the most prevalent and easily identifiable APDU Types are the Confirmed Requests and the corresponding Simple/Complex Acknowledgements.

Over the 8 month capture period nearly 700 million Confirmed Request packets were transmitted on the network. These triggered a resounding 670 million Acknowledgments. The 30 million difference can be accounted for between Error packets and messages that were never responded to. Combined Confirmed Requests and Acknowledgements (Complex, Simple, Segment) make up approximately 79% of the total communications seen over the whole dataset.

The next most popular APDU Type sent over the network, after Confirmed Requests and Acknowledgments, were Unconfirmed Requests. These were used heavily when *slave* devices wanted to tell the *master* controller about a local change that occurred. Unconfirmed Requests accounted for 19.5% of the total traffic transmitted in our studied dataset and were critical for keeping the controller up-to-date with current events and occurrences within each BA device.

The remaining 1% of traffic is made up of Errors (16.6 million), Rejects (78,000), Aborts (3,190), and Unknown (64,000) packets. There are 7 Error classes in BACnet [45], the first results from circumstances that affect the functioning of an entire BACnet device. The second relates to issues with identifying, accessing, and manipulating BACnet objects, both BACnet defined and otherwise. Even if this error is transmitted a service request could still be completed. The third error class deals with problems related to the properties of BACnet objects and service requests can still be completed if this error is thrown. The fourth has to do with problems of a BACnet resources that affect a device's ability to carry out a protocol service request. The fifth error class is rarely used and pertains to problem with protocol security. The sixth type of errors are considered fatal and signal the inability to execute a service request due to any issue with the service request itself. The last error class encompasses problems experienced with any virtual terminal services, this includes failures to open/close sessions or sudden terminations. Similarly, there are 10 Reject classes. They are all related to syntax errors, buffer overflows, invalid tags, incorrect number of arguments passed, undefined enumerations, unrecognized services, and incon-

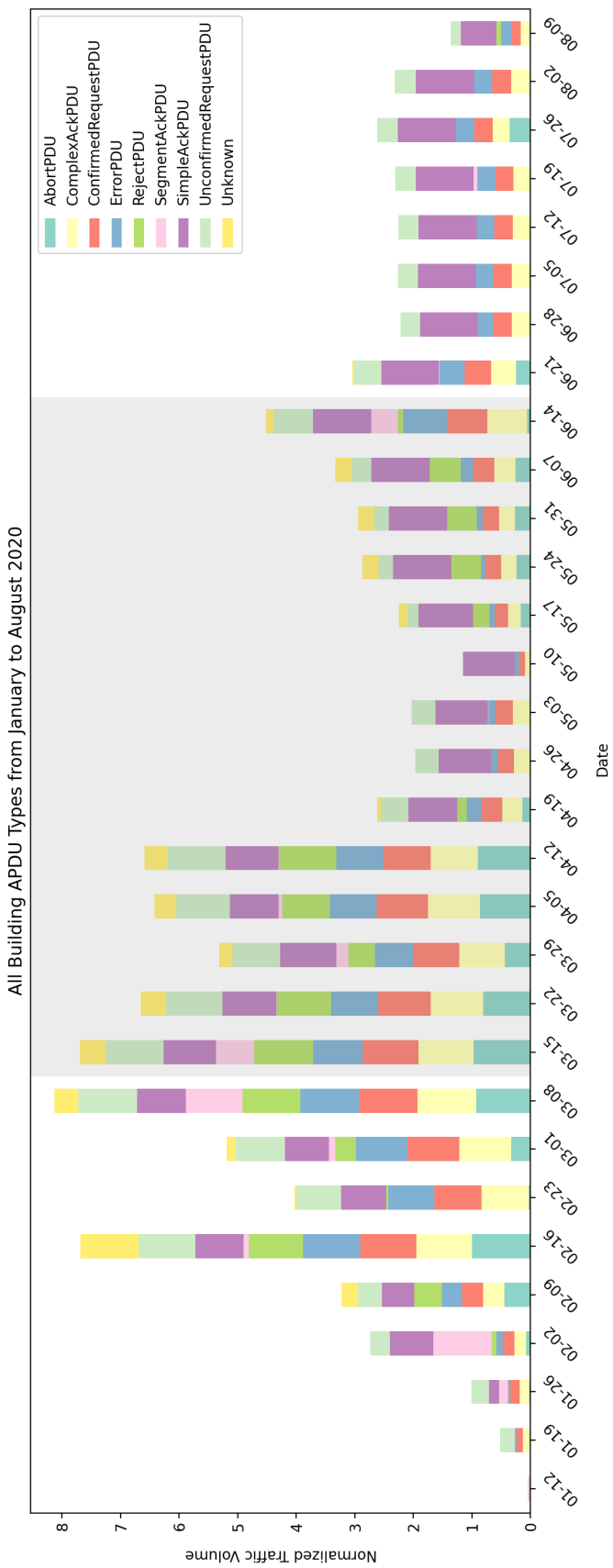


Figure 3.16: Buildings 1-9 Normalized Weekly APDU Type Traffic Breakdown

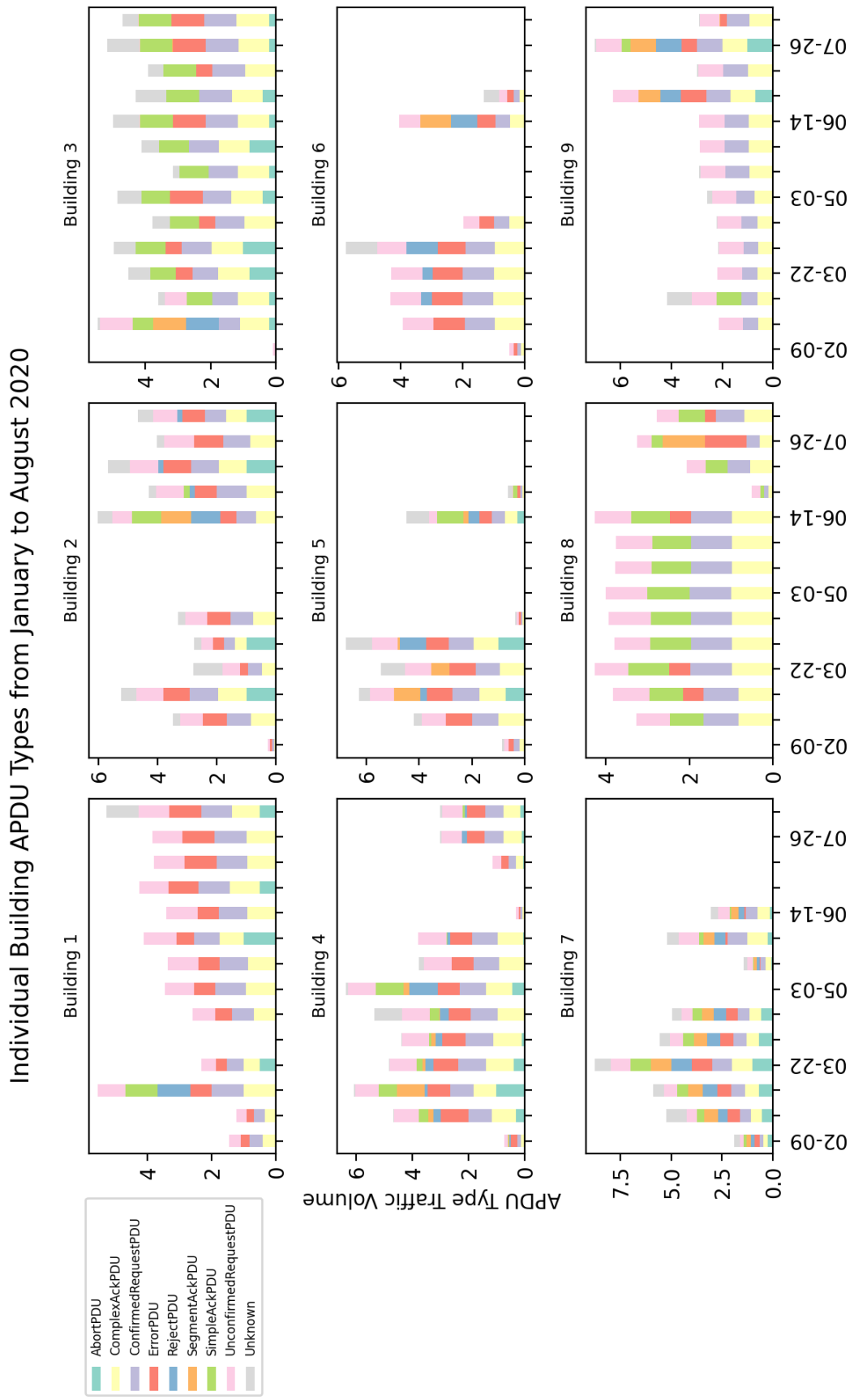


Figure 3.17: Individual Buildings Normalized Biweekly APDU Types

sistent/invalid/missing/out of range parameters. There are also 5 Abort classes defined in the protocol and these are more severe than the previously mentioned because they cause the device to quit communication altogether. Abort reasons could be due to buffer overflow, invalid APDUs, unsupported segmentation, higher priority tasks being handled, and or a miscellaneous issue.

Figure 3.17 shows the APDU Type distribution for each building in the dataset. We see most commonly among the buildings that Confirmed Requests and Complex Acknowledgments dominate most individual building traffic, which makes sense as that is what was seen in the all buildings APDU Type figure.

3.4.4 Building APDU Service Choice Trends

Similar to the last subsection, now we will look at the APDU Service Choices within the campus building dataset. Figure 3.18 shows the distribution of the APDU services choices from our university dataset over the 8 month collection period. Additionally, Figure B.4 in the appendix shows the full dataset APDU Service Choices by volume.

From the figures we see that 1 billion packets, nearly 58% of all the traffic observed on the network were Confirmed Private Transfers. These service choices are *heartbeat* messages that are sent from controller to field device to ensure the device is still online and functioning. At 19.5% of the total traffic volume, the next most popular service choice was Unconfirmed Private Transfers. Similar to the Confirmed Private Transfers, Unconfirmed Private Transfers are sent from the controller to the field devices to ensure its alive, but there are 2 differences. The first is the fundamental difference between Confirmed and Unconfirmed messages, which is the requirement of an Acknowledgement. The second is that the Confirmed Private Transfers are sent regularly, while the Unconfirmed are more random or sporadic in nature. This could possibly be due to the controller having not heard from a particular device in a time period and wanting to specifically follow up with it. With almost 340 million packets transmitted between 3 service choices (Read Property, Read

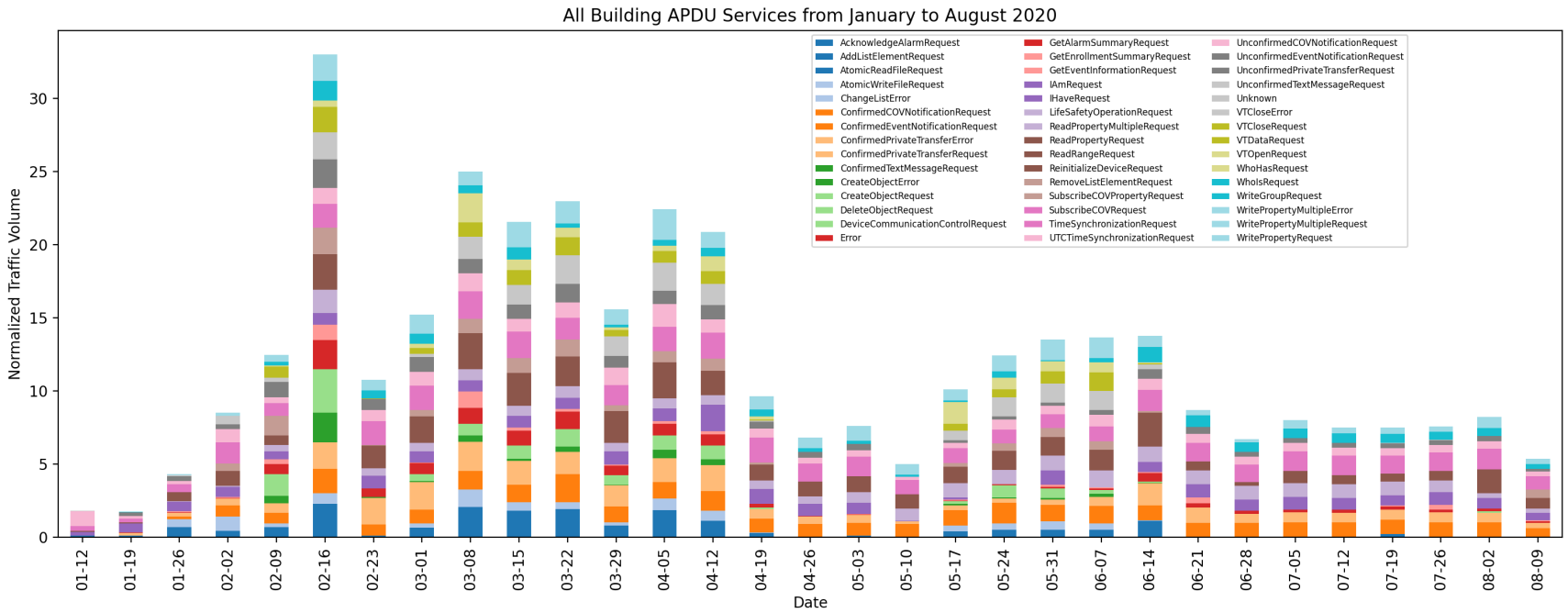


Figure 3.18: Buildings 1-9 Normalized Weekly APDU Services Traffic Breakdown

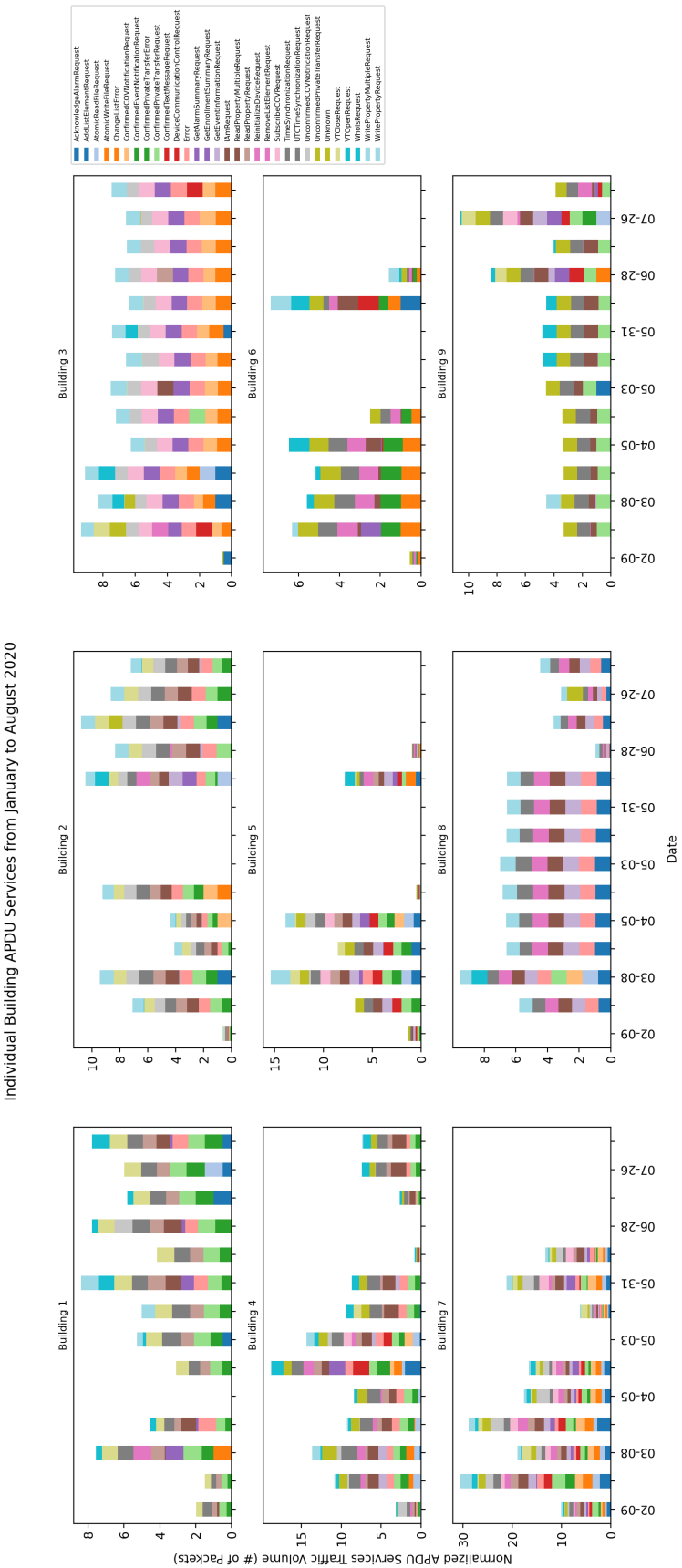


Figure 3.19: Individual Buildings Normalized Biweekly APDU Services

Property Multiple, and Read Range), another 19.5% of traffic were devices reading data points. Reads are particularly important on the university BAN because they are how data gets propagated from the field devices up to the management workstations where the building operators use building management software (BMS) to read them from a GUI. A much smaller ratio of traffic (1.9%) was dedicated to Confirmed COV Notifications, this traffic is the field device's way of informing the its subscribers that there has been an update in one or more of its object or property values. This assumes that interested devices on the network have *subscribed* to learn of events as they occur in specific BACnet devices. Once subscribed the COV (Change of Value) notifies the subscriber of the updated information, this is valuable because it gives field devices which are typically passive servers, the ability to report new events [46]. Of the 1% APDU Service Choice Error traffic, about half (8.46 million) are Confirmed Private Transfer Errors and the other half (8.15 million) are a generic Error type. This is significant, because it gives a scale for just how many of the heartbeat messages sent over the BAN result in error. These errors very likely stem from missed Acknowledgements expected by senders, hence why there are no Unconfirmed Private Transfer Errors. In our studied dataset a very small percentage of traffic (0.04% or 696 thousand packets) were dedicated to network discovery, from this we can insinuate that most devices on the network have been on the network for some time. This follows logic, because it is commonplace for new devices to send several network discovery messages as they join and try to fill their routing tables. For this dataset the low number of network discovery messages is on par with our observation of low device turnover rates where few devices were added and removed from the network over the data collection period. A few other very infrequently transmitted service choices of note were Time Synchronizations, Virtual Terminal Requests, and some Unknown message types (malformed packets).

Figure 3.19 shows the distribution of APDU Service Choices for each building. Some interesting findings were the high level of Errors transmitted, particularly in Buildings 3 and 8. Additionally the individual building APDU Service Choice figure highlight the non-

trivial presence of Reinitialize Device Requests in several buildings, namely Buildings 6, 7, and 8. It is no coincidence that these packets are seen in buildings with higher Errors, because it is usually manually triggered from an operator in an attempt to *refresh* a device.

3.4.5 Traffic Size Trends

In this last subsection we look at the packet sizes of traffic sent over the network. Figure 3.20 shows the average packet size per day in bytes for each of the 9 buildings in the dataset. The plot is visually similar to that of Figure 3.10, but the bars here indicate packet length. Overall our observations over the dataset show that packets of less than 50 bytes are the norm, but this will be heavily dependent on the packet payload.

For Building 1, 59% (36.8 million) of the packets transmitted over the data collection period were between 0 - 25 bytes. This is reflected in the average byte size shown in Figure 3.10. Packets sized from 25 - 50 bytes made up 22.5% of the traffic, 50 - 75 byte packets accounted for 12.5% of the traffic and the last significant packet size range was 5.4% from 75 - 100 bytes. There were overall very few packets from this building that exceeded 150 bytes. Upon further investigation of the very large packets, particularly the ones that exceeded 400 bytes, it was discovered that they were just response packets with large payloads. Buildings 6, 7, and 9 also exhibited very similar packet size trends to what was observed in Building 1. Each of these buildings also had the majority ($>$80%) of their traffic sized $<$50 bytes. One thing that makes Building 9 slightly different is that of that $>$80%, $\sim$50% are 25 - 50 bytes and $\sim$30% are 0 - 25 bytes. These smaller packets are normal on field bus networks and included periodic heartbeats, requests and responses for data, time synchronizations and more. An interesting finding in the Building 7 data however, was that it contained $\sim$25 packets which exceeded 1000 bytes. Upon investigation of this it was found that those packets were malformed or erroneous.

Buildings 2, 3, and 5 all had similar size trends with approximately 90% of the traffic sized less than 50 bytes. Of these packets about half were between 0 - 25 bytes and the

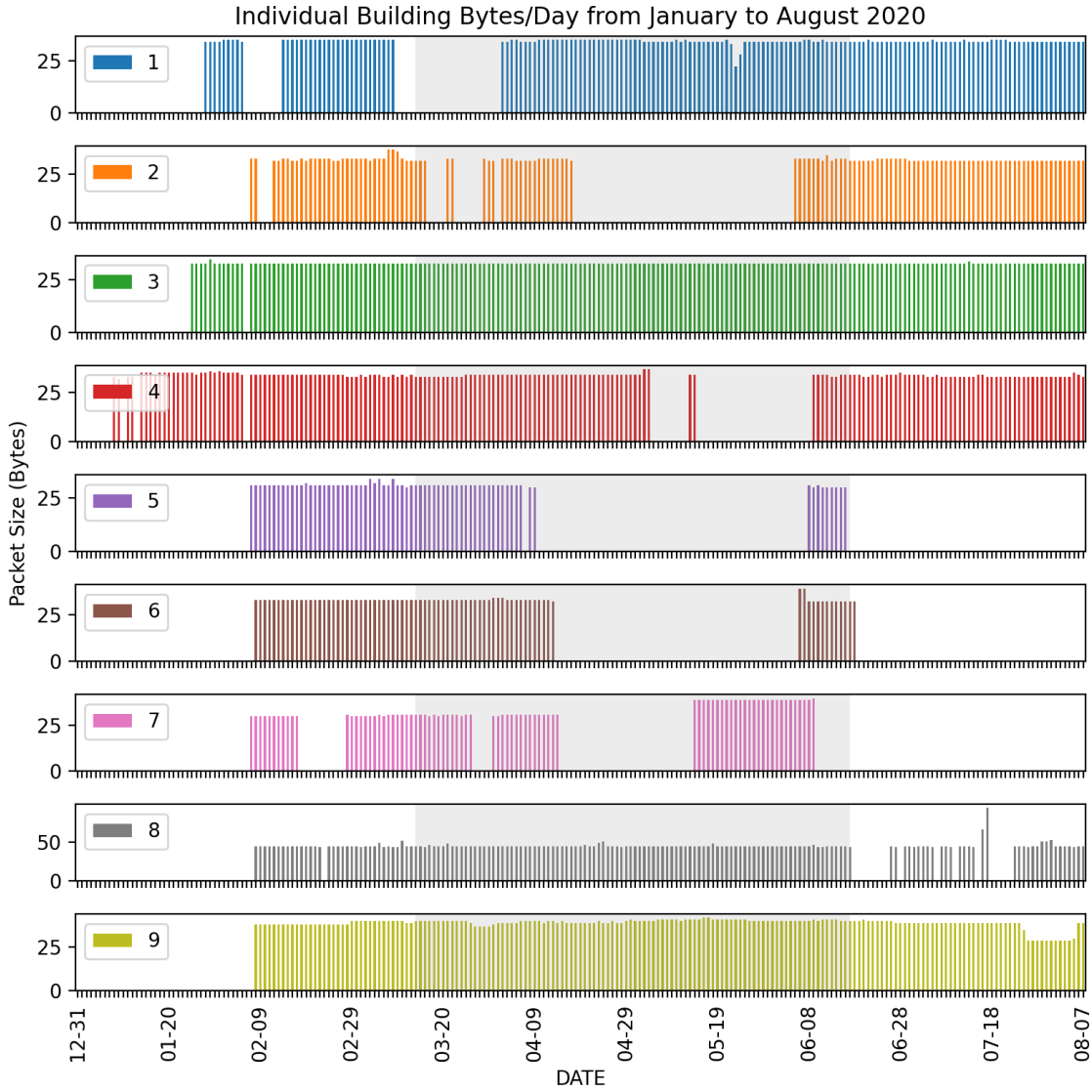


Figure 3.20: Individual Buildings Mean Packet Size Per Day from January to August 2020

other half were 25 - 50 bytes. Just like with Building 1, the Building 2 and 5 datasets had a few thousand very large (>400 byte) packets transmitted which again turned out to be a result of large payloads of building data from field device to controller.

Lastly, Buildings 4 and 8 had somewhat different packet size distributions over the entire data collection period than the other buildings. The Building 4 traffic, which was among the largest in our university campus dataset, consisted of about 80% packets sized less than 25 bytes. The other packet sizes transmitted in bulk were 25 - 50 bytes ($\sim$4%) and 50 - 75 bytes ($\sim$12%). Unlike most of the buildings in our dataset, nearly 40% of the Building 8 traffic were packets sized between 50 - 75 bytes. This is very visible in Figure 3.20 and can be attributed to the slightly larger than normal Read Property Multiple response packets sent from a particular field device to the controller. The remainder of the traffic ($\sim$60%) from this building were less than 50 bytes, which is normal in comparison with other buildings.

3.5 Outlier Detection Case Study

In order to accurately detect attacks on BASs we must first understand what normal or benign operation looks like. To do this we have trained machine learning models from the BAN traffic of Building 10 from our university campus dataset. Building 10 was selected for the case study because of its large size and composition. It is the only building within our dataset that exclusively uses BACnet for all its BA communications and it has a wide variety of physical devices. The case study data explored from Building 10 comes from August 16 - 22 of 2020, the first full week of campus reopening and the official start of the Fall 2020 semester.

The underlying premise of our case study was to develop building models from what is considered *normal or benign* traffic to see if we can successfully detect *abnormal or malicious* traffic flows. To accomplish this we extracted features (source and destination address, network controller number, packet length, APDU type, and APDU service choice)

from every packet. These features are described in subsection 3.3.3 in detail and once extracted were stored as a 12GB CSV file.

We then used three semi-supervised machine learning techniques (kNN, Isolation Forest, and CBLOF) to train 3 building models with the PyOD [47] python library for outlier detection. These models were selected for their popularity in the outlier detection domain and for their ease of use. For the purpose of this work, we considered our campus dataset as *normal or benign* and without anomalies. This is a commonly made assumption in this domain and is considered valid because our manual analysis offered no contradictory evidence. To ensure anomalies, as classified by highly unusual or otherwise invalid parameters, were included in the dataset we injected 8 classes of anomalies (inclusive of 2 of the most common BACnet attacks) into the Building 10 network traffic. The anomalies were designed as follows -

- Anomaly 1 - Spoofed the APDU Service Choice. Valid Service Choices as defined by the BACnet protocol should be between 0 and 31. The injected packets included values up to 50.

- Anomaly 2 - Spoofed the APDU Type. Valid APDU Types as defined by the BACnet protocol should be between 0 and 7. The injected packets included values up to 50. Additionally, the BACnet specification mandates APDU Type should correspond with the APDU Service Choice, therefore Unconfirmed types are always paired with Unconfirmed service choices etc. Some of our injected packets also violated this requirement with mismatched types and service choices.

- Anomaly 3 - Spoofed the source and destination address. Valid BACnet MS/TP addresses should be between 0 to 127. The injected packets included values from 0 to 256. In addition to this, we did not enforce uniqueness between source and destination addresses.

- Anomaly 4 - Spoofed the network controller number. The injected packet included

values for the controller number that did not exist truly exist on the network.

- Anomaly 5 - Spoofed the packet size. The injected packet may have contained very small or very large packets with sizes not seen before, but that do not exceed the maximum transmission unit (MTU) of 1500 bytes.
- Anomaly 6 - Combination of Anomalies 1 - 5.
- Anomaly 7 - Network Discovery attack. Stream of numerous remote device management messages, such as Who-Is, Who-Has, I-Have, and I-Am, for simulating a node performing network reconnaissance.
- Anomaly 8 - DoS attack. Floods the controller with what would be an inundating number of packets.

For our evaluation of each algorithm we divided the Building 10 data into 70% train and 30% test. The train data was injected with 1% anomaly type 6 traffic for shaping the possible variations of malformed packets and attacks. Since we manually injected the traffic, we labeled the anomalies as *malicious or class 1* and all other data as *normal or class 0*. For testing we injected different anomalies 1 - 8 and recorded both the precision (shown in Table 3.2, Table 3.3, and Table 3.4) and the ROC. Overall we observed high accuracy across each of the 3 building models developed when sufficient injected anomalies were present in the test dataset. In the following subsections we discuss our results from each machine learning algorithm.

3.5.1 k-Nearest Neighbors (kNN)

K-nearest neighbors is a model that classifies data points based on the points that are most similar to it. It uses test data to make an estimation of which label should be assigned to each unclassified data point [48]. Figure 3.21, shows the Precision-Recall Curve, ROC Curves, and Confusion Matrix for a test run with 20 injected anomalies of type 2. The

curves and Table 3.2 show that the kNN algorithm had the best classification performance of all 3 built models. Even with only 20 injected anomalies the precision was very high and there were almost no misclassifications. This came at the cost of training and testing time, the kNN model was routinely the slowest model to test.

Table 3.2: kNN Model Precision Results

# of Injected Anomalies	**A1**	**A2**	**A3**	**A4**	**A5**	**A6**	**A7**	**A8**
20	0.863	0.918	1	0.952	0.818	0.9	1	0.643
200	0.896	0.908	0.934	0.962	0.972	0.997	0.99	0.768
2000	0.903	0.984	0.929	0.932	0.912	0.973	1	0.966

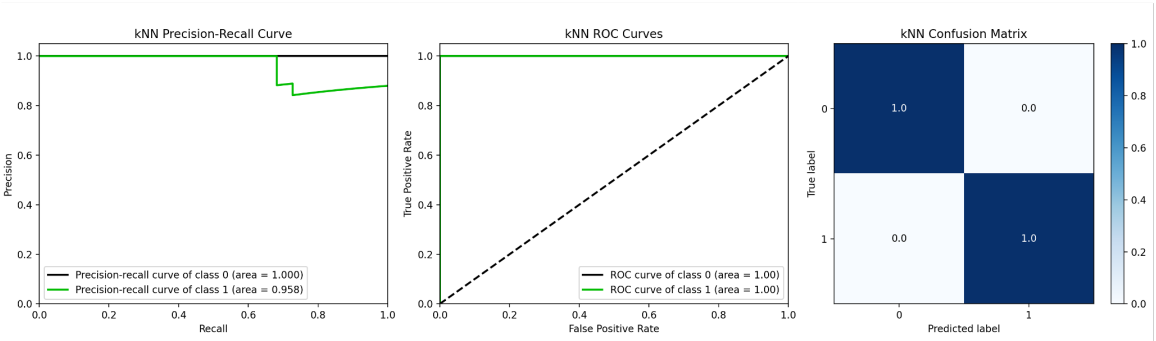


Figure 3.21: kNN

3.5.2 Isolation Forest (IF)

Isolation forests are commonly used in outlier detection problems and are built on the idea that anomalies are easier to identify through data partitioning than normal data. The algorithm essentially consists of a collection of decision trees from random subsets of the data and then aggregates the anomaly scores from each tree to determine a final anomaly score for a point [49]. Figure 3.22 shows the Precision-Recall Curve, ROC Curves, and Confusion Matrix for a test run with 20 injected anomalies of type 2. We can see from the figure the precision measurements in Table 3.3 that the number of injected anomalies greatly effected the model performance. Except in the case of anomaly 8, significant improvement was observed as the number of injected data points increased.

Table 3.3: IF Model Precision Results

# of Injected Anomalies	**A1**	**A2**	**A3**	**A4**	**A5**	**A6**	**A7**	**A8**
20	0.545	0.454	0.954	0.545	0.523	1	0.979	0.22
200	0.776	0.808	0.9254	0.813	0.828	0.995	0.99	0.269
2000	0.717	0.791	0.982	0.836	0.871	0.992	0.992	0.0898

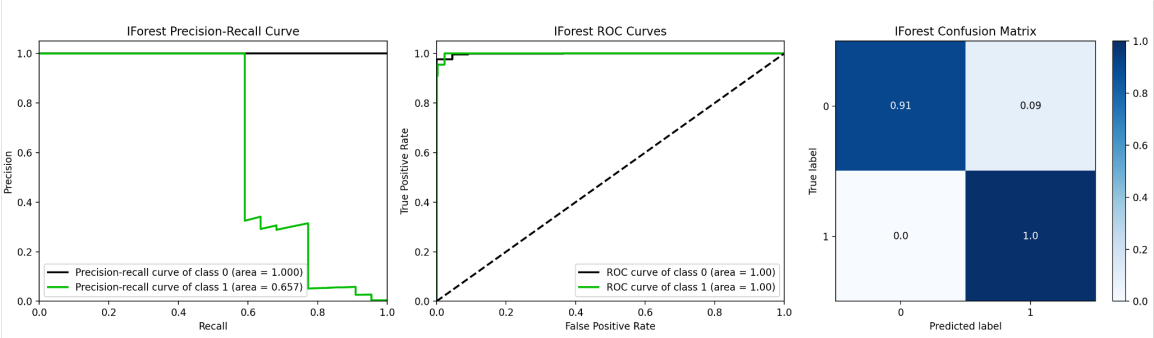


Figure 3.22: Isolation Forest

3.5.3 Cluster Based Local Outlier Factor (CBLOF)

CBLOF is an algorithm that calculates the outlier score based on the distance of each instance to its cluster center multiplied by the instances belonging to its cluster [50]. Figure 3.23 shows the Precision-Recall Curve, ROC Curves, and Confusion Matrix for a test run with 20 injected anomalies of type 2 and we see very low precision. This is a stark difference between the ROC analysis curves which are very close to 1. This difference can be attributed to the high class imbalance between *malicious* and *normal* traffic in the test dataset. In a real network the number of anomalies would be extremely low in comparison to the vast amount of regular traffic so it is important that our building models be able to make a distinction in the presence of very few anomalies.

Table 3.4: CBLOF Model Precision Results

# of Injected Anomalies	**A1**	**A2**	**A3**	**A4**	**A5**	**A6**	**A7**	**A8**
20	0.047	0.090	0.190	0.227	0.952	0.954	0.56	0
200	0.454	0.416	0.666	0.544	0.956	0.977	0.95	0.372
2000	0.844	0.8423	0.933	0.8704	0.989	0.989	0.980	0.937

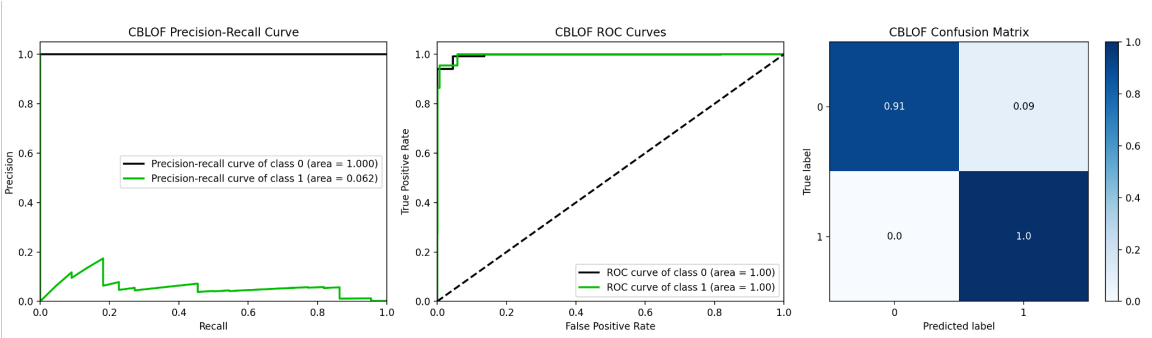


Figure 3.23: CBLOF

3.6 Conclusion

In this chapter we presented the first and most detailed building automation network traffic analysis at the field layer. We examine the effect of the COVID-19 shutdown on a large university campus dataset, consisting of 9 buildings, over the course of 8 months from January to August of 2020. In our analysis we found that there were 2 types of buildings on the campus, static and dynamic. The static buildings were driven more by their pre-programmed schedules than the building environment and therefore had fairly consistent traffic flows from the Pre-COVID to Ramp-Up time periods. The dynamic buildings on the other hand, in addition to their pre-programmed schedules, reacted to feedback from the environment and as a result differences in traffic could be noted from the 3 time ranges. In addition to this, we successfully trained 3 machine learning models on a subset of data from our university building dataset to detect anomalies in the traffic and observed high accuracy predictions. Our work has shown that machine learning can be leveraged to precisely detect anomalies in BASs. We also uncover that due to the generally predictable nature of these networks even small datasets can be used to generate very robust models.

CHAPTER 4
UNDERSTANDING AND EVALUATING BUILDING AUTOMATION SYSTEM SECURITY

4.1 Introduction

In the early days of building automation, BASs (sometimes referred to as Building Management Systems or BMSs) consisted of hardwired centralized systems which primarily interfaced with one or two isolated subsystems through pneumatic sensors/controllers and direct digital control (DDC) [1][51]. As technology has advanced the prominence of data communication protocols and networked systems has allowed process controllers to be placed on individual devices which can transmit data back to the centralized system. At the start of the Internet age there was another evolution where centralized controllers began being connected to the Internet. In modern day smart buildings, a lot of the central control automation has been dispersed to cloud endpoints where the information can be accessed from nearly any connected computer.

Traditionally, the main purpose of BASs has been to assist facility managers in building administration, increase occupant comfort, and conserve energy. Formerly, this was accomplished through providing a few basic core services like Heating, ventilation, and Air Conditioning (HVAC), lighting and shade control, and sometimes even elevator management. Other safety and security oriented services, such as fire and life safety systems, access control, and alarm intrusion systems, were typically operated and managed by completely separate systems [52]. Today, with demands for more building control, less operation costs, and higher energy efficiency/self-sustainability the industry has evolved towards smart or intelligent buildings.

Much like smart homes, smart buildings are more connected than previous genera-

tions of buildings and introduce added levels of intelligence through the inclusion of smart devices (smart sensors, smart actuators, etc). They allow for more seamless control and monitoring by building managers and owners. In [51] smart buildings are defined as buildings equipped with integrated technology systems such as building automation, life safety, telecommunications, user systems, and facility management systems. As seen with nearly all major technological advancements, the progression of the BAS domain has come at the cost of an increased threat surface and higher magnitude security breach repercussions. According to [53] the building automation system (BAS) market is forecasted to grow from 75 billion USD in 2019 to 121.5 billion USD in 2024, a time span of just five years. This growth will be due largely to the swift penetration of the Internet of Things in BASs, the increased focus on energy efficient and eco-friendly buildings, and the growing infrastructure demands of developing countries. Even IoT for smart buildings in the global market is expected to expand from $6.3 billion in 2017 to $22.2 billion in 2026 [54]. As the domain grows so have the attacks vectors which enable the exploit of BAS vulnerabilities and in the last few years the amount of attacks against BASs have escalated without signs of slowing down.

Researchers hacked the Google Australia building control system in 2013 due to Google's failure to patch their system [4]. Previously found vulnerabilities in the Tridium Niagara AX platform were leveraged by the researchers to obtain the admin password for the BMS. Then again in 2018, a Google employee successfully hacked into the Google Sunnyvale access control system when he noticed the devices were all using the same hard-coded encryption key [5]. In early 2019 attackers targeted the Nortek Security & Control's Linear eMerge E3 access control system and used it to launch DDoS attacks [55]. These aforementioned attacks barely scratch the surface and do not even begin to address the host of indirect vulnerabilities that surface from the interconnection of BASs to traditional IT networks. One such example is that after thorough investigation of the Target hack [56], which exposed about 110 million customers, in the November-December time frame of 2013 it

was revealed that the attackers made the initial intrusion through network credentials stolen from an HVAC vendor [6].

The assurance of smart buildings which are globally becoming commonplace, starts with securing of the BAS domain. As is typically the case, secure design has not been accounted for during the evolution of BASs and there are still massive amounts of trust placed on poorly integrated legacy devices and undertrained staff. Given that the life cycle of BASs greatly exceed traditional IT systems and require high availability, the undertaking of security is too much for one party to bear alone. It requires a combination of knowledge from both building automation experts and system security researchers to sufficiently fortify buildings which will ultimately protect not only assets, but human lives.

The goal of this work was to survey and systematize the literature in BAS security for analyzing the security properties of building automation devices and deriving an evaluation framework based on the vulnerabilities, attacks, and defenses proposed in the literature, for characterizing device security posture. We built a multi-protocol testbed consisting of devices from each of the three most prominently deployed BAS communication protocols (BACnet, KNX, and LonWorks) to evaluate our proposed framework for measuring device security. Our building automation testbed is the largest of its kind referenced in the literature to the best of our knowledge. From our systematization we uncover insights and determine the open and neglected research areas as well as discuss the issue of liability in BAS security. The major contributions of this chapter can be summarized as follows:

- Systematize the research literature in the BAS security domain to understand the scope of exploitable vulnerabilities, attack techniques, and proposed countermeasures.

- Leverage the systematization to propose a framework for evaluating the security posture of building automation (BA) devices from the network and device perspectives. The framework gives building automation system operators a reproducible means for measuring the security assurredness of their deployments.

- Investigate the proposed evaluation framework methodologies we build and perform a security assessment on the largest multiprotocol BAS testbed discussed in the literature, to the best of the authors knowledge. The testbed contains devices from three of the most widely used BAS protocols, BACnet, LonWorks, and KNX.

- Uncover several side channel vulnerabilities never discussed previously in the literature. Additionally our analysis shows a significant research gap in the vulnerability assessment of the BA devices from the firmware/software and side channel points of view.

- The results from the systematization and testbed evaluation are analyzed in detail and the findings are explained for possible future research.

The remainder of this chapter is organized as follows, in section 4.2 we give background on the field of building automation, the communication standards, and protocols. In section 4.3 we discuss our systemization methodology, evaluation objectives, and threat model. In section 4.4 we breakdown the literature in the BAS domain by layer and in section 4.5 and section 4.6 we discuss our proposed framework and testbed evaluation. Finally in section 4.7 we conclude and discuss interesting findings as well as open challenges in the domain.

4.2 Background

A building automation system is, generally speaking, a distributed computer based controls platform used for monitoring and managing electrical and mechanical building equipment. Modern day BASs may consist of several interconnected or isolated components, including HVAC systems, access control systems, fire and life safety systems, lighting control systems and more. BAS applications can range from large custom built environments to smaller off-the-shelf deployments. A key challenge of these systems is balancing the optimization of operational costs while reducing energy consumption and maintaining user

comfort. All the while aiming to improve precise, intelligent autonomous building control and integrate various subsystems. To facilitate these goals building automation networks are routed with communications from all manner of field devices, controllers, and human machine interfaces (HMIs).

In terms of security BASs differ from traditional IT systems in many way. In [57] four key unique characteristics that make securing BASs difficult are identified -

- Limited Resource Availability - Sensors and actuators may be energy constrained
- Diverse Topologies - Network structure may vary by building and security solutions need to be adaptable
- Physical Access - Devices may be reasonably accessed by an adversary and thus sensitive data could get leaked
- Continuous Maintenance - Device and system maintenance are critical because firmware patching and upgrading can be essential to proper operation, but should be upgraded in secure manner to avoid compromise

Compared to traditional ICSs or CPSs, smart buildings are much more “open”, accessible, and interconnected. From a networking standpoint the BAS architecture can be broken down into three levels, management, automation and field device [39], an example architecture is shown in Figure 3.2 and discussed further in subsection 4.2.1.

4.2.1 Building Automation Network Architecture

The field level is where physical interaction with the environment takes place, these are sensors and actuators that monitor and control equipment. The automation level operates on data from the field level and handles the automation logic and control statements. At the management level remote operator intervention takes place and HMIs report relevant network and system details [39].

The lowest tier of Figure 3.2 represents the field level. Field devices, typically small and/or embedded, control and measure the physical world. This level is made up of sensors and actuators for performing local operations, like valve opening and closing. Networked field devices communicate most commonly via wired signals and protocols, but can also be wireless. Some examples of field device sensors are occupancy sensors for detecting motion and presence, temperature sensors for measuring heat energy, and photosensors for detecting light. Common actuators used in BASs include locks for access control and fans for HVAC.

The middle tier of Figure 3.2 represents the automation level. This level contains controllers which gather data from the field devices and drive the system actuators based on those signals as well as the programmed control technique. Devices used at the automation level include Programmable Logic Controllers (PLCs), supervisory controllers, and data acquisition units. Automation devices conduct distributed control tasks through BAS network protocols and can even perform routing to field devices.

The topmost tier of Figure 3.2 represents the management level. Most of the monitoring and configuration from the building administrator/manager perspective occurs here and information from the entire BAN is available. IP networks are primarily used at this level to send and receive instructions to and from the automation level. Management devices include workstations, HMIs, and other computing devices used to facilitate the cohesion between operators and systems.

Through the BAN, devices diligently work together performing operations to maintain the facility. Just as buildings vary in size and function, so do BAS control components. They come from assorted vendors and often times use differing communication technologies. Given the long length of the average BAS life cycle, BANs tend to consist of several different legacy systems and subsystems. The networking of diverse devices and communication standards oftentimes presents a complex challenge to holistic building automation security.

4.2.2 BAS Industry Standards and Protocols

In modern building automation systems the primary wired protocols used are BACnet, LonWorks, and KNX. According to a recent study [58] (report included 21 countries from North/Latin America, Europe, Asia, and the Middle East), BACnet is the leading BAS protocol with over 60% market share globally. It is followed by KNX, which is on the rise, and LonWorks/proprietary protocols, which have been on the decline as the demand for open protocols has increased (shown in Figure 4.1 [58]).

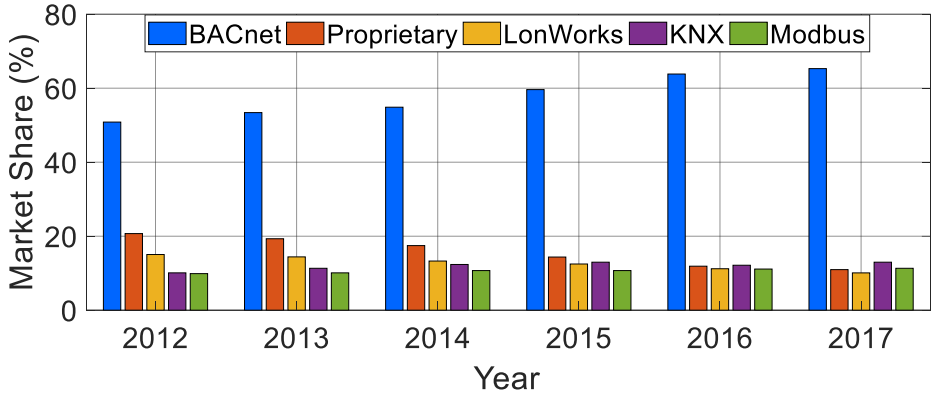


Figure 4.1: Global BAS communication protocol market share from 2012 to 2017

BACnet

BACnet, previously introduced in section 3.2, is a non-proprietary open communication standard. BACnet is essentially an unconnected peer network where any device can send service requests to any other device. One of its highlights is that it was designed to embrace object modeling and as such a physical device is characterized by a group of standardized objects. BACnet devices must communicate to one another over a network. The most popular implementations or *flavors* are BACnet/IP and BACnet MS/TP. Figure 3.3a [59] shows the BACnet protocol stack, consisting of an application, network, and data link layer.

KNX

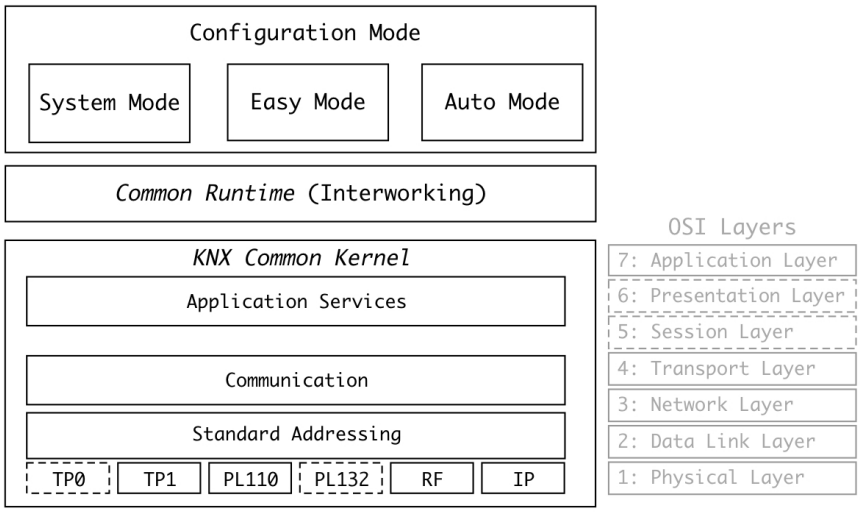
The KNX standard was created in 1999 by the Konnex Association, presently known as the KNX Association. It formed from a combination of three now deprecated standards: European Home Systems Protocol (EHS), BatiBUS, and European Installation Bus (EIB or Instabus) [60]. While KNX is primarily deployed in the European building automation markets, the standard's global presence is rapidly growing.

KNX has been used in residential and commercial building automation deployments for HVAC, lighting, security, remote access, blind and shutter control, visualization, and energy management. Some features that make KNX a prominent contender in the building automation space are its standardized commissioning procedures, vendor independent applications, tree topology capable of supporting large networks, backwards compatibility with EIB, and wide variety of transmission media.

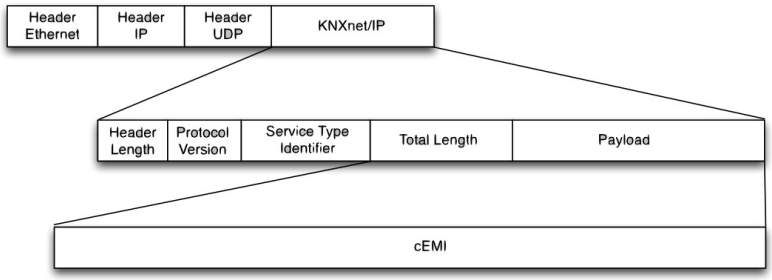
KNX can operate over twisted pair (KNX TP), radio frequency (KNX RF), IP (KNX/IP), or power line (KNX PL). KNX TP is an open standard in which both power and data run over the bus cabling to each node with a recommended maximum length of approximately 1,000m/3,280ft. Information is transmitted between bus devices via telegrams which consists of 4 fields. KNX TP telegrams contain a 1 byte control field, 5 byte address field, up to 16 byte data field, and a 1 byte checksum. The KNX protocol stack and an example IP frame are given for reference in Figure 4.2 [61]. In many modern installations KNX TP is used in conjunction with KNX/IP which allows the tunnelling and/or routing of KNX frames encapsulated within IP frames. The less commonly implemented KNX PL is transmitted over an existing 230V mains network and KNX RF is transmitted via radio signals (either uni or bidirectionally).

LonWorks

The development of LonWorks (Local Operating Network) began by the Echelon Corporation in 1988 with the goal of making an easy and cost effective way to build open control



(a) KNX Protocol Stack



(b) KNX/IP Frame

Figure 4.2: KNX Protocol Diagrams

systems. The majority of LonWorks devices play a role in buildings projects from various domains like HVAC, lighting, but are also used in many other markets such as transportation, utility, process control, and home automation [60]. LonWorks supports several physical layer technologies (twisted pair, power line, fiber optic, wireless) and includes security features that provide authentication and confidentiality. Though these rarely are configured in real BAS operational environments [51].

LonWorks is a peer-to-peer network with no single master, so control devices communicate directly with one another. Addressing is typically broadcast to all devices on the network through binding at the time of network commissioning using specialized network management tools. The most common kind of communication transmitted on LonWorks

networks are network variable updates [62]. A network variable is a data element that may consist of a single value or a collection of data. Typically network variables are collections of data where the data can be represented in over 170 varying data types known as Standard Network Variable Types (SNVTs). The data type depends on the variable being transmitted and is scaled according to the SNVT definition.

Much like the OSI model from traditional IT networking, LonWorks has 7 layers, physical, link, network, transport, session, presentation, and application, each of which provide key services. Figure 4.3 [59] shows a standard Lon protocol frame sent from node to node.

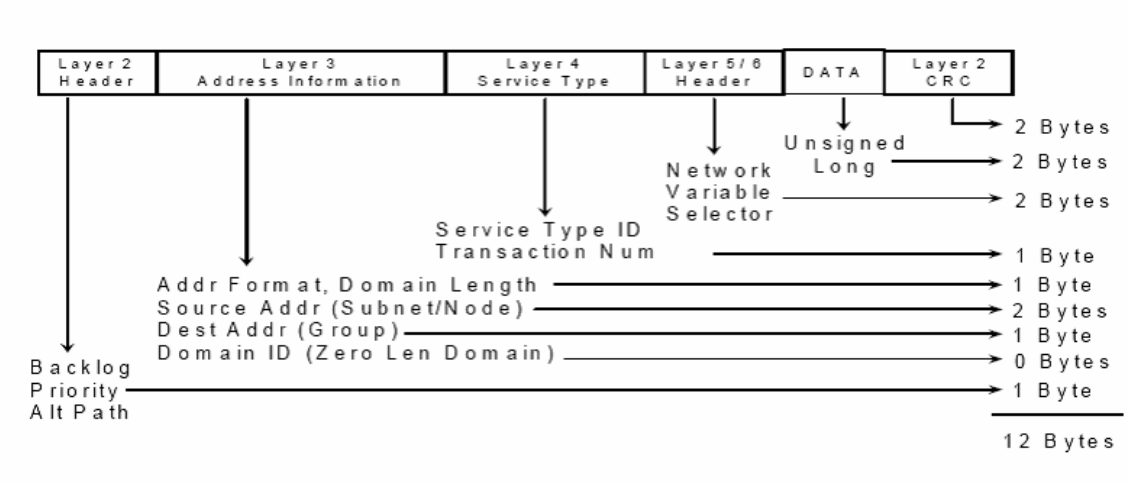


Figure 4.3: LonTalk Protocol Frame Format

Other Protocols

In addition to BACnet, KNX and LonWorks there are several other communication protocols and standards used in the BAS domain. Some examples are Modbus, DALI, Zigbee, and EnOcean. These protocols are out of the scope of this work due to their very small market share of currently installed building automation system deployments [58].

4.3 Evaluation and Systemization Overview

4.3.1 Systematization Literature Selection

The literature reviewed for the systematization was selected based on the following criterion -

- Disruption - The research touches on a hot area that is of high interest to the community
- Impact - The research is regarded as significant based on the quality of the contribution and/or the number of citations
- Merit - The research is one of first to uniquely explore a given security exigency
- Scope - The research focuses on building automation system security

4.3.2 Literature Review Taxonomy

In our exploration of BAS security literature we found that related work can be taxonomized by five criteria: layer, protocol, target, attack, and countermeasure. The remainder of this section discusses each of them in detail.

Layer refers to the building automation network layer, this can be either management, automation, or field. For the purposes of this systematization, each paper is evaluated by the layer of its key contribution.

Protocol is the BAS communication protocol(s) discussed in the work. This may be BACnet, KNX, LonWorks, etc.

Target is the subject of the attack(s) or countermeasure(s) discussed or proposed in the work. Targets are sub-classified into five categories: management network (MGT NW), field network (FD NW), management device (MD), interconnection device (ICD), or field device (FD). The MGT NW refers to the communication link, most commonly an IP backbone, between nodes on the management layer. The FD NW is the network media (twisted

pair, power line, radio frequency, etc.) used to connect field devices. MDs are the workstations and other devices used at the management level to monitor and control the BAS. ICDs are nodes, such as gateways and routers, that physically link the BA network layers. FDs are the sensors, actuators, and controllers that 'do work' and report back up the chain. Figure 4.4 shows an illustrative example of each of the five targets where the referenced target is highlighted in red.

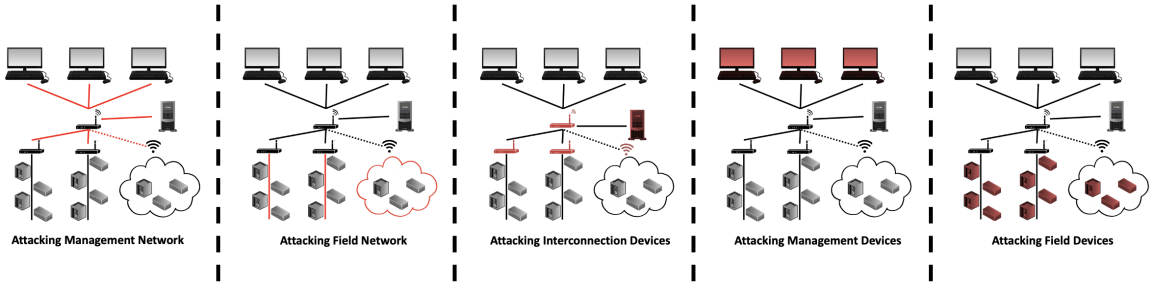


Figure 4.4: Five Types of Building Automation Systems Targets

Attack can be subdivided into network and device attacks. Network attacks are sub-classified even further into four categories: interception (attacks on network confidentiality), modification (attacks against data integrity), injection (fabrication or attacks against authentication and authorization), and interruption (attacks against network availability). Device attacks are sub-classified into three categories: software (attacks which exploit weakness in a software implementation), physical (manipulation of hardware or physical intrusions), and side-channel (attacks which exploits observed system parameters)

Countermeasure refers to the action taken to counteract, defend, or mitigate a threat. This category is sub-classified into three categories: prevention (techniques and strategies to stop threats before they cause harm), detection (methods or techniques to inform relevant parties of security breach before, after, or while they occur), and hybrid (solutions which offer any combination of attack prevention and detection).

4.3.3 Attack and Defense Taxonomy

The attack and countermeasure categories can be further categorized by the attack vector, attack impact, defense technique, and liable party for defense implementation.

Vector is divided into six classes of attack vectors referenced in the literature for gaining access to BASs. First are insecure services, this includes devices running insecure programs on common ports, using weak passwords or default configurations, etc. Next is the utilization of insecure or unpatched legacy systems and software. These are resources that have become compromised due to lack of updating, upgrading, or patching. Another attack vector is the compromise of hardware or firmware, this includes physical attacks (vandalism and theft) as well as chip or firmware application modifications. Then there is protocol insecurity which could be due to inbuilt protocol vulnerabilities such as the lack of confidentiality, integrity, and authentication mechanisms or poor cryptography key management policies and procedures. The next attack vector stems from human error and can be the result of poor security training, system misconfigurations, social engineering, or even bad security practices in general. Finally denial of service (DoS) can be used against BASs to overwhelm resources and perform a malicious payload.

Impact has four classes that can be ascertained from the research literature. The first is physical damage where a device or network resource is destroyed. Next is data leakage (information exfiltration) or corruption (information is destroyed or made unusable). Then there is loss of network access and/or privilege loss, for example if surveillance equipment is compromised or if root access is reconfigured on a MD. The final class is financial loss, where economic resources are affected by the attack.

Defense, just as with the six attack vectors, is comprised of six defense techniques observed from the BAS security literature. First were intrusion prevention systems (IPSs) for assuring BA networks and devices in attempts to elude attack. Next were intrusion detection systems (IDSs) for informing facility managers and system administrators of abnormal activity. Cryptography was also a defense employed, whether it be through the develop-

ment of a new cryptoanalysis techniques or the improvement of existing algorithms re-imagined for BASs. Along the same line new and improved BAS protocols were proposed which address vulnerabilities is current implementations. Another defense is the use of secure software, this includes software protection techniques which aim to prevent vulnerabilities and hinder attacks. Finally the last class of defense was general advice to follow guidelines of best security practice. These typically listed security recommendations to follow as their main contribution, some examples are upgrading physical security to increase tamper-proofness, staying up-to-date with patching, and implementing firewalls.

Scope of Liability corresponds to the liable party for each defense, just as impact corresponds to attack. We found that there can be six responsible parties when it comes to BA defense implementation. These are the entities who should take action to realize the defenses proposed in the literature (or be held accountable for the consequences of attack). The first party are system integrators, these are the people hired to monitor and install BA devices and configure networks. Then is the building occupant, person or persons who utilize building services. Next is the building owner, followed by the facility manager (group that controls and directs building operation). Then are the BAS manufacturers who develop the BA technology (devices, protocols, etc). Finally are the vendors and distributors who sell the BA devices, software, and systems.

4.3.4 Evaluation Objectives

In section 4.5 we propose a BAS evaluation framework which offers a systematic means for measuring the security assurredness of BA devices from the network and device level. Our framework will be useful for facility managers in need of evaluating the security posture of their existing BA deployments as well as new deployments in order to incorporate security from early project stages. Fundamentally the research questions (RQs) we aim to explore with our evaluation framework are:

- **RQ1 -** Can the security posture of BASs be measured?

- **RQ2 -** What are the core assessment criteria to consider when exploring BAS security?
- **RQ3 -** How does BA device interoperability affect the overall building automation system's security?

We answer these questions by experimenting and evaluating our framework on the largest multiprotocol BAS testbed discussed in the literature.

4.3.5 Threat Model

The threat model considered in this work are remote attackers with IP access to the BAN, as well as local attackers with limited physical access. Given the nature and accessibility of *smart buildings*, it is imperative to consider means by which intruders can use non-networked approaches to affect BAS security.

4.4 Systemization of Knowledge

This section presents a systematization of building automation system security research with papers spanning over the last 20 years. Table 4.1 gives a breakdown of each of the works categorized by the building automation layer of their main contribution. Each paper is classified according the protocol(s) it addresses, the attack or countermeasure type, and the target of the attack/countermeasure. Table 4.2 the attack vectors, impacts, defense mechanisms, and liable/responsible parties for each work are highlighted. The remainder of this section discusses the literature in detail. Note, this systemization focuses on representative work of the field and does not depict the field in totality.

4.4.1 Management Layer

Most BAS security research has been performed at the management level because its the most easily accessible. Remote access to BA deployments must occur from this level and

Table 4.1: Systematization of the Current Literature by Building Automation Layer

Layer	Reference	Protocol			Target					Attack							Countermeasure		
										Network				Device					
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Management	[63]	✓	✓		✓						✓						✓		
	[64]		✓					✓	✓	✓	✓						✓		
	[65]	-	-	-	✓	✓				-	-	-	-	-	-	-	✓		
	[66]	-	-	-	✓	✓	✓		✓				✓	✓	✓		-	-	-
	[67]	✓			✓	✓			✓	✓			✓	✓				✓	
	[68]	-	-	-	✓	✓			✓					✓	✓				✓
	[69]	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓		✓				✓
	[70]	✓			✓		✓			-	-	-	-	-	-	-			✓
	[12]	✓	✓	✓	✓					✓	✓	✓	✓	✓	✓	✓	✓		
	[71]	✓	✓	✓			✓		✓					✓	✓		✓		
	[72]	-	-	-	✓	✓			✓	✓	✓	✓	✓				✓		
	[73]		✓	✓	✓		✓			-	-		-	-	-	-	✓		
	[74]	✓			✓			✓		✓							✓		
	[75]	✓					✓		✓					✓	✓	✓	✓		
	[76]	-	-	-					✓	-	-	-	-	-	-	-		✓	
	[13]	-	-	-	✓	✓	✓	✓	✓	✓				✓	✓		✓		
	[28]	✓			✓		✓			✓			✓					✓	
	[77]	✓	✓		✓					-	-	-	-	-	-	-	✓		
	[78]	✓	✓		✓					✓							✓		
	[79]	-	-	-	✓			✓	✓		✓	✓	✓				✓		
	[80]	-	-	-	✓		✓		✓	✓		✓					✓		
	[81]	-	-	-	✓								✓	✓			-	-	-
	[82]	✓			✓	✓					✓	✓	✓					✓	
	[83]	✓			✓	✓				-	-	-	-	-	-	-		✓	
	[84]	-	-	-	✓		✓						✓				✓		
	[85]		✓		✓				✓	✓	✓	✓					✓		
	[35]	✓			✓	✓			✓	✓	✓	✓	✓					✓	
	[86]	-	-	-			✓			-	-	-	-	-	-	-	✓		
	[16]	✓				✓			✓	✓	✓		✓					✓	
	[87]	-	-	-	✓				✓		✓		✓		✓		-	-	-
	[88]	-	-	-	✓	✓			✓		✓	✓	✓		✓		✓		
	[89]	✓	✓	✓	✓					✓			✓	✓			-	-	-
	[90]	-	-	-	✓					✓	✓	✓	✓		✓		✓		
	[91]	✓			✓				✓				✓				✓	✓	
	[92]	✓			✓				✓	✓	✓		✓					✓	
	[93]	-	-	-	✓					-	-	-	-	-	-	-	✓		
	[94]	✓			✓	✓				-	-	-	-	-	-	-		✓	
	[95]	-	-	-	✓			✓		✓			✓					✓	
	[96]	-	-	-	✓					✓							✓		
	[97]	-	-	-	✓		✓		✓	-	-	-	-	-	-	-		✓	
Automation	[98]	-	-	-				✓		-	-	-	-	-	-	-	✓		
	[99]	-	-	-					✓	-	-	-	-	-	-	-		✓	
	[57]	-	-	-		✓			✓		✓	✓	✓	✓	✓				✓
	[15]	✓				✓				-	-	-	-	-	-	-		✓	
	[29]	✓				✓	✓		✓				✓				-	-	-
	[100]	✓			✓	✓			✓		✓		✓					✓	
	[101]	-	-	-	✓				✓	✓			✓	✓	✓			✓	
	[102]	-	-	-			✓		✓		✓						-	-	-
	[103]	✓				✓			✓	✓	✓	✓	✓					✓	
	[104]		✓						✓	-	-	-	-	-	-	-	✓		
	[105]	-	-	-	✓	✓				-	-	-	-	-	-	-	✓		
	[106]		✓		✓	✓		✓				✓					✓		
	[107]	✓			✓	✓				✓	✓							✓	
	[108]	-	-	-	✓				✓	✓	✓	✓					✓		
	[74]	✓			✓	✓		✓		✓	✓	✓	✓		✓			✓	
	[109]	✓	✓	✓	✓	✓				-	-	-	-	-	-	-	✓		
	[110]	-	-	-		✓			✓			✓	✓					✓	
	[111]	✓			✓	✓				✓		✓	✓				✓		
	[112]	✓	✓			✓	✓		✓	✓	✓		✓				✓		
	[113]	-	-	-		✓			✓	✓			✓				✓		
Field	[114]	-	-	-		✓			✓	-	-	-	-	-	-	-			✓
	[115]	-	-	-					✓		✓	✓					-	-	-
	[116]	-	-	-					✓		✓	✓	✓			✓	✓		
	[117]		✓			✓			✓	✓							-	-	-
	[118]	-	-	-	✓	✓	✓		✓	-	-	-	-	-	-	-	✓		
	[119]		✓		✓				✓						✓		✓		
	[120]			✓	✓				✓	-	-	-	-	-	-	-			✓
	[121]	-	-	-		✓			✓	✓			✓		✓		✓		
	[122]	✓	✓	✓					✓	-	-	-	-	-	-	-	✓		
	[123]	✓				✓				✓	✓			✓	✓		✓		
	[124]	✓	✓		✓			✓		-	-	-	-	-	-	-	✓		
	[125]	✓	✓	✓	✓			✓	✓	✓		✓	✓			✓	✓		
	[126]	-	-	-	✓	✓			✓	-	-	-	-	-	-	-	✓		
	[127]	✓	✓	✓		✓			✓	✓	✓	✓	✓		✓	✓	-	-	-
	[128]	✓	✓	✓		✓			✓	✓	✓	✓	✓		✓		✓		
	[129]	✓	✓	✓					✓			✓					✓		
	[130]	-	-	-	✓		✓							✓			✓		
	[131]	-	-	-	✓	✓			✓				✓					✓	
	[132]	-	-	-		✓			✓		✓	✓						✓	
	[133]	-	-	-					✓	✓	✓		✓				✓		
	[134]	-	-	-					✓						✓		-	-	-
	[135]		✓			✓			✓	✓								✓	
	[136]	-	-	-		✓			✓			✓	✓				✓		

Header Legend

Protocol
1: BACnet
2: KNX
3: LonWorks

Target
4: Management Network
5: Field Network
6: Management Device
7: Interconnection Device
8: Field Device

Network Attacks
9: Interception
10: Modification
11: Injection
12: Interruption

Device Attacks
13: Software
14: Physical
15: Side Channel

Countermeasure
16: Prevention
17: Detection
18: Hybrid

Table 4.2: Systematization in Depth by Contribution

Reference	Vector						Impact				Defense						Scope of Liability					
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
[63]	✓			✓	✓		✓		✓	✓						✓	✓	✓		✓	-	-
[98]	-	-	-	-	-	-	-	-	-	-			✓				✓			✓	✓	✓
[99]	-	-	-	-	-	-	-	-	-	-		✓					✓	✓		✓		
[64]			✓	✓				✓		✓			✓			✓	✓	✓			✓	
[65]	-	-	-	-	-	-	-	-	-	-	✓				✓						✓	
[57]				✓				✓			✓					✓	✓	✓		✓		
[15]	-	-	-	-	-	-	-	-	-	-		✓						✓		✓		
[29]				✓		✓				✓		✓						✓		✓		
[100]				✓		✓	✓	✓		✓	-	-	-	-	-	-						
[66]			✓	✓	✓			v		✓	✓						✓	✓		✓		
[67]	✓		✓	✓				✓		✓		✓						✓		✓		
[68]	✓			✓				✓		✓	✓						✓	✓		✓		
[69]			✓	✓	✓		-	-	-	-		✓	✓			✓			✓	✓		
[70]	-	-	-	-	-	-	-	-	-	-						✓		✓	✓			
[12]	✓	✓	✓		✓	✓		✓		✓						✓		✓		✓		
[114]	-	-	-	-	-	-	-	-	-	-					✓		✓	✓				
[71]	✓	✓		✓				✓								✓	✓	✓		✓	✓	✓
[72]	✓				✓	✓		✓		✓			✓			✓	✓	✓		✓		
[73]	-	-	-	-	-	-	-	-	-	-			✓	✓							✓	
[74]				✓				✓						✓							✓	
[75]		✓			✓			✓	✓							✓		✓		✓		
[76]	-	-	-	-	-	-	-	-	-	-		✓					✓	✓		✓		
[101]	✓							✓				✓					✓	✓		✓		
[13]	✓	✓	✓				✓		✓							✓		✓		✓		
[115]				✓			✓				-	-	-	-	-	-				✓		
[102]		✓								✓					✓						✓	
[116]	✓	✓		✓		✓	✓	✓								✓					✓	✓
[28]	✓			✓	✓			✓		✓		✓						✓		✓		
[117]				✓				✓		✓	-	-	-	-	-	-	✓	✓		✓	✓	
[103]	✓		✓					✓		✓		✓						✓				
[77]	-	-	-	-	-	-	-	-	-	-						✓	✓	✓		✓	✓	✓
[78]					✓					✓						✓	✓			✓		
[104]	-	-	-	-	-	-	-	-		-			✓				✓				✓	
[79]	✓						✓			✓					✓		✓	✓		✓		
[80]				✓			✓	✓		✓			✓		✓		✓	✓		✓	✓	✓
[118]	-	-	-	-	-	-	-	-	-	-	✓				✓					✓		
[119]				✓				✓		✓			✓	✓	✓		✓			✓		
[105]	-	-	-	-	-	-	-	-	-	-						✓	✓					
[106]				✓				✓		✓						✓	✓	✓		✓		
[81]	✓	✓						✓			-	-	-	-	-	-	-	-	-	-	-	-
[82]				✓				✓		✓		✓					-	-	-	-	-	-
[83]	-	-	-	-	-	-	-	-	-	-		✓					✓	✓		✓		
[84]						✓	✓				✓						✓	✓		✓		
[85]				✓				✓		✓			✓	✓		✓					✓	✓
[35]	✓		✓				✓	✓	✓	✓		✓					✓	✓		✓		
[107]	✓		✓	✓				✓		✓		✓					✓	✓		✓		
[108]	✓	✓								✓			✓	✓	✓						✓	
[74]				✓			✓					✓								✓		
[120]	-	-	-	-	-	-	-	-	-	-						✓	✓	✓		✓		
[86]	-	-	-	-	-	-	-	-	-	-			✓		✓		✓	✓		✓		
[121]	✓						✓	✓		✓					✓		✓	✓		✓	✓	
[122]	✓	✓			✓		✓	✓		✓					✓	✓					✓	
[123]				✓		✓		✓		✓					✓						✓	
[124]	-	-	-	-	-	-	-	-	-	-				✓			✓				✓	
[16]				✓				✓		✓		✓					✓	✓		✓		
[125]				✓		✓		✓					✓	✓							✓	
[109]	-	-	-	-	-	-	-	-	-	-						✓					✓	✓
[87]	✓					✓	✓	✓	✓		-	-	-	-	-	-	-	-	-	-	-	-
[88]	✓				✓	✓			✓	✓						✓	✓	✓		✓		
[89]	✓			✓		✓		✓		✓	-	-	-	-	-	-	-	-	-	-	-	-
[90]				✓				✓								✓	✓	✓			✓	
[126]	-	-	-	-	-	-	-	-	-	-			✓		✓					✓		
[91]				✓	✓	✓		✓	✓							✓	✓	✓		✓		
[127]				✓			✓				-	-	-	-	-	-	-	-	-	-	-	-
[128]	✓			✓			✓	✓		✓			✓	✓		✓					✓	
[110]				✓			✓					✓					✓			✓		
[129]				✓				✓		✓			✓		✓		✓			✓		
[130]		✓			✓					✓					✓						✓	
[137]				✓			✓					✓					✓			✓		
[131]					✓	✓	✓	✓				✓					✓	✓		✓		
[111]				✓				✓			✓						✓			✓		
[93]	-	-	-	-	-	-	-	-	-	-					✓						✓	
[94]	-	-	-	-	-	-	-	-	v	-		✓					✓			✓		
[95]				✓				✓		✓		✓			✓		✓			✓		
[96]	✓							✓			✓						✓	✓		✓		
[132]				✓						✓		✓					✓	✓		✓		
[112]				✓		✓		✓								✓	✓	✓		✓		
[133]						✓		✓								✓	✓	✓		✓		
[134]					✓				✓		-	-	-	-	-	-	-	-	-	-	-	-
[113]		✓				✓			✓	✓						✓	✓	✓		✓		
[135]				✓				✓				✓					✓			✓		
[136]						✓		✓			✓				✓		✓			✓		
[97]	-	-	-	-	-	-	-	-	-	-		✓					✓	✓		✓		

Header Legend

Vector
1: Insecure Services
2: Insecure or Legacy Software
3: HW/FW Modification or Failure
4: Insecure Protocol
5: Human Error
6: DoS

Impact
7: Physical Damage
8: Data Leakage/Corruption
9: NW Access or Privilege Loss
10: Financial Loss

Defense
11: IPS
12: IDS
13: Cryptographic Method
14: New/Improved Protocol
15: Secure Software
16: Best Practice

Scope of Liability
17: System Integrator
18: Building Owner
19: Building Occupant
20: Facility Manager
21: Manufacturer
22: Vendor/Distributor

thus finding vulnerabilities to exploit is key.

Attacks

Antonini [89] performs a security analysis of BA protocol vulnerabilities (namely BACnet, KNX, Modbus, and LonWorks). It found that security features are not intrinsically present and furthermore that there is very limited supported bandwidth for data transmissions. Given these issues the authors highlight one a cryptographic approach to address the underlying confidentiality, integrity, and availability issues. In Frimans's [87] security threat analysis it was found that the threats with the greatest impact and probability are associated with (1) unauthorized cloud data access (2) interruption of building climate control systems and (3) destruction of HVAC systems and/or loss of goods/equipment.

Countermeasures

The overwhelming majority of research at the management level explore various BAS insecurities and propose best practices or give security recommendations and guidelines as solutions. In [13] [71] cybersecurity plans for smart buildings are discussed from the business perspective and the issues uncovered are evaluated from the vantege point of a broad range of building stakeholders. Similarly the authors in [70] proposed an impact assessment methodology for measuring business continuity. The key idea being to consider and score building automation and control system software modules based on the weight of their role in supporting business processes and their interconnectedness to other software modules.

In [12] it was found that there is no single document for facility operators and security professionals to consult for understanding risks in the domain. Due to this fact they and many researchers [12, 88, 63, 78, 77, 69, 72] in the literature propose or make call-to-actions for more comprehensive all encompassing BAS security measures which are widely accessible and open to be referenced by all. The recommendations can be subdivided into

two categories, improvements to technology (protocols, software, etc.) and improvements to security education (training facility operators, designers, etc) which would effect security practices. Some of the most widely stressed suggestions included calls for -

- Physical Security - Isolation and implementation of tamper resistant measures for building automation devices
- Secure External Access to Network Resources - Use of intrusion detection mechanisms, firewalls, authenticated protocols with encryption
- Secure Software - Ensuring the use of system software that conform to good risk management policy and is kept up to date
- Operator Security Training - Teaching operators the necessary BAS security mechanisms and measures

The commonalities of the recommendations among the literature demonstrate the urgency of the need for improvement and Wendzel et al. [77] assert this change will require the collaboration of academia and industry.

While general comprehensive approaches have the advantage of being far-reaching there are a handful of works [91, 90, 85, 75] that give specific guidance for particular threats. In [91] security improvement steps are given for mitigating the risk of BACnet amplification attacks with digital certificates (like SSL) recommended for mitigating man-in-the-middle (MITM) threats [90].

When quantifying BAS availability and security a recurring proposed methodology in the literature is to employ the use of Attack Tree Analysis (ATA) and Markov models [66, 68, 96]. The fundamental tenet among them is that ATA can be utilized to explore or *simulate* BAS intrusions by computing the probability of system wide destabilization given the probability of individual building automation subsystem failure. Markov models are then used to illustrate the various possibilities of recovery and system states following

discrete system faults. Hachem et al. [84] propose a similar but novel approach which entails the simulation and security modeling of systems to uncover emerging cascading attacks. By discovering attacks at the architecture stage of a building's software life cycle they resolve to save cost, future development time, and mitigate high impact attacks.

Just like in traditional IT security, protecting the network is a cardinal requirement in BASs. This is made even more challenging given that the, arguably, most vulnerable part of BANs (IP components) lie heavily in the management layer. To combat attacks researchers have adopted intrusion detection techniques from the IT world. Szlósarczyk [111] and Kaur [16] both proposed the adaptation of traffic normalization as seen in TCP/IP networks to analyze BACnet traffic and protect receivers against fuzzing attacks. The implementation consisted of a BACnet testbed of virtual machines (VMs) to represent devices inclusive of a Snort normalization node extended to BACnet/IP and configured with rules to remove exploitable ambiguities in the traffic. Research has also shown that through traffic flow monitoring and analysis network activities and security issues can be uncovered. One approach accomplished this by comparing volume and entropy based techniques in [28] and another by combining statistical analysis with the application use case to develop anomaly detectors on a university BAN [74]. Others leverage traffic analyses for identifying anomalies involving the use of unsupervised machine learning techniques [83] and computing probability maps from visualized flows [94].

A non-trivial issue experienced both in traditional IT and BANs is the dilemma of network level device identification. Most BAN operators do not know about all of the devices on their networks and this becomes a major issue when distinguishing malicious actors/behavior from improperly documented or malfunctioning nodes. To mitigate this researchers have proposed situational awareness techniques to identify and label devices according to their network behavior or *role*. That information is then used to pinpoint unusual activity and map out the network [82] [35]. Related works use BA device and network documentation collected both online and offline to learn expected network device behavior

and extract rules to enforce them in addition to sending alerts upon detecting violations [137]. This research was limited by the fact that it could only retrieve device documentation to characterize and generate rules once the device in question transmitted data packets with specific keywords and could only process documentation rendered in a few predefined formats. Esquivel-Vargas et al. [67] improved upon the overall approach, by automating the document retrieval and specification extraction process, but the aforementioned device transmission limitation remains. The use of honeypots in traditional IT networks dates back decades, but it has only more recently become commonplace in CPS networks. During a ten week experiment period in 2019, Bauer et al. [97] set up eight DDC4200 (direct digital control device) honeypots and observed 62,611 unauthorized login attempts from various countries worldwide. In [76] the authors unified physics based security analytics for detecting cyber attack with actionable resilience policies to keep a building EMS functional during attack.

Protocol security is an especially hot topic in BASs since technology changes and the interconnection of systems has left them open to vulnerabilities that original protocol designers could have never imagined. In [138] a model for implementing multilevel security to prevent covert channels in BAS protocol environments is proposed. The developed model was based on an analysis of BACnet and operates under the assumption that larger organizations are already separated into physically disparate security zones which can be directly implemented into BANs with hierarchical router structures. Research in the same vein by Granzer et al. [73] offers a protocol agnostic architecture for secure data exchange between control applications. In order to accomplish this they define key primary and secondary security objectives which must be satisfied in all management nodes on the network by means of a new multiprotocol communication stack.

Secure software and hardware are paramount for the overall security of BASs and thus substantial effort [79, 93, 95, 80, 86] has been made by the research community to point out flaws in current deployments as well as propose improvements. Hernandez-Ramos et

al. [86] propose a platform for secure management of IoT enabled smart building services via an ARM compliant framework which extends the City Explorer software. The framework presented in [80] aimed to provide a secure communication foundation by leveraging the seL4 microkernel to enforce global communications policies in BANs with minimal impact to network performance. Caranica et al's [95] proposed building management solution varied slightly as they created a universal threat management system of open source tools to provide cost effective security solutions for budget restricted parties. The security architecture is highly customizable and relies on logically separated network segments. Researchers in [93] explored the rarely discussed area of usable security in BAS and found that a key user concern is the loss of data due to cloud supplier compromise. As a result they designed a distributed private cloud computing model that allows multiple cloud computing suppliers to manage data together, but grant none access to read all the data. Much like the research of Wendzel [138], Boyer et al. [79] propose the use of multi-tier web servers and authentication systems to prevent the exploitation of least privilege in BASs. Their technique protects lower tiers from potentially compromised higher tiers by enforcing *redundant authentication*, a method for requiring higher tiers to provide evidence of authentication upon making requests to lower tiers.

4.4.2 Automation Layer

Located below the management layer, in Figure 3.2, is the automation layer. This is where all of the complex control logic and automation processes are implemented. Many of the buildings critical controllers operate at this level. The controllers are key for relaying information from sensors and actuators at the field layer to operator workstations at the management layer.

Attacks

Attack vectors in the BAS domain over the last several years have grown and are trending towards increases in magnitude as well as the potential to harm building occupants. The first BAS-specific malware was presented in [101, 81] using known and unknown zero day vulnerabilities to infect and persist in BANs. To demonstrate their attack the researchers built a testbed of BA devices (IP camera, two PLCs, Phillips Hue light bulb and motion sensor, and an IoT thermostat) interconnected on an IP network. The malware exploits vulnerabilities in the IP camera to copy itself to the devices and access a misconfigured database software before finally connecting to the access control PLC and performing its malicious payload. Similarly in [113] researchers explore the usage of enterprise IoT device drivers as an attack mechanism and demonstrate examples of how the vulnerabilities can impact the security of smart buildings. They name the series of attacks (Dos, remote control, resource farming) PoisonIvy and all stem from the download and installation of a malicious unverified driver which compromised a system controller.

The attack presented in [106] takes advantage of the well-known lack of authentication in the KNX protocol and use the eibd software to inject packets onto the KNX network of the St. Regis ShenZhen hotel. By connecting to hotel's IP network and monitoring traffic from the room's iPad they were trivially able to identify requested actions such as temperature and television control. With this knowledge they learned the hotel's KNX/IP room addressing scheme and could send arbitrary actions to any room of their choice. They report these vulnerabilities to the hotel and suggest the implementation of a secure tunnel from each room iPad to the KNX/IP router.

Countermeasures

In [15, 29] rule based and anomaly based IDSs are proposed, respectively. The former approach first builds a normal model of BACnet traffic characterized by Network Protocol Data Unit (NPDU) and Application Protocol Data Unit (APDU) data. The researchers

generate abnormal traffic and test their approach on a testbed fire alarm system where they obtain good classification accuracy, but false positive rates too high for a real implementation. In the latter, network data is (1) collected, (2) modeled into Protocol Context Aware Data Structures (PCADS) and Sensor-DNA (s-DNA), (3) analyzed with Discrete Wavelets Transform (DWT) and then (4) rule based anomaly behavior analysis methods are performed to classify the data and explore countermeasures. The detection rate on each of the six attack categories defined by the researchers exceeded 90%, but much like the former approach the false positive rates were too substantial to deploy in practice. Additionally, both approaches extend intrusion detection into intrusion prevention with *protective actions*, but given their false positive rates this could lead to the dropping of legitimate network traffic. Peacock's [100] approached the anomaly detection of known and unknown threats in BACnet with machine learning. The author gathered BACnet data from a university campus to develop a simulation environment for generating experimental normal and anomalous network traffic. Graph analysis, clustering, time series, and HMM analyses were run for detecting anomalous network traffic and were found to be able to detect known/unknown threats at the cost of a high false positive rate.

Lightweight formal system specifications are combined with anomaly detection in [103] to identify network abnormalities and generate actionable results. The main contribution of this work is mapping network traffic that would otherwise be incomprehensible to known process variables via operator domain knowledge. Fauri et al. [107] pick up where [82, 35, 137] leave off with their role-based intrusion detection approach. Unlike former approaches device roles are dynamically learned from network traffic and are not dependent upon any vendor specific device descriptions, allowing the IDS to adapt with the BAN over time. Additionally they offer greater contextualization in the anomaly alerts and network map to be better understood by network operators. Another IDS proposed by [110] aimed to expose sequence attacks which use chains of seemingly innocuous events to perform a malicious payload. As a result the developed sequence aware IDS studied patterns of ICS

network events to extract their semantic meaning and models known behaviors over time. The architecture is experimentally investigated using discrete-time Markov chains and the initial results show good attack detection rates with low false positives. In the very early work of Luo et al [99] multiple building sensors (with redundancy) are used to detect BAS faults and unsafe events. The experiments offer promising results with detection accuracy as high as 95%, but only apply to very particular BAS environments.

The work of [98] presents a novel encryption scheme to protect big data network flow in environments where applications are data-driven. The approach is called SymCpAbe and it blends the best parts of symmetric and attribute-based encryption schemes to provide an efficient and flexible solution for sensitive data. SymCpAbe protects data from sensors with a symmetric key algorithm that additionally protects its generated symmetric keys with CP-ABE encryption (performed by a third party as to not tie up resource limited devices). Cavalieri et al [104] present a cryptographic scheme specifically for addressing authentication and encryption in KNX. Their proposal requires the addition of a new device 'controller' to each KNX network which oversee and coordinate network authentication. The controller is responsible for distributing network keys to every network device and by doing so becomes a single point of failure in the network.

Very few works take the approach presented in [108] for offering authentication and key management services. The research leverages a software defined networking (SDN) framework for facilitating secure communication between smart devices and a central SDN controller via an extension to the Open Flow message protocol. A demo of the system implemented with a server for the SDN controller and an Arduino Uno for the smart device. It was found that losses of up to 0.5s over a regular Open Flow authentication initialization session can be sustained.

Manufacturer and vendor 'lock in' can be major hurdles for building administrators, once one flagship system is installed it can be expensive and sometimes infeasible to integrate a different one. This is due to the lack of inbuilt interoperability within BASs. To

overcome this limitation researchers in [65] designed a lightweight multilayer architecture that allows remote access and has multi-vendor support. The proposed system provides an interface for multiple applications over a shared API via SSL. The solution presented in Wang et al. [102] also aimed to secure critical applications. Their approach was based on the concept of transforming BAS microcontrollers into security anchors that would protect the critical functionality of controllers even if part of the system becomes compromised. They implemented this technique on some security enhanced microkernel architectures and found that attack impacts were reduced for the two attack simulation scenarios tested.

Time and time again researchers have poked holes at and found exploits in the security of BAS protocols. Reliability and robustness to malicious manipulations require the support of advanced security mechanisms. Granzer et al. [109] identify seven functional requirements to be fulfilled for maximum effectiveness of BAS solutions in security critical environments. Additionally five domain specific challenges which prevent the direct mapping of IT security mechanisms to BASs is presented and a gap in the delivery of data availability by the studied BAS standards is exposed. Further evaluation by Liu et al. [57] led to the proposal of a protocol security assessment covering all phases of device interaction. The taxonomy is used to assess the security of Thread, a native IP BAS protocol, and enhancements to Thread's security are discussed.

After security and vulnerability analyses of the BAS domain, both [105] and [112] find there are areas necessary to address to achieve better security functionality. These include the fortification against bus interference, assured local and system level management controls, use of encryption with keys and certificates, and intelligent traffic filtering at routers.

4.4.3 Field Layer

Attacks

When seeking to attack, system reconnaissance for learning the attack surface is usually the first place to start. In [64] an analysis of a real KNX/IP network attack surface revealed a

host of vulnerabilities which the authors wanted to corroborate with previous BAS research. In support of this a testbed was built to mimic a real BAS deployment and assist with the investigation of network susceptibility to practical exploits in KNX. The attacks show the ease of reverse engineering vendor specific KNX protocol components and confirm previously discovered exploitable vulnerabilities can be catastrophic to BANs. From the digital forensic analysis of a university building's KNX BAN, Mundt et al. [117] [127] assert it impossible to secure a publicly accessible KNX installation. Experimental research results provided show that even preventing physical access to BASs does little to improve security.

Three attacks leveraging proactive participation demand scheme of BASs in smart grids are examined in [134]. The first attack requires accessing metering devices and can leak private customer usage data, the second performed false data injection for submitting fake demand bidding information to mislead market operators, and third spoofed network traffic to manipulate the electricity guideline price received by other customers. In [115] the researchers explore models of the general sensor integrity attack in targeted mode where an attacker is goal oriented. The experimental design pertained to an HVAC control system where the attackers end goal could be to over cool or under cool the room. The results from testbed experimentation revealed the system to be more prone to the under cool attack than over cool.

Countermeasures

Using the Failure Mode Effects and Analysis (FMEA) method Abdulmunem et al. [131] present an approach to, in the event of BAS compromise, learn the extent of device vulnerability and failure as well as failed component effect on the overall system performance. Additionally Intervention Mode Effects and Criticality Analysis (IMECA) methods are used to study attacks and measure the degree to which intervention mechanisms for shutting down or placing critical systems on stand-by, impact comprehensive BAS security.

Markov models were used to implement the assessments and compute mean time to failure and mean time to repair for system analysis. Other techniques like the Physical Intrusion Detection System proposed in [135] aim to leverage device or component analysis for classifying whether or not an attack takes places. In this work Mundt et al. train Support Vector Machines with sensor bus data collected from a KNX office building deployment to detect physically intrusive actions inside the building. For system evaluation a testbed was built and simulated attacks were crafted. The results show that in the case of a single attacker anomalies are hard to detect in peak hours, but as the degree of attack increases so does the detection rate.

In their proof of concept Mays et al. [132] develop techniques for analyzing the proprietary INSTEON BAS protocol. By using software defined radios they were able to increase packet capture rates from 40% to 75% and perform protocol analysis which enabled the design and deployment of honeypots. Experimental testing was performed to determine the authenticity of the mapping of the honeypot network to a genuine one as well as the honeypot's targetability by measuring how closely its network presence aligned with real devices. In test environments the honeypot was found to have high authenticity and targetability, but requires logging capabilities before testing in the wild.

KNX is among the most security problematic of the BAS protocols deployed and used today. As a result there have been a handful of proposed extensions and improvements in the literature [128, 125, 119]. Granzer et al. [128] developed EIBsec to enable mechanisms which provide authentication services and guarantees data confidentiality, integrity, and freshness. EIBsec handles key management and distribution and is backwards compatible with existing KNX devices taking advantage of standard network infrastructures to build in security. The secure communication protocol designed in [125] was tested on a KNX BAN, but is generic to any BAN. This approach provides security guarantees against eavesdropping and data modification. The novelty of this solution lies in the multiparty key agreement scheme which was based on the discrete logarithm problem and shown equiv-

alent to a computational Diffie Helman problem. The protocol implements logical time slotting and a periodically refreshed ephemeral shared secret key. Experimental results indicate secure commands can be sent with the worst case timing of 1.3s if 40 KNX nodes are in the collision domain and key agreement in one minute. Glanzer et al. [119] propose a security extension to KNX that claims to be resistant against hardware faults and malicious parties. KNX security routers with two separate KNX interfaces (one to the standard KNX and one to the secured KNX) are introduced. The feasibility of the solution was tested with Raspberry Pis for the security routers on a KNX test BAN and found to introduce significant communication overhead and require redundancy to withstand DoS attacks. Similarly the research in [124] presented a vertically integrated networking approach that allowed for direct device data point mapping, which removes the need for the use of gateways. The implementation builds on a multiprotocol communication stack and the proof of concept device can support both KNX and BACnet.

Hardware solutions for addressing security in BAS are few and far between, but Halemani et al. [114] developed a proof of concept low cost interface for the control of electrical loads. The prototype runs on a Beagle Bone Black connected to a CAN network with custom messages for BAS intrusion detection. To preserve BA device authenticity and prevent foreign devices from joining the BAN, Fischer et al. [126] present a security trust anchor approach. The goal is to incorporate authentication into an ordinary BAS communication stack through the inclusion of an off-the-shelf hardware security module extended to perform challenge response authentication.

In recent years there has been more synergy between the Architecture, Engineering, Construction (AEC), Building Automation System and security domains. The research in [118] embodies this integration with an approach that maps Building Information Models (BIM - typically used in the AEC industry), to the BAN via a building management system for identifying early vulnerabilities in the construction process. The work in [130, 78, 122] address securing software running on multiple device classes while preventing attacks. To

accomplish this a formalized model for expressing control application software security is presented. Wang et al. [123] present a real time OS architecture designed specifically for BAS and CPS controllers.

Sensor value accuracy is critical to the proper function of any BAS, to ensure this correctness researchers [136] propose a sensor correction model for abating data integrity attacks based on a deep learning framework. Their method used a denoising autoencoder model to learn spatial correlation among sensors. Experimentation showed the proof of concept model successfully stopped attacks on a testbed network, but has a high overhead under normal non-attack network conditions. In their work Schwaiger et al. [129] redesign a smart card security system for Lonworks to increase authentication and support unacknowledged services on BANs.

Much field level security research has resulted in community findings in the form of suggestions of best practice and guidelines for secure operation of BASs. Qi et al. [133] explore the implementation of demand response in smart buildings and the cyber physical security challenges associated with it. BAS failures that could lead to demand response failures were identified and can be summarized as: (1) any interruption, interception, or modification of information between a system and demand response automation server, and (2) malware attacks based on the compromise of a system or server. The authors provide four recommendations for enhancing security in BASs: (1) utilization of trusted system architectures (2) design of access control models, (3) implementation of secure communication, and (4) incorporation of privacy guarantees. Botnets are reimagined for the BAS domain in [116] and shown feasible through wardriving, remote command execution, and remote vulnerability exploitation. For counteracting the attacks described in this work, a call to action is made for the collaboration of academia, industry, and government to develop detection and mitigation techniques for bot infections and mechanisms for protecting new/existing BASs. As a seeming response to the call, Novak et al. [120] introduce a common approach for engineering safe and secure BA field level technology. The proposal

is based on the fundamental tenet of harmonizing the safety and security of the device at every stage of its life cycle.

4.5 Building Automation Security Framework

In what follows, we present a comprehensive framework for assessing the security of BASs. We study and analyze building automation security from two perspectives: i) Device, ii) Network. The distinction between these is that the device level considers BA devices individually without factoring their larger role or station in the network.

There are three domains of attack to consider when analyzing device security in BASs: i) Software/Firmware, ii) Side-Channel, and iii) Physical [52]. Physical security is critically important, because due to the nature of BASs many devices can be accessed anytime [75]. Adversaries can leverage physical security weaknesses as well as other unprotected channels to exploit vulnerabilities in device software and firmware. They could even use externally observable device parameters to collect side-channel data about device operation. Fortifying devices on these fronts lowers a device's susceptibility to manipulation from outside forces. To measure BAS security posture from the device level, the proposed framework uses 8 assessment criteria spanning the Software/Firmware, Side-Channel, and Physical attack domains.

BA device security from the network level can be qualitatively derived through the evaluation of 7 criteria of device-network association. These range from the establishment of the physical network media (Network Access), to when a device joins (Commissioning) and leaves (Leaving), as well as all the interactions in between (Communication, Upgrading and Patching, Data Warehousing, Intrusion Detection and Prevention). Our proposed evaluation framework draws from the Five Phases of Interaction explained in Liu et. al.[57], but is modified to include consideration for the greater context of a BA device's role in the network. A detailed explanation about each assessment criterion is given in the following subsections.

4.5.1 Device Level Assessment Criteria

In this section, we discuss the 8 assessment criteria of building automation device assurance as they pertain to the standalone devices, separate from the underlying network. Additionally, each section breaks down the procedure of device evaluation for each criterion.

Software/Firmware

BA devices can be thought of as embedded systems which run firmware on computationally limited resources. In addition to their own personal firmware, many devices (such as access control systems) require the installation of additional software on operator workstations (management devices) for facilitating the management of device settings and capabilities [139]. These software packages oftentimes have out-of-date OS requirements which can be a potential point of insecurity. Much like traditional IT systems, vulnerabilities in device software and firmware can be exploited by adversaries to threaten the confidentiality, integrity, and availability of the entire network [140]. For this reason our proposed security evaluation framework leverages the individual vulnerability assessment from multiple well-known repositories to define the Software/Firmware security posture of each BA device.

Evaluation Procedure -

(i) Locate device software/firmware name and version (search the web, check logs, refer to device manual);

(ii) **SW/FW score:** Search the i) ICS-CERT Advisory [141] and ii) CVE Details website [142] for the software/firmware vulnerability assessment and CVSS score[1];

(iii) If the device requires any additional devices or systems for operation, repeat steps 1 and 2 for each of them;

[1] If multiple CVSS scores are found, take the maximum

(iv) **OS Score:** If there are OS requirements for the operation or commissioning of the device, search the CVE Details website for the OS vulnerability assessment and CVSS score[2];

(v) **Vendor Score:** Search the CVE Details website for device's vendor CVSS score.

The scores derived from this procedure (OS, Vendor SW/FW) are indicated in Table 4.4 by their numeric (1-10) CVSS score or (–) if not applicable. The combination of these 3 scores provides a representative outlook of the software/firmware security stance for a standalone BA device. Here, the CVSS scores serve as well-known and community accepted vulnerability scoring metrics [143].

Side-Channels

For the sake of this research, side-channels are considered the unintentional means of communicating system operational information. Examples of side-channel signals could be electromagnetic emissions from processors, acoustic noise from computer fans, or the overall power consumption of a machine [144].

Side channel signals can be leveraged to either infer secret information about the running BA devices, or intentionally affect their performance (i.e., DoS, or malicious command injection) during their normal operation. For example, researchers in [145] leveraged the analysis of the total power consumption in a computer system to infer encryption keys. As another example, it was shown in [146] that a laser-based signal can intentionally affect the performance voice-controllable systems. In this evaluation framework, we focus on the influence of side-channel signals which can actively disrupt or control the state(s) of BA devices.

Type of Side-Channels For comprehensiveness, our evaluation framework considers a wide range of side-channel signals including mechanical energy, thermal energy, magnetic

[2]See footnote footnote 1

fields, and electromagnetic waves as detailed in Table 4.3. These side-channels are of particular importance in BASs because they can threaten the availability and integrity of data transmission which could stall or even prevent normal system operation(s). For each side channel modality, the evaluation framework score is obtained as discussed below.

Table 4.3: Side-Channels and Signal Generators

Category	**Modality**	**Generators**
Mechanical	Acoustic	Speaker
Thermal	Temperature	Heat Gun
Magnetic	Quasi-Static Magnetic Field	Permanent Magnet
Electromagnetic	Infrared Waves Visible Light	IR Flashlight Projector

Evaluation Procedure - The side channel assessment has four possible outcomes (●/○/◑/◐). If there is an influence on a device's output (or internal variables) from the side-channel signal, we consider this side channel modality to be a potential attack vector for the evaluated device. This is denoted in Table 4.4 for a given side channel category as, all side channel modalities influenced the device - ● or none of the side channel modalities influenced the device - ○. For the electromagnetic (EM) category there are two modalities, so the influence from only IR waves is denoted as (EM - ◑) and the influence from only Visible Light is denoted as (EM - ◐). The selection of side-channels experiments to perform on a BA device should be made carefully. Each BA device is made up of different components that serve various purposes (i.e. lighting devices commonly use PIR sensors). To have the highest chance of identifying a potentially influential side-channel, one should choose a side-channel modality with respect to the main medium(s) of operation for the given device. For each of the modalities, the following general testing rules were applied:

(i) Place the signal generator within a reasonable range (e.g., $< 1ft$) of the evaluated device;

(ii) If the signal generator supports user-defined patterns, use a single frequency/amplitude component one at a time (the frequency/amplitude can be changed to find the most

influential condition);

(iii) For each experiment, record the response of the device (taking note of all relevant variables), with and without the presence of the side-channel signal. The response without the presence of the side-channel signal is placed in the control group, whereas the response with the presence of the side-channel signal is placed in the experiment group. Repeat the process N times (for example $N = 5$). Perform a statistical test with the null hypothesis that the injected signal does not affect the output of the device.

Physical

Physical security for BASs can be difficult to achieve due to the open access nature of some buildings. BA devices are generally accessible not only by authorized building operators and facility managers, but many times by building users as well. An example of this is a thermostat in each room of a large hotel. These devices are both sensors and controllers of the HVAC system for a predefined space which can be adjusted by room occupants to accommodate for their comfort. This kind of visibility cannot always be avoided, but instead, should be accounted for with countermeasures to prevent calamity. In 2014, an attack on the St. Regis ShenZhen luxury hotel through the room's iPad controller interface enabled the remote control of every room from the compromised device [147]. With no consideration to physical device security, BASs are vulnerable to these kinds of attacks and more. If physical security barriers to access cannot be applied, every device should at least be tamper-proof. Tamper-proofness hinders compromise from tampering such as obtaining network secrets and easily performing data manipulation. Additionally, there should be tamper protection built inside device hardware for defending the memory, processor, and the intellectual property of the software/firmware [148]. The evaluation framework for physical security of BA devices is explained as follows.

Evaluation Procedure - The physical assessment has four possible outcomes (P -

●/●/●/●). We define three benchmarks for scoring a device's physical security, 1) barriers to physical access, 2) tamper-proofness, and 3) regular device inspections. Barriers may include physical intrusion detection systems (e.g., cameras, access control mechanisms, motion detectors, etc.). Regular (e.g., monthly, biweekly, etc.) inspections should be performed by system operators to ensure physical device integrity. For every BA device, each benchmark should be checked and if the device has none of the mentioned factors, then (P - ●). If the device has only one of the aforementioned factors, then (P - ●). If the device has two of the aforementioned factors, its security is (P - ●). If the device has all the three aforementioned factors together, its security is (P - ●). Although we cannot compare the discussed factors with each other deterministically, combining them seems reasonable from the security experts point of view [149].

4.5.2 Network Level Assessment Criteria

The main goal of this stage is to assess the security of the given BAS from the network perspective. The BAS network evaluation has the following assessment categories: 1) Network Access, 2) Commissioning, 3) Communication, 4) Upgrading and Patching, 5) Leaving, 6) Data Warehousing, and 7) Intrusion Detection and Prevention.

Network Access (NA)

In BASs, it can be assumed that the network exists within an untrusted environment. In this environment, physical access to the network medium cannot always be controlled, especially in public buildings where people may freely connect to the IT networks. Common network media for BASs include, powerline, radio frequency (RF), wired twisted pair cable, Ethernet, and range in ease of compromise [150]. It is safe to assume that for a short amount of time, physical access to a BAN may be inevitable for a determined attacker. As a result, one way to enhance BAN security is to put physical and network barriers in place to decrease access time. For the physical barriers it is recommended that BANs be in-

stalled in protected environments with physical intrusion detection systems in place where the access is hard and takes excessive time. Furthermore, there should be tamper-proof hardware where possible and manipulation detection mechanisms for maintaining medium integrity. For the network barriers, the BAN should combine network isolation methods from the building's traditional IT network by means of firewalls, VPNs/VLANs, network segmentation and interrupt techniques [151]. Interrupt techniques can guarantee data availability against DoS attacks through anti-spamming services, content filtering, disabling port scans, and many more. Network access credentials should be distributed on an as-needed basis only, and wherever possible secure channels should be used. To be secure, a channel must employ unbroken cryptography techniques for avoiding unauthorized interference of data, providing data integrity, and assurance against data injection and interception. This is especially important when the remote Internet access to the network is a BAS operating requirement.

Evaluation Procedure - The network access assessment has three possible outcomes (NA - ●/●/●). Check the network barriers to the BAN and physical barriers to the network media. If there are very little or no networked and physical barriers for BAN access this is a poor security practice (NA - ●). If there are a reasonably insurmountable number of networked/physical barriers to access and remote BAN access *is possible* via an insecure Internet channel this is also a poor security practice (NA - ●) which leaves open a large attack surface. Alternatively if the BAN that *is remotely connected to the Internet*, connected through a secure channel, this is (NA - ●) good form, but should be regularly monitored for compromise. Here, a well-rounded practice is to implement networked/physical barriers, maintain secure channels where needed, and prohibit remote BAN access via the Internet (NA - ●).

Commissioning (CX)

In BASs, commissioning is defined as the process in which a new network device finds the correct network, mutually authenticates with the commissioner, establishes trust to gain proper credentials and configuration, and creates a communication channel with other devices in the network [152]. Secure commissioning practices should guarantee security during the network detection, authentication, and delivery of network secrets. Such techniques can effectively defend against spoofing, man-in-the-middle (MITM), and denial of service (DoS) attacks as well as protecting the network from leaking secrets to unauthorized third parties [152].

Evaluation Procedure - The commissioning assessment has three possible outcomes (CX - ●/●/●). Review the initial device network association scheme, if the commissioning procedure does not utilize state of the art (unbroken and generally regarded as secure) mechanisms for authorizing and authenticating network devices (CX - ●), this is a poor security implementation and attackers can perform spoofing, MITM, or DoS attacks in the network. If sufficient authentication and authorization are used, but there is no encryption of certificates or shared secrets (CX - ●), this is an improvement upon the former, but still an insecure practice. The recommended approach is for devices to perform sufficient authentication/authorization and using encryption for certificates, shared secrets, or other verifiable credentials (CX - ●).

Communication (Comms)

Following the commissioning of a device, data should be sent and received between devices using valid network credentials. Secure communication should provide secure routing information exchange, message forwarding, and device-to-device data delivery. This could mean for example enabling the encryption option in BACnet for the BACnet communication protocol or using more secure protocol alternatives like BACnet Secure Connect [153]. Additionally, secure channels should be used whenever possible and interrupt countermea-

sures taken to guarantee data availability.

Evaluation Procedure - The communication assessment has four possible outcomes (Comms - ●/●/●/●). For each BA device review its network traffic. If device communication and/or data exchange *is in* plain text, this is a very poor security practice (Comms - ●). If data is not transmitted in plain text, but network senders and receivers *are not* authenticated and using state of the art encryption, the implementation is susceptible to attacks on data integrity and confidentiality (Comms - ●). If on the other hand data is encrypted and devices authenticated, but session keys/nonces are not used, the replay attack could be possible on the BAS (Comms - ●). A recommended approach is to ensure data encryption with the use of session keys/nonces for guaranteeing data freshness and to always maintain up-to-date device authentication techniques (Comms - ●).

Upgrading and Patching (UP)

One of the important security aspects of BASs is to upgrade and patch the system devices and protocols for mitigating network vulnerabilities and countering state-of-the-art attacks and exploits. The frequency (e.g., every one, two, three, and etc. months) of the upgrades/patches along with the underlying method (e.g., authenticated online, hard copy shipping) play a prominent role in building a secure BAS [154].

Evaluation Procedure - The upgrading and patching assessment has four possible outcomes (UP - ●/●/●/●). Review each device's manual and vendor website, if there are no mechanisms for upgrading/patching, local or remote (UP - ●). In the same vein, if remote device upgrading/patching is possible, but the delivery mechanism is not via a secure channel (UP - ●), this could leave network devices vulnerable to software/firmware attack. Remotely performing upgrading and patching over secure channels (UP - ●) is a better practice, but the recommended approach is to locally perform upgrades and patches with only vendor verified secure media (UP - ●).

Leaving (L)

Leaving refers to the removal or decommissioning of a device from the BAS network and the destruction of sensitive information about the commissioned device and network [155]. This information destruction should be performed in a secure manner, or risk the exposure of network credentials to malicious parties which could threaten the security of the entire BAN.

Evaluation Procedure - The leaving assessment has two possible outcomes (L - ●/●). Check each device's network decommissioning scheme, if the procedure does not include the destruction of local and network credentials this is a poor security practice (L - ●). Removing network credentials used by devices that are no longer active as well as ensuring the propagation of this removal throughout the network is the recommended method (L - ●).

Data Warehousing (DW)

Data warehousing is a key part of any BAS and is the practice of storing network traffic, metrics, and other data for a fixed period of time. This saved information can be used for backup purposes or for restoring the system to a normal state in the event of an emergency cyber incident. Historical data can also help system security analysts perform network forensics following a cyberattack which helps them better understand system vulnerabilities and weak points, as well as the extent of the incident. The data warehouse must store the system logs in a secure environment and in an encrypted way [156].

Evaluation Procedure - The data warehousing assessment has three possible outcomes (DW - ●/●/●). Review all data warehousing methods, if historical network data details are not collected (DW - ●), this is bad practice. Without historical data, there can be no network baseline for troubleshooting and debugging. Conversely, if network data backups are collected, but stored unencrypted and remain accessible on the main BAN (DW - ●), this is prone to snooping and could be stolen. The recommended approach is to collect network

data backups and store them encrypted and off the BAN or offline (DW - ●) entirely.

Intrusion Detection and Prevention (IDP)

An intrusion detection and prevention (IDP) system in a BAS is defined as a software application which monitors the network traffic and detects any malicious activities or policy violations [157]. There are two general types of IDP systems: i) signature based, and ii) anomaly based. The signature-based IDP system analyzes the network traffic to recognize any suspicious patterns similar to the known malware signatures. The anomaly-based IDP system analyzes patterns of the network traffic to find suspicious activity and can leverage machine learning to do so.

Evaluation Procedure - The intrusion detection and prevention assessment has four possible outcomes (IDP - ●/●/●/●). Consider the currently implemented intrusion detection and protection techniques for each device, if there are none then (IDP - ●). IDP techniques/mechanisms are necessary and useful for monitoring BAS state and overall network performance. If there is an IDP system, then it should be regularly updated (IDP - ●). An out of date IDP system (IDP - ●) may fail to alert on critical security events and render itself effectively useless. The recommended implementation is to use multiple combinations of IDP mechanisms and techniques, making sure to keep them all up-to-date (IDP - ●).

4.6 Multiprotocol Testbed Evaluation

In the following subsections, we will discuss the result of our evaluation for each device by protocol. The evaluation covers 19 real BA devices that make up the largest multiprotocol testbed, to the authors knowledge, in the literature. The devices were selected based on their membership in one of the three most popular BAS domain types: HVAC, lighting, and access control [158]. Our evaluation also includes the ICDs (gateways, routers, etc.) necessary for communicating from the local testbed to the engineering workstation.

Table 4.4: Testbed Evaluation Results

	Device Name	Device Level								Network Level						
		Software/Firmware			Side-Channel				Physical							
		OS*	Vendor	SW/FW	Mech	T	Mag	EM	P	NA	CX	Comms	UP	L	DW	IDP
BACnet	Laidlaw Orbis	4.3	–	–	○	○	○	○	●	●	●	●	●	●	●	●
	JCI TEC3612	4.3	6.5	–	○	○	○	○	●	●	●	●	◒	●	●	●
	CControls BASstat	4.3	–	–	○	○	○	○	●	●	●	●	●	●	●	●
	Wattstopper LMPL-201	4.3	–	–	–	–	–	–	●	●	●	●	●	●	●	●
	Wattstopper LMPX-100	4.3	–	–	○	●	○	◒	●	●	●	●	●	●	●	●
	Wattstopper LMLS-400	4.3	–	–	○	○	○	◒	●	●	●	●	●	●	●	●
	CControls BASrouter	–	–	–	–	–	–	–	●	●	●	●	◒	●	●	●
KNX	eelectron TR32A29KNX	4.3	–	–	○	○	●	○	●	●	●	●	●	●	●	●
	Schneider MTN6005	4.3	6.8	–	○	○	○	○	●	●	●	●	●	●	●	●
	Schneider MTN6221	4.3	6.8	–	○	○	○	○	●	●	●	●	●	●	●	●
	Siemens 5WG1258	4.3	6.6	–	○	●	○	●	●	●	●	●	●	●	●	●
	Schneider MTN631619	4.3	6.8	–	○	●	○	●	●	●	●	●	●	●	●	●
	Siemens N148/22	–	6.6	–	–	–	–	–	●	●	●	●	●	●	●	●
LonWorks	A Plus LonServer	4.3	–	–	○	○	○	○	●	●	●	●	●	●	●	●
	Honeywell T7350h	4.3	7.5	–	○	○	○	○	●	●	●	●	◒	●	●	●
	JCI TEC2216	4.3	6.5	–	○	○	○	○	●	●	●	●	●	●	●	●
	Hubbell LXPSCMFT	4.3	–	–	○	○	○	◒	●	●	●	●	●	●	●	●
	e-controls e-Multisensor	4.3	–	–	○	●	○	◒	●	●	●	●	●	●	●	●
	Echelon i.LON SmartServer	–	7.3	9.8	–	–	–	–	●	●	●	●	◒	●	●	●

* The OS of the engineering workstation which was used for device commissioning and/or control.

4.6.1 BACnet Results

For assessing our proposed BAS security evaluation framework on the BACnet protocol we used the following 7 devices:

- **Laidlaw Orbis:** Door Controller with Card Reader
- **JCI TEC3612:** Thermostat
- **CControls BASstat:** Thermostat
- **Wattstopper LMPL-201:** Lighting Controller
- **Wattstopper LMPX-100:** PIR Occupancy Sensor
- **Wattstopper LMLS-400:** Closed Loop Photosensor
- **CControls BAS router:** Multi-network Router

Software/Firmware

The Laidlaw Orbis door controller requires access control software to be installed on an engineering workstation for adding users, registering access cards, and generally managing the access control system. The only known vulnerabilities reported about this device are those associated with the OS of the engineering workstation. The software runs on Windows 10 and its CVSS score (OS - 4.3) is referenced in the Table 4.4. The other BACnet testbed devices require no common software for commissioning, but were managed with Yabe [159] running on Windows 10 (OS - 4.3). Another finding was that while there are no published vulnerabilities associated with the SW/FW of any of the BACnet testbed devices, the vendor of the JCI TEC3612 has a known CVSS score (Vendor - 6.5). We should note that the lack of reported vulnerabilities does not imply promising security, but rather suggests there has been little security analysis to date on these devices.

Side-Channel

For the acoustic side-channel, we varied the frequency of the sinusoidal waves from 100 to 15000 Hz by steps of 10 Hz. The amplitude of all the sinusoidal waves was set to 0.1 (and then varied from 0.0 to 1.0). *More Details on the Statistical Tests:* For each device once the data collection process completed for all frequency values, we obtained a frequency response from the control group and a frequency response from the experiment group, as shown in Figure 4.5 for the Wattstopper LMLS-400 (photosensor). Both the control group and the experiment group varied over frequency, and this is because the data collection process lasted for 12 hours. As time progressed, the testbed was subject to increased sunlight which affected both the control group and the experiment group. The control group helped us understand the influence of the environment over time, but to reduce the influence of environmental change, we subtract the control group from the experiment group. Additionally, we take the difference between adjacent values in the control group (i.e., threshold), to see how much variation exists in the pure signal. If the difference between the experiment group and the control group exceeds the threshold, then we consider the influence of the injected side-channel on the selected attribute of the target device to be statistically significant. From Figure 4.5, we can conclude that the acoustic side-channel does not have significant influence on the WattStopper LMLS-400. Similar results were observed for every BACnet testbed devices, and thus (Mech - ○). For the sake of brevity, plots showing the lack of influence due to the acoustic side-channel are excluded.

For the EM side-channel, both optical and IR experiments were performed. For the optical experiments, a projector was used to emit light (white/red/black) to fully cover each testbed device. White light resulted in the highest lux values, representing extreme daylight, red light resulted in mid-level lux values slightly above normal/ambient conditions, and black light was broadcast as the control equivalent of ambient/normal conditions. Table 4.5 shows the full multiprotocol testbed optical side-channel results. Multiple tests were run at both day and night time hours to confirm the experimental results and it was found that

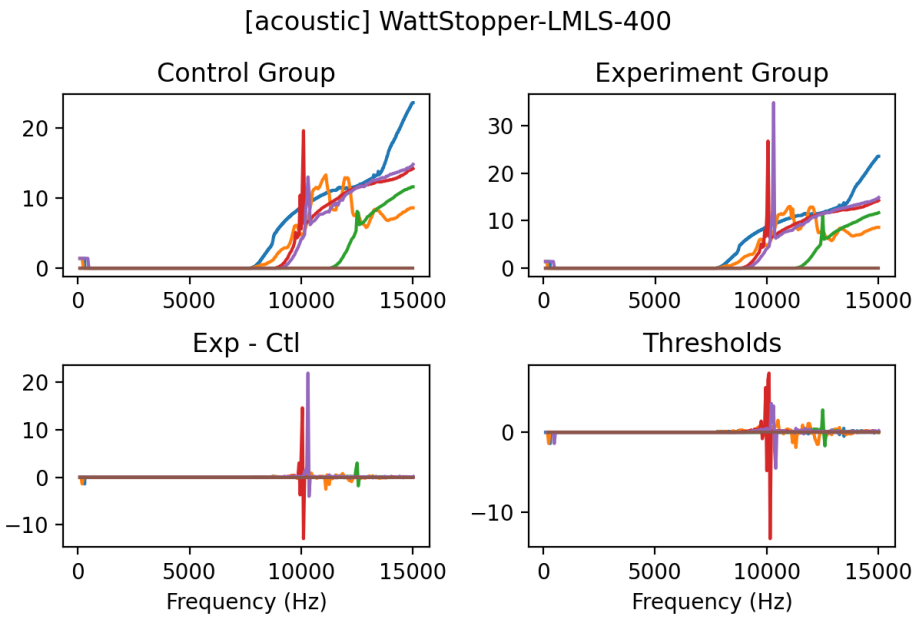


Figure 4.5: Acoustic side-channel analysis results for the light level of the Wattstopper LMLS-400.

there was no significant influence on the operation of any BACnet device due to the optical side-channel. The other half of the EM experiments were performed with IR flashlights and it was found that Wattstopper LMPX-100 and LMLS-400 were highly susceptible to interference. For the experimental design, a light chamber (with a single opening cut at the top) was constructed to fully enclose each device, depicted in Appendix E. Two IR flashlights were used, 850nm and 940nm, to shine IR beams into the opening of the light chamber. It was discovered that the LMPX-100 falsely detected motion activity when the 940nm IR light was shone into the otherwise dark chamber, which could be used by an attacker to remotely feign room activity when the room is truly empty. LMPX-100 responded because it operates by the pyroelectric sensor, which simply converts IR lights into voltages. When a human being is present, the IR lights from the human being activates LMPX-100. However, since LMPX-100 does not distinguish sources of IR lights, any IR light source over a certain power can activate LMPX-100. The LMLS-400 began reporting extremely high lux values contrary to the true dark nature of the chamber. This could be used by an attacker to prevent the trigger of other BA devices, such as a lighting system such

that a space stays dark for dubious activity to go undetected. The LMLS-400 responded to IR light because its light sensors, usually CCD or CMOS, inherently respond to IR lights whose wavelengths are close to those of visible lights. Normally a camera has a filter in the lens to filter out IR lights. If such a filter is missing, the light sensor will be affected by IR lights. For these reasons, both of these devices have an (EM - ◑) and the rest of the BACnet testbed devices have (EM - ○).

Table 4.5: Optical side-channel experimental results, table values represent the average influence observed over 10 trials. Binary observations encoded as 1 for influence or 0 for no influence.

	Device Name	Projector Brightness Percentage				
		0%	25%	50%	75%	100%
BACnet	Laidlaw Orbis	0	0	0	0	0
	JCI TEC3612	0	0	0	0	0
	CControls BASstat	0	0	0	0	0
	Wattstopper LMPL-201	0	0	0	0	0
	Wattstopper LMPX-100	0	0	0	0	0
	Wattstopper LMLS-400	0	0	0	0	0
	CControls BASrouter	–	–	–	–	–
KNX	eelectron TR32A29KNX	0	0	0	0	0
	Schneider MTN6005	0	0	0	0	0
	Schneider MTN6221	0	0	0	0	0
	Siemens 5WG1258	0	0.7	0.8	1	1
	Schneider MTN631619	0	0	0	0.9	1
	Siemens N148/22	–	–	–	–	–
LonWorks	A Plus LonServer	0	0	0	0	0
	Honeywell T7350h	0	0	0	0	0
	JCI TEC2216	0	0	0	0	0
	Hubbell LXPSCMFT	0	0	0	0	0
	e-controls e-Multisensor	0	0	0	0	0
	Echelon i.LON SmartServer	–	–	–	–	–

For the thermal side channel experiments, a heat gun was stationed approximately 10 inches below each target device and systematically transitioned between the on (400°F) and off (no heat applied) state as well as between an in-motion and motionless state. It was found that for the LMPX-100 when the heat gun was applied motionless, movement was registered by the device. This is a significant influence (T - ●) because the occupancy sensor operates by detecting an increase of IR energy in the environment, regardless of the

sources of the IR lights. The thermal side channel could be used by an attacker to feign activity in a space for distracting from other malicious activity in the room.

For the magnetic side channel, the experimental design involved presenting a stack of powerful permanent magnets to each device of interest and monitoring device operation and network traffic. Due to the constant nature of the magnetic field from the permanent magnet there was no interference or influence observed for any of the BACnet testbed devices (Mag - ○). Appendix E shows the side-channel experimental design for each modality explored in this chapter.

Physical

The entire testbed is located in an entry controlled room which requires the presentation of specific user access cards (physical barriers are present). Also, we perform regular inspections of the testbed devices every week (the third criteria is satisfied). However, we could not find any evidence of any tamper proofing for any BACnet testbed device (the second criteria is not satisfied). Given we have satisfied two criteria, the physical security evaluation of all BACnet testbed devices is (P - ●).

Network Access

All testbed devices were installed on a single platform with basic physical barriers and no network protection. Therefore, if physical entry can be obtained to the testbed the network would be readily accessible by potential attackers, this gives us (NA - ●) for all testbed devices.

Commissioning

Due to the lack of secure mechanisms employed for the commissioning of the BACnet testbed devices any device can be connected or replaced in the network without authentication. This is a poor (yet standard in real BASs) security implementation and so (CX - ●)

for all BACnet testbed devices.

Communication

In the BACnet protocol, devices on the network communicate in plain text, which leaves them vulnerable to snooping, MITM attacks, and many others (Comms - ●). It should be noted that BACnet Secure Connect (BACnet/SC) is an addendum to the BACnet protocol recently released by the ASHRAE BACnet Committee. It provides a secure, encrypted data link layer that is specifically designed to meet the requirements, policies and constraints of minimally managed to professionally managed IP infrastructures [153].

Upgrading/Patching

For the majority of the BACnet testbed devices (Laidlaw Orbis, CControl BASstat, Wattstopper LMPL-201, Wattstopper LMPX-100, and Wattstopper LMLS-400), there are no defined methods for upgrading/patching (UP - ●). This is a bad security practice which could leave their software/firmware potentially vulnerable to attacks. For the two other devices (JCI TEC3612 and CControl BASrouter), there are at least unverified upgrades available online through the vendor website (UP - ●).

Leaving

Once BACnet devices are connected to a BAN, a routing table is automatically saved in the memory of the device for enabling connections. By disconnecting a single device from the network, the attacker could extract this routing table for performing network reconnaissance and figuring out network topology. Since there are no secure leaving techniques employed, all of the BACnet testbed devices have (L - ●).

Data Warehousing

For all of the studied protocols, we collect network traffic in the engineering workstation for data analysis. Because this data is stored unencrypted (DW - ●) for all testbed devices. With this practice, in the event of compromise, a potential attacker could snoop on or steal the backed up network traffic.

Intrusion Detection and Prevention

In our testbed, we do not have any IDP system implemented for any of the protocols (IDP - ●) for all testbed devices.

4.6.2 KNX Results

For assessing our proposed BAS security evaluation framework on the KNX protocol we used 6 devices:

- **eelectron TR32A29KNX:** Door Controller/Card Reader
- **Schneider MTN6005:** CO2, Humidity, & Temp. Sensor
- **Schneider MTN6221:** Thermostat
- **Siemens 5WG1258:** Presence Detector
- **Schneider MTN631619:** Movement Sensor
- **Siemens N148/22:** IP Interface

Software/Firmware

The KNX protocol uses an exclusive software, ETS5 [160], for network commissioning. Consequently, the OS CVSS score required by the engineering workstation software was given (OS - 4.3). Additionally, most of the BA devices in the KNX protocol testbed are

from prominent vendors such as Siemens (Vendor - 6.6) and Schneider (Vendor - 6.8), whose CVSS scores are publicly available. However, there are no published vulnerabilities associated directly with the software/firmware of any KNX testbed devices. Just as with the BACnet protocol, this does not mean that they do not have any vulnerabilities, just that there should be more analysis by the ICS device security community for identifying zero day vulnerabilities.

Side-Channel

We performed the acoustic side-channel analysis for KNX the same way as was done for BACnet, and just as was found in Section subsubsection 4.6.1, the influence was negligible on every device (Mech - ○), and so, the detailed results are excluded for brevity.

For the EM side channel, both optical and IR experiments were performed just as described for BACnet in Section subsubsection 4.6.1. Two devices, Siemens 5WG1258 and Schneider MTN631619, were determined to be vulnerable to the optical side channel. For the Siemens device, it was discovered that upon shining white light directly at the device there was a *blinding* effect where the device could not accurately detect motion for several minutes. After the extensive periods of white light applied, the sensor consistently (> 5 min) became erroneously unresponsive to motion activity (EM - ◐). This could be leveraged by an attacker to *blind* the device then move around a space freely doing whatever they please, meanwhile the sensor detects nothing. The Schneider on the other hand suffered from extremely degraded performance upon extensive exposure (> 5 min) to the white light. During the exposure, the device could only detect some motion activity in the room as opposed to all activity as it did before the light was applied. We suspect that the visible lights may have activated pyroelectric sensors in the occupancy sensors, just as they would for infrared lights. The visible lights were so strong in intensity that the voltages in the occupancy sensors got saturated. As a result, the occupancy sensors could no longer respond to legitimate IR light. Both the Siemens 5WG1258 and Schneider MTN631619

were also found to be vulnerable to the IR side channel. By placing the Siemens presence detector in the light chamber and shining the both the IR850 and IR940 flashlights, false brightness values were read on the scale of 100x above the true brightness measurement. Similar results were observed with the Schneider movement sensor, and additionally both flashlights (IR850 and IR940) were able to trick the sensor into believing motion activity occurred when there was not. This means that with a long range IR transmitter properly aimed, an attacker could potentially stand outside of a building's window or door and trigger false activity. This activity could set off security alarms if motion detectors are used and lead to costly false alarms or even serve as diversions for other malicious payloads. For this reasons these devices have a (EM - ●). For every other KNX device no influence was observed from the EM side channel (EM - ○).

For the thermal side channel, we identified the same side channel influence as with the BACnet devices previously discussed in the Siemens 5WG1258 and Schneider MTN631619 (T - ●). This makes sense because PIR devices work by detecting heat energy in the environment [161]. No other KNX device experienced any significant influence due to the thermal side channel (T - ○).

For the magnetic side channel, the same experimental design was used as described for BACnet. The only device that showed influence to the magnetic side channel was the eelectron card reader (Mag - ●). In the presence of the magnets, the reader struggled to scan access control cards. Table 4.6 shows the measured operational delay from each BA device in the presence of the magnets. We speculate, that with a much stronger magnetic field (generated by the presence of several magnets), an attacker could effectively cause DoS in the door access control system. There was no interference or influence observed for any other KNX testbed device (Mag - ○).

Table 4.6: Average delay(s) of 10 run magnetic side-channel experiments

	Device Name	Number of Magnets					
		0	1	4	8	12	16
BACnet	Laidlaw Orbis	0	0	0	0	0	0
	JCI TEC3612	0	0	0	0	0	0
	CControls BASstat	0	0	0	0	0	0
	Wattstopper LMPL-201	0	0	0	0	0	0
	Wattstopper LMPX-100	0	0	0	0	0	0
	Wattstopper LMLS-400	0	0	0	0	0	0
	CControls BASrouter	–	–	–	–	–	–
KNX	eelectron TR32A29KNX	0	2.99	5.04	6.37	6.32	7.04
	Schneider MTN6005	0	0	0	0	0	0
	Schneider MTN6221	0	0	0	0	0	0
	Siemens 5WG1258	0	0	0	0	0	0
	Schneider MTN631619	0	0	0	0	0	0
	Siemens N148/22	–	–	–	–	–	–
LonWorks	A Plus LonServer	0	0	0	0	0	0
	Honeywell T7350h	0	0	0	0	0	0
	JCI TEC2216	0	0	0	0	0	0
	Hubbell LXPSCMFT	0	0	0	0	0	0
	e-controls e-Multisensor	0	0	0	0	0	0
	Echelon i.LON SmartServer	–	–	–	–	–	–

Physical

The KNX testbed devices are installed in the same room and on the same platform as the BACnet devices. Thus they also satisfy two assessment criteria (physical access barriers and regular device inspection). We could not find any evidence to suggest the tamper-proofness of any KNX testbed device, therefore (P - ●).

Network Access

Just as with the BACnet protocol, the KNX network can be accessed physically without any network barrier. Hence (NA - ●) for all devices.

Commissioning

The commissioning of the KNX testbed devices are performed using ETS5 (the common engineering workstation software) and a physical programming button located on each device. Although there is some level of authentication in the commissioning process, there are no certificates exchanged, and no implementation of encryption (CX - ●) for all KNX testbed devices.

Communication

Similar to the BACnet protocol, KNX devices communicate over the network in plain text and so (Comms - ●) for all the testbed devices.

Upgrading/Patching

For the eelectron TR32A29KNX access control device, there is no available upgrading/patching mechanism defined by the vendor (UP - ●). The rest of the KNX testbed devices use exclusive vendor specific software for upgrading/patching, but the process is not encrypted and thus susceptible to MITM attacks (UP - ●).

Leaving

Unlike the BACnet protocol, after disconnecting a device from the KNX BAN, no information about the KNX network is stored locally. Therefore all the KNX testbed devices have (L - ●).

Data Warehousing

As mentioned in the BACnet data warehousing section, there is a global system for the collection and storage of network traffic for every testbed protocol evaluated, so (DW - ●) for KNX testbed devices.

Intrusion Detection and Prevention

Just as with BACnet, there is no IDP system so (IDP - ●).

4.6.3 LonWorks Results

For assessing our proposed BAS security evaluation framework on the LonWorks protocol we used 6 devices -

- **A Plus LonServer:** Door Controller with Card Reader
- **Honeywell T7350h:** Thermostat
- **JCI TEC2216:** Thermostat
- **Hubbell LXPSCMFT:** Controller & Photosensor
- **e-controls e-Multisensor:** Light & Motion Sensor
- **Echelon i.LON SmartServer:** Router

Software/Firmware

Similar to KNX, there is a common engineering workstation software, IzoT Commissioning Tool [162], required for LonWorks device commissioning. Thus the OS CVSS score for this software is given for every device (OS - 4.3) in Table 4.4. Interestingly, there are three vendors (JCI, Honeywell, and Echelon) which have CVSS scores (Vendor - 6.5/7.5/7.3), but the only device that has a SW/FW vulnerability is the Echelon i.LON SmartServer (SW/FW - 9.8). For this router there were multiple scores from the various repositories, but the maximum was given for representing the highest reported level of insecurity. This score is suggests severe vulnerability that could be exploited remotely through an attacker with basic security knowledge. As with the other testbeds, even though the other devices do not have any published vulnerabilities, they could have undiscovered zero day vulnerabilities.

Side-Channel

We performed the acoustic side-channel analysis for LonWorks the same way as was done for BACnet, and just as was found in Section subsubsection 4.6.1. The influence was negligible on every device (Mech - ○). So, the detailed results are excluded for brevity.

For the EM side channel, both optical and IR experiments were performed as described with BACnet in Section subsubsection 4.6.1. There was no significant influence observed due to the optical experiments on any LonWorks device, but for the IR experiments both the e-controls and the Hubbell were influenced (EM - ◑). They both were unable to make accurate brightness readings and suffered from highly enlarged values. Additionally, the e-controls could be triggered to detect motion when the IR940 flashlight was shone into the light chamber from above. No other LonWorks testbed device exhibited influence (EM - ○).

For the thermal side channel just as with BACnet and KNX, we identified the same side channel influence on the Lonworks e-controls e-Multisensor (T - ●). If heat was applied (@400°F 10 inches away) whether in motion or stationary motion activity was detected by the sensor. No other LonWorks testbed device experienced any significant influence due to the thermal side channel (T - ○).

For the magnetic side channel, we found no significant influence on any testbed devices (Mag - ○).

Physical

Since the LonWorks testbed devices were installed on the same platform as the KNX and BACnet devices, the same two criteria are satisfied (physical access barriers and regular device inspections). Additionally, there is no evidence that any of the LonWorks testbed devices are tamper-proofed. Therefore, just as with the other protocol devices (P - ●).

Network Access

The LonWorks testbed device NA evaluation scores are the same as the BACnet and KNX (NA - ●), for reasons previously mentioned.

Commissioning

Similar to KNX, the commissioning of the LonWorks testbed devices was done through IzoT Commissioning Tool (common engineering workstation software) and a physical programming button located on each device. In the case of LonWorks the programming button press, can be replaced by entering the device serial number. While this provides some authentication, there are no encryption methods utilized and thus (CX - ●).

Communication

Similar to the previous two protocols, the LonWorks testbed devices communicate in plain text and so all the testbed devices have (Comms - ●).

Upgrading/Patching

The A Plus LonServer, Hubbell LXPSCMFT, and e-controls e-Multisensor have no defined upgrading/patching mechanisms (UP - ●). For the Honeywell T7350h there is vendor exclusive software used for upgrading/patching, but the process is not encrypted (UP - ●). As for the remaining devices (JCI TEC2216 and Echelon i.LON SmartServer), there are unverified upgrades available on their respective vendor websites so (UP - ●).

Leaving

Similar to the KNX testbed devices, after disconnecting a LonWorks device from the BAN no information stored about network is readily available for use by a potential attacker (L - ●).

Data Warehousing

As discussed previously, every testbed network uses the same system for the storing network details (DW - ●).

Intrusion Detection and Prevention

Just as with the other protocols there is no implemented IDP system for any LonWorks testbed device so (IDP - ●).

4.7 Conclusion and Discussion

From our systematic literature review and evaluation framework results, we have identified several interdisciplinary open challenges that should be explored by a combination of the security research community, BAS operators, and the AEC industry for moving forward in BAS security domain.

4.7.1 Systemization Findings

The first finding is that complexity grows proportionally with the number of sensors and actuators used in a BAS and only grows more complex over time. We can see this complexity exhibited in the BAS communication space. Protocols of past were wired and serial in their communication schemes, but now wireless techniques are being adapted. With more interactions between network components and devices there has been an enlargement of the BAS attack surface which has ultimately translated into more vulnerabilities. As a research community there should be a greater effort in uncovering systematic means for detecting and resolving potential security vulnerabilities as they develop. As the smart building industry continually endeavors towards smart cities this will be of even greater importance in order to prevent widespread insecurity proliferation.

Another area we uncovered is the lack of empirical results and evaluations in performed

in the literature. We found few works that utilized testbeds and even fewer that leveraged real systems for analysis. This is likely due to the inherently difficult nature of evaluating BASs. Testbeds can be expensive to build and partnering with industry for accessing real systems is not always an option. Even once an industry partnership can be made, full-scale realistic security evaluations may not be permissibly performed due to concerns about potential infrastructure damage. Our findings show that universities are working to reverse this trend and they are more commonly making their resources available for research purposes. There is of course more work to be done in this area of interdisciplinary partnership and collaboration for researching, designing, and developing secure BAS solutions.

Two challenges that our systemization research found to be barely touched in the literature are stakeholder liability and human factor considerations. Throughout the life cycle of a building automation systems several groups may be contracted to work on them. From the building design through operation and demolition multiple parties are responsible for various system aspects, but there is typically no one wholly liable for the entire building. This trickles down to the BAS as well, when we consider network security this typically falls under the jurisdiction of the IT department, but facilities managers and operators also must have stake in order to keep everything running. Not to mention the complexities of the recursive relationships between contractors, who may be involved during any phase of building existence, but only deal with sub-parts of the physical system. With this level of entanglement there is a clear need for a better integration of supply-chain and organizational factors which would assist in defining BAS liability at each life cycle stage. This in turn will remove the ambiguity of identifying BAS security responsibility and lead to more secure BAS deployments.

In regard to human factor considerations the literature is very sparse in discussions of the people perspective of BAS security. Human beings are oftentimes the source of various security issues from system misconfigurations, to lack of security training and awareness. In addition to this, there is little insight on the impact of attacks like social engineering and

phishing which could exploit them. The open challenge here is quantifying the effects of human shortcomings on BAS security and designing usable security mechanisms to counter them.

4.7.2 Evaluation Findings

SW/FW Findings

Our detailed analysis shows that there are few known software/firmware vulnerabilities associated with BA devices in modern vulnerability databases. This does not make them more secure than common IoT devices, which have lots of known vulnerabilities. It just means that BA devices yet to be carefully examined by ICS security researchers. The current low number of discovered vulnerabilities empower attackers to find the zero day vulnerabilities for carrying out their malicious exploits. This is similar to what we observed in the ICS domain before the Stuxnet attack [163] in 2010. As shown in Figure 4.6 [164, 165, 166, 167, 168], the number of discovered ICS vulnerabilities exploded following the discovery of Stuxnet, it was a wake up call which pushed security researchers to look for security flaws in ICS devices. Ultimately, this has been beneficial for both legitimate consumers, for installing more secure devices in their plants, and vendors for patching their existing devices against discovered vulnerabilities. We believe that a similar trend could emerge in the BAS domain. Therefore, it would be beneficial to populate and enrich the vulnerability databases through the careful analysis BA device software/firmware.

Side-Channel Findings

Our analysis has also revealed that the BA devices are severely vulnerable from the side channel perspective. This is a relatively new attack vector which can be employed by adversaries to compromise BASs. There currently exists no effective mechanism or framework for learning of the possible side channel vulnerabilities associated with a given BA device. Legitimate users would need to perform very sophisticated analysis to identify these inse-

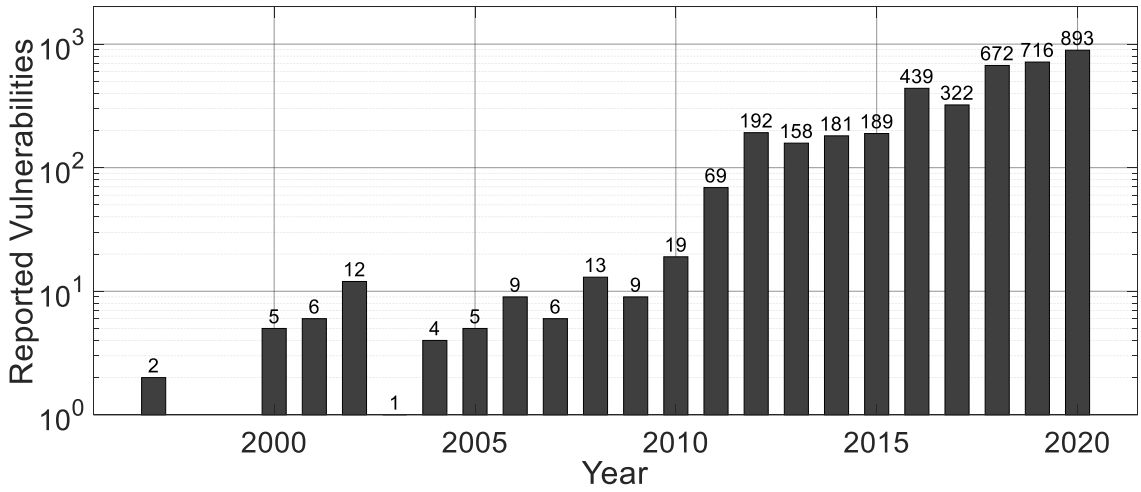


Figure 4.6: The number of discovered vulnerabilities in the ICS domain. Note- The scale is logarithmic and no data are available for 1998-1999.

curities. As a result we propose one possible future work in this domain as follows. ICS research community should develop a standard means for evaluating side channel vulnerabilities of BA devices and release their findings in a public database to be used by vendors (for patch development or hardware redesign) and customers (for identifying the most secure devices for their buildings). Independent security researchers could then contribute to this database by analyzing different devices for side channel vulnerabilities and report them to the database. The vulnerability database could be BAS specific and include the software/firmware as well as side channel security flaws for different devices from various vendors.

4.7.3 Conclusion

This chapter systematized the existing literature for building automation system security through a layer based approach. We leveraged the insights gathered from the systematization to develop a BAS security evaluation framework from the device and network level. We evaluated our proposed framework on the largest multiprotocol testbed (to the authors knowledge) presented in the literature. In our evaluation we answered our three research questions (RQs) as follows -

- **RQ1**: Can the security posture of BASs be measured?

  - Yes, but quantitatively making security judgements is an open research problem out of the scope of this work. As a result our proposed evaluation framework uses qualitative measures to make security recommendations based on well regarded security best practices and guidelines while taking into consideration the limitations of building automation systems.

- **RQ2**: What are the core assessment criteria to consider when exploring BAS security?

  - From investigation of the literature we identified 16 assessment criteria (section 4.5 for measuring the security assurredness of BA devices. These criteria span both the device and network level and take into consideration various aspects of BAS interactions. Some examples include physical access, software and firmware review, and even side-channel exploitability.

- **RQ3**: How does BA device interoperability affect the overall building automation system's security?

  - Four of the sixteen assessment criteria used in our proposed security evaluation framework consist of components involving device-device communication. Ensuring individual device integrity is critical to the soundness of the entire network. One compromised node could bring the whole system and thus device interoperability is tantamount with overall BAS security.

CHAPTER 5

INTRUSION DETECTION IN BUILDING AUTOMATION SYSTEMS

5.1 Introduction

Intrusion Detection Systems (IDSs) are typically hardware devices or software tools that monitor traffic for unusual activity and alert when the activities are detected. There are two umbrella classes under which most IDSs can be divided into, host based and network based [169]. Host based IDSs are installed on and exclusively monitor one host device, while network based IDSs are installed strategically on networks so they can monitor and collect traffic from multiple hosts. IDSs use a variety of techniques for performing detection including signature based, anomaly based, and specification based [27]. Signature based IDSs detect attacks by looking for patterns/signatures of malicious behavior, anomaly based IDSs detect attacks by creating a baseline of normal then alerting on deviations, and specification based IDSs use system documentation to determine expected behavior and detect attacks that stray from this behavior.

As the threats against BASs have grown, so have the popularity of intrusion detection systems proposed in the literature. The proposals vary from flow analysis techniques [28], to rule and anomaly based techniques [15, 29], and even specification based approaches [16, 137, 34]. Many of these works, [28, 15, 29, 16], suffer from false positives much too high to deploy in practice. BASs are critical systems with operational dependencies tied to many building functions. Erroneous IDSs which incorrectly identify and drop normal packets that were deemed *unsafe*, could lead to major BA device malfunction and potentially endanger the lives of building occupants.

Resultingly, there has been a increased need for more robust intrusion detection techniques. In a stride towards this, our approach leverages the use of two types of BAS device

documentation to generate rules for supplementing an IDS. Attacks on BASs generally occur in the form of unauthorized commands to insecure/vulnerable devices. In this work we perform network analysis of BACnet BAN traffic, monitoring the source and destination, to alert on traffic that violate pre-specified rules about the sender/receiver behavior. Our approach is lightweight, cleanly deployed, and passive in nature. We emphasized BACnet in this work because it is the most prevalently used BAS protocol to date [58], but our general methods apply to any protocol so long as sufficient documentation exists.

The previous chapters of this research focused on characterizing and exploring BANs through passive traffic analysis of a real BAS deployment on a university campus and systematizing the BAS literature to formulate an evaluation framework for measuring BA device security posture with an assessment performed on a multi-protocol testbed. In this chapter we will combine BAS insights acquired from the previous reconnaissance steps (Chapter 3 and 4) with Building Information Modeling (BIM) and file processing methods to develop robust network rules for specification based BAS intrusion detection systems. The main contributions of this chapter are -

- Development of the first specification based IDS that leverages BIM models as a supplement to Protocol Implementation and Conformance Statement (PICS) files
- Proposal of high accuracy documentation extraction and rule generation techniques
- Evaluation of intrusion detection rules on testbed of real BACnet devices

The remainder of this chapter is organized as follows, in section 5.2 we give the threat model, assumptions, and goals of this work. In section 5.3 we discuss our specification based IDS rule extraction approach and the BAS data sources that supplied it. Then in section 5.4 we discuss our testbed and evaluation results. Finally, section 5.5 summarizes and concludes the chapter.

5.2 Threat Model, Assumptions, and Goals

Building automation systems are different from most ICSs in there inherently open and accessible nature. By design buildings are constructed as infrastructure and can range from high security architecture like prisons to unbarred public entities like libraries or hospitals. Due to this fact it is infeasible to assume attackers can never gain access to a building's BAS. Thus for the purpose of this work we assume an adversary can perform both network and device exploits. From the device perspective they could leverage physical access to tamper with BAS firmware and alter field/control device behavior. From the network perspective we assume attackers can gain access to the BAN and send/receive messages from the BA devices. Network access could be obtained either locally (through physical connection) or remotely, since many modern BANs are Internet connected and searchable. Web search engines like Shodan [170] and Censys [171] have been developed specifically for the purpose of scanning networks for devices and uncovering threat surfaces. Further, we assume that our management layer network monitor can be compromised, but can be verified against a field layer tap which is trusted and adds redundancy.

For this research, we assume the building automation network layout documents are available to the BAS operator or are apart of the domain knowledge. This information can be accessed and used as a basis for the expected totals and types of BAS devices on a network. The premise is reasonable because BAN layout details can be deduced from commonly used building management software (Metasys [172], Niagara [173], etc) network views, aggregated from BAS network scanning tools (Yabe [159]), or even dynamically queried through programming APIs (BACpypes [174], BAC0 [175]). This mapping should at minimum include basic information mapping a device's network address with its vendor and model number.

The adversary's goal, for the sake of this research, is to accomplish three of the most salient attacks against BACnet networks [69]: interception, interruption, and modifica-

tion. Interception attacks are those concerned with stealing information about a BAN. This would primarily be useful for network reconnaissance such as learning device models/types and their capabilities and location. These attacks can be particularly hard to detect, because they do not disrupt control processes. Interruption techniques, such as DoS attacks, however are completely disruptive in nature with the end goal of making legitimate network resources unavailable. This can be accomplished in many ways, one of the most common are flooding attacks that aim to take advantage of BA device limited computing resources and overwhelm a device until it erroneously drops offline or can otherwise no longer do its job. The last and potentially most heinous type of studied attack are modification attacks which try to alter a control process and interfere with device operation. Sending unauthorized packets to devices which rework their standard object and property list is one way to do this.

5.3 Methodology

In this section we discuss our specification based IDS rule generation process for BACnet BANs. Our method starts with n processes for each data source utilized, but eventually converges to one. Here $n = 2$ because we leveraged both BACnet PICS files and BIM models for rule creation. These processes are semi-automated in nature (with some human intervention required) and can be run in parallel. A high level overview of our system model is given in Figure 5.1.

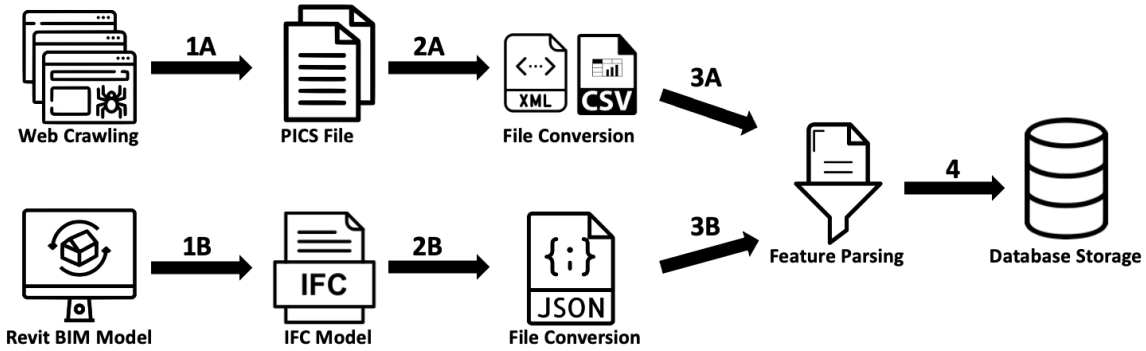


Figure 5.1: Data Source Specification Extraction Process

For our method PICS processing begins with web crawling on several BA manufacturer, vendor, and distributor websites. The web crawler performs keyword searches and then downloads all matching documents. The documents are then converted to XML/CSV format and parsed for relevant features before being stored in a database. Not so similarly, BIM models must be customized per building and thus our method starts with using Revit [176] to create a building model and infusing the model with relevant BACnet data. After model creation it is exported to an Industry Foundation Classes (IFC) file then to IFCJSON for ease of processing. IFC is the most commonly used data exchange format for BIM modeling and has been approved as an ISO 16739 international standard [177]. In the AEC industry IFC is useful for facilitating data exchange between different software applications across a building's life cycle and IFCJSON is just a JSON representation of the IFC data. Lastly, just as with PICS, the IFCJSON file(s) are parsed for features and stored in the database. Overlap is maintained by the inclusion of both PICS files and BIM model definitions in our database. By design PICS files are supposed to support a rigid set of features, but in practice and as implemented by BA manufacturers this is not always the case. Some common shortcoming of PICS files are -

- Erroneous = Oftentimes include grammatical, spelling, and other errors in the text
- Incomplete = Some documents do not contain full device details and capabilities as they are defined by the BACnet specification
- Difficult to Parse = Non-standard document formatting and layouts which vary per manufacturer lead to problematic information extraction
- Scattered Availability = Not always discoverable or readily available due to lack of standardized distribution mechanisms
- Non-specific = Generic elements for a device or series of devices, but deployed device characteristics could differ

Our proposed method attempts to overcome these complication with redundancy through BIM models. In addition to maintaining details that ordinary PICS may have, BIM models can be supplemented with *as-built* details. Once all data source processing is completed intrusion detection rules are defined as any object, property, service, or predetermined trait which is not in the listed in the database's set of capabilities for a given device. Then for any packet that is seen on the network it can be quickly analyzed and cross-referenced with the database for compliance. If it does not comply an alert is raised and BAS operators are notified.

It should be noted that not every violation will correspond to malicious behavior or an attack. In this context anomalous behavior is defined as device activity that contradicts allowable behaviors given by device documentation. For example, if there is a BACnet temperature sensor on the network which only supports the *ReadProperty* service (allowing other devices to read its temperature), it would be anomalous for this device to transmit a *WriteProperty* message or attempt to modify any other device. In the following subsections we talk about the PICS and BIM processing steps in more detail.

5.3.1 PICS

According to the BACnet specification [178], *all devices conforming to the BACnet protocol shall have a Protocol Implementation Conformance Statements that identifies all of the portions of BACnet that are implemented.* This means there should be a written document, example shown in Figure Figure 5.2, for every BACnet device which theoretically contains the following -

1. Basic information identifying the vendor and describing the BACnet device.
2. The BACnet Interoperability Building Blocks (BIBBs) supported by the device.
3. The standardized BACnet device profile to which the device conforms, if any.
4. All non-standard application services that are supported along with an indication

for each service of whether the device can initiate the service request, respond to a service request, or both.

5. A list of all standard and proprietary object types that are supported.

6. For each object type supported: any optional properties that are supported, which properties can be written-to using BACnet services, if the objects can be dynamically created or deleted using BACnet services, and any restrictions on the range of data values for properties.

7. The supported data link layer options, both real and virtual.

8. Whether segmented requests and responses are supported.

BACnet Protocol Implementation Conformance Statement

Vendor Name: McQuay International
Product Name: MicroTech Unit Controller
Product Model Number: Varies

Product Description

The MicroTech Unit Controller provides control for a variety of HVAC applications. It is connected to the BACnet network through a BACdrop gateway for BACnet. The MicroTech Unit Controller will appear to the BACnet network as a BACnet node offering BACnet services.

The MicroTech Unit Controller will appear to be BACnet riding on ISO 8802-3.

BACnet Conformance Class Supported

Class 1	☐	Class 4	☐
Class 2	☐	Class 5	☐
Class 3	☒	Class 6	☐

BACnet Functional Groups Supported

Clock	☐	Files	☐
HHWS	☐	Reinitialize	☐
PCWS	☐	Virtual Operator Interface	☐
Event Initiation	☐	Virtual Terminal	☐
Event Response	☐	Device Communications	☐
COV Event Initiation	☐	Time Master	☐
COV Event Response	☐		

(a) Device Description, Supported Classes, and Functional Groups

BACnet Standard Application Services Supported

Application Service	Initiate Requests	Executes Requests
AcknowledgeAlarm	☐	☐
ConfirmedCOVNotification	☐	☐
ConfirmedEventNotification	☐	☐
GetAlarmSummary	☐	☐
GetEnrollmentSummary	☐	☐
SubscribeCOV	☐	☐
UnconfirmedCOVNotification	☐	☐
UnconfirmedEventNotification	☐	☐
AtomicReadFile	☐	☐
AtomicWriteFile	☐	☐
AddListElement	☐	☐
RemoveListElement	☐	☐
CreateObject	☐	☐
DeleteObject	☐	☐
ReadProperty	☐	☒
ReadPropertyConditional	☐	☐
ReadPropertyMultiple	☐	☒
WriteProperty	☐	☒
WritePropertyMultiple	☐	☒

(b) APDU Services Supported

Figure 5.2: BACnet PICS document for a MicroTech Unit Controller

As stated earlier, what was found in practice is that a device's PICS file is not always accessible and does not always conform exactly to the BACnet. If the file can be found it

must be assessed and processed in accordance to its particular formatting and layout. To accomplish this our techniques has 4 steps, the first 3 are unique to the data source.

1A) Web Crawling

The first step in PICS processing is web crawling. In order to enable the thorough search and collection of as many PICS files as possible we used the Google Custom Search JSON API. The search engine takes in multiple queries and performs a search of PICS on up to 10 pages worth of results. We were limited to 10 queries/day and used them to explore specific BACnet vendor websites. Some vendor/manufacturer sites we investigated included *Johnson Controls*, *Siemens*, and *Honeywell* among others. Our method allows for easy domain swapping of different vendor website as well as interchangeable keywords. These terms filter out the majority of the search results allowing us to pinpoint relevant documents. After performing a search: the results are downloaded as a large JSON (shown in Figure 5.3, programmatically parsed in Python, and the appropriate PICS PDF files are downloaded and saved by query.


```
{
  "kind": "customsearch#search",
  "url": {
    "type": "application/json",
    "template": "https://www.googleapis.com/customsearch/v1?q={searchTerms}&num={count?}&start={startIndex?}&lr={language?}&safe={safe?}&cx={cx?}&sort={sort?}&filter={filter?}&gl={gl?}&cr={cr?}&googlehost={googleHost?}&c2coff={disableCnTw
{siteSearchFilter?}&exactTerms={exactTerms?}&excludeTerms={excludeTerms?}&linkSite={linkSite?}&orTerms={orTerms?}&relatedSite={relatedSite?}&dateRestrict={dateRestrict?}&lowRange={lowRange?}&highRange={highRange?}&searchType={searchType}&
{imgType?}&imgColorType={imgColorType?}&imgDominantColor={imgDominantColor?}&alt=json"
  },
  "queries": {
    "request": [
      {
        "title": "Google Custom Search - BACnet PICS",
        "totalResults": "1580",
        "searchTerms": "BACnet PICS",
        "count": 10,
        "startIndex": 1,
        "inputEncoding": "utf8",
        "outputEncoding": "utf8",
        "safe": "off",
        "cx": "d1c84e538e2231c8c"
      }
    ],
    "nextPage": [
      {
        "title": "Google Custom Search - BACnet PICS",
        "totalResults": "1580",
        "searchTerms": "BACnet PICS",
        "count": 10,
        "startIndex": 11,
        "inputEncoding": "utf8",
        "outputEncoding": "utf8",
        "safe": "off",
        "cx": "d1c84e538e2231c8c"
      }
    ]
  },
  "context": {
    "title": "PICS Search"
  },
  "searchInformation": {
    "searchTime": 0.317112,
    "formattedSearchTime": "0.32",
    "totalResults": "1580",
    "formattedTotalResults": "1,580"
  },
  "items": [
    {
      "kind": "customsearch#result",
      "title": "BACnet Protocol Implementation Conformance Statement (PICS)",
      "htmlTitle": "\u003cb\u003eBACnet\u003c/b\u003e Protocol Implementation Conformance Statement (\u003cb\u003ePICS\u003c/b\u003e)",
      "link": "https://www.downloads.siemens.com/download-center/download?A6V11572295",
      "displayLink": "www.downloads.siemens.com",
      "snippet": "Nov 13, 2018 ... 2/163. Siemens. Desigo™ V6.1 BACnet Protocol Implementation Conformance \nStatement (PICS). CM110665en_12. Building Technologies.",
      "htmlSnippet": "Nov 13, 2018 \u003cb\u003e...\u003c/b\u003e 2/163. Siemens. Desigo™ V6.1 \u003cb\u003eBACnet\u003c/b\u003e Protocol Implementation Conformance \u003cbr\u003e\nStatement (\u003cb\u003ePICS\u003c/b\u003e). CM110665en_1
      "formattedUrl": "https://www.downloads.siemens.com/download-center/download?...",
      "htmlFormattedUrl": "https://www.downloads.siemens.com/download-center/download?...",
      "pagemap": {
        "cse_thumbnail": [
          {
            "src": "https://encrypted-tbn3.gstatic.com/images?q=tbn:ANd9GcSmw4lJGoVc_GlebxY3jte5eGyxxdYaMnZ1-P81xYILX9Y-B-s0Nc75O32S",
            "width": "189",
            "height": "267"
          }
        ],
        "metatags": [
          {
            "moddate": "D:20181126115128+01'00'",
            "creator": "Microsoft® Word for Office 365",
            "creationdate": "D:20181126115128+01'00'",
            "producer": "Microsoft® Word for Office 365",
            "title": "BACnet Protocol Implementation Conformance Statement (PICS)"
          }
        ],
        "cse_image": [
          {
            "src": "x-raw-image:///3543436095432976b27f522e32b4f54533c58b3e8b1efe4be49c7ab9a1b119212"
          }
        ]
      },
      "mime": "application/pdf",
      "fileFormat": "PDF/Adobe Acrobat"
    },
```


Figure 5.3: Sample Web Crawl Results from PICS Crawler with Keyword 'BACnet PICS'

2A) File Conversion

The second step is file conversion. In order to process the PICS files in a more interpretable format we first convert them from PDF to XML and CSV. PDF analysis can be challenging due to the number and types of symbols, positioning, text arrangement, and tables used. Unlike text files that tend to lose a lot of critical information on conversion, the XML format records information about each character's position, width, and height for reference when parsing. Additionally, the CSV format is useful for extrapolating data from tables and other structured attributes that may get distorted or lost in the XML. To accomplish this we utilize the PDFMiner Python library, a text extraction tool especially designed for PDFs. We feed in the PICS file to PDFMiner and the output is the PICS file divided by pages, text boxes, table lines, and text in both XML and CSV format.

3A) Feature Parsing

The third step in PICS processing is feature extraction. Before beginning feature parsing we filter out all non-PICS from the downloaded files (XMLs and CSVs). To do this files are scanned for the keywords "PICS" and "Protocol Implementation Conformance Statement", if found they move on to get parsed otherwise they are discarded.

Passing XML files get passed through three different XML parsers, with techniques that vary in the handling of checkboxes, tables, and bullet points. These parsers were derived from the three most common converted formats of web crawled PICS files and was found to have good coverage across many different vendors. After parsing, the extracted information is taken from whichever parser returns the most complete results (no missing values). If none of the XML parsers return complete results, then the CSV file is parsed for the missing details. If the CSV parser also fails to return any results, the file is deemed unparsable, and the logs are notated for manual extraction. If parsing completes successfully (for an appropriately defined PICS file) we can extract 7 attributes. These features are vendor name, product name, product number(s), standard device profile, required BIBBs,

optional BIBBs, and standard object types.

Figure 5.4: Sample PICS PDF File Converted to XML and CSV

5.3.2 BIM

As the building automation industry transitions from single centralized BMSs that are made up of isolated sub-systems to more integrated and communicative distributed systems, so are the technologies used in building construction, planning, and management being updated. This revitalization can be best witnessed through the widespread adoption of BIM in the architecture, engineering, and construction (AEC) community. According to Autodesk [179], *building information modeling (BIM) is a process that begins with the creation of an intelligent 3D model and enables document management, coordination and simulation during the entire life cycle of a building project*. The life cycle starts at the planning stage, then to building design and construction, and lasts through the operation and maintenance phases. BIM models are used to design and document buildings and infrastructure at various levels of granularity and are passed between multiple industrial teams that all play critical roles in the project. Presently BIM models contain detailed information about the structural components of a building including walls, doors, windows, and even mechanical and plumbing components such as HVAC and piping. Figure 5.5 shows the 6 BIM levels of development (LOD) [180] which define the industry standard for the level of refinement achieved in building models. The lowest is LOD 100 and it provides only conceptual

details with basic information. The highest level is LOD 500 and at this stage elements are modeled as they exist and were built in the constructed building with high modeling accuracy.

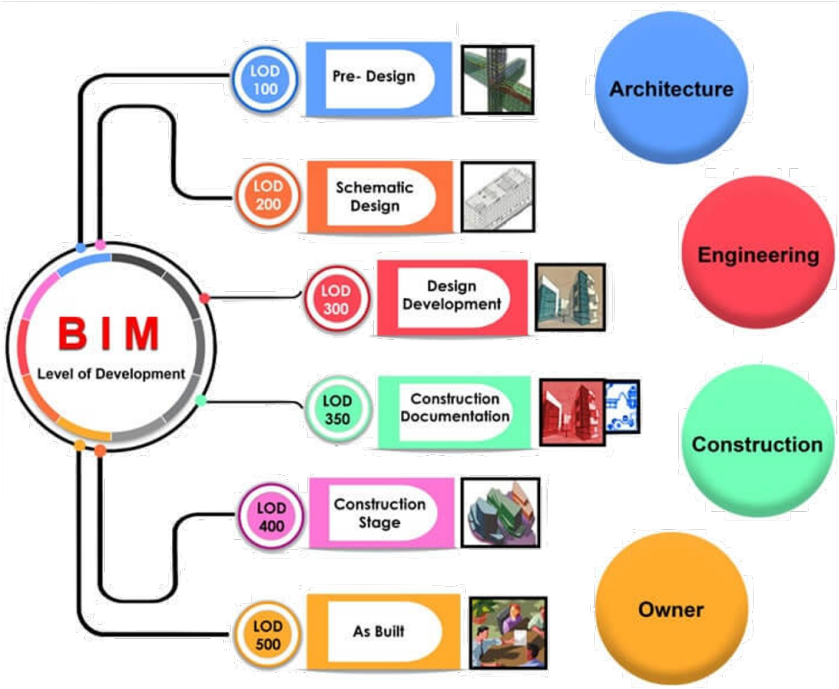


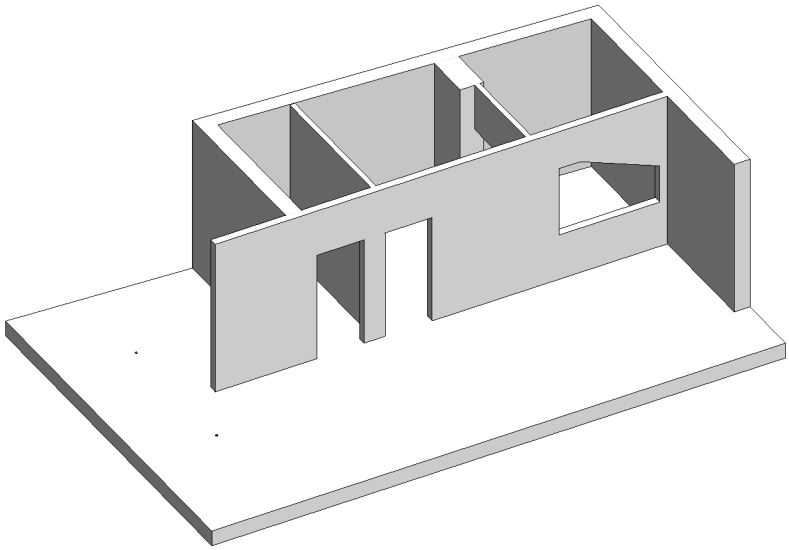
Figure 5.5: BIM Model Levels of Development [181]

Recent work [182] has taken great stride to include BACnet data into BIM models for facilitating information exchange in BIM assisted building automation design and operation. As the industry shift continues towards smart buildings, BIM representations will be utilized as the building's *digital twin*. As a step towards that, the infusion of BAN data in BIM (LOD 350 and above) is critical for maintaining the most complete representation of a building. Moving forward we expect to see BIM models used more and more for building network security.

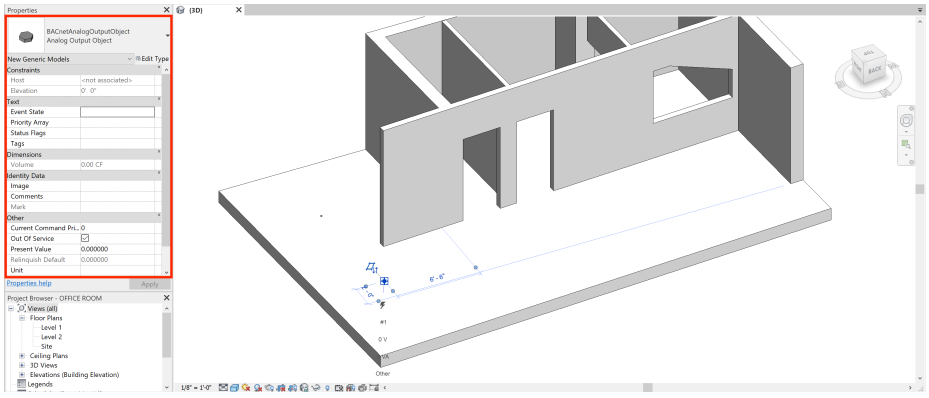
1B) Revit Modeling

For this work we created simple Revit models infused with BACnet data to expose device network characteristics, such as its properties, services, objects, and allow for the addition

of other BAS operator defined components. We consider our models to be LOD 350 and greater because at those levels building elements interface with various systems and as the AEC industry continues to evolve device networking will be one of those subsystems. Figure 5.6a [183] shows a sample rendering created in Revit for a small office. For this work we integrated BACnet details into Revit model in 2 ways.



(a) Sample 3D Rendering of a Small Office in Revit



(b) Sample 3D Rendering of a Small Office in Revit with BACnet Infused Data

Figure 5.6: Revit Rendering

The first is through the use of Revit Families (RFA) as proposed in [182]. Tang et al. established an Industry Foundation Class (IFC) subset schema for ensuring that BAS infor-

mation complying with the BACnet protocol can be represented in the IFC data model for information exchange between BIM tools throughout various project stages. The schema defines RFAs for each BACnet object (including property details) as shown in Figure 5.6b. We agglomerate multiple RFAs, representing several different BACnet objects that make up a physical BACnet object in each Revit model. The second method supplements the first by adding a variable amount of BACnet details into the Revit Project Information field to be parsed out of the IFC in step 3A. Figure 5.7 shows a screenshot of the sample Revit model BACnet details exported as an IFC file.

```
#763= IFCDISTRIBUTIONCONTROLELEMENT('1$zJ0hwlr5gQiyYakGtdgf',#41,'BACnetAnalogOutputObject:Analog Output Object:347976',$,'Analog Output Object',#762,#755,'347976',$);
#749= IFCCONTROLLERTYPE('1$zJ0hwlr5gQiyYakGtdiq',#41,'Analog Output Object',$,$,(#805,#899,#821),(#748),'347861',$,.MULTIPOSITION.);
#763= IFCPROPERTYSINGLEVALUE('Present Value',$,IFCREAL(0.00),$);
#784= IFCPROPERTYSINGLEVALUE('Description',$, IFCTEXT('Default'),$);
#778= IFCPROPERTYSINGLEVALUE('Status Flags',$, IFCTEXT('Default'),$);
#794= IFCPROPERTYSINGLEVALUE('Object Identifier',$, IFCTEXT('Default'),$);
#768= IFCPROPERTYSINGLEVALUE('Out of Service',$,IFCBOOLEAN(.F.),$);
#781= IFCPROPERTYSINGLEVALUE('Units',$,IFCTEXT('Default'),$);
#773= IFCPROPERTYSINGLEVALUE('Priority Array',$, IFCTEXT('Default'),$);
#777= IFCPROPERTYSINGLEVALUE('Status Flags',$, IFCTEXT('Default'),$);
#792= IFCPROPERTYSINGLEVALUE('Relinquish Default',$, IFCREAL(0.00),$);
```

Figure 5.7: BACnet Details from Sample IFC File of Small Office Model

In total for each device we infused 9 BACnet attributes into a BIM model. All device BACnet objects and their corresponding properties are added via the first modeling method with RFAs. The remaining 7: vendor name, product name, product number(s), required/optional BIBBs, segmentation, hours of operation, and packet size. Five of the aforementioned features can be derived from PICS files and serve for redundancy purposes. The remaining 2, hours of operation and packet size, are derived from BAS operator domain knowledge. Working knowledge of the BAN is required to define for each device values that make sense. For instance, if a gym is only operational from 6AM until 6PM and all the lights are turned off at closing it would be anomalous to see traffic from the BAS lighting system at 11PM. In this scenario the hours of operation on the light devices could be set as the business hours. Similarly with the packet size metric, if under normal conditions a device transmits packets of 50 bytes or less it would again be anomalous to see that device sending 300 byte packets. These 2 features serve as just a starting point for the customization that can be added to a BIM file through operator domain knowledge.

2B) File Conversion

JSON is a lightweight data-interchange format used prevalently on the web for sharing data. IFCJSON is format that was introduced in the AEC industry to address a handful of problems with IFC. Some of the problems include lack of awareness from the programming and development communities leading to increased data extraction efforts and the lack of adaptability to update with fast paced distributed design and construction products/projects. With IFCJSON there is a balance between compatibility with IFC schema and best practice human readable JSON representations. For file conversion our method leveraged a code base by buildingSMART [184] to convert our BIM models from IFC to IFCJSON. Figure 5.8 shows the sample output of the buildingSMART IFCJSON converter (left) as well as the refined IFCJSON (right) with only BACnet features which was fed into the feature parser.

Figure 5.8: IFC JSON after converting from raw IFC file (left) and IFC after separating BACnet features

3B) Feature Parsing

Similar to the PICS feature parsing, the goal of the BIM feature parser was to pull out the 9 attributes from the IFCJSON file. The heavy lifting of the data having primarily already been handled at stage 1B and 1A, feature extraction was fairly straightforward. The Python JSON library was used with keyword searches for each of the 9 desired attributes and the data was saved as a pandas dataframe and as intermediary before database storage.

5.3.3 Database Storage and Rule Formulation

Once the PICS files and BIM models are fully parsed they are stored in a local database instance. For this project, MongoDB [185] is used as the document database because of its scalability and flexibility, as well as its compatibility with Python. The data is stored in 2 separate collections within the database organized by device and divided into 7 features in the PICS collection and 9 features in the BIM collection as previously described.

As a final step once all specifications are parsed and stored in the database, the rules are straightforward. For any device indexed in the database if a network packet or series of network packets are identified which contradict the expected device behavior as defined by the database specifications, this is considered a violation or an anomaly. Essentially, there is a list of valid elements (properties, objects, services, etc) for every BACnet device in the database. For every packet seen on the network it gets -

1. Matched to 0, 1, or 2 entries in the database. This corresponds to either no documentation, one PICS/BIM element list, or both a PICS and BIM element list.

2. If the device has no documentation, an alert is raised to network administrators with details for future inclusion. If the device has at least 1 entry its application layer attributes are analyzed against the database element list.

3. If the packet's attributes are not found to match the elements in at least 1 of its database entries an alert is generated.

5.4 Testbed Experimentation and Evaluation

For evaluating our proposed specification based intrusion detection rule formulation approach we built a BACnet HVAC testbed consisting of 6 real BAS devices. Table 5.1 gives an overview of each device in the testbed. Additionally, as a pre-processing step to our evaluation, we manually created 4 BIM models for each of the 4 BAS device types in the

testbed and scraped 42 PICS files from the Internet with our web crawler. Each of these documents was then passed through all of the appropriate IDS rule formulation software framework steps.

Table 5.1: BACnet HVAC Testbed Devices

Vendor Name	**Product Name**	**Product Model Number**	**# of Devices**
Contemporary Controls	BASstat	BAST-221C B2	2
Johnson Controls	Terminal Equipment Controller	TEC3612-00-000	2
Siemens	RDY BACnet Thermostat	RDY2000BN	1
Temco Controls	Tstat8	Tstat8-H-C02	1

5.4.1 Implementation

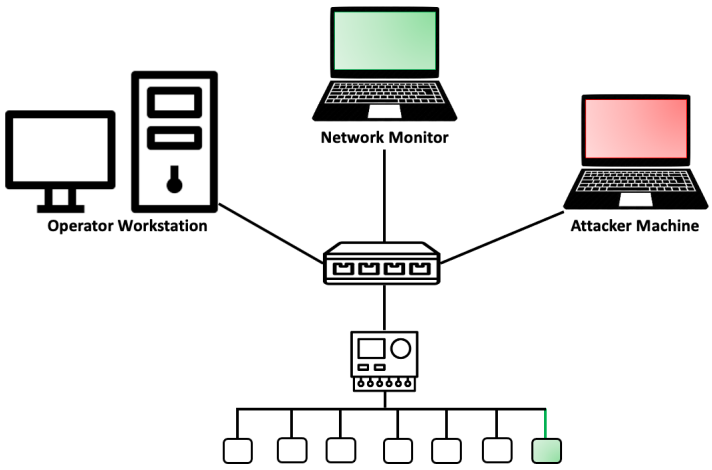


Figure 5.9: Testbed Experimentation Setup

There were 3 key machines utilized in our implemented approach, these are illustrated in Figure 5.9. The first machine is the operator workstation (Dell OptiPlex Desktop), in a real BAS this would be the machine used by the facility managers to monitor the building systems statuses. The Yabe software was used on the operator workstation to generate traffic and supervise device properties and statuses. A sample screen capture from the operator workstation software is shown in Figure 5.10. This machine was also running the

specification based IDS program which was developed using the Python Scapy [186] tool. Scapy is a packet manipulation program that can be used to forge and decode packets for many protocols. It is also commonly used for network scanning, probing, and attacking. For the purpose of this research Scapy was used for real time network level packet captures, with each packet received analyzed for attributes matching its specification. For our proof of concept implementation we assume a network mapping of the BA device type/name and address exists. This is a valid assumption because experienced BAS operators have domain knowledge that can close this gap, otherwise several publicly available BMSs can provide this information.

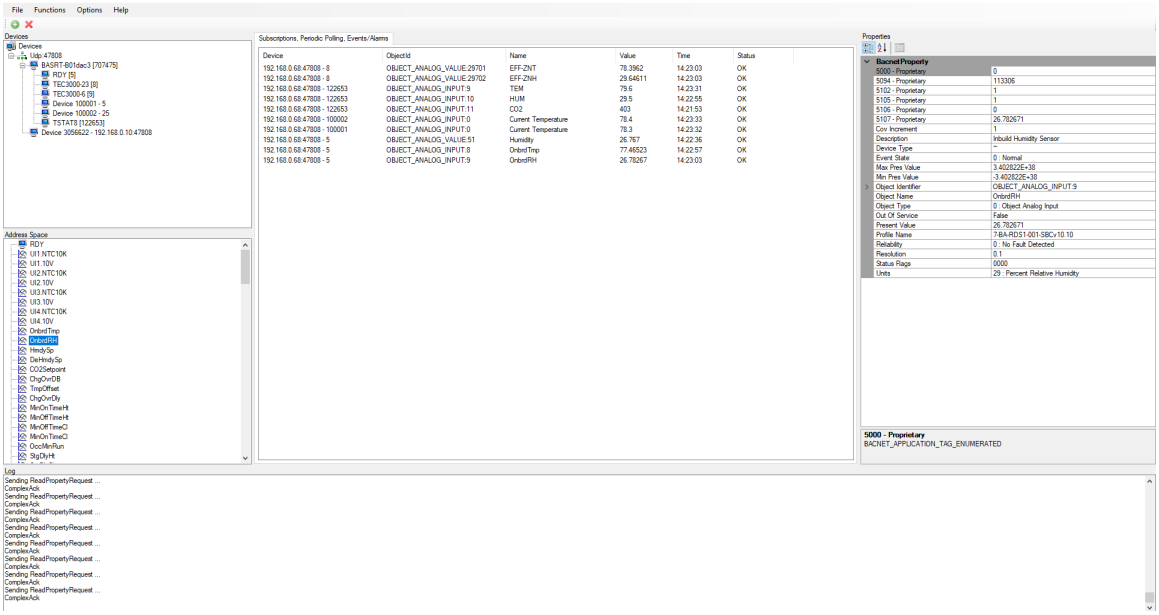


Figure 5.10: Screenshot of Yabe from Operator Workstation

The attacker machine (Dell XPS Laptop) was used to simulate attacks and test the IDS rules by injecting malicious packets onto the network with the Python BAC0 library. Our executed attacks are discussed in more detail in subsection 5.4.2.

The network monitor machine (Macbook Pro) was connected to the network via a mirror port on the switch (Netgear SG300-28) and monitor traffic from all nodes. The field layer tap (shown in green on the bottom of Table 5.1) was implemented with the BACnet capture tool, described in section 3.3, serves as a redundant checkpoint for ensuring the

BAS controller has not been compromised. This node captured field level traffic which was compared for consistency with the network monitor's IP level traffic.

5.4.2 Evaluation Results

For all the devices shown in Table 5.2 our intrusion detection rule formation software framework was run to extract the vendor name, product name, product number(s), standard device profile, required BIBBs, optional BIBBs, and standard object types from their PICS files. The table gives the precision and recall of our algorithm for the 3 most important BACnet details extracted from the file. For the purpose of this research, we define precision as the percentage of BACnet details that were correctly extracted from the documentation. Essentially measuring how often the extracted value matches the value represented in the document. Recall, is then the percentage of details that were extracted with respect to the total number of details that could be extracted. For example, recall is 33% if the goal was to extract 12 data points, but only 4 are extracted in practice. Just as with the PICS files, Table 5.3 gives the precision and recall of all the device BIM models that our algorithm ran.

For the purpose of evaluating the rule extraction mechanisms and the quality of the extracted rules in practice, we executed 5 attacks against our implemented testbed from 3 of the most salient BACnet attack domains (interception, interruption, modification). In the following subsections we elaborate on the specification interpretation results of our approach and discuss the outcome of unleashing 2 interception attacks, 2 interruption attacks, and 1 modification attack on our testbed.

Specification Interpretation

With our method we extracted about 857 BIBBs, 40 device profiles, and 472 object types from PICS files. Our approach had an overall precision score of 95.24% and an overall recall score of 94.64%. Most device PICS files were extracted in an automated fashion

Table 5.2: PICS file attribute extraction results. Table values represent number of BACnet attributes successfully extracted per device.

	Device	BIBBs	Device Profile	Object Types	Precision	Recall
CControls	BASgatewayLX	0	0	0	0%	0%
	BASpi-IO	8	1	6	100%	100%
	BASrouterLX	12	1	1	100%	100%
	BASstat	6	1	6	100%	100%
	BAScontrol20	0	0	0	0%	0%
	Portable BAS Router	6	1	1	100%	100%
	BASview	6	1	10	100%	100%
Honeywell	Eagle AX	35	1	15	100%	100%
	FieldServer	3	1	2	100%	100%
	TB7800 Thermostat	6	1	8	100%	100%
	TB7300 Thermostat	6	1	7	100%	100%
	TB7200 Thermostat	6	1	7	100%	100%
	TB7600 Thermostat	6	1	8	100%	100%
	EAGLEHAWK NX	54	1	18	100%	100%
	Excel Web II	7	1	4	100%	100%
	IRM Controller	19	1	24	100%	100%
	IRM N4	20	1	10	100%	100%
	Novar XIO	12	1	32	100%	100%
	VLCA-1688	19	1	15	100%	100%
	WEB-8000	35	1	16	100%	100%
	EAGLE	37	1	19	100%	100%
JCI	FAC	29	1	15	100%	100%
	ATC1510	19	1	10	100%	100%
	Color TEC	26	1	11	100%	100%
	E-Link Gateway	18	1	5	100%	100%
	FEC	16	1	10	100%	100%
	FX Controller	11	1	10	100%	100%
	FX-PC Controller	16	1	10	100%	100%
	NAE/NCE	45	1	28	100%	100%
	TEC	23	1	11	100%	100%
Siemens	PXC Unitary EC	39	1	15	100%	93.75%
	TC Unitary EC	39	1	15	100%	93.75%
	RDY Thermostat	13	1	8	100%	100%
	Intelligent Valve	12	1	10	100%	100%
	PXC Modular	42	1	15	100%	93.75%
	Climatix	26	1	20	100%	100%
	Desigo PX	45	1	25	100%	100%
	ATEC/PTEC	9	1	7	100%	100%
	PXC Compact	42	1	15	100%	93.75%
	VAV Compact Controller	9	1	9	100%	100%
Temco	Tstat8	12	1	7	100%	100%
	Bacnet Transducer	12	1	7	100%	100%

Table 5.3: BIM model attribute extraction results. Table values represent number of BACnet attributes successfully extracted per device.

Device	BIBBs	Object Types	Properties	Segmentation	Hours	Packet Size	Precision	Recall
CControls BASstat	6	6	–	2	1	1	100%	100%
JCI TEC	23	11	104	2	1	1	100%	100%
Siemens RDY	13	8	97	2	1	1	100%	100%
Temco Controls Tstat8	12	7	52	2	1	1	100%	100%

with no assistance, but over our evaluation we observed 3 errors responsible for reduced scoring on some files. The first error comes from the BASgatewayLX PICS file, which we discovered was automatically web crawled and downloaded. This file, it turns out was password protected and set with view only permissions. As a result, the feature parser failed to copy the searched BACnet details from the document. The second error that occurred was due to the format of the PDF document crawled for the BAScontrol20 device. This PDF file is a scanned document and as a result there is technically no text in the document, it was represented as a single image. The final error reoccurred in 4 Siemens PICS files and upon further investigation it was attributed to a misspelling in the device object types. The typo was the exact same in each file and read as ”Binary Inpute” instead of the searched keyword ”Binary Input”. This misspelling threw off the feature parsers search and resulted in a single missing object type for each device.

From the 4 BIM models our method extracted 43 BIBBs, 54 object types, 253 properties, 8 segmentation capabilities, 4 hours of operation, and 4 packet sizes. Due to the straightforward and structured format of the BIM models our algorithm ran with both 100% precision and recall.

Interception Attack

The first attack we performed on our testbed was the network discovery attack. This consists of an attacker running scripts for performing reconnaissance on a BACnet network. The goal of this is to learn the BA devices present on a network for determining potentially exploitable vulnerabilities. In this case the attacker's network discovery script requests 5 device properties, namely *vendor-identifier*, *vendor-name*, *application-software-version*, *model-name*, and *description*. From the attacker machine presented in section 5.3, we executed a custom script to probe each testbed device and triggered 7 alerts.

- CControls BASstat - Our testbed contains 2 of these devices, each which generated 0 alerts because their PICS file had no supported properties. Due to this a special

catchall (*.*) IDS condition was enacted that allows all property requests.

- JCI TEC - In our testbed there are 2 devices of this type which generated the same 3 alerts each. Alerts were triggered from the lack of the *vendor-identifier*, *vendor-name*, and *model-name* properties.

- Siemens RDY - No alerts were triggered by this device since all the requested properties are implemented.

- Temco Controls Tstat8 - One alert triggered from the lack of the *model-name* property.

The second attack we performed on our testbed was an impersonation attack. This attacker aims to trick BA devices into falsely updating their local routing tables or internal network map by impersonating a device that they are not. This type of attack would typically follow a reconnaissance step, leveraging learned network details to impersonate devices for snooping traffic. To do this a malicious and gratuitous *I-Am* message is sent from the attacker machine to redirect a legitimate devices traffic to itself. For this attack, we separately chose each testbed device as a target device and injected 3 false *I-Am* packets from the attacker machine impersonating the non-target devices. From these experiments, 9 alerts were triggered as follows -

- **Sender** [BASstat or RDY or Tstat8] —> **Receiver** [TEC] = 3 packets triggered the same 3 alerts due to the impermissible initiation of the *I-Am* service from all senders

- **Sender** [TEC or RDY or Tstat8] —> **Receiver** [BASstat] = 3 packets triggered 2 alerts due to the impermissible initiation of the *I-Am* service from the RDY and Tstat8 devices

- **Sender** [TEC or BASstat or Tstat8] —> **Receiver** [RDY] = 3 packets triggered 2 alerts due to the impermissible initiation of the *I-Am* service from the BASstat and Tstat8 devices

- **Sender** [TEC or BASstat or RDY] —> **Receiver** [Tstat8] = 3 packets triggered 2 alerts due to the impermissible initiation of the *I-Am* service from the BASstat and RDY devices

Interruption Attack

The third attack performed on our testbed was a DoS attack that attempts to leverage the *ReinitializeDevice* service to cause sporadic device restarts and interrupt BAS operation. We executed a custom script to inject the packet from the attacker machine and triggered 2 alerts.

- CControls BASstat - One alert was triggered for each device instance because the service is not supported
- JCI TEC - No alerts were triggered because the service is supported
- Siemens RDY - No alerts were triggered because the service is supported
- Temco Controls Tstat8 - No alerts were triggered because the service is supported

The fourth attack performed on our testbed was a flooding attack. The goal of the attacker with this attack would be to overwhelm the BA device's buffer with a stream of packets that are very large. We simulated this attack by injecting 500 byte packets filled with garbage addressed as if from the legitimate testbed device, but truly from the attacker machine. For this experiment we generated 6 packets from each device and triggered 6 alerts. Each alert came from the BIM model specification that states packets should be sized 300 bytes and less.

Modification Attack

The fifth and final attack performed on our testbed was a modification attack. The goal of the attacker in this scenario is to change the name of a *File* object stored locally on a

BA device. This file could be a restore file which contains default configuration details a device needs to recover from data corruption or any other BACnet Standard/Vendor Defined file. To carry out this attack, packets with the *WriteProperty* service to the *File* object's *ObjectName* property with a garbage string are injected from the attacker machine to each testbed device and 6 alerts were generated.

- CControls BASstat - One alert was generated from each of these devices due to the unsupported object type
- JCI TEC - No alerts were triggered because the object type and property are supported
- Siemens RDY - Two alerts were generated, one for unsupported object type and the other for the unsupported object property
- Temco Controls Tstat8 - Two alerts were generated, one for unsupported object type and the other for the unsupported object property

Some of these attacks highlight the main shortcoming of our approach and specification based intrusion detection techniques at large. We cannot detect attacks that leverage expected device behavior to perform a malicious payload. Although we are still successful in detecting many attacks the combination of multiple intrusion detection techniques will be useful as an intelligent attacker may be able to circumvent discovery.

5.5 Conclusion and Future Work

In this chapter we presented a semi-automated framework for gathering and extracting BACnet specifications and leveraging them into actionable intrusion detection rules. Our proposed technique is the first to investigate the use of building information models for specification based intrusion detection. We show how simple rule extraction mechanisms can be used to secure BASs and evaluate our approach on a testbed of real building automation devices. Our evaluation shows that BACnet PICS files can be valuable sources of

information for the generation of specification based rules to enforce BA device behavior, but their non-standardized formatting and erroneous nature often make them imperfect data sources. To supplement PICS files we developed BIM models infused with BACnet data. Our BIM models offered simpler feature extraction and higher data correctness. Additionally, with the supplementation of BIM, we implemented custom rules specific to the BAS deployment and rooted in operator domain knowledge. This expanded the intrusion detection rule space, allowing for more detailed rules than could be gleaned from PICS alone. Future work in this direction could aim to improve our approach and investigate intrusion detection in BASs in 3 main ways -

- Fortification of feature parsing methods to include optical character recognition (OCR) of scanned documents, programmatic mechanisms for exporting PDFs downloaded with password protection, and programmed parser resilience to minor typos for handling PICS grammatical/spelling errors

- Extension of the given intrusion detection approach to include anomaly based techniques for a more well-rounded intrusion detection system that not only checks that expected behavior matches the documentation, but also compares present behaviors with past behaviors to identify inconsistencies

- Inclusion of more data sources for specification extraction such as BACnet device EDE files, instruction manuals, and quick start guides

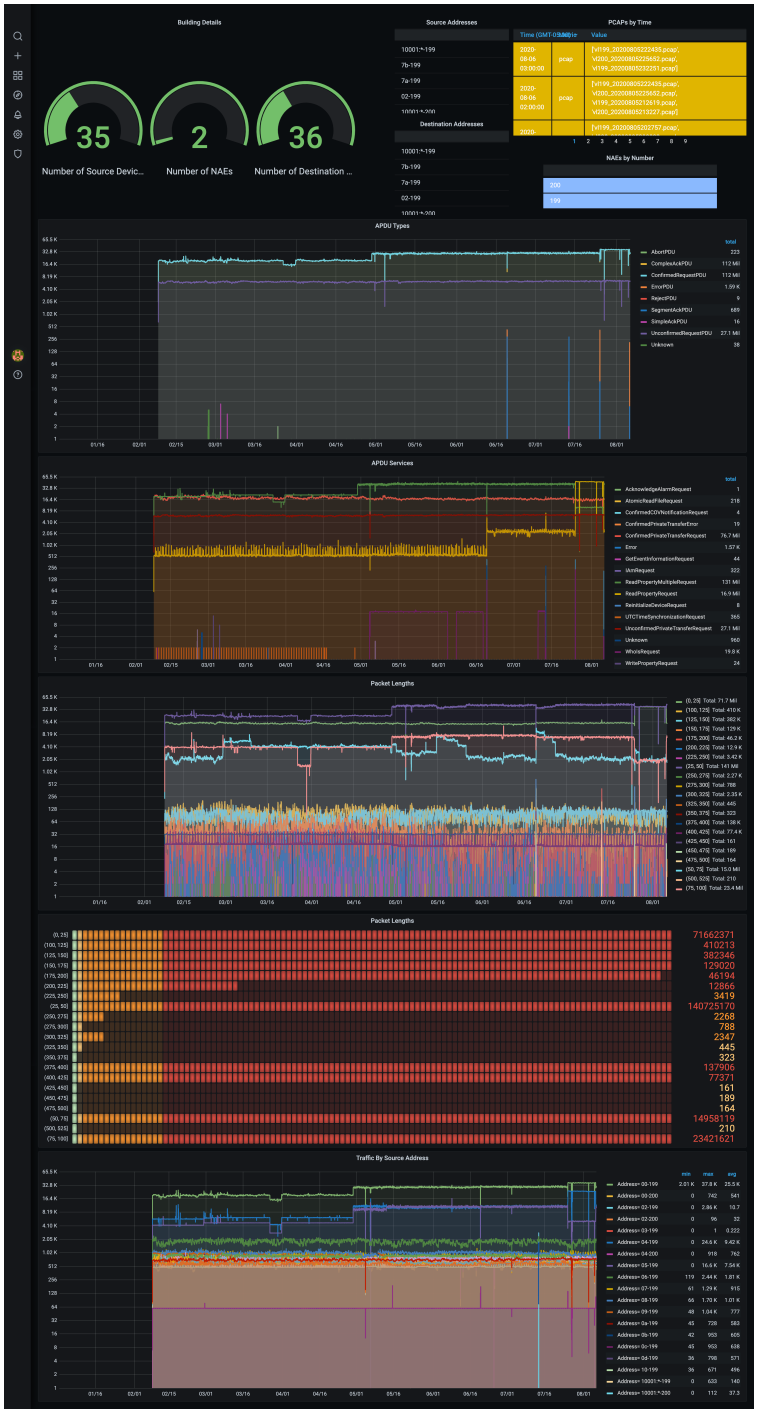
CHAPTER 6
SUMMARY OF CONCLUSIONS

For decades building automation systems have existed as siloed and isolated subsystems which operators interacted with through offline (low-tech) means. With the paradigm shift in the domain towards *smarter* buildings and the integration of the Internet of Things there are several open security challenges to be addressed. In this research we tackled some of these challenges by first, performing the largest field layer longitudinal BAN traffic characterization discussed in the research literature. Our study of a university campus building automation network uncovered that some buildings are responsive to environmental feedback (i.e. building use) and adjust accordingly, while others run a pre-programmed schedule with little deterrence. Then we performed a systematization of the BAS security literature and identified several research gaps, as well as proposed a BA device security framework. We evaluated our framework on the largest multi-protocol testbed discussed in the literature and discovered several device side-channel vulnerabilities. Lastly, we developed a semi-automated software framework for gathering BAS resources such as PICS files and BIM models to generate specification based intrusion detection rules. Overall, the methodologies proposed in this research serve as a solid basis for the BAS security research community to leverage as strides are continually taken towards the protection and safeguarding of smart building resources.

Appendices

APPENDIX A

REAL TIME WEB BAN DASHBOARD EXAMPLE



APPENDIX B

CAMPUS CHARACTERIZATION GRAPHS

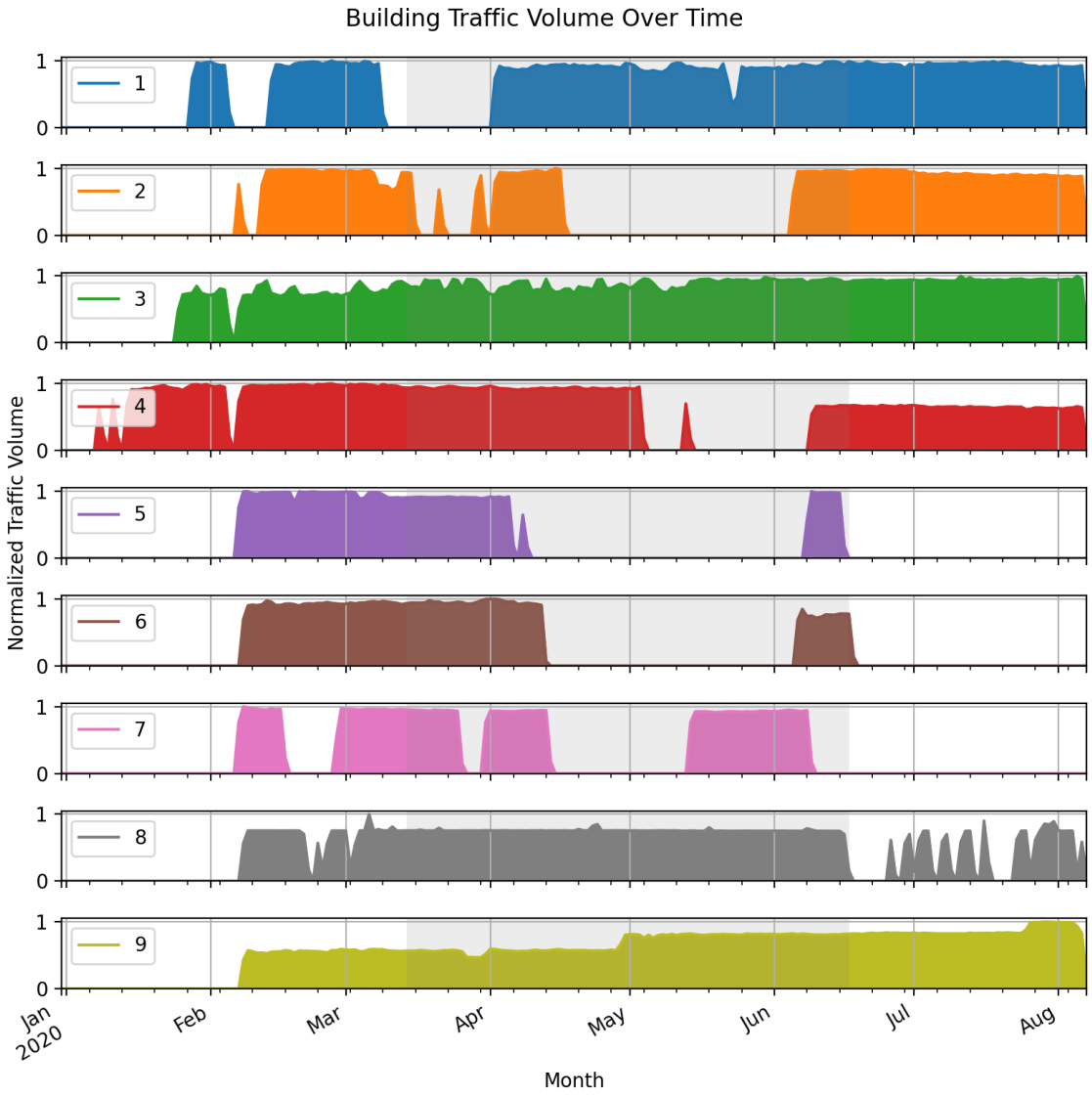


Figure B.1: Normalized Area Shaded Individual Building Traffic from January to August

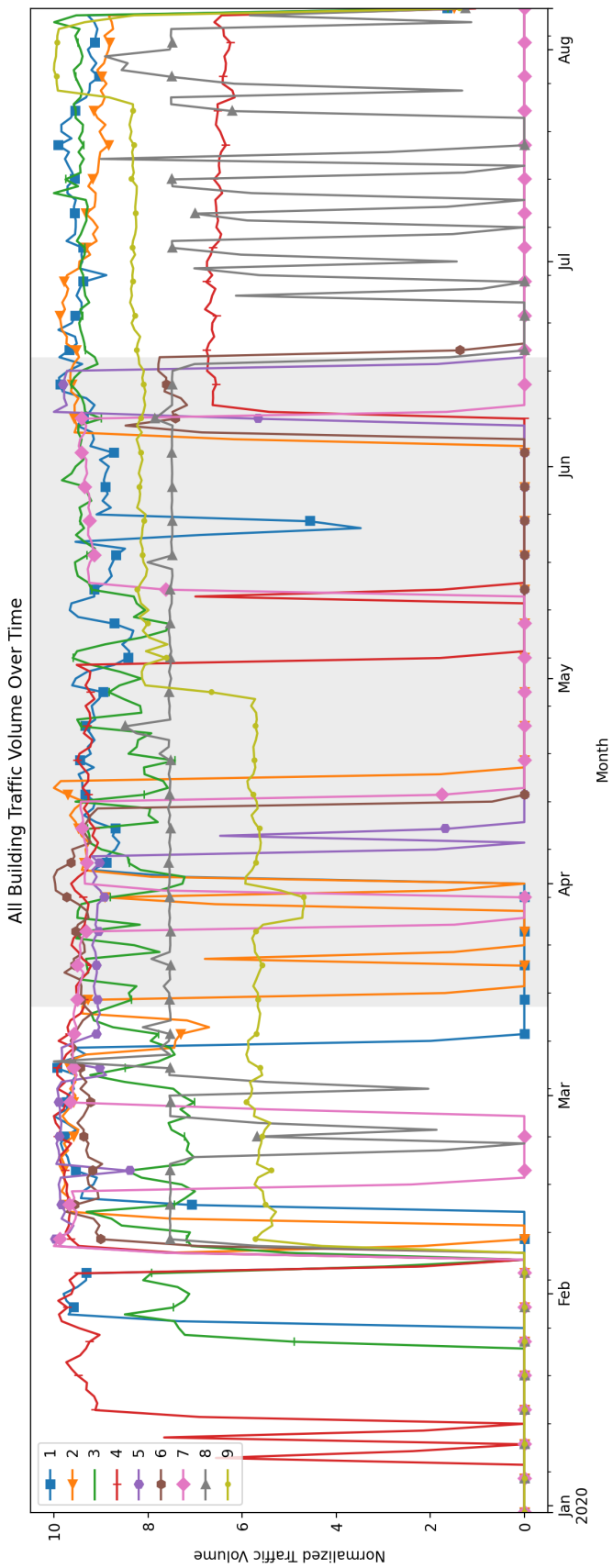


Figure B.2: All Building Normalized Traffic from January to August

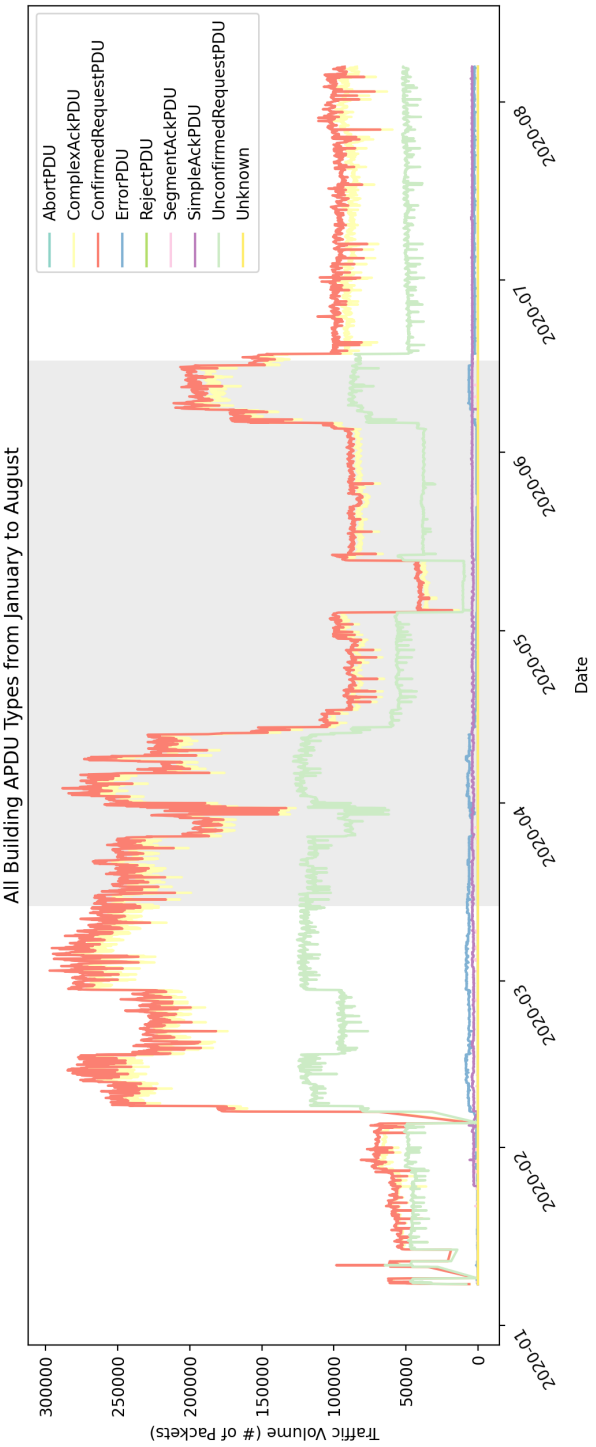


Figure B.3: All Building APDU Type Traffic from January to August

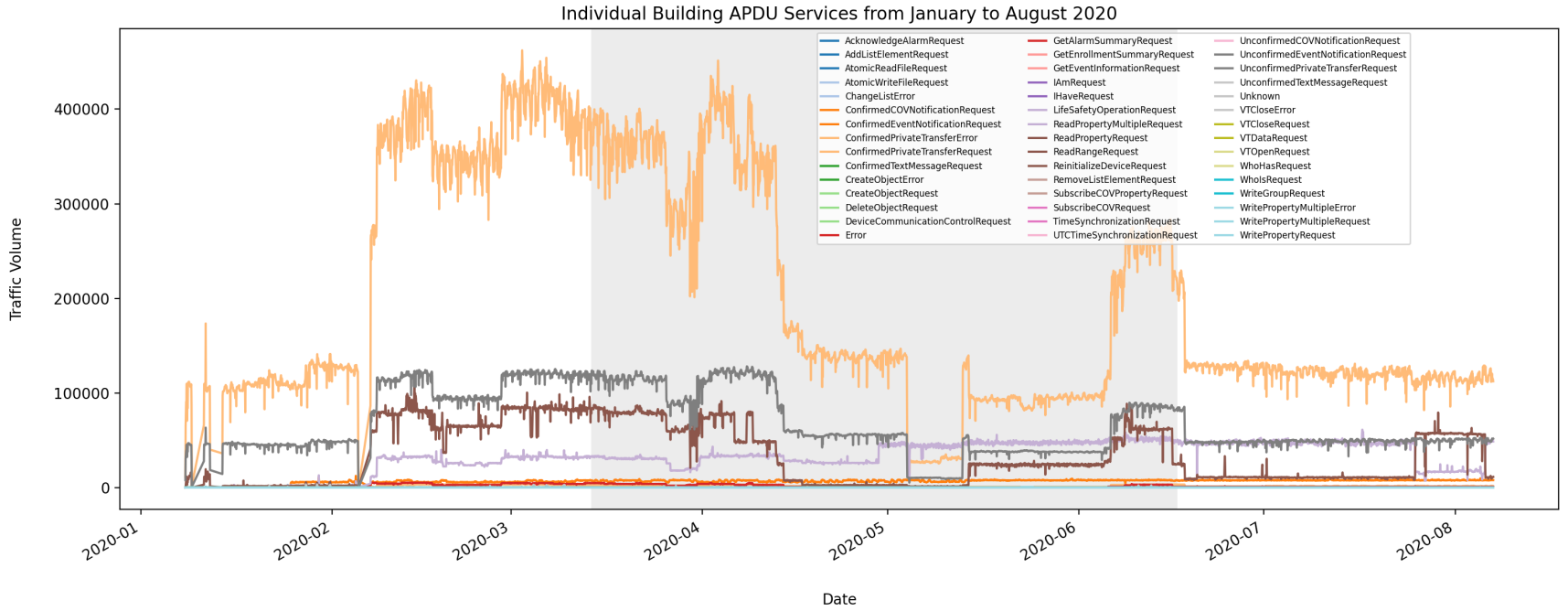


Figure B.4: All Building APDU Services Traffic from January to August

APPENDIX C

CAMPUS CHARACTERIZATION CASE STUDY FIGURES

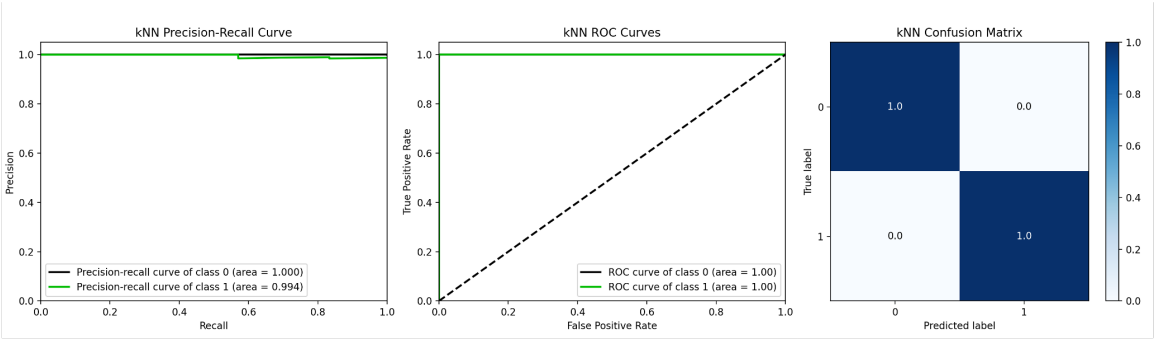


Figure C.1: kNN 200 Injected Anomalies

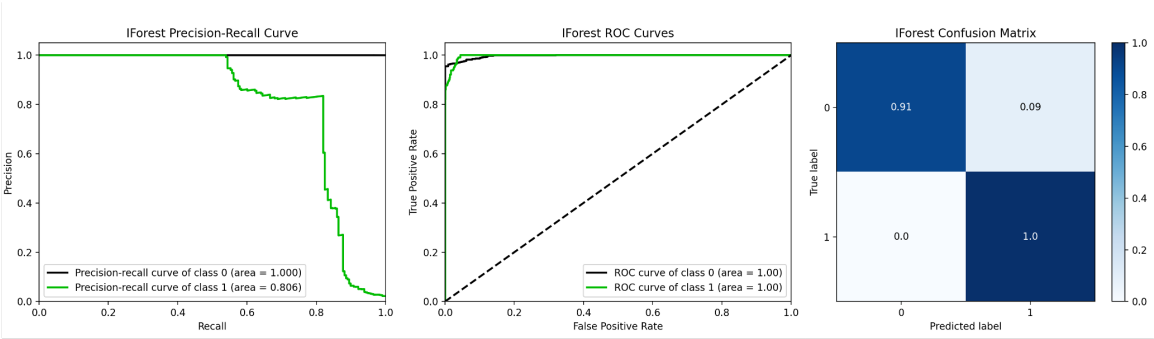


Figure C.2: Isolation Forest 200 Injected Anomalies

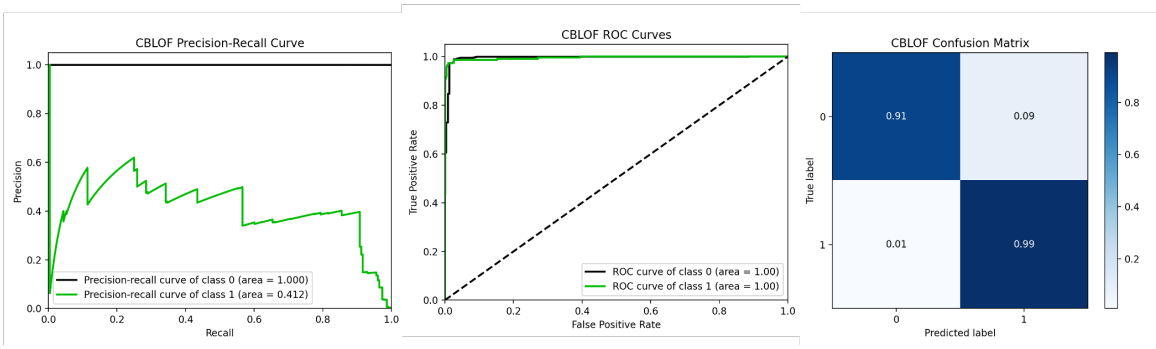


Figure C.3: CBLOF 200 Injected Anomalies

APPENDIX D

PHYSICAL TESTBED

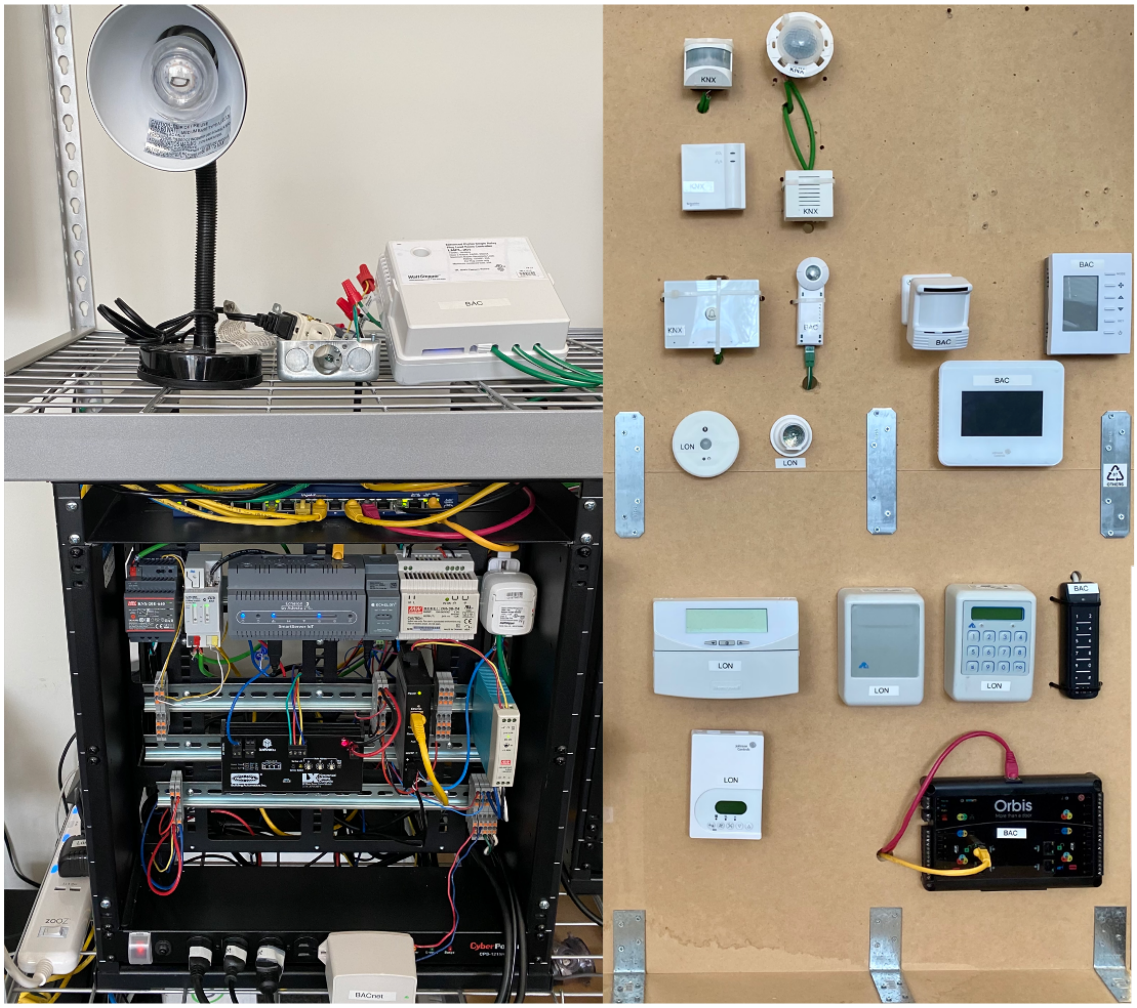


Figure D.1: Multi-protocol Building Automation System Testbed. Controllers (left) and networking equipment (left). Physical BA devices (right).

APPENDIX E

SIDE-CHANNEL EXPERIMENTAL DESIGN

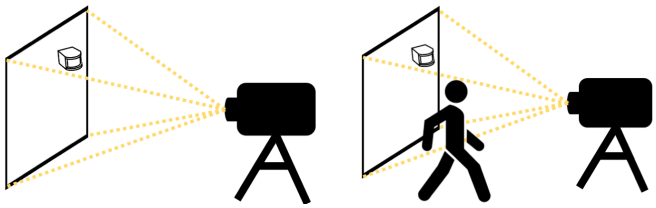


Figure E.1: For the optical side channel different levels of brightness applied (left). Reaction to user motion recorded (right).

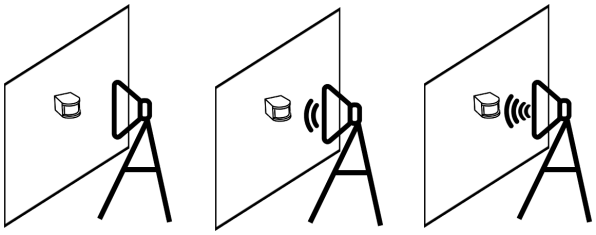


Figure E.2: For the acoustic side channel low (left) to high (right) frequency tones projected at devices.

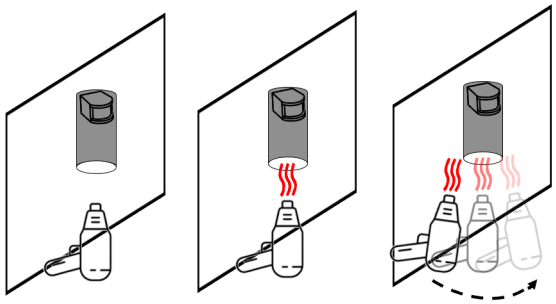


Figure E.3: For the thermal side channel, no heat applied (left), then static heat applied (center), and lastly heat applied in motion (right).

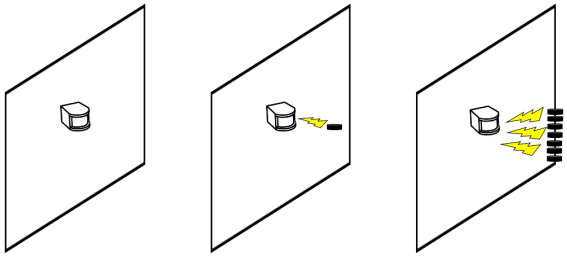


Figure E.4: For the magnetic side channel, the number of magnets presented to each device varied from none (left) to sixteen (right) to increase the strength of the magnetic field.

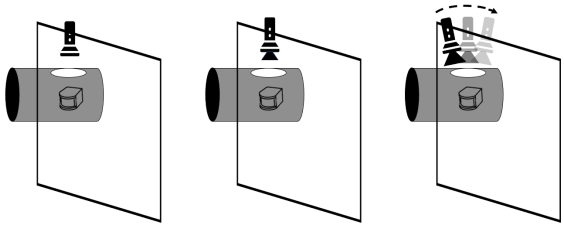


Figure E.5: For the IR side channel two types of IR flashlights were used to vary the IR radiation levels at each device from none (left), to a static maximum (center), to maximum in motion (right).

REFERENCES

[1] "Cyber security, building automation, and the intelligent building," *Intelligent Buildings International*, vol. 4, no. 3, pp. 169–181, 2012.

[2] D. Snoonian, "Smart buildings," *IEEE spectrum*, vol. 40, no. 8, pp. 18–23, 2003.

[3] W. Atkinson, *The future of building automation systems — electrical contractor magazine*, https://www.ecmag.com/section/systems/future-building-automation-systems, 2017.

[4] K. Zetter, *Researchers hack building control system at google australia office — wired*, https://www.wired.com/2013/05/googles-control-system-hacked/, 2013.

[5] Forbes. (2018). "Google's doors hacked wide open by own employee."

[6] KrebsOnSecurity. (2014). "Target hackers broke in via hvac company."

[7] W. Granzer and W. Kastner, "Security analysis of open building automation systems," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6351 LNCS, pp. 303–316, 2010.

[8] M. Peacock and M. N. Johnstone, "An analysis of security issues in building automation systems," *Australian Information Security Management Conference*, 2014.

[9] Alessio Antonini, Alessandro Barenghi and G. P. Dipartimento, *Security Analysis of Building Automation Networks: Threat Model and Viable Mitigation Techniques*, October. 2013, ISBN: 9783642414879.

[10] S. Cavalieri and G. Cutuli, "Implementing encryption and authentication in knx using diffie-hellman and aes algorithms," in *2009 35th Annual Conference of IEEE Industrial Electronics*, IEEE, 2009, pp. 2459–2464.

[11] M. Cheminod, L. Durante, and A. Valenzano, "Review of security issues in industrial networks," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 277–293, 2012.

[12] D. J. Brooks, M. Coole, P. Haskell-Dowland, M. Griffiths, and N. Lockhart, "Building Automation & Control Systems: An Investigation into Vulnerabilities, Current Practice & Security Management Best Practice," p. 210, 2017.

[13] K. Khaund, "Cybersecurity in smart buildings inaction is not an option anymore," no. September, 2015.

[14] D. G. Holmberg, J. J. Bender, and M. A. Galler, "Using the bacnet (r) firewall router," *ASHRAE American Society for Heating, Refrigeration and Air Conditioning Journal*, vol. 48, no. ASHRAE American Society for Heating, Refrigeration and Air Conditioning Journal, 2006.

[15] Z. Pan, S. Hariri, and Y. Al-Nashif, "Anomaly based intrusion detection for building automation and control networks," in *IEEE/ACS Int. Conf. on Computer Systems and Applications (AICCSA)*, 2014, pp. 72–77.

[16] J. Kaur, J. Tonejc, S. Wendzel, and M. Meier, "Securing BACnet's pitfalls," *From book ICT Systems Security and Privacy Protection*, vol. 428, no. June, 2014.

[17] A.-S. M. Q. Abdulmunem and V. S. Kharchenko, "Availability and security assessment of smart building automation systems: Combining of attack tree analysis and markov models," in *2016 Third International Conference on Mathematics and Computers in Sciences and in Industry (MCSI)*, IEEE, 2016, pp. 302–307.

[18] E. J. Byres, M. Franz, and D. Miller, "The use of attack trees in assessing vulnerabilities in scada systems," in *Proceedings of the international infrastructure survivability workshop*, Citeseer, 2004, pp. 3–10.

[19] P. A. Khand, "System level security modeling using attack trees," in *2009 2nd International Conference on Computer, Control and Communication*, IEEE, 2009, pp. 1–6.

[20] V. Paxson, "End-to-end internet packet dynamics," in *ACM SIGCOMM Computer Communication Review*, ACM, vol. 27, 1997, pp. 139–152.

[21] R. R. R. Barbosa, R. Sadre, and A. Pras, "A first look into scada network traffic," in *Network Operations and Management Symposium (NOMS), 2012 IEEE*, IEEE, 2012, pp. 518–521.

[22] S. S. Jung, D. Formby, C. Day, and R. Beyah, "A first look at machine-to-machine power grid network traffic," in *Smart Grid Communications (SmartGridComm), 2014 IEEE International Conference on*, IEEE, 2014, pp. 884–889.

[23] D. Formby, A. Walid, and R. Beyah, "A case study in power substation network dynamics," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 1, no. 1, p. 19, 2017.

[24] M. Peacock, "Anomaly Detection in BACnet / IP managed Building Automation Systems Edith Cowan University," 2019.

[25] R. Krejčı, P. Čeleda, and J. Dobrovoln, "Traffic measurement and analysis of building automation and control networks," in *IFIP International Conference on Autonomous Infrastructure, Management and Security*, Springer, 2012, pp. 62–73.

[26] Z. Zheng and A. N. Reddy, "Safeguarding building automation networks: Thedriven anomaly detector based on traffic analysis," in *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, IEEE, 2017, pp. 1–11.

[27] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 16–24, 2013.

[28] P. Čeleda, R. Krejčı, and V. Krmıček, "Flow-based security issue detection in building automation and control networks," in *Meeting of the European Network of Universities and Companies in Information and Communication Engineering*, Springer, 2012, pp. 64–75.

[29] Z. Pan, J. Pacheco, and S. Hariri, "Anomaly behavior analysis for building automation systems," in *IEEE/ACS Int. Conf. of Computer Systems and Applications (AICCSA)*, 2016, pp. 1–8.

[30] Z. Pan, S. Hariri, and J. Pacheco, "Context aware intrusion detection for building automation systems," *Computers & Security*, vol. 85, pp. 181–201, 2019.

[31] M. N. Johnstone, M. Peacock, and J. den Hartog, "Timing attack detection on bacnet via a machine learning approach," 2015.

[32] J. Tonejc, S. Güttes, A. Kobekova, and J. Kaur, "Machine learning methods for anomaly detection in BACnet networks," *Journal of Universal Computer Science*, vol. 22, no. 9, pp. 1203–1224, 2016.

[33] M. Caselli, E. Zambon, J. Amann, R. Sommer, F. Kargl, E. Zambon, J. Amann, and F. Kargl, "Specification Mining for Intrusion Detection in Networked Control Systems Specification Mining for Intrusion Detection in Networked Control Systems," *Proceedings of the 25th USENIX Security Symposium*, pp. 791–806, 2016.

[34] H. Esquivel-Vargas, M. Caselli, and A. Peter, "Automatic Deployment of Specification-based Intrusion Detection in the BACnet Protocol," pp. 25–36, 2017.

[35] M. Kapsalakis, "Passive situational awareness and threat detection in building automation networks," *Thesis*, Oct. 2017.

[36] D. Fauri, M. Kapsalakis, D. R. dos Santos, E. Costante, J. den Hartog, and S. Etalle, "Leveraging semantics for actionable intrusion detection in building automation

systems," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11260 LNCS, no. 700665, pp. 113–125, 2019.

[37] G. K. Ndonda and R. Sadre, "A public network trace of a control and automation system," *arXiv preprint arXiv:1908.02118*, 2019.

[38] J. Achenbach, A. Eunjung Cha, and F. Stead Sellers, "A viral tsunami: How the underestimated coronavirus took over the world," *The Washington Post*, Mar. 2021.

[39] W. Kastner, G. Neugschwandtner, S. Soucek, and H. M. Newman, "Communication systems for building automation and control," *Proc. of the IEEE*, vol. 93, no. 6, pp. 1178–1203, 2005.

[40] Cimetrics. (2020). "What is BACnet?"

[41] A. Martin. (2017). "BACnet: A Technical Summary."

[42] A. S. 135. (2021). "BACnet - A Data Communication Protocol for Building Automation and Control Networks."

[43] Amazon. (2017). "Amazon AWS."

[44] S. K. Alghoul, "A comparative study of energy consumption for residential hvac systems using energyplus," *American Journal of Mechanical and Industrial Engineering*, vol. 2, no. 2, pp. 98–103, 2017.

[45] WEBfactory. (2019). "BACnet Status Codes."

[46] Chipkin. (2021). "BACNET - WHAT IS THE BACNET CHANGE OF VALUE (COV)."

[47] Y. Zhao, Z. Nasrullah, and Z. Li, "Pyod: A python toolbox for scalable outlier detection," *Journal of Machine Learning Research*, vol. 20, no. 96, pp. 1–7, 2019.

[48] A. Mucherino, P. J. Papajorgji, and P. M. Pardalos, "K-nearest neighbor classification," in *Data Mining in Agriculture*. New York, NY: Springer New York, 2009, pp. 83–106, ISBN: 978-0-387-88615-2.

[49] F. T. Liu, K. M. Ting, and Z. Zhou, "Isolation forest," in *2008 Eighth IEEE International Conference on Data Mining*, 2008, pp. 413–422.

[50] Z. He, X. Xu, and S. Deng, "Discovering cluster-based local outliers," *Pattern Recogn. Lett.*, vol. 24, no. 9–10, pp. 1641–1650, Jun. 2003.

[51] S. Wendzel, J. Tonejc, J. Kaur, A. Kobekova, H. Song, G. Fink, and S. Jeschke, *Cyber security of smart buildings*. Wiley, 2017.

[52] W. Granzer, F. Praus, and W. Kastner, "Security in building automation systems," *IEEE Trans. Ind. Electron.*, vol. 57, no. 11, pp. 3622–3630, 2010.

[53] Market and M. R. Private. (2019). "Building automation system market by communication technology, offering (facilities management systems, security & access control systems, fire protection systems, and building energy management software), application, region - global forecast to 2024."

[54] J. Bourne. (2017). "IoT for intelligent buildings will surpass $22 billion by 2026, research says."

[55] B. Mühlberg. (2020). "Hackers use smart building access control systems to launch DDoS attacks."

[56] M. Kassner. (2015). "Anatomy of the target data breach: Missed opportunities and lessons learned."

[57] Y. Liu, Z. Pang, G. Dán, D. Lan, and S. Gong, "A taxonomy for the security assessment of ip-based building automation systems: The case of thread," *IEEE Trans. Ind. Inform.*, vol. 14, no. 9, pp. 4113–4123, 2018.

[58] B. Inc., "World penetration of communications protocols 2018," BSRIA, Tech. Rep., 2018.

[59] INCIBE. (2017). "Security in protocols for building automation."

[60] S. Electric, "Guide to Open Protocols In Building Automation," *Scheinder Electric*, 2015.

[61] Frecks, *Knx model*, 2018.

[62] Control solutions Minnesota, *Lonworks 101 - introduction to LonWorks*, 2020.

[63] T. Brandstetter and K. Reisinger, "(in)security in building automation - how to create dark buildings with light speed," *Black Hat*, pp. 1–15, 2017.

[64] A. Antonini, F. Maggi, and S. Zanero, "A practical attack against a KNX-based building automation system," in *Int. Symp. for ICS & SCADA Cyber Security Research (ICS-CSR)*, 2014, pp. 53–60.

[65] S. Wendzel, T. Rist, E. André, and M. Masoodian, "A secure interoperable architecture for building-automation applications," in *Proc. of the Int. Symp. on Applied Sciences in Biomedical and Communication Technologies*, 2011, pp. 1–5.

[66] Q. A. A.-S. Mustafa, A.-K. A. Waleed, and V. Kharchenko, "Ata-based security assessment of smart building automation systems," *Radioelectronic and Computer Systems*, no. 3, pp. 30–40, 2016.

[67] H. Esquivel-Vargas, M. Caselli, and A. Peter, "Automatic deployment of specification-based intrusion detection in the BACnet protocol," in *Proc. of the 2017 Workshop on Cyber-Physical Systems Security and Privacy*, 2017, pp. 25–36.

[68] A.-S. M. Q. Abdulmunem and V. S. Kharchenko, "Availability and security assessment of smart building automation systems: Combining of attack tree analysis and Markov models," in *3rd Int. Conf. Mathematics and Computers in Sciences and in Industry (MCSI)*, 2016, pp. 302–307.

[69] D. G. Holmberg and D. Evans, *BACnet wide area network security threat assessment*. US Department of Commerce, National Institute of Standards and Technology, 2003.

[70] H. Esquivel-Vargas, M. Caselli, E. Tews, D. Bucur, and A. Peter, "BACRank: Ranking building automation and control system components by business continuity impact," in. Springer, Cham, Aug. 2019, pp. 183–199.

[71] Foundation, IoT Security, "Can you trust your smart building?" *White Paper*, no. June, 2019.

[72] T. Novak, A. Treytl, and P. Palensky, "Common approach to functional safety and system security in building automation and control systems," in *IEEE Conf. Emerging Technologies and Factory Automation*, 2007, pp. 1141–1148.

[73] W. Granzer and W. Kastner, "Communication services for secure building automation networks," in *IEEE Int. Symp. on Ind. Electron.*, 2010, pp. 3380–3385.

[74] Z. Zheng and A. N. Reddy, "Safeguarding building automation networks: Thedriven anomaly detector based on traffic analysis," in *26th Int. Conf. Computer Communication and Networks (ICCCN)*, 2017, pp. 1–11.

[75] D. Fisk, "Cyber security, building automation, and the intelligent building," *Intelligent Buildings International*, vol. 4, no. 3, pp. 169–181, 2012.

[76] K. Paridari, A. E.-D. Mady, S. La Porta, R. Chabukswar, J. Blanco, A. Teixeira, H. Sandberg, and M. Boubekeur, "Cyber-physical-security framework for build-

ing energy management system," in *ACM/IEEE 7th Int. Conf. on Cyber-Physical Systems (ICCPS)*, 2016, pp. 1–9.

[77] S. Wendzel, "How to increase the security of smart buildings?" *Communications of the ACM*, vol. 59, no. 5, pp. 47–49, 2016.

[78] F. Praus and W. Kastner, "Identifying unsecured building automation installations," *IEEE Int. Conf. Emerging Technologies and Factory Automation*, pp. 1–4, 2014.

[79] J. P. Boyer, R. Hasan, L. E. Olson, N. Borisov, C. A. Gunter, and D. Raila, "Improving multi-tier security using redundant authentication," in *Proc. of the ACM Workshop on Computer Security Architecture*, 2007, pp. 54–62.

[80] R. Habeeb, "Improving the security of building automation systems through an sel4-based communication framework," *MSc Thesis, University of South Florida*, 2018.

[81] D. R. dos Santos, M. Dagrada, and E. Costante, "Leveraging operational technology and the Internet of things to attack smart buildings," *Journal of Computer Virology and Hacking Techniques*, pp. 1–20, 2020.

[82] D. Fauri, M. Kapsalakis, D. R. dos Santos, E. Costante, J. den Hartog, and S. Etalle, "Leveraging semantics for actionable intrusion detection in building automation systems," in *Int. Conf. Critical Information Infrastructures Security*, 2018, pp. 113–125.

[83] J. Tonejc, S. Güttes, A. Kobekova, and J. Kaur, "Machine learning methods for anomaly detection in BACnet networks," *Journal of Universal Computer Science*, vol. 22, no. 9, pp. 1203–1224, 2016.

[84] J. E. Hachem, V. Chiprianov, M. A. Babar, T. A. Khalil, and P. Aniorte, "Modeling, analyzing and predicting security cascading attacks in smart buildings systems-of-systems," *Journal of Systems and Software*, vol. 162, pp. 1–17, 2020.

[85] A. Judmayer, L. Krammer, and W. Kastner, "On the security of security extensions for IP-based KNX networks," in *10th IEEE Workshop on Factory Communication Systems (WFCS)*, 2014, pp. 1–10.

[86] J. L. Hernández-Ramos, M. V. Moreno, J. B. Bernabé, D. G. Carrillo, and A. F. Skarmeta, "SAFIR: Secure access framework for IoT-enabled services on smart buildings," *Journal of Computer and System Sciences*, vol. 81, no. 8, pp. 1452–1463, 2015.

[87] N. Friman, *Security analysis of smart buildings*, 2020.

[88] T. Tenkanen and T. Hämäläinen, "Security assessment of a distributed, Modbus-based building automation system," in *IEEE Int. Conf. Computer and Information Technology (CIT)*, 2017, pp. 332–337.

[89] A. Antonini, A. Barenghi, G. Pelosi, and S. Zonouz, "Security challenges in building automation and SCADA," in *Int. Carnahan Conf. on Security Technology (ICCST)*, 2014, pp. 1–6.

[90] S. Krishnan, M. Anjana, and S. N. Rao, "Security considerations for IoT in smart buildings," in *IEEE Int. Conf. Computational Intelligence and Computing Research (ICCIC)*, 2017, pp. 1–4.

[91] O. Gasser, Q. Scheitle, C. Denis, N. Schricker, and G. Carle, "Security implications of publicly reachable building automation systems," in *IEEE Security and Privacy Workshops (SPW)*, 2017, pp. 199–204.

[92] M. Caselli, *Intrusion detection in networked control systems : from system knowledge to network security*. Universiteit Twente, 2016, pp. 791–806, ISBN: 9789036541770.

[93] W. Bo, Y. Zhang, X. Hong, H. Sun, and X. Huang, "Usable security mechanisms in smart building," in *IEEE Int. Conf. on Computational Science and Engineering*, 2014, pp. 748–753.

[94] J. Tonejc, J. Kaur, A. Karsten, and S. Wendzel, "Visualizing BACnet data to facilitate humans in building-security decision-making," in *Int. Conf. Human Aspects of Information Security, Privacy, and Trust*, 2015, pp. 693–704.

[95] A. Caranica, A. Vulpe, and O. Fratu, "Tenable smart building security flow architecture using open source tools," in *World Conf. on Information Systems and Technologies*, 2018, pp. 118–127.

[96] V. Kharchenko, Y. Ponochovnyi, A.-S. M. Q. Abdulmunem, and A. Andrashov, "Availability models and maintenance strategies for smart building automation systems considering attacks on component vulnerabilities," in *Advances in Dependability Engineering of Complex Systems*, Springer, 2017, pp. 186–195.

[97] J. Bauer, J. Goltz, T. Mundt, and S. Wiedenmann, "Honeypots for threat intelligence in building automation systems," in *Computing, Communications and IoT Applications (ComComAp)*, 2019, pp. 242–246.

[98] S. Pérez, J. L. Hernández-Ramos, S. N. Matheu-Garcıa, D. Rotondi, A. F. Skarmeta, L. Straniero, and D. Pedone, "A lightweight and flexible encryption scheme to protect sensitive data in smart building scenarios," *IEEE Access*, vol. 6, pp. 11 738–11 750, 2018.

[99] R. C. Luo, S. Y. Lin, and K. L. Su, "A multiagent multisensor based security system for intelligent building," in *Proc. of IEEE Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems*, 2003, pp. 311–316.

[100] M. Peacock, "Anomaly detection in BACnet / IP managed building automation systems," Ph.D. dissertation, Edith Cowan University, 2019.

[101] D. dos Santos, C. Speybrouck, and E. Costante, "Cybersecurity in Building Automation Systems (BAS)," Forescout Technologies, Tech. Rep., 2019.

[102] X. Wang, R. Habeeb, X. Ou, S. Amaravadi, J. Hatcliff, M. Mizuno, M. Neilsen, S. R. Rajagopalan, and S. Varadarajan, "Enhanced security of building automation systems through microkernel-based controller platforms," in *IEEE Int. Conf. on Distributed Computing Systems Workshops (ICDCSW)*, 2017, pp. 37–44.

[103] D. Fauri, D. R. Dos Santos, E. Costante, J. den Hartog, S. Etalle, and S. Tonetta, "From system specification to anomaly detection (and back)," in *Proc. of the Workshop on Cyber-Physical Systems Security and Privacy*, 2017, pp. 13–24.

[104] S. Cavalieri and G. Cutuli, "Implementing encryption and authentication in KNX using Diffie-Hellman and AES algorithms," in *35th Ann. Conf. of IEEE Ind. Electron.*, 2009, pp. 2459–2464.

[105] J. M. Porteous, "Intelligent buildings and their effect on the security industry," in *Proc. Int. Carnahan Conf. on Security Technology*, 1995, pp. 186–188.

[106] J. Molina, "Learn how to control every room at a luxury hotel remotely: The dangers of insecure home automation deployment," *Black Hat USA*, 2014.

[107] D. Fauri, M. Kapsalakis, D. R. dos Santos, E. Costante, J. den Hartog, and S. Etalle, "Role inference+ anomaly detection= situational awareness in BACnet networks," in *Int. Conf. on Detection of Intrusions and Malware, and Vulnerability Assessment*, 2019, pp. 461–481.

[108] N. Xue, X. Huang, and J. Zhang, "S2net: A security framework for software defined intelligent building networks," in *IEEE Trustcom/BigDataSE/ISPA*, 2016, pp. 654–661.

[109] W. Granzer and W. Kastner, "Security analysis of open building automation systems," in *Int. Conf. on Computer Safety, Reliability, and Security*, 2010, pp. 303–316.

[110] M. Caselli, E. Zambon, and F. Kargl, "Sequence-aware intrusion detection in industrial control systems," in *Proc. of the ACM Workshop on Cyber-Physical System Security*, 2015, pp. 13–24.

[111] S. Szlósarczyk, S. Wendzel, J. Kaur, M. Meier, and F. Schubert, "Towards suppressing attacks on and improving resilience of building automation systems - an approach exemplified using BACnet," *Sicherheit 2014–Sicherheit, Schutz und Zuverlässigkeit*, pp. 407–418, 2014.

[112] J. de las Morenas, C. M. da Silva, G. S. Funchal, V. Melo, M. Vallim, and P. Leitao, "Security experiences in iot based applications for building and factory automation," in *IEEE Int. Conf. on Industrial Technology (ICIT)*, 2020, pp. 322–327.

[113] L. P. Rondon, L. Babun, A. Aris, K. Akkaya, and A. S. Uluagac, "PoisonIvy: (In)secure practices of enterprise IoT systems in smart buildings," *arXiv preprint arXiv:2010.05658*, 2020.

[114] R. Halemani and A. Rajagopal, "Building automation security using can and IoT," in *Int. Conf. on Applied and Theoretical Computing and Communication Technology (ICATCCT)*, 2015, pp. 471–476.

[115] C. M. Calimbahin, S. Pancho-Festin, and J. R. Pedrasa, "Domain-based attack models in building automation systems," in *IEEE Int. Conf. on Emerging Technologies and Factory Automation (ETFA)*, 2019, pp. 1752–1758.

[116] S. Wendzel, V. Zwanger, M. Meier, and S. Szlósarczyk, "Envisioning smart building botnets," *Sicherheit 2014–Sicherheit, Schutz und Zuverlässigkeit*, pp. 1–11, 2014.

[117] T. Mundt, A. Dähn, and H.-W. Glock, "Forensic analysis of home automation systems," in *Workshop on Hot Topics in Privacy Enhancing Technologies (HotPETs)*, 2014, pp. 1–18.

[118] S. Charalambous, "Incorporating smart building security with BIM," Ph.D. dissertation, Imperial College London, 2017.

[119] H. Glanzer, L. Krammer, and W. Kastner, "Increasing security and availability in KNX networks," *Sicherheit 2016-Sicherheit, Schutz und Zuverlässigkeit*, pp. 1–12, 2016.

[120] T. Novak and A. Gerstinger, "Safety and security-critical services in building automation and control systems," *IEEE Trans. Ind. Electron.*, vol. 57, no. 11, pp. 3614–3621, 2009.

[121] F. Praus and W. Kastner, "Secure control applications in building automation using domain knowledge," in *IEEE Int. Conf. on Ind. Inform.*, 2010, pp. 52–57.

[122] F. Praus, "Secure control applications in smart homes and buildings," *PhD Dissertation, TU Wien*, pp. 1–212, 2015.

[123] X. Wang, M. Mizuno, M. Neilsen, X. Ou, S. R. Rajagopalan, W. G. Boldwin, and B. Phillips, "Secure rtos architecture for building automation," in *Proc. of the First ACM Workshop on Cyber-Physical Systems-Security and/or Privacy*, 2015, pp. 79–90.

[124] C. Reinisch, W. Granzer, and W. Kastner, "Secure vertical integration for building automation networks," in *IEEE Int. Workshop on Factory Communication Systems*, 2008, pp. 239–242.

[125] A. Antonini, A. Barenghi, and G. Pelosi, "Security analysis of building automation networks," in *Nordic Conf. on Secure IT Systems*, 2013, pp. 199–214.

[126] T. Fischer, C. Lesjak, A. Hoeller, and C. Steger, "Security for building automation with hardware-based node authentication," in *IEEE Int. Conf. on Emerging Technologies and Factory Automation (ETFA)*, 2017, pp. 1–6.

[127] T. Mundt and P. Wickboldt, "Security in building automation systems - a first analysis," in *Int. Conf. On Cyber Security And Protection Of Digital Services (Cyber Security)*, 2016, pp. 1–8.

[128] W. Granzer, W. Kastner, G. Neugschwandtner, and F. Praus, "Security in networked building automation systems," in *IEEE Int. Workshop on Factory Communication Systems*, 2006, pp. 283–292.

[129] C. Schwaiger and A. Treytl, "Smart card based security for fieldbus systems," in *IEEE Conf. on Emerging Technologies and Factory Automation*, vol. 1, 2003, pp. 398–406.

[130] F. Praus, W. Kastner, and P. Palensky, "Software security requirements in building automation," *Sicherheit 2016-Sicherheit, Schutz und Zuverlässigkeit*, 2016.

[131] M. Q. A. Al-Sudani, W. A.-K. Ahmed, and V. Kharchenko, "The method of IMECA-based security assessment: Case study for building automation system,", no. 1, pp. 138–144, 2016.

[132] C. Mays, M. Rice, B. Ramsey, J. Pecarina, and B. Mullins, "Defending building automation systems using decoy networks," in *Int. Conf. on Critical Infrastructure Protection*, 2017, pp. 297–317.

[133] J. Qi, Y. Kim, C. Chen, X. Lu, and J. Wang, "Demand response and smart buildings: A survey of control, communication, and cyber-physical security," *ACM Trans. on Cyber-Physical Syst.*, vol. 1, no. 4, pp. 1–25, 2017.

[134] T. Wei, B. Zheng, Q. Zhu, and S. Hu, "Security analysis of proactive participation of smart buildings in smart grid," in *IEEE/ACM Int. Conf. on Computer-Aided Design (ICCAD)*, 2015, pp. 465–472.

[135] T. Mundt, S. Wiedenmann, J. Goltz, J. Bauer, and M. Jung, "Detecting intrusive behaviour of people in a building through data analysis and anomaly detection in home automation systems," in *Int. Conf. on New Technologies, Mobility and Security (NTMS)*, 2019, pp. 1–7.

[136] C. M. Calimbahin, S. Pancho-Festin, and J. R. Pedrasa, "Mitigating data integrity attacks in building automation systems using denoising autoencoders," in *Int. Conf. on Ubiquitous and Future Networks (ICUFN)*, 2019, pp. 390–395.

[137] M. Caselli, E. Zambon, J. Amann, R. Sommer, F. Kargl, E. Zambon, J. Amann, and F. Kargl, "Specification mining for intrusion detection in networked control systems specification mining for intrusion detection in networked control systems," *Proc. of the 25th USENIX Security Symposium*, pp. 791–806, 2016.

[138] S. Wendzel, B. Kahler, and T. Rist, "Covert channels and their prevention in building automation protocols: A prototype exemplified using BACnet," in *IEEE Int. Conf. on Green Computing and Communications*, 2012, pp. 731–736.

[139] -. (2021). "Remsdaq."

[140] J.-b. Hou, T. Li, and C. Chang, "Research for vulnerability detection of embedded system firmware," *Procedia Computer Science*, vol. 107, pp. 814–818, 2017.

[141] -. (2021). "CISA ICS-CERT Advisory."

[142] ——, (2021). "CVE Details."

[143] P. Mell, K. Scarfone, and S. Romanosky, "Common vulnerability scoring system," *IEEE Security Privacy*, vol. 4, no. 6, pp. 85–89, 2006.

[144] M. Devi and A. Majumder, "Side-channel attack in Internet of things: A survey," in *Applications of Internet of Things*, 2021, pp. 213–222.

[145] S. Mangard, E. Oswald, and T. Popp, *Power analysis attacks: revealing the secrets of smart cards*. New York: Springer US, 2007.

[146] T. Sugawara, B. Cyr, S. Rampazzi, D. Genkin, and K. Fu, "Light commands: Laser-based audio injection attacks on voice-controllable systems," in *USENIX Security Symp.*, 2020, pp. 2631–2648.

[147] K. Zetter. (2014). “Here’s How Easy It Could Be for Hackers to Control Your Hotel Room.”

[148] B. Ali and A. I. Awad, “Cyber and physical security vulnerability assessment for IoT-based smart homes,” *Sensors*, vol. 18, no. 3, p. 817, 2018.

[149] T. I. S. Committee. (2015). “Best Practices for Planning and Managing Physical Security Resources: An Interagency Security Committee Guide.”

[150] J. Zhang, “Distributed network security framework of energy internet based on Internet of things,” *Sustainable Energy Technologies and Assessments*, vol. 44, pp. 1–10, 2021.

[151] E. D. Knapp and J. T. Langill, *Industrial Network Security: Securing critical infrastructure networks for smart grid, SCADA, and other Industrial Control Systems*. Syngress, 2014.

[152] A. Kabulov, I. Yarashov, and D. Vasiyeva, “Security threats and challenges in IoT technologies,” *Science and Education*, vol. 2, no. 1, pp. 1–9, 2021.

[153] BACnet International, *BACnet Secure Connection*, 2020.

[154] C. Ebert, “Risk-based security engineering,” *Project Risk Management: Managing Software Development Risk*, 2021.

[155] S. Kesavan, J. Senthilkumar, Y. Suresh, and V. Mohanraj, “Iot device onboarding, monitoring, and management: Approaches, challenges, and future,” in *Challenges and Opportunities for the Convergence of IoT, Big Data, and Cloud Computing*, IGI Global, 2021, pp. 227–255.

[156] M. Conti, D. Donadel, and F. Turrin, “A survey on industrial control system testbeds and datasets for security research,” *arXiv preprint arXiv:2102.05631*, pp. 1–46, 2021.

[157] R. Rastogi, R. Jain, and P. Jain, “Iot applications in smart home security: Addressing safety and security threats,” in *Artificial Intelligence Paradigms for Smart Cyber-Physical Systems*, IGI Global, 2021, pp. 251–277.

[158] A. Richardson. (2021). “Building Automation Systems: What You Need To Know.”

[159] Morten Kvistgaard, Frédéric Chaxel, Adam Guzik, Christopher Günther, Thamer Al-Salek, *Yet another bacnet explorer*, version 1, Feb. 19, 2021.

[160] The KNX Association, *What is ets5 professional?* Version 5.74, Aug. 15, 2020.

[161] J. Cook. (2018). “The Right Tool for the Job: Active and Passive Infrared Sensors.”

[162] The Echelon Corporation, *Izot commissioning tool (ct)*, version 1, Oct. 1, 2020.

[163] D. Kushner. (2013). “The Real Story of Stuxnet.”

[164] NCCIC. (2016). “ICS-CERT Annual Vulnerability Coordination Report Industrial Control Systems Cyber Emergency Response Team 2016.”

[165] F. Inc. (2016). “Overload CRITICAL LESSONS FROM 15 YEARS OF ICS VULNERABILITIES 2016 Industrial Control Systems (ICS) Vulnerability Trend Report.”

[166] O. Andreeva, S. Gordeychik, G. Gritsai, O. Kochetova, E. Potseluevskaya, S. I. Sidorov, and A. A. Timorin. (2016). “INDUSTRIAL CONTROL SYSTEMS VULNERABILITIES STATISTICS.”

[167] I.-C. Kaspersky. (2018). “Threat Landscape for Industrial Automation Systems in H2 2017.”

[168] C. R. Team. (2020). “CLAROTY BIANNUAL ICS RISK & VULNERABILITY REPORT: 2H 2020.”

[169] R. Di Pietro and L. V. Mancini, *Intrusion Detection Systems*, 1st ed. Springer Publishing Company, Incorporated, 2008, ISBN: 0387772650.

[170] Shodan. (2021). “Shodan is the world’s first search engine for internet-connected devices.”

[171] Censys. (2021). “Attack surface management continuous internet discovery and complete cloud visibility.”

[172] J. Controls. (2021). “Metasys building automation system.”

[173] Tridium. (2021). “The open alternative.”

[174] J. Bender. (2015). “Welcome to BACpypes.”

[175] C. Tremblay, *Bac0*, https://github.com/JoelBender/bacpypes, 2021.

[176] Autodesk. (2021). “Multidisciplinary bim software for higher-quality, coordinated designs.”

[177] buildingSMART International. (2021). “Industry foundation classes (ifc) - an introduction.”

[178] *Bacnet - a data communication protocol for building automation and control networks*, ASHRAE 135-2016, ASHRAE SSPC 135, Jun. 2016.

[179] A. Inc. (2019). "What are the benefits of bim?"

[180] TrueCADD. (2019). "Level of development (lod)."

[181] S. INC. (2020). "Bim level of development (lod) 100, 200, 300, 400 500."

[182] S. Tang, D. R. Shelden, C. M. Eastman, P. Pishdad-Bozorgi, and X. Gao, "Bim assisted building automation system information exchange using bacnet and ifc," *Automation in Construction*, vol. 110, p. 103 049, 2020.

[183] S.PRASANTH. (2017). "Hr room."

[184] buildingSMART International. (2020). "Ifcjson-4."

[185] MongoDB. (2021). "The database for modern applications."

[186] P. Biondi. (2021). "Packet crafting for python2 and python3."