Vulnerability Assessments of Building Management Systems

Raymond Chan, Forest Tan, Ulric Teo, Brandon Kow

To cite this version:

Raymond Chan, Forest Tan, Ulric Teo, Brandon Kow. Vulnerability Assessments of Building Management Systems. 14th International Conference on Critical Infrastructure Protection (ICCIP), Mar 2020, Arlington, VA, United States. pp.209-220, 10.1007/978-3-030-62840-6_10 . hal-03794630

HAL Id: hal-03794630
https://inria.hal.science/hal-03794630

Submitted on 3 Oct 2022

Chapter 1

VULNERABILITY ASSESSMENT STUDIES FOR SMART BUILDING MANAGEMENT SYSTEM

Raymond Chan1, Forest Tan2, Ulric Teo3 and Brandon Kow4

Abstract A building management system (BMS) enables the capability to control the infrastructure within a building. The elevator system, power meters, gas, and water supply system can be monitored and administered by the BMS. In short, a BMS can be considered a miniature industrial control system, which is much more common to reach and found by the users. Due to this reason, the cybersecurity of BMS is a hot topic, as the recent trends are to include Internet of things (IoT) devices, sensors, and internet connection capabilities of the BMS. A vulnerability assessment should take place for BMS operators to understand the risks and impact of cyberattacks. In this paper, we performed two case studies on two vulnerability assessments of the smart BMS subsystem.

Keywords: Vulnerability Assessment, building management system, cyber-security, Smart cities

1. Introduction

In recent years, the smart city has become a popular trend around the world. It encourages the integration of the IoT to have more convenient administrating, monitoring, and analyzing of the surrounding physical environment. A BMS (a.k.a., building automation system) is one of the systems that will install IoT devices and sensors. A BMS includes a power control system, water and gas supply system, elevator system, and fire alarm system. All those subsystems are critical for building operations. The reasons for installing IoT devices and sensors are to enhance monitoring and control capabilities; reduce the malfunctioning of the subsystems; and perform predictive maintenance.

The smart building management system (BMS) is a new term that describes combining the use of IoT devices with a traditional building management system. Due to the original architecture of the BMS, installing a new IoT system allows the external connection to bring a new security loophole into the existing BMS. Moreover, the communication between the BMS and IoT devices creates potential attacks. This is because there are no standards for telling the vendors to use a common protocol to communicate between devices. The communication medium is also not protected and does not have to be secure. Therefore, it is necessary to conduct a vulnerability assessment for the BMS.

In this paper, we introduce the existing work on BMS security, followed by two vulnerability assessment case studies and testing approaches for the BMS subsystem. Last but not least, we provide possible future works in this area.

The rest of this paper is organized as follows. Section 2 summarizes existing works about security vulnerabilities and BMS protection. Section 3 introduces the vulnerability assessment case studies for two smart BMS subsystems. Section 4 gives a conclusion and proposes possible future contributions.

2. Related work

The threat analysis of the elevator control system is one of our previous works showing the possibilities of performing attacks on the BMS subsystem [3]. We also proposed a methodology to conduct forensic analysis on the programmable logic controller [4]. For smart city security related research, Baig [2] presented the security landscape and the security threats of a smart city. He mentioned that building automation systems (BAS) protocols are inherently insecure due to the amount of trust given to sensors and controllers. The BAS has to have forensic preparation, and the protection of BAS can prevent significant problems in a smart city. Khatoun [5] introduced security and privacy concerns and countermeasures in smart cities. Al-Dairi [1] provided an overview of the major security problems and influencing factors in smart cities. He pointed out that the key issues in smart city's security are the emergent integration of technologies, which leads to an unbound attack surface.

For BMS security related research, Wang [9] introduced a threat model of the infrastructure, assessment, and mitigation approach of a smart city. Mundt [7] conducted a security analysis of building automation systems and introduced possible attacks on it. Paridari [8] proposed a cyber-physical-security framework to a Building Energy Management

System and a security information analytics algorithm to recover resilient control from an attack. Minoli [6] reviewed the technical challenges faced by the IoT in the smart building arena. The fragmented nature of the industry leads to possible security issues in Smart Buildings. Wang [10] proposed an approach to secure the Microkernel-Based Controller, which is used in building automation systems.

To summarize, no study has been conducted to assess the real vulnerabilities of the Smart BMS and what could be done if we can attack a real BMS subsystem. To fill up this gap, we perform a comprehensive study on conducting a vulnerability assessment on some subsystems in Smart BMS. To discover the possible findings, we conduct real cyberattacks and analyze the impact of the attacks.

3. Case study

In this section, we analyze two BMS subsystems, namely the Building Energy Metering System (BEMS) and Smart Lighting System (SLS). Both systems are located in the advanced cybersecurity lab at the Singapore Institute of Technology (SIT).

3.1 Case study 1: Building Energy Metering System (BEMS)

The aim of this case study is to understand the Modicon M580 programmable logic controller, Unity Pro XLS, which programs the M580 and the Industrial Control System (ICS), as well as to perform penetration testing to test the security vulnerabilities of the M580 PLC.

Figure 1 shows the network architecture of the ICS system that is currently being worked on. In the supervision network, it consists of the workstation that contains the supervision software, which allows the operator to carry out modifications to the system. The SCADA System, which is the SMART X, can be found on the same network. The next section of the ICS architecture is the production network, which consists of the M580 Hot standby Primary and Secondary PLC, and the Relay Input & Output (RIO) Drop, which facilitates communication between the input and output of the PLC and M580s.

3.1.1 Architecture of Building Energy Metering System. One key component of the supervision network is the SmartX controller. The SmartX AS-P server performs critical functionalities such as logic control, trend logging, alarm supervision, communication support, and connectivity to the Input/Output and field buses making use of the

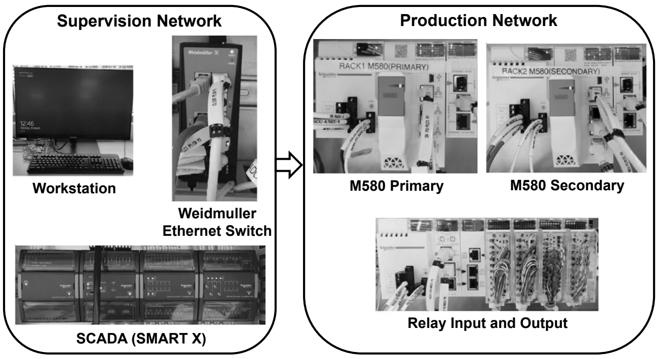
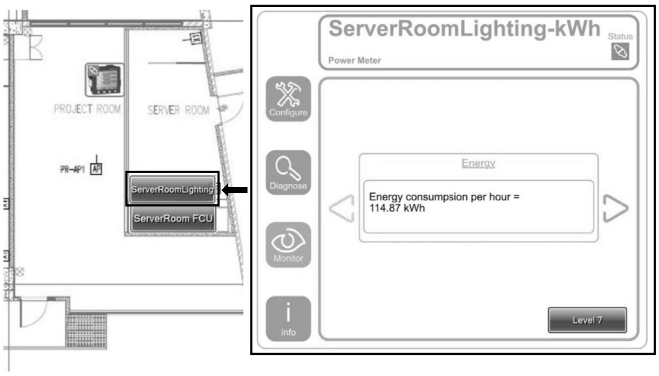


Figure 1. Architecture of the Power monitoring subsystem



Energy consumpsion per hour =
114.87 kWh

Figure 2. Server Room Lighting status

EcoStruxure BMS software. The intelligence of the EcoStruxure BMS ensures fault tolerance in the system and provides a fully-featured user interface via workstation and web station.

To fully understand the supervision network, some work needs to be done in order to ease the work for future operators using the SCADA system. One of the achievements was the creation of a floor plan to observe the status of each power meter located in different rooms of the building, simulating supervisory control within the structure.

As shown in Figure 2, a floor plan was created, which enabled the operator to check the status of the fans, lights, and every other electric appliance in the room.

3.1.2 Schneider Electric M580 Hot Standby System.

Acting as the PLC for this current ICS is the M580 Hot Standby System. It was designed to eliminate any downtime. The system achieves high availability through redundancy. Both primary and standby programmable automation controls (PACs) are configured with identical hardware and software.

The M580 is currently located in SIT @ Nanyang Polytechnic. One of the M580s acts as the primary, which runs the application by executing program logic communication on the sensors and actuators via RIO drops. Secondary M580 acts as the standby programmable automation controller (PAC). The primary PAC updates the standby PAC at the start of every scan and checks the health of the Ethernet RIO network link and the hot standby connection between the essential and backup CPUs. The secondary M580 is ready to assume control within one scan if the first stop prematurely or breaks down. It checks the health of the primary PAC and identifies modules in both primary and standby racks, the application version running in both primary and standby M580, firmware versions of the primary and standby CPUs, and the health of the links between primary and standby CPUs.

3.1.3 Vulnerability assessment. First, we have to find out the vulnerabilities of the ICS system. In this experiment, various methods were used to conduct penetration testing of the M580 PLC system. The first step of penetration testing is to detect the IP addresses of the M580. Various methods were used to perform these device scanning attacks, and the two most effective ones were Nmap (Network Mapper) scan and PLCSCAN. An Nmap scan was conducted in search of M580. Nmap has detected two IP addresses – 192.168.10.1 (primary M580) and 192.168.10.2 (standby M580). Nmap has shown the open http and ftp ports on these PLC, which hackers could use to exploit.

The second type of device scanning attack is known as PLCSCAN. As compared to the Nmap scan, which scans for open ports on the devices in the computer network, PLCSCAN only scans for specific ports such as port 102 (Siemens PLC) and port 502 (Modbus). PLCSCAN is used to detect M580. A range of IP addresses was typed into Kali Linux with PLCSCAN software, and the Modbus port 502 was discovered by PLCSCAN. It pulled out information such as the PLC module, the firmware of the PLC, and the version.

Simple Network Management Protocol (SNMP) is an Internet standard protocol for collecting and organizing information about managed devices on IP networks and for changing that data to change device behavior. It is one of many scans done after receiving the host's IP address. It is conceivable to get the default network name of the remote SNMP server. An aggressor can utilize this data to acquire learning about the

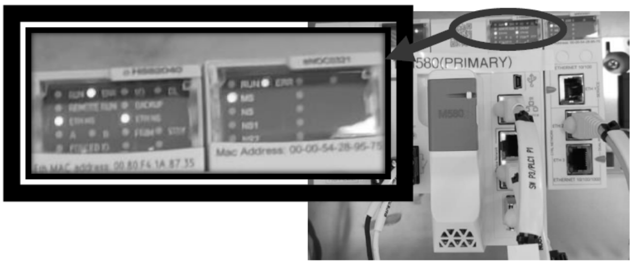


Figure 3. M580 after DoS attack

remote host or to change the design of the remote framework. So, Kali Linux is used to replicate the device scanning attack.

After performing both Nmap and PLCSCAN, the IP addresses of the M580 are known. These are the results that were found. Attackers can receive the network configuration of the M580, the different network routes configured, and also all the listening ports on TCP and UDP. After detecting the M580's IP address, the next step of penetration testing is exploiting the M580. Several methods were explored to test system vulnerabilities, and these were the most efficient and effective.

A DoS attack was performed in the M580. By exploiting function codes, the hackers can send a 'kill switch' into the M580, replicating a DoS attack. The results of the DoS attack on the M580 are shown in Figure 3. It rendered the functionality of the M580 unavailable. The M580 has been downed, and any connectivity and communication towards the M580 have been cut.

3.2 Case study 2: Smart Lighting System (SLS)

The smart lighting system (SLS) that is used for the project is modeled after a common smart office lighting setup and is established on a wireless local area network (WLAN). There is a total of 15 lights, and seven multi-function sensors and the arrangement of hardware are shown in Figure 4.

3.2.1 Architecture of Smart Lighting System.

The system consists of the following components: A supervisory system/web application -A local server which contains services such as web server hosting and Representational State Transfer Application Programming Interface (REST API) -that also double up as a client that access the

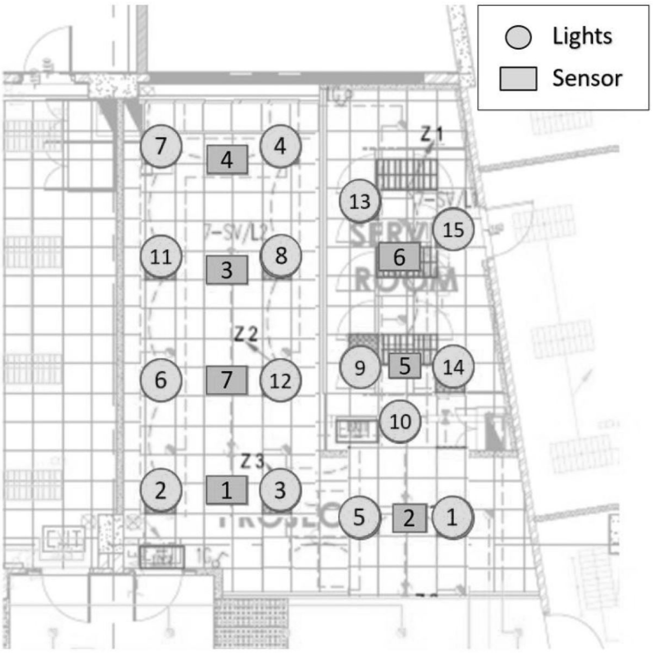


Figure 4. Placement of sensors and lights

web interface hosted within the supervisory system to control the lights.

A ZigBee gateways -Relays data packets from the router to the ZigBee drivers in the form of Zigbee wireless transmission.

Zigbee drivers/controller -Controls the lights by the data packets received from the gateway as well as transmitting data packets from the lights and multi-function sensors back to the supervisory system.

Multi-function sensors -Contains Passive Infra-red, photo sensor and thermal sensor sense occupancy, light intensity and temperature respectively and transmit the information to obtain raw data back to the supervisory system via the Zigbee drivers/controller and gateway.

The data travel through the system in this specific order where the web interface enables the user to act to control the lightings. The web interface uses Apache Tomcat which is an open-source Java Servlet Container that runs web applications based on Java scripts as well as used as an HTTP server. The instructions are sent to the internet router via ethernet cable which is responsible for routing the instructions to the correct IP address which in this case will be the ZigBee gateway's IP address. The gateway will transmit it via ZigBee wireless to the various ZigBee driver/controller, and they will direct the instructions to the specific hardware.

3.2.2 Communication protocol analysis. An extensive analysis was conducted on the system to find out where the vulnerabilities were, and this was achieved using an open-source packet analyser known as Wireshark to capture incoming and outgoing data packets transmitting over the network. The vulnerability discovered was that the system uses a communication protocol known as Hypertext Transfer Protocol (HTTP) which is part of an application layer protocol sent over Transmission Control Protocol/Internet Protocol (TCP/IP) and the default port number is 80. This protocol has three phases which are initialisation phase, data exchange phase and termination phase. It uses a request-response API where an HTTP request contains the URL of the host and the request verb. A response will be sent from the other host containing the requested data as well as the status code.

3.2.3 Vulnerability assessment. In this section, three types of attacks will be introduced, which are man-in-the-middle (MitM), Packet Tampering, and denial of service (DOS). A Raspberry Pi 3B+ installed with Kali Linux and equipped with a WIFI antenna will be used to conduct the attacks on the smart lighting system. Figure 5 shows the architecture diagram and where the attacks will be focused on. For the purpose of the experiments, it is assumed that the attacker has a breach into the same WLAN network as the smart lighting system.

The attack that will be conducted will be an active MitM attack known as ARP poisoning/spoofing. This type of attack is where the attacker intercepts an original communication channel between the gateway and client so that the data are passed through the attacker.

The result shows that the data packets were successfully routed through the attacker and was able to retrieve the information sent by the system to the gateway. Based on the reconstructed data, the HTTP API command and the host IP address can be reverse engineered to form the same command used by the system.

Furthermore, other relevant information, such as the method allows for access control, which is POST, GET, OPTIONS, PUT, and DELETE can be retrieved. This provides the attacker to know that other forms of HTTP request parameters that can be used. For example, an attacker may issue out a DELETE request to permanently delete the resources rendering anyone from gaining the resource.

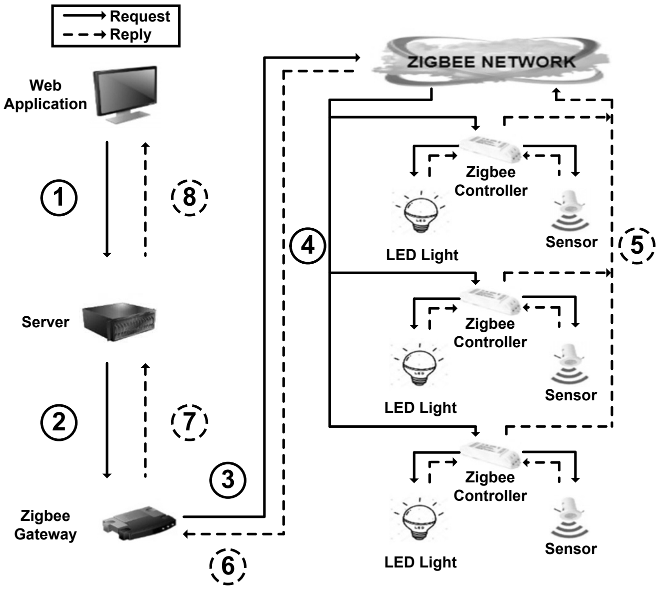
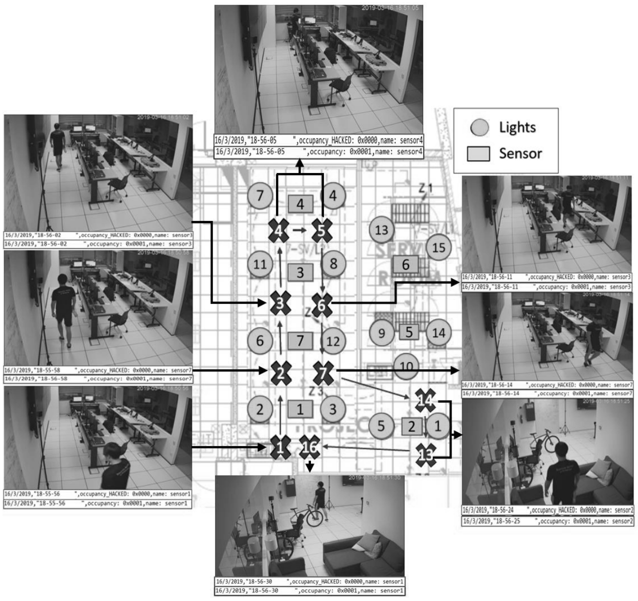


Figure 5. Architecture of Smart Lighting System

Lastly, the requested data were shown clearly in plain-text at the bottom. As the system sends out these requests in a fixed period, the total number of lights and sensors and their corresponding HTTP API command were able to be determined. An observation was made that there was no authorization needed to request the data from the gateway if the HTTP request is sent in the correct format. While MitM attack is one of the oldest forms of cyber-attack, analysts predict that there will be an increase in MitM attack happening on IoT devices due to a lack of standard mitigation against it. From the experimental results, MitM attack was able to spy and collect data and information, and this was achieved through the exploitation of HTTP vulnerability.

A web parameter tampering attack is an act of manipulating parameters within a data packet as they are exchanged between two parties. This attack can be executed by various forms of software such as Webscarab or Burp Suite, depending on the integrity of the system; they can be used for other forms of attack such Cross-site Scripting (XSS) or Structured Query Language (SQL) injection attack. This attack aimed to modify the occupancy data such that the sensor does not detect anyone and re-transmit back to the system. This causes the system to log the modified data instead of the actual data.

The occupancy data was successfully modified, as shown in Figure 6 below. The picture with the box highlighted in red shows the data logs



```
16/3/2019,"18-56-05     ",occupancy_HACKED: 0x0000,name: sensor4
16/3/2019,"18-56-05     ",occupancy: 0x0001,name: sensor4
```

```
16/3/2019,"18-56-02     ",occupancy_HACKED: 0x0000,name: sensor3
16/3/2019,"18-56-02     ",occupancy: 0x0001,name: sensor3
```

```
16/3/2019,"18-55-58     ",occupancy_HACKED: 0x0000,name: sensor7
16/3/2019,"18-56-58     ",occupancy: 0x0001,name: sensor7
```

```
16/3/2019,"18-55-56     ",occupancy_HACKED: 0x0000,name: sensor1
16/3/2019,"18-55-56     ",occupancy: 0x0001,name: sensor1
```

```
16/3/2019,"18-56-11     ",occupancy_HACKED: 0x0000,name: sensor3
16/3/2019,"18-56-11     ",occupancy: 0x0001,name: sensor3
```

```
16/3/2019,"18-56-14     ",occupancy_HACKED: 0x0000,name: sensor7
16/3/2019,"18-56-14     ",occupancy: 0x0001,name: sensor7
```

```
16/3/2019,"18-56-24     ",occupancy_HACKED: 0x0000,name: sensor2
16/3/2019,"18-56-25     ",occupancy: 0x0001,name: sensor2
```

```
16/3/2019,"18-56-30     ",occupancy_HACKED: 0x0000,name: sensor1
16/3/2019,"18-56-30     ",occupancy: 0x0001,name: sensor1
```

Figure 6. Result of a Packet Tampering attack

collected from the attacker, and the box highlighted in blue is from the system. The keyword and the value shown in the data log by the system have been changed and displayed following what the attacker wants. Comparing the occupancy values of the attacker and system shows that the system logs “0x0000,” which means there was no one detected, while the attacker logs “0x0001,” which indicates there was someone, and this was supported by the CCTV footage. Throughout this time, the person moves around the room. The system continues to log the modified data displaying as though no one is present. This form of attack is not limited to altering parameters, and it can be used to inject viruses or malware that create a backdoor to other devices on the network.

Using the information obtained from the MitM attack, the next form of attack can be conducted; it is known as a denial-of-service (DoS) attack, which is a type of cyberattack that causes the services of the target to be unusable by flooding the traffic and overloading the system. DOS attacks are launched from a single device to a target through Ethernet or wireless connections. It can be upgraded to a larger scale in which the flooding attacks are launched from numerous devices that are distributed globally on the Internet; this is known as a distributed denial-of-service (DDoS) attack. The result of this was a very useful

DoS attack, as it was able to achieve full denial-of-service of the system, preventing it from sending any requests.

The last DoS attack that was conducted was an application layer attack known as an HTTP unbearable load king (HULK) attack. These types of attacks consist of legitimate requests, such as GET/POST requests, that are aimed at crashing the target server by targeting Apache or Windows vulnerabilities, which also makes them harder to detect compared to the rest of the DoS attack. This attack was only able to achieve partial denial of service of the system as, during the particular point of the attack, the system was able to send out its request. The number of successful requests made by the system varied among the five runs of the attack, which lasted for 30 seconds. This could be due to the latency when the attacker was transmitting the packets, which resulted in the system getting its request through.

4. Conclusion and future works

To conclude, we conducted two vulnerability assessment case studies on two Smart BMS subsystems. These case studies show that it is possible to perform attacks on those subsystems. The case studies also indicate that it is necessary to conduct a vulnerability assessment of the smart BMS after deployment to understand possible attacks and mitigate their impact if the system has been compromised.

In the future, we are going to set up a smart living lab on our campus. We will test more subsystems, propose methodologies to secure those systems, and determine how to verify that they are resilient to cyberattacks.

References

[1] A. AlDairi and L. Tawalbeh, Cyber security attacks on smart cities and associated mobile technologies, *Procedia Computer Science*, vol. 109, pp. 1086–1091, 2017.

[2] Z.A. Baig, P. Szewczyk, C. Valli, P. Rabadia, P. Hannay, M. Chernyshev, M. Johnstone, P. Kerai, A. Ibrahim, K. Sansurooah, N. Syed and M. Peacock, Future challenges for smart cities: Cyber-security and digital forensics, *Digital Investigation*, vol. 22, pp. 3–13, 2017.

[3] R. Chan and K.P. Chow, Threat analysis of an elevator control system, in *Critical Infrastructure Protection XI*, M. Rice and S. Shenoi (Eds.), Springer, Cham, Switzerland, pp. 175–192, 2017.

[4] R. Chan and K.P. Chow, Forensic analysis of a Siemens programmable logic controller, in *Critical Infrastructure Protection X*,

M. Rice and S. Shenoi (Eds.), Springer, Cham, Switzerland, pp. 117–130, 2016.

[5] R. Khatoun and S. Zeadally, Cybersecurity and privacy solutions in smart cities, *IEEE Communications Magazine*, vol. 55(3), pp. 51–59, 2017.

[6] D. Minoli, K. Sohraby and B. Occhiogrosso, IoT considerations, requirements, and architectures for smart buildings — energy optimization and next-generation building management systems, *IEEE Internet of Things Journal*, vol. 4(1), pp. 269–283, 2017.

[7] T. Mundt and P. Wickboldt, Security in building automation systems - a first analysis, *Proceedings of the IEEE International Conference on Cyber Security and Protection of Digital Services*, 2016.

[8] K. Paridari, A. El-Din Mady, S. La Porta, R. Chabukswar, J. Blanco, A. Teixeira, H. Sandberg and M. Boubekeur, Cyber-physical-security framework for building energy management system, *Proceedings of the Seventh IEEE International Conference on Cyber-Physical Systems*, 2016.

[9] P. Wang, A. Ali and W. Kelly, Data security and threat modeling for smart city infrastructure, *Proceedings of the IEEE International Conference on Cyber Security of Smart Cities, Industrial Control System and Communications*, 2015.

[10] X. Wang, R. Habeeb, X. Ou, S. Amaravadi, J. Hatcliff, M. Mizuno, M. Neilsen, S. Rajagopalan and S. Varadarajan, Enhanced security of building automation systems through microkernel-based controller platforms, *Proceedings of the Thirty-Seventh IEEE International Conference on Distributed Computing Systems Workshop*, pp. 37–44, 2017.