Covert and Side Channels in Buildings and the Prototype of a Building-aware Active Warden

Steffen Wendzel

1 Faculty of Mathematics and Computer Science, University of Hagen, Germany
2 Faculty of Computer Science, Augsburg University of Applied Sciences, Germany

Abstract—Covert channels and side channels are barely discussed topics in the area of building automation. We define a building in the context of multilevel security (MLS) and show that covert channels and side channels exist in building automation. Additionally, we present a system called the building-aware active warden to eliminate covert/side storage channels in building automation systems (BAS). Active wardens aim to remove malicious (covert) elements in communications and are a well-known means from the area of network covert channels and steganography. Within the last years, new models, such as the network-aware active warden, were developed. The presented building-aware active warden is an adoption of the concept of a network-aware active warden to building automation. Building-aware active wardens modify or drop building automation commands as well as building information requests from users based on their security levels to enhance a building’s security. We extended an interoperable system for building automation supporting hardware from two vendors for the purpose of a building-aware active warden and for providing an unified application programming interface.

I. INTRODUCTION

Covert channels and side channels in building automation systems are still unhandled threats. We propose an active warden as an approach to handle such malicious communication channels. The diversity of the involved topics caused us to provide brief introductions before we describe the main idea of an building-aware active warden.

Covert Channels and Side Channels: A covert channel is a communication channel that was never designed to be used as a communication channel [1]. Thus, using network covert channels it is possible to send information in a way prohibited by a security policy [2]. Network covert channels can be used to keep a low profile when transferring illicit information [3]. Network covert storage channels utilize storage attributes of network protocols while network covert timing channels modify the timing behavior of network data (e.g. packet ordering/intervals) [3].

In contrast to covert channels, side channels leak information without an explicit sender [4] (e.g. cryptographic algorithms can leak key information through their cpu consumption, memory consumption or timing behavior).

A number of publications (e.g. [5], [6], [7], [8]) describe how to implement covert channels in network packets and packet timings. However, means against covert channels are available as well (e.g. [5], [9], [10], [11], [12], [13], [14], [15]).

Building Automation: Building automation systems (BAS) are systems within buildings which are used to control and monitor the building [16], mainly in the context of heating, ventilation and air-conditioning (HVAC). For instance, building automation systems are used to optimize a room’s indoor climate, to automatically open/close windows, to control the lightening and to monitor access requests to rooms. Due to the increasing number of automated buildings and their increasing capabilities, the importance and need for security in such environments increases. Security in networked building automation systems (BAS) was discussed in previous publications: [17] analyzed security aspects of fieldbus systems in building automation. [18] introduced EIBSec, a secure version of the EIB protocol using AES and providing confidentiality, integrity, data freshness, authentication, key management and key distribution functionality. The authors also explained possible economic impacts of security threats in BAS by giving the example of a BAS user who turns on a building’s lighting at night. [16] gives a more detailed view on building automation security by analyzing general aspects (such as intrusion detection) and also mentioned the possible threat of side channels (a detailed discussion of side channels was not presented).

Active Wardens: A traffic normalizer is a gateway with the capability to prevent malicious communication (e.g. caused by botnet software). Traffic normalization is usually implemented as part of a firewall system [19] and is capable of blocking and modifying network packets. For instance, a typical technique of a normalizer is to clear bits in network packets [20].

Since a traffic normalizer actively modifies traffic, it is called an active warden [21]. A passive warden does only monitor events (e.g. network traffic) to alert in case malicious communication (e.g. steganographic information transfer) is taking place. Active wardens with support for so-called active mapping reduce the problem of data that can be interpreted in multiple ways [22] by mapping a network (including its policies). Lewandowski et. al. presented an idea based on active mapping that is called the network-aware active warden [22]. Such systems contain knowledge about the network topology as well as they are capable to apply stateful traffic inspection to eliminate covert channels [23].

Our approach of a building-aware active warden is based on the idea of the network-aware active warden. In contrast to the network-aware active warden, our system does not only have knowledge of a network topology (in this case, the network topology of a building), but is also aware of a building’s users, their roles and security levels. In contrast to a firewall system,
we do not only block requests, but also modify them in order to allow partial requests to the BAS as long as they conform with the security policy.

We extend the existing approaches in BAS security by investigating covert channels and side channels in automated buildings. Additionally, we describe the design and implementation of the building-aware active warden for the network communication within the BAS by extending existing approaches with the concept of multilevel security (MLS). The building-aware active warden aims to prevent malicious communication within the BAS. We analyze the differences between the existing approaches for BAS security and the building aware-active warden. Open problems in eliminating side channels and covert channels in buildings are discussed as well.

The remainder of this paper is organized as follows. Section II introduces the problem of covert channels and side channels in building automation systems and explains our adoption of multilevel security (MLS) to building automation systems. Section III discusses the design and implementation of the building-aware active warden while Section IV discusses the results of our approach and explains a covert timing channel not eliminated by our concept. Section V concludes.

II. COVERT CHANNELS AND SIDE CHANNELS IN BUILDING AUTOMATION

We define a building as a multilevel secure (MLS) system. While this is not meaningful for small homes, the situation differs for public buildings and company buildings. Thus, our approach focuses on larger buildings where the concept of MLS can be applied. Since multilevel security systems must contain a set of levels, we adopt these levels from the organizational structure of a given company or public institution. In our model, as usual in multilevel security, a high security level dominates the lower security level. For associating persons with levels, we link levels in organizational charts to MLS levels. If, for instance, a company contains three classes of associated people: The CEO, some managers, and other staff, there must be three security levels (more can be applied, if necessary). Fig. 1 shows a sample MLS structure applied to an organizational chart.

Regarding to the Bell-LaPadula model, we must ensure that two rules can be applied to the building’s security levels [2]:
1) There must be no way to write confidential data from a higher level to a lower level (no write-down, NWD).
2) There must be no way to read confidential data from a higher level by a lower level (no read-up, NRU). If an attacker can bypass one of these rules, a covert channel or a side channel can be created.

In building automation, a side channel occurs when a low-level user (e.g. an employee) can monitor behavior of higher levels. Example 1: An employee uses a side-channel to monitor whether the company’s director is turning on his office’s lighting and thus is currently located in his office. This can either represent a write-down, i.e. the user can eavesdrop events occurring in the BAS, or a read-up, i.e. the user is able to request information of a higher level using the BAS.

A covert channel occurs when a high-level user sends information to a low-level user, which is less likely because such information can in most cases be transferred without the BAS. However, a covert channel can occur in large organizations. Example 2: Alice and Bob cooperate to steal money and are located in different floors of the same building. Alice can use the BAS (e.g. by turning on the light in Bob’s room) to signal Bob to raise attention (e.g. by creating some smoke for a fire alarm). While the building is getting evacuated, Alice can steal the money while not being linked to the fire alarm which was raised in another floor. Thus, Bob will be aspersed but can probably prove that he was at the location of the fire at the moment the money was stolen. Example 3: A (previously payed) warden could signal a prisoner (by turning off the lighting in the prisoner’s cell) that he will not be in the prisoner’s floor for a few moments. The prisoner can try to escape within this time and the BAS write-down is no obvious detectable communication.

In our model, an attacker is either internal (i.e. an employee with direct access to the BAS interface) or external (i.e. only remote (web) access to the BAS is provided). The goal of the adversary is (1) to obtain confidential data (e.g. detecting the location of inhabitants or whether someone is currently in the office) either by a read-up or a write-down using a side channel, or (2) to send confidential data to a lower level by using a covert channel by bypassing the NRU or NWD rule.

In our model, each person is linked to some devices as shown in Fig. 2. Thus, it is obviously easy to apply the mentioned MLS rules NWD and NRU. However, if two persons with different security levels share a device (e.g. two persons of two security levels both need access to the meeting room and both have access to the lighting in the meeting room), a conflict comes up (Fig. 2): If the device in such a shared room is linked to the higher security level, the low-level person is not allowed to access the device. If, on the other hand, the device is linked to the low level, the low-level user is able to obtain information via side channels. To solve this conflict, temporary permissions for low-level persons or temporary level-downgrades for the devices located in a room are thinkable.

![Fig. 1. Sample organizational chart with MLS levels.](image-url)
An additional problem occurs when users of a given MLS level are allowed to access all devices at their security level. For instance, a level 2 accountant can access level 2 research data. A solution for these kind of problems is to additionally apply role based access control (RBAC) as described in the next section.

Due to the low-level improvements such as EIBSec [18], the eavesdropping on the lower communication level by external attackers can be prevented since they cannot send/read encrypted frames to the building automation network. For internal attackers, such problems can also be solved by using asymmetric cryptography. At the moment, symmetric algorithms are used since the chips in the devices are not powerful enough to handle asymmetric algorithms in acceptable time [18]. However, this problem will be solved by more powerful hardware. Thus, our focus relies on the higher communication level (the middleware) where we concentrate on both, the prevention of write-downs (eavesdropping) as well as the prevention of read-ups due to prohibited information requests.

III. DESIGN AND IMPLEMENTATION OF A HIGH-LEVEL APPROACH

A first prototype was implemented to illustrate the usefulness of our security approach. The prototype is based on a previously developed middleware (described later in this Sect.) which can handle both, the HomeMatic automation system (www.homematic.com) by eq-3 and the CurrentCost energy consumption monitoring system (www.currentcost.com) by CurrentCost Ltd. Both systems are end-user systems but can be used to verify our approach nevertheless.

As already discussed, the focus of our building-aware active warden is on higher communication levels and not on the direct network access level. Therefore, the building-aware active warden is required to be located between the building’s end-user software and the building automation system itself (see Fig. 3). In our case, it is implemented as a network service running on a stand-alone system. Our prototype is designed for building automation systems containing a coordinator instead of peer components, i.e. there is a single interface the active warden can use to interact with the BAS. Other BAS architectures (multi-agent based) will be taken into account in future work.

Nowadays, users have – in many cases – direct access to a (part of the) building control software (e.g. to the HomeMatic web-interface) without any multilevel-secure protection means. Before we continue, a short exemplary introduction shall be provided for one system (the HomeMatic). The HomeMatic system contains multiple components (cf. Fig. 4): A user-accessible control interface, a central control unit (CCU), and a number of sensors and actuators used to monitor and control a building. The system works as follows: A sensor (e.g. temperature sensor) sends its value (e.g. measurement value of the current temperature) to the CCU. The CCU displays the current value on the user-interface (e.g. IPhone-interface or web-interface). On the other hand, a user can control the building via the control interface: The user clicks a button to trigger an event (e.g. turning on an electrical device), which is registered by the CCU. The CCU forwards the command to the actuator and the actuator executes the command.

Our approach prevents such direct access by giving direct access only to the building-aware active warden. All end-user software only has indirect access via the active warden’s unified API. In other words, there is no need to provide end-users and end-user software direct access to the building management system anymore, and all applications are forced to use the active warden’s API. We discuss the implementation details in Sect III-B.

A drawback of this approach is the fact that not all interactions are based on high-level software, e.g. a switch turning on/off the light in a room cannot be routed through the building-aware active warden. However, this drawback can be neglected since covert channels based on direct interactions require a low-level person’s direct access to a high-level device (e.g. lighting switch in a manager’s room) and can only be solved by low-level protection means (physical access control (PAC) [24] as well as the use of encryption, such as with the previously mentioned EIBSec [18]).

Maña et. al. already developed a role-based access control (RBAC) middleware for embedded systems (specially building...
automation) [25] and the IT4SE project developed a similar RBAC middleware called the home analytical system interface (HASI) for building automation with a focus on privacy for energy awareness-related applications [26]. We extended the HASI middleware to implement the building-aware active warden. Therefore, we decided to link users to security levels and to implement verifications for the previously described NRU and NWD rules.

Our building-aware active warden is based on a local database used as an information source for evaluating the policy-conformity of a user’s request via the API. The database contains information about a building’s users, their levels (in MLS context), their associated rooms and their associated devices.

For instance, a user X is linked to the role “head of IT” and is linked to a high security level $L_H$. User X has access to every room and every device with an associated level $\leq L_H$ and additionally associated with the role “head of IT”. On the other hand, a user Y is linked to the role “trainee” and is therefore linked to a lower level. He has access to a few rooms and can control and monitor only few devices, but cannot access rooms associated with a higher security level or associated with another role. However, a user can be associated with multiple roles, if necessary.

A. Architecture and Normalization

The architecture of the building-aware active warden is shown in Fig. 5 and based on the design of HASI presented in [26]. The building-aware active warden differs from HASI’s original model by providing the capability of altering traffic (instead of only blocking it like a firewall) and by addressing covert channel specific problems by applying the previously described multilevel secure (MLS) context.

The HASI multilayer architecture provides an interface for applications on the Unified Application Programming Interface (UAPI) layer, e.g. for energy consumption monitoring applications. The UAPI layer handles the communication with the building-aware active warden. Since an application using the API and the active warden are not necessarily located on the same system, the communication within both layers is done via a SSL encrypted connection. The UAPI layer additionally abstracts all low-level socket actions and all multiplexing, i.e. it is capable of handling multiple applications at the same time.

We implemented a simple communication protocol based on HASI’s communication protocol for the interaction between the user’s applications and the building-aware active warden. By receiving and verifying a user’s identification, the active warden can associate a user’s role, a user’s level and associated rights with the current connection. These information are (as described previously in Sect. III) used to apply modifications.

To get the current list of hardware available within all associated buildings, a user’s application needs to send a Hardware Listing Request (HLR) using the API. If the active warden receives a HLR, it verifies the permissions of the authenticated user associated with the connection. All hardware information a user has read-permission for is then read from the hardware layer and transferred to the UAPI. If a device, a room, a floor or a whole building is not accessible for a given user (the related role and MLS level), access will not be granted and information about these elements is not included in the HLR response message (e.g. most employees will only be allowed to control building automation objects in their own office rooms).

If a status information request (e.g. requesting the current temperature value of a temperature sensor) or a modification request (e.g. “turn off lighting in the meeting room”) is received, an access verification (for modification rights of the user and the associated role and level) is done by the active warden too.

A request is normalized by the active warden in a way that only the allowed parts of a request will be executed. To implement these features, modification requests contain a building identifier, the number of included hardware parameters ($n$), hardware identifiers, $n$ request types (e.g. floating point value or integer value), and $n$ values.

B. Implementation

As mentioned earlier, we developed a implementation of an inter-operable building automation architecture based on HASI by Rist et. al. [27] – a system with focus on energy-awareness for interoperable building automation. We extended the approach by adding the functionality to alter a user’s requests and to link a user to a security level within the MLS concept. The already developed support for two building automation/monitoring systems (HomeMatic as well as CurrentCost) was kept and the API and middleware layers were only required to be extended for our purposes.

The system is implemented in Python using a MySQL backend and is designed to run on low-cost hardware. Low-cost hardware prevents the drawback of consuming too much energy in BAS environments which aim to reduce the overall energy consumption.

The API supports different hardware-related request types dependent on the underlying requirements of each component. For instance, a power switch for a device can either be turned on or off, thus, switches can be controlled using a boolean parameter. On the other hand, there are floating point...
parameters. For instance, a user can request to set the heating level in a room using a floating point value as parameter.

For setting new values, the active warden not only verifies the permission of a user to set the state of a device but also ensures the policy-conformity of a value. If, for instance, the light is requested to be turned on at 2AM while the policy defines that working is not allowed between 1AM and 4AM, the active warden will drop that request.

Like other active wardens, the building-aware active warden also applies modifications to a user’s requests. We implemented such modifications for two situations: (1) The user wants to apply a number of settings to different hardware components. In case, the user is only allowed to apply a subset of the rules, the request is modified in a way that it only contains the allowed components. (2) If a floating value (such as setting the heating level in a room to 70%) is not allowed, but a can be allowed in a limited way, the request is modified (e.g. setting the heating level to 30% instead of 70%).

C. Emergency Situations

In our security model, the granted access for individuals is limited due to security levels and roles. However, in emergency situations (e.g. fire), a user should be able to get access to every exit door and window. To achieve this goal, we propose the implementation of an emergency functionality such as a special button in every floor or even in every room – such a button must not necessarily use the provided security architecture but can send direct low-level control commands to the BAS instead.

The emergency button can give the users control to all devices required to leave the building on a fast way without taking care of security roles and levels. Such an approach does not prevent covert channels or side channels but the safety of inhabitants is a mandatory requirement which we rate higher than limiting malicious communication channels.

IV. Results

In this section, different aspects achieved through our side and covert channel prevention concept using our prototype implementation of a building-aware active warden are evaluated.

Enforcing high-level access: While some existing low-level security approaches provide varying features (e.g. data encryption or protection for replay attacks), no global approach to limit malicious communication can be provided by low-level systems. Thus, our system administratively prohibits direct low-level access for future applications and therefore limits the threat of covert/side channels by applying RBAC as well as MLS features to the BAS. Since the building-aware active warden enforces policy-conformity, different threads which are already addressed by other authors (such as limiting economic impacts of turning on the lighting at night [18]) are limited as well.

Read-ups and write-downs: The presented system is able to eliminate “read-ups” by low-level applications which are based on our active warden middleware. The threat of write-downs can also be eliminated if the NWD rule of the Bell-LaPadula model is enforced.

To finalize example 1 of section II: The employee has no longer access to information of his manager’s office due to the applied BLP model. Example 2 is also handled since the active-warden can deny the communication between Alice and Bob due to the BLP model as well as due to RBAC. The same situation occurs for example 3 since BLP and RBAC prevent all non BAS-administrators to control the prisoner cell’s BAS components.

However, in practical situations, e.g. in a company where a leader wants to control all devices of a building, the NWD rule cannot be applied completely. A possible solution to at least reduce the threat of covert channels in such a case can be provided nevertheless using RBAC: A high-level process is forced to use a shared low-level resource for signaling covert information, and the process can only do so if its role is associated with the required low-level receiver’s role. The write-down to other low-level resources not associated with a given role is prevented. Thus, if the high-level sender and the low-level receiver are not associated with the same role and a shared device, the write-down is prevented.

The Problem of covert timing channels: While our approach is designed for storage channels (both, covert storage channels and side storage channels), there are problems for preventing timing channels. In case of shared resources, such as the devices in a meeting room which is used by processes of multiple MLS layers, “storage” information can be set invisible to low-level processes if the room is in use of high-level processes (e.g. the low-level process cannot monitor and control the devices for the duration of a meeting). However, a high-level process can alter a timing information by making those rooms “invisible” for arbitrary time intervals (the low-level process can detect such alternations in the context of elapsing time, which results in a covert timing channel).

Hu presented an approach called “fuzzy time” to reduce the capacity of covert timing channels in 1991 [10]. The capacity was reduced by introducing fuzzy time values (i.e. timing values with slightly incorrect values) by the VAX security kernel to virtual machines. We do not expect that the approach of “fuzzy time” can be successfully applied to building automation in practice since the timing intervals used in building automation (e.g. opening and closing windows) are already long in comparison to timings of kernel events in the VAX security kernel. Such a limitation would only result in a minimal reduction of the already low capacity timing channels in a BAS (e.g., opening and closing windows can take multiple seconds). Due to that already limited capacity, we consider BAS timing channels as a limited threat. By applying various covert timing channel detection algorithms (such as by Berk et. al. [9] or Zander et. al. [3]) we expect similar detectability for BAS as already exist for network timing channels. On the other hand, covert timing channels can become a threat if multiple protocols are used simultaneously, i.e. multiple shared devices are used at the same time, as described by Wendzel and Keller for covert storage channels [28]. However, while such techniques are thinkable for covert timing channels, their possibilities are – in comparison to covert storage channels
and their micro protocols – very limited. Currently there are neither publications nor implementations focusing on the idea of protocol switching covert timing channels.

Low-level storage and timing channels: However, we expect the existence of low-level covert storage channels and covert timing channels which are not in our scope, since they require a direct media access for both, sender and receiver. For instance, a covert channel’s sender could write a PDU to the wire (e.g. an EIB frame) and signal hidden information by altering unused frame attributes. Since such low-level covert channels require a direct media access, existing low-level protection means (as mentioned in Sect. I) can hinder the creation of such channels by forcing authentication and encryption as well as by providing limited access. Due to our high-level approach, no user or application must be granted a direct access to the wire since indirect API access is provided.

Legacy software: Our high-level approach can be implemented in BAS step by step since legacy applications which require a direct BAS access can be used in parallel. This allows backward-compatibility but old applications which are not controlled by the middleware can enable covert channels and side channels.

V. CONCLUSION AND FUTURE WORK

This work described the possibility to create side channels and covert channels within building automation systems (BAS) and explained the concept of a building-aware active warden aiming to eliminate those channels.

We explained the usefulness of defining a BAS as multilevel secure (MLS) system since we can prevent covert storage channels and side storage channels by applying the Bell-LaPadula model. To apply MLS, we link the hierarchy of organizations (as given by organizational charts) to the levels of users (e.g. employees) and additionally make use of an existing RBAC concept for building automation systems.

Future work will include the adoption of our middleware to building automation systems currently not supported by the building-aware active warden (such as multi-agent based architectures). Additionally, we aim to prove the usefulness of other covert channel prevention, limitation and detection means in the context of building automation. An important step in this direction would be to find ways to limit and detect covert timing channels in buildings.

ACKNOWLEDGEMENTS

The author would like to thank Jörg Keller for his valuable contributions and guidance, and Thomas Rist who lead the development of the HASI middleware which served as the base for the development of the building-aware active warden.

REFERENCES