--- Poster published in Proc. SECURWARE 2014 --

A Cost-Efficient Building Automation Security Testbed for Educational Purposes

Jaspreet Kaur, Michael Meier, Sebastian Szlósarczyk and Steffen Wendzel

Fraunhofer FKIE
Bonn, Germany
Email: {jaspreet.kaur, michael.meier,
sebastian.szlosarczyk, steffen.wendzel}@fkie.fraunhofer.de

I. Introduction

Building automation systems (BAS) are concerned with the control, monitoring and management of services such as heating, ventilation and lighting in buildings. They originate at a time when security was not of utmost importance. Nowadays, due to the rising desires for the increasing number of features and the inter-connectivity of BAS with the Internet, security must no longer be neglected. Besides, the lack of security in BAS did not led to an increase of security expertise among many people working in the field.

In order to perform research on the security of BAS, educational organizations such as universities as well as small and medium sized enterprises (SMEs) are required to gain access to BAS hardware and software components, which is linked to high costs and thus not affordable by many institutions. Our poster aims on sharing knowledge on the setup of BAS testbeds for universities and SMEs.

We present a low-cost testbed to educate students and employees in the field of BAS fundamentals and BAS security. The testbed allows to perform BAS security research for the *building automation control and network* (BACnet) protocol suite [1]. BACnet is a BAS protocol stack used worldwide and integrated into products by more than 760 vendors. Out of the four BACnet layers, we consider the two most important ones i.e. the network and the application layer: The protocols on these two layers do not depend on the low-level communication protocol used and their protocol structure remains the same in all BACnet environments, including IP-based BACnet as well as Ethernet- or MS/TP-based BACnet networks. Most of the functionality of our testbed is thus related to these two layers.

Covert channels are an emerging threat to BAS. Covert channel-based hidden data exfiltration (e.g. of surveillance data) can be created and monitored within the testbed. The testbed does support the evaluation of countermeasures for network covert channels in BAS.

The remainder of this paper is structured as follows. Section II highlights the related work on covert channels in BAS. Section III presents the description and implementation of the testbed and Section IV concludes.

II. Covert Channels in BAS

Covert channels are hidden communication channels not foreseen by a system's design. These channels are used to transfer secret information in a stealthy manner and aim on hiding the fact that communication is taking place. Like within Internet protocols (IPv4, IPv6, TCP etc.), covert channels also exist in BAS protocols. Covert channels in BAS can be used for different purposes [2]:

- Covert channels realize data exfiltration over the BAS network in order to bypass sophisticated commercially available data leakage protection (DLP) means, which do not foresee data leakage protection in BAS protocols [3].
- Moreover, covert channels allow bypassing BAS-internal protection means with policy breaking communication flows (e.g., for the undesired observation of sensor values).

The *BACnet Firewall Router* (BFR) [2] is the first approach to integrate simple firewall functionality into BACnet. The BFR can be used to enforce multi-level security to counter covert channels [3]. BFR is an open source project that implements filters for BACnet messages and is capable to realize NAT, software-side network switching, and routing.

Another defensive mechanism for eliminating covert channels in BACnet communications is *traffic normalization* [4]. A traffic normalizer is integrated into routers that inter-connect BACnet network segments in order to monitor the traffic exchanged between the devices. Besides being able to detect a set of anomalies, a normalizer drops or modifies packets containing malicious or incompliant content. The normalizer therefore uses normalization rules as a basis, which enforces the known protocol specification. With these capabilities, traffic normalizers are a simple and efficient way of defeating some of the known covert channels by eliminating ambiguities in network traffic.

III. Testbed

We developed a testbed that can be used to demonstrate BAS fundamentals, to teach BAS security, for educating students and employees, and to perform research on BAS security without requiring expensive BAS hardware. The testbed is highly configurable as each BACnet device is represented by virtual Linux machines (Linux VMs), what allows the

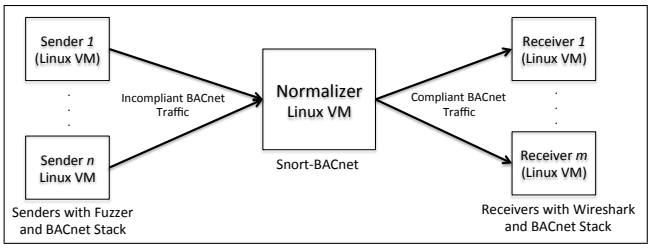

Figure 1. Virtual testbed for BACnet traffic

integration of a high number of devices (the maximum number is only limited by the computing resources used for the Linux VMs). Moreover, it can be used to teach students how covert channels can be implemented in BAS and how they can be defeated using traffic normalization.

The principal requirements for our virtual testbed are

1) cost-efficiency,
2) availability as open source, and
3) easy configurability for various scenarios.

Furthermore, since the major goal of establishing the testbed is for educational purposes, it should be easy to use with basic Linux command line skills, thus reducing the training time required. Considering these aspects, we choose the following components to fulfill our requirements (as shown in Figure 1):

1) Linux machines with the open source *BACnet stack* [5] to act as BACnet devices; these systems run as Linux VMs,
2) a Linux machine running *Snort* [6] with our Snort BACnet extension to act as a protecting traffic normalizer (we introduced this Snort extension in [7]),
3) our protocol fuzzer based on *Scapy* [8].

Arbitrary BACnet traffic can be generated on each Linux VM. BACnet traffic can be exchanged between all Linux VMs in the local network as well as with remote Linux VMs for which the traffic needs to pass the central Linux VM that acts a traffic normalizer.

With this setup, we could reduce the costs of monitoring, testing and analysis operations in the context of teaching and research. Typical BAS environments for BACnet cost at least a few thousand EUR up to multiple ten thousand EUR, while virtual machines consume standard resources which may be easier accessible, i.e. which already present at many universities. As the BACnet stack requires only limited computing power, old PC systems and laptops can easily be used as non-virtual devices.

Researchers, teachers, and students can dynamically observe the behavior and relationship of the components involved in the environment eliminating the need to analyze the complicated real-world environment. Network flows can be efficiently monitored with the help of *Wireshark* providing the opportunity to examine the inner details while transmitting the BACnet traffic from one end to the other. Thus, comprehensive testing can be carried out effectively without damaging real hardware or influencing real (or even critical) BAS operation. Furthermore, this ensemble offers a prominent advantage of setting up BAS research clusters between different universities using the BACnet/IP.

Using our Scapy fuzzer installed on all Linux VMs, it is easy to create covert channel traffic that can be observed using Wireshark at the receiver side. The behavior of the covert channel traffic can be observed at the normalizer VM as well as at the receiving VMs. Research concepts on detecting, limiting, and preventing network covert channels in BAS can be implemented as proof-of-concept codes into our existing traffic normalizer code that provides the necessary interface. This allows easy integration and evaluation of covert channel countermeasures.

However, while our testbed is linked to various advantages, two significant drawbacks were determined as well while performing research and teaching tasks. Firstly, the temporal behavior (e.g. response time) of the simulated BACnet devices on Linux VMs differs from typical embedded BACnet devices. For instance, performing Denial-of-Service (DoS) tests in our testbed cannot be directly adapted to non-virtualized BACnet environments. Secondly, the availability of application layer features of the used open source BACnet stack is minimal and does not posses the full spectrum of application layer services as foreseen in the BACnet standard, making various application-related tests impossible.

IV. Summary

Our poster shares knowledge and experiences for the cost-efficient realization of a BAS testbed. To this end, we decided to use BACnet, which is one the most popular BAS protocols, as a basis for our work. The testbed allows teaching and researching different aspects; especially of the BACnet network layer but due to drawbacks it cannot be recommended when it comes to application layer-related research and teaching. Moreover, the timing behavior of our testbed differs greatly in comparison to embedded BACnet environments.

References

[1] "ISO standard 16484-5:2014: Building automation and control systems (BACS) – part 5: Data communication protocol," 2012.

[2] "BACnet Firewall Router (BFR)," http://bfr.sourceforge.net/.

[3] S. Wendzel, B. Kahler, and T. Rist, "Covert channels and their prevention in building automation protocols: A prototype exemplified using BACnet," in Proc. GreenCom 2012, Nov 2012, pp. 731–736.

[4] M. Handley and V. Paxson, "Network intrusion detection: Evasion, traffic normalization, and end-to-end protocol semantics." in 10th USENIX Security Symposium, 2001.

[5] "BACnet Stack," http://bacnet.sourceforge.net/.

[6] "Snort," http://www.snort.org/.

[7] 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." in Sicherheit 2014. LNI 228, 2014, pp. 407–418.

[8] "Scapy," http://www.secdev.org/projects/scapy/.