SAFECARE

Integrated cyber-physical security for health services

Specification of the BMS Threat Detection System

Deliverable 5.3

Lead Author: FST

Contributors: CCS, CSI

Deliverable classification: PU

Co-funded by the Horizon 2020 Framework Programme of the European Union

Version Control Sheet

Title	*Specification of the BMS Threat Detection System*
Prepared By	*Mario Dagrada, Daniel dos Santos*
Approved By	*Fayçal Hamdi, Elisabetta Biasin*
Version Number	*1.0*
Contact	mario.dagrada@forescout.com

Revision History:

Version	Date	Summary of Changes	Initials	Changes Marked
V.01	24.10.2019	Initial draft for ToC and content	MD	
V0.2	08.11.2019	Adding new content to all sections	MD, DdS	
V1.0	14.11.2019	Adding reviewers' comments	MD	

The research leading to these results has received funding from the European Union's Horizon 2020 Research and Innovation Programme, under Grant Agreement no 787002.

Contents

List of Figures

List of Tables

Table of Acronyms

Acronyms	Description
BAS	Building Automation System
BMS	Building Management System
BTDS	BMS Threat Detection System
CoAP	Constrained Application Protocol
DICOM	Digital Imaging and Communications in Medicine
DoS	Denial of Service
HDO	Healthcare Delivery Organization
HL7	Health Level 7
POCT	Point of Care Testing
HVAC	Heating, Ventilation and Air Conditioning
IDS	Intrusion Detection System
IoT	Internet of Things
IT	Information Technology
TCP	Transmission Control Protocol

MQTT	Message Queue Telemetry Transport
OT	Operational Technology
PLC	Programmable Logic Controller
VLAN	Virtual Local Area Network
MITM	Man-in-the-middle
DPI	Deep Packet Inspection

Executive Summary

The challenge of SAFECARE is to bring together the most advanced technologies from the physical and cyber security spheres to achieve a global optimum for systemic security and for the management of combined cyber and physical threats and incidents, their interconnections and potential cascading effects. The project focuses on health service infrastructures and works towards the creation of a comprehensive protection system, which will cover threat prevention, detection, response and, in case of failure, mitigation of impacts across infrastructures, populations and environment.

Over a 36-month timeframe, the SAFECARE Consortium will design, test, validate and demonstrate 13 innovative elements, developed in the Document of Actions, which will optimize the protection of critical infrastructures under operational conditions. These elements are interactive, cooperative and complementary, aiming at maximizing the potential use of each individual element. The consortium will also engage with leading hospitals, national public health agencies and security Stakeholders across Europe to ensure that SAFECARE's global solution is flexible, scalable and adaptable to the operational needs of various hospitals across Europe, and meets the requirements of newly emerging technologies and standards.

This deliverable (D5.3) details the specification of an innovative network-based Intrusion Detection System (IDS) to be developed by Forescout that leverages in-depth protocol parsing and is specifically designed to protect healthcare Building Management Systems (BMS) from cyber-attacks. This IDS, called BMS probe in the context of SAFECARE, will combine whitelisting (machine-learning based) approaches and blacklisting (attack-specific) approaches to detect a wide range of possible attacks to BMS.

More specifically, this document highlights the main security challenges of BMS, the requirements for the BMS-specific threat detection system, a detailed system architecture, and the interconnections with other SAFECARE components, namely the Advanced Malware Analyzer and the Cyber Threat Monitoring System.

The SAFECARE Project

Over the last decade, the European Union has faced numerous threats that quickly increased in their magnitude, changing the lives, the habits and the fears of hundreds of millions of citizens. The sources of these threats have been heterogeneous, as well as weapons to impact the population. As Europeans, we know now that we must increase our awareness against these attacks that can strike the places we rely upon the most and destabilize our institutions remotely. Today, the lines between physical and cyber worlds are increasingly blurred. Nearly everything is connected to the Internet and if not, physical intrusion might rub out the barriers. Threats cannot be analyzed solely as physical or cyber, and therefore it is critical to develop an integrated approach in order to fight against such combination of threats. Health services are at the same time among the most critical infrastructures and the most vulnerable ones.

They are widely relying on information systems to optimize organization and costs, whereas ethics and privacy constraints severely restrict security controls and thus increase vulnerability. The aim of this proposal is to provide solutions that will improve physical and cyber security in a seamless and cost-effective way. It will promote new technologies and novel approaches to enhance threat prevention, threat detection, incident response and mitigation of impacts. The

project will also participate in increasing the compliance between security solutions and European regulations about ethics and privacy for health services. Finally, project pilots will take place in the hospitals of Marseille, Turin and Amsterdam, involving security and health practitioners, in order to simulate attack scenarios in near-real conditions. These pilot sites will serve as reference examples to disseminate the results and find customers across Europe.

1 Introduction

A smart building integrates physical systems and digital infrastructures, allowing devices to communicate with each other using network protocols, such as BACnet, KNX, and Zigbee [1] [2]. Current examples of smart buildings include not only Healthcare Delivery Organizations (HDOs) such as hospitals and clinics, but also other critical and non-critical facilities, such as airports, residences and office spaces.

Building Management System (BMS), also known as Building Automation System (BAS), is the term used to refer to the automatic centralized control of a smart building's Heating, Ventilation and Air Conditioning (HVAC), lighting, physical access and other automated systems. In the past, the subsystems of a BMS were independent from one another and control took place with a hardwired or serial connection. This configuration made such systems relatively safe from attacks. In recent times, we have witnessed an increasing adoption of standard IT communication (e.g. with the use of IP-based networks) to control BMSs. Furthermore, while BMSs have traditionally operated as standalone entities, they are now often paired with Internet of Things (IoT) devices [3].

With the introduction of the Internet of Things, such systems may even be connected to the Internet; hence, attackers can exploit vulnerabilities on protocols and devices to launch attacks on a building. In 2017, more than 16000 building automation devices have been found accessible, and possibly exploitable, on the Internet [4], including devices in hospitals and clinics [5]. There are well-known cases of real-world attacks, including the 2013 hacking of Google's Sydney office [6]; the 2016 attack that turned off the heating systems in two buildings in Finland [7]; and the 2017 attack that locked hotel guests in their rooms in Austria [8].

These attacks can lead to economic loss or even harm building occupants [9] [10]. Attacks on smart buildings can, e.g., cause blackouts by damaging power systems; block access to exits or grant access to restricted areas by tampering with physical access control; or damage data centers by stopping air conditioning. These worrying scenarios have shown that BMS are a new Achilles' heel of cyber security and this threat cannot be ignored, especially in the healthcare domain.

In healthcare-related buildings, BMS subsystems include fire detection, access management, video monitoring, power supply, and air-cooling systems. Technical implementation of these systems mainly relies on Programmable Logic Controllers (PLC). These systems use specific ports and protocols (often proprietary), and thus require dedicated tools to analyze the network traffic. Moreover, these systems share the network with medical devices, IT systems and other Operational Technology (OT) systems.

This document details the specification of an innovative network-based Intrusion Detection System (IDS) that leverages in-depth protocol parsing and is specifically designed to protect healthcare BMS from cyber-attacks. This IDS, called BMS probe in the context of SAFECARE, will combine whitelisting (machine-learning based) approaches and blacklisting (attack-specific) approaches to detect a wide range of possible attacks to BMS.

Specifically, the BMS probe will adopt anomaly detection techniques to identify anomalous devices, services and communication patterns. For instance, by applying frequency-based analysis, the BMS probe will be able to identify devices that behave differently from their pairs as well as to detect abnormal usage of legitimate services. The BMS probe will improve incident detection by sending security events to the Cyber Threat Monitoring System in case of suspicious activity on the network traffic.

The rest of this document is organized as follows. Section 2 presents the main security challenges of BMS. Section 3 describes the requirements for the specialized threat detection system. Section 4 details the system architecture of the BMS probe. Section 5 shows the interconnections with other SAFECARE components, namely the Advanced Malware Analyzer and the Cyber Threat Monitoring System. Finally, Section 6 concludes this report.

2 Building Automation Security

In this Section, we discuss the network architecture of building automation systems (Section 2.1), the main security threats to which these systems are exposed (Section 2.2), and some examples of attack scenarios (Section 2.3).

2.1 Network Architecture

Building automation networks are usually organized in three levels. The *field level* contains sensors and actuators that interact with the physical world. The *automation level* implements the control logic to execute appropriate actions. The *management level* is used by operators to monitor, configure, and control the whole system.

Devices in these levels communicate via network packets to share their status and send commands to each other. Sensors send their readings to controllers, which in turn decide what actions to take, and communicate their decisions to actuators. For instance, a sensor reads the temperature of a room and provides it to a controller, which decides to switch a fan on or off, according to a setpoint configured by a management workstation.

Devices are also typically grouped in subsystems according to their functionalities. For example, smoke detectors are part of the fire alarm system, whereas badge readers are part of the access control system. Ideally, these subsystems' networks should be segmented from each other, and especially from IT networks, although that is not always the case in practice. Sometimes different subsystems are configured in different VLANs for network segmentation, but misconfigurations allowing cross-VLAN communication (VLAN hopping) are common [11].

The architecture of a typical smart building network is shown in Figure 1, where systems including Video Surveillance, Access Control, and HVAC are connected. The OT devices in the different subsystems use either proprietary or standard domain-specific protocols such as BACnet, KNX, and LonTalk [12] to communicate. More recently, IoT devices like smart lights, smart locks, smart electrical plugs, and other sensors and actuators started being deployed alongside building automation systems [13] using protocols such as Message Queue Telemetry Transport (MQTT) and the Constrained Application Protocol (CoAP) to achieve machine-to-machine communication and establish a common message bus.

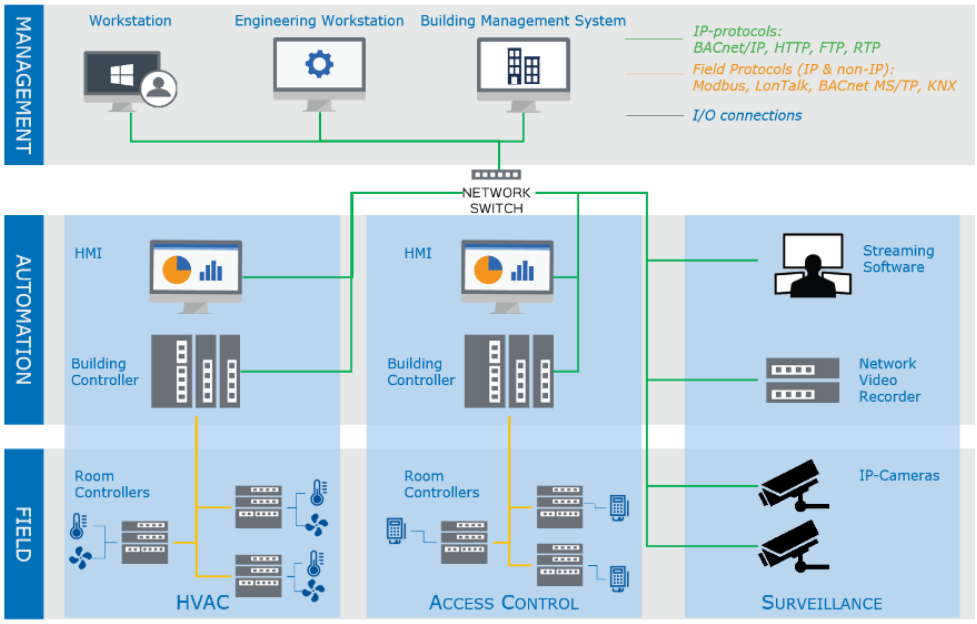

Figure 1 - The architecture of a typical Building Automation Network

In the specific case of HDOs, besides the building automation and general IoT devices mentioned above, there are often many connected medical devices sharing the network. Although these medical devices are not part of the building automation network per se, their traffic is often mixed with the traffic of other IT and OT devices. As discussed above, this happens because of misconfigurations and a general lack of proper network segmentation [14]. It is important to understand the relationship between medical devices and the rest of the network because those are the most critical devices in an HDO and because attackers may pivot from other IT and OT devices into medical devices or vice-versa [15].

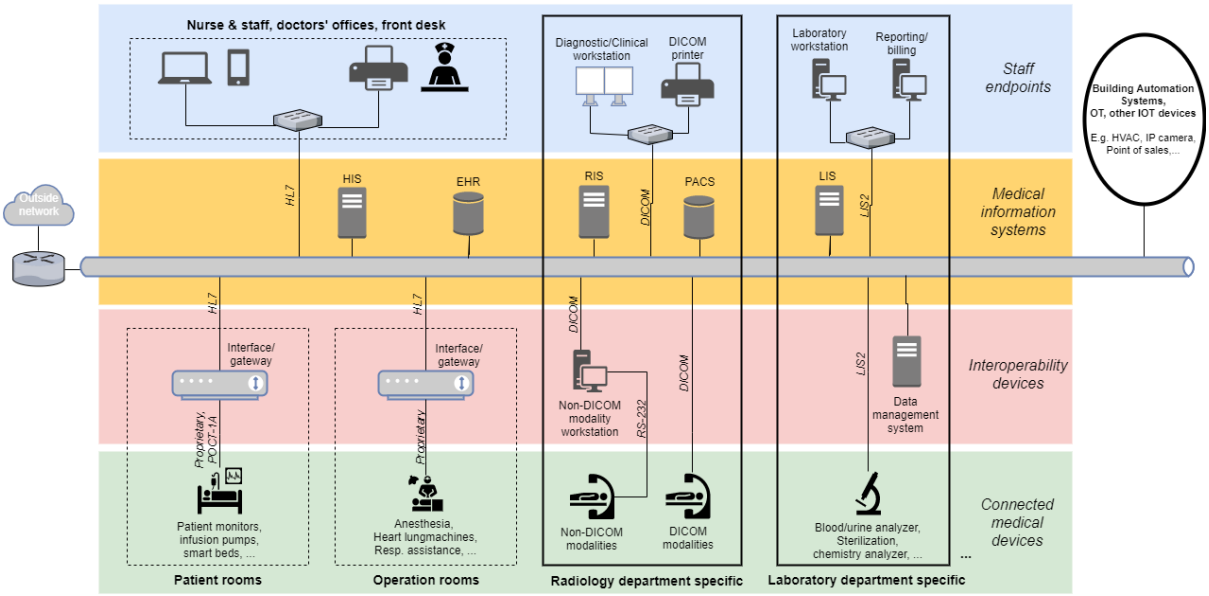

Figure 2 - HDO network

Figure 2 shows a network model for HDO networks, depicting the diverse IT, OT, IoT, and medical devices, as well as their interconnection and networking protocols. HDOs are generally divided into several departments, which deliver specific clinical care (e.g., radiology and laboratory) or organizational services (e.g., administration or accounting). On the Figure, we represent two of those departments in the plain-line boxes (while keeping in mind that there would be likely more departments in a real HDO). While some departments can have specialized equipment related to their operations or care to deliver (e.g., imaging modalities in radiology department), there also are certain device types that can be found in multiple departments. As an example, the lower left part of the Figure shows a patient room and an operation room with their respective devices to be likely connected to the network: multiple operation rooms can be found in the surgery department for instance. In addition, there are resources that can be found ubiquitously across an HDO such as IT devices. The Figure depicts some of these devices in the upper left part: for instance, doctors will have their own computers querying databases for patients' electronic health records, while the front desk personnel can create records to admit new patients.

Based on their purpose, we classify the devices seen on the network of an HDO into 4 categories. *Connected medical devices* support clinical care, while *interoperability devices* assure communication for some devices on the network. Then *medical information systems* store and manage clinical data and finally *staff endpoints* provide human interfaces to information systems.

Connected medical devices correspond to devices that support clinical care and can be further divided into active or passive devices [14]. Active medical devices are meant to deliver medical treatment. This type of devices includes for instance insulin pumps, heart defibrillators, or any equipment sustaining patients' life. Passive medical devices monitor patient information (e.g., vital signs and test results) and report/alert events or need for treatment to clinical staff. These include patient monitors, laboratory equipment, imaging devices, among others. Depending on the network protocol used by the devices discussed above, they may be connected to interoperability devices, which will convert network data into an interoperable format, allowing it to be further processed and/or stored by medical information systems. Such systems form the backbone of an HDO, as they collect, store and manage various types of healthcare data. For example, health, radiology, and laboratory information systems (respectively HIS, RIS and LIS), will manage electronic medical records, radiology pictures from imaging modalities and laboratory analysis results, respectively.

Medical devices in HDOs transmit data on the network using standard or proprietary protocols, such as: HL7v2, which is the most widely used interoperability and data exchange protocol in medical networks; DICOM, which defines both the format for storing medical images and the communication protocols used to exchange them; and POCT1-A and LIS2-A2, which are used for point-of-care and laboratory devices, respectively.

2.2 Security Threats

Recent versions of building automation protocols support some security features to provide data authentication, confidentiality and integrity, but their implementation is usually optional. Besides, many buildings still operate legacy versions of these protocols, which have little or no built-in security [16]. Many smart buildings operate with data being exchanged without any kind of authentication, and devices in them are programmed to process every message received, which means that any attacker that manages to reach the network where those devices are located can control them. Regardless of the protocol employed, IoT and building automation devices are

notoriously vulnerable to things like command injection and memory corruption vulnerabilities, due to poor coding practices which allow attackers to bypass their security features and gain full control of them.

Software and network vulnerabilities are not the only cause for concern for facility managers. Recently, a hacker in the Netherlands shut down the cooling system used to store pharmaceutical drugs in a supermarket [17]. This hacker was a disgruntled former employee, who logged in remotely from Norway directly into the building automation system with an old set of credentials. He succeeded in accessing and shutting down the cooling system, but timely response from the store management contained the damages and mitigated the risk. A key takeaway from this incident should be that insider threats are a valid risk for any organization, and a BAS can be hacked by someone with a little know-how and motive.

The landscape discussed above opens smart buildings to exploitation by both internal and external attackers, who have different backgrounds and motives [9] [10]. **Internal attackers** are building employees or occupants, who have authorized access to the building and prior knowledge of systems and devices. They may exploit vulnerabilities or directly perform unauthorized actions. Their motives are varied and may include financial gain, espionage, or revenge. System administrators, operators and other personnel may also be considered internal "attackers" when their unintended mistakes disrupt the normal functioning of the building. **External attackers** are unknown to the building's systems and act from the outside. They may get access to systems via social engineering techniques or by exploiting network vulnerabilities. External attackers may be hackers, criminals or competitors, with diverse motives.

Attacks on building automation systems can have varying degrees of complexity and goals. Besides attacks that attempt to take control of the functions of a building [18] [19] [20], more subtle attacks have also been theorized. For instance, researchers have demonstrated how to use building automation networks as botnets [21] and how to use the HVAC system to bypass "air gaps" (i.e. reach isolated networks) via a covert thermal channel [22].

After reviewing the relevant literature (see, e.g., [4] [5] [9] [10] [16] [18] [19] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32] [33] [34]), we identify the following four main categories of attacks on building automation networks and devices. The following list is not exhaustive, but is representative of the prevailing kinds of attacks that must be detected by an intrusion detection system monitoring a BAS network.

1. Network Reconnaissance (or Snooping). The attacker wants to gain knowledge of network topology and gather information about assets, services, objects and properties. This knowledge can then be used to, e.g., discover vulnerabilities and plan the next steps of an attack or to determine whether someone is present in their office or home [35].

2. Device Writing Access (or Tampering). The attacker wants to tamper with the normal operations of a building automation system by changing internal values of variables in devices. For instance, an attacker might tamper with the setpoint of the temperature in a room (as exemplified in the first threat scenario above) or with the response of an access control request (as exemplified in the second scenario).

3. Traffic Redirection (or Spoofing). The attacker impersonates a legitimate network device to read, modify or reject the intercepted messages that will either never reach their legitimate destination or will be tampered with before delivery.

4. Denial of Service (DoS). The attacker wants to disable the communication between devices or to make a whole network unavailable. With this attack, building operations can be interrupted and a manual reset of devices might be necessary.

Many malicious actors have motivations to attack HDOs [36] [14]. For instance, individual cyber criminals or criminal organizations usually try to reap money from their cyberattacks, either directly (via, e.g., ransomware and cryptomining), or indirectly (via, e.g., stolen information and use of infected computers in botnets). For examples of reported attacks on HDOs, see, e.g., [37] [38] [39] [40].

Security research in the medical space usually focuses either on devices or network protocols. Vulnerabilities in specific medical devices have been found over the past years (see, e.g., [41] [42] [43] [44]), and the number of security advisories in the medical space has been growing [45]. Currently, there is a trend of research into protocol insecurity in healthcare, where many protocols support neither encryption nor authentication (or support them without enforcing their usage, in the case of DICOM), a situation similar to what is found in BAS devices. More precisely, vulnerabilities of the following protocols have been demonstrated:

- HL7 standards can be abused in several ways [46] [47] [48] [49]. While they are used to exchange patient data between systems, researchers have shown that these standards are often insecurely implemented. Consequently, as the HL7 data is sent over unencrypted and unauthenticated communications, it is possible for an attacker to intercept and modify information in transit, which could ultimately cause safety hazards to patients.
- In a similar fashion, unencrypted DICOM communications could also allow an attacker to tamper with medical images, misleading medical staff to wrong diagnostics. As demonstrated in a proof of concept [50], the researchers implemented an attack in which tumors could be added or removed from CT scan images while being transferred over the network. Such attacks could lead to dramatic consequences for patients.
- Proprietary protocols have also caught the attention of security researchers. Douglas McKee has shown [51] how one could intercept a patient's vital signs sent by a GE patient monitor over their RWHAT protocol. Once intercepted, the malicious actor could modify the patient signs arbitrarily.

2.3 Example Attack Scenarios

To better illustrate the consequences of attacks to building automation systems, we will briefly discuss three example attack scenarios, each with a different impact on people, devices, and business operations.

Data centers: Many organizations use large data center facilities to store and process their data. Electronic devices used in a data center are susceptible to damage from high temperatures and depend on robust cooling and air conditioning systems, which are now connected to the BAS network. If an attacker is able to access the HVAC system of a data center by exploiting a device or network vulnerability, they can raise a temperature setpoint to disable the air conditioning. As a result, the facility will overheat, leading to equipment damage or, more probably, to safety mechanisms shutting down the data center. It is expected that safety mechanisms shutting down data centers will kick-in after less than a minute of high temperature [52]. In either case, the organization's normal operation will be severely affected.

Physical access control: HDOs usually employ access control systems to grant or deny access to certain areas of a hospital. These systems are comprised of access badges, badge readers, controllers, and databases that store user credentials. When a user swipes their badge on a reader, their credentials travel through the network to reach a controller that accesses a database to check whether or not the user has access to the area behind the badge reader. If the user has access to that area, then the controller sends a signal to an actuator to open the door. Otherwise, the access is not permitted. An attacker who has access to the automation network of the hospital is able to send malicious commands to control the doors and gain access to forbidden areas. Furthermore, the attacker can perform a combination of this and the previously described attack scenario. They could lock all doors of the building and increase/decrease the temperature to cause an insufferable condition for patients or people working in the building.

Medical devices. As described in Section 2.1, HDOs use many different types of active and passive medical devices to deliver care to patients and these devices are sometimes reachable in the same network as building automation and other IoT devices. Active medical devices can be compromised to cause harm by denying or modifying treatment, whereas passive medical devices can affect patient health through a clinician by reporting false information and medical events or not reporting medical events. Other attack possibilities involve altering medical records, altering work orders, altering medicine inventory, and modifying test results. An attacker who wants to harm a patient directly could achieve this goal in at least two ways. First, the attacker could use a building automation or IoT device as a pivot point to exploit a known vulnerability in a medical device connected to the patient to cause a DoS or tamper with a critical setting (e.g., rate of drug delivery in an infusion pump). Second, the attacker could use a building automation or IoT device as a man in the middle between a target medical device and its related information system in order to tamper with the sensitive traffic between these two systems, thus sending potentially dangerous commands or tampering with test results.

3 Requirements

This section presents different types of requirements for the BMS Threat Detection System. The requirements mentioned in the SAFECARE requirement analysis deliverable D3.4 "Initial requirements analysis" that are relevant for building automation security are also integrated in this section.

3.1. Functional Requirements

The main goal of the BMS threat detection system is to leverage the network traffic in order to detect possible attacks to building automation devices in an HDO. The specific functional requirements for the solution are as follows.

- Security trend analysis: The solution should analyze network traffic from connected devices to detect trends that indicate potential security attacks.
- Device misuse detection: The solution should be able to detect suspicious events from the network traffic, such as dangerous operations performed on a device.
- Post-incident analysis: The solution should facilitate forensic investigations of security incidents.
- Alert generation: The solution should generate timely alerts for the detected security events and send them to relevant stakeholders.
- Input to risk management model: The solution should provide insights about the security posture of the devices and its environment that becomes input to the risk management model of the devices.
- Accuracy: The solution should be able to distinguish likely threats from normal usage with a reasonable degree of accuracy.
- Vulnerability detection: The solution should inform relevant stakeholders (e.g. operators) of passively detected system vulnerabilities, even if they are not being actively exploited. For example, the threat detection system should detect and inform the operators if the devices have unpatched vulnerable components.
- General intrusion detection checks: The solution should be able to detect general attack methods used by common hacking tools such as port scans and man-in-the-middle.
- Specialization: The solution should offer BMS-specific functionality over a general-purpose intrusion detection system product.

3.2. Performance requirements

The requirements in terms of overall performance of the security analytics solution are as follows.

- Non-interference: Threat detection will not interfere with communication between, or the functionalities of, building automation devices and other existing infrastructure, both in terms of CPU load and network traffic.
- Event detection time: Threat detection should happen as soon as possible after the suspicious event has happened on the network.
- Alert generation time: Events can be immediately forwarded to a responder or to SOC operators.

3.3. Other requirements

Some requirements that are not covered in the above sections are mentioned below. These requirements are inherited from the requirements provided by SAFECARE deliverable D3.4 "Initial requirements analysis".

- Security updates: When a new relevant vulnerability is published, the solution should receive an update that allows it to detect exploitation of this issue. There should be an easy and a resilient way to get the updates.
- Portability: The solution should not rely too much on the specifics of a deployment environment, such as a particular brand of firewall or router being in use, or operators using one type of operating system or browser.
- Customizability: Operators should be able to manually tweak the configuration to reduce the number of false positives/negatives. That is, the ones who get alerts should also be able to add some custom rules to the threat detection system.
- Scalability: The solution should be scalable across different groups of operators examining different types of events, scalable in different situation such as when the network throughput increases.
- Traceability: Sufficient information should be provided in the generated alerts so that responders can identify an issue, and the actors involved with it.

4 System Architecture Specifications

This Section describes the architecture of the BMS Threat Detection system (BTDS). We first outline the overall architecture of the BTDS. We then give an overview on the main network protocols which need to be supported to enable monitoring of a wide range of BMS networks. Finally, we describe the different detection engines which can timely raise alerts if suspicious activity is detected in the network traffic.

4.1 Overall architecture

As described in Section 3, one of the main requirements of the BTDS is not to disrupt operational continuity of the building automation network in the HDO due to its criticality for patients' and building occupants' safety.

In order to cope with this requirement, the BTDS is conceived as a network intrusion detection system (NIDS) only based on passive monitoring detection modules. This reduces the interference with critical network operations.

The BMS Threat Detection system high-level architecture is depicted in Figure 3 below.

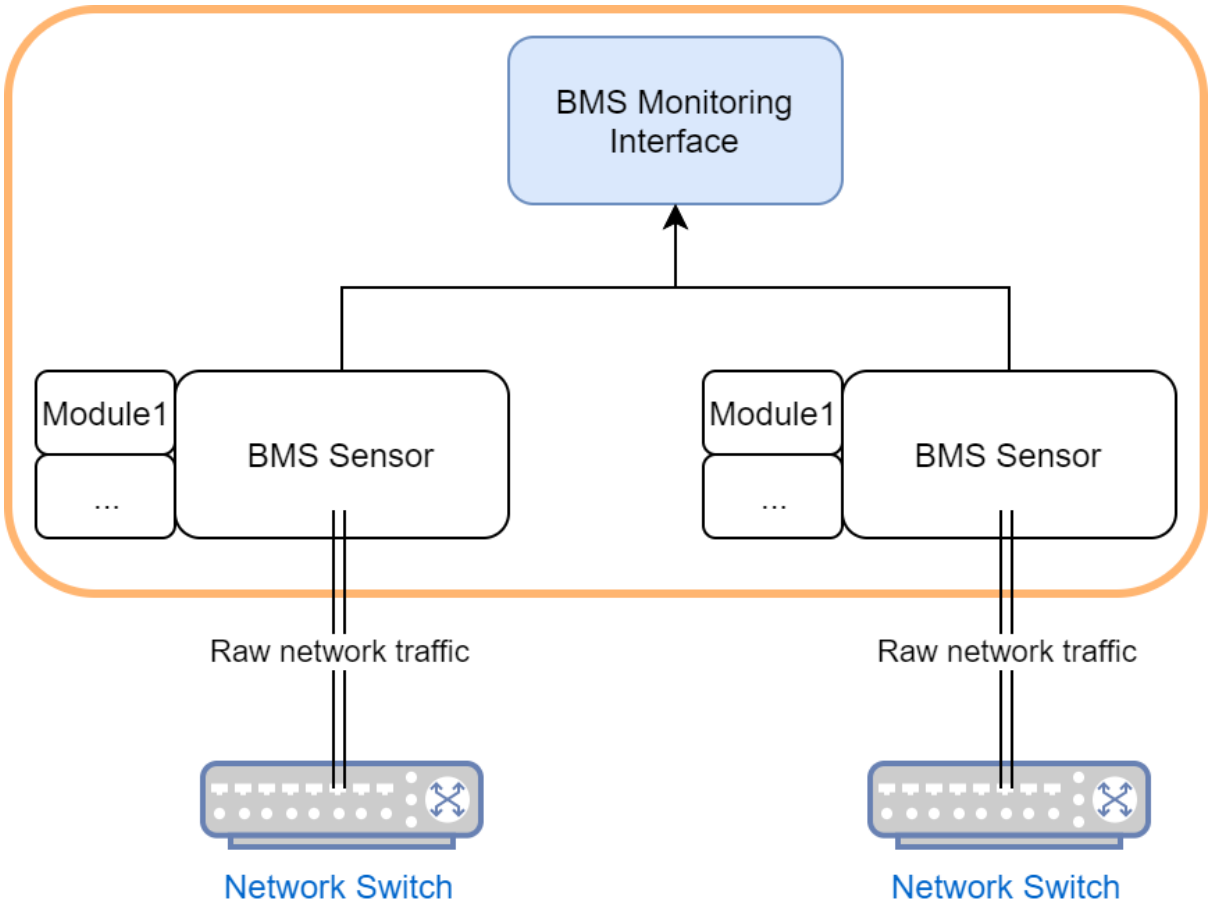

Figure 3 - BMS Threat Detection System overall architecture

More in details, the BTDS encompasses the following components:

- *BMS Sensor*: this is the core of the BTDS. It consists of an advanced and efficient network sniffer, which intercepts and dissects the traffic passing on the wire in a completely passive fashion using deep packet inspection techniques [53]. A sensor is connected to a

pre-configured Switched Port Analyzer (SPAN) port of a switch located in the BMS network. The SPAN port mirrors all traffic going through the switch. To satisfy the requirements, the BMS sensor can dissect several open standard and proprietary protocols commonly used in building automation and medical systems.

- *Detection modules*: these modules are the passive detection engines incorporated in the BMS sensor. These modules are responsible for analyzing the network traffic parsed by the sensor and raising alerts in case of suspicious activity such as the change in the logic of a BMS controller. For each alert raised, a short packet capture before and after the suspicious activity can be stored by the sensor in order to facilitate post-incident forensics analysis. Each module is fully configurable and can be turned off if needed. The available detection engines will be described more in details in Section 0.
- *BMS Monitoring Interface*: this component receives, aggregates and visualizes the data coming from the sensors placed in the BMS network. The BMS monitoring interface provides the user with actionable information on the assets present in the network, assigns a risk scoring to those assets to enable correct asset prioritization and allows real-time control over what is happening in the network through visualization analytics. Finally, the BMS monitoring interface is responsible for sending the alerts raised by the detection engines to third-party systems such as the Cyber Threat Monitoring System in the SAFECARE platform.

4.2 Supported Network Protocols

As discussed in the previous section, the core of the BTDS is the BMS sensor capable of dissecting building automation and medical network protocols. This component provides the evidence extracted from raw network traffic, which is then fed into the detection engines for raising security alerts. For protocols that support file transferring, the BMS sensor can also dissect the files and make them available in the monitoring interface.

In order to protect building automation and medical HDO networks as required within the SAFECARE project, the BMS Threat Detection system is extended with specific parsers for protocols commonly found in the aforementioned networks. In this section we detail the most important protocols which are needed to meet the coverage requirements outlined in Section 3.

4.2.1 Building Automation Protocols

The following paragraphs provide a description of three among the most widely used protocols used in building automation systems: BACnet, LonWorks and the Tridium Niagara protocol stack.

BACnet

Building Automation and Control network (BACnet) is a general purpose, multi-stack network protocol specifically devised to control several building automation systems such as HVAC, lighting and access control. BACnet is by far the most widely used network protocol in building automation systems and its integration allows the BTDS to cover large part of the use cases for HDO building networks. The importance of this protocol justifies the level of detail provided in this section.

BACnet uses a four-layer collapsed architecture including the physical, data-link, network and application layers of the ISO-OSI model. There are seven different combinations (options) for the physical/data-link layers that the BACnet devices can use; the network and application layers are used by all the options, as Figure 4 shows.

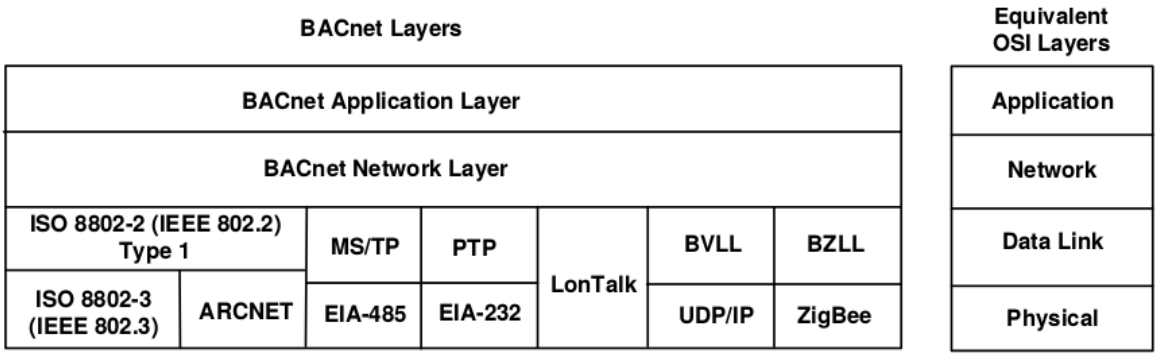

Figure 4 – BACnet collapsed layers architecture

Devices using different data-link layers are not able to communicate directly, so the BACnet Network Layer is used for this purpose. For instance, a device that uses BACnet/IP (BACnet over UDP/IP with BVLL data-link layer) cannot communicate with a device that uses BACnet MS/TP (Master-Slave / Token-Pass). Special devices (BACnet Routers, described later) are used to perform these connections by extending the BACnet Network Layer to include the source/destination Network and the source/destination address. The following paragraphs focus on the BACnet/IP protocol, which is the most adopted one.

Datalink Layer

The BACnet/IP (B/IP) protocol uses UDP/IP with the default port set to 47808 (0xBAC0). More UDP ports are supported by BACnet in the following ranges: 47809-47823 and 49152-65535.

The Data-Link layer used in this case is called BACnet Virtual Link Layer (BVLL) and its fields are shown in Figure 5:

- *BVLC type*: The first byte is the BACnet Virtual Link Control type that has the value 0x81, which identifies the B/IP option.
- *BVLC function*: The second byte defines if the message is a Unicast, Broadcast, or a Forwarded PDU. Also, it used for B/IP devices (Foreign Devices) that do not belong to any BACnet subnetwork (described later) to register to it.
- *BVLC length*: defines the length of the entire BACnet PDU
- *Data*: Transfer data according to the BVLC function

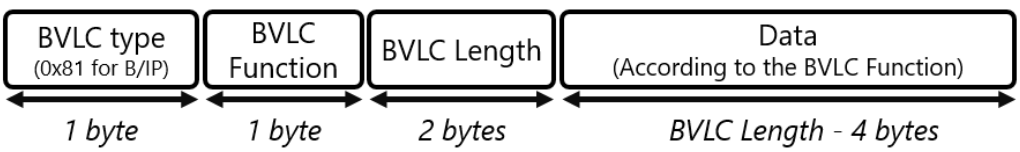

Figure 5 – BACnet Virtual Layer Link

Application Layer

BACnet Application Layer is on top of the BACnet Protocol stack and it is used to exchange data between BACnet devices. This layer contains the APDU, which is divided into the Application Protocol Control Information (APCI) and the Application Service Data Unit (ASDU). The APCI is the header of each APDU, while the ASDU has the actual data, such as the present value of a thermostat sensor.

The first byte of the header defines the type of the APDU. BACnet protocol has 8 types of APDUs defined in Table 1.

APDU Type	Description
Confirmed Request PDU	A client requests a service by a server and waits for a response, such as the ReadProperty service
Unconfirmed Request PDU	A client requests a service by a server without waiting for a response, like the I-Am service
Complex ACK PDU	A server uses this PDU to reply to a Confirmed Request PDU and to send data back to the client
Segment ACK PDU	When a Confirmed Request or a Complex ACK is transmitted in segments, a Segment ACK is sent by the receiver when the segment arrives
Simple ACK PDU	A server uses this PDU to reply to a Confirmed Request PDU, which does not need a complex response, namely just a confirmation that the request was executed
Error PDU	If an unexpected PDU was received then the device sends one of these PDU types, depending on the case
Reject PDU	
Abort PDU	

Table 1 – BACnet Application Protocol Data Unit

Transaction

Each Confirmed-Request needs a response (Complex-ACK, Simple-ACK, Error-PDU, Reject-PDU, Abort-PDU). For a successful communication between a client and a server, a transaction established and maintained by the Transaction State Machine of each BACnet device. Each transaction is identifier by an Invoke_ID [0...255], the BACnet client address and the BACnet server address.

Segmentation

BACnet supports segmentation for the Confirmed-Request and the Complex-ACK. The segmentation is used because of the different maximum NPDU length accepted by different physical/data-link layer options and BACnet devices. For BACnet/IP and BACnet/Ethernet the maximum is 1497 Bytes, for BACnet/MSTP, BACnet/Zigbee is 501 Bytes, and for BACnet/LonTalk is 228 Bytes.

The Segmented_Message flag defines whether the message is segmented, while the More_Follows flag indicates whether more segments follow or the last segment was received and the reassembling process can start. In case of a segment, the sequence-number byte is used.

The BACnet APDU conveys the actual information that the BACnet devices want to exchange. To do that, the devices use services (described later) and according to the service different data are transmitted. The service used is declared in the service_choice byte. All the APDU types have a service_choice byte.

BACnet Services, Objects and Properties

BACnet is an object-oriented protocol. Devices contain BACnet Objects, which in turn contain BACnet Properties, each of which contains a value. BACnet supports 54 object types. Each device must have one *device object* and as many objects of the other types supported by this device (according to the vendor), for instance *analog-input* objects.

Each object has a unique instance number (ID). The ID is a 22-bit numbers. That means that each device has to have a unique ID in the whole BACnet network, which allows 4194303 devices. Every device can have 4194303 objects of each object type. By combining the device and the object ID, the object becomes unique in the whole BACnet network (e.g.: Device ID: 5, Analog_Input object with ID: 200).

BACnet devices can exchange information by using BACnet services. The BACnet APDU is responsible for transmitting this information. The data are included in the Service-Request or Service-ACK field of the ASDU. There are four main categories of BACnet Services used for different purposes.

- **Alarm and Event Services**: are used to manage communication related to events, such as change of value of BACnet objects (e.g., ConfirmedCOVNotification).
- **File Access Services**: are used to access and modify files saved in BACnet devices (e.g., AtomicReadFile, AtomicWriteFile).
- **Object Access Services**: are used to access and modify BACnet objects and their properties (e.g., CreateObject, ReadProperty).
- **Remote Device Management Services**: are used to discover a remote device, to synchronize clock, or to initialize it (e.g., who-Is, TimeSynchronization, ReinitializeDevice)

According to the service used in the request, a Complex-ACK or a Simple-ACK is used for the response. For instance, the ReadProperty request needs a Complex-ACK with the values of the requested properties. On the other hand, a WriteProperty request needs a simple-ack that identifies that the request was executed properly, while an Error-PDU identifies an error (there is an error code and class in the message) during the execution of the request.

The services and objects supported by each device are specified in a Protocol Implementation Conformance Statement (PICS) document that the device's vendor publishes. PICS are available to the following page: http://www.bacnetinternational.net/btl/

LonWorks

LonWorks is a platform that provides a set of resources for Building Automation and control of elements such as HVAC or lighting. LonWorks belongs to the Echelon Corporation, who submitted

the networking communication protocol in 1999 (later known as LonTalk) and was accepted as a standard since then [54].

The LonWorks-based communication protocol [55] is one of the most widely deployed technologies for building automation worldwide, even if it is a proprietary platform. This is due to its relative low cost and its great compatibility with other manufacturers devices.

The protocol conforms to ISO/IEC 14908 (worldwide), EN 14908 (Europe), ANSI/CEA-709/852 (U.S.) and is also standardized in China. This protocol acts in a way that reminds a standard Local Area Network (LAN), and in fact, "LON" stands for Local Operating Network.

The protocol is characterized by the following features:

- LonWorks is suited for use with different types of transmission media, such as twisted pair cables, power line, RF, fiber optics or IP (both TCP/IP and UDP/IP), which makes it very flexible.
- Straightforward installation with a choice of different cabling topologies (e.g. star or line).
- The connection of objects via bindings (e.g. standard network variables, standard configuration properties) can be defined at the project engineering stage or can be adapted in the field. This simplifies the engineering process and helps prevent errors.

The network stack of LonWorks is presented in Figure 6.

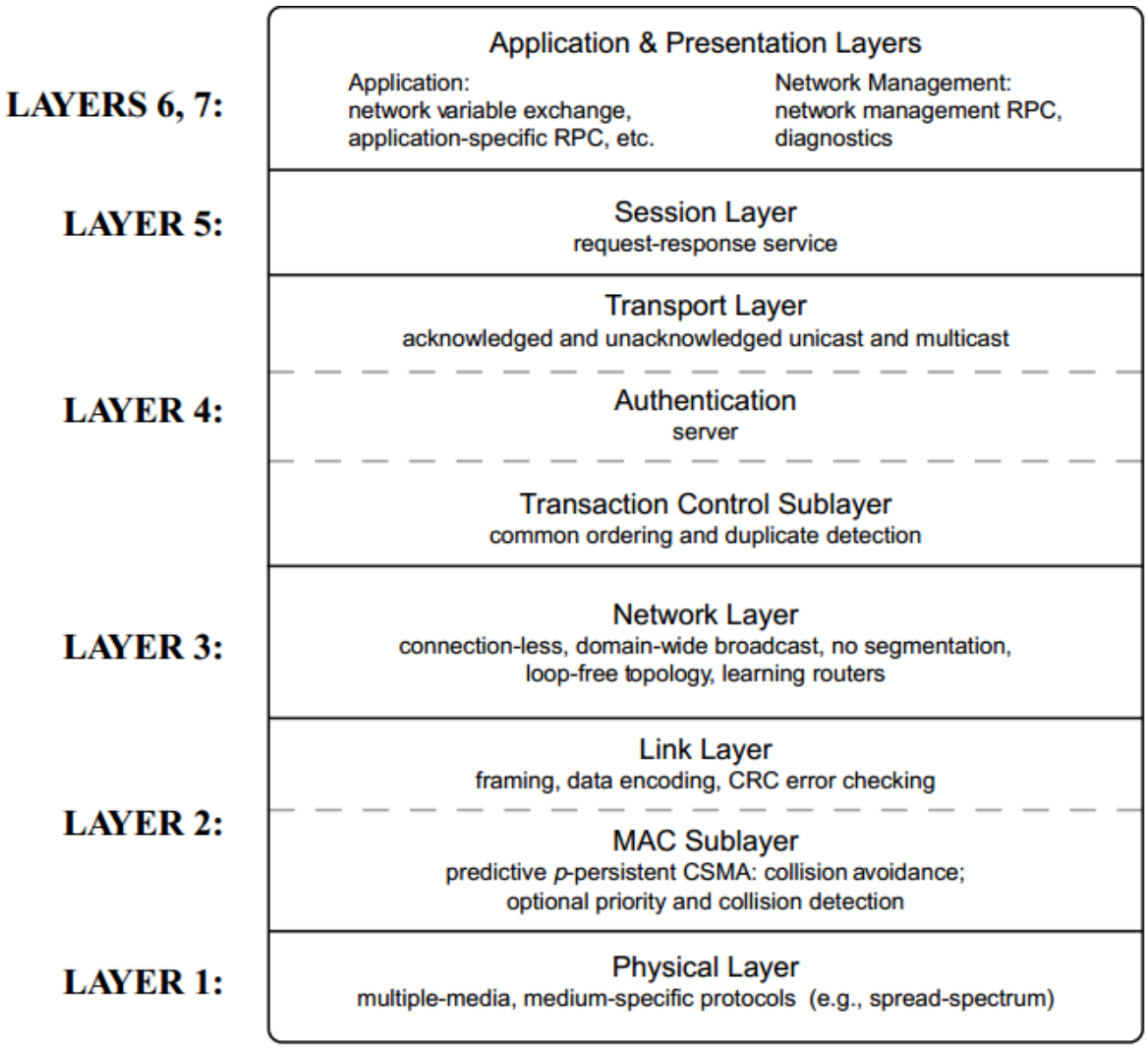

Figure 6 – LonWorks protocol stack

It is important to mention that, even if LonTalk (LonWorks protocol for communication) redefines the stack, many LonWorks applications have a great compatibility on IP-based devices. The tunneling standard ISO/IEC 14908-4 is used in LonWorks to offer this compatibility: IP-aware systems can be integrated with new LonWorks applications, network management tools. Other applications belonging to the platform offer interactive compatibility using Web Services, IP-routing etc.

One of the most important concepts regarding LonWorks is the Neuron chip. A Neuron chip is an integrated circuit that is installed in every LonWorks device and offers the protocol a hardware environment for its execution and treatment. This 8-bit processor acts like a 3-in-one microcomputer: two of the microprocessors are used for the communication protocol, and the last one is used for the node control application. This way, Neuron guarantees a better optimization of the internal operations for LonWorks and this is translated in a relief of the computational cost for the network response.

LonTalk the main rival of BACnet in builing automation. Both are standards, widespread and satisfying for most of client's needs. To increase the flexibility of both the protocols, since 2014 it is possible to use BACnet over LonTalk. This extends BACnet capabilities making possible to use

it along all the automation processes. This improves not only the compatibility, but also the features in Building Automation both protocols have to offer, and the spread of usage for them.

Tridium Niagara Protocol Stack

Tridium Inc. develops the widely used Niagara product line used in building automation controllers, particularly for access control.

Tridium Niagara, through its API and Java Virtual Machine, supports a slew of protocols. This platform supports the typical web services protocols that you would expect to find in the TCP/IP stack. Tridium also supports most of the open protocols in building automation, such as LonWorks, BACnet, and Modbus.

However, Tridium also develops its own proprietary protocols, which are described in the following.

Fox Protocol

The Fox protocol facilitates the communication between stations and a workbench software, as shown in Figure 7. Because Fox is used to communicate between devices without a driver, third-party systems cannot communicate to Tridium. Architecturally, Fox sits at the top levels or application/transport levels of the TCP/IP Stack. Fox utilizes port 1911 to communicate with other Tridium devices.

Fox features include:

- Layered over a single TCP socket connection
- Digest authentication (username/passwords are encrypted)
- Peer to peer
- Request / response
- Asynchronous eventing
- Streaming
- Ability to support multiple applications over a single socket via channel multiplexing
- Text based framing and messaging for easy debugging
- Unified message payload syntax
- High performance
- Java implementation of the protocol stack

NiagaraD

NiagaraD is the protocol for communication between workbench and daemon services. Whereas many competitive offerings utilize simple web services calls against XML data in their databases, Tridium utilizes a proprietary protocol to call upon data.

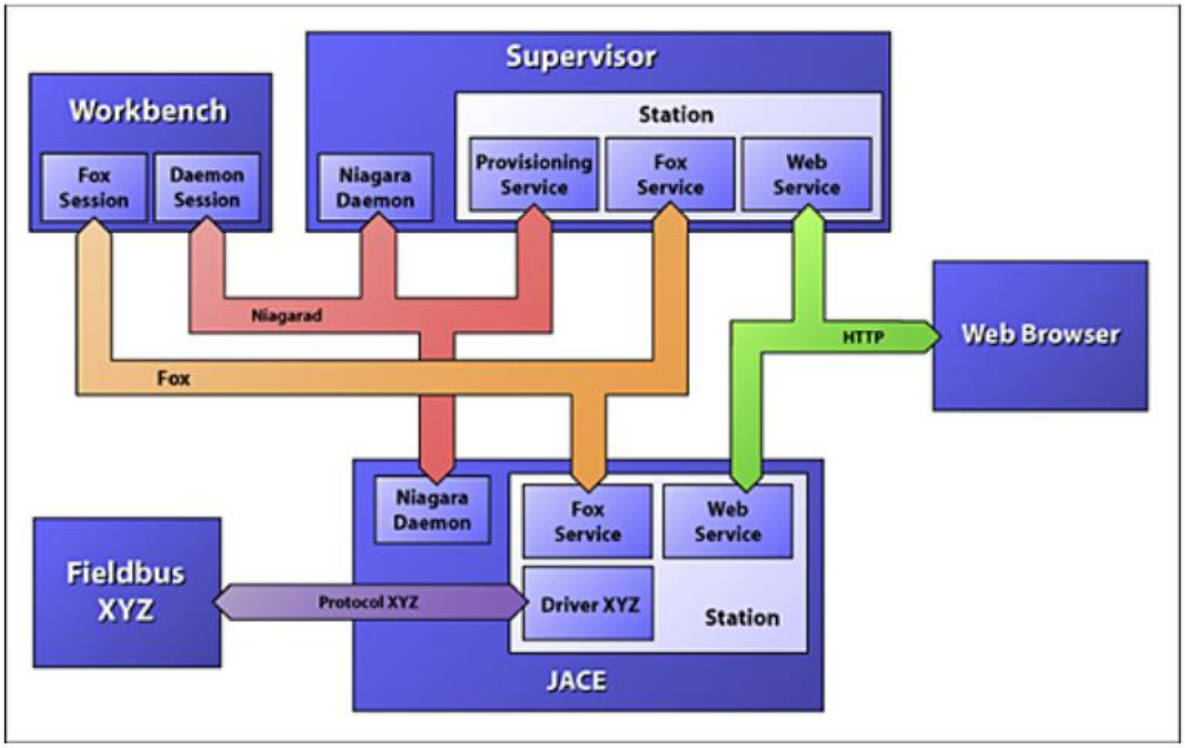

Figure 7 - Niagara Framework communication

Platform tools require a platform connection, which is different from a station connection. When connected to a NiagaraAX platform, the workbench communicates (as a client) to that host's platform daemon (also known as "**niagarad**" for Niagara daemon), a server process. Unlike a station connection, which uses the Fox protocol, a client platform connection requires Workbench, meaning it is unavailable using a standard Web browser.

A NiagaraAX host's platform daemon monitors a different TCP/IP port for client connections than does any running station (if any). By default, this is port 3011. Finally, the platform daemon uses "host-level" authentication for logon access. This means a user account and password separate from any station user account and should be considered the highest level access to that host.

4.2.2 Healthcare Protocols

To provide better threat detection, particularly for patient data exfiltration, the BTDS needs to dissect also protocols commonly used in medical devices network to exchange information of patient and medical devices. We describe here two open standard protocols widely used in HDOs: Health Level 7 (HL7) and Point of Care Testing 1A (POCT-1A).

Health Level 7

Health Level 7 is by far the most widely used network protocol for exchanging patient information in hospital networks. Its standard is developed by the HL7 Group.

The purpose of the HL7 Group is to facilitate communication in healthcare settings by providing standards for the exchange of data among healthcare computer applications. Among those standards are also network protocols, which are the focus of this section.

There are 3 protocols currently supported by the HL7 group. The standards of the protocol exist exclusively on the 7th layer (application) of the OSI model (hence "Level 7"). For lower levels, there are sometimes guidelines provided by the HL7 group, but they are not standardized. This is mostly because the HL7 messaging protocol is a very flexible protocol throughout as it needs to fit numerous varied environments.

This section focuses on HL7 version 2, which is the most widely used flavor of HL7. Besides HL7v2, the HL7 group defines HL7v3 [56] and HL7 FHIR [57].

HL7v2 has been first released in 1987. Newer versions are all backwards compatible and the newest at time of writing is HL7v2.8.2. HL7v2 is a very simple and easy to understand protocol but is very broad in specifications. The encoding uses human-readable characters (ASCII by default) and employs character delimiters to separate messages and data.

An example HL7v2 message is shown below:

```
MSH|^~\&|ZIS|1^AHospital|||199605141144||ADT^A01|20031104082400|P|2.3||
|AL|NE|||8859/15|<cr>
EVN|A01|20031104082400.0000+0100|20031104082400<cr>
PID|||""|10||Vries^Danny^D.^^de||19951202|M|||Rembrandlaan^7^Leiden^^730
1TH^""^^P||""|""||""|||||||""|""<cr>
PV1||I|3w^301^""^01|S|||100^van den
Berg^^A.S.^^""^dr|""||9||||H||||20031104082400.0000+0100<cr>
```

The payload of a HL7v2 can be divided into several parts, most of which have predefined specifications about how they should be constructed. Structure and different message parts for HL7v2 are shown in Figure 8 and Figure 9.

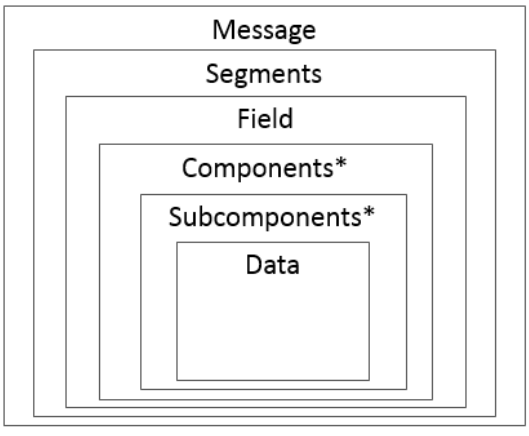

Figure 8 – Structure of a HL7v2 message

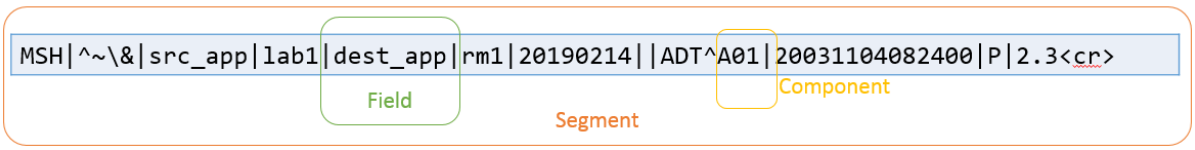

Figure 9 – HL7v2 message parts

In the following, the different message parts are described more in detail.

Messages are the main data transferred between systems (usually a single network packet). A message has defined types with a defined sequence of segments.

Segments contain multiple fields separated by a "segment terminator" character. A segment starts with a three-character literal value identifier (i.e. MSH), may be defined as required or optional, and may be repeated.

Data fields contain data or a component and subcomponents **s**eparated by a "field separator" character. **Components** contains data or subcomponents. **Subcomponents** contain only **Data**, which may have variable length and whose contents are specified by in which sequence the data fields (and possible components) are in a segment.

Most messages in HL7v2 use the MSH type messages. MSH specifies a defined message from the standard. There are many messages defined, the standard contains a complete set of messages that a HDO should need. But there might be need for special messages on a local level, it is possible to create custom messages in HL7v2. For each message there is a specification of what segments it should contain in a specific sequence.

Some of the most commonly used messages are:

- ACK – General acknowledgement
- ADT – Admit, Discharge, Transfer
- BAR – Add/change billing account
- DFT – Detailed financial transaction
- ORM – Order (Pharmacy/treatment)
- ORU – Observation result (unsolicited)
- RDE – Pharmacy/treatment encoded order
- SIU – Scheduling information unsolicited

A complete list of the message types can be found in the HL7v2 specification [58].

Point of Care Testing 1A

POCT-1A (where POCT stands for Point of Care Testing) is a standard messaging protocol for medical diagnostic devices which can carry out tests directly at the patient point of care, i.e. for instance the hospital bed, instead that in central medical laboratories. The standardization committee for POCT-1A is the National Committee for Clinical Laboratory Standards (NCCLS).

The diagnostic tests which can be carried out using POCT-1A devices are, for example, blood, glucose, urine analysis, pressure measurement and so on. Therefore, we expect that the devices using POCT-1A for exchange messages will be small portable devices which can be commonly found at the patient bed in a hospital or in the general practitioner office.

The general architecture of a POCT-1A network is depicted in Figure 10.

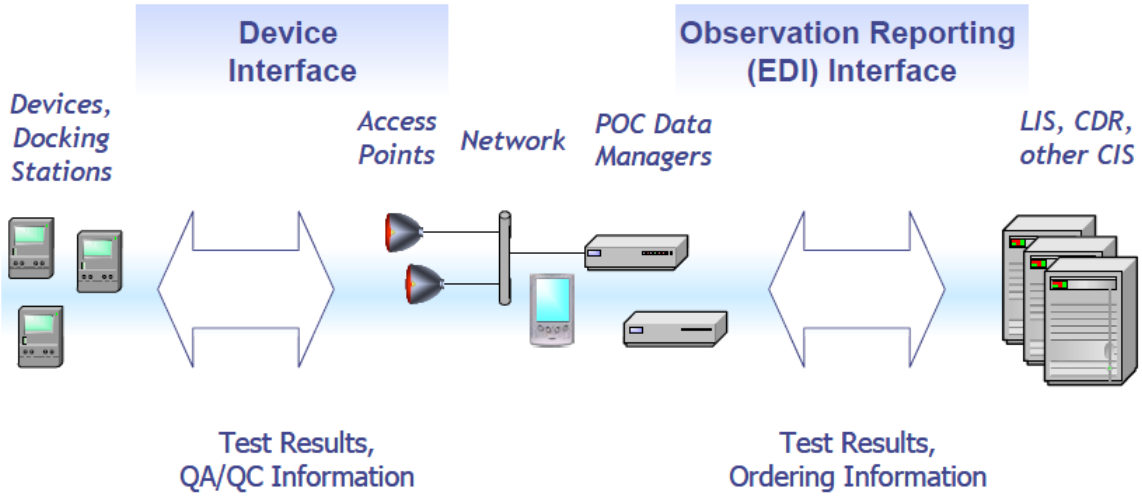

Figure 10 - POCT-1A typical network communication architecture

The components of this architecture are:

- Device, docking stations: these are called Point of Care (POC) devices in the specification and encompass all the devices which are used for diagnostics.
- POC Data Managers: these are called Observation Reviewer in the specification and they are generally IT medical workstations running Windows.
- Laboratory Information System (LIS) and other information systems: these are the storage servers used for medical information and they are not taken into account by the specification.

The POCT-1A protocol is only spoken between POC devices and observation reviewers. Furthermore, the specification gives a hint on the devices which can use POCT-1A protocol: The devices that are within the scope of this specification are hand-held devices; test modules that are part of other instrumentation (a patient monitor, for example); or small, bench-top analyzers.

POCT-1A is a purely application layer protocol which assumes the existence of a robust and reliable transport method. The transport protocol used is usually TCP. POCT-1A messages uses clear-text XML format for storing information and leverage the HL7v3 information model for encoding message data types. The POCT-1A device messaging layer (DML) standardize the communication between device and observation reviewer, which we are interested in. The standard message flow between POC devices and observation reviewers is shown in Figure 11.

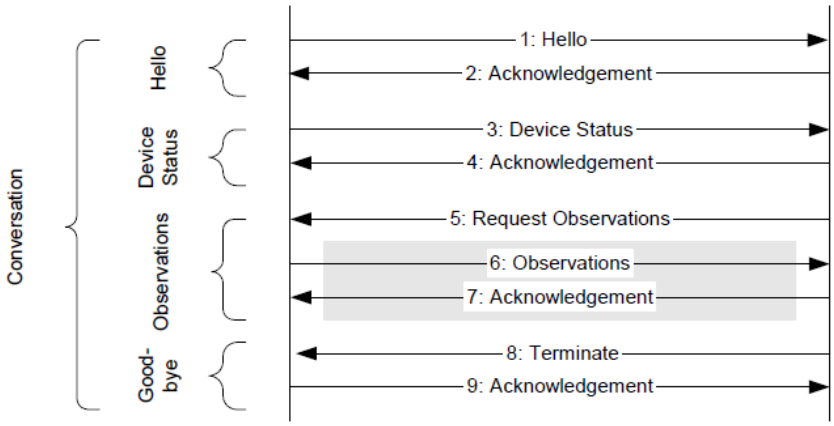

Figure 11 - POCT-1A usual message flow

In the above picture the device is on the left and the observation reviewer on the right part of the graph.

Every message uses a set of codes as root for the XML message. The message types in the following table are required to have minimal compliance with the POCT-1A specs.

Code	Name	Direction
HEL.R01	Hello initial message	C → S
DST.R01	Device status	C → S
OBS.R01	Observation information	C → S
ACK.R01	Acknowledgment	C ↔ S
REQ.R01	Request	S → C
EOT.R01	End of topic	C → S
END.R01	Terminate the conversation	C ↔ S
ESC.R01	Escape due to some unavailability	C ↔ S

Table 2 - POCT-1A required message types

4.3 Threat Detection Engines

As mentioned in Section 4.1, the BTDS is composed of several detection modules, which are the main building blocks for satisfying the requirements of Section 3. These modules are described more in detail in this Section.

4.3.1 Signature-based detection module

The signature-based module provides several pre-configured checks and controls to detect weaknesses and threats at an early stage and offers intelligence about the cause and remediation of a detected problem. The checks are conveniently divided into three categories:

- **Networking:** To detect device and network misconfigurations, such as hosts not receiving NTP responses or connectivity issues.
- **Operations:** To detect problems and threats to the building automation operations, such as malfunctioning or misbehaving devices or the use of potentially dangerous operations (e.g., restart/reset commands).
- **Security:** To detect security threats and vulnerabilities, such as the use of insecure protocols or protocol versions (e.g., TELNET or SSHv1), exploits of known vulnerabilities, Indicators of Compromise (IoCs) and user defined blacklists (e.g., blacklisted IPs).

These signatures are completely configurable, and each individual check can be enabled or disabled for the entire network or only for some of the monitored hosts in order to achieve the necessary flexibility to accommodate most of the use cases required by an HDO.

4.3.2 Anomaly-based communication detection module

The anomaly-based detection engine is used to model network communications within a local network environment, i.e. a network with a limited number of (known) hosts communicating with each other. The anomaly-based engine can model network communications by the following features that span across the network protocol stack (OSI model):

- IP addresses: the source and destination hosts. IP is assumed as the network-layer protocol.
- L4 protocol: the transport-layer protocol in use (TCP or UDP).
- L4 ports: the source and destination ports used by the transport protocol.
- L7 protocol: the application-layer protocol (e.g., BACnet, HTTP, SMB, etc.).
- L7 message groups: what application-layer messages the sender sends to the recipient (e.g., read, write, delete).

Modeling is done by means of communication rules. A communication rule defines an action to be performed by the engine when the observed network communication matches the IP addresses, L4 protocol, L4 ports, L7 protocol and L7 message groups specified in the rule.

The most common actions which can be defined are: *allow* and *alert*. If the action is *allow*, the rule defines a whitelisted communication. If the action is *alert*, the rule defines a blacklisted communication and an alert is raised when the communication is detected. The anomaly-detection module can be set in learning mode in order to automatically detect the rules from network traffic. One rule is created for each combination of source IP address, destination IP address, L4 protocol, destination L4 port and L7 protocol. When set in detecting mode the anomaly-based engine checks the network communications for a matching rule and reacts according to the specified action. The rules are checked at different stages of a network communication.

4.3.3 Malformed packet detection module

The Malformed Packet Detection module performs protocol parsing related checks, such as detecting malformed packets and packets that do not comply with the protocol specification. For example, a packet that contains an additional parameter or contains an alphabetic character instead of a numeric one is malformed. A packet that has a parameter with value 300, while the specification only allows numeric values between 0 and 255, does not comply with the protocol

specification. The module shows the status of the module and gives an overview of the enabled protocols and protocol checks.

4.3.4 Port scan detection module

The Port Scan Module is used to detect TCP port scanning activity. The Port Scan Module must detect horizontal and vertical scans from a single source or from multiple different sources (distributed scans).

In a horizontal scan, a group of IPs is scanned for a single port. In a vertical scan, one IP is scanned for multiple ports. Two classes of TCP port scanning techniques exist: TCP SYN scans and scans using Out Of State TCP packets. Both techniques work in a similar way, in that the attacker either sends a TCP SYN packet or an Out Of State packet and waits for a response from the destination host. Based on (the lack of) the response, the attacker can determine whether a scanned port is open, closed or filtered.

An Out Of State TCP packet has TCP flags set that are invalid based on the current state of the TCP stream. For example, in the normal case, a TCP stream is initiated by sending a TCP SYN packet. If instead an attacker sends a TCP ACK packet as the first packet in a stream, this is considered an Out Of State packet. Also, some invalid ag combinations such as having no flags set at all are considered Out Of State packets. The TCP specification does not specify how an implementation should respond to such invalid packets. Some implementations may ignore invalid packets while others reply in some way, giving the attacker some information about the destination's operating system and state of the destination port.

4.3.5 Man-in-the-middle detection module

The Man-in-the-middle Module is used to detect MITM attempts using various techniques. The MITM Module must be able to detect Address Resolution Protocol (ARP) poisoning, ARP port stealing, the ARP Re-ARP stage of an ARP poisoning attack, Internet Control Message Protocol (ICMP) Redirect (spoofing), Dynamic Host Configuration Protocol (DHCP) responses from unknown hosts and DHCP response spoofing techniques. MITM attacks are extremely dangerous. In most cases, the attacker is already present on the local network and can use MITM techniques to spy on communication between different hosts on the network or to alter key parts of this communication.

5 Interconnections with SAFECARE platform components

The BTDS integrates within the cyber-security solution of the SAFECARE platform. Figure 12 shows where the solution sits in the global architecture of the SAFECARE system. It depicts all the physical security solutions, cyber security solutions and integrated solutions within SAFECARE and the interconnections between them. For more information about the global architecture, refer to SAFECARE deliverable D6.1.

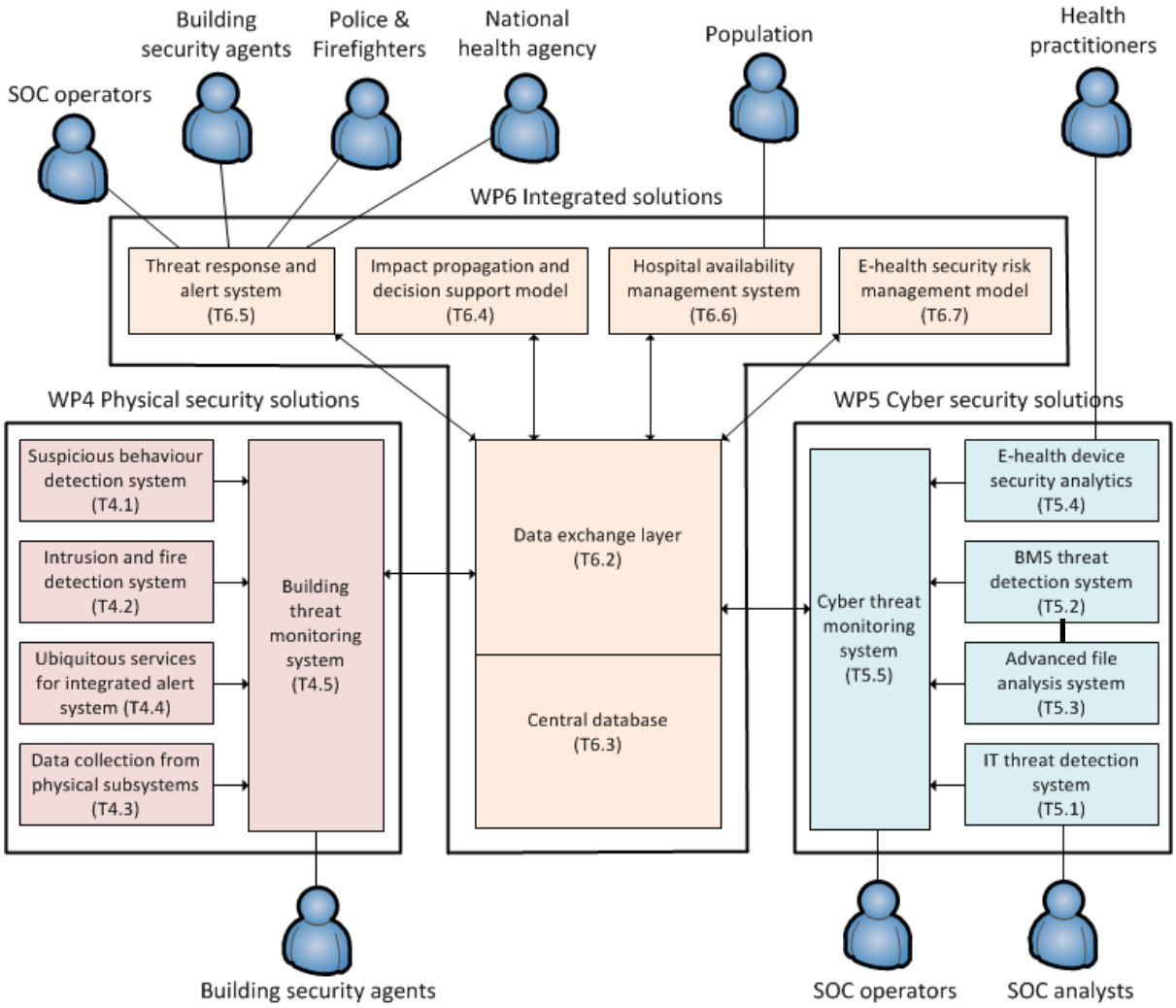

Figure 12 - SAFECARE Global Architecture

In the following Sections, we will specify how the BTDS communicates with its directly connected components in the architecture, namely the Cyber Threat Monitoring System and the Advanced File Analyzer.

5.1 Connection with Advanced Malware Analyzer

Among the functionalities of the BTDS to be developed during the SAFECARE project, there is the capability of the BMS sensor to parse and dissect files from the raw network traffic of several widely used protocols in building automation systems such as SMB.

In order to send the files to external systems, the BMS sensor dissector engine dissects the files and makes them available to the BMS monitoring interface. The monitoring interface sends the dissected files as payload of HTTPS post requests to TLS authenticated clients. An event-driven forwarding mechanism triggers a callback to send automatically any newly dissected file available in the BMS monitoring interface. This mechanism is used to send the dissected files to the Advanced File Analysis System, another component of the SAFECARE cybersecurity solution.

5.2 Connection with Cyber Threat Monitoring System

The BTDS is also directly linked with the Cyber Threat Monitoring System. The main role of the latter component is to integrate the information acquired by the different detection systems composing the SAFECARE cyber-security solution into an incident which is correlated with the physical security information and stored in the central database.

The alerts generated by the BTDS are sent to the Cyber Threat Monitoring System via the Syslog protocol. In the following, we provide an example of a Syslog message representing an alert raised when resetting a building controller using the BACnet protocol:

```
CEF:0|SAFECARE|BTDS|BACnet Device Reinitialization
Command|severity=HIGH|cat=alert alert_type=bacnet_device_reset
id=1 smac=00:0a:0a:0a dmac=00:0b:0b:0b src=192.168.1.1 src_risk=HIGH
dst=192.168.1.2 dst_risk=MEDIUM src_port=47809 dst_port=47810
l4proto=udp l7proto=bacnet module=SignatureModule timestamp= 2019-10-
25T10:34:24.461+02:00 msg={Potentially dangerous BACnet operation: a
BACnet device or operator has instructed another BACnet device to
either reboot, reset itself to an initial configuration, start/end
backup, or start/end/abort restore procedure. This operation may be
part of a regular maintenance but can also be used to carry out a
Denial of Service attack.}
```

6 Conclusion

This deliverable detailed the specification of an innovative network-based intrusion detection system that leverages in-depth protocol parsing and is specifically designed to protect healthcare building management systems from cyber-attacks. More specifically, this document highlighted the main security challenges of BMS, the requirements for the BMS-specific threat detection system, a detailed system architecture, and the interconnections with other SAFECARE components, namely the Advanced Malware Analyzer and the Cyber Threat Monitoring System.

Early testing with an incomplete prototype of the BMS sensor shows that this system is able to effectively detect both common attacking procedures and threats specifically related to the operation of the BMS network. This isachieved while limiting the number of false positive alerts sent to the Cyber Threat Monitoring System.

The next SAFECARE deliverable related to the BMS threat detection system (D5.3) will be released on M22 and will describe the implementation of the system, which will be based on the specification provided in this document.

References

[1] P. Domingues, P. Carreira, R. Vieira and W. Kastner, "Building automation systems: Concepts and technology review," *Computer Standards & Interfaces,* vol. 45, no. 1, pp. 1-12, 2016.

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

[3] Memoori, "The Collision of IT & OT is Shaping the Future of Buildings in the IoT Age," 2018. [Online]. Available: https://www.memoori.com/collision-ot-shaping-future-buildings-iot-age/.

[4] O. Gasser, Q. Scheitle, C. Denis, N. Schricker and G. Carle, "Security implications of publicly reachable building automation systems," in *IEEE Security & Privacy Worshops*, 2017.

[5] M. Fuentes and N. Huq, "Securing connected hospitals," TrendMicro, 2017.

[6] K. Zetter, "Researchers hack building control system at Google Australia office," Wired, 2013. [Online]. Available: https://www.wired.com/2013/05/googles-control-system-hacked/.

[7] SecurityLedger, "Let's get cyberphysical: Internet attack shuts off the heat in Finland,," 2016. [Online]. Available: https://securityledger.com/2016/11/lets-get-cyberphysical-ddos-attack-halts-heating-in-finland/.

[8] D. Bilefsky, "Hackers Use New Tactic at Austrian Hotel: Locking the Doors," 2017. [Online]. Available: https://www.nytimes.com/2017/01/30/world/europe/hotel-austria-bitcoin-ransom.html.

[9] D. Holmberg, "Bacnet wide area network security threat assessment," NIST, 2013.

[10] T. Mundt and P. Wickboldt, "Security in building automation systems - a first analysis," in *Proceedings of Cyber Security*, 2016.

[11] Haboob Team, "VLAN Hopping Attack," [Online]. Available: https://www.exploit-db.com/docs/english/45050-vlan-hopping-attack.pdf.

[12] O. Hersent, D. Boswarthick and O. Elloumi, The Internet of Things: Key Applications and Protocols, Wiley, 2012.

[13] K. Walker, "The impact of the internet of things on buildings," 2018. [Online]. Available: http://www.smartbuildingsmagazine.com/features/the-impact-of-the-internet-of-things-on-buildings.

[14] ISE, "Securing Hospitals: A Research Study and Blueprint," 2016. [Online]. Available: https://www.securityevaluators.com/hospitalhack/.

[15] TrapX Labs, "Anatomy of an Attack: MEDJACK (Medical Device Hijack)," 2015. [Online].

[16] S. Wendzel, J. Tonejc, J. Kaur and A. Kobekova, "Cyber Security of Smart Buildings," in *Security and Privacy in Cyber-Physical Systems*, Wiley, 2017.

[17] D. Telegraaf, "Hack met medicijnen: 'Hoe haal je het in je hoofd?!'," 2018. [Online]. Available: https://www.telegraaf.nl/nieuws/2841336/hack-met-medicijnen-hoe-haal-je-het-in-je-hoofd.

[18] B. Bowers, "How To Own a Building: Controlling the Physical World with BacNET Attack Framework," 2013. [Online]. Available: https://www.youtube.com/watch?v=d3jtmv6Y9uk.

[19] T. Brandstetter, "(in)Security in Building Automation: How to Create Dark Buildings with Light Speed," 2017. [Online]. Available: https://www.youtube.com/watch?v=PyOhwYgpGfM.

[20] B. Rios, "Owning a Building: Exploiting Access Control and Facility Management Systems," 2014. [Online]. Available: https://www.youtube.com/watch?v=wvO3puWSGgQ.

[21] S. Wendzel, V. Zwanger, M. Meier and S. Szlosarczyk, "Envisioning Smart Building Botnets," *GI Sicherheit,* 2014.

[22] Y. Mirsky, M. Guri and Y. Elovici, "HVACKer: Bridging the Air-Gap by Attacking the Air Conditioning System," 2017. [Online]. Available: https://arxiv.org/abs/1703.10454.

[23] D. Holmberg, "Using the BACnet Firewall Router," *ASHRAE Journal,* vol. 48, no. 11, pp. B10-B14, 2006.

[24] W. Granzer, F. Praus and W. Kastner, "Security in Building Automation Systems," *IEEE Transactions on Industrial Electronics,* vol. 57, no. 11, pp. 3622-3630, 2010.

[25] P. Celeda, R. Krejci and V. Krmicek, "Flow-based security issue detection in building automation and control networks," in *Proceedings of EUNICE*, 2012.

[26] Z. Pan, S. Hariri and Y. Al-Nashif, "Anomaly based intrusion detection for building automation and control networks," 2014.

[27] S. Szlosarczyk, S.Wendzel, J. Kaur and F. Schubert, "Towards suppressing attacks on and improving resilience of building automation systems - an approach exemplified using bacnet," in *GI Sicherheit*, 2014.

[28] J. Kaur, J. Tonejc, S. Wendzel and M. Meier, "Securing BACnet's Pitfalls," in *Proceedings of IFIP SEC*, 2015.

[29] M. Johnstone, M. Peacock and J. d. Hartog, "Timing attack detection on bacnet via a machine learning approach," in *Proceedings of AISM*, 2015.

[30] M. Caselli, E. Zambon, J. Amann, R. Sommer and F. Kargl, "Specification mining for intrusion detection in networked control systems," in *Proceedings of USENIX Security* , 2016.

[31] P. Morgner, S. Mattejat and Z. Benenson, "All Your Bulbs Are Belong to Us: Investigating the Current State of Security in Connected Lighting Systems," *arXiv e-prints,* 2016.

[32] J. Tonejc, S. Guttes, 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] H. Esquivel-Vargas, M. Caselli and A. Peter, "Automatic deployment of specification-based intrusion detection in the bacnet protocol," in *Proceedings of CPS-SPC*, 2017.

[34] Z. Zheng and A. Reddy, "Safeguarding building automation networks: THE-driven anomaly detector based on traffic analysis," in *Proceedings of ICCCN*, 2017.

[35] F. Mollers and C. Sorge, "Deducing user presence from inter-message intervals in home automation systems," in *Proceedings of IFIP SEC*, 2016.

[36] HIMSS, "2019 HIMSS Cybersecurity Survey," 2019. [Online]. Available: https://www.himss.org/2019-himss-cybersecurity-survey.

[37] MITRE, "APT18," [Online]. Available: https://attack.mitre.org/groups/G0026.

[38] Symantec, "New Orangeworm Attack Group Targets the Healthcare Sector in the U.S., Europe, and Asia," [Online]. Available: https://www.symantec.com/blogs/threat-intelligence/orangeworm-targets-healthcare-us-europe-asia.

[39] Symantec, "Whitefly: Espionage Group has Singapore in Its Sights," [Online]. Available: https://www.symantec.com/blogs/threat-intelligence/whitefly-espionage-singapore.

[40] FireEye, "DoubleDragon - APT41, a dual espionage and cyber crime operation," [Online]. Available: https://content.fireeye.com/apt-41/rpt-apt41.

[41] B. Rios, "Security of Medical Devices," S4x13, 2013. [Online]. Available: https://youtu.be/5rn2L45uLD8.

[42] B. Rios, "Infusion Pump Teardown," S4x16, 2016. [Online]. Available: https://youtu.be/pq9sCaoBVOw.

[43] D. Regalado, "Inside the Alaris Infusion Pump, not too much medicine, plz," DEF CON 25 IoT Village, 2017. [Online]. Available: https://youtu.be/w4sChnS4DrI.

[44] S. Hanna, R. Rolles, A. Molina-Markham, P. Poosankam, K. Fu and D. Song, "Take Two Software Updates and See Me in the Morning: The Case for Software Security Evaluations of Medical Devices," in *HealthSec*, 2011.

[45] Y. Xu, D. Tran, Y. Tian and H. Alemzadeh, "Poster: Analysis of Cyber-Security Vulnerabilities of Interconnected Medical Devices," in *Proceedings of CHASE*, 2019.

[46] A. Duggal, "Understanding HL7 2.X Standards, Pen Testing, and Defending HL7 2.X Messages," Black Hat US, 2016. [Online]. Available: https://youtu.be/MR7cH44fjrc.

[47] D. Haselhorst, "HL7 Data Interfaces in Medical Environments: Attacking and Defending the Achille's Heel of Healthcare," SANS, [Online]. Available: https://www.sans.org/reading-room/whitepapers/vpns/paper/38010.

[48] D. Haselhorst, "HL7 Data Interfaces in Medical Environments: Understanding the Fundamental Flaw in Healthcare," SANS, 2017. [Online]. Available: https://www.sans.org/reading-room/whitepapers/vpns/paper/38005.

[49] M. Bland, C. Dameff and J. Tully, "Pestilential Protocol: How Unsecure HL-7 Messages Threaten Patient Lives," Black Hat US, 2018. [Online]. Available: https://i.blackhat.com/us-18/Thu-August-9/us-18-Dameff-Pestilential-Protocol-How-Unsecure-HL7-Messages_Threaten-Patient-Lives.pdf.

[50] Y. Mirsky, T. Mahler, I. Shelef and Y. Elovici, "CT-GAN: Malicious Tampering of 3D Medical Imagery using Deep Learning," in *USENIX Security*, 2019.

[51] D. McKee, "80 to 0 in Under 5 Seconds: Falsifying a Medical Patient's Vitals," 2018. [Online]. Available: https://securingtomorrow.mcafee.com/other-blogs/mcafee-labs/80-to-0-inunder-5-seconds-falsifying-a-medical-patients-vitals/.

[52] ActivePower, "Data Center Thermal Runway," [Online]. Available: http://powertechniquesinc.com/wp-content/uploads/2015/08/Active-Power-WP-105-Data-Center-Thermal-Runaway.pdf.

[53] R. Antonello, S. Fernandes, C. Kamienski and et. al., "Deep packet inspection tools and techniques in commodity platforms: Challenges and trends," *Journal of Network and Computer Applications,* vol. 35, no. 6, 2012.

[54] International Standard Organization (ISO), "ISO/IEC 14908-1:2012," [Online]. Available: https://www.iso.org/standard/60203.html.

[55] LonMark International, [Online]. Available: www.lonmark.org.

[56] Health Level 7 Group, "HL7 v3," [Online]. Available: http://www.hl7.org/implement/standards/product_brief.cfm?product_id=186.

[57] Health Level 7 Group, "HL7 FHIR," [Online]. Available: https://www.hl7.org/fhir/overview.html.

[58] Health Level 7 Group, "HL7 v2.8.2," [Online]. Available: http://www.hl7.org/implement/standards/product_brief.cfm?product_id=403.