Journal of Computer Virology and Hacking Techniques (2024) 20:301–314
https://doi.org/10.1007/s11416-023-00479-w

ORIGINAL PAPER

Demonstration of new attacks on three healthcare network protocols in a lab environment

Guillaume Dupont[1] · Daniel dos Santos[2] · Stanislav Dashevskyi[2] · Sangavi Vijayakumar[1] · Sashaank P. Murali[1] · Elisa Costante[2] · Jerry den Hartog[1] · Sandro Etalle[1]

Received: 20 May 2022 / Accepted: 28 March 2023 / Published online: 24 July 2023

Abstract
Healthcare delivery organizations such as hospitals are complex infrastructures comprising a broad range of networked devices. They include connected medical devices which can deliver health care, support hospitals' operations, and can exchange patients' data over healthcare network protocols. Previous research has pointed out weaknesses in the implementations of some of these protocols, and demonstrated how they could be abused by malicious actors in hospitals. There are still other healthcare network protocols for which we have limited knowledge, and no security analysis can be found in the literature. This can represent an issue, as these technologies may also have vulnerabilities which could, if exploited, impact hospitals' operations and patients' data. For this reason, we investigate in this paper three healthcare protocols found in hospital networks: the *POCT1-A* and *LIS02-A* standards used by some point-of-care and laboratory devices, and the proprietary protocol *Data Export* used by some Philips patient monitors. We explain how to build a test lab to perform security research on medical devices, in which we demonstrate four attacks highlighting how the selected protocols can be abused. This research provides greater knowledge of threats relevant to healthcare delivery organizations, and helps to enhance network security monitoring capabilities such as intrusion detection systems. More specifically, signatures can be created to detect attacks on these protocols and datasets can be assembled to assist the development and testing of hospital-specific intrusion detection systems.

Keywords Healthcare · Connected medical devices · Network security

1 Introduction

Healthcare Delivery Organizations (HDO) are complex infrastructures where a broad range of information technology, operational technology, and Internet of Things (IoT) devices are increasingly interconnected [1]. Specialized IoT devices, often referred to as Internet of Medical Things (IoMT) [2], are used to support clinical care. They can, for example, deliver treatment (e.g., infusion pumps), track patients' vital signs (e.g., patient monitors), or perform analysis on samples (e.g., blood analyzers). These IoMT devices can communicate with other systems within the HDO by sending and receiving data over healthcare network protocols.

While these technologies and increased connectivity improve the efficiency and quality of care, they can also introduce threats. A growing body of security research conducted on healthcare network protocols and medical devices demonstrates the risks of attacks which can have critical consequences for patients (e.g., [3–8]). Examples include the ability to manipulate the amount of drug delivered by infusion pumps [9, 10], or the interception and modification of patients' medical data such as real-time vital signs [3] and even CT scans [11].

The network analysis we performed in [12] reveals the usage of other healthcare protocols for which, to the best of our knowledge, no security research has been performed. This represents an issue for HDO and patients as these protocols could potentially be abused in similar ways mentioned above. This situation calls for security research to be conducted on these protocols in order to gain a greater understanding of the risks these technologies may introduce.

✉ Jerry den Hartog
j.d.hartog@TUE.nl

Guillaume Dupont
g.f.c.dupont@tue.nl

[1] Eindhoven University of Technology, Eindhoven, The Netherlands

[2] Forescout Technologies B.V, Eindhoven, The Netherlands

This knowledge can contribute to enhancement of the security posture of HDO as it can be leveraged to implement and/or fine-tune security controls. More specifically, protocol security research can enable the improvement of network security monitoring solutions such as intrusion detection systems (IDS) for HDO in multiple ways. For example, new signatures for IDS can be created, giving them the ability to detect malicious activities. Additionally, one can create network traffic datasets relevant to HDO, which can be used for the testing of novel environment-specific IDS and other security tools. In fact, as shown in [13], current datasets available are often limited or ill-suited to reflect the modern threat landscape (e.g., lack of real-network attacks, or large number of deprecated threats).

Performing security research in HDO can be challenging for two reasons. First, experimentation in "live environments" may be not feasible, as it can introduce risks for HDO's operations by altering the functioning of devices. Common security testing procedures such as port scanning can crash systems that are connected to patients and deliver care [14]. Second, obtaining network traffic data to conduct research offline can be difficult since such data is likely to contain patients' information and therefore cannot be shared due to regulations. Consequently, it is essential to have dedicated labs to perform security research in safe and controlled environments, yet realistic enough to guarantee the validity and accuracy of the experimentation.

In this paper we conduct security research on three healthcare protocols and investigate how they can be abused. These protocols are the *POCT1-A* and *LIS02-A* standards used to connect point-of-care and laboratory devices with laboratory information systems (LIS), and the proprietary protocol *Data Export* which can be used to collect data from Philips patient monitors. After introducing previous work done on healthcare protocols in Sect. 2, we start by detailing our experimental setup in Sect. 3, including the objectives, methodology, the selection of targets, and explain how to build a lab containing medical devices and other systems required for our research. We then introduce in Sect. 4 our attacker model and demonstrate four attacks targeting these protocols, highlighting the potential impact to HDO's operations and patients' safety and privacy. The contribution of our work is three-fold: 1) the lab setup we present can be used as a blueprint to pursue security research on medical devices in a safe environment, 2) the demonstration of attacks highlights how attackers could exploit weaknesses in healthcare networks, and 3) security measures in HDO can be improved with the creation of new signatures and datasets to assist the development and testing of HDO-specific IDS (Data sharing not applicable to this article as no datasets were generated).

2 Related work

Some of the literature focuses on certain healthcare network protocols, including non-proprietary protocols such as Health Life Seven (HL7) or Digital Imaging and COmmunications in Medicine (DICOM), and proprietary ones like RWHAT. HL7 standards are used to exchange patient data between systems, and is considered to be "the most widely implemented standard for healthcare in the world" [15]. Several studies have shown how it can be abused in hospitals due to insecure implementation. Duggal explains in [16] how to attack HL7 2.x standards. He outlines the critical parts of HL7 messages, and demonstrates ways to abuse HL7 interfaces, including how to perform denial of service (DoS) attacks and how to fuzz HL7 services running on machines in hospitals. Haselhorst proposes similar attacks in [7]. He shows how to execute DoS attacks by exhausting the number of simultaneous connections supported by an HL7 interface in order to block further legitimate connections. Additionally, he points out the lack of authentication by default, allowing anyone on the network to craft and send HL7 messages. Dameff et al. [17] developed a tool that exploits the lack of authentication and encryption in most HL7 deployments. Their tool automates the process of performing a man-in-the-middle attack and changes laboratory results from healthy values to those indicating serious illness.

DICOM is the international standard involved in medical imaging processes, and is used for the exchange and storage of images captured by imaging modalities (e.g., CT scanners) [18]. Chantzis et al. explain in [4] how to create a custom protocol parser in Wireshark,[1] and how to develop a DICOM Service scanner for the Nmap Scripting Engine.[2] This scanner can identify DICOM Service Providers on the network, test their configuration and launch attacks. Mirsky et al. developed in [11] a tool which can intercept patients' CT scans over the network and modify them by adding or removing tumors.

McKee investigates in [3] the RWHAT proprietary protocol used by some bedside patient monitors. He demonstrates the feasibility of modifying patient's vital signs recorded by a patient monitor while being transmitted to a central monitoring system (CMS). Tampering with vital signs can lead to serious consequences for a patient, including "extended hospitalization, additional testing, and side effects from medications prescribed to control heart rhythm and/or prevent clots" [3].

[1] https://www.wireshark.org/.

[2] https://nmap.org/book/man-nse.html.

3 Experimental setup

We explain in this section the objectives of our research and the selection of protocols and medical devices. Finally, we present the design and setup of our testing lab.

3.1 Objectives & methodology

We aim at discovering vulnerabilities in a selection of healthcare protocols and medical devices, and demonstrating attacks exploiting them. More specifically, our objective is to create a number of novel network attacks which could impact HDO's operations, patients' safety and data privacy. The purpose of these attacks is to enable the creation of new IDS signatures and datasets relevant to this environment, ultimately improving network monitoring capabilities [13]. While this work follows the examples of previous healthcare security research, we focus on different protocols.

Based on our research conducted on HDO networks in [12] we identify and select healthcare protocols for which no attack has been yet published (to the best of our knowledge). We then acquire devices using those protocols. Our objective is to first understand the communication patterns between the devices and how the protocols operate. We specifically look at how the packets are formed, and what kind of data they convey, among other things. We then investigate whether these communications can be abused by, for instance, spoofing devices, intercepting packets and tampering values on the fly. Furthermore, we also want to identify if the medical devices can be remotely controlled by sending commands via these protocols, similarly to other protocols like BACnet [19].

While it is important to understand how the devices and protocols work to find weaknesses, we limit the explanation of their functioning and specifications to the amount necessary for our study. We focus here on the demonstration and consequences of the novel attacks we found, and refer the curious readers to the specifications of the devices and protocols for deeper understanding.

3.2 Targets

The protocols selected for this study are POCT1-A2 [20], LIS02-A [21] and Data Export [22]. The two first protocols are used for interconnecting point-of-care and laboratory devices with information systems such as laboratory information systems (LIS) or data management system (DMS) [20, 21]. These protocols enable the exchange of clinical results and patient data between laboratory instruments and information systems [21]. Additionally, they can be used by clinical staff to issue test orders and remote requests to devices. Data Export is a proprietary protocol which can be used by Philips bedside patient monitors to connect them to a computer [22]. It provides a means for clinical researchers to collect data such as measurements (e.g., heart rate, arterial oxygen saturation), electrocardiogram waves, patient demographic information and patient monitors' system data.

We acquire several devices that use the selected protocols. Obtaining these devices enables us to generate real data and observe "live" communications. This situation helps greatly in understanding how the protocols operate, and also to test attacks. We choose a Siemens DCA Vantage blood analyzer for the analysis of both POCT1-A2 and LIS02-A, and a Philips IntelliVue MP50 patient monitor for Data Export. The blood analyzer measures the glycaemia in patients with diabetes and detects early kidney disease, while the patient monitor is used to track in real-time vital signs of a patient.

Note: The LIS02-A standard is a revision of the former ASTM E1394-97 standard [21]. While the specifications of the Siemens DCA Vantage cover the usage of ASTM, this information applies to LIS02-A.

3.3 Healthcare lab design

We start our security research by setting up a test lab containing the aforementioned medical devices and other systems required. It enables us to experiment with attacks in a controlled and safe environment. We present below our lab by first introducing its architecture, then the tooling installed on the devices, and finally the basic functioning of the devices and protocols.

3.3.1 Architecture

The lab we build is depicted in Fig. 1. It consists of the aforementioned medical devices, as well as a collection of computers, interconnected all together via a network switch.

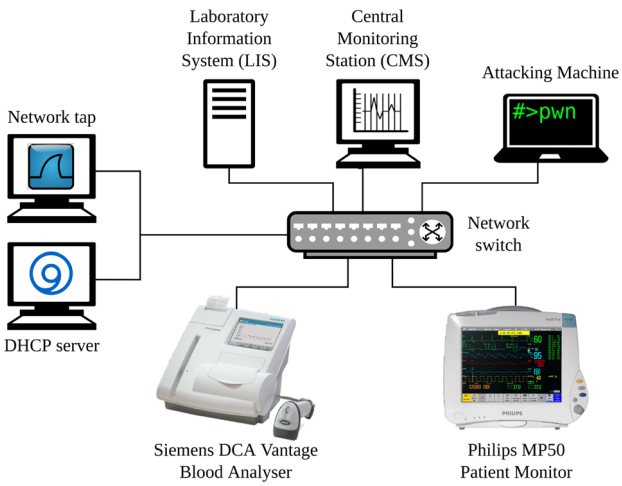

Fig. 1 Network layout of our healthcare lab

The devices require to be provisioned an IPv4 address upon boot before being able to communicate. To fulfill this requirement we add a Raspberry Pi 3 to the network and configure it as a *DHCP server*. For the purpose of analyzing protocols, it is important to be able to observe real-time network traffic as it is generated by the devices. For this reason, we configure one port of the switch as a SPAN port on which all network traffic is mirrored, and we connect a computer to it which will be used as a *network tap*.

Regarding the protocols of interest, we can generate and analyze POCT1-A2 and LIS02-A traffic by deploying a *LIS* server which communicates with the *blood analyzer* and stores test results. For Data Export, we consider in our research a scenario where this protocol is used to interconnect the *patient monitor* with a computer acting as a *central monitoring station* (this computer is further referred to as the CMS). Such monitoring stations are commonly used in hospitals to remotely display the patients' vitals coming from several monitors, allowing nurses to watch over multiple patients from a single location [3]. However, as stated in the protocol's specifications, the protocol is designed to enable data collection for clinical research purposes, and must not be used as real-time alarming system due to the risk of data loss [22] (p. 10). Moreover, the specifications also state that the Data Export interface "cannot be accessed via the Local Area Network when the monitor is connected to the Philips LAN" [22] (p. 10).

Although our architecture goes against the intended protocol design, we do so in our lab to highlight two things. First, the methodology we present can still be replicated to analyze other protocols used in HDO to interconnect patient monitors and CMS. Second, as data collected for clinical research can be used for the development of machine learning models [23], it is relevant to consider how protocol abuse could be leveraged by threat actors to perform machine learning-specific attacks such as model poisoning attacks [24]. With this in mind, we deploy in our lab a computer configured to act as a *CMS* on which we can observe the real-time readings sent by the *patient monitor*.

Finally, we connect a laptop (referred to as *Attacking Machine* in Fig. 1) to the switch to represent an attacker on the network. We elaborate on the capabilities of the attacker later in Sect. 4.

3.3.2 Software & tooling

For the lab, as depicted in Fig. 1 to be operational, we need to configure the devices and install or simulate related software and services. While the medical devices are deployed with their default configuration and the network switch is configured as described above, we install/develop the necessary tools on the other devices as follow:

- **LIS:** Since no suitable open-source solution for LIS can be found, we implement a simple LIS02 server using Python ASTM and a POCT01 server according to the communication specifications of the Siemens DCA Vantage analyzer [25]. This setup is for proof of concept only in our lab, and is not used in HDO nor recommended by Siemens.
- **CMS:** We install the IxTrend Express software[3] on the computer acting as a CMS. The program records medical signals from patient monitors for clinical research, and is selected for its support of the Data Export protocol. As mentioned above, this deployment goes against Philips' recommendations, and is only meant for proof of concept in our lab.
- **Network tap:** We install the software Wireshark[4] to capture and analyze network traffic. This tool is a powerful protocol analyzer, enabling us to inspect packets as they are passing on the network.
- **DHCP server:** We configure the Raspberry Pi 3 as DHCP server by installing Bind9.[5] The device answers to the DHCPDISCOVER messages broadcasted by the devices on the network and provides IPv4 addresses to them.
- **Attacking machine:** We install the tool Ettercap[6] for traffic redirection and packet modification, and Scapy[7] for packet crafting. Additionally, we deploy on this machine a copy of our custom LIS02 server which is used as a *rogue LIS server*. We elaborate further on how these tools are used when presenting the attacks and proofs of concept. Note: In a real situation, an attacker would likely use a broader collection of tools to guarantee the success of her objectives. However, we consider for our scenario only the tools required for the execution of the attacks we present in Sect. 4.

3.3.3 Basic functioning of the selected medical devices

We describe below how the devices in the lab interact with each other over the healthcare communication protocols we are interested in. In particular, we look at how the devices are configured to operate on the network, then how communications are established and how data is exchanged.

Siemens DCA Vantage blood analyzer:

The configuration of the device is done manually by adjusting the settings directly on the device. The relevant configuration options for our study can be found in the Eth-

[3] https://www.ixellence.com/index.php/en/home/17-default-en/products.

[4] https://www.wireshark.org/.

[5] https://gitlab.isc.org/isc-projects/bind9.

[6] https://www.ettercap-project.org/.

[7] https://scapy.net/.

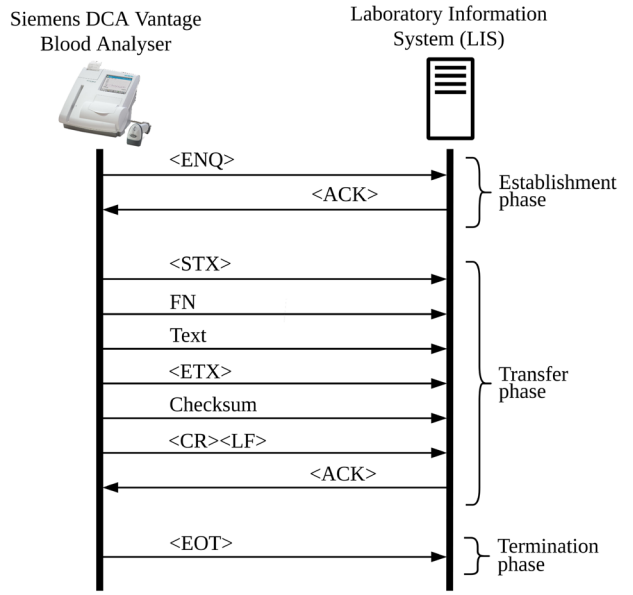

Fig. 2 Communications between the Siemens DCA Vantage and the LIS over LIS02-A protocol

ernet port settings menu. Particularly, the IP address of the blood analyzer can be set to be either automatic (provided by the DHCP server), or static (entered manually on the device). Next, we specify the IP address and port number of the LIS server we want to connect to. Finally, we select which protocol to use: the Siemens DCA Vantage can communicate with the LIS server over the LIS02-A and POCT1-A2 protocols. Below we describe the basic functioning of both these protocols.

LIS02-A: When using LIS02-A, a *session* is created every time an operator wants to transmit data from the Siemens DCA to the LIS server, according to the procedure depicted in Fig. 2. As described in the Siemens DCA's specifications [25] (p. 6), a session consists of three distinctive phases: *Establishment*, *Transfer* and *Termination* phase. In the Establishment phase, the Siemens DCA Vantage requests to establish the direction of communication by sending an inquiry character <ENC>, which is acknowledged by the LIS if it is ready to receive data. In the Transfer phase, the laboratory instrument can send messages as follow: it sends a Start of Text character <STX>, followed by the frame number FN (value from 0 to 7) and the Text corresponding to the content of the message to send. Finally, it finishes the transmission by sending a End of Text character <ETX>, a Checksum and the end of frame characters <CR><LF>. The LIS replies back with an <ACK> if the checksum value is correct. Finally, in the *termination phase*, the blood analyzer sends an End of transmission character <EOT> to the server, after which both devices return to their initial state.

Various information can be carried in LIS02-A messages: data about the laboratory instrument and the information system, patient (e.g., name, birth date, address, phone number, known diagnosis, etc.), test order (e.g., specimen ID, priority, ordering physician, etc.), and results (e.g., measurement value, unit, reference ranges, result abnormal flags). Figure 3 depicts an example of a LIS2-A message containing the result of Hemoglobin A1C test (or HbA1c test): by measuring the amount of glucose in the blood, these tests may be used to diagnose diabetes in adults [26]. LIS02-A messages consist of multiple *records* (each line represents a different record), which carry specific information.

```
H|\^&|||DCA Vantage^01.00.00.00^A123456|||||||P||20120415170532<CR>
P|1|654321|||Doe^John<CR>                    1              2
C|1|I|age^51|G<CR>   3
O|1||434^8622<CR>
R|1|^^^HbA1c|2.5|%|4.0 to 6.0|<|||||20120415180010<CR>
C|1|I|1.000^0.0 %^NGSP|G<CR>                          4
L|1|<CR>
```

Fig. 3 Example of a LIS02-A message containing the result of a HbA1c test (example inspired from [21])

In Fig. 3, some information of interest for our study is indicated by red boxes. The first record is the *Message Header Record* (H) and contains sender information including the product-code, software-version and serial number (box 1 on the figure), as well as the timestamp of message transmission following the format YYYYMMDDHHmmSS (box 2). The second record is the *Patient Information Record* (P), specifying the patient ID and name (box 3). The fifth record is the *Result Record* (R), displaying the universal test ID ("HbA1c"), the measurement ("2.5"), the unit ("%"), the reference range ("4.0 to 6.0"), and the timestamp of the beginning of the test analysis (box 4). A complete overview of LIS02-A messages can be found in [21].

POCT1-A2: The Siemens DCA Vantage can communicate with the LIS over flows of POCT1-A2 messages referred to as *conversations*, which consist of a number of *topics* exchanged. As described in the Siemens DCA's specifications [25] (p.36), two types of conversations are supported: *Basic profile*, where a conversation is initialized, data is transmitted and the conversation is terminated, and *Continuous Mode*, where the conversation remains open after initialization, allowing the blood analyzer to send status change, device events and observations (i.e., test results) as they occur.

As depicted in Fig. 4, a Basic profile conversation starts with the *Hello* topic, in which the Siemens DCA Vantage informs the LIS that it is ready to communicate. After receiving an acknowledgment of the LIS, the blood analyzer sends a *Device status* message indicating its current level of readiness (e.g., ready, busy, locked). After acknowledgment, the conversation goes to the *Observations* topic where the LIS requests observations (i.e., test results). The Siemens DCA Vantage sends them back, the LIS acknowledges the reception, and the blood analyzer finishes the transmission by sending an *End of Topic* message. The LIS then sends a direc-

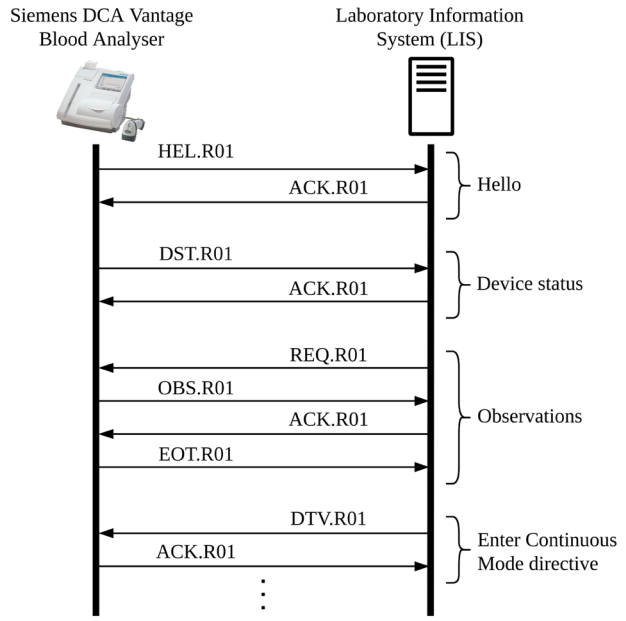

Fig. 4 Communication between the Siemens DCA Vantage and the LIS over POCT1-A2 protocol

```
<OBS.R01>
    <HDR>
        <HDR.control_id V="10003"/>
        <HDR.version_id V="POCT1"/>
        <HDR.creation_dttm V="2006-10-02T20:00:15-00:00"/>
    </HDR>
    <SVC>
        <SVC.role_cd V="OBS"/>
        <SVC.observation_dttm V="2006-10-02T18:34:20-00:00"/>
        <SVC.reason_cd V="NEW"/>
        <SVC.sequence_nbr V="333"/>                              1
        <PT>
            <PT.patient_id V="987654"/>
            <PT.name V="Jane Doe">
                <FAM V="Doe"/>
                <GIV V="Jane"/>                                  2
            </PT.name>
            <OBS>
                <OBS.observation_id V="HbA1c" SN="SIEM" SV="1.0"/>
                    <TRANSLTN V="HbA1c"/>
                </OBS.observation_id>
                <OBS.value V="2.5" U="%"/>
                <OBS.method_cd V="M"/>
                <OBS.interpretation_cd V="L"/>
                <OBS.normal lo-hi limit V="[4.0;6.0]" U="%"/>    3
            </OBS>
        </PT>
        <RGT>
            <RGT.name V="DCA HbA1c"/>
            <RGT.lot_number V="9012"/>
            <RGT.expiration_date V="2008-05-31"/>
        </RGT>
    </SVC>
</OBS.R01>
```

Fig. 5 Example of a POCT1-A2 message containing the result of a HbA1c test (similar to the test presented in Fig. 3)

tive to the device to enter the Continuous Mode: according to the device's specifications, the Siemens blood analyzer requires a Basic profile conversion to be followed by a Continuous Mode conversation [25]. From there on, the communication is maintained and a number of other topics can be used [25] (p. 37). For our study, understanding the functioning of the Basic Profile conversation is sufficient. The details of the Continuous Mode conversation can be found in [25].

In the Observations topic, there is a number of message types that are supported by the Siemens DCA Vantage, such as Request Observation Message (REQ.R01, as shown in Fig. 4), Patient Observations (OBS.R01), Device Status (DST.R01) or even Remote Command Directive (DTV.SIEM.DVCMD). The exhaustive list along with descriptions can be found in [25] (p. 34). Patient Observation messages contain the results of a patient test. Such results are retrieved from the blood analyzer at the end of a test, or when a device operator performs a test recall. An example of an HbA1c result formatted according the POCT1-A standard is depicted in Fig. 5. It represents the POCT1-A equivalent of the LIS02-A test result shown previously in Fig. 3. The POCT1-A standard uses the XML format for the exchange of messages. The relevant parts of the message for our study are highlighted in red boxes on the figure. The first box contains the Service (SVC) information, indicating that the message is a patient test results ("OBS"), and it also includes the measurement timestamp and the sample sequence number. The second box displays the patient data (PT), which are the patient ID and name. The third box contains the actual observation results (OBS): the unique test identifier ("HbA1c"), the test value and unit ("2.5%"), as well as the reference range ("[4.0;6.0]") among other things.

Philips MP50 patient monitor:

For the purpose of our study, we first need to configure the patient monitor to communicate with the computer acting as a CMS (referred to as "the computer" in this section) over the Data Export protocol. Data Export is a message-based request/response protocol, and uses the Local Area Network (LAN) interface and the standard UDP/IP transport protocol [22]. The configuration is completed as follow. After being connected to the switch and booted up, the Philips MP50 begins by requesting an IP address on the network using the BOOTP protocol. The DHCP server answers to that request by supplying an address. The patient monitor then starts multicasting *Connect Indication Event* messages in a Data Export packet over UDP to port 24005, signaling its presence on the network. These messages contain information about the patient monitor such as the serial number, product number, hardware revision, appliance software, and software revision. The Philips MP50 keeps on sending these messages until an association is established with a computer.

The computer on the network can establish an *association* with the patient monitor as described in detail in [22] (p.25). The association is created (and terminated) according to a certain sequence of messages depicted in Fig. 6.

When the computer receives a *Connect Indication* message, it can send an *Association Request* message. The Philips MP50 receives it and sends back an *Association Result* message, to either accept or refuse the association. If the association is accepted, the patient monitor sends next an *MDS Create Event Report* message, containing information

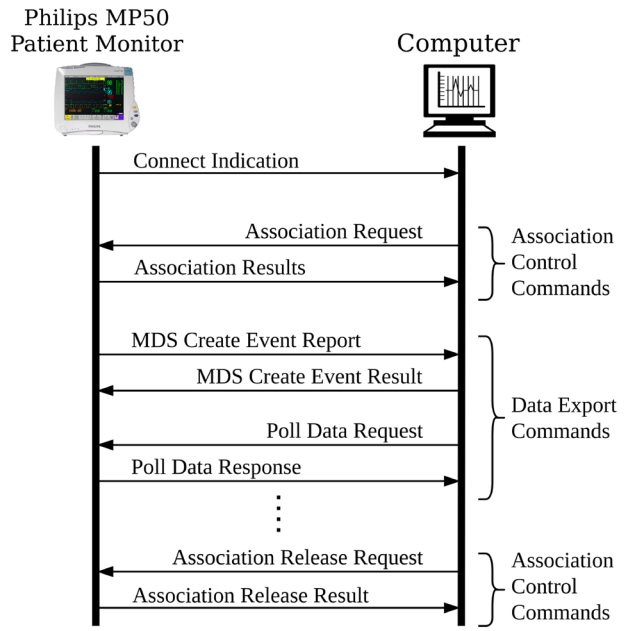

Fig. 6 Communication sequence between the Philips MP50 patient monitor and a computer to establish an association, transfer data and close the connection

about its system and its configuration. This message must be acknowledged by the computer with an *MDS Create Event Result* message. From this moment on, the computer can send *Poll Data Request* messages to the patient monitor to retrieve information such as vitals measurements, alerts (e.g., patient alarms), patient demographics or system attributes (e.g., version numbers). The Philips MP50 replies to these requests by sending *Poll Data Response* messages containing the data queried. The computer can chose to close an association by sending an *Association Release Request* message to the patient monitor, which will be acknowledged in return with an *Association Release Result* message from the patient monitor. In case of a communication issue, the Philips MP50 can send an *Association Abort* message to the computer, indicating that the association has been stopped (not shown in Fig. 6).

3.4 Remarks and communication with manufacturers

Before deploying the devices in the lab as described above, we first performed a vulnerability assessment on the two pre-owned medical devices which were obtained via an online auction site. We leveraged various sources detailing IoT device assessments such as [27] to structure our work. Starting with a function evaluation, we set up and configured the devices to operate according to their "normal specifications" in order to understand how they work. We then performed a physical inspection to evaluate the physical attack surface of the devices. Based on the results, we then attacked the exposed interfaces identified.

We spotted several issues with the DCA Vantage blood analyzer which were reported to Siemens and we worked together through the responsible disclosure process to address the following findings. The DCA Vantage is operated by the users via the restrictive user interface called the kiosk app [4]. This application limits the actions that can be executed by a regular user (e.g., a nurse), and provides an authentication mechanism to allow a privileged user (e.g., technician) to perform maintenance operations and change the system configuration. While users should not be capable of escaping from the kiosk app, we found that it is possible to break out from this application by connecting a keyboard to the exposed USB port and using specific key-stroke combinations. This allows us to gain access to the underlying OS and file system, granting us the possibility to run arbitrary code and retrieve files stored on the system. This *Improper Access Control* vulnerability was reported to Siemens and assigned the number CVE-2020-15797. We also found that the DCA Vantage contains several pieces of sensitive and confidential information. For instance, we extracted a database in the system in which we found data such as administrator password, test results and device logs. The password corresponds to the four-digit pin required to authenticate a privileged user (for the authentication mechanism mentioned above). This vulnerability, commonly referred to as *Use of Hard-coded Password*, was reported to Siemens and assigned the number CVE-2020-7590.

No vulnerability was found in the Philips IntelliVue MP50 patient monitor in the time allotted for the assessment. While the attacks on the Data Export protocol presented in this paper demonstrate the risk of (semi-)plain text protocols in general, they do not constitute an attack on the Philips device per se (i.e., they do not exploit a vulnerability present in the device). Therefore the device is not considered "vulnerable" and there is no specific vulnerability disclosure related to these attacks. Our results were not formally discussed with contacts at Philips. The attacks in this paper have limited impact, especially if the device's interface and protocol are used as intended and documented by the manufacturer [22].

Finally, our findings also reveal potential privacy issues. The test results stored in the database we retrieved are presumably from a previous owner. Although we did not investigate further the prevalence of sensitive data in second-hand medical devices, this shows that special care should be taken to data confidentiality when disposing of medical devices. There should be a process in place to securely decommission devices, as well as functions implemented in medical devices to allow for the secure erasure of data.

4 Experimentation

In this section we demonstrate in our lab four attacks on the selected protocols.

4.1 Attacker model

The attacker model we choose for our analysis is one of an attacker who wants to interfere and disrupt HDO's operations and patient care by launching cyber attacks against medical devices and HDO infrastructure. The attacker can be driven by various motivations, such as social and economical destabilization, profit through ransom, or sheer "amusement".

This attacker model assumes the threat actor to be already inside the network, and can interact with the targeted devices. We discussed in our previous work [12] how networks in HDO can be found improperly segmented, making our model tangible. The consequence is that sensitive devices (and data) can be reachable not only by legitimate systems used by medical staff, but also by adversaries who have access to the network [11]. We consider an attacker having access to the network in one of the two following ways: via remote access or via physical access. Remote access to an HDO's network can be established, for example, through the compromise of a device reachable over the Internet or via phishing [28]. Physical access can be obtained by connecting a rogue device to an exposed network socket [11]. The ease of obtaining such access is common in hospitals since sockets are often used in patient rooms to connect medical devices that interface directly with patients [29].

An attacker having access to the network can leverage a man-in-the-middle (MITM) position. This consists in the attacker's ability to redirect communication flows to her, and therefore allowing her to be *logically* in between devices. Such position enables her to intercept, read and/or modify packets on the network. Additionally, we consider an attacker who has already performed network reconnaissance activities [30] and identified the target devices and their respective IP addresses and ports.

4.2 Attacks

We implement four attacks using the protocols POCT1-A, LIS02-A and Philips Data Export, as described in Table 1. For each attack, we outline in the table its type and impact on the confidentiality, integrity and availability (CIA) of assets according to the STRIDE threat classification model [31]. This popular model is commonly used to identify weaknesses in technology [4] and has also been successfully applied to cyber physical systems [32, 33]. Following the STRIDE framework, the four attacks we demonstrate in this section are classified along the following three types:

- **Spoofing**: The attacker pretends to be a legitimate system.
- **Tampering**: The attacker violates the integrity of data by modifying it.
- **Denial of service (DOS)**: The attacker violates the availability of data and system by disrupting system operation.

We present each attack below. We first outline the goal and impact, and then introduce the technical background relevant to the understanding of the attack before explaining the scenario of the attack. We finish with a demonstration of our proof of concept where we implement the aforementioned scenario and execute the attack in our lab.

4.2.1 Attack 1: retrieving test results

Goal and impact: This attack consists in the interception of blood test results from the DCA Vantage analyzer. It impacts the confidentiality of patient data, where the attacker obtains a number of sensitive information contained in the POCT1-A packet. This attack highlights the risks to patients' data privacy, and can be reproduced with any two devices communicating over an unencrypted and unauthenticated POCT1-A channel.

Technical background:

Let us first consider an example of a blood test result stored in the Siemens DCA Vantage as shown in Fig. 7.

It displays the result of a Hemoglobin A1C test (or HbA1c test) of 66 mmol/mol, which could be indicative of diabetes. This result, dating from the 26th of July 2018, was already stored in the device when we received it (see Remarks above). We use this test result throughout our experimentation and attacks below. For sanitary reasons, we did not conduct blood tests ourselves in our lab. The interested reader can refer to

Table 1 Attacks on medical device protocols implemented in the lab

Attack	Type	Impact on CIA	Protocol	Description
1	Spoofing	Confidentiality	POCT1-A2	Retrieve test results stored in blood analyzer
2	Tampering	Integrity	LIS02-A	Change test results sent from blood analyzer to LIS
3	Denial of service	Availability	Data Export	Abort connection between patient monitor and CMS
4	Tampering	Integrity	Data Export	Change pulse rate readings displayed on CMS

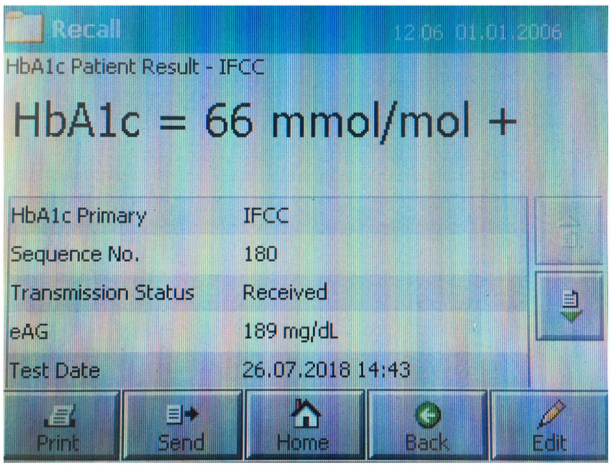

Fig. 7 HbA1c test result displayed on the screen of the Siemens DCA Vantage

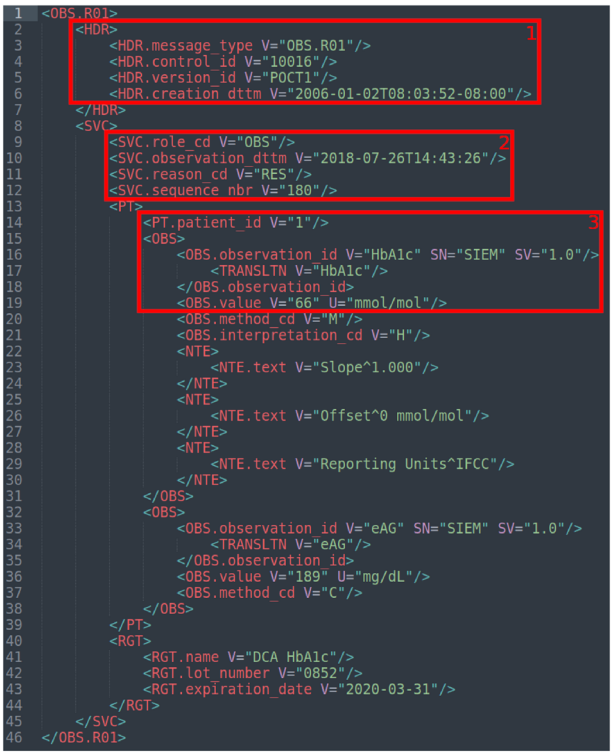

Fig. 8 Details of the POCT1-A packet containing the HbA1c test results sent from the Siemens DCA Vantage to the LIS. The relevant parts of the message for our study are highlighted in red boxes. Box 1 is the header of the message which specifies, among other things, the message type ("OBS.R01" for Patient Observations message), a control ID which uniquely identifies the message, and a timestamp corresponding to the date and time at which the message was sent. Box 2 highlights the Service section of the message, as introduced before. Box 3 contains the patient ID and the result value for the HbA1c test

the DCA Vantage Operator's manual [34] to learn how these tests are practically performed in an HDO setting.

When the operator (e.g., a nurse or a lab analyst) chooses to send a test result to the LIS, a POCT01-A packet containing the result is generated and sent over an established connection between the DCA Vantage and LIS.

We execute the transmission of the aforementioned test (pressing the "Send" button on the Siemens DCA's screen shown in Fig. 7). By using our network tap, we capture the packet and retrieve the payload in which we find the XML-formatted test result, as shown in Fig. 8. The content of the payload is somewhat similar to the example shown in Fig. 5. As expected, the data and test results contained in the payload are the same as the ones displayed on the blood analyzer (see Fig. 7).

Attack scenario: An attacker can retrieve test results in one of these two ways. The first way consists in passively intercepting POCT1-A packets as they are transmitted on the network. Since the packets are sent in clear text, the attacker can simply sniff the network traffic and examine the content. The limitation of this approach is that the attacker has to wait for an operator to manually initiate the transmission of the results to the LIS.

The second way an attacker can retrieve the test results is by sending a specific command via POCT1-A to the blood analyzer (or any other POCT1-A devices). Knowing that a LIS server can send commands via POCT1-A to a device, an attacker can hijack the communications between a POCT1-A device and a legitimate LIS server (e.g., via ARP cache poisoning) and spoof the server. She can then send a limited set of remote commands to the target device. Depending on the commands supported by the device, the attacker will be able to, for example, force the device to send all pending test results to the LIS server (which she will intercept), or update the list of device administrators for example. These commands can be useful for the attacker depending on the motivations and/or objectives.

A malicious actor can easily perform such an attack in a hospital by connecting a small device like a Raspberry Pi embedding a rogue LIS server to the network via an exposed network socket. A similar attack targeting the DICOM protocol has been successfully demonstrated in a real hospital setting in [11].

Proof of concept: We demonstrate in our lab the second scenario described above by performing the following steps. From the attacking machine, we first run our rogue LIS server and perform an ARP cache poisoning attack with Ettercap to spoof the legitimate LIS server. This forces the DCA Vantage to communicate with our rogue server. Following the device's specifications [25], our server is configured to respond to the blood analyzer's "Hello message" (HEL.R01) with an acknowledgment message (ACK.R01). Our server then requests pending tests results by sending a "Request message" (REQ.R01) to the target device. The DCA vantage replies by sending the test results. After a short conversation

```
00000000: 0231 487c 5c5e 267c 7c7c 4443 4120 5641  .1H|\^&|||DCA VA
00000010: 4e54 4147 455e 3034 2e30 342e 3030 2e30  NTAGE^04.04.00.0
00000020: 305e 5330 3133 3032 337c 7c7c 7c7c 7c7c  0^S013023|||||||
00000030: 507c 7c32 3030 3630 3130 3230 3833 3532  P||2006010208352
00000040: 310d 507c 310d 4f7c 317c 7c31 3830 5e30  1.P|1.O|1||180^0
00000050: 3835 327c 7c7c 7c7c 7c7c 7c7c 7c7c 7c7c  852|||||||||||||
00000060: 7c7c 7c7c 7c7c 7c7c 7c43 0d52 7c31 7c5e  |||||||||C.R|1|^
00000070: 5e5e 4862 4131 637c 3636 7c6d 6d6f 6c2f  ^^HbA1c|66|mmol/
00000080: 6d6f 6c7c 7c48 7c7c 437c 7c7c 3230 3138  mol||H||C|||2018
00000090: 3037 3236 3134 3433 3134 0d43 7c31 7c49  0726144314.C|1|I
000000a0: 7c31 2e30 3030 5e30 206d 6d6f 6c2f 6d6f  |1.000^0 mmol/mo
000000b0: 6c5e 4946 4343 5e31 3839 206d 672f 644c  l^IFCC^189 mg/dL
000000c0: 7c47 0d4c 7c31 7c4e 0d03 3032 0d0a       |G.L|1|N..02..
```

Fig. 9 Details of the LIS02 packet containing the HbA1c test result sent from the Siemens DCA Vantage to the LIS. The format of the packet is somewhat similar to the example given above in Fig. 3. We find relevant information for our study highlighted in red such as the Message Header record, the result value, and the checksum ("02") (Color figure online)

```
----------------------------
RECEIVED!
----------------------------
HEADER: ['H', [[None], [None, '&']], None, None, ['DCA VANTAGE', '04.04.00.00', 'S0113023'],
 None, None, None, None, None, None, 'P', None, '2006010208352']

PATIENT: ['P', '1']

ORDER: ['O', '1', None, ['180', '0852']], None, None, None, None, None, None, None, None, No
ne, None, None, None, None, None, None, None, None, None, None, None, None, None, 'C']

RESULT: ['R', '1', [None, None, None, 'HbA1c'], '66', 'mmol/mol', None, 'H', None, 'C', None
, None, '20180726144314']

COMMENT: ['C', '1', 'I', ['1.000', '0 mmol/mol', 'IFCC', '189 mg/dL', 'G']]

TERMINATOR: ['L', '1', 'N']
```

Fig. 10 The test result containing a value of 66 mmol/mol is received by the LIS server

sequence (detailed in [25]) a continuous conversation mode is established between the DCA Vantage and our rogue device. In this mode, all further test results will be sent directly to our rogue LIS server. Moreover, the DCA Vantage will accept a limited set of commands from the server, such as to update the list of the device's operators (OPL.R01).

4.2.2 Attack 2: changing test results

Goal and impact: The goal of this attack is to tamper with a test result being sent from the DCA Vantage analyzer to the LIS over the LIS02-A protocol. With such ability, an attacker can modify healthy results to abnormal ones, or the other way around. This impacts the integrity of the patient data, which could ultimately affect the patient's health. This attack can be reproduced with any two devices communicating over unencrypted and unauthenticated LIS02-A.

Technical background: When the operator chooses to send a test result to the LIS via the LIS02-A protocol, a packet such as the one shown in Fig. 9 is generated and sent over the network. When the LIS server receives the packet, it displays the results, as shown in Fig. 10 and stores them internally.

Attack scenario: Let us consider a scenario where the attacker wants to tamper with this data flow and send incorrect test results to the LIS. The procedure to implement this attack consists in the following steps. The attacker first executes a MITM attack to redirect the communication flow to her. When the test result is being sent by the laboratory device, she can intercept and drop the original packet. The attacker

```
----------------------------
RECEIVED!
----------------------------
HEADER: ['H', [[None], [None, '&']], None, None, ['DCA VANTAGE', '04.04.00.00', 'S0113023'],
 None, None, None, None, None, None, 'P', None, '2006010208357']

PATIENT: ['P', '1']

ORDER: ['O', '1', None, ['180', '0852']], None, None, None, None, None, None, None, None, No
ne, None, None, None, None, None, None, None, None, None, None, None, None, None, 'C']

RESULT: ['R', '1', [None, None, None, 'HbA1c'], '41', 'mmol/mol', None, 'H', None, 'C', None
, None, '20180726144314']

COMMENT: ['C', '1', 'I', ['1.000', '0 mmol/mol', 'IFCC', '189 mg/dL', 'G']]

TERMINATOR: ['L', '1', 'N']
```

Fig. 11 Due to the tampering attack, the test result value received by the LIS server is 41 mmol/mol, instead of the 66 mmol/mol value sent by the blood analyzer

then crafts a new packet with a modified test result, and adjusts the checksum value by computing a new one before sending the packet. Finally, the crafted packet is received and stored by the LIS.

Proof of concept: We demonstrate the aforementioned scenario by implementing an attack in which we change the result of an HbA1c test which could be indicative of diabetes (66 mmol/mol) to a normal result (41 mmol/mol). We use the test result depicted in Fig. 7.

We follow the procedure outlined above by using Ettercap to establish a MITM position and create a custom filter that modifies test results and checksum accordingly. An example of filter creation is provided on the Ettercap's GitHub repository.[8]

We begin the attack by running Ettercap from the attacking machine and hijacking the connection between the DCA Vantage and the LIS server. Next, we execute the test result transmission from the blood analyzer using the test result already stored in the device (depicted in Fig. 7). In a normal situation, when the analyzer sends the test result (value of 66 mmol/mol), the LIS would receive the packet as shown in Fig. 10. However, after launching our attack, the Ettercap filter we developed intercepts and modifies the packet by setting the test result to the value of our choice (41 mmol/mol) and adjusts the checksum. The crafted packet is then sent to the LIS server and we can observe the success of the attack upon reception, as displayed in Fig. 11.

4.2.3 Attack 3: disconnecting a patient monitor

Goal and impact: The goal of this attack is to close an ongoing Data Export association between the patient monitor and the computer acting as a CMS (recall that this setup is for proof of concept only, as explained in Sect. 3.3). In a (hypothetical) hospital environment, the staff remotely monitoring a patient loses the real-time information about vital readings. As a DoS attack, the availability of the patient monitor's data is impacted. This attack could be launched on multiple patient

[8] https://github.com/Ettercap/ettercap/blob/master/share/etter.filter.examples.

monitors which could result in the disruption of HDO operations, ultimately leading to delays in care with potential consequences to patients' health.

Technical background: The Data Export protocol supports the transmission of commands. As described in its specifications [22], there is a number of commands that can be sent, allowing the monitor and a computer to interact with each other in various ways. One of the commands is the *Association Abort* command. A patient monitor can send this message in the case of communication problems to close the association [22].

Attack scenario: An attacker can leverage this command to terminate the association between a patient monitor and the CMS, effectively causing a denial of service. She can spoof the targeted monitor and send an Association Abort message to the CMS. To do so, the attacker crafts a packet containing the Association Abort message as payload according to the specifications [22] (p.337). The payload consists of the following bytes:

```
0x19 0x2e 0x11 0x01 0x03 0xc1 0x29 0xa0
0x80 0xa0 0x80 0x30 0x80 0x02 0x01 0x01
0x06 0x02 0x51 0x01 0x00 0x00 0x00 0x00
0x61 0x80 0x30 0x80 0x02 0x01 0x01 0xa0
0x80 0x64 0x80 0x80 0x01 0x01 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
```

The attacker can then set the source IP address to the one of the targeted patient monitor and can finally send the packet to the right port of the CMS (24105 by default). In case of another port being used instead, the attacker would first have to sniff the traffic to observe over which port the CMS and the monitor are communicating. Upon reception of this packet, the association between the CMS and the patient monitor will be closed until manually re-established. By repeatedly sending Association Abort commands, the attacker can prevent effective use of the Data Export protocol.

Proof of concept: We implement this attack in our lab by performing the following steps. Using Scapy, we create a UDP packet containing the aforementioned payload. We then set the packet's source IP address and source port to the patient monitor's values in order to spoof it. Finally, we set the destination IP address to the CMS's, and the destination port to 24105, the default port for Data Export protocol.

We send the packet from the attacking machine and we observe the result on the screen of the CMS, as shown in Fig. 12. Upon packet reception, the CMS displays an error message informing that the patient monitor has closed the association. No data can be received any longer by the CMS, and the association has to be manually re-established.

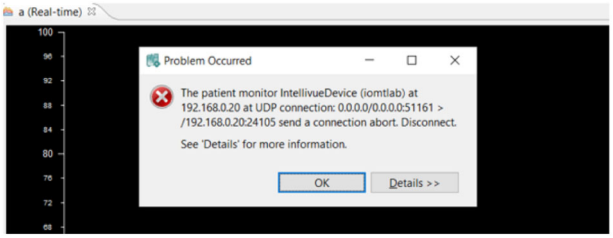

Fig. 12 CMS displaying the result of an Association Abort message

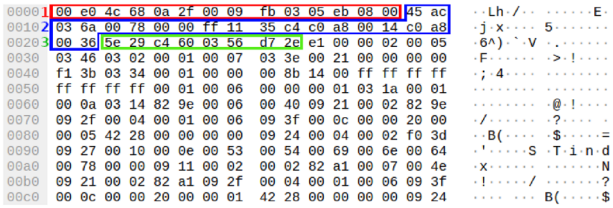

Fig. 13 Example of a Data Export packet sent by the Philips MP50 patient monitor to the CMS captured with Wireshark. Box 1 (in red) represents the Ethernet envelop, box 2 (in blue) the IPv4 envelop, box 3 (in green) the UDP envelop, and the remaining of the packet corresponds to the payload in the Data Export format. The packet is 888 bytes long (the remaining of the payload is truncated) (Color figure online)

4.2.4 Attack 4: changing a patient's vital readings

Goal and impact: The goal of this attack is to tamper with the vital readings sent from the patient monitor to the computer acting as a CMS over the Data Export protocol (recall that this setup is for proof of concept only, as explained in Sect. 3.3). In a (hypothetical) hospital environment, the personnel monitoring a patient remotely sees incorrect readings, unbeknownst to her. Similarly to Attack 2, tampering with the data conveyed in healthcare protocol packets violates the integrity of a patient's health information, which could ultimately lead to adverse effects on the patient's health.

Technical background: The patient monitor captures patient's vitals such as pulse, blood pressure and oxygen saturation through sensors connected to the patient's body. This information is displayed on the device's screen (see Fig. 15 as example), and can be exported over the Data Export protocol for clinical research purposes. As outlined in Sect. 3.3, we use in our lab this protocol to interconnect the patient monitor with a computer acting as a CMS. The patient monitor encodes the data according to the protocol specifications [22]. An example of a Data Export packet sent by the monitor to the CMS is depicted in Fig. 13.

While certain clear-text key words can give an idea about the type of data present in the packet, one cannot directly understand the information in the payload. The protocol being a binary protocol, understanding the payload format requires a bit of reverse-engineering. The publication of the protocol specifications makes this task easier, as we show below.

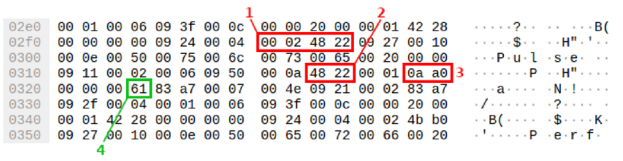

Fig. 14 Data Export packet transmitted from the patient monitor and captured with Wireshark. It shows the patient's pulse rate in the payload of the packet. The bytes related to the pulse rate as found in the protocol specifications are highlighted in red (boxes 1, 2 and 3). The actual encoded patient's pulse rate can be seen in green (box 4) (Color figure online)

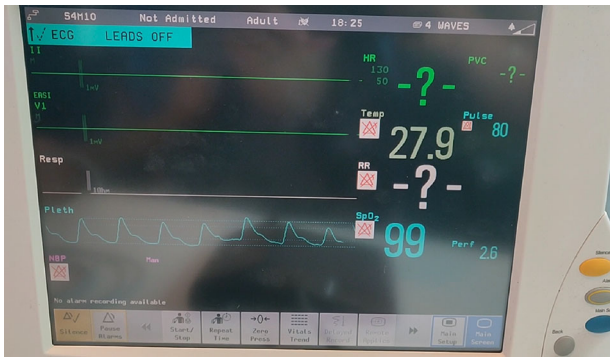

Fig. 15 Patient vitals and other information shown on the display of the Philips MP50 patient monitor. The pulse rate reading (80 beats per minute) is displayed on the right-hand side of the screen. The pulse is measured by the finger plethysmograph that is attached to one of our fingers, simulating an actual patient being monitored. Some vitals are missing (value set to –?–) because the sensors are not connected to the patient monitor

Attack scenario: An attacker wants to modify the pulse rate of a patient to 0 beat per minute to simulate a patient "flat-lining". Such attack could trigger an immediate emergency response from the staff and, when executed on multiple patient monitors, could be disruptive. This attack is achieved by modifying specific bytes of the Data Export packets as they are sent from the monitor to the CMS.

To execute this scenario, an attacker can follow a similar procedure as shown with Attack 2: she can use Ettercap to establish a MITM position after having created a filter that captures and replaces the real-time vital readings in Data Export packets to a value of 0 beat per minute. The challenge in this context is to first identify where and how the pulse rate is encoded in a packet, which requires some reverse engineering.

Note that, depending on the motivations, the attacker could also change the pulse rate values to any arbitrary values, or even change other vital readings with the same procedure, such as blood pressure and oxygen saturation, as listed in the Data Export specifications [22].

Proof of concept: To demonstrate this attack, we proceed with the three following steps. In the first step, we want to identify at which offset the pulse readings are located in the packets. To do so, we need to understand how the pulse rate is collected and processed by the patient monitor. In our lab setting, the pulse rate is measured via a finger plethysmograph connected to the monitor. We find in the Data Export documentation the information related to the pulse rate recorded by the plethysmograph and how this information is encoded [22] (p.118). In particular, we see the encoding of the label (`0x00024822`), the observed value (`0x4822`) and the unit (`0x0AA0`). With this information at hand, we can search in Data Export traffic for packets with these byte values in the payload.

We generate and capture traffic by using our patient monitor's plethysmograph connected to one of our fingers. Using Wireshark, we find the packets sent by the patient monitor containing bytes with the values related to the pulse data (see the bytes highlighted in Fig. 14) as identified in the protocol specifications.

Moreover, by comparing the pulse value displayed on the patient monitor with the values displayed on the CMS, we identify the two bytes encoding the actual pulse rate value measured by the monitor, which are located 6 bytes after the pulse's unit (`0x0a 0xa0`). In the example given in Fig. 14, the value is 0x61 in hexadecimal, which translates in decimal into a pulse rate of 97 beats per minute. Having identified where the pulse value is located in the packet, we can now calculate the offset of the byte we want to change: offset 0x323.

In the second step, we create an Ettercap filter to capture and modify the packets containing patients' pulse readings. Similarly to Attack 2 presented before, Ettercap is used to establish a MITM position and intercept the packets of interest: the ones having a payload containing the values `0x00 0x02 0x48 0x22` (the pulse label), `0x48 0x22` (the observed value) and `0x0A 0xA0` (the unit). We then modify the pulse value to the desired value and forward it to the CMS to display this information.

We can then execute the attack as follow. First, we attach the plethysmograph to one of our fingers to generate data and simulate a patient being monitored. In Fig. 15 is depicted the reading displayed on the patient monitor, which is a normal pulse of 80 beats per minute. We then execute the attack. The result can be observed in Fig. 16, as seen by hospital staff on the CMS. After the execution of the attack, the pulse suddenly drops from a legitimate range oscillating between 70 and 80 beats per minute to 0 until the attack is stopped.

5 Conclusion

In this article we investigate three healthcare protocols found in HDO. A number of weaknesses are found and we demon-

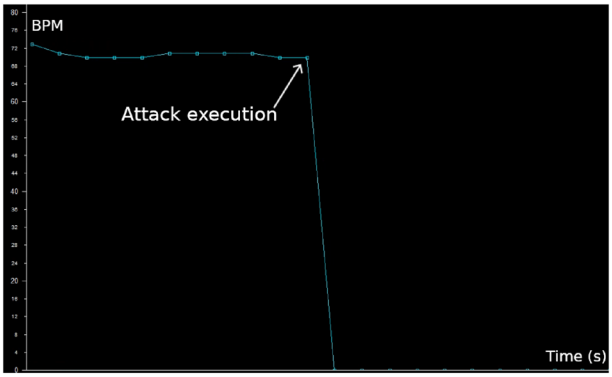

Fig. 16 Flat-lining spoofing attack observed on the CMS: before the attack execution, the pulse values oscillate between 70 and 80 beats per minute, corresponding to the actual pulse readings. After execution, the pulse value drops to 0 beat per minutes as long as the attack is executed

strate four attacks exploiting them in our lab. These practical scenarios highlight new risks to HDO and patients by violating the confidentiality, the integrity and the availability of data transmitted between medical devices and other systems. The state of the network segmentation in healthcare networks that we observed in our previous analysis [12] lets us assume that many HDO would be exposed to such threats. It is therefore necessary to protect these environments by designing network monitoring tools suited to address the threats specific to HDO, as they can impact greatly hospital's operations and patients' safety and privacy.

The research presented in this article contributes to such design. It can improve security measures such as IDS by issuing signatures for newly found vulnerabilities. Additionally, new attack datasets can be assembled to test the signatures and support the implementation of novel IDS tailor-made for HDO. Finally, our methodology and blueprint for designing a healthcare lab allows researchers to pursue vulnerability research efforts for the healthcare industry in a safe environment.

Declarations

Conflict of interest The authors have no competing interests to declare that are relevant to the content of this article. The authors have no relevant financial or non-financial interests to disclose.

References

1. O'Brien, G., Edwards, S., Littlefield, K., McNab, N., Wang, S., Zheng, K.: Securing Wireless Infusion Pumps in Healthcare Delivery Organizations. Special Publication (NIST SP), National Institute of Standards and Technology, Gaithersburg (2018). https://doi.org/10.6028/NIST.SP.1800-8
2. Gatouillat, A., Badr, Y., Massot, B., Sejdic, E.: Internet of medical things: a review of recent contributions dealing with cyber-physical systems in medicine. IEEE Internet Things J. **5**(5), 3810–3822 (2018). https://doi.org/10.1109/JIOT.2018.2849014
3. McKee, D.: 80 to 0 in Under 5 Seconds: Falsifying a Medical Patient's Vitals. https://bit.ly/3MDb70P. Accessed 12 Apr 2023 (2018)
4. Chantzis, F., Stais, I., Calderon, P., Deirmentzoglou, E., Woods, B.: Practical IoT Hacking. No Starch Press, San Francisco (2021)
5. Kramer, D.B., Baker, M., Ransford, B., Molina-Markham, A., Stewart, Q., Fu, K.: Security and privacy qualities of medical devices: an analysis of FDA postmarket surveillance. PLoS ONE (2012). https://doi.org/10.1371/journal.pone.0040200
6. Taylor, C.R., Venkatasubramanian, K., Shue, C.A.: Understanding the security of interoperable medical devices using attack graphs. In: 3rd International Conference on High Confidence Networked Systems. HiCoNS '14, pp. 31–40. Association for Computing Machinery, New York (2014). https://doi.org/10.1145/2566468.2566482
7. Haselhorst, D.: HL7 Data Interfaces in Medical Environments: Attacking and Defending the Achille's Heel of Healthcare. Technical report, SANS (2017)
8. Rushanan, M., Rubin, A.D., Kune, D.F., Swanson, C.M.: SoK: security and privacy in implantable medical devices and body area networks. In: IEEE Symposium on Security and Privacy, pp. 524–539 (2014). https://doi.org/10.1109/SP.2014.40
9. Donovan, F.: Wireless Infusion Pumps Could Increase Cybersecurity Vulnerability. https://bit.ly/3AVHTCD. Accessed 12 Apr 2023 (2018)
10. McKee, D., Laulheret, P.: McAfee Enterprise ATR Uncovers Vulnerabilities in Globally Used B. Braun Infusion Pump. https://bit.ly/38fdQeY. Accessed 12 Apr 2023 (2021)
11. Mirsky, Y., Mahler, T., Shelef, I., Elovici, Y.: CT-GAN: Malicious tampering of 3D medical imagery using deep learning. In: 28th USENIX Security Symposium (USENIX Security 19), pp. 461–478. USENIX Association, Santa Clara (2019)
12. Dupont, G., dos Santos, D.R., Costante, E., den Hartog, J., Etalle, S.: A matter of life and death: analyzing the security of healthcare networks. In: Hölbl, M., Rannenberg, K., Welzer, T. (eds.) ICT Systems Security and Privacy Protection, pp. 355–369. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58201-2_24
13. Hindy, H., Brosset, D., Bayne, E., Seeam, A., Tachtatzis, C., Atkinson, R., Bellekens, X.: A Taxonomy of Network Threats and the Effect of Current Datasets on Intrusion Detection Systems. arXiv preprint arXiv:1806.03517 (2020)
14. Florek, C.: Medical Device Security, Part 2: How to Give Medical Devices a Security Checkup. https://bit.ly/3s7Hlp4. Accessed 12 Apr 2023 (2019)
15. Health Level Seven International: HL7 Messaging Standard Version 2.9. https://bit.ly/3PdtyYD. Accessed 12 Apr 2023 (2019)
16. Duggal, A.: Understanding HL7 2.X Standards, Pen Testing, and Defending HL7 2.X Messages. Black Hat US 2016. https://youtu.be/MR7cH44fjrc (2016)

17. Dameff, C., Bland, M., Levchenko, K., Tully, J.: Pestilential Protocol: How Unsecure HL7 Messages Threaten Patient Lives. Black Hat US 2018. https://youtu.be/66x3vfac8rA (2018)
18. NEMA: DICOM PS3.1 2022b—Introduction and Overview. https://bit.ly/3Nch8yF. Accessed 12 Apr 2023 (2022)
19. Peacock, M., Johnstone, M.N., Valli, C.: An exploration of some security issues within the BACnet protocol. In: Mori, P., Furnell, S., Camp, O. (eds.) Information Systems Security and Privacy, pp. 252–272. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93354-2_12
20. Dunka, L.J., et al.: POCT01-A2—Point-of-Care Connectivity, 2nd edn. Standard, Clinical and Laboratory Standards Institute (CLSI), Wayne (2006)
21. Mountain, P.J., et al.: LIS02-A2—Specification for Transferring Information Between Clinical Laboratory Instruments and Information Systems, 2nd edn. Standard, Clinical and Laboratory Standards Institute (CLSI), Wayne (2004)
22. Philips: Data Export Interface Programming Guide. https://www.documents.philips.com/doclib/enc/fetch/2000/4504/577242/577243/577247/582636/582882/X2%2C_MP%2C_MX_&_FM_Series_Rel._L.0_Data_Export_Interface_Program._Guide_4535_645_88011_(ENG).pdf. Accessed 12 Apr 2023 (2015)
23. Ngiam, K.Y., Khor, W.: Big data and machine learning algorithms for health-care delivery. Lancet Oncol **20**(5), 262–273 (2019)
24. Schwarzschild, A., Goldblum, M., Gupta, A., Dickerson, J.P., Goldstein, T.: Just how toxic is data poisoning? A unified benchmark for backdoor and data poisoning attacks. In: International Conference on Machine Learning, pp. 9389–9398. PMLR (2021)
25. Siemens Healthcare Diagnostics: DCA Vantage Analyzer Host Computer Communications Link. https://bit.ly/3HqAkWU. Accessed 12 Apr 2023 (2011)
26. MedlinePlus: Hemoglobin A1C (HbA1c) Test. https://bit.ly/3up6FJV. Accessed 12 Apr 2023 (2021)
27. Heiland, D.: IoT Security Testing Methodology. https://bit.ly/3umYVZ5. Accessed 12 Apr 2023 (2017)
28. HIMSS: 2019 HIMSS Cybersecurity Survey. https://bit.ly/34v2q4u. Accessed 12 Apr 2023 (2019)
29. ISE: Securing Hospitals: A Research Study and Blueprint. https://bit.ly/3GkFp1v. Accessed 12 Apr 2023 (2016)
30. Achleitner, S., La Porta, T., McDaniel, P., Sugrim, S., Krishnamurthy, S.V., Chadha, R.: Cyber deception: virtual networks to defend insider reconnaissance. In: ACM 8th CCS International Workshop on Managing Insider Security Threats, pp. 57–68 (2016). https://doi.org/10.1145/2995959.2995962
31. Shevchenko, N., Chick, T.A., O'Riordan, P., Scanlon, T.P., Woody, C.: Threat Modeling: a Summary of Available Methods. Technical report, Carnegie Mellon University Software Engineering Institute Pittsburgh United-States (2018)
32. Karahasanovic, A., Kleberger, P., Almgren, M.: Adapting threat modeling methods for the automotive industry. In: 15th ESCAR Conference, pp. 1–10 (2017)
33. Martins, G., Bhatia, S., Koutsoukos, X., Stouffer, K., Tang, C., Candell, R.: Towards a systematic threat modeling approach for cyber-physical systems. In: Resilience Week (RWS), pp. 1–6. https://doi.org/10.1109/RWEEK.2015.7287428. IEEE (2015)
34. Siemens Healthcare Diagnostics: DCA Vantage Operator's Guide. https://bit.ly/3Holjoo. Accessed 12 Apr 2023 (2012)