Hardening Infusion Pump Communication Software for Medical Device Cybersecurity

Robert Smigielski

About the Author

Robert Smigielski is senior embedded software designer at B. Braun Medical Inc. in Allentown, PA. Email: Robert.Smigielski@bbraun.com

Cybersecurity attacks on medical devices pose serious concerns and challenges in the medical community. Such unauthorized access may result in changes to a protected health information, infusion therapy, and device function. For example, wireless infusion pumps transmit and receive electronic data across a variety of information technology (IT) networks and communicate patient data and therapies to and from electronic medical records (EMR) systems. Industry has been working on ways to prevent cyberattacks on infusion pumps, which could impact healthcare delivery organization (HDO) computer networks and, ultimately, quality of care and patient safety. Methods to prevent cyberattacks on infusion pumps include implementing technological improvements designed to enhance patient care.

The National Cybersecurity Center of Excellence (NCCoE) at the National Institute of Standards and Technology (NIST) has analyzed the cybersecurity risk factors associated with the infusion pump ecosystem. In NIST Special Publication 1800-8, *Securing Wireless Infusion Pumps in Healthcare Delivery Organizations*, NCCoE utilized a questionnaire-based risk assessment to develop implementation strategies for HDOs to use standards-based, commercially available cybersecurity technologies to better protect the infusion pump ecosystem, including patient information and drug library dosing limits.[1]

Infusion pumps are designed with internal computer systems that require routine updates to maintain software security. IT offers "hardening" as an accepted method to prevent access to systems' critical information or functionality.[2] This article describes the importance of software hardening as a preferred method of protection for medical infusion pumps and emphasizes the importance of testing and documentation during protocol development, which ensures that the hardening methodology is valid and reliable. In keeping with B. Braun's goal of sharing expertise, this article will also describe our testing protocol process to encourage collaboration with our industry partners and health systems.

What Are the Dangers of Cyberattacks?

HDOs, patients, regulators, and medical device manufacturers are understandably concerned about securing devices against intrusion and guaranteeing device function. Cybersecurity includes the technologies, processes, and practices designed to protect networks, computers, programs, and data from attack, damage, and unauthorized access by creating methods to analyze and protect communication systems. Faceless bad guys, or "bad actors," as they're known to IT, seek to either attack or capture communication systems, seizing opportunities to steal personal information, plant exploitation programs to attack other devices, lock down billing or personal information for ransom, or steal security-related information for later use.[3,4]

How Can Devices Be Protected?

How can medical device manufacturers create and maintain devices to meet the healthcare needs of patients while protecting them from people who would attempt to compromise medical devices and IT infrastructures? The use case story is a reaction of medical devices to specific and general attempts to corrupt control features and/or data communicated by devices. To protect the data, developers should be aware of the motivations, goals, and methods of potential hackers.

Dealing with Known Threats

Medical device engineers must define the features susceptible to exploitation by probing devices for all possible points of control, creating a list of software attack points, and searching each point to generate responses from the devices. These attack points are a combination of known software features and multiple possible ways to exploit that feature to gain information, disable, or take control of the device.

How Are Device Manufacturers Informed of Security Concerns?

To inform and educate users about threats, NIST produces and distributes a publicly available list of known software vulnerabilities called the National Vulnerability Database. These vulnerabilities are tagged with information that includes software product manufacturers or owners, software item names, and software versions. These vulnerabilities are updated daily and maintained by NIST online.

Medical device developers and testers use the software vulnerability list information to determine if their medical device could be vulnerable to historic or newly discovered issues. Medical device manufacturers can use the list provided by NIST to search for vulnerabilities applicable to their device. The manufacturer keeps a record of its specific software bill of materials, which is the key input to searching the NIST software vulnerability list. The results of the search are further refined by the device manufacturer to determine which vulnerabilities apply and which cannot have an effect on the device.

Medical device engineers must define the features susceptible to exploitation by probing devices for all possible points of control, creating a list of software attack points, and searching each point to generate responses from the device.

Addressing Security Concerns

The methods described in this article include quality control checks and approved methods to test the features of a device with respect to verifying that specific software vulnerabilities are mitigated. These documented steps describe how engineers can test for a specific, directed vulnerability and achieve a specific expected result. The documentation follows the process of testing software- or firmware-enabled features delivered as a functional part of the medical device. If the test reveals that the device is vulnerable, then the test is a failure, and the resulting information is sent to the team to analyze the root cause. If the test is successful, the feature was not adversely affected, and the device continues to operate as expected, then the test has passed. These tests are performed to address cybersecurity vulnerabilities in the end product to protect the customer and patient from security risks.

The Food and Drug Administration, medical device manufacturers, researchers, and industry are addressing ways to improve and maintain the cybersecurity of medical devices throughout device life cycles. One promising process for securing systems is hardening.[4,5]

Preventing Vulnerabilities

Preventing security issues can be one way to mitigate risk by the device manufacturer. Scientific and research-backed processes demonstrate that system-level hardening can improve the cybersecurity of medical devices. NIST 800-123 *Guide to General Server Security*, defines hardening as "configuring a host's operating system (OS) and applications to reduce the host's security weaknesses."[2] Commercial computer server devices are typically viewed as black box systems running in data centers to provide specific services to other computing devices or people. Examples of servers include email servers, file servers, and database servers. However, a server for a software-enabled system listens for incoming requests for data and responds with answers. For security purposes, these same servers undergo commercial processes to prevent cybersecurity issues from being present in the delivered system.

Software input/output communication ports and software-functional interfaces can fail in the presence of unexpected or malformed inputs, a concept termed "fuzz."

Improving Medical Device Security through Hardening

Infusion pumps include features that operate as servers since they must respond to requests for information from the traditional server computers deployed in the HDO network. Because the infusion pump executes some server software features, the infusion pump software must undergo risk analysis and mitigation similar to a traditional server device. Hardening is an industry recognized process by which a computing device is configured to perform a specific set of functions, transforming a computer from a general-purpose device (e.g., a laptop or desktop computer) into a device that focuses on a specific job. Because a computer system can be configured to provide specific services, the following aspects of hardening may be applied to reduce risks associated with unprotected computer software and hardware interfaces in an infusion pump[1]:

- Disabling unused or unnecessary communication ports and services
- Changing manufacturer default administrative passwords
- Securing any remote access points
- Ensuring firmware version is up-to-date

How can medical device manufacturers capture these goals with specific instructions and verify that software communication ports are protected during the life cycle of the product? Also, how can each software and firmware update applied during the lifetime of the device undergo this testing to verify the system remains within a known risk state? The answer is to apply the disciplines of device fuzz testing, device penetration testing, source code review, and methods to verify error rejection.

Verifying Device Hardening with Fuzz Testing

Software input/output communication ports and software-functional interfaces can fail in the presence of unexpected or malformed inputs, a concept termed "fuzz." Fuzz consists of bad data that are provided to the device. This can happen when a software feature encounters real-world conditions but the data presented to the computing device are not as expected by the software designers. Such situations occur when other devices or humans provide unexpected data to the devices.

For example, consider a situation in which a malevolent actor targets an infusion pump with a denial-of-service attack aimed at a software-controlled data port used to receive a drug library. The network used by the infusion pump is maliciously filled with garbage data in an attempt to cripple the network's data stream. Next, the culprit device is disabled and removed. The customer expects that the infusion pump will continue to operate normally since it ignored all the invalid data. This is a scenario of successful operation in the presence of fuzz data.

Fuzz data can also occur when an attacker deliberately attempts to gain access to the system to take control. A system must be robust to avoid such attacks. The Institute of Electrical and Electronics Engineers defines robustness as "the degree to which a system can function correctly in the presence of invalid inputs or stressful environmental conditions."[4] A system that functions properly despite the unpredictable is essential in the medical systems world. B. Braun uses fuzz testing techniques to test infusion pumps. During validation, the fuzz testing protocol is used to send sets of bad data to the data communication and control paths in order to expose inappropriate or faulty behavior. The devices are considered to have passed the test only if they respond by rejecting the data and avoid behaving unpredictably (i.e., crashing or locking up) (B. Braun internal document).

Devices are considered to have passed the [fuzz] test only if they respond by rejecting the data and avoid behaving unpredictably (i.e., crashing or locking up).

Verifying the Source Code for Issues

Hackers can take advantage of memory leaks and invalid use of memory to gain control of a computer system. Using knowledge of a system's software, they can exploit existing known memory leaks, inject special instructions, force the leak, and possibly take control of the device. This is only one way to hack a device.

Static analysis is a method used to find these invalid memory usage mechanisms. In the case of a B. Braun Space Infusion Pump, the application cppcheck (an off-the-shelf, open-source program) performs a static analysis of the source code that comprises the pump's application set. The goal of this process is to expose problems introduced in the written source code, which consists of millions of lines of control data. Cppcheck results are then analyzed to verify that all indicated issues have been resolved. The tests are executed with each software release to verify that the system has been corrected.

Operating systems are often configured with many communication ports in fully open states to help test the new installation of the computer. The hardening process verifies that the least number of ports are available to maintain a state of cybersecurity.

Verifying the Operating System for Open Ports

OSs provide a feature of logical ports used as communication channels for sending and receiving data (e.g., data used to configure a pump, communicate with the pump drug library server, other well-defined interfaces in EMR integration). The open-source program nmap is used to probe and analyze any OS's logical ports to verify whether the port is open for communication. A hacker with knowledge of a system's software can exploit an open port with a direct attack to gain control of the device. The nmap program is part of the fuzz testing protocol to verify which ports are open for use and those that are not open for use. This testing protocol is critical for limiting the logical communication ports to the minimum necessary for proper function. An analogy is building a rather strange house that includes a front door and a back door but no windows. To walk into the house, you must use the front or the back door. With only two ways to get inside of the house, the next level of protection can be focused on securing the only ways to get inside.

Computer OSs are often configured with many communication ports in fully open states to help test the new installation of the computer. The hardening process verifies that the least number of ports are available to maintain a state of cybersecurity.

Verifying that Invalid Data Is Rejected

The fuzzing technique includes conducting special tests that change each known data field in a drug library into a known invalid value. Then the data, both valid and invalid, are communicated to the infusion pump in a quality assurance test environment. The test determines the acceptance of any invalid drug libraries. After every field in a drug library is altered and tested for an expected rejection, a known error-free drug library is transmitted to the pump. The test then verifies whether the drug library has been accepted. This final step confirms whether the mix of valid and invalid data has corrupted the drug library manager software and enabled the acceptance of valid data even after presentation of invalid data (B. Braun internal document).

The B. Braun Infusomat Space Infusion Pump

The B. Braun Infusomat Space Pump communication software relies on a wireless hardware module that connects to a Wi-Fi computer network. The infusion pump uses this communication channel to send and receive data to the B. Braun software applications, including the Space OnlineSuite for drug library management and Space DoseTrac and Space DoseLink for reporting, documentation, and EMR integration. Following an approved protocol helps verify that the communication path is protected against invalid access and invalid data coming across the communication channel. The testing protocol verifies that the communication software does not crash or lock up when invalid data are introduced. The protocol also tests and verifies that the unused software ports in the communication subsystem are closed. These features are critical to verify the hardening of the Wi-Fi communication module used by the infusion pump (B. Braun internal document).

Open-source tools (e.g., nmap and cppcheck) are used to interrogate the SpaceCom communication subsystem, present known invalid data, and analyze the resulting behavior. These tools are used to probe each and every logical software communication port managed by the OS. This follows a "keep-it-simple" philosophy in the sense that open communication ports are in use and all other ports are disabled (B. Braun internal document).

Conclusion

A proactive approach is necessary to identify ways that the device manufacturer can improve cybersecurity to prevent access to devices and system software. The approach described in this article can be used to identify, neutralize, and lock out unauthorized access, thereby permitting only approved access. This approach includes:

- Hardening to conduct a device cybersecurity risk assessment.
- Researching and monitoring the NIST common vulnerabilities and exposures list to learn about cybersecurity issues and threats in the OS and applications.
- Configuring system software to manage access for software processes.
- Disabling unused software services and communication ports. ■

References

1. **National Institute of Standards Technology.** *Securing Wireless Infusion Pumps In Healthcare Delivery Organizations NIST Special Publication 1800-8.* Gaithersburg, MD: U.S. Department of Commerce, Computer Security Division, Information Technology Laboratory; 2017.
2. **National Institute of Standards Technology.** *Guide to General Server Security Special Publication 800-123.* Gaithersburg, MD: U.S. Department of Commerce, Computer Security Division, Information Technology Laboratory; 2008.
3. **Food and Drug Administration.** Postmarket Management of Cybersecurity in Medical Devices Draft Guidance for Industry and Food and Drug Administration Staff. Available at: www.fda.gov/downloads/medicaldevices/deviceregulationandguidance/guidancedocuments/ucm482022.pdf. Accessed Oct. 1, 2017.
4. **Codenomicon, Medical Device Innovation, Safety & Security Consortium.** MDISS Technical White Paper Series Fuzz Testing: Improving Medical Device Quality and Safety. Available at: https://fuzzinginfo.files.wordpress.com/2012/12/codenomicon-mdiss-fuzz-framework-16.pdf. Accessed Oct. 1, 2017.
5. **Food and Drug Administration.** Guidance for Industry—Cybersecurity for Networked Medical Devices Containing Off-the-Shelf (OTS) Software. Available at: www.fda.gov/MedicalDevices/DeviceRegulationandGuidance/GuidanceDocuments/ucm077812.htm. Accessed Oct. 1, 2017.