The 10 most common security vulnerabilities in medical devices

Security Testing Medical Devices (Software and Hardware)[1]

The European Medical Device Regulation (EU) 2017/745 (MDR) and the In Vitro Diagnostic Regulation (EU) 2017/746 (IvDR) regulate the testing under security aspects of software, firmware, microcode and mobile apps for controlling medical devices (here generally referred to as software) such as pacemakers, X-ray machines, tomographs, ultrasound devices and also patient management software as well as networks etc. The ordinances are currently scheduled to come into force gradually by May 2020 (medical devices) and by May 2022 (in-vitro diagnostics). These regulations correspond to regulations in the international area (USA, Canada, China, Japan etc.), so that the security-related IT requirements and the requirements for technical data protection measures are comparable worldwide – such as MDCG 2019-16 and the „Leitfaden zur Nutzung des MDS2 aus 2019: Sicherheit von Medizinprodukten“ of the Alliance for Cyber-Security.

Medical equipment and systems in practices and hospitals are mostly networked and connected - more or less directly - to the Internet, so that IT security measures are absolutely necessary to protect against unauthorized access and modification of (patient) data. Similarly, it must be ensured that the software controlling the devices is not changed - starting with the boot process.

In particular, the objectives of confidentiality, integrity and availability of the data and systems (known to computer scientists) are to be achieved. The state of the art (ISO 27034) is a security testing process that checks the software for security-relevant errors - security gaps - during development. Such a process contains the following 6 methods:

- Security Requirements Analysis - Risk Analysis
- Threat modeling for the analysis of the security architecture
- Static source code analysis - code reading of the source code
- Penetration testing of the executable code
- Dynamic Analysis - Fuzzing of the executable code
- Conformity testing to identify backdoors and cover channels

Each method is necessary because it helps identify different types of vulnerabilities.

Based on the experience with security tests since 2016, the following are the 10 most common and serious security vulnerabilities and challenges in medical devices:

- **Privacy**. Insufficiently protected patient data: Sensitive data must be protected against unauthorized access and manipulation using appropriate technical measures. The data must also be protected against physical access and data transmission by means of encryption.
- **Authentication**. Inadequate authentication before device access: To ensure that only authorized persons access the medical device (and the data), authentication must be performed before any interaction with the software or device.
- Generic/weak **default configurations**: The choice of default passwords for devices or applications must be based on strong, device-specific passwords. Identical passwords for devices that are sold many times over massively reduce the security level.
- **Outdated software** versions. The entire system must be considered. In addition to self-developed applications, the latest updates must also be rolled out to the operating system as well as to used libraries and third-party software.
- **Update process**. Missing or insufficient signature verification: To ensure the authenticity and integrity of a software update and to prevent unauthorized changes, a signature verification must be performed before the update is executed.
- **Best Practices**. For encryption, password storage, etc., best practice methods should be used instead of developing your own solutions.
- **Too narrow a focus**. Functions obtained from third party vendors must also be security checked. For example, this also applies to HL7/DICOM protocol implementations, which are typically integrated in the form of third-party libraries.
- **Secure Coding** Standards. For example, when programming in C/C++, corresponding best practices must be implemented. For example, "banned functions" should not be used, as they are susceptible to attacks such as buffer overflows.
- **Never Trust User Input**. Regardless of whether data is provided over the network via HL7/DICOM or locally, user data must be validated before processing.
- **Least Privilege**. Applications may only be given those privileges that are indispensable for the operation of the system. In the event of a successful attack, the extent of damage can thus be limited.

In all cases, the respective manufacturer has - as evidenced by a repeat test - eliminated the identified security gaps so that the software, firmware, app or microcode etc. can be considered secure according to the state of the art.

[1] Lars Ubbenhorst und Prof. Dr. Hartmut Pohl, softScheck GmbH