CIS 607 Computer Security in the Physical World Week 6:
Implantable Medical Device Security

Scribed by Hannah Pruse
February 18, 2012

Introduction

On February 15th, Paul Elliot presented a talk on the security of Implantable Medical Devices (IMDs). The presentation focused on the paper titled “Pacemakers and Implantable Cardiac Defibrillators: Software Radio Attacks and Zero-Power Defenses” by Halperin et al. that was featured in the proceedings of the 2008 IEEE Conference. A similar paper, titled “Hacking Medical Devices for Fun and Insulin: Breaking the Human SCADA System” was also included in the talk.

Paul indicated his presentation would remain lighthearted, due to the solemn nature of the subject at hand. He related the story of how, as a child, he discovered that one of this relatives used a pacemaker by circulating a shock pen around a family gathering and subsequently receiving reprimand about the danger of his actions.

What are IMDs?

Implantable medical devices (IMDs) are small apparatuses that can be inserted into the human body to perform a medical task, such as monitoring vitals, delivering drugs, and administering emergency care. Almost 25 million people in the United States alone depend on such devices to treat and prevent serious conditions such as diabetes, Parkinson’s disease, and cardiac arrhythmia.

Paul’s presentation focused on implantable cardiac defibrillators (ICDs) that are used in patients with severe heart disorders. The two main functions of these devices are to administer therapies for irregular heartbeats through pacing and defibrillation, and to communicate patient data to healthcare professionals through an external programmer device. Wireless technology was introduced to these devices in 2003-2004, allowing easy access to information within the device and reducing the need for numerous follow-up surgeries.

Another necessary function of the device is testing. When an ICD is initially placed in a patient, its life-saving capabilities need to be tested with a V-fibrillation test. To do so, the device delivers shocks in such a way to place the heart in an irregular rhythm, then applies treatments to revert the heart to its normal functioning.

Problems and Initial Takeaway

It is known that no piece of software is perfect or completely accurate. Paul demonstrated how this principle remains true within medical software by showing an incident with arrhythmia tracking software that seemingly reversed time within its logbook. Further examples of medical software and device failure included software recalls, and the fatal Therac-25 incident, in which discrepancies between a radiation treatment device’s screen and its internal readings resulted in the death and injury of several patients.

Paul explained the pressure for medical software developers to quickly release their products before sufficient quality testing is performed, as these devices are vital in supporting and improving the quality of life for many people. Furthermore, the legal system protects developers after their products are approved by the FDA, leading to issues for consumers.

The initial takeaway at this point is that, as with any software, it is difficult to prevent unintentional problems and bugs in medical devices and software. A proposed solution would be to improve security by taking on an adversarial perspective when testing and designing products. Such a method is shown to be effective from the Tylenol scare in
1982, which resulted in laws on tamper-evident packaging. If such attacks were considered before product release, many problems could be resolved.

ICD Exploits

From the main functionalities of the ICD, it becomes obvious how an adversary may want to exploit the device. Simply by having the ability to induce irregular heart rhythms, the device can cause serious problems, should it be compromised. An attacker may also wish to eavesdrop on the medical telemetry that the ICD issues.

Three types of adversaries are examined in this presentation:

a. An insider with a commercial programmer, who is the most dangerous type of attacker

b. A passive adversary, who wishes to eavesdrop on ICD communications

c. An active adversary, who can send commands to the ICD

The passive and active adversaries can utilize a universal software radio peripheral (USRP) for their attacks. The USRP is linked to a daughterboard with GNURadio that sends and receives radio signals. The USRP also has an oscilloscope to detect wireless frequencies and can convert RF data to PC-readable data.

Successful Attacks

To accomplish any type of an attack, the adversary must decode the language used by the ICD. Three steps are necessary:

a. Find the correct frequency

b. Determine the modulation scheme used by both the ICD and the programmer

c. Demodulate and decode the communications to binary

Through reverse engineering, the paper authors determined the modulation schemes of each device and discovered that all the communications were transmitted in the clear. Paul showed a short video describing how a group of students working under the paper authors’ supervision were able to easily translate the device communications to meaningful binary data.

With this information, the authors were able to perform eavesdropping attacks, in which they could obtain medical telemetry from the ICD. Another exploit of telemetry, which was just speculated by the authors, was a DoS attack by forcing the ICD into a state of constant wireless communication.

The authors also deployed replay attacks by sending pre-recorded commands from a commercial programmer to the ICD via the software radio. These attacks enable an untrusted third party to determine a device’s existence, modify the patient data it contained, and alter the type and time of therapies given. With replays, the V-fibrillation test that is usually performed after surgery could be replicated on command.

Proposed Defenses

The goals of the proposed defenses are to prevent attacks and ensure patient privacy. However, security in medical devices must take into consideration a set of significant constraints to protect the patient. Added security and privacy should not interfere with the device’s ability to perform life-sustaining functions in emergency situations. For example, while the use of cryptography may protect the privacy of ICD communications, the increased computation needed may significantly reduce the battery life of the device. Furthermore, key loss could prove incredibly dangerous and possibly fatal in emergency situations.

To mitigate the security risks, while keeping in mind the constraints of the problem, the authors proposed a series of zero-power defenses that draw no power from the main battery. Their solution utilized the WISPer, a small device with an attached piezo element that is implanted along with the ICD. The suggested defenses also keep the patient in the loop when security-sensitive events occur.
Zero-Power Notification

When any form of wireless communication occurs between the ICD and any outside source, the WISPer emits a chirping sound and vibrates to alert the patient.

The notification system was tested to ensure the chirping could be heard by surrounding the device in ground beef and placing it under a centimeter of bacon. This simulated the approximate depth under human flesh that the device is usually placed. Tests showed that the WISPer could be heard from one meter away, which is far greater than the distance of the implanted device to the ear.

Zero-Power Authentication

A simple cryptographic system was also implemented to conserve the privacy of medical telemetry. Each programmer stores a master key in secure hardware, and each ICD holds a unique ID. The key for each IMD is a hash of the master key and the ID of the IMD.

The protocol is a simple challenge/response. The programmer requests to have authorization from the ICD, and the ICD responds with its ID and a nonce. The programmer then responds with response $R = RC5(K, N)$, where $K$ is the master key and $N$ is the nonce. The ICD and the programmer can then communicate.

This scheme can help mitigate the risk of battery DoS attacks, as proper authorization must be obtained before any power-consuming operations can take place.

Key Management

As with many cryptographic systems, there is still the problem of key management. The proposed system uses an audio signal that is only audible to the programming head that is in contact with the patient’s skin. In this way, the patient is aware of any key exchange attempts.

While the audio signal can only be picked up by the nearby programming head, an eavesdropper can still utilize electromagnetic signals to obtain key information.

Limits and Additional Concerns

This study only focused on a single ICD device, and there are numerous IMDs available to exploit. The attacks on the single ICD proved time-consuming and could only be performed within a very short distance of a few centimeters.

Other Devices

Paul’s presentation also included a study called “Hacking Medical Devices for Fun and Insulin” by Jerome Radcliffe. This paper focused on exploits of insulin monitors and pumps similar to those detailed in the work by Halperin et al. Radcliffe’s work, however, showed how the independent processing of the insulin pump and monitor can be beneficial for warding off attacks, as the patient is still in charge of administering his or her own treatments.

Future Directions for IMDs

The future direction of ICD development focuses on increasing the wireless communication distance between an IMD and a commercial programmer to facilitate constant readings. While this may be more convenient for the patient and medical worker, it can introduce many security concerns, as seen in the study of Halperin et al.

Less human input to IMD functionality is also a goal for future device developments. Closing this human feedback loop, however, makes attacks easier, especially with the increasing functionality of cell phones that may have software radio capabilities.

What are Other Solutions?

Since weak FDA regulations are a large factor in faulty software release, improving these regulations would be a beneficial step. This includes increasing the quality of software reviews before product release, stressing threat models during software testing, revoking authorization privileges for lost devices, and requiring proper authorization to
use commercial programmers. Furthermore, commercial programmers should be removed from the ICD’s trusted computing base, as this study has shown how malicious insiders with this technology can be a major threat.

**Final Thoughts**

IMDs are important devices that preserve the length and quality of life for millions of people around the world. As a result of this importance, devices are often rushed to the market. Like any software, they already suffer from unintentional bugs and are becoming targets for malicious attacks.

We have seen examples of malicious attacks on an ICD device and proposed defenses. We have also seen how these attacks can be extended to target other devices, namely insulin monitors and pumps.

**Takeaway**

The production of implantable medical devices needs to continue; however, testers should ensure that they model a wide array of attacks, even if they have not occurred in the wild. There are unique conflicts in the medical filed when dealing with security, and proposed security defenses must take the safety of the patient into consideration.

**Discussion**

_The following are discussions following the presentation. The quotations are merely paraphrases of actual conversations._

**Prof. Butler:** What made this paper [Halperin et al] an Oakland paper, while the other [Radcliffe] was a Black Hat?

**Paul:** The Radcliffe paper was mainly postulated, not thorough, storyboarded, and there were no proposals for defenses. Bother were sensationally written, but the Halperin paper had more explanatory material and proposals.

**Prof. Butler:** Submitted papers must make scientific contributions. What were they in this [Halperin et al] paper?

**Paul:** They were drawing attention to issues in this important area of technology and showing how possible these attacks are. They also demonstrate that these things should be regulated.

**Prof. Butler:** From the papers we have seen, what would be a way to start attacking a system?

**Adam:** Determining the protocols used in the system

**Joe:** Finding the attack model

**Dan:** Yes, finding if we want to gather information from the system or shut it down.

**Reza:** With many of the systems we covered, there are logical threat models. For example, in the voting system paper is it vital to consider all players as an attacker.

**Getty:** The voting systems paper showed that the developers were underestimating the attackers. If code is written in a really clever/intricate way, it is harder to find how attackers will exploit it.