Syndrome: Spectral Analysis for Anomaly Detection on Medical IoT and Embedded Devices- Experimental Demonstration

Nader Sehatbakhsh, Hope Hong, Benjamin Lazar, Barry Johnson-Smith, Oghuzhan Yilmaz, Monjur Alam, Alireza Nazari, Alenka Zajic, and Milos Prvulovic

Recent advances in embedded and IoT (internet-of things) technologies are rapidly transforming health-care solutions and we are headed to a future of smaller, smarter, wearable and connected medical devices. IoT and advanced health sensors have provided more convenience to patients and physicians where physicians can now wirelessly and automatically monitor patient's state. While these medical embedded devices provide a lot of new opportunities to improve the health-care system, they also introduce a new set of security risks since they are connected to networks and run off-the-shelf operating systems. More importantly, these devices are extremely hardware and power constrained, which in turn makes securing these devices more complex. Implementing complex malware detectors or antivirus on these devices is either very costly or infeasible due to these limitations on power and resources.

This demo is the actual hardware implementation and live demonstration of our recent work "Syndrome: Spectral Analysis for Anomaly Detection on Medical IoT and Embedded Devices" that will appear in the Proceedings of HOST 2018. In this paper, we proposed a new framework called Syndrome for "externally" monitoring medical embedded devices. Our malware detector uses electromagnetic (EM) signals involuntary generated by the device as it executes a (medical) application in the absence of malware, and analyzes them to build a reference model. It then monitors the EM signals generated by the device during execution and reports an error if there is a statistically significant deviation from the reference model. To evaluate Syndrome, we use an open-source software to implement a real-world medical device, called a Syringe Pump, on a well-known embedded system *Arduino Uno.* We also implement a control-flow hijack attack on SyringePump and use Syndrome to detect and stop the attack.

Our setup for the hardware demo is shown in Figure 1. Our setup includes an Arduino Uno board which has an Atmega Microcontroller. The microcontroller is receiving commands from a Serial connection that is connected to a PC. It is also sends information to an LCD as a user-interface module. It also drives a DC motor based on the commands that it receives from the Serial communication and either injects or withdraws some amount of medicine (or fluid) that is defined by the user through the Serial communication. The step motor is then connected to an actual syringe that contains some amount of fluid and can move in both directions to either inject or withdraw the fluid.

In the first step of the demo, we show how a buffer overflow attack can be performed on the Arduino device using the Serial communication. Since the data is sent by the user from a PC to the device, a malicious user can hijack the control-flow by exploiting a vulnerability existing

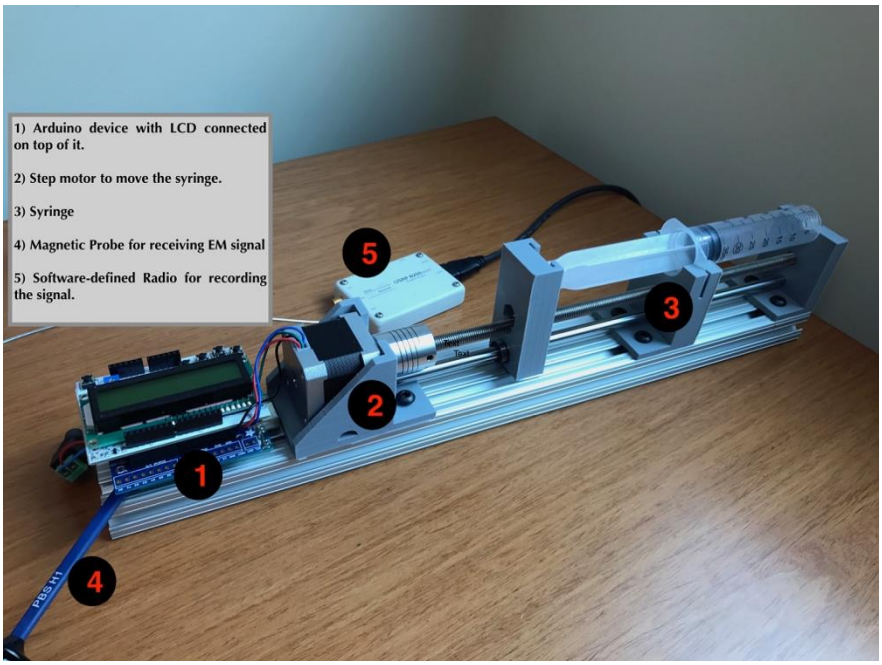

Figure 1. Hardware demo setup including 1) an Arduino device (with Atmega microcontroller), 2) an LCD connected on top of the device, 3) step motor, 4) syringe, 5) magnetic probe, and 6) Software-defined radio 7) a PC (not shown in the figure) for analyzing the data.

in the *readSerial* function on the SyringePump code. The attack is done by sending a very large string via Serial that will cause the buffer overflow on the microcontroller's stack. The actual malicious code is also injected with large string which will cause the control flow to be hijacked by the attacker. After hijacking, the Program Counter will jump to the attacker's malicious code and execute it. Using this attack, we will show how the syringe can move with any arbitrary amount of fluid or in any arbitrary direction. This could be extremely dangerous given that this device is used to inject (or withdraw) a very specific and exact amount of medicine to a patient's body.

The second step in our demo is showing how our Syndrome can help to immediately detect such a malicious activity. For that, we use a magnetic probe that is placed close to the microcontroller to receive EM signals emanated by the device. This probe is connected to a software-defined radio that can collect and send the data to a PC. On a PC we use a software framework developed by our team to analyze the signal and constantly compare the receiving signals to a reference model. If there is a significant deviation between the two signals, our framework will report a possible anomaly and the operator can immediately stop the operation. We will show how our framework can successfully detect all the instances of buffer-overflow attack.