Demonstration of Built-in Secure Register Bank (BSRB) Protection Scheme for Embedded System Security

Participant: Sean D. Kramer, Zhiming Zhang
Advisor: Professor Qiaoyan Yu
Affiliation: University of New Hampshire

1. Description of Research

Embedded systems are being utilized in the technology environment now more than ever before. As the world creeps closer to full-scale adoption of the Internet of Things (IoT), the importance of embedded systems will reach an all-time high. The concern with this rapid growth is the lack of security options available. Researchers forecast that the IoT security market will be worth over $36 billion by the year 2021 [1]. Current forms of security mechanisms, both software and hardware-based, offer a level of protection, but do have limitations such as significant performance degradation or power overhead [2, 3]. Thus, we propose a new, hardware-based protection scheme. Our goal is to reduce the amount of overhead and drawbacks associated with existing protections while maintaining a high level of security.

2. Features of Research Targeted in Demo

In this demonstration, we showcase the ability of our security scheme to thwart the execution of software exploits. This hardware demo consists of two parts. The first part introduces the development of our experimental setup and evaluation environment. For the second portion, we take an embedded system used in real-world applications and hijack its functionality by using return-oriented programing attacks [4]. Through this demonstration, we plan to teach just how easy it can be to exploit an existing system, and how our protection scheme will help to protect user data.

We utilize an FPGA development board, running a Linux OpenRISC operating system. On the device, a series of exploits are executed. These exploits consist of a stack buffer-overflow, data pointer and function pointer corruption, and a format string vulnerability. Each of these attacks simulate a return-oriented programming exploit [2, 4], a popular method of disrupting embedded systems.

The second portion of our demonstration is a real-world application of an attack. We will execute a type of return-oriented programming attack to an Engine Control Unit (ECU) for automobiles. This serves the purpose of showing the audience a real-world example of how an embedded system can be compromised through a software attack. Next, we redo the attack procedure, with our protection scheme implemented. Instead of successfully breaking into the ECU, our protection scheme will halt the exploit before being executed.

Figure 1: Experimental Set-up of Demonstration

3. What Will the Audience See?

While the demonstration is occurring, the audience will be able to witness the real-time output from the FPGA device. Our research uses the FPGA board connected to computer running a Linux Ubuntu operating system. The audience can view the output from the FPGA while the different types of exploits are being executed. By monitoring the output from the device, the viewers can witness the attacks happening in real-time, and the potential effects that these types exploits can have to an embedded system. After the security scheme is implemented into the device, viewers will see how the example programs are supposed to have run without interference from an unwanted attack.

The second part of the demonstration focuses on attacking an embedded system used in real-world applications. We will perform this by uploading a form of return-oriented programming attack to a PIC16F876A microcontroller. Our audience will watch as the exploit will cause the ECU to lose control Once our countermeasure has been implemented. The exploit will be redone, and the audience will watch as the ECU retains control instead of being compromised by the exploit.

4. References

[1] http://www.marketsandmarkets.com/PressReleases/iot-security.asp
[2] T. Dang, P. Maniatis, and D. Wagner, "The Performance Cost of Shadow Stacks and Stack Canaries," in *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security (ASIACCS)*, New York, NY: ACM, 2015, pp. 555–566.
[3] K. Piromsopa and R. J. Enbody, "Secure Bit: Transparent, Hardware Buffer-Overflow Protection," in IEEE Transactions on Dependable and Secure Computing, vol. 3, no. 4, pp. 365-376, Oct.-Dec. 2006.
[4] L. Davi, P. Koeberl, and A.-R. Sadeghi, "Hardware-assisted fine-grained control-flow integrity: Towards efficient protection of embedded systems against software exploitation," in *2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC)*, IEEE, 2014, pp. 1–6.