Tampering with Motes: Real-World Attacks on Wireless Sensor Networks

Alexander Becher Zinaida Benenson Maximillian Dornseif

RWTH Aachen University

Abstract: Most security protocols for wireless sensor networks (WSN) assume that the adversary can gain full control over a sensor node through direct physical access (node capture attack). But so far the amount of effort an attacker has to undertake in a node capture attack is unknown. In our project we evaluate different physical attacks against sensor node hardware. Detailed knowledge about the effort needed for physical attacks allows to fine tune security protocols in WSNs so they provide optimal protection at minimal cost.

1 Introduction

Wireless sensor networks (WSN) consist of a large amount of *sensor nodes*, which are small low-cost wireless computing devices equipped with different sensors. Sensor nodes are also called *motes* because of their small size and the envisioned deployment pattern: They are supposed to be spread over a large geographic area, organize themselves into an ad hoc network, and operate unattended for months or even years. Many intended applications have high security requirements. For an overview of security issues in sensor networks, see e.g. [PSW04].

One of possible attacks on WSNs is called *node capture* where an adversary gains full control over sensor nodes through direct physical access. Many newer security mechanisms for WSNs take node capture into account. It is usually assumed that node capture is "easy". Some security mechanisms are verified with respect to being able to resist capture of 100 and more sensor nodes out of 10,000 [HK04]. However, to the best of our knowledge, nobody ever tried to determine the actual cost to attack currently available sensor nodes. Thus our project was set out to verify the assumption that node capture is easy.

Contributions of our work so far Study of relevant literature revealed that node capture attacks are not so easy as usually assumed in the literature on WSNs. They require expert knowledge, costly equipment and other resources, and, most important, removal of nodes from the network for a non-trivial amount of time. We conclude that removal of a sensor node from the deployment area can be noticed by its neighbors, as well as by the sensor node itself, and the affected sensor node can be timely excluded from the network.

In experimental setups, we are evaluating possibilities to attack unattended sensor nodes *in the field*, without disruption of the regular node operation. So far we found some attacks, and also countermeasures.

Related Work Attacks on embedded systems, that is, on microcontrollers and smart cards, were studied, e.g., in [AK98, SA03, Sko05]. All described attacks require sensor nodes to be taken to a laboratory. Even if a mobile laboratory could be moved into the deployment area, all attacks would require at least disruption of the regular node operation, and in most cases, also physical destruction of sensor nodes.

2 Design Space for Physical Attacks on Sensor Nodes

We classify attacks according to two main categories. The first one is the amount of control gained over the sensor node, or the amount of information extracted from the node. In order of decreasing severity these are: (1) gaining complete read/write access to the **microcontroller**; (2) reading out RAM or flash **memory**, in whole or in part; (3) influencing **sensor readings**; and (4) manipulating **radio** communications.

The second category is the time necessary to carry out the attack. More precisely, the time during which the node cannot carry out its normal operation. We suggest three categories namely (1) **short** attacks of less than five minutes, e.g., creating plug-in connections and making a few data transfers over these; (2) **medium** duration attacks of less than thirty minutes, usually requiring some mechanical work such as soldering; and (3) **long** attacks which can only be carried out in a specialized laboratory.

3 Examples of In-the-Field Attacks and Countermeasures

Our project set out to actually implement attacks on WSNs in an attempt to measure the effort needed, evaluate the effectiveness of various security mechanisms, and come up with new ones. In the following section we will describe some of them.

JTAG The IEEE 1149.1 JTAG standard is designed to assist testing. It can be used for reading and writing arbitrary memory. All nodes we have examined had a JTAG connector on the board allowing easy access. Of the two microcontrollers we encountered on nodes during our research the TI MSP430 has a fuse which can be blown to disable all JTAG functionality. The Atmel ATmega128 has software-controlled lock bits to disable memory access via JTAG. With JTAG access, an attacker equipped with an appropriate adapter cable and a portable computer is capable of taking complete control over the sensor node.

Bootstrap Loader On certain nodes the canonical way of programming is by using the bootstrap loader (BSL) through an USB interface. The BSL enables reading and writing

memory independently of the software on the microcontroller. The BSL requires the user to transmit a password before carrying out any operation of value to an attacker.

The BSL password has a size of $16 \cdot 16$ bit and is equivalent to the flash memory content at addresses 0xFFE0 to 0xFFFF where the interrupt vector table is located.

An attacker might be able to guess the password by **brute force**. Certain restrictions apply to the values of the individual bytes reducing the key space from 256 bit to a much lower value. All code addresses have to be aligned to 16 bit word boundaries, resulting in a fixed least significant bit of every interrupt vector. Also, with the compilers we tested, both the reset interrupt vector and unused interrupt vectors were fixed. Finally code is placed by the compiler in a contiguous area of memory starting at the lowest flash memory address. We conclude that a BSL password in a worst case scenario may have an entropy of only 40 bit.

A possible brute force attack can be performed by connecting a computer to the serial or USB port and consecutively try passwords. By breaking the specification of the serial communication on the nodes we were able to try 83 passwords per second. This means that a brute force attack can be expected to succeed on the average after about 2^{32} s $\approx$ 128 a. As 128 years is well beyond the expected life time of current sensor nodes, a brute force attack can be assumed to be impractical.

One consequence of the fact that the password is equal to the interrupt vector table is that anyone in **possession of an object file** of the program stored on a node also possesses the password. Even someone who only has the source code of the program still can get the password if he has the same compiler as the developer, since he can produce an image identical to the one on the deployed node.

In order to avoid this form of attack, we proposed and tested a technique called **interrupt vector randomization**. We have written a program that preprocesses a program image before installation on a node. Our tool replaces the original interrupt vector table with a series of 16 randomly chosen, previously unused code addresses. Unconditional branch instruction to the original interrupt handler routine are placed at each of these addresses.

3.1 External Flash

Some applications store valuable data on the external EEPROM. Probably the simplest form of attack is eavesdropping on the conductor wires connecting the external memory chip to the microcontroller, possibly using a suitable logic analyzer. A more sophisticated attack would connect a second microcontroller to the I/O pins of the flash chip. If the node's microcontroller will not access the data bus while the attack is in progress, it can be completely unnoticed. Instead of using her own chip, the attacker could simply overwrite the program in the nodes's microcontroller and put her own program on it to read the external memory contents. This operation is even possible without knowledge of the BSL Password. While this causes "destruction" of the node from the network's point of view, in many scenarios this might not matter to the attacker.

So far we were not able to carry out these attacks in practice.

4 Conclusion

We systematically investigated physical attacks on current sensor node hardware, paying special attention to attacks which can be executed directly in the deployment area, without interruption of the regular node operation. We found out that most serious attacks, which result in full control over a sensor node (node capture), require absence of a node in the network for a substantial amount of time.

Thus, in order to design a WSN secure against node capture attacks, *standard precautions* for protecting microcontrollers from unauthorized access, such as disabling the JTAG interface, or protecting the bootstrap loader password can improve resilience greatly. We developed a method of protecting the bootstrap loader password by randomization of the interrupt vector table. This allows the developers to make source code of their products public without fearing that their WSN can be taken over easily.

We are interested in further developing other attacks and countermeasures on WSNs.

Mechanisms for *revocation* of a sensor node which was absent for too long from the network by its neighbors should be developed. This is our future work. Note that depending on the WSN design, local revocation could be insufficient. For example, if an attacker removes a single sensor node from the network and successfully extracts the node's cryptographic keys, the attacker would be able to clone nodes, i. e., to populate the network with new sensor nodes which all use the cryptographic keys of the captured sensor node. Thus, a WSN should also be protected from *node cloning*.

References

[AK98] Ross J. Anderson and Markus G. Kuhn. Low Cost Attacks on Tamper Resistant Devices. In *Proceedings of the 5th International Workshop on Security Protocols*, pages 125–136, London, UK, 1998. Springer-Verlag.

[HK04] Joengmin Hwang and Yongdae Kim. Revisiting random key pre-distribution schemes for wireless sensor networks. In *SASN '04: Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*, pages 43–52. ACM Press, 2004.

[PSW04] Adrian Perrig, John Stankovic, and David Wagner. Security in wireless sensor networks. *Commun. ACM*, 47(6):53–57, 2004.

[SA03] Sergei P. Skorobogatov and Ross J. Anderson. Optical Fault Induction Attacks. In *CHES '02: Revised Papers from the 4th International Workshop on Cryptographic Hardware and Embedded Systems*, pages 2–12, London, UK, 2003. Springer-Verlag.

[Sko05] Sergei P. Skorobogatov. Semi-invasive attacks - A new approach to hardware security analysis. Technical report, University of Cambridge, Computer Laboratory, April 2005. Technical Report UCAM-CL-TR-630.