

Embedded in Thin Slices

Internet of Things Security (Part 2)

Side-Channel Attacks

```
if ($?GET['defaults'] == 1)
{
    $reboot_needed = 1;
    $response = ` /apps/cmdxmlin defaults `;
}
```

In this next part of his article series on IoT security, Bob takes a look at side-channel attacks. What are they? How much of a threat are they? And how can we prevent them?

By
Bob Japenga

COLUMNS

Growing up in the 50s and early 60s, I spent hours in the basement of one of my friend's houses reading *Mad Magazine*. One of our favorite parts of the magazine was the "Spy vs. Spy" cartoons (**Photo 1**). It consisted of endless gags where one side tried to out-smart the other while the other tried to use comical countermeasures. I never guessed that embedded systems design would one day become the playground of attacks and countermeasures in our own real-world "Spy vs. Spy." Yet it truly has come to this.

Our company MicroTools has, over the years, created a number of successful Internet of Things (IoT) products for various customers. In that time, we have recognized that security is becoming more and more important—not necessarily to our customers but for our customers. More on that later. And we have found that to build airtight secure systems is hard and getting harder. Our countermeasures are met with ever more sophisticated attacks much in the spirit of Spy vs. Spy. To help us continue to create ever more secure embedded systems, we have begun to design some IoT devices using microprocessor chips that claim to have built in security features. Just last month I began reading the data sheet about one of the chips we were using (Microchip's SAM5D2) and found the following statement:

[The chip has] several hardware features that safeguard memory content, authenticate

software reliability, detect physical attacks and prevent information leakage during code execution.

Okay, I understood all of those security features except one. I wasn't sure what they meant by "information leakage during code execution." So, I began researching the literature to find out what they meant. The terminology was different from what I used and discovered that they were talking about side-channel attacks. What the chip maker was referring to when they mentioned "information leakage" was that the chip had built-in features that prevented some forms of side-channel attacks.

SIDE-CHANNEL ATTACK DEFINED

With respect to IoT security, a side-channel attack is one that uses information leaked from the embedded system to gain access to the device and its data. Side-channel attacks do not require access to the system through open IP ports or by connecting to an external communications channel. Instead they use certain known physical characteristics of the device to obtain access.

There are many different classes of side-channel attacks that are well documented on the web and in this magazine so we won't repeat that information here. Also, a class of attacks that are sometimes classified as side-channel attacks are fault attacks. These are where you extract security information by injecting a fault through temperature, voltage

(range or glitches) or electro-magnetic radiation. Colin O'Flynn touches on one of these in his *Circuit Cellar* 330 (January 2018) article. I do not consider these side-channel attacks even though the Wikipedia article on Side-Channel Attacks considers them as such. Let me just mention a few classes of side-channel attacks here:

Power Analysis attack: There are two types of power analysis attacks: Simple Power Analysis and Differential Power Analysis. Simple Power Analysis is based on the assumption that every instruction has a different power profile and the algorithm is known that it is trying to break. From these profiles, a hacker with physical access to the embedded system can extract the private keys from that device if the execution path of the code depends on the data being processed. Hackers feed keys in one bit at a time and watch for the change in execution path based on the change in the power profile when a correct bit is used.

Differential Power Analysis uses statistical methods to not just decode the instruction but the actual data being used with the instruction. Differential Power Analysis allows the hacker to reverse engineer the code simply from the “leaked” power information.

The standard security algorithms AES and DES are trivially broken through power analysis. One researcher took 50 different products of a wide range of functionality and extracted the private keys from each.

The web site www.chipwhisper.com (the work of *Circuit Cellar* columnist O'Flynn) provides open source tools to perform power analysis. It also contains many good instructional videos and demonstrations of how power analysis works.

Timing attacks: Using timing attacks to extract security information has been documented since 1996! There are two types of timing attacks: those that require physical access and those that don't. Basically, the time through your code is different based on the success or failure of some of your security code. A hacker can measure the time of your embedded system's response and deduce keys and passwords. *Circuit Cellar* columnist Colin O'Flynn has done a great job demonstrating a timing attack that requires physical access. There is also a lot of work done demonstrating that statistical analysis of a timing attack can extract key security parameters (keys and passwords) remotely.

Temperature attacks: Just as your embedded design creates a power profile and a timing profile—both of which leak vital information about its inner workings—the temperature profile of your chip leaks the same vital information. Yes, but can this

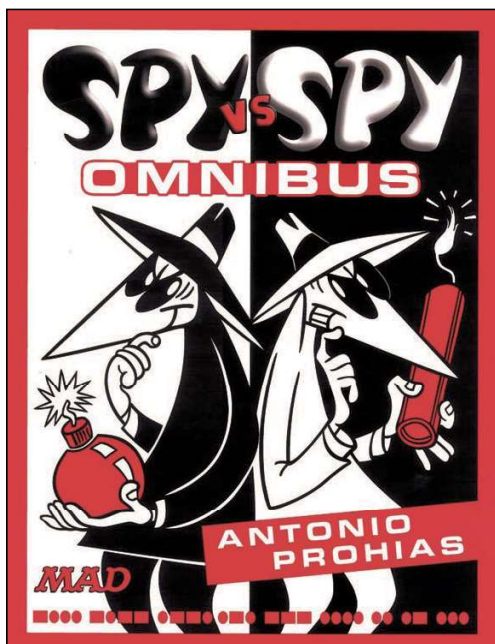


PHOTO 1

Embedded systems design today has to deal with attacks and countermeasures in our own real-world “Spy vs. Spy” scenario, like the classic Mad Magazine cartoon.

really be exploited? Researchers state that the “temperature side-channel has a very low bandwidth limiting practical attacks” and remains theoretical in a practical attack. But it is something for us to keep in mind as we attempt to design air-tight security in our embedded systems. I am convinced that detecting the lid removal from the microprocessor will prevent these for a long time.

Electromagnetic attacks: A few of you might remember writing code for the Altair 8800 (**Photo 2**) to play music on our AM radios. Didn't we have anything better to do? We thought it was so cool. Basically, the electromagnetic radiation from your embedded system is emitting unique music to the ears of hackers. And from that music they too are able to extract security information from your device. And that truly is music to their ears. This has been done successfully since 2001.

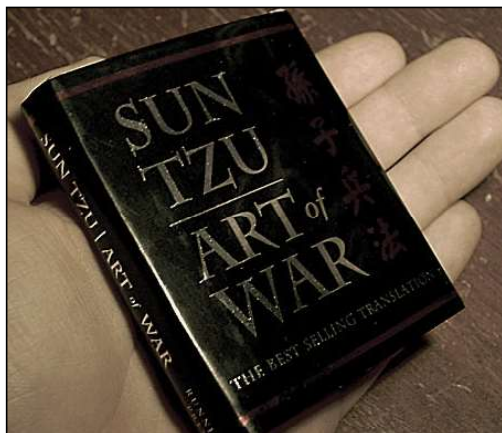


PHOTO 2

The Altair 8800 is a microcomputer designed in 1974 by MITS and based on the Intel 8080 CPU.

PHOTO 3

Sun-Tzu's "The Art of War" advises you should know your enemy and know yourself. In embedded systems, you need to know your enemy and your product. You also need to be aware of all of the possible threats to your system.



HOW MUCH OF A THREAT?

A valid question at this point is: "Who would go to the trouble of performing a side-channel attack?" I cannot find any evidence of a real-world embedded system fleet that was compromised in this way. Most attacks have come because we have just left the barn door open for our secure information to get out. I would love for any of you to send me any examples of a real-world side-channel attack. Are we taking countermeasures without any real-world attackers? Is this just "vs Spy" where there is no attacker? You might say that some hackers have gone after some smart cards to obtain the money from the card but nobody is interested in my control systems that make glass containers or my solar power monitoring equipment. This is where I think we are wrong. Let me offer two reasons why we should be serious about side-channel attacks (and the same applies to fault injection attacks as things we need to be concerned about.)

Ransomware: I first heard about ransomware in about 2012. There were some well documented attacks on individuals and organizations where the attacker hijacked their data and asked for a moderate amount of ransom money to get the data back. The amounts were set very cleverly at the level where it was much cheaper to get your data back by paying the ransom than to get the data back through other means. With this in mind, what would happen to your company if through a side-channel attack, someone was able to stop all glass container machines around the world. How much would you pay the extortionist to start the machines back up? With the advent of Bitcoin, the money can

now be securely transferred to the extortionist with no traceability. Or perhaps you make a digital garbage can that is deployed by the thousands in hundreds of large cities around the world. If some side-channel attacker was able to stop all of the digital garbage cans from talking to the host, what would you pay to get that feature back?

My point is that there are countries and terrorist organizations with vast resources of very smart technical people with time on their hands who are strapped for cash. What looks like a lot of effort (extracting private keys through side-channel attacks) with little benefit instead could reap significant financial rewards.

Public Relations: You might say "ransomware is not an issue for me." Fine, then my second and equally powerful argument for thoroughly understanding all the possible threats and taking appropriate action is public relations. There are literally thousands of researchers out there who would love to make a name for themselves and demonstrate a real-world side-channel attack on your system. Don't get me wrong. We should appreciate the effort being expended to test the security of our systems before a malicious attacker breaks in. The question we need to ask is: What would happen to your company if you got written up in trade journals that your device was susceptible to a real side-channel attack? I guess if you have no competition or your competition is also all being written up, then you won't lose anything. But I doubt that this is a real-world scenario. We all have skilled and professional competitors who are just waiting for the chance to overtake our market share.

This doesn't mean that we pull out all the stops and attempt to prevent every form of side-channel attack. But by using wise risk assessment methodologies with a detailed knowledge of who your enemy is, you should select the countermeasures that best minimize your risk.

One of the challenges that I alluded to earlier is that most of our customers are not willing to pay for these security features up front. What is the cost-benefit analysis of this security feature? That is almost impossible to estimate because we don't know the probability of a hacker actually using these techniques. And we don't know the public relations cost of a failure. That is why we as designers have to build it into the products we design somewhat independent of our customers' demand. It is for their own good. I'll leave the hard question of how to do that up to you.

HOW TO PREVENT THEM?

Here are my suggestions:

Know your enemy and your product: At the beginning of this article I said that

For detailed article references and additional resources go to:
www.circuitcellar.com/article-materials

RESOURCES


Microchip Technology | www.microchip.com

embedded security has evolved into a kind of “Spy vs Spy” cartoon strip. Those cartoons were launched during an intense period of the cold war. So, we can look to the master of making war, the 6th century BC military general Sun-Tzu who wrote “The Art of War” (**Photo 3**). From it we can glean our first suggestion: “... If you know your enemies and know yourself, you will not be put at risk even in a hundred battles.” You need to be very aware of all of the possible threats to your system. Then as part of a systematic risk assessment process you need to identify how you will mitigate that risk. Perhaps physical access by hackers is not possible with your design—for example a controller for automatically testing one of your products. If they had physical access they could wreak other havoc without the need for side-channel attacks. So, you can eliminate a large number of side-channel attacks because you know your device.

Minimize leakage: This is easier said than done. But here are some suggestions. Find out what your chip and OS provide. Then enable as many of the security features as feasible. Where possible, use security algorithms with constant path execution. This can mitigate electromagnetic, power and thermal attacks. If possible, use spread spectrum

clock frequencies to mitigate timing attacks. Where possible, encapsulate your design with tamper protection.

CONCLUSION

Attacking a topic like this (pun intended) is daunting. IoT security is a large and important topic. Side-channel attacks have the potential of making or breaking your product. Hopefully, with this article and the many other fine articles in *Circuit Cellar*, we have added to your knowledge base, if only in thin slices. 

ABOUT THE AUTHOR

Bob Japenga has been designing embedded systems since 1973. In 1988, along with his best friend, he started MicroTools, which specializes in creating a variety of real-time embedded systems. MicroTools has a combined embedded systems experience base of more than 200 years. They love to tackle impossible problems together. Bob has been awarded 11 patents in many areas of embedded systems and motion control. You can reach him at rjapenga@microtoolsinc.com.



Verilog HDL

With the right tools designing a microprocessor can be easy.

Okay, maybe not easy, but certainly less complicated. Monte Dalrymple has taken his years of experience designing embedded architecture and microprocessors and compiled his knowledge into one comprehensive guide to processor design in the real world.

Monte demonstrates how Verilog hardware description language (HDL) enables you to depict, simulate, and synthesize an electronic design so you can reduce your workload and increase productivity.

cc-webshop.com

