

Embedded in Thin Slices

Internet of Things Security (Part 6)

Identifying Threats

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

In this final part of his Internet of Things Security article series, this time Bob returns to his efforts to craft a checklist to help us create more secure IoT devices. This time he looks at developing a checklist to evaluate the threats to an IoT device.

COLUMNS

By
Bob Japenga

A number of years ago (there were woolly mammoths around if I remember correctly), I attended a conference on the Ada programming language. Ada was created for the United States' Department of Defense to replace the myriad of programming languages that were deployed by the DoD at that time. The language was named after the first programmer, Augusta Ada King Lovelace, a colorful character in her own right and the only legitimate daughter of the poet Lord Byron. Ada is credited with publishing the first algorithm for use on a computing machine: Charles Babbage's famous analytical engine.

At the conference I attended a breakout session on algorithms. In the conference room next door, a popular speaker, whose name I don't remember, held another breakout session. About ten minutes into the session, we heard a deafening chant coming from the conference room next door that repeated over and over: "I don't care." The speaker was making a point that, as software designers, we should not care about everything. There are legitimate things for which we need to say: "I don't care." We need to identify them as not relevant to the task at hand and emphatically say: "I don't care."

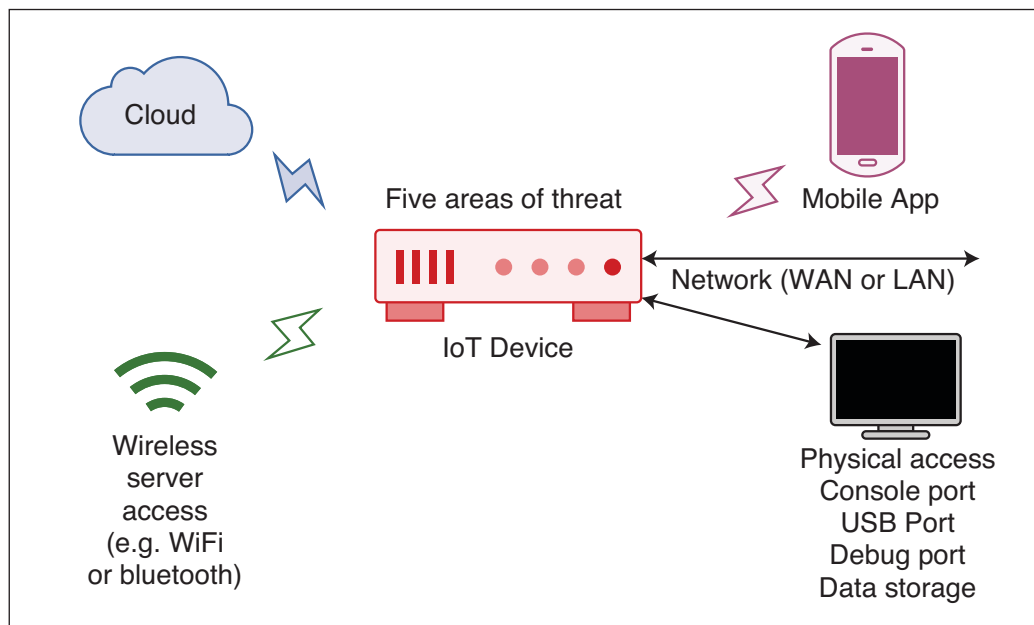
Although I remember nothing from the breakout session on algorithms, I have never forgotten this principle: "There are some

things that we just don't care to address when designing embedded systems." Certainly, there is much to be said for thoroughness in design, but when we—with well thought through analysis—determine that some aspect of a design is a "don't care" we need to let it go.

In designing secure IoT devices this is a very important principle. The threats are diverse and difficult to number. The assets are important and of differing value. This month we will continue to build our checklist for IoT security. Last time we created a checklist to help you identify the assets that you want to protect. This month we will add to that checklist with some questions to help you identify and quantify the threats.

IDENTIFYING THE THREATS

We need to start with definitions. A good working definition for a threat would be: "a person or thing likely to cause damage or danger." Although this is a good definition, for the purpose of building our checklist, I want to expand upon it a little. Here's why: In most cases "I don't care" who the threat is, nor do I care what their capabilities are. Keep in mind that, if there is a threat with very little capabilities, that threat could get passed on. They can always sell either their knowledge or their access to the device to someone who has the capabilities to create a security

**FIGURE 1**

Shown here are the five areas of threat I've identified for IoT devices.

breach with the device. Let me illustrate that. Imagine there are two threats: One is a disgruntled former employee with little or no capability of reverse engineering your design in order to find a security flaw. The second is an organization with deep pockets and highly skilled hackers. If any of the assets that we identified in the first part of the checklist are worth a significant chunk of change, the former employee can always sell what they have to this other organization. With all that in mind, in general "I don't care" about who the threat is.

But I do care about the activities of these threat agents. This is in line with the way the OWASP Top Ten IoT Security Threats is laid out. The Open Web Application Security Project (OWASP) is a worldwide organization focused on improving the security of software. I introduced OWASP as a valuable resource in my August 2016 column (*Circuit Cellar* 313) when we discussed their list of the top ten security vulnerabilities. The list was updated in 2017 and worthwhile to review [1]. OWASP also provides what it calls the top ten threats to IoT devices. We will look at these a little later in this article. They agree with my assessment that we don't care who it is or what their capability is. What we care about is the action that they can take.

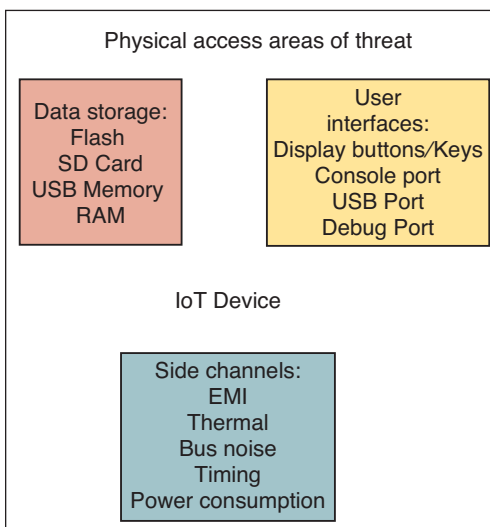
When thinking about threats to the security of our IoT device, I would identify five areas of threat as shown in **Figure 1**: access to the physical device; access to the wireless services on the device; access to the network (LAN or WAN) the device is on; access to the cloud server used by the device; and access to the mobile app used by the device. Anyone who has access to one or more of these is a threat agent. So, the beginning of our checklist

needs to analyze what harm could be done by such a threat agent who gained access to any of these five areas of threat. Not all of your IoT devices have all of these areas of threat but most have a majority of them. For each of the areas of threat we need to ask the question: What would be the potential cost if someone with a lot of time, highly skilled hackers and a lot of money got access to one of these areas of threat without permission?

This provides the first five elements of the Threat portion of the IoT Security Checklist. Let's look at each of these.

FIVE THREAT ELEMENTS

Physical access: Not all IoT device designers will consider physical access to the device an area of threat. For example, we are currently working with a client who has determined that there is very little risk of an unauthorized person having physical access

**FIGURE 2**

Shown here are the access areas of threat if physical access is a threat area for your device.

OWASP Top Ten IoT Security Threats

1. Insecure Web Interface	Anyone who has access to the web server (external/internal users)
2. Insufficient Authorization/Authentication	Anyone who has access to the web interface, mobile interface or cloud interface
3. Insecure Network Services	Anyone who has access to the device via a network
4. Lack of Transport Encryption	Anyone who has access to the network the device is on
5. Privacy Concerns	Anyone who has access to the device itself; the network the device is on; the mobile app; the cloud app
6. Insecure Cloud Interface	Anyone who has access to the Internet
7. Insecure Mobile Interface	Anyone who has access to the mobile app
8. Insufficient Security Configurability	Anyone who has access to the device
9. Insecure Software/Firmware	Anyone who has access to the device itself; the network the device is on; The update server
10. Poor Physical Security	Anyone who has physical access to the device

TABLE 1

Top 10 IoT security threats identified by the Open Web Application Security Project (OWASP)

to their device. For most cases this is true. The device is only touched by employees and is physically inaccessible to everyone else. But I have not pushed them to protect the assets accessible through physical access for other reasons. I have gone along with that assessment because the assets available inside the device are minimal. But if the assets were valued higher, I would push back more strongly primarily due to the potential of a disgruntled or greedy insider handing the unit off to a qualified hacker.

If physical access is a threat area for your device, then the following access areas portrayed in **Figure 2** need to be protected: access to data storage; access to user interfaces; access to USB ports; access to console ports; access to side channel information; and access to debug ports.

Mobile app: Many of our IoT devices have a mobile app associated with it. Although not strictly part of the IoT, it is certainly something that needs to be considered when designing your IoT device. Certainly, one approach is to limit who can put your mobile

app on their phone or tablet. This certainly provides a great physical barrier to access. But the integration of Google's Play Store and Apple's App Store with your phone and tablet makes for very easy deployment and is very tempting to us designers. Surely the next line of defense is to drastically limit what the mobile app can access. Again, this is the power of the mobile app interface and you hate to lose it. Requiring a username and strong password is your next line of defense. For now, our job is to identify what harm someone bent on destroying your business would do if they were given unlimited access to your mobile app. How your mobile app communicates to the device is another concern we'll look at next.

Wireless access: Your IoT device may provide several wireless ways to connect to it: cellular, Wi-Fi, Zigbee, Thread, Bluetooth, IrDA and Near Field Communication (NFC) are some of the most common. At this point in our checklist we need to ask: What if an unauthorized person got on your device wirelessly? What harm could be done? What if someone could perform a man-in-the-middle attack? The most recent Bluetooth hacking technique [2] shows us that even secure transmissions can have holes in their implementations allowing for man-in-the-middle attacks. So, we cannot just rely

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

References [1] through [3] as marked in the article can be found there.
A link to Bob's IoT Checklist is also available there.

on secure transmissions as our only source of protection. I think about this every time I connect over Bluetooth to my OBD2 (on-board diagnostics) interface in my car. What would happen if someone could get on that interface and muck with my on-board computer? There's no doubt that providing good access control through usernames and passwords, encrypting and authenticating all traffic and limiting physical access are all in your arsenal of protection. For now, we are concentrating on evaluating the harm nefarious access over the wireless interfaces on your IoT device could do.

Cloud access: Like mobile access, your cloud access is not strictly part of the IoT device. But again, we must pursue the questions: What if an unauthorized person got on your cloud interface? What harm could be done? The cost of that harm will help you to evaluate the amount of security you need to provide to the cloud interface. Clearly, we don't want to use unencrypted transmissions. HTTPS provides encryption for us. But we found that on one of our major projects the cell modem chip only supported HTTP. So, we needed to encrypt the transmissions ourselves. Secure user access is pretty standard for cloud interfaces. But again, don't rely on these layers. Seriously address what harm a malicious hacker intent on destroying your company could do if they had full access to your IoT cloud interface.


IoT network: Some of our IoT devices still have an Ethernet interface and provide some form of local area networking (LAN) or wide area networking (WAN). But this could be any wired network interface. Again, we need to look hard at what someone could gain from watching the traffic on the network. Our company's most serious security breach came because of a little used Ethernet port that provided unencrypted traffic to a Link Local address. A researcher sniffed it out and found a security flaw.

Finally, to fill out our checklist, we've created items from the OWASP Top Ten project [3]. **Table 1** provides their list of threats. We have taken each one of these and created a checklist to use in evaluating whether or not these threats have been addressed in your design. To see our updated IoT Checklist, go to the *Circuit Cellar* article materials webpage.

CONCLUSION

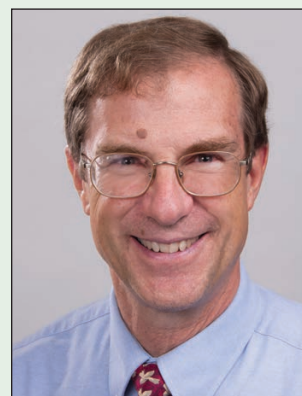
The Aesop fable of the crow and the pitcher is instructive for us here. A crow was dying of thirst. He found a pitcher with a little water at the bottom. He found that if he dropped one little pebble in the bottom of the pitcher, eventually he could reach the water. Assuring IoT security is a gigantic task. We access life giving water one pebble at a time. We eat elephants one bite at a time. We

create secure IoT products one small step at a time. Checklists can help us assess whether or not we have covered the security issues we know about—one step at a time. Not knowing everything is why this document cannot be static. Take this resource and make it yours. There are things that you will need to say: "I don't care" about this threat. Tailor it in your project plan for each project. But use checklists in developing secure IoT products.

Next time we will begin a new topic on Bluetooth Mesh. Of course, 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.



FlexRes® Oscilloscopes

Speed or resolution, analog or digital.
Why compromise?

pico
Technology

PicoScope® 5000D Series

- FlexRes flexible 8 to 16 bit hardware resolution
- Up to 200 MHz analog bandwidth
- Up to 512 MS capture memory
- 16 digital channels (on MSO models)
- Serial decoding as standard (18 protocols)
- 70 dB SFDR at 100 kHz



Try our software for free at
www.picotech.com/A174