

Embedded in Thin Slices

Internet of Things Security (Part 4)

The Power of Checklists

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

In this next part of his article series on IoT security, Bob looks at how checklists and the common criteria framework can help us create more secure IoT devices. He covers how to create a list of security assets and to establish threat checklists that identify all the threats to your security assets.

By
Bob Japenga

In 1934, the United States Army Air Corps solicited proposals from major aircraft manufacturers for a new long range bomber. Boeing, Douglas and Martin all submitted their proposals and conducted a fly-off in 1935. Boeing's submittal—what was to become the B-17 or Flying Fortress [1] (**Figure 1**)—exceeded the specifications including carrying five times as many bombs as requested. However, during the fly-off, the pilots forgot to unlock the control surfaces at takeoff and the plane crashed, killing the test pilot and a Boeing employee. The Air Corps determined that the error was due to the complexity of the operational procedures for flying the plane. This complexity caused the pilot and crew to miss an obvious step at takeoff.

Although the award went to the Douglas B-18, the Air Corps loved the B-17's potential and, through a loop-hole, ordered 13 for testing. The test pilots and Boeing were determined to make the plane safe and flyable. What they came up with was the lowly checklist [2] (**Figure 2**). The rest, as they say, is history. The B-18 ended up playing a minor role in World War II while over 12,000 B-17 were built and were instrumental in the Allied victory. Some of you may remember from either the book or the movie *Unbroken* [3], Louis Zamperini's desire to fly on the B-17 and his disappointment when he was assigned to the B-24. Zamperini dramatically captures the reputation that B-17 carried with it.

Atul Gawande is a surgeon who saw

many situations in hospitals where the correct diagnosis or response was missed not because of ignorance but because it was overlooked. And he wanted to change the way we function in a variety of hospital settings. In his book, *The Checklist Manifesto* [4] he outlines the cause, the cure and the results of using checklists in hospitals. The cause is that the "volume and complexity of what we know has exceeded our individual ability to deliver its benefits correctly, safely or reliably. Knowledge has both saved us and burdened us." The cure he came up with was the lowly checklist. The results were eye popping. For example, he found that when "doctors and nurses in the ICU create their own checklists for what they think should be done each day, the consistency of care improves to the point where the average length of patient stay in intensive care dropped by half." The book covers a wide variety of scenarios where the lowly checklist dramatically improves quality.

So, what does this have to do with IoT security? Basically, my thesis is that, although IoT security is complex, most of our failures have come, not from lack of knowledge but from lack of execution. I think checklists can help us get better at IoT security. We use them at our company for releasing software so why not for security concerns in our designs? Over the next several articles I would like to present a checklist for you to use as you develop your next generation of IoT product. I am not advocating a simplistic vacation or chores checklist. ISO/IEC 27003:2016 [5]

**Figure 1**

Boeing's long range bomber design that what was to become the B-17 (shown here) exceeded the specifications including carrying five times as many bombs as requested.

Annex A provides a checklist more like what I am looking for. It includes a project phase, the description, the documented output and so on. We will explore this more next time.

COMMON CRITERIA

If you are not on Jack Ganssle's *Embedded Muse* newsletter mailing list you should. Jack provides a perfect blend of detail and breadth to connect you with the latest tools and ideas for embedded system design. In his March article [6], Jack described the most recent Embedded World Conference held in Germany. When talking about presentations on IoT security and bemoaning the lack of solid solutions, he threw out the following statement: "the Common Criteria is a start but hardly the end-all metric." If there is one thing I have learned in the last 46 years of embedded systems design, it's that I don't know much. I never heard of the Common Criteria. Perhaps it's new to you too.

Various efforts have been made to systematize security both from a requirements perspective and from a verification perspective. Some very similar standards have come out of these efforts: ISO/IEC 15408 [7], ISO/IEC 18045 [8] and ISO/IEC 27k [9] to name a few. Common Criteria [10] was created to provide the framework for establishing and ensuring the security of systems which can be software, hardware, firmware or any combination thereof. It seems that the goal was to provide a means for independent labs to have a way to validate the security of a device.

This is an admirable goal. Much as the standards community has done for safety, it

looks like there was a desire to create common criteria for a class of devices. For example, for most electronic systems we design, there is a general testing requirement safety standard (IEC 61010-1) [11]. If the equipment is a medical device, there is an additional safety standard (ISO/IEC 60601-1) [12]. If the device has software in it, another safety standard (ISO/IEC 62304-1) [13] is called out. If the device is an Oxygen Concentrator, there is a further safety standard (ISO/IEC 80601-2-69) [14]. When we developed a portable oxygen concentrator, for safety certification we tested against these four standards. Independent test labs across the globe can certify that our

RESTRICTED

APPROVED B-17F and G CHECKLIST	
REVISED 3-1-44	
PILOT'S DUTIES IN RED COPILOT'S DUTIES IN BLACK	
BEFORE STARTING	ENGINE RUN-UP
1. Pilot's Preflight—COMPLETE	1. Brakes—Locked
2. Form 1A—CHECKED	2. Trim Tabs—SET
3. Controls and Seats—CHECKED	3. Exercise Turbos and Props
4. Fuel Transfer Valves & Switch—OFF	4. Check Generators—CHECKED & OFF
5. Intercoolers—Cold	5. Run up Engines
6. Gyros—UNGAUGED	BEFORE TAKEOFF
7. Fuel Shut-off Switches—OPEN	1. Tailhook—Locked
8. Gear Switch—NEUTRAL	2. Gyro—Set
9. Cowl Flaps—Open Right—	3. Generators—ON
OPEN LEFT—Locked	AFTER TAKEOFF
10. Turbos—OFF	1. Wheel—PILOT'S SIGNAL
11. Idle cut-off—CHECKED	2. Power Reduction
12. Throttles—CLOSED	3. Cowl Flaps
13. High RPM—CHECKED	4. Wheel Check—OK right—OK LEFT
14. Autopilot—OFF	BEFORE LANDING
15. De-icers and Antifreeze, Wing and Prop—OFF	1. Radio Call, Altimeter—SET
16. Cabin Heat—OFF	2. Crew Positions—OK
17. Generators—OFF	3. Autopilot—OFF
STARTING ENGINES	4. Booster Pumps—On
1. Fire Guard and Call Clear—LEFT RIGHT	5. Mixture Controls—AUTO-RICH
2. Master Switch—ON	6. Intercooler—Set
3. Battery switches and inverters—ON & CHECKED	7. Carburetor Filters—Open
4. Parking Brakes—Hydraulic Check—On—CHECKED	8. Wing De-icers—Off
5. Booster Pumps—Pressure—ON & CHECKED	9. Landing Gear
6. Carburetor Filters—Open	a. Visual—Down Right—DOWN LEFT
7. Fuel Quantity—Gallons per tank	b. Tailhook Down, Antenna in, Ball Turret Checked
8. Start Engines: both magnetos on after one revolution	c. Light—OK
9. Flight Indicator & Vacuum Pressures CHECKED	d. Switch—OFF—Neutral
10. Radio—On	10. Hydraulic Pressure—OK, Valve closed
11. Check Instruments—CHECKED	11. RPM 2100—Set
12. Crew Report	12. Turbos—Set
13. Radio Call & Altimeter—SET	13. Flaps 1/2—Down
	FINAL APPROACH
	14. Flaps—PILOT'S SIGNAL
	15. RPM 2200—PILOT'S SIGNAL

RESTRICTED

Figure 2

The test pilots and Boeing were determined to make the B-17 safe and flyable. To aid this goal they came up with a checklist.

Common Terms	Possible Descriptions
Catastrophic	Results in company bankruptcy or loss of life
Critical	Results in loss of product or brand (loss of all sales) OR major down turn in the sales of all products
Serious	Results in significant loss of sales of product OR significant costs to rectify.
Minor	Results in minor down turn in sales of product OR minor (with respect to sales) costs to rectify.
Negligible	Results in public relations embarrassment resulting in no measurable down turn in sales of product.

Table 1

Security severity levels

	Negligible	Minor	Serious	Critical	Catastrophic
Frequent	21	13	7	4	1
Probable	22	16	9	5	2
Occasional	23	18	11	6	3
Remote	24	19	14	10	8
Improbable	25	20	17	15	12

Table 2

Semi-qualitative probability levels

	Negligible	Minor	Serious	Critical	Catastrophic
Frequent	21	13	7	4	1
Probable	22	16	9	5	2
Occasional	23	18	11	6	3
Remote	24	19	14	10	8
Improbable	25	20	17	15	12

Table 3

The data from Tables 1 and 2 are combined here into a matrix generating a probabilistic severity index.

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.



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

References [1] through [17] as marked in the article
can be found there.

device has been tested to these standards. For a lot of devices, certification is not required by law but may be required for marketing purposes. For example, some installers would not install our solar energy monitors unless it passed certain safety standards.

In spite of the immense effort that went into creating this framework, as Jack Ganssle stated, this is only a start. In fact, so far it seems to me to fall significantly short of the goal. What is needed would be to provide the standards to which our IoT devices can be certified as secure. This is not my area of expertise (writing international standards), but I think there could be criteria based on the given technology. For example, there would be a standard for Bluetooth security, one for Wi-Fi security, one for Ethernet security, one for cell modem security and so on.

For Bluetooth for example, the standard should delineate the threats associated with its use. The ISO27K [15] initiative claims to be doing something like that. But it is too general to be very useful. For example, ISO/IEC 27033-6:2016 describes the threats, security requirements, security control and design techniques for wireless IP network access. It has a large section on Bluetooth. But it is not comparable to the safety standards. There are not concrete specific requirements in these standards as there are in safety standards. As of yet, I don't see how I can go to an independent lab and say: "Please test my IoT device to 27033-6."

Not being the faint of heart in terms of going where angels have feared to tread, I would like to begin creating a checklist that we can actually use in designing until such standards are created.

IDENTIFYING THE SECURITY ASSETS

The very first thing that is needed is to create a list of all of the security assets for a project. Since this may evolve over time, this needs to be a living document. Sometimes our customers "discover" new security assets as the project moves along. Security is expensive and you need to know exactly what it is that you want to secure. Some things are obvious. For example, if you are using any encryption, the keys are security assets. Usernames and passwords are also entities that you want to protect. Others are not so obvious. For example, for some systems (like medical systems), the names of the clients and their use of the device need to be secured because of HIPAA. In other products, who is using it and how is being used is not something that needs to be protected.

Also, not all items need to be secured at the same level of security. I would recommend

Threat	27033-6 Clause	Description
Unauthorized Access	7.2	An unencrypted Wi-Fi or Bluetooth network would allow SSID's, usernames, passwords and other access information to be obtained.
Packet Sniffing	7.3	Unencrypted Wi-Fi networks allow a packet sniffer to eaves drop on all traffic. Special concern should be given to installation or factory reset conditions. 3G and 4G networks are somewhat secure without encryption – but rogue cell towers and devices used by law enforcement like the Stingray family of devices open the flood gates to packet sniffing on all unencrypted networks.
Rogue Wireless Access Point	7.4	For the LVAD heart pump we designed, we had a Wi-Fi access point located on the patient's device. This allowed doctors and health technicians to log into the device. A rogue wireless access point could be an evil twin and be used to obtain credentials and other access information. In 2014, a security company published that they had discovered more than a dozen rogue cell towers across the US. Rogue femtocell network extenders are other possible paths for an attacker.
Denial of Service Attack	7.5	We documented an unusual denial of service attack in <i>Circuit Cellar</i> June 2016 (Issue 311). Denial of service can happen both with jamming and overloading. We also mentioned in that article that one of our design's batteries could be depleted too soon by flooding the device with unexpected traffic.
Bluejacking	7.6	Bluejacking occurs when an attacker sends an unsolicited message to a Bluetooth enabled device.
Bluesnarfing	7.7	Bluesnarfing exploits vulnerabilities in the way Bluetooth is implemented on the device to obtain data from the device.
IMSI Catching	7.8.2	One of our goals when we create symmetrical key encryption is to provide a unique key for each device. A convenient way to do that is to have one private key that is modified by the IMSI number. Since there are devices that can capture the IMSI from your device, this creates a threat should the private key be disclosed or discovered.
Device Tracking	7.8.3	A possible threat is an attacker who is able to track your device based on the wireless network.

something similar to the five levels of qualitative severity in risk management for medical devices [16]. **Table 1** provides a possible list of severity levels. In addition to the severity level, following the 14971 standard, a table of semi-qualitative probability levels should be created. **Table 2** provides one such table. And then, finally, combine this into a matrix generating a probabilistic severity index as shown in **Table 3**.

So, we have the first element on the checklist: Create a list of assets to be secured—including a quantitative index as to the probabilistic severity of the risk should the asset be lost or compromised.

THREAT CHECKLISTS

Once you have established your first element on the checklist and created the list of security assets with their probabilistic index, we need identify all of the threats to each asset. This is an area that requires

a significant amount of experience and expertise. Experience comes when you discover a security breach in an existing product that caught you flat footed. (Of course, this never happens to us!) Expertise comes through research into threats that you never imagined. *Circuit Cellar* has published a number of articles by several of authors that can add to your expertise. The ISO/IEC standards and the Common Criteria can provide some help [17]. But generally, these are going to lag the security breaches in the field. For example, ISO/IEC 27033-6:2016 provides a list of threats for wireless networks. **Table 4** provides a summary of these threats.


Security is complicated. But so is surgery and flying a new airplane. Checklists can help. Next time we will start developing ours in earnest. But of course, only in thin slices. 

Table 4

ISO/IEC 27033-6: Threats to Wireless Networks