EMBEDDED IN THIN SLICES

The Internet of Things (Part 7)

DoS Attacks with a Twist

Bob continues investigating real world security vulnerability and what we can learn from it as we design our Internet of Things devices. This month he focuses on uncommon DoS attacks.

By Bob Japenga

Last fall, I was driving to pick up my grandson for an evening of fun watching the first game of the post season with our beloved Cubbies. We were going to miss the first hour and I had forgotten to start the DVR to record it. So my wife did it remotely from my phone. Wow! I love that Comcast provides that feature.

Then this week I started having second thoughts about this. It was announced that Comcast's home security system was vulnerable to a pretty simple attack that would leave your home completely unprotected. Completely. I starting thinking. If I can tell my DVR to record a show remotely, that means there may be a path from the general Internet to my home computer network. Is it secure? Hmmm! Read on and decide for yourself.

This month we will look at a form of the Denial of Service (DoS) attacks we mentioned last time. This time with a twist. The Comcast vulnerability could be triggered by jamming the RF security network used in the home security system. The past two articles we have picked some security vulnerabilities that were low hanging fruit. Leaving ports open and unauthenticated and hard coded passwords is pretty simple stuff. I think the next big actual security attack will be not be as obvious as this. I will be something outside the box. And probably quite simple. Comcast's susceptibility to a security vulnerability seems real obvious now and very simple to cause. But remember, these are complex systems designed by real people and real organizations who can miss obvious things. Jamming creates a whole slew of options for generating security vulnerabilities that we probably never thought of. Hopefully, this article will get you thinking outside the box about the security of the IoT systems you design.

STEALING YOUR CAR

You may remember the scare that went around the Internet a number of years ago. I got an email from a friend about how people are lurking in shopping mall parking lots and grabbing the security codes that you use to unlock your car with your FOB. I went to my trusty source of rumor debunking (www.snopes.com) and found that early automobile Remote Keyless Entry systems in the 1980s were indeed susceptible to such a steal. But the automotive industry quickly changed to rolling random codes in their wireless communication between the car and the FOB. The code changes every time the FOB is used. A repeated key code is rejected. Brilliant. So in the words of Snopes.com: "It is theoretically possible for a thief armed with the right technology and the ability to manipulate it correctly to snatch a modern keycode from the air and use it to enter a vehicle."[1]

Thanks to jamming, a very clever hacker has made the theoretical real and found a way to grab anyone's code. And he sells it for $30. Let's reopen that Snopes file! Here is how that works: If this new "theoretically possible" device is near your car when you try to unlock it, your first attempt will fail. But that won't surprise us because we frequently press the button twice because it didn't work the first time. We pushed the wrong button. We are out of range. When we push it a second time, it works. In the meantime, this clever device has recorded the code from your first and second push.

How does it do it you ask? Jamming. During your first attempt to unlock your car (see **Figure 1**, Step 1), the device jams the signal with two radios that transmit RF on the two common frequencies used by cars and garage door openers. That is why your door didn't unlock the first time. While transmitting a pair of jamming signals (preventing your car from hearing it), the device listens on a third receiver's radio blocking the jamming signal and records your first code.

You try to open the door again (see Figure 1, Step 2) and this time the device jams the signal again and stores the second code. The car doesn't get your second push. Then the hacking device sends the first code that it stored which unlocks the door. Now the device has squirreled away a code that will unlock your car or open your garage. Gosh these hackers are sharp!

For over 20 years this technology has been considered "secure" by the auto industry and the public. Now through a very clever technique using jamming, there is a device that can capture the codes to most of the cars and garage door openers made.

BREAKING INTO YOUR HOME

How does this relate to the Comcast vulnerability? Let's look at what happened. Comcast markets a security system for your home which monitors the opening and closing of windows and doors through some battery operated and wireless sensors including a motion detector and a camera. It can also be used for home automation. The base station communicates to the sensors over ZigBee. A security consultant and his team from a company called Rapid7 checked the system for security vulnerabilities. They found security holes that allowed an intruder to fool the system such that it would not report the opening of doors and windows. It would in fact, report that everything was secure when in fact it was not.

Rapid7 found that "all it takes to open windows or doors without detection is interrupting the 2.4-GHz radio frequency band used by the base station to communicate with window and door sensors. Rather than alerting the user to a change in state of the security system, the ZigBee-based system continues to report that the sensors are intact, doors are closed, and no motion is detected—while any movement in the doors remains unmonitored."[2]

How did they interrupt the signal from the base station? Jamming. It seems that the system is designed with a flaw known in the industry as "Not Failing Securely." In fact, this flaw is classified as Common Weakness Enumeration (CWE) 636 on the MITRE we site. (By the way, they have done a great job "naming the trees" in the forest of security vulnerabilities. I highly recommend that you become familiar with it.). MITRE defines CWE 636 as follows: "When the product encounters an error condition or failure, its design requires it to fall back to a state that is less

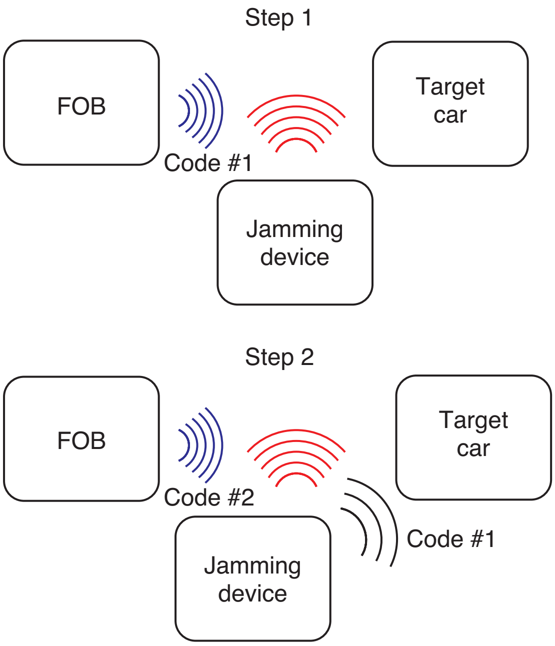

FIGURE 1
Step 1: The owner unlocks the car the first time. The jamming device detects Code #1, records it, and jams the reception of the car. Step 2: The owner unlocks the car the second time. The jamming device detects Code #2, records it, jams the reception of the car of Code #2, and sends Code #1 to the car.

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. With 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.

secure than other options that are available, such as selecting the weakest encryption algorithm or using the most permissive access control restrictions."[3]

That's exactly what happens with the Comcast "security" system. If you jam the ZigBee communications between the base and the sensors, the base station fails with the system wide open to an undetected intrusion. The base station thinks everything is warm and cozy. It thinks that the perimeter is unbreached.

There are other problems uncovered in the report. (It takes a long time for the sensors to come back on line after jamming; the sensors don't report the history while they were offline so you never know that the break-in occurred; and the fact that Comcast provides a sign for you to put on your lawn to tell potential hackers where to invade!) But we want to concentrate on jamming during this article.

WHAT CAN WE LEARN?

The folks from Rapid7 suggest two things that could be done. They suggest that the base station should issue some kind of low level alert when it loses the connection to the sensors. The other thing they suggest is that the sensors keep track of what happens while the connection was lost. I agree with the second suggestion, but know how hard it is to implement the first. Let's see if I can illustrate this.

A number of years ago, a neighbor installed a security system. It so frustrated the local police with false alarms that they required him to disconnect it. Nowadays, many municipalities have ordinances against this. Therein is the challenge for designing secure IoT devices that get every positive failures right and don't have any false positive failures. False positives will get you booted out. If the Comcast system has too many low level alerts, you will always ignore them. I don't know if there is an algorithm that Comcast could have used to prevent false positive alerts. My experience with ZigBee is that it can be flakey because it uses so low power. These devices are battery powered so Comcast had to minimize power usage and thus minimize traffic. We are doing that now with an IoT device that has to run for 7 years outside with 6 AA batteries. There is a lot of engineering that goes into optimizing RF power usage, the number of transmissions per day, the issue of retries, and so on. And that engineering requires trade-offs. It appears that Comcast erred on the wrong side of the trade-offs.

I know the people at Comcast are smart and perhaps they can implement Rapid7's first solution. But let it be said by me that it ain't easy to prevent false positives. That said, it is a problem they must solve.

Concerning the second suggestion, it may work after the fact but it is not the first line of defense. The enemy is over the wall (literally) and by reporting that a breach happened we are just boiling the hot oil. And by time they stop jamming, they are gone. Nonetheless, this is a must have. As I said in the first article, we need a multilayered defense. This is one. Not just logging history but knowing what critical things need to get recorded and reported while we are disconnected from the host. This takes careful systems design.

Are there other things we can do? Clearly Rapid7's first suggestion does the trick in the ivory tower world where there are very few low level alerts. But as designers we need to be smarter than that. What do we know about the cause of the disruption? Can we detect jamming? Can we tell the difference between an intentional jamming signal and the normal disruptions? Can we choose ZigBee chips for the base station that can

circuitcellar.com/ccmaterials

REFERENCES

[1] D. Mikkelson, "Lock Stalk: Warning About Thieves Using 'Code Grabbers' to Record Remote Keyless Entry Signals," 2014, www.snopes.com/autos/techno/lockcode.asp.

[2] E. Chickowski, "XFINITY Security System Flaw Allows Sneak Attacks By Jamming Radio Frequency," 2016, http://www.darkreading.com/informationweek-home/xfinity-security-system-flaw-allows-sneak-attacks-by-jamming-radio-frequency/d/d-id/1323769.

[3] MITRE, "CWE-636: Not Failing Securely ('Failing Open')," Common Weakness Enumerations, https://cwe.mitre.org/data/definitions/636.html.

RESOURCES

C. Thompson, "A Hacker Made a $30 Gadget That Can Unlock Many Cars That Have Keyless Entry,"
Business Insider, 2015, www.techinsider.io/samy-kamkar-keyless-entry-car-hack-2015-8.

report this information? We need to have clear specifications about false positives and false negatives. What is acceptable? Then we need to test it to death.

Finally, I would say that we need to think more outside the box. I find that when we have a problem, our engineers do a great job brainstorming every possible way that we can solve it. But we need to radically push the "What if…" questions. Exposure to articles like this is a good way to start thinking outside the box. Keep subscribing to *Circuit Cellar*!

Did the automobile industry ever think about that the jamming technique used by the code grabber? Think how many smart people are in that industry. No one thought of this for years. What about in your IoT devices? What if someone jammed your device? How would it react? Writing this article got me thinking about the design I mentioned earlier. We have algorithms for a retry mechanism that are based on real world (without jamming) scenarios. But if you jammed our device 24/7, I think the battery would wear out 70% earlier because we would be retrying far more often than we predicted would happen in the real world. This device has to live on the street for seven years in major metropolitan cities. In our case, the jamming would not be done to harm our device or to break security, but perhaps the police would intentionally jam all cell phones during some period of time (during a riot or a demonstration). Our unit wasn't designed to run seven years in that jamming scenario. What I am saying is: Work hard to think like a hacker and think outside the box. Jamming is just one scenario. Think about what intentionally jamming would do to your IoT device. Would it force you into some unexpected mode? Would it cause some security breach? Would it cause the system to wear out prematurely? Have you tested it in that mode?

GET SECURE

Security is very hard to build into any product. It takes careful thought and hard work. This month we saw how one simple technique completely disabled a security system. It is easy for us to Monday morning quarterback these flaws, but designing secure IoT devices is not for the faint of heart. We will continue to look at some more real problems next time—but of course, only in thin slices.