Luis Garcia, Tim Herrmann, Joe Maiocco

In July 2020, researchers at a lab working with Tencent reported that they had discovered a method (labeled BadPower) that manipulates fast charger firmware in order to harm connected devices.[1] Specifically, BadPower corrupts the firmware of these types of chargers so that they supply more power to the devices they are charging than the devices can normally withstand. Rather than communicating with the plugged-in devices to establish a safe voltage boundary, the infected chargers deliberately increase the voltage passed into the devices, forcing the device components to melt or ignite. BadPower can infect chargers through two attack vectors. First, an attacker can connect a targeted charger to a special firmware-modifying setup that causes the corruption. Second, attackers can load malicious code onto the devices themselves, and these devices transfer this code to the chargers during the charging process.

The Tencent research team experimented with a handful of fast chargers available to consumers and tested how these chargers would react when they came into contact with the BadPower exploit. Out of the 35 fast chargers tested, roughly half were vulnerable. The team determined that the risk of BadPower infection could be mitigated somewhat if the firmware was updated. However, they also noted that updates were not always an option for consumers. Along with the fast charger models that they tested, the team also checked out the chips that were central to the designs of the fast chargers themselves. Roughly half of these chips did not support any option for updating, which was a massive oversight on the part of the vendors who created and supplied them. In the end, analysts suggested that both the firmware for the fast chargers should be made further inaccessible to potential attackers and that charging devices should be better protected from any overloads.

A handful of hardware design decisions facilitated the unfortunate exploitation of the flaws discovered in the fast charger models tested by Tencent. Many of these flaws

[1] "BadPower attack corrupts fast chargers to melt or set your device on fire" by Catalin Cimpanu. July, 2020.

are analogous to potential dangers present in the development and maintenance of software as well. Just as two or more hardware subsystems create a larger overall system, two or more pieces of software can connect to form a complex system. Larger systems of either hardware or software offer more services and have the capacity to perform a greater variety of tasks. However, the interconnectedness of system components also increases the number of internal operations that may go wrong. Even though the BadPower vulnerability was found directly in the fast chargers' firmware, it had the greatest impact on the connected charging devices. In a similar example from the realm of software design, poor text filtration in website text fields may not be an immediate issue for the website itself. Yet slight oversights may allow attackers to use SQL injection to cause major damage to the website's accompanying database(s). One flaw in one system component can cause the whole system to malfunction.

The Tencent fast charger report also illustrates how threats do not just exist in pure hardware or pure software systems. Technical systems composed of both hardware and software possess the greatest number of risks. These complex types of systems typically offer potential attackers more vectors from which they can launch their assaults. The fast chargers that the researchers experimented with could be compromised either initially through the firmware itself, or by infecting devices that would later pass on the bug to the chargers. Going back to the example of SQL injection for comparison, hackers could attack databases through the website as previously mentioned. Alternatively, if these malicious actors knew the location of the site where the physical databases were being maintained, they could launch a direct attack on the supporting hardware there, especially if they had connections to someone with legitimate access to the site. Thus, software developers and engineers must recognize the threats posed to the systems they design and subsequently put preventative measures in place to decrease the likelihood that their vulnerabilities will be exploited.

Another major flaw in the fast charger models that the team at Tencent tested was the fact that many of the chips could not be updated. Normally, new versions of hardware are released, firmware is updated, and software is patched in order to improve upon previous iterations. Perfect finished products do not exist and never will; functional products do exist and are high in demand. As such, fixing additional problems as they arise after a product has shipped has become a popular way for developers to handle bugs and vulnerabilities. Patched software can safely be used by consumers until hackers discover new exploitable aspects, which then results in more patches being released. The exchanges back and forth between developers and hackers may seem frustrating and pointless in the long run, but actually the opposite holds true. Updatable systems survive in the market for much longer. Given this reality, the fact that some of the tested fast charger chips could not be updated is even more astounding. Products may ship with flaws, but these flaws *must* be addressed as soon as possible in order for the products to remain as viable choices for customers. Designers and developers who fail to build systems that can adapt in the face of failures basically choose to lose their customer base over time.

This last point lends itself to some of the supporting concepts of software development: developers want to deliver applications to customers that are needed and timely. Customers, both in the contractual sense and as general consumers, want products for long-term use. Very few real-life scenarios require software (or for that matter, hardware) that exists for a brief period of time and then gets thrown away forever. Instead, the systems that engineers produce must serve constituents and endure for many months and even years. Thus, systems must be capable of evolving over time so that they can fully respond to the new needs of constituents, and so that they can reduce the likelihood of large-scale malfunctions caused by any attacks from malicious actors or by the undesired side effects of certain internal components. Any system that fails to do so should not last in the market for long.