Pipeline
Technology for Service Providers.
www.pipelinepub.com Volume 15, Issue 4

IoT Security: Down to Fundamental and Up to C-Level

By: Erez Kreiner

Cybersecurity has become a major factor in the risk calculation of almost any company, no matter if it is a large enterprise or small home business. Various solutions are offered by many vendors to overcome the security gaps in companies' networks and devices, and a lot of effort is invested in searching for the "holy grail" of security, the one that will supply a fully protected environment for all devices.

As we all know, this search will probably last forever, as hackers become more sophisticated and rogue nations pool their resources to maliciously attack the larger global community and economy. We can continue to combat the latest threats with the latest cybersecurity solutions, but it's becoming increasingly more important to analyze the situation from the other side—the side of the attacker trying to hack into a system, or actually place his malicious code in a persistent manner inside an organization's network, devices or machines.

The hacker's desire is to change the software and data that is stored in the memory and his "holy grail" is to change the firmware, which is a fundamental brick in any computing mechanism. Computed devices are made of chips, which are millions of electronic circuits that can be condensed into very small areas, creating chips of memory, CPUs, GPUs, or communication, which is commonly called hardware. To make these chips a living and breathing computer device, software is fused in. The lowest software level that connects it to the hardware is called firmware. In most—if not all—cases firmware is a piece of code that is not subject to change by anyone besides the vendors and, many times, firmware is rarely changed at all. When firmware is updated, however, it presents an opportunity for attackers to place malicious code within the firmware, as few security organizations are thoroughly protecting firmware over the air (FOTA) updates.

Attackers try to gain control of networks and devices in many ways, but all the attacks can be categorized into three main vectors:

1. **Remote attack**—the attacker will probably make use of an Internet connection and he does not have any physical contact with the target.
2. **Close attack**—when the attacker has some kind of physical contact with the target himself or, through proxies—such as a connected thumb drive—has access to the company network or to communication equipment, etc.
3. **Supply chain attack**—when the attackers take advantage of the relationship between the company and its business partners and slide the malicious code into a product of a third party, a product that is later supplied to the target.

The common denominator for all these attack vectors and the attacks that utilize these vectors is the attempt to modify the code that runs on the system, either by modifying the code itself or by modifying the parameters—through configuration and calibration—that affect the way the code is executed. In addition, changing and "playing" with the firmware can ensure the attacker that his malicious code would have a long-lasting life in the targeted systems, and that the attempts of the security components to discover its existence will, in most cases, fail.

Focusing on firmware is a result of the huge revolution we all experience now, the IoT or IIoT, where the basic idea and basic meaning is to connect all sorts of devices to the Internet, especially devices that do not include powerful CPUs. Most CPUs are built with poor resources and limited computing power. In these simple devices, the software is mainly or solely firmware.

To make it a bit more colorful, the list of connected devices includes industrial robots, cars, home machines like air conditioners, routers, and 95 percent of the equipment in any new, smart or semi-smart building. It includes the protected relays in the electricity grid and the smart meters attached to every home. The list goes on and on…

One can almost claim that any electric or electronic device will include a few electronic chips, with a non-volatile memory containing the critical code to its operation.

Out-of-the-box approach

The connected device's ecosystem and its attacker landscape have several key components which make them vulnerable to hackers:

a. Most security processes are handled within the CPU.
b. The CPU has many interfaces to many components.
c. Security software vs. hacking software is an endless circle that needs to be broken by introducing another dimension to the game.

A new approach is needed to address these concerns and protect these devices and should include:

1. Defending the non-volatile memory.
2. Controlling write/read attempts, independent of the host CPU or OS.
3. Ensuring a secured channel for content updates.
4. Deploying a solution that is agnostic to the CPU, OS and memory brands.
5. Creating a management platform that is able to securely monitor updates.

To create this breakthrough in security, organizations must understand security technologies and the gaps that keep CIOs and CEOs awake at night. Understanding the business needs of enterprises and companies will lead to the development of the technology from the embedded devices to the management platform and create tools to provide information consumed by many key players from IT managers up to C-level executives.

Focusing the efforts on defending the non-volatile memory is the outcome of the fact that the "holy grail" mentioned above remains the main target for attackers. Attackers want their attacks to be persistent, to stay in control of devices and networks, and to easily be hidden. They also want to easily manage their future attacks.

If an authorized party can control the write and read lines, it avoids any capability to manipulate the data or the code stored inside the memory device. A software-only security solution, even if very sophisticated, trying to overcome the security gap can be compared to Bobby Fisher trying to win a basketball game. CISOs need to do more than apply common methods to protect content or the firmware through encryption for example, as encryption cannot protect against attempts to destroy the data.

What is needed is a truly innovative security approach process in which various components run in the memory itself while the management platform runs in the company's secured area, taking advantage of its full capabilities. Each flash-enabled device self-registers to the management platform during its first operation using a unique un-cloned key. Thus, if even one end-device (or many) is breached—a huge task by itself—there is no impact on other devices, which remain secure.

This solution should protect the root of trust between the cloud and the device, from provisioning time throughout the device's entire lifecycle and after, ensuring that only an authorized entity can update and change the device's critical elements.

It's important that any security protecting IoT devices from embedded to cloud contain the following features:

- Protecting endpoints with limited resources

- Interfacing to external management systems
- Securing and validating new content, including firmware, data and software
- Fully backward compatible
- Providing ironclad security
- No latency
- Working with all CPUs and all OSs, and CPU agnostic
- Protecting CPU “takeover”
- Securing FOTA updates
- Protecting systems from reverse engineering

If organizations focus on protecting the persistent memory, recent famous attacks could have most likely been prevented. If the device’s flash memory was protected, security flaws like VPNFilter and Mirai would not exist. And these security flaws are damaging to organizations with IoT devices. For example, the Mirai malware changed code in security cameras, routers and other sorts of connected devices, turning them into bots in a botnet that was later utilized in attacking Amazon, Twitter, Spotify, DYN and many others. There’s also an issue with security flaws such as Meltdown and Spectre, as these vulnerabilities demonstrate a fundamental flaw with CPU design. While chips vendors have sent software patches to rectify the security issue, these patches will have limited results against current and future breaches resulting from internal design flaws, coding errors and external hacking, all of which still have huge implications for a number of connected devices from the medical field to smart cities. If the firmware of the said routers or cameras had security built in or on top of the persistent memory, then the content could not be changed and could only be updated and managed by the organization’s owner.

Organizations need an end-to-end, embedded-to-cloud solution for managing, protecting and firmly securing IoT and connected edge devices, an approach that prevents all attack vectors from overwriting, modification, manipulation, and erasure of memory content. Until then, we’ll never find the “holy grail” of cybersecurity protection.

==========

NanoLock Security is an Israeli start-up with an innovative, out-of-the-box approach and technology in the arena of managing and securing connected and IoT devices. NanoLock has offices in New York, Israel and Tokyo.