



[Home \(https://embedi.com\)](https://embedi.com/) / [Blog \(https://embedi.com/blog\)](https://embedi.com/blog/) /

[Analytics \(https://embedi.com/blog/categories/analytics/\)](https://embedi.com/blog/categories/analytics/) / Killchain of IoT Devices. Part 2

Categories:

➤ [Analytics \(https://embedi.com/blog/categories/analytics/\)](https://embedi.com/blog/categories/analytics/)

➤ [Research \(https://embedi.com/blog/categories/research/\)](https://embedi.com/blog/categories/research/)

Tags:

[#ATM \(https://embedi.com/blog/tags/atm/\)](https://embedi.com/blog/tags/atm/) [#CISCO \(https://embedi.com/blog/tags/cisco/\)](https://embedi.com/blog/tags/cisco/)

[#cybersecurity \(https://embedi.com/blog/tags/cybersecurity/\)](https://embedi.com/blog/tags/cybersecurity/)

[#D-Link \(https://embedi.com/blog/tags/d-link/\)](https://embedi.com/blog/tags/d-link/) [#DJI \(https://embedi.com/blog/tags/dji/\)](https://embedi.com/blog/tags/dji/)

[#exploitation \(https://embedi.com/blog/tags/exploitation/\)](https://embedi.com/blog/tags/exploitation/)

[#firmware-security \(https://embedi.com/blog/tags/firmware-security/\)](https://embedi.com/blog/tags/firmware-security/)

[#hardware \(https://embedi.com/blog/tags/hardware/\)](https://embedi.com/blog/tags/hardware/)

[#hijacking \(https://embedi.com/blog/tags/hijacking/\)](https://embedi.com/blog/tags/hijacking/) [#intel \(https://embedi.com/blog/tags/intel/\)](https://embedi.com/blog/tags/intel/)

[#Microsoft \(https://embedi.com/blog/tags/microsoft/\)](https://embedi.com/blog/tags/microsoft/) [#mobile \(https://embedi.com/blog/tags/mobile/\)](https://embedi.com/blog/tags/mobile/)

[#Office \(https://embedi.com/blog/tags/office/\)](https://embedi.com/blog/tags/office/) [#RCE \(https://embedi.com/blog/tags/rce/\)](https://embedi.com/blog/tags/rce/)

[#router \(https://embedi.com/blog/tags/router/\)](https://embedi.com/blog/tags/router/) [#SCADA \(https://embedi.com/blog/tags/scada/\)](https://embedi.com/blog/tags/scada/)

[#vulnerabilities \(https://embedi.com/blog/tags/vulnerabilities/\)](https://embedi.com/blog/tags/vulnerabilities/)

Popular articles:

[Killchain of IoT Devices. Part 2 \(https://embedi.com/blog/killchain-iot-devices-part-2/\)](https://embedi.com/blog/killchain-iot-devices-part-2/)

[Killchain of IoT Devices. Part 1 \(https://embedi.com/blog/killchain-iot-devices-part-1/\)](https://embedi.com/blog/killchain-iot-devices-part-1/)

22 June, 2017

Killchain of IoT Devices. Part 2



Category: Analytics (<https://embedi.com/blog/categories/analytics/>)

Tags: #cybersecurity (<https://embedi.com/blog/tags/cybersecurity/>), #hardware

(<https://embedi.com/blog/tags/hardware/>), #vulnerabilities (<https://embedi.com/blog/tags/vulnerabilities/>)

Thousands of security incidents related to embedded-devices show us that devices do not provide the security level we can rely on. Sure thing, manufacturers use some technics like secure boot, firmware signature, etc., but mostly they take such measures only for expensive and enterprise devices. Just think: what kinds of security mechanisms are implemented in smart-DVRs, locks, TVs, routers? How many of them verify firmware before start working? Moreover, there are a lot of developers who don't use security development lifecycle (SDL). Consequently, manufacturers can admit lots of mistakes in their products and enable violators to use other but still simple techniques to conduct an attack. Hence, customers have no choice but to buy vulnerable devices and being hacked. Understanding implementation of various attacks aimed at these smart-gadgets, allow both a manufacturer and a consumer to make our world safer. So, last time we started our discussion about security of embedded devices and the kill-chain model. We explained why it is important to know what the kill-chain model is and discussed first three steps of it.

Now we are at the most crucial step of an attack – exploitation. At the exploitation step of the kill-chain model, we have a huge range of vulnerabilities to deliberate on.

One of the most exploited vulnerabilities in smart-devices is a weak password policy or a complete lack of password protection, which is even more common. Sometimes users simply forget to set strong passwords, though everyone has heard it helps to increase security level. Nevertheless, some manufacturers may forget to delete their test accounts or create backdoors, so there is no guarantee that a user has the only admin account.

From time to time, researchers discover these backdoors, like it happened with Huawei HG8245 router that contained two administrator accounts enabled by default. Consequently, cyber-criminals collect both the most popular passwords that are set by users and passwords set by manufacturers for default accounts. No wonder, it will be as easy as a piece of cake for an attacker to conduct further attacks having all the credentials.

According to the facts listed previously, at the weaponization step (/analytics/killchain-of-iot-devices-part-1/), we can't use traditional security measures from the PC-world. In best cases, developers manage to adapt traditional ways of protection, but they can't do it for everything (for example, common antimalware systems require a lot of memory space and computing capacity).

This fact directly affects security and cryptography capabilities of embedded devices. Smart-gadgets usually provide lightweight symmetric cryptography algorithms. So, devices are vulnerable to eavesdropping at the pairing parts: protocols, based on symmetric cryptography, and imply keys exchange for creation a session key. Hence, an attacker can influence this process even worse.

Another problem is lack of cryptography. It may sound like a joke, but there are lots of smart-devices that have a poor communication security: devices send confidential information in plain-text, like login/password, location, names of used services, photos and so on.

Also, devices may provide poor authentication. This vulnerability type may be caused by logic problems or by poor cryptography. Smart-gadgets may have no capabilities to distinguish legal from non-legal users, and therefore everyone can obtain secret data or get control capabilities without any efforts just by connecting to device ports, sending SMS to any services or other methods. For example, manufacturers may unintentionally allow intruders to get access over a smart-device via control panel by connecting to 192.168.1.1:80 with admin login and no password.

From time to time, some software parts of embedded devices should be updated. The way updating process is implemented by a manufacturer can be extremely inconvenient for users, who neither have time or inclination to update their devices nor have the skills required to do this. Consequently, users reject the update process completely. Also, manufacturers either don't release new firmware updates or just don't provide this feature, therefore users may be breached, thanks to good old, vulnerable software.

Another side of this problem is a verification of incoming updates. Cyber-criminals may attempt to release their malicious updates that can compromise smart-gadgets. That is why manufacturers should implement validation verification services for incoming updates.

The problem similar to verification of updates is verification of firmware: there are security services that allow controlling a device booting process and protecting it against bootkits and other malware. This security technique is nothing new for the world of IT: TPM and TXT produced by Intel for PCs enable checking integrity of a device firmware and storing keys securely. So, secure boot increases manufacturers' and consumers' confidence, that smart-devices are based on trusted firmware created by the manufacturer and that nobody can install any malware, for example, at the transportation stage. Unfortunately, secure boot technique for smart-gadgets isn't widely available and, consequently, consumers can't trust even unboxed devices.

Again, as embedded devices can be placed outdoors, an attacker can directly interact with their hardware. Embedded-devices have some pins and interfaces that can be used by an attacker. For example, these interfaces may allow everyone to interact with a device firmware or other important constituents through them. Hence, these interfaces could give full control over devices to intruders.

Also, an attacker can use some unorthodox methods over hardware constituents. Heating or cooling chips may lead to unpredictable results. One of the latest news shows us, that hackers can fool accelerometers with specially recorded sounds.

Intruders can even use device resources for their own needs by stealing IMEI, SIM card information, and others. Some devices store information in plain-text. Therefore, if devices are reversed, attackers will be able to learn a user's e-mail address, Wi-Fi password and any other important information and use it in their further attacks.

Unfortunately, devices may suffer from denial-of-sleep or denial-of-service. Long-term ping of a smart-device by an attacker can knock it out or reduce its capacity.

We covered ways of exploiting devices that are new to traditional security approaches. Nevertheless, it would be wrong to remain silent about well-known, but most exploited and still dangerous vulnerabilities. According to Ponemon Institute research, 80{69c90e256be360ad980ca26f621f2003f22fc8a173440dabd89573a4d6bbc248} of applications in the IoT world aren't tested for vulnerabilities. Consequently, almost two out of three applications have at least one vulnerability from OWASP TOP 10 list that we are going to discuss soon. Mostly, applications have a poor input data validation and intruders try exploiting these security flaws. For instance, TP-Link M5350, a 3G/Wi-Fi router, has a XSS vulnerability, that enables a remote attacker to get admin credentials by sending a simple SMS, that contains script. You can find lots of free pentesting tools that enable finding the most common vulnerabilities. Intruders can be extremely low-skilled and even don't know how hacker stuff works, but their aims will be hit in two clicks. Another example is buffer overflow vulnerability that allows cyber-criminals to change process' address space and to escalate their privilege or crash the application process. All in all, attackers have cut their teeth on these attacks and therefore there is no doubt they will use their honed skills on smart-gadgets as well.

Summarizing this step, we should mention, that even for an advanced user maintaining smart-gadgets in a secure state could be much harder, than a traditional PC. Sometimes it's just impossible to configure devices safely because manufacturers don't give you such methods (devices have no screen or have limited settings, etc). Some devices don't provide features that allow you to explore or even navigate in their filesystems, therefore, malware doesn't even need to be hidden. Also, some devices have special security services that are disabled by default and users don't know anything about security potential of their gadgets.

Other steps of the IoT killchain model are similar to the traditional killchain model and depend only on an attacker's aim. Cyber-criminals can hack your device to spy on you and rob your house. What is even more dangerous, several hacked devices, if used together, may provide more detailed information about their owner,

thus, leading to blackmailing. Mirai shows us, that, if combined, compromised devices can shut down DNS servers by conducting continuous DDoS attacks. Consequently, hackers will be even able to kill people by using smart-gadgets, as it was with a malfunctioning robot in 2015.



(<https://www.linkedin.com/shareArticle?mini=true&url=https://embedi.com/blog/killchain-iot-devices-part-2/&title=Killchain of IoT Devices. Part 2>) (<https://twitter.com/embedi/status/988888888888888888>)

f Part 2
() 2) 2)

Subscribe to our newsletter to stay in touch






Enter your email here

Submit

Solutions (<https://embedi.com/solutions/>) Blog (<https://embedi.com/blog/>)

Our news (<https://embedi.com/news/>) Resources (<https://embedi.com/resources/>)

About (<https://embedi.com/about/>)

 (<https://www.facebook.com/Embedi/>)  (<https://www.linkedin.com/company/embedi>)
 (https://twitter.com/_embedi_)  (<https://github.com/embedi/>)
 (<https://www.youtube.com/channel/UC9XR2noZynNxvQcg0G9ooKA>)

2016 - 2018 ©