Attacks against Embedded Systems. Threats and Solutions

Mario Bufu  
  
Univesitatea Politehnica TimisoaraTimisoara, Romania  
mario.bufu@student.upt.ro

*Abstract*—Embedded system are a core component in domains like automotive, industrial machines, healthcare. Ensuring security in embedded systems is necessary because some systems are used in safety systems, this security is hard to achieve and no matter how much countermeasures it is implemented, the 100% security cannot be guaranteed*. This paper provides some examples of attacks against embedded systems and some solution to decrease the possibility of a successful attack*.

# Introduction

An embedded system is a combination of CPU, memory, I/O interfaces and other optionally components like sensors, ADC, displays, power supply. Usually an embedded system controls physical operation, this requires that the embedded system to respond in real time to the data from outside, usually data captured by sensors. Embedded systems are commonly found in consumer, industrial, automotive, home appliances, medical, commercial and military applications. The industrial trend shows that embedded systems are important in many application domains, such as automotive electronics, avionics, consumer electronics, railways, telecommunications, and healthcare [1].

Security is very important because these embedded systems are often use in safety systems, malfunction or misbehaving will cause physical damage to people or other objects [2]. As the technology advances, many approaches have been proposed to secure embedded systems, but securing those systems is challenging due to various facts like resource limitation, cost, power consumption [3].

# Attack scenarios

In [4] is presented 5 dimension that will classify the attacks on embedded systems based on information from the public CVE (Common Vulnerabilities and Exposures) database. According to this the attacks against embedded systems ca be classified as follows: precondition, vulnerability, target, attack method and effect of attack.

For preconditions the following types of attacks are observed:

* **Internet facing device**: potentially exploitable by a remote attacker if the device is connected to the Internet. The attacker does not necessarily need to have access privileges; the only requirement is that the attacker can potentially discover the device and send messages to it via the network
* **Local or remote access to the device**: requires the attacker to have some privileges that allow for logical access to the services or functions provided by the device
* **Direct physical access to the device**: the attacker access the device physically
* **Physically proximity of the attacker**: attacks on wireless devices may only require to be within the radio range of the target device
* **Miscellaneous**: general category to represent each appearing in only one or a few records
* **Unknown**: some CVE records and other sources do not provide sufﬁcient amount of information to determine the preconditions of a potential attack

The vulnerabilities observed are:

* **Programming errors:** buffer overﬂow problems, and memory management problems such as using pointers referring to memory locations that have been freed
* **Web based vulnerability:** web-based management interface through which they can be conﬁgured and updated.
* **Weak access control or authentication:** many devices use default or weak passwords, and some devices have hard-coded passwords that provide backdoor access to those who know the hard-coded password.
* **Improper use of cryptography:** use of weak random number generators for generating cryptographic keys, or vulnerabilities in the protocols that use cryptographic primitives
* **Unknown**: some CVE records do not contain information about the vulnerability itself

For target we have the following types:

* **Hardware**
* **Firmware/OS**
* **Application**

Attack methods observed:

* **Control hijacking attacks:** divert the normal control ﬂow of the programs
* **Reverse engineering:** an attacker can obtain sensitive information by analyzing the software in an embedded device
* **Malware:** infect an embedded device with a malicious software
* **Injecting crafted packets or input:** injection of crafted packets is an attack method against protocols
* **Eavesdropping:** attacker only observes the messages sent and received by an embedded device
* **Brute-force search attacks:** weak cryptography and weak authentication methods can be broken by brute force search attacks. Those involve exhaustive key search attacks against cryptographic algorithms such as ciphers and MAC functions
* **Normal use:** attack that exploit an unprotected device or protocol through normal usage
* **Unknown**: Some CVEs that described vulnerabilities but did not identify any particular attack method.

The effect of the attacks can be:

* **Denial-of-Service (DoS):** malfunctioning or completely halting the device
* **Code execution:** execution of attacker supplied code on the embedded device
* **Integrity violation:** modiﬁcation of ﬁles and conﬁguration settings, as well as the illegitimate update of the ﬁrmware or some applications
* **Information leakage:** leakage of some information that should not be obtained by the attacker
* **Illegitimate access:** attacker gain privileges that put him in control of the system
* **Financial loss:** sending and SMS or making a call to a premium number from a compromised smart phone
* **Degraded level of protection:** a device is tricked into using weaker algorithms or security policies than those that it actually supports
* **Miscellaneous:** attacks cause users to be redirected to malicious websites or trafﬁc to be redirected
* **Unknown**: no speciﬁc attack effect is identiﬁed

A graphical representation of frequencies and path used for attacks can be found Fig. 1.

.

# Mechanisms against physical and side-channel attacks

The Federal information processing standard FIPS1402 provides four increasing qualitative levels of physical security requirements intended to reﬂect diﬀerent degrees of tamper resistance [3].

Security level 1 represents the minimum physical protection with production grade enclosure that may be removable; level 2 requires the use of tamper-evident covers or seals which allow the detection of a physical compromise, while level 3 mandates the use of more advanced tampering detection and response mechanisms, level 4 adds to the requirement of previous levels the condition that the system must include environmental failure protection (EFP) or undergo environmental failure testing (EFT).

Concerning side-channel attacks several prevention software and hardware approaches have been proposed. Software-based countermeasures include randomization of the instruction execution sequence, introducing dummy instructions, balancing Hamming weights of the internal data and bit splitting to name. On the hardware level, randomization can also be applied on the clock signal or the power consumption [3].

# Mechanisms against software attacks

Those mechanisms typically target objectives like:

* Ensure privacy and integrity of sensitive code and data during every stage of software execution in an embedded system [3].
* Determine with certainty that it is safe from a security standpoint to execute a given program [3].
* Remove security loopholes in software that make the system vulnerable to such attacks [3].

Software-based solutions:

* Secure bootstrapping: A common approach adopted by the proposed techniques in this area is to use the notion of trust boundaries across the system’s hardware and software components; this notion allows the detection of illegal accesses carried out by malicious software [3].
* Software authentication and validation: Every known application in the embedded system should undergo a validation step before its execution. This can be done through computing a hash of the application code and comparing it to a pre-computed golden value [3].
* Operating system and application enhancement: On system level it is recommended a strong process isolation so that the private resources of one process can be protected from another process; process-level attestation that authenticate the code before that it establishes communication channel with corresponding device. In the application level, adopted techniques include the use of encryption wrappers that allow dynamic run-time encryption/decryption of the software code to prevent static code analysis attacks; code obfuscation that consists of transforming the application’s code, so it becomes less intelligible to humans, and thus preventing reverse engineering [3].

# Conclusion

This paper provides an overview of what types of attacks were used in the past, what are the most approaches used and most vulnerable parts of and embedded software and some solution that could help to make an embedded system more secure. No matter how much countermeasure we use we cannot guarantee that an embedded system is 100% secure.

##### References

1. P. Marwedel, Embedded system design: Embedded systems foundations of cyber-physical systems. Springer Science & Business Media, 2010.
2. R. Langner, “Stuxnet: Dissecting a cyberwarfare weapon,” Security & Privacy, IEEE, vol. 9, no. 3, pp. 49–51, 2011.
3. Khelladi, Lyes & Challal, Yacine & Bouabdallah, Abdelmadjid & Badache, N.. (2008). On Security Issues in Embedded Systems: Challenges and Solutions. International Journal of Information and Computer Security. 2. 10.1504/IJICS.2008.018515.
4. D. Papp, Z. Ma and L. Buttyan, "Embedded systems security: Threats, vulnerabilities, and attack taxonomy," 2015 13th Annual Conference on Privacy, Security and Trust (PST), Izmir, 2015, pp. 145-152.