Embedded Systems Security: Threats, Vulnerabilities, and Attack Taxonomy

Dorottya Papp[*][†], Zhendong Ma[†], Levente Buttyan[*]

[*]CrySyS Lab
Budapest University of Technology and Economics, Hungary
{dpapp, buttyan}@crysys.hu

[†]Digital Safety & Security Department
AIT Austrian Institute of Technology, Austria
zhendong.ma@ait.ac.at

Abstract—**Embedded systems are the driving force for technological development in many domains such as automotive, healthcare, and industrial control in the emerging post-PC era. As more and more computational and networked devices are integrated into all aspects of our lives in a pervasive and "invisible" way, security becomes critical for the dependability of all smart or intelligent systems built upon these embedded systems. In this paper, we conduct a systematic review of the existing threats and vulnerabilities in embedded systems based on public available data. Moreover, based on the information, we derive an attack taxonomy for embedded systems. We envision that the findings in this paper provide a valuable insight of the threat landscape facing embedded systems. The knowledge can be used for a better understanding and the identification of security risks in system analysis and design.**

I. Introduction

An embedded system is a computing system built into a larger system, designed for dedicated functions. It consists of a combination of hardware, software, and optionally mechanical parts. Thus the term refers to any computing systems other than general purpose PC or mainframe computers [1]. Often they are Cyber-Physical Systems (CPS) due to the integration of computation and physical processes [2]. The industrial trend shows that they are the driving force in many application domains that build smart or intelligent systems, including areas such as automotive electronics, avionics, consumer electronics, railways, telecommunications, and healthcare [3].

Security is an important issue because of the roles of embedded systems in many mission and safety-critical systems. Attacks on cyber systems are proved to cause physical damages [4]. However, comparing to conventional IT systems, security of embedded systems is no better due to poor security design and implementation and the difficulty of continuous patching [5]. Although many approaches have been proposed in the past to secure embedded systems [6], [7], various facts such as deployment scale, resource limitations, the difficulty of physical protection, and cost consideration all make it very challenging to secure them [8], particularly for devices with remote control, maintenance and operation functions.

Having a comprehensive view and understanding of an attacker's capability, i.e. knowing the enemy, is prerequisite for security engineering of embedded systems. Security analysis, secure design and development must take into account the full spectrum of the threat landscape in order to identify security requirements, innovate and apply security controls within the boundary of constraints. As a result, in the scope of embedded systems security, the following questions arise:

- What are the main causes of those successful attacks?
- What are the main vulnerabilities?
- What are the commonalities of the attacks?
- How can we use the knowledge to improve the security of embedded systems?

In this paper, we conduct a systematic review of existing threats and vulnerabilities. We focus on two sets of data, i.e., the exposures of attacks on embedded systems in security conferences and literature, and the published vulnerabilities specific to embedded systems. Based on the data, we derive an attack taxonomy to systematically identify and classify common attacks against embedded systems. We envision that the comprehensive knowledge of attacks and their implications will contribute to savvy design decisions for mission and safety critical systems.

In the following, Sec. II discusses the related work, Sec. III describes our method to conduct the review, Sec. IV surveys the known attacks and vulnerabilities. Based on the findings, Sec. V classifies attacks in a taxonomy specific to embedded systems. The correctness and feasibility of the taxonomy is evaluated in Sec. VI, followed by the conclusion in Sec. VII.

II. Related work

In [9], [10], [11], the uniqueness of embedded systems security and possible countermeasures to software and hardware attacks are elaborated. Various attack scenarios are also discussed. Vulnerabilities in general computer and IT systems are studied in [12], [13]. An empirical study focusing on embedded systems vulnerability is included in [14].

For attack taxonomy for general computer and IT systems, ENISA [15] maintains a list of existing incident taxonomies.

978-1-4673-7828-4/15/$31.00 ©2015 IEEE

Among them is the common language security incident taxonomy developed at the Sandia National Laboratories which divides an incident into attackers, tools, vulnerability, action, target, unauthorized results, and objective. [16] proposes a cyber attack taxonomy that classifies cyber attacks into attack vector, operational impact, defense, information impact, and target, abbreviated to AVOIDIT. Common Attack Pattern Enumeration and Classification (CAPEC) [17] defines structured description of IT security attacks. It organizes attack patterns into 11 categories, such as data leakage attacks, resource depletion, injection etc. Hansman and Hunt [18] proposed a four dimensional approach to attack taxonomy, including attack vector, target, vulnerabilities and exploits, and the possibility of having a payload or effect beyond itself. The information in each dimension is further described in several hierarchical levels of details.

For attack taxonomy for cyber-physical systems, Zhu *et al.* [19] provided a taxonomy of cyber attacks on Supervisory Control and Data Acquisition (SCADA) systems. Cyber attacks are classified according to their targets: hardware, software and the communication stack. The attacks on software are grouped into exploitation of embedded operating systems without privilege separation, buffer overflow, and SQL injection. The attacks on communication stack are classified into network, transport, and application layer, as well as the implementation of the protocols. In their identification of potential attacks on avionics embedded systems, Dessiatnikoff *et al.* [20] provided a categorization of attacks against onboard aerospace systems. Attacks are categorized into two major classes: attacks against core functions and against fault-tolerance mechanisms. For each subcategory, the authors provide examples and emphasize the impact of such attacks. Yampolskiy *et al.* [21] placed their emphasis on how cyber attacks might influence the physical space and proposed a taxonomy that could be used to categorize cross-domain attacks as well. Their proposed taxonomy has six dimensions, organized in three groups, i.e. targets, effects, and attacks.

Existing taxonomies do not cover the full spectrum of embedded systems. For example, [19] is created for SCADA systems and [20] focuses on onboard aerospace systems. While the taxonomy in [21] is intended to capture cross-domain effects of cyber attacks, it is a generic and abstract classification of the semantics of attacks without concrete attack information. In addition, the lack of structure in the taxonomy makes it difficult to efficiently process attack information. In our approach, we extend the existing attack taxonomies and modify their contents and structures.

III. Our Approach

The first part of our research was a systematic review of existing threats and vulnerabilities. To collect detailed description of attacks on embedded systems, we go through proceedings of computer security conferences with a focus on computer hacking such as DefCon and BlackHat, and follow the referenced white papers and links. We also search the Internet for media reports, blogs, and mailing lists related to attacks on embedded systems. Besides, we include scientific papers with a practical focus. The information obtained by this research is described in Sec. IV.

Due to researchers' specific interests, the cost incurred in security testings, and the non-disclosure agreement forced by the vendors or asset owners, the published attacks/hacks only reflect a fraction of the whole threat landscape. To gain a comprehensive view, we study the information on vulnerabilities related to embedded systems, which we consider as the other side of the same coin. Our main information source is the Common Vulnerabilities and Exposures (CVE) [22] database. CVE is the most comprehensive aggregation of security vulnerabilities. Each entry in the CVE database is assigned a standardized identifier which can be used to share vulnerability information across different organizations. At the time of our research, the database contains more than 60,000 entries, not all are related to embedded systems. We improvise several techniques on-the-fly to filter and extract relevant entries from the general vulnerability data, and manually analyze the selected entries. The result of the analysis is a set of attack classification criteria that serves as a basis for our attack taxonomy. The process and analysis are described in Sec. V. To validate our proposed taxonomy, we apply it to classify all embedded systems-related CVE entries since 2005 in an automated way. The results described in Sec VI confirm the validity of the constructed taxonomy.

IV. Attacks on embedded systems

This section lists some examples of attacks against embedded devices and systems and looks into the attackers' capabilities and their implications. Although not comprehensive, in our view, the examples are very representative and cover a broad range of application domains such as industrial systems, communications, and consumer devices.

[23] presents a timeline for critical infrastructure. Noteworthy attacks date back to the 1982 and the number of attacks have been increasing since 2001. [24] presents vulnerabilities and possible exploits of key management in wireless devices. For example, one of the devices is shipped with a graphical user interface with default values to configure the device. The implementation of the interface generates a passphrase which is later used to generate the AES key. However, the Pseudo-Random Number Generator is seeded by the `srand()` function using the current time and generator itself is the `rand()` function. As a result, the attacker is capable of calculating the passphrase and the encryption key and can intercept all communication on the target wireless network. [25] demonstrates remote attacks against SCADA devices using the ModBus protocol. The vulnerability exploited is within the design of the protocol: it lacks encryption and authentication. As a result, a device exploitation can be easily achieved with a carefully crafted packet. RuggedCom devices can be attacked via hard-coded credentials in the operating system [26]. The default account is present in the system to support password recovery, so can not even be disabled. However, attackers with the

knowledge of the MAC address can use this account to connect to the device and take full control of it.

[27] presented multiple attacks against satellite communication systems originating from the ground segment. In one of the attack scenarios, the man-machine interface of the airplane onboard SATCOM unit requires administrator password for restricted configurations and control mechanisms. The generation algorithm uses the device serial number (can be found printed on the device) plus a hard-coded string, which makes it easy to guess the password. Thus the attacker has access to all configurations and can disable critical parts related to the safety of the aircraft. [28] implemented a rogue carrier for satellite systems. Their method allows the attacker to become an illegitimate user of services provided. Firstly, the attacker must select its target, an artificial satellite. Then, the attacker point his antenna to the target and searches for unused, legal frequency for clients. If such a frequency is found, the attacker is free to transmit and receive as he wishes. However, the attacker still has to avoid detection: he has to sniff packets send by the operator to legitimate clients and do exactly as the operator packet asks. As stated in their talks, the method works because even if the satellite supports encryption, turning it on causes performance to drop significantly. As a result, operators turn it off because it is the service customers pay for, not the security of the service. [29] investigates the Automatic Dependent Surveillance-Broadcast (ADS-B) protocol and presents practical attacks using the vulnerabilities the protocol has: no authentication, no encryption and no challenge-response mechanisms. As a result, messages can be sniffed, spoofed or replayed. The attacker may confuse pilots and hinder them in performing their tasks.

[30] presents an attack against a smart home automation device, the Nest Thermostat. Pressing a button for 10 minutes on the device initiates a global reset. Afterwards, there is a small time window during which the device accepts code from USB sticks connected to it and uses that code for booting without any cryptographic checks on the code. An attacker can use this vulnerability to install an SSH server and access the home network of the user. However, physical access is needed to the device to launch the attack, so either the attacker has to break into the house or compromise the device during transport. [31] presents physical and remote attack surfaces in cars. For example, the authentication protocol between the Telematics Unit and the center relies on a challenge response mechanisms. However, the random number generator is seeded with the same constant each time it is initialized. As a result, an observed response packet can be replayed by the attacker to authenticate himself as the Telematics Call Center, getting full control over the car. A possible attack against a wireless home automation device is presented in [32] which is used for controlling electrical outlets. The implementation of the Home Network Administration Protocol contains a buffer overflow which can be used to execute arbitrary code on the device. Since the device controls the power outlet to any device physically connected to it, the attacker has the ability to damage the connected device. The D-Link DIR-815 Wireless-N Dual Band Router contains a command injection vulnerability which allows the attacker to get remote access to the device as demonstrated by [33]. The vulnerability lies with the packet parsing: strings inside backticks are considered commands and executed on the router.

[34] discusses a case study of malicious firmware updates to a HP-RFU (Remote Firmware Update) LaserJet printer. The vulnerability which enables this attack comes from the fact that the printer has to accept printing jobs in an unauthenticated way (as dictated by the standard) and that the firmware is updated by printing to the memory. Thus, an attacker can send a printing job to the device, instructing it to update its firmware with the malicious code provided. [35] discusses attacks against a fireworks control system. The protocol used by the system provides no encryption, nor authentication, which allows the attacker to sniff packets and thus learn the addresses of each device. Now, the attacker might wait for the operator to `arm` the system, the attacker can immediately send the `digital arm` and `fire` commands. The system will immediately fire its pyrotechnics loads and may cause physical harm to the operator. The attack can be automated as well, since arbitrary Python code can be uploaded to the devices. [36] demonstrates multiple attacks against an automated external defibrillator. For example, the firmware upgrade software package shipped with the device contains a buffer overflow vulnerability which may result in arbitrary code execution. Another vulnerability is the use of CRC as a digital signature. Combining these two vulnerabilities allows the attacker to harm patients by setting shock protocols and shock strengths or launch a cyberattack against the IT system in which the device is deployed.

V. An attack taxonomy for embedded systems

In this section, we identify the classification criteria as the basis for deriving an attack taxonomy. For this purpose, we analyzed reported vulnerabilities related to embedded systems that we obtained from the public CVE database.

Analyzing CVE data was a major challenge in our work. The CVE database has more than 60,000 records, in which only a small part is relevant to embedded devices. CVE records do not contain meta-information that would make it easy to identify which records are related to embedded systems. Therefore, we applied heuristics to identify and extract the relevant subset. Specifically, we implemented a script that matched CVE records to a whitelist and a blacklist of keywords that we defined, and selected those entries whose textual description contains at least one word from our whitelist and did not contain any word from our blacklist. Our script identified 3826 relevant CVE records, which was still infeasible to read and analyze manually. In addition, the set of selected CVE records was quite biased in the sense that a large subset of the records was related to devices produced by a small number of embedded device manufacturers (e.g., 3306 out of the 3826 records were related to CISCO products).

Next, we randomly sampled the 3826 CVE records based on the target mentioned at [37] with the limitation that only

35 entries may relate to same target type (hardware, operating system or application) mentioned in [37]. However, this resulted in the over-representation of some large companies (e.g. Cisco), so we had to set the limit of 9 for the number of CVE records from the same company. The result of the steps mentioned above was a sample set of 106 CVE records. We also added one CVE entry for which [37] did not provide information about the affected product. Finally, we manually analyzed all the 106 CVE records in the selected sample set, and identified appropriate criteria based on which the vulnerabilities in the CVE records can be classified.

Assuming that our selected sample set is representative, the identified classification criteria form the basis of a taxonomy of attacks against embedded devices. In order to check that indeed all 3826 CVE records can be categorized according to the criteria derived from the analysis based on the 106 samples, we later created a script that classifies all 3826 records according to our criteria in an automated way. Our script classified the majority of the records with no problem, and we were able to handle the few exceptions manually by refining our set of classification criteria. At the end of this process (cf. Sec. VI), we obtained a final set of criteria that allowed for the classification of all 3826 CVE records related to embedded device vulnerabilities.

A. Identify attack taxonomy classification criteria

Based on existing attack taxonomies (cf. Sec. II) and attacks (cf. Sec. IV), we defined 5 dimensions along which attacks against embedded systems can be classified: (1) precondition, (2) vulnerability, (3) target, (4) attack method, and (5) effect of the attack. The precondition dimension contains possible conditions that are needed to be satisfied in order for the attacker to be able to carry out the attack. The vulnerability dimension contains different types of vulnerabilities that can be exploited by the attacker. The target dimension contains possible attack targets by which we mean a specific layer of the system architecture or the embedded device as such if no specific layer can be identified as a target. The attack method dimension contains various exploitation techniques that the attacker can use. The effect dimension contains possible impacts of an attack.

We populated the above dimensions by going through the 106 selected CVE records and observing the preconditions and the type of the vulnerabilities described in the CVE records, as well as the targets, methods, and effects of the potential attacks that may exploit the described vulnerabilities. As for the preconditions, we observed the following types of requirements for the attacker:

- **Internet facing device:** Many vulnerabilities in the CVE records are potentially exploitable by a remote attacker if the device is connected to the Internet. The attacker dose 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:** This precondition requires the attacker to have some privileges that allow for logical access to the services or functions provided by the device. This logical access can be restricted to local access or it can be a remote access capability (e.g., via the Internet). Often, the privileges required by the access are normal user privileges, and not administrator privileges.
- **Direct physical access to the device:** Direct physical access requires the attacker to access the device physically. However, the attacker might not need any privileges to access the services of the device.
- **Physically proximity of the attacker:** In some cases, the attacker does not need physical access. It is sufficient that the attacker can be in the physical proximity of the device. For instance, attacks on wireless devices may only require to be within the radio range of the target device.
- **Miscellaneous:** We observed a number of other preconditions in CVE records, each appearing in only one or a few records, and we decided to create this general category to represent them. An example for a miscellaneous precondition is when the target device has to run some software or has to be configured in a certain way for the vulnerability to be exploitable.
- **Unknown:** Some CVE records and other sources do not provide sufficient amount of information to determine the preconditions of a potential attack; in these cases, we classify the precondition as unknown.

The vulnerabilities that we observed in the CVE records have the following types:

- **Programming errors:** Many of the vulnerabilities in the selected CVE records stem from programming errors, which may lead to control flow attacks (e.g., input parsing vulnerabilities leading to buffer overflow problems, and memory management problems such as using pointers referring to memory locations that have been freed).
- **Web based vulnerability:** Many embedded devices have a web based management interface through which they can be configured and updated. However, the web server applications running on those devices are typically rarely updated. Hence, those devices are exposed to web based attacks that exploit unpatched vulnerabilities in the web based interface of the device.
- **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. Such vulnerabilities make it possible for attackers to bypass access control mechanisms rather easily with minimal effort.
- **Improper use of cryptography:** Some devices use cryptographic mechanisms for authentication purposes or for preserving the confidentiality of some sensitive information. Often, cryptographic mechanisms are not used appropriately, which leads to fatal security failures. Examples include the use of weak random number generators for generating cryptographic keys, or vulnerabilities in the protocols that use cryptographic primitives.
- **Unknown:** Similar to precondition, some CVE records

do not contain information about the vulnerability itself, while they described the target and the effect of the potential attacks exploiting the unspecified vulnerability.

Regarding the target of the potential attacks, we distinguish the following layers of the embedded system architecture: **hardware**, **firmware/OS**, and **application**. When no specific layer can be determined from the CVE record, or when a potential attack can target multiple layers, we identify the **device** itself as the target of the attack. Note that we do not make a clear distinction between the firmware and the operating system (OS), because in many cases, embedded devices have no real OS, but their firmware provides typical OS functionality (such as control to resources). In addition, we observed that the attacks may not target the embedded devices themselves, but many CVE records report potential attacks on the **protocol** used by those devices for communication or device management. The attacks that we observed in the selected CVE records cover a wide range of methods. We grouped them in the following types:

- **Control hijacking attacks:** This type of attacks divert the normal control flow of the programs running on the embedded device, which typically results in executing code injected by the attacker.
- **Reverse engineering:** Often, an attacker can obtain sensitive information (e.g., an access credential) by analyzing the software (firmware or application) in an embedded device. This process is called reverse engineering. By using reverse engineering techniques, the attacker can find vulnerabilities in the code (e.g., input parsing errors) that may be exploited by other attack methods.
- **Malware:** An attacker can try to infect an embedded device with a malicious software (malware). There are different types of malware. A common characteristic is that they all have unwanted, potentially harmful functionality that they add to the infected system. A malware that infects an embedded device may modify the behavior of the device, which may have consequences beyond the cyber domain. For instance, the infamous Stuxnet worm reprogrammed PLCs in an uranium enrichment facility, which ultimately led to the physical destruction of the uranium centrifuges controlled by the infected PLCs.
- **Injecting crafted packets or input:** We observed that injection of crafted packets is an attack method against protocols used by embedded devices. A similar type of attack is the manipulation of the input to a program running on an embedded device. Both packet and input crafting attacks exploit parsing vulnerabilities in protocol implementations or other programs. In addition, replaying previously observed packets or packet fragments can be considered as a special form of packet crafting, which can be an effective method to cause protocol failures.
- **Eavesdropping:** While packet crafting is an active attack, eavesdropping (or sniffing) is a passive attack method whereby an attacker only observes the messages sent and received by an embedded device. Those messages may contain sensitive information that is weakly protected or not protected at all by cryptographic means. In addition, eavesdropped information can be used in packet crafting attacks (e.g., in replay type of attacks).
- **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, and dictionary attacks against password based authentication schemes. In both cases, brute-force attacks are feasible only if the search space is sufficiently small. Unfortunately, we observed CVE records that report such vulnerabilities.
- **Normal use:** This refers to the attack that exploit an unprotected device or protocol through normal usage. This is because we observed CVEs that reported on potential attacks where the attacker simply used some unprotected mechanism as if he was a legitimate user. For instance, the attacker can access files on an embedded device just like any other user if the device does not have any access control mechanism implemented on it.
- **Unknown:** We observed some CVEs that described vulnerabilities but did not identify any particular attack method that would exploit those identified vulnerabilities.

The effect of the above attacks can be the following:

- **Denial-of-Service:** Many CVE records identify potential attacks that lead to denial-of-service conditions such as malfunctioning or completely halting the device.
- **Code execution:** Another large part of the analyzed CVE records identify execution of attacker supplied code on the embedded device as the effect of potential attacks. This also includes web scripts and SQL injections, not only native code of the device.
- **Integrity violation:** A commonly obeservable effect of potential attacks is the integrity violation of some data or code on the device. This includes modification of files and configuration settings, as well as the illegitimate update of the firmware or some applications on the device.
- **Information leakage:** In some cases, the effect of the attack is the leakage of some information that should not be obtained by the attacker.
- **Illegitimate access:** Many attacks result in the attacker gaining illegitimate access to the device. This not only includes the cases when an attacker, who otherwise has no access to the device, manages to logically break into it, but also cases when the attacker has already some access, but he gains more privileges (i.e., privilege escalation).
- **Financial loss:** Certain attacks enable the attacker to cause financial loss to the victim e.g. by making calls from a smart phone. Actually, most attacks can lead to financial loss in a general sense, so we use this criterion to represent only those attacks whose primary goal is to cause financial loss. A typical example would be an attack which aims at sending and SMS or making a call to a premium number from a compromised smart phone.

- **Degraded level of protection:** In some CVE records, we observed that the potential attack results in a lower level of protection than expected. An example would be when a device is tricked into using weaker algorithms or security policies than those that it actually supports.
- **Miscellaneous:** Some attacks cause users to be redirected to malicious websites or traffic to be redirected. In these cases, there is not enough information about what happens exactly to the redirected user or traffic, but there is information about the effect.
- **Unknown:** In some CVE records, no specific attack effect is identified. This is mainly the case when the attack method is not identified either, and the CVE record contains only information about a vulnerability.

B. Examples

In the rest of this section we will show how the taxonomy can be applied to CVE entries through two examples. Firstly, the description of CVE-2006-2560 is as follows:

```
Sitecom WL-153 router firmware before 1.38
    allows remote attackers to bypass access
    restrictions and conduct unauthorized
    operations via a UPnP request with a
    modified InternalClient parameter, which
    is not validated, as demonstrated by using
     AddPortMapping to forward arbitrary
    traffic.
```

The description mentions a remote attacker without authentication, who has to be able to send packets to the **Internet facing** device. The description also states, that the exploitation is done via a modified parameter, which means that the attack method is **Injecting crafted packets or input**. The entry also tells us that the parameter is not validated by the **firmware**, so the vulnerability is a **programming error**. By exploiting this vulnerability, the attacker can conduct unauthorized operations, so the effect is **illegitimate access**.

An an other example, let us consider CVE-2008-1262:

```
The administration panel on the Airspan WiMax
    ProST 4.1 antenna with 6.5.38.0 software
    does not verify authentication credentials
    , which allows remote attackers to (1)
    upload malformed firmware or (2) bind the
    antenna to a different WiMAX base station
    via unspecified requests to forms under
    process_adv/.
```

The attacker remotely sends messages to the **Internet facing** device. He is able to upload a malformed **firmware**, which results in an **integrity violation**. The antenna does not verify authentication credentials and thus implements **weak access control or authentication**. As a result, the attacker does not need to do anything suspicious: he has to access the device, type in credentials when prompted and upload the modified firmware which is a completely **normal usage**.

VI. Evaluation of the taxonomy

To evaluate our taxonomy, we applied it to the 3826 CVE subset related to embedded systems. Applying the taxonomy was done in a half-automated and iterative way. We created a Python script[1] that used expressions of CVE entries related to a particular category in each dimension of our taxonomy. For example, obtaining some kind of access to the system is common in the Illegitimate Access category of the Effect dimension. When the script encountered an entry for which it could not determine the correct category, it displayed the description of the entry and the relevant expression had to be added manually. We repeated this procedure until all CVE entries were categorized.

The immediate result of our script shows that a lot of CVE entries can be put into multiple categories. And this outcome is natural if we look at some examples. For example, CVE-2010-0597 tells us that the vulnerability “allows remote authenticated users to read or modify the device configuration, and gain privileges or cause a denial of service (device reload)”. Reading the configuration discloses sensitive information to the attacker, writing the configuration violates the integrity of the configuration, privilege escalation is a type of illegitimate access and then there is denial of service, which is a single category in itself. It depends on the attacker what effect he wishes to have. And if his actions had multiple effects, his attack could be categorized into multiple categories in the Effect dimension. This observation stands for the attack method as well since there are attacks which require multiple steps to be performed by the attacker. The description of CVE-2009-1477 states that certain switches have hard-coded SSL private keys which allows the attacker to decrypt HTTPS sessions. Exploiting this vulnerability requires the attacker to obtain the hard-coded key from a previous installation, then he has to sniff the channel for messages to decrypt. Our taxonomy allows an attack to be classified into multiple categories, thus variations of attacks can be handled easily. However, to create statistics and for the sake of manageability, we needed to simplify the output of our script. We chose the following approach: when a CVE entry could be categorized into multiple categories, we have manually selected the scenario we think has the most possibility of happening.

The output of our script was a table in which each row represents a vector from our taxonomy, a possible attack scenario. Vectors have five coordinates, each coordinate representing a category from our five dimensions. Figure 1 shows vectors on a parallel coordinates diagram. Each path on the diagram is an attack scenario described by the vector, the categories the path touches in each dimension show a different aspect of the attack. The thicker a path is, the more CVE entries mention it as a possible scenario. It is clear from the figure that most attacks require only that the attacked device should have a public IP address. Local or remote access (some kind of user authentication) is also often needed to launch the attack.

The attacker has multiple methods available, but according to the CVE entries, most of the time he will either inject crafted inputs and arguments, or perform a control hijack

[1] The created scripts can be found at http://www.hit.bme.hu/~buttyan/publications.html#PappMB15pst

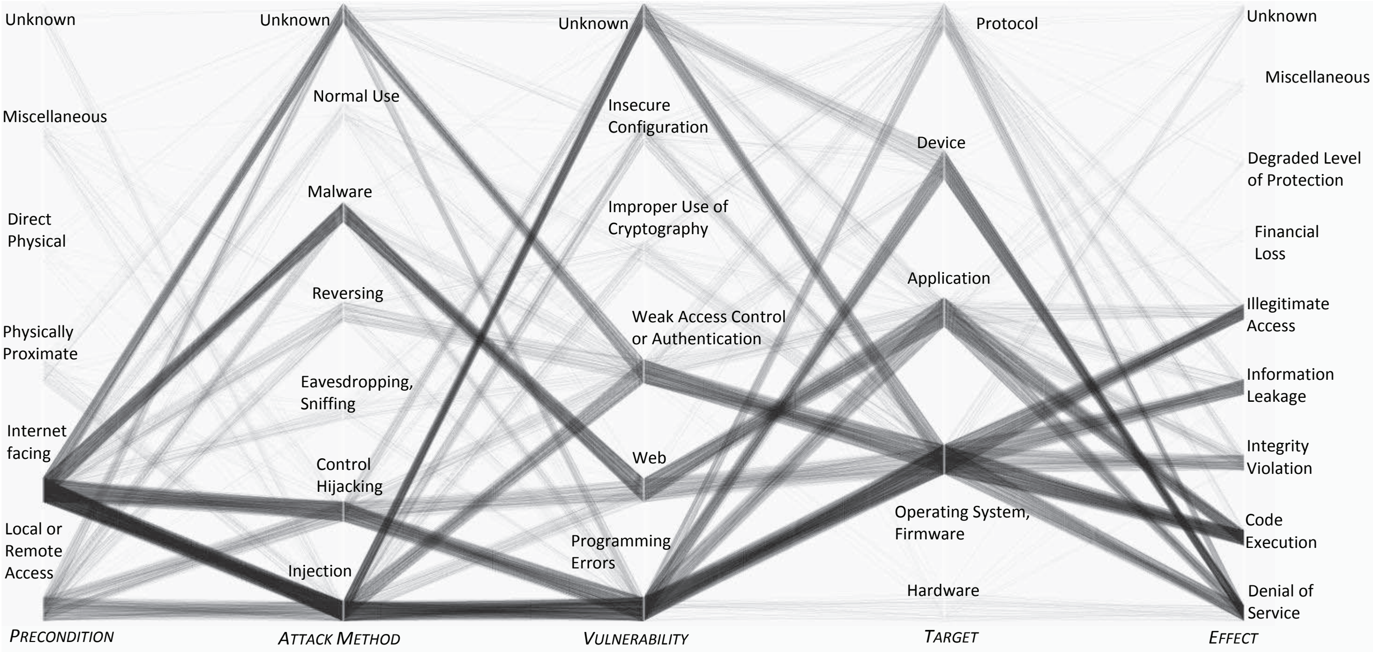

Fig. 1. Common Attack Scenarios

by exploiting buffer overflows or embedding commands into parameters. Another common exploitation method is the use of malware, when the attacker injects scripts into web pages or is able to install a compromised firmware. There are numerous CVE entries which do not state how the vulnerability could be exploited.

CVE entries suggest that embedded systems have three common vulnerabilities: programming errors, web-based vulnerabilities and weak access control or authentication. Figure 1 also shows common ways to exploit the vulnerabilities. A programming error can be exploited by control hijacking and crafted inputs. A web-based vulnerability is often exploited by a malicious script. Weak access control or authentication is also exploitable by crafting inputs e.g. directory traversal vulnerabilities. It must also be stated that a considerable amount of CVE entries do not disclose the vulnerability.

The possible target of the attack is almost always (although sometimes indirectly) mentioned in the entries. Operating systems and firmware suffer the most attacks: programming errors in these pieces of software and weak access control or weak authentication enable attacks at the lowest software-based level but the undisclosed vulnerabilities often affect these pieces of software as well. Applications could be targeted via exploiting programming errors and web-based vulnerabilities. Many black lines touch the Device category because when the entry did not provide any information about which part of the system if affected by the attack, we classified the target into the Device category. Another observation here concerns protocols: this category is closely related to programming errors which tells us that most of the time the implementation of protocols contain exploitable vulnerabilities and it is not the design of the protocol that is flawed.

Embedded systems can be affected by attacks in multiple ways. Denial of service situations are quite commonly mentioned, especially if there is no accurate information on the target of the attack. The attacker may also be able to execute code, either his own or a program installed on the system. Operating systems and firmware suffer the most integrity violation: the attacker might compromise sensitive files or install a new firmware. Information leakage affects operating systems and firmware as well in situations when the attacker cannot modify an arbitrary file but is able to read said file or when sensitive information e.g. version information is disclosed. Naturally, illegitimate access is closely related to operating systems and firmware through privilege escalation and impersonation.

VII. Conclusion

This paper provides a comprehensive overview of embedded systems security by describing both attacks and vulnerabilities. It enables us to create an attack taxonomy which we used to classify and describe common attack scenarios against embedded systems. The attack taxonomy derived in this paper provides information on how an embedded system can be attacked. Moreover, the structured knowledge can assist analysis and design of systems including or based on embedded devices during system development lifecycle.

The presented attack taxonomy also helps us to forecast trends in embedded-system security. Considering the attacks and vulnerabilities discussed in this paper and the recent trends in machine-to-machine communications, in our opinion, Internet facing devices will continue to suffer the majority of attacks. What is more, the vulnerabilities and errors identified in our taxonomy are similar to errors that arose in traditional

IT systems. However, traditional IT systems already have solutions and tools to address these issues. We anticipate that the solutions will be deployed in embedded systems with modifications tailored for the needs of this field.

This taxonomy is developed within a large-scale research project addressing embedded systems security for safety- and mission-critical systems. Our next step will be to further validate the taxonomy in realistic settings through different use cases led by industry. Moreover, the taxonomy and the knowledge will be applied to security analysis of cyber-physical systems to identify and enumerate threats in a systematic way with reduced error and uncertainty.

Acknowledgment

This work is partially supported by the EU ARTEMIS project EMC^2 (contract no. 621429) and Austrian Research Promotion Agency (FFG).

References

[1] F. Vahid and T. Givargis, "Embedded system design: A unified hardware/software approach," *Department of Computer Science and Engineering University of California*, 1999.

[2] E. A. Lee, "Computing foundations and practice for cyber-physical systems: A preliminary report," *University of California, Berkeley, Tech. Rep. UCB/EECS-2007-72*, 2007.

[3] P. Marwedel, *Embedded system design: Embedded systems foundations of cyber-physical systems*. Springer Science & Business Media, 2010.

[4] R. Langner, "Stuxnet: Dissecting a cyberwarfare weapon," *Security & Privacy, IEEE*, vol. 9, no. 3, pp. 49–51, 2011.

[5] B. Schneier, "Security risks of embedded systems," https://www.schneier.com/blog/archives/2014/01/security_risks_9.html, January 2014.

[6] S. Parameswaran and T. Wolf, "Embedded systems security - an overview," *Design Automation for Embedded Systems*, vol. 12, no. 3, pp. 173–183, 2008.

[7] D. Kleidermacher and M. Kleidermacher, *Embedded systems security: practical methods for safe and secure software and systems development*. Elsevier, 2012.

[8] D. N. Serpanos and A. G. Voyiatzis, "Security challenges in embedded systems," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 12, no. 1s, p. 66, 2013.

[9] P. Kocher, R. Lee, G. McGraw, A. Raghunathan, and S. Moderator-Ravi, "Security as a new dimension in embedded system design," in *Proceedings of the 41st annual Design Automation Conference*. ACM, 2004, pp. 753–760.

[10] P. Koopman, "Embedded system security," *Computer*, vol. 37, no. 7, pp. 95–97, 2004.

[11] S. Ravi, A. Raghunathan, P. Kocher, and S. Hattangady, "Security in embedded systems: Design challenges," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 3, no. 3, pp. 461–491, 2004.

[12] H. Holm, M. Ekstedt, and D. Andersson, "Empirical analysis of system-level vulnerability metrics through actual attacks," *Dependable and Secure Computing, IEEE Tran. on*, vol. 9, no. 6, pp. 825–837, 2012.

[13] L. Bilge and T. Dumitras, "Before we knew it: an empirical study of zero-day attacks in the real world," in *Proceedings of the 2012 ACM conference on Computer and communications security*. ACM, 2012, pp. 833–844.

[14] A. Costin, J. Zaddach, A. Francillon, and D. Balzarotti, "A large-scale analysis of the security of embedded firmwares," in *Proceedings of the 23rd USENIX Security Symposium*. San Diego, CA, USA: USENIX Association, 2014, pp. 95–110. [Online]. Available: https://www.usenix.org/system/files/conference/usenixsecurity14/sec14-paper-costin.pdf

[15] European Union Agency for Network and Information Security (ENISA), "Existing indicent taxonomies."

[16] C. Simmons, C. Ellis, S. Shiva, D. Dasgupta, and Q. Wu, "Avoidit: A cyber attack taxonomy," University of Memphis, Technical Report CS-09-003, 2009. [Online]. Available: http://si.lopesgazzani.com.br/docentes/marcio/SegApp/CyberAttackTaxonomy_IEEE_Mag.pdf

[17] T. L. Nielsen, J. Abildskov, P. M. Harper, I. Papaeconomou, and R. Gani, "The capec database," *Journal of Chemical & Engineering Data*, vol. 46, no. 5, pp. 1041–1044, 2001.

[18] S. Hansman and R. Hunt, "A taxonomy of network and computer attacks," *Computers and Security*, vol. 24, no. 1, pp. 31 – 43, 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167404804001804

[19] B. Zhu, A. Joseph, and S. Sastry, "A taxonomy of cyber attacks on scada systems," in *(iThings/CPSCom'11*, Washington, DC, USA, 2011.

[20] A. Dessiatnikoff, Y. Deswarte, E. Alata, and V. Nicomette, "Potential attacks on onboard aerospace systems," *IEEE Security and Privacy*, vol. 10, no. 4, pp. 71–74, Jul. 2012. [Online]. Available: http://dx.doi.org/10.1109/MSP.2012.104

[21] M. Yampolskiy, P. Horvath, X. D. Koutsoukos, Y. Xue, and J. Sztipanovits, "Taxonomy for description of cross-domain attacks on cps," in *Proceedings of the 2Nd ACM International Conference on High Confidence Networked Systems*, ser. HiCoNS '13. New York, NY, USA: ACM, 2013, pp. 135–142.

[22] MITRE Corporation, "Common Vulnerabilities and Exposures," https://cve.mitre.org/, 1999-2015.

[23] M. Keefe, "Timeline: Critical infrastructure attacks increase steadily in past decade," 2012. [Online]. Available: http://www.computerworld.com/article/2493205/security0/timeline--critical-infrastructure-attacks-increase-steadily-in-past-decade.html

[24] L. Apa and C. M. Penagos, "Compromising Industrial Facilities from 40 Miles Away ," ser. BlackHat, 2013.

[25] E. Forner and B. Meixell, *Out of Control: SCADA Device Exploitation*, Cimation, 2013. [Online]. Available: https://media.blackhat.com/us-13/US-13-Forner-Out-of-Control-Demonstrating-SCADA-WP.pdf

[26] J. W. Clarke, "RuggedCom - Backdoor Accounts in my SCADA network? You don't say..." http://seclists.org/fulldisclosure/2012/Apr/277, April 2012.

[27] R. Santamarta, *SATCOM Terminals: Hacking by Air, Sea, and Land*, IOActive, Inc., 2014. [Online]. Available: https://www.defcon.org/images/defcon-22/dc-22-presentations/Cerrudo/DEFCON-22-Cesar-Cerrudo-Hacking-Traffic-Control-Systems-UPDATED.pdf

[28] J. Geovedi and R. Irayndi, "Hacking a bird in the sky: Exploiting satellite trust relationship," 2008. [Online]. Available: http://www.sliim-projects.eu/docs/Hacking/D1T1-Jim_Geovedi-Hacking_a_Bird_in_the_Sky_2.0.pdf

[29] A. Costin and A. Francillon, "Ghost in the air(traffic): On insecurity of ads-b protocol and practical attacks on ads-b devices," 2012. [Online]. Available: http://s3.eurecom.fr/docs/bh12us_costin.pdf

[30] G. Hernandez, O. Arias, D. Buentello, and Y. Jin, "Smart nest thermostat: A smart syp in your home," ser. Black Hat, 2014.

[31] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, and T. Kohno, "Comprehensive experimental analyses of automotive attack surfaces," in *Proceedings of the 20th USENIX Conference on Security*, ser. SEC'11. Berkeley, CA, USA: USENIX Association, 2011, pp. 6–6.

[32] "Hacking the d-link dsp-w215 smart plug," http://www.devttys0.com/2014/05/hacking-the-d-link-dsp-w215-smart-plug/, /DEV/TTYS0, May 2014.

[33] Z. Cutlip, "Dlink dir-815 upnp command injection," http://shadow-file.blogspot.hu/2013/02/dlink-dir-815-upnp-command-injection.html, February 2012.

[34] A. Cui, M. Costello, and S. J. Stolfo, "When firmware modifications attack: A case study of embedded exploitation," in *Proceedings of NDSS Symposium 2013*, 2013.

[35] A. Costin and A. Francillon, "Short paper: A dangerous 'pyrotechnic composition': Fireworks, embedded wireless and insecurity-by-design," in *Proceedings of the 2014 ACM Conference on Security and Privacy in Wireless &#38; Mobile Networks*, ser. WiSec '14. New York, NY, USA: ACM, 2014, pp. 57–62.

[36] S. Hanna, R. Rolles, A. Molina-Markham, P. Poosankam, K. Fu, and D. Song, "Take two software updates and see me in the morning: The case for software security evaluations of medical devices," in *Proceedings of the 2Nd USENIX Conference on Health Security and Privacy*, ser. HealthSec'11. Berkeley, CA, USA: USENIX Association, 2011, pp. 6–6.

[37] "CVE Details." [Online]. Available: http://www.cvedetails.com/