Shaila I Kolhar. International Journal of Engineering Research and Applications
www.ijera.com
ISSN: 2248-9622, Vol. 6, Issue I1, November 2016, pp. 79-86

RESEARCH ARTICLE OPEN ACCESS

Embedded System Security System: Threats, Vulnerabilities and Attack Taxonomy

Shaila I Kolhar
Selection grade Lecturer
shailavarun@gmail.com

Date of Submission: 02-11-2016 Date of Acceptance: 16-11-2016

ABSTRACT:

In the post-PC age, embedded systems are the engine that powers innovation across numerous industries, including transportation, healthcare, and process control. Security is becoming more important for the reliability of any smart or intelligent system based upon embedded systems as they grow more widespread and "invisible" in everyday life. In this study, we use publicly accessible data to undertake a comprehensive evaluation of the many risks and vulnerabilities facing embedded systems today. Furthermore, we use this data to develop an attack taxonomy specifically for embedded systems. The results of this article should provide light on the nature of the risks that embedded systems face, according to the authors. This information may aid in the study and identification of potential security issues during the design phase of a system.

Keywords: Embedded System, CVE, malware, vulnerability, unknow attack

I. INTRODUCTION

An embedded system is a computer that is part of a larger system and serves a specific purpose. Hardware, software, and even mechanical components may all be a part of it. This means that the word encompasses all computers that are neither personal computer or large mainframes [1]. They often take the form of Cyber-Physical Systems (CPS) owing to their hybrid nature [2]. The current industrial trend demonstrates their central role in several application domains that construct smart or intelligent systems [3]. These domains include automotive electronics, avionics, consumer electronics, trains, telecommunications, healthcare, and more.

Embedded systems play crucial roles in many missions and safety-critical systems, making security a paramount concern. It has been shown that physical harm may result from attacks on cyber systems [4]. Poor security design and implementation, as well as the challenge of ongoing patching [5], mean that embedded system security is no better than that of traditional IT systems. This is especially true for devices with remote control, maintenance, and operation capabilities, even though many methods have been proposed in the past to secure embedded systems [6], [7]. Factors such as deployment scale, resource limitations, the difficulty of physical protection, and cost consideration make it very challenging to secure embedded systems [8].

Security engineering of embedded systems requires an in-depth familiarity with the capabilities of a potential attacker. To determine security needs, generate novel solutions, and effectively implement security controls within the bounds of available resources, it is necessary to conduct a thorough study of the threat environment before beginning secure design and development. Therefore, the following concerns emerge within the context of embedded system security.

- What factors primarily contribute to the success of the attacks?
- What are the main vulnerabilities?
- What are the commonalities of the attacks?

In this article, we analyze the current security landscape in detail and identify potential weak spots. We zero in on two groups of information: first, the vulnerabilities particular to embedded systems that have been reported, and second, the exposures of attacks on embedded systems in security conferences and literature. From

this information, we construct a taxonomy of attacks to precisely name and categorize prevalent threats to embedded systems. We hope that by understanding the full scope of assaults and their consequences, designers of mission- and safety-critical systems will be better equipped to make informed choices.

II. RELATED WORK

The distinctiveness of embedded system security and potential defenses against software and hardware threats are discussed in [9]. Different kinds of attacks are also covered. There have been several studies on the topic of computer and IT system vulnerabilities [13]. In [14], you'll find the results of an empirical investigation on the susceptibility of embedded systems.

ENISA [15] keeps track of existing incident categories for attacks on broad computer and IT systems. Among them is Sandia National Laboratories' common language security incident taxonomy, which classifies an event according to its attackers, tools, vulnerabilities, actions, targets, authorized outcomes, and objectives. Using the acronym AVOIDIT (for "attack vector," "operational impact," "defense," "information impact," and "target"), [14] provides a taxonomy for cyber assaults. IT security assaults are described in detail in Common Attack Pattern Enumeration and Classification (CAPEC) [15]. It classifies various forms of cyberattack into eleven distinct categories, such as "data leakage," "resource exhaustion," "injection," and so on. The four dimensions of attack taxonomy outlined by Hensman and Hunt [13] are attack vector, target, vulnerabilities and exploits, and the potential for a payload or impact beyond itself. There are several hierarchical layers of description for the data in each dimension.

Zhu et al. [11] presented a cyber-attack taxonomy for SCADA systems, which is a key component of cyber-physical systems. Hardware, software, and the protocol stack are the three main areas of a network that are often attacked during a cyberattack. Buffer overflow, SQL injection, and the exploitation of unprivileged embedded operating systems are the most common forms of software attack. Attacks on the communication stack may be broken down into the following categories: the network layer, the transport layer, the application layer, and the protocol layer. Desatnik et al. [2] identified probable assaults on avionics embedded systems and categorized them based on their impact on on-board aeronautical systems. There are two main types of attacks: those that target essential features and those that aim to disable error-handling safeguards. The authors illustrate the effect of each kind of assault using examples. A taxonomy for classifying cross-domain assaults was presented by Yampolskiy et al. [1], who focused on the potential effects of cyber-attacks on the real-world assaults, too. The taxonomy they propose is based on six factors targets, effects, and assaults are the three main categories.

The current taxonomies are inadequate since they do not Embedded devices. For instance, [9] is made for use in SCADA systems. aerospace on-board systems, like [2] does. While The purpose of [1]'s taxonomy is to record inter-domain impacts of cyber assaults, but rather a broad categorization of abstract attack semantics Not only that, but the disorganized taxonomy makes it hard to handle attack data effectively. In In our method, we build upon previous classifications of cyberattacks to shift the focus and format of them.

III. PROPOSED WORK

We began our investigation by compiling a comprehensive inventory of potential dangers. We go through the proceedings of computer security conferences like DefCon and BlackHat, which concentrate on computer hacking, and follow the links and white papers linked therein to compile extensive descriptions of attacks on embedded devices. We also scour the web for articles, blogs, and mailing groups that discuss hacking attempts on embedded devices. In addition, we include scientific studies with real-world applications. In Section IV, we detail the findings of this study.

The disclosed attacks/hacks only represent a subset of the whole threat environment due to researchers' specialised interests, the expense of security testing, and the non-disclosure agreement applied by vendors or asset owners. As the other side of the same coin, we also look at data on vulnerabilities in embedded systems to round up our understanding. The Common Vulnerabilities and Exposures (CVE) [2] database is our primary resource for this information. The Common Vulnerability Enumeration (CVE) is the largest database of security flaws. To facilitate the exchange of vulnerability data across institutions, the CVE database assigns a unique number to each entry. When we last checked, the database included over

60,000 items, albeit not all of them pertained to embedded systems. We make use of several improvised methods to filter and retrieve pertinent elements from the larger vulnerability data, and then manually analyze the extracted data. The investigation led to the development of a set of criteria for categorizing attacks, which forms the foundation of our taxonomy of security threats. Analysis and procedure are outlined in Section V. We use our suggested taxonomy to automatically categorize all CVE entries linked to embedded systems from 2005 to verify its usefulness. The findings presented in Section VI provide more evidence that the crafted taxonomy is sound.

IV. ATTACK ON EMBEDDED SYSTEMS

This section details many assaults that have been launched against embedded devices and systems and analyses the capabilities and consequences of these attacks. We don't think the examples are exhaustive, but we do think they're illustrative of a wide variety of possible uses, including industrial systems, communications, and consumer gadgets.

In [3], a schedule for essential facilities is provided. The first notable assaults occurred in 1982, and the number of attacks has steadily grown since 2001. Key management in wireless devices is discussed in [4], along with its flaws and potential for abuse. One of the gadgets, for instance, may be configured using the on-box wizard and a set of predetermined values shown in graphical form. A passphrase is created by the interface implementation and used to create the AES key. The srand() method uses the current time to seed the Pseudorandom Number Generator, while the rand() function acts as the generator itself. This allows the attacker to deduce the password and encryption key, allowing them to eavesdrop on all traffic on the victim's wireless network. The ModBus protocol is exploited in a remote attack on SCADA equipment, as seen in [5]. The protocol is flawed since it does not include security measures like encryption or authentication. This means that given the right packet, it's rather simple to abuse a device.

numerous ground-based assaults against satellite communication systems were shown. In one potential attack scenario, an administrator password is needed to access certain settings and controls in the airplane's onboard SATCOM unit's user interface. The password is easily guessed since the generation technique employs the device's serial number (which is written on the device) in addition to a hard-coded text. As a result, the attacker may change any setting and turn off any portion of the plane that has anything to do with its safety. For satellite networks, [8] introduced a rogue carrier. Using this technique, an attacker may trick a service into thinking they are a real user and steal their data. The first step for an attacker is to zero in on a certain satellite in the sky. The attacker then directs his antenna towards the intended victim and looks for free, legal channels to broadcast on.

If an attacker discovers such a frequency, he may use it to send and receive data at will. The attacker still has a challenge, though: he must remain undetected while sniffing operator packets destined for genuine customers and acting in accordance with their requests. They were successful in their plan because, as they had discussed, turning on encryption dramatically reduces performance, even if the satellite is capable of handling it. Therefore, operators disable it since users are paying for the service itself, not the security of the service. In [9], the flaws in the Automatic Dependent Surveillance-Broadcast (ADS-B) protocol, such as the lack of authentication, encryption, and challenge-response protocols, are examined and presented in the form of attack scenarios. This makes it possible to intercept, spoof, or replay communications. The attacker risks disorienting pilots and making it harder for them to do their jobs.

Describes an assault against the Nest Thermostat, a smart home automation device. The gadget may be reset entirely by holding down a button for 10 minutes. After that, the device has a brief window where it will take programs from USB sticks and boot from them without doing any cryptographic checks. An attacker may exploit this flaw by setting up an SSH server on the user's machine and gaining access to the user's private network. However, to start the assault, the attacker must either get into the residence or compromise the device while it is in transit. Vehicles are vulnerable to both direct and indirect attacks, as seen in [1]. For instance, the Telematics Unit's authentication protocol with the control center uses a challenge response method. The random number generator, however, is always initialized with the same constant. This means an attacker who has seen a response packet may impersonate the Telematics Call Centre and take complete control of the vehicle

by just replaying the packet. [2] describes a hypothetical attack against a wireless home automation system that may be exploited to switch on and off electrical outlets. There is a buffer overflow in the implementation of the Home Network Administration Protocol, which may be exploited to run arbitrary code. The attacker may harm the connected gadget by cutting power to it via the device's control of the outlet. As shown in [3], an attacker may get remote access to a D-Link DIR-815 Wireless-N Dual Band Router by exploiting a command injection vulnerability. The router's processing of packets is flawed since strings enclosed in backticks are interpreted as instructions and carried out.

describes a scenario in which an HP-RFU (Remote Firmware Update) LaserJet printer was subjected to rogue firmware upgrades. This attack is possible because printers are required by standard to accept printing jobs without authentication and the firmware is updated by writing to the memory. This allows an adversary to command the device to install malicious firmware by sending a print job to it. The vulnerabilities of a fireworks control system are discussed in [5]. The system's protocol lacks both encryption and authentication, making it easy for an attacker to discover the IP addresses of connected devices by packet sniffing. While the operator arms the system, the attacker may now submit digital arm and fire orders instantly. The pyrotechnic payload will be discharged instantly, potentially harming the operator. Since any Python code can be uploaded to the devices, the attack may also be automated. Multiple assaults against an AED are shown in [6]. The firmware update software that is pre-installed on the device, for instance, has a buffer overflow flaw that might allow for the execution of arbitrary code. The use of CRC as a digital signature also presents a security risk. By exploiting these flaws simultaneously, an attacker might cause injury to patients by altering shock protocols and shock intensities, or they could launch a cyberattack on the IT system the device is connected to.

V. AN ATTACK TAXONOMY FOR EMBEDDED SYSTEMS

Here, we detail the parameters used to categorize attacks and thereby derive a taxonomy. To do this, we used information on embedded-system vulnerabilities found in the open CVE database. The analysis of CVE data posed a significant obstacle to our progress. Only a fraction of the more than 60,000 items in the CVE database are applicable to embedded devices. Meta-information that would make it clear which CVE entries pertain to embedded systems is missing from the database. As a result, we used heuristics to zero in on and extract the useful subset. To be more specific, we built a script to compare CVE records against keywords we provided, both whitelisted and blacklisted, and then prioritized entries whose textual descriptions included at least one whitelisted term but none of the blacklisted ones. Even if our software found 3,826 relevant CVE records, it would be impossible to examine and analyze them all by hand. Furthermore, there was a substantial amount of bias in the set of chosen CVE data, with a disproportionate number of records pertaining to products made by a limited number of embedded device manufacturers (e.g., 3306 of the 3826 entries were associated with CISCO products).

5.1: Identify attack taxonomy classification criteria

We established 5 dimensions along which attacks against embedded systems may be categorized based on current attack taxonomies (cf. Section II) and assaults (cf. Section IV): There are five stages to an attack: (1) setup, (2) exposure, (3) selection, (4) execution, and (5) fallout. Possible criteria that must be met by the attacker before the assault may be carried out are catalogued in the precondition dimension. The vulnerability axis categorizes the many flaws that an adversary can use to gain an advantage. When we talk about potential attack targets, we're referring to either a particular layer of the system architecture or, if no such layer can be found, the embedded device itself. Multiple exploitation strategies may be found in the assault method dimension. Possible results of an assault are listed in the affect dimension.

By combing through 106 hand-picked CVE entries, we were able to fill the dimensions with information on the possible victims, vectors, and outcomes of attacks that may exploit the vulnerabilities detailed within. We identified the following prerequisites that an attacker must meet:

a) Internet Facing Devices: If the device is online, a remote attacker might theoretically exploit any number of the vulnerabilities listed in the CVE records. An attacker simply needs the ability to find the device and send it messages over the network; privileged access is not required.

b) Local or remote access to the device: This prerequisite necessitates the attacker's possession of some kind of privilege that grants them logical access to the device's services or functionalities. This logical access might be limited to users in the same physical location, or it can be a remote access capability (through the Internet, for example). In many cases, just standard user permissions, and not administrator permissions, are necessary for the access in question.

c) Direct Physical Access to the device: To get direct physical access, an attacker must physically reach the target device. However, the attacker may not even need any special permissions to use the device's features.

d) Physically Proximity of the attacker: In other instances, an attacker's access to a target is not even necessary. To compromise a gadget, an attacker needs just be physically close to it. In the case of wireless devices, an attacker may need just to be in radio contact with their intended victim.

e) *Unknown*: We label the prerequisite as unknown when the CVE record or other source does not give enough information to assess the preconditions of a possible attack.

In terms of where attacks are most likely to occur, we categorise embedded systems into three distinct layers: hardware, firmware/OS, and application. Whenever the CVE record does not specify which layer is at risk, or when an attack might be launched against more than one layer, we consider the device itself to be at risk. We do not draw a strict line between firmware and operating system (OS) here since many embedded devices do not have a true OS but instead rely on firmware to perform OS-like tasks (such as resource management). We also found that assaults may not be aimed directly against the embedded devices themselves, but rather at the protocol that is utilised by such devices for communication and administration. We found a broad variety of attack techniques documented in the chosen CVE reports. We classified them as follows:

a) Control Hijacking Attack: These assaults often cause the embedded device to execute malicious code by rerouting the control flow of the programs operating on the device.

b) Reverse Engineering: The software (firmware or application) in an embedded device may often be analysed by an attacker to get sensitive information (such as an access credential). The term "reverse engineering" describes this method. An attacker may use reverse engineering to discover flaws in the code (such as incorrect input processing) that can then be exploited using other forms of attack.

c) Malware: Malicious software (malware) might be installed on an embedded device by an attacker. Malware may be broken down into subcategories. All these infections have one thing in common: they introduce dangerous features to the infected system that weren't there before. Malware that infects an embedded device might potentially alter the device's behavior, which could have far-reaching effects. For instance, the notorious Stuxnet virus reprogrammed PLCs at a uranium enrichment plant, causing the destruction of the centrifuges they controlled.

d) Injecting crafted Packets or inputs: We found that one way to attack protocols used by embedded devices is to introduce specially designed packets into the network. Manipulating the input to a programme on an embedded device is another kind of attack of a similar nature. Parsing flaws in protocol implementations or other programs are what packet and input crafting attacks take advantage of. Reusing packets or packet pieces that have already been detected is also a sort of packet crafting that may be used to effectively trigger protocol failures.

e) Brute-force search attack: Brute-force search attacks are effective against insecure encryption and authentication schemes. These include dictionary attacks on password-based authentication systems and exhaustive key search attacks on cryptographic algorithms like cyphers and MAC functions. In both circumstances, the search space must be sufficiently constrained for a brute-force assault to be practical. Regrettably, we found CVE records that mention such vulnerabilities.

VI. EVALUATION OF THE TAXONOMY

We used the 3826 CVEs that are specifically relevant to embedded systems to test our classification. A semi-automated and iterative process was used to apply taxonomy. For each dimension of our taxonomy, we developed a Python script1 that evaluated expressions of CVE entries matching that dimension. In the Illegitimate Access subcategory of the Effect dimension, for instance, unauthorized entry into the system is a regular occurrence. If the script ran upon an item for which

it could not automatically establish a category, it would show the entry's description and the user would need to enter the appropriate phrase. This process was continued until all CVE entries were labelled.

Our script's first output reveals that many CVEs may be assigned to many different types. And if we look at several cases, we see that this result is inevitable. An example of such a flaw is CVE-2010-0597, which "allows remote authenticated users to read or modify the device configuration and gain privileges or cause a denial of service (device reload)". Denial of service is its own category, whereas reading the configuration may reveal sensitive information to an attacker, writing the configuration can compromise its integrity, gaining elevated access privileges is an example of unauthorized access, and so on. The intended result is up to the attacker. If his actions had a wide range of consequences, his assault may be broken down even further along the Effect axis.

Since there are attacks that need the execution of several sequential stages, this remark is also applicable to the attack methodology. According to the CVE-2009-1477 description, an attacker may decrypt HTTPS sessions on vulnerable switches since the switches' SSL private keys are stored in plaintext. The attacker must first get the hard-coded key from a prior installation, and then sniff the channel for messages to decode in order to exploit this issue. Our taxonomy provides for the classification of an attack into numerous categories, making it easy to manage variants of assaults. However, we wanted to streamline the output of our script to generate statistics and make it more manageable. When a CVE entry could be assigned to numerous categories, we picked the one that we believe to have the greatest likelihood of occurring.

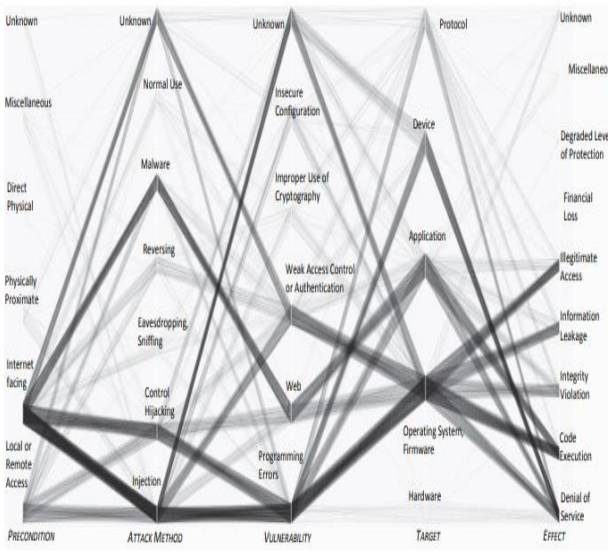

Figure No. 1: Common Attack Scenarios

Our software generated a table where each row represents a taxonomic vector, or potential assault scenario. Each of the five positions in a vector stands in for one of the five dimensions we use. Vectors are shown in parallel coordinates in Figure 1. Each line represents a possible assault scenario as specified by the vector; the dimensions in which the line passes provide further information about the attack. More CVE entries list a route's potential outcomes as a vulnerability the thicker the path. The diagram makes it apparent that a public IP address is all that's needed for most assaults. In many cases, the assault must additionally have either local or remote access (through user authentication of some kind).

The CVE entries suggest that although the attacker might use a variety of techniques, he most often injects specially crafted inputs and arguments or performs a control takeover by taking advantage of buffer overflows or embedding instructions into parameters. Malware injection into websites or malicious firmware installations are two more prevalent forms of exploitation. Many of the CVE listings don't explain how the flaw may be used in an attack.

CVE entries imply that programming mistakes, web-based vulnerabilities, and insufficient access control or authentication are the three most prevalent types of vulnerabilities in embedded systems. Common exploits are shown in Figure 1 as well. Control may be taken over and inputs can be manipulated to take advantage of a code flaw. A

malicious script will often target a web-based vulnerability. Inadequate authentication or access control, such as directory traversal flaws, may potentially be exploited with carefully crafted inputs. Furthermore, it should be noted that a substantial portion of CVE listings conceal the vulnerability.

The entries nearly invariably (though sometimes indirectly) describe the likely target of the assault. The most common targets of assaults are the system's operating system or firmware due to programming faults or insufficient access control or authentication. However, these systems are also often affected by unknown vulnerabilities. Programming flaws and online security holes might be used to compromise applications. Due to the lack of specificity in the entry about the component of the system that was compromised, several dark lines extend into the Device category. Another thing to note is that when it comes to protocols, the implementation of such protocols often contains exploitable flaws, rather than the design of said protocols themselves.

VI. CONCLUSION

This article describes both threats and weaknesses in embedded system security, providing a thorough review of the topic. As a result, we were able to define and explain typical attack scenarios against embedded devices inside an attack taxonomy. In this research, we develop a taxonomy of possible attacks against embedded systems. When applied to the system development lifecycle, the organized knowledge may aid in the analysis and design of systems that include or are based on embedded devices.

We can better foresee future developments in embedded-system security thanks to the attack taxonomy we offer here. We believe that Internet-facing devices will continue to bear the brunt of assaults due to the prevalence of the attacks and vulnerabilities outlined in this research and the current trends in M2M communications. In addition, our taxonomy has found vulnerabilities and mistakes that are quite like those that occurred in older forms of information technology. However, these problems can already be solved and techniques to combat them may be found in more conventional IT systems. We expect the solutions to be implemented in embedded systems with changes to meet the requirements of this sector. This taxonomy is created as part of a comprehensive study focusing on the safety and reliability of embedded systems used in critical infrastructure. The next phase of our work will consist of validating the taxonomy in real-world contexts through a variety of industry-driven use cases. In addition, this taxonomy and knowledge will be used to security analysis of cyber-physical systems, allowing for the systematic identification and enumeration of risks with decreased error and uncertainty.

REFERENCE

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

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

[7]. 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.

[8]. 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.

[9]. 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.

[10]. 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.

[11]. 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

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

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

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

[15]. 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.