IoT/Embedded vs. Security: Learn from the Past, Apply to the Present, Prepare for the Future

(Invited Keynote Paper)

Andrei Costin
Faculty of Information Technology
University of Jyväskylä
Jyväskylä, Finland
ancostin@jyu.fi

***Abstract*—It is expected there will be 50 billion IoT/embedded connected devices by 2020. At the same time, multiple recent studies revealed that IoT/embedded devices and their software/firmware is plagued with weaknesses and vulnerabilities. Moreover, various recent and prominent attacks, such as the *Mirai* botnet targeting Commercial Off-The-Shelf (COTS) IoT/embedded devices, and the *ROCA* attack targeting secure embedded hardware chips (in their many form-factors), clearly demonstrate the need to secure the many layers and components of the highly fragmented and heterogeneous ecosystem of IoT/embedded devices. In this paper we aim to explore, discuss and exemplify some research aspects and directions that could be used to help improve the long-term security posture for IoT/embedded devices.**

I. Introduction

It is expected there will be 50 billion IoT/embedded connected devices by 2020 [12]. At the same time, multiple recent studies revealed that IoT/embedded devices and their software/firmware is plagued with weaknesses [16] and vulnerabilities [13], [15]. Moreover, various recent and prominent attacks, such as the *Mirai* botnet targeting Commercial Off-The-Shelf (COTS) IoT/embedded devices [5], and the *ROCA* attack targeting secure embedded hardware chips (in their many form-factors) [25], clearly demonstrate the need to secure the many layers and components of the highly fragmented and heterogeneous ecosystem of IoT/embedded devices. In this paper we aim to explore, discuss and exemplify some research aspects and directions that could be used to help improve the long-term security posture for IoT/embedded devices.

II. Insights and Viewpoints

A. (In)security Research: Use It or Lose It?

1) The Past: The research related to (in)security and exploitation of IoT/embedded devices has a (un)surprisingly long history. Attacks on cryptographic implementations inside security embedded chips goes back as far as 2001 [9]. The prospects of practical feasibility to attack and exploit networked embedded devices dates back to at least 2002 [19]. The idea of large botnets based on embedded "dumb" devices was explored by researchers back in 2007 [7], [6], until the first malware targeting embedded devices (e.g., ADSL modems) that constructed a full-fledged botnet was discovered in 2009 and dubbed *Psyb0t* [8]. The extent and implications of default password usage in embedded devices have been studied at large scale back in 2010 [16].

2) The Present: However, it took the community and the vendors a decade (or more!) to just start reacting to the numerous wake-up calls related to those attacks and vulnerabilities. At present, there are many instances of large scale or impacting vulnerabilities and attacks that could have been checked, fixed or prevented most likely years ago. One example is the *Mirai* botnet that abused default credentials and created a DDoS botnet at large scale [5]. Another example is the *ROCA* attack that allows practical factorization to recover private RSA keys – it abuses a vulnerability in generation of RSA keys that are used by a software library implemented in various cryptographic smartcards, security tokens and other secure hardware chips from Infineon [25]. Unfortunately, all the mentioned research and works from the past, while being widely cited and well-known, were apparently either not taken seriously nor really put in practice (e.g., by vendors, by large organizations) in order to prepare a timely and effective security defense for IoT/embedded devices.

3) The Future and Key Takeaways: Very likely, many of the problems, vulnerabilities, attacks and exploits that endanger the security and privacy of IoT/embedded devices in the short and medium term future, may have been already discussed, explored and researched. However, finding the relevant and applicable research papers to follow and act upon in the currently immense amount of published work is similar to *finding the needle in the haystack*. Additionally, the discovery of relevant research could be made potentially more challenging by what is called *the Sturgeon's law*, also known in its modern critical thinking form as *"90% of everything is crap"* (which anecdotally is backed-up by the 10%-20% acceptance rates of most top peer-reviewed academic venues). Therefore, the key takeaways could be summarized as:

> *The attacks and defenses for IoT/embedded devices for the upcoming short and medium term future most likely have been already discussed and explored.*
>
> *Following the above patterns and numbers, the key at this point in time is to know how to find those 10% of relevant research, likely and mainly published during (but not only) 2007–2014.*
>
> *Even decades after a good original research is conducted, there is still a lot of potential and place to create even better research with global impact.*

ISSN 2305-7254

B. Security Knowledge (CVEs) vs. ML/AI

1) The Past: It may have been a normal practice in the past to not have CVEs for all the vulnerabilities, in particular back when the embedded devices were seen more as an outlier rather than a serious and dangerous threat. As an example of long-time missing CVE information, let us consider the case of *Hydra-2008.1* (also known as *Hydra D-Link*, and not to be confused with *THC-Hydra* [4]). It is perhaps the first malware known (since 2008) to attack embedded devices, such as D-Link routers, and whose purpose was to build an IRC-based botnet. To the best of our knowledge, the vulnerability it exploited did not (and still does not!) have a CVE number, while the vulnerability is most commonly referred to in various technical reports as: *"a D-Link authentication bypass exploit"* [21].

2) The Present: At present, given the wealth of information and tools, there should be no place for excuse to not *properly and timely* create, document and track particular information pieces of vulnerabilities, exploits, and malwares. This can be done even post-factum, e.g., *CVE-1999-1122* was assigned 10 years (!) after the original disclosure. However, as of this writing, even a decade after *Hydra-2008.1* was released, the same (or extremely similar) vulnerabilities in D-Link devices are being rediscovered over and over again – in 2013 [30], and in 2017 [32], [18]. What is worst, to the best of our knowledge, none of those (re)discoveries yet have a properly traceable entry in the CVE, though some of them have Exploit-DB (EDB) entries [18]. As another illustrative example, let us consider the case of Snort rule *ET 2020857* known as *"ET EXPLOIT Belkin Wireless G Router DNS Change POST Request"*. First, it contains a wrong Exploit-DB *reference:url*, i.e., the correct EDB-ID should be *6305* instead of *3605*. Second, it is missing the *reference:cve* which should be *reference:cve,CVE-2008-1244*. Third, the initial rule was created 7 years (!) after the original CVE and vulnerability was disclosed. Finally, despite being recently updated in 2017, the wrong *reference:url* was not corrected, and the missing *reference:cve* was not added.

Nowadays, the ML/AI techniques and knowledge reached a particular maturity level, and are being actively applied in various domains. Cybersecurity is no exception to that, and many propose ML/AI as a solution (or even "panacea") for the ever increasing amount and complexity of cybersecurity attacks and exploits [17], [10], [14]. However, according to Hand et al. [23], the quality of data is *critical* to machine learning: *"The effectiveness of a data mining exercise depends critically on the quality of the data. In computing this idea is expressed in the familiar acronym GIGO – Garbage In, Garbage Out"*. Moreover, missing data is altogether a blocking barrier for any AI/ML system. As we discuss in more detail later in Section II-C, in certain cases and attacks, a large proportion of exploited (or just disclosed) vulnerabilities for IoT/embedded devices are not documented within CVE or similar databases. In the end, how an AI/ML instance is supposed to learn and apply the knowledge (e.g., alert, defend, prevent), if there is nothing to learn from in the first place?

3) The Future and Key Takeaways: Hence, we can outline in this context at least the following challenges. First, it is to ensure that the data produced by and used in cybersecurity-related advisories and reports are maximally accurate, updated, informative, and machine-readable. Also, it must be ensured that the security and vulnerability knowledge "survives" and evolves no matter the circumstances, e.g., shutdown of *milw0rm* [2] vs. shutdown of *OSVDB* [3]. Second, it is to ensure that cybersecurity organizations employing ML/AI on the above mentioned data take a practical, effective and efficient approach to data quality improvement [20]. However, it is a totally different challenge to find out how many and which organizations actually implement such data quality improvement processes. In addition, it can prove uneasy to verify AI/ML data quality claims of an organization in a easy, trustworthy and secure manner, without compromising the "intellectual property" related to the organizations' data and its competitive edges. Therefore, the key takeaways could be summarized as:

> *To handle effectively, efficiently and securely vulnerabilities in nearly 50 billion devices, vulnerability details must be properly and timely documented, updated and tracked in CVE and similar databases.*
>
> *As a community of cybersecurity researchers and professionals, we should continuously help improve the knowledge about vulnerabilities (e.g., CVE, exploits), in particular for IoT/embedded and other emerging fields. For example, #CVECleanupChallenge could be one of the many possible solutions.*
>
> *It would be virtually impossible to build reliable and robust ML/AI cybersecurity solutions based on missing, incorrect or incomplete security and vulnerability data (still found in CVE and similar databases).*

C. "Low-Hanging Fruit" Vulnerabilities

It has been demonstrated over and over again that, at a large scale, the IoT/embedded devices are literally plagued with vulnerabilities [13], [15]. Yet, little is being done to completely eradicate the "evil root cause" of some of those vulnerability classes. Let us consider for example Cross-Site Request Forgery (CSRF, also known as XSRF) vulnerabilities. The basic idea behind CSRF is simple: on behalf of a victim user (e.g., bank client, router admin), an attacker is able to perform actions of her choosing on a vulnerable target application, using an URL link or other HTTP-related content.

1) The Past: The CSRF vulnerabilities and their attack potential has been known in detail since 2005 [11]. In fact, the first CVE related to CSRF is documented as *CVE-2002-1648* and it dates back to 31st December 2002. The first prevention solutions were also proposed a decade ago [22]. However, despite the extensive research on CSRF attacks and defenses, the CSRF "climbed" the OWASP Top10 chart and stayed there for nearly a decade: it ranked 5th in OWASP Top10 2007 [26], it kept its 5th position in OWASP Top10 2010 [27], it went down to 8th in OWASP Top10 2013 [28]. The IoT/embedded devices and their firmware are no exception to this class of vulnerabilities. The *CVE-2006-5175*, documenting the first CVE-traceable CSRF vulnerability in IoT/embedded firmware, dates back to 10th October 2006 which is around the same time CSRF made it to the OWASP Top10 2007 [26].

2) The Present: Fortunately for the most web-enabled computing environments, the CSRF was recently declared "dead" by many leading practitioners and organizations [24].

Proven and reliable defense and prevention solutions were developed or standardized such as *Same-Site Cookies* [33]. For these reasons, CSRF is finally out from the chart according to OWASP Top10 2017 [29]. On the other hand, the state of IoT/embedded devices relative to the simple-to-fix vulnerabilities, such as CSRF, is worrying. According to recent research, CSRF vulnerabilities (numbers-wise) were among the top to affect web-enabled IoT/embedded devices. Moreover, CSRF affected the most number of firmware images in that particular experiment dataset [15]. At the same time, while CSRF is declared "dead" for the traditional web computing environments, its exploitation is thriving in the IoT/embedded world. Recently there were discovered at least several notorious malware families/campaigns that exploited dozens of CSRF vulnerabilities (most even undocumented and untraceable in CVE and similar databases) in IoT/embedded devices [31]. In fact, as of this writing and based on some of our internal research and data (joint work in progress with Jonas Zaddach, to be soon released [1]), our estimations for the (IoT-)malware/campaigns exploiting CSRF in IoT/embedded devices are as follows. First, only around 20% of the CSRF vulnerabilities observed during the attacks have a CVE number assigned. Second, at least 60% of those vulnerabilities do not even have a CVE or an alternative entry (e.g., VU#, EDB).

3) The Future and Key Takeaways: Relative to even the simplest types of vulnerabilities, the future of IoT/embedded security may look grim if we consider several facts. First, it took *at least a decade* for a relatively simple vulnerability class such as CSRF to get off the OWASP Top10 chart. Second, there are plenty of CSRF-vulnerable IoT/embedded devices and firmware, and those vulnerabilities are very actively exploited. Third, most of those CSRF vulnerabilities are undocumented and untraceable in CVE and similar databases, making it virtually impossible to track them by users and practitioners, and very hard to follow-up on them with relevant vendors. Finally, we did not even consider the more complex vulnerability classes. However, the future of IoT/embedded security can be definitely improved. As of this writing and according to the same internal research and data, we estimate that by closing the CSRF attack surface in IoT/embedded devices known to be affected (e.g., by applying the simple CSRF fixes mentioned above), it is possible to neutralize at least 7% of the analyzed *malware families/campaigns* that target or abuse IoT/embedded devices during their attack lifecycle. Therefore, the key takeaways could be summarized as:

> *Some vulnerabilities are relatively easy to fix. Future-proof solutions already exist (e.g., Same-Site Cookies), and may be safely borrowed from the world of OSes and software for PCs/servers.*
>
> *Even for the "low-hanging fruit" vulnerabilities in IoT/embedded, the CVE-related takeaways from above are still more than applicable.*

REFERENCES

[1] "Firmware.RE Project," http://firmware.re.

[2] "History of Exploit-DB," https://www.exploit-db.com/history/.

[3] "OSVDB: FIN," https://blog.osvdb.org/2016/04/05/osvdb-fin/.

[4] "THC-Hydra," https://github.com/vanhauser-thc/thc-hydra/.

[5] M. Antonakakis, T. April, M. Bailey, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, D. Menscher, C. Seaman, N. Sullivan *et al.*, "Understanding the Mirai Botnet," in *USENIX Security Symposium*, 2017.

[6] K.-H. Baek, S. Bratus, S. Sinclair, and S. W. Smith, "Attacking and Defending Networked Embedded Devices," in *Workshop on Embedded Systems Security (WESS)*, 2007.

[7] ——, "Dumbots: Unexpected botnets through networked embedded devices," *Darthmouth College Computer Science, TR2007-591*, 2007.

[8] T. Baume, "Netcomm NB5 botnet – psyb0t 2.5L," 2009.

[9] M. Bond and R. Anderson, "API-level attacks on embedded systems," *Computer*, vol. 34, no. 10, pp. 67–75, 2001.

[10] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016.

[11] J. Burns, "Cross Site Request Forgery," *An introduction to a common web application weakness, Information Security Partners*, 2005.

[12] Cisco and D. Evans, "The Internet of Things – How the Next Evolution of the Internet Is Changing Everything," https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf.

[13] A. Costin, J. Zaddach, A. Francillon, and D. Balzarotti, "A Large Scale Analysis of the Security of Embedded Firmwares," in *USENIX Security Symposium*, 2014.

[14] A. Costin, A. Zarras, and A. Francillon, "Towards Automated Classification of Firmware Images and Identification of Embedded Devices," in *IFIP International Conference on ICT Systems Security and Privacy Protection (IFIP SEC)*. Springer, 2017, pp. 233–247.

[15] ——, "Automated Dynamic Firmware Analysis at Scale: A Case Study on Embedded Web Interfaces," in *ACM Symposium on Information, Computer and Communications Security (ASIACCS)*, 2016.

[16] A. Cui and S. J. Stolfo, "A quantitative analysis of the insecurity of embedded network devices: results of a wide-area scan," in *Annual Computer Security Applications Conference (ACSAC)*. ACM, 2010.

[17] S. Dua and X. Du, *Data mining and machine learning in cybersecurity*. CRC press, 2016.

[18] Embedi, "Enlarge your botnet with: top D-Link routers (DIR8xx D-Link routers cruisin for a bruisin)," https://embedi.com/blog/enlarge-your-botnet-top-d-link-routers-dir8xx-d-link-routers-cruisin-bruisin/.

[19] FX, FtR, and kim0, "Attacking networked embedded systems," 2002.

[20] D. J. Hand, H. Mannila, P. Smyth *et al.*, "Principles of Data Mining," *MIT Press Books*, vol. 1, 2001.

[21] M. Janus, "Heads of the hydra. malware for network devices," *Securelist, August*, 2011.

[22] N. Jovanovic, E. Kirda, and C. Kruegel, "Preventing cross site request forgery attacks," in *Securecomm and Workshops*. IEEE, 2006.

[23] D. Loshin, *The practitioner's guide to data quality improvement*. Elsevier, 2010.

[24] J. Mannino, "OWASP Top 10 2007–2017: The Fall of CSRF," 2017.

[25] M. Nemec, M. Sys, P. Svenda, D. Klinec, and V. Matyas, "The Return of Coppersmith's Attack: Practical Factorization of Widely Used RSA Moduli," in *24th ACM Conference on Computer and Communications Security (CCS'2017)*. ACM, 2017, pp. 1631–1648.

[26] OWASP, "TOP 10–2007: The ten most critical web application security risks," *The Open Web Application Security Project*, 2007.

[27] ——, "TOP 10–2010: The ten most critical web application security risks," *The Open Web Application Security Project*, 2010.

[28] ——, "Top 10–2013: The ten most critical web application security risks," *The Open Web Application Security Project*, 2013.

[29] ——, "Top 10–2017: The ten most critical web application security risks," *The Open Web Application Security Project*, 2017.

[30] R. Paleari, "Unauthenticated remote access to D-Link DIR-645 devices," http://roberto.greyhats.it/advisories/20130227-dlink-dir.txt.

[31] Proofpoint and Kafeine, "Home Routers Under Attack via Malvertising on Windows, Android Devices," https://www.proofpoint.com/us/threat-insight/post/home-routers-under-attack-malvertising-windows-android-devices.

[32] M. Schwartz, "SSD Advisory – D-Link 850L Multiple Vulnerabilities," https://blogs.securiteam.com/index.php/archives/3364.

[33] M. West and M. Goodwin, "Same-site Cookies," https://tools.ietf.org/html/draft-west-first-party-cookies-07, 2016.