International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. , No. (Mon-20..)

http://dx.doi.org/10.12785/ijcds/XXXXXX

A Survey on IOT Firmware Security

Abdulkarim Katbi[1], Mustafa Hammad[2] and Riadh Ksantini[1]

[1]*Department of Computer Science, University of Bahrain, Sakhir, Kingdom of Bahrain*
[2]*Department of Software Engineering, Mutah University , Jordan*

Received Mon. 20, Revised Mon. 20, Accepted Mon. 20, Published Mon. 20

Abstract: The recent advancements in technology and the rapid movement towards cyber-connected societies have led to the unprecedented use of Internet-of-things (IoT) devices as an effective enabler of such new trends. In fact, IoT devices are continuously penetrating all personal and critical domains like homes, infrastructures, manufacturing, and the military. Despite the numerous benefits, the security concerns of IoT are rising. This is because of the constrained nature of IoT devices and the possible devastating effects of successful attacks on them. While there are many attempts to address the security concerns in IoT from different perspectives, the literature lacks any study that surveys all of the available approaches that focuses on the area of IoT firmware (software) security. Therefore, the aim of this paper is to survey the existing approaches that shed light on IoT firmware security issues and preserving mechanisms. In line with this, current challenges and future research directions were outlined.

Keywords: Internet of Things, IoT, Software, Firmware, Security

1. Introduction

Internet-of-Things (IoT) devices are becoming increasingly essential to today's lives. They are present everywhere including home appliances, industrial and manufacturing, automotive, and healthcare sectors. In fact, the current number of IoT devices exceeded 8 billion and is expected to reach more than 41 billion by 2027 [1]. Besides the massive number of IoT devices, IoT in general aim to facilitate and augment the lives of people in unprecedented ways. For example, IoT devices can be used to reduce electricity bills by managing smart lamps, providing auto-driving capabilities in vehicles, and providing protection to homes and facilities.

Due to the pervasive and heterogeneous nature of the IoT, many device types, standards, communication protocols, and implementation schemes exist. In fact, the heterogeneity of IoT devices helps to provide optimal solutions for specific domain applications or use cases. However, this poses many problems and challenges that can hinder their effective implementation and use. Among the various issues is the issue of IoT device security

Since security is a cornerstone of any technology, it is particularly important in the field of IoT devices. Recent studies demonstrated the effect of breaching the security of IoT firmware and the possible devastating consequences of it. For example, Ronen et al. [2] demonstrated that an attack on smart lamps being operated in a factory or a town could lead to an immediate shutdown of the lighting systems. A more serious attack is to target the patient-related IoT devices to affect their operability or tamper their actual readings, which could threaten the life of the people [3]. The exploitation of vulnerabilities in IoT firmware and weak device firmware credentials have led to massive denial of service attacks on IoT devices caused by Mirai botnet [4]. Other IoT firmware security breach examples include vehicle hacking [5], IP cameras distributed denial of service [6], and spamming scenarios of general home appliances [7]. Thus, attempting to secure all dimensions of the IoT ecosystem is of utmost importance. One of the essential elements of securing IoT devices is the protection of IoT firmware.

The purpose of this paper is to shed light on the security mechanisms being proposed and/or adopted by scholars to protect the firmware of IoT devices. In order to do so, a comprehensive survey in the area of IoT firmware security is conducted. To demonstrate how the surveyed studies contribute to the work of IoT firmware security, an IoT firmware security taxonomy is provided. Additionally, the illustration of the basic concepts and comparison among different works was also presented.

2. Background and overall taxonomy

IoT devices are characterized by limited computational capacity, restricted power consumption, and domain-specific application scenarios. Thus, the process of developing IoT devices comprises the design and development of hardware-related equipment and the actual firmware. Figure 1 il-

E-mail address: a.katbi@uob.edu.bh, hammad@mutah.edu.jo, rksantini@uob.edu.bh

lustrates the main concepts and processes involved in the creation and dissemination of IoT devices and their associated firmware. The first process comprises an IoT-related software development stage. It includes determining the IoT firmware's requirements, specifications, and design considerations. The outcome of the development process is the source code of IoT devices. Since the same manufacturer could produce mass amounts of different IoT devices, reusing the code is a common theme in the development and production process. Commercial off-the-shelf (COTS) IoT-specific code components exist as well. Such components are an attractive solution for enabling faster and easier production of IoT devices. Therefore, some manufacturers prefer to use the already developed components and integrate them into the design and implementation of IoT devices. Thus, firmware code could be almost totally developed by the commercial off-the-shelf components with minimal configuration and adaptation. Alternatively, only some portions of the source code could be constructed from COTS components and the remaining are in-house developed code.

The need to add new features, support more advanced IoT hardware or deal with certain security issues dictate the development of newer versions of the IoT firmware. With this respect, newer versions could have the same blocks of an older version of the source code augmented by the new functionality. Alternatively, the newer version could undergo a major reconstruction activity in which the majority of the code is new. In both situations, IoT firmware security threats exist. With respect to the first case, old undetected vulnerabilities could evolve into newer versions making them prone to attacks. In the second case, the use of COTS components, and even the newly developed code could have undiscovered security vulnerabilities. Therefore, the examination of IoT firmware security should take into account the above scenarios.

Once the source code is compiled, IoT firmware in binary format is created and becomes ready to be loaded into the hardware of the IoT device. With this respect, most IoT manufacturers do not provide the source code of their IoT devices. Rather, a compressed, binary-based firmware is made available to the users [8]. For legacy IoT devices, even the compressed firmware sometimes cannot be found and is not provided by the manufacturer of IoT devices. To obtain it, some techniques can be applied to extract the compressed firmware directly from the IoT hardware. Even though the extraction task seems to be possible, it is a challenging activity in which the success of it cannot be guaranteed.

The lack of source code for IoT devices makes investigating the security of IoT firmware a challenge. This is because traditional static analysis techniques for investigating security and vulnerability issues in the code cannot be implemented. This leads to adopting the option of reverse engineering the IoT firmware, in some way, to get a certain level of knowledge regarding the possible vulnerable elements of the firmware. Reverse engineering the IoT firmware pose many issues such as the ability to accurately identify the functions of the source code and the attributes of the firmware. Therefore, this is an active area of scholars' focus with respect to IoT firmware security.

Another area of security concern is the update process of IoT firmware. Many IoT devices offer either manual or automatic update options once newer versions become available. Updating the IoT firmware could happen over the air by means like 5G, Wi-Fi, and Zigbee protocols. Thus, the update process has to be secure. Malicious and tampered IoT firmware could reside in an IoT device as a result of an insecure update process. Consequently, the security and functionality of IoT devices are affected.

To address the above security concerns, there exist different approaches proposed in the literature to secure IoT firmware. With this respect, it is notable that all of the approaches tend to consider specific elements of the IoT firmware development, rollout, and update process that are shown in Figure 1. However, different analysis and evaluation tools are adopted by scholars in an attempt to cover and provide robust solutions for detecting vulnerabilities and securing IoT firmware.

Figure 2 presents the overall taxonomy of the approaches that focused on IoT firmware security. Three broad categories of approaches have been identified. In particular, IoT firmware security can be examined from the perspective of software-based solutions, hardware-based solutions, and approaches that focus on the process of updating the firmware in a secure way.

Software-based solutions can be examined from the perspective of an algorithm or type of technique being followed to detect IoT firmware vulnerabilities and secure the IoT firmware. With this regard, three main categories of techniques have been identified. The general and fuzzing-based approaches utilize either software-based algorithms or fuzzing techniques in order to discover the vulnerabilities of IoT firmware. Classical Machine Learning (ML) and Deep learning (DL)-based techniques are also utilized to train models that can learn the IoT firmware characteristics and assist in detecting the vulnerabilities. Finally, blockchain-based techniques adopt the different implementations of blockchain, like public or private blockchain methods, to provide a secure environment for IoT devices, in general, and IoT firmware, in particular.

Hardware-based solutions focus on the hardware of IoT as a possible mechanism to secure the IoT firmware. With this regard, the work in the literature presents several studies that attempt to propose a new IoT hardware design that can enforce IoT firmware security and prevent intruders from reverse engineering the compressed IoT firmware. Additionally, other approaches involve conducting performance evaluations among various IoT devices to pinpoint effective

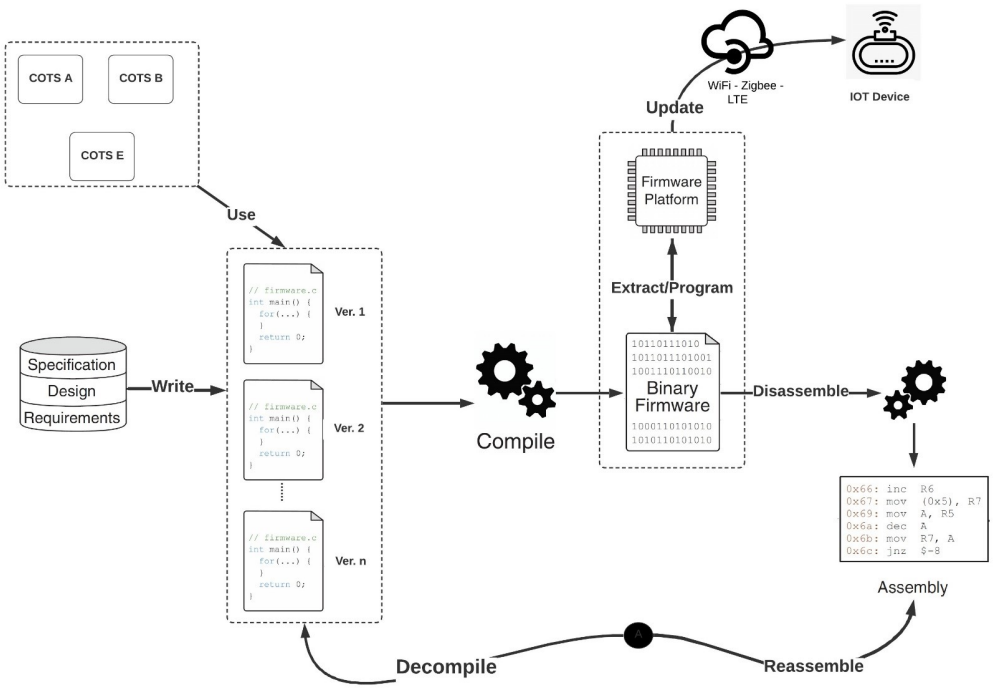

Figure 1. IoT firmware security focus areas.

and weak design issues, as well as utilizing hardware components as the basis for firmware security analysis and protection.

Secure IoT firmware roll-out is the third category of approaches, primarily focused on proposing new frameworks, systems, and protocols that concern providing a secure environment for updating the IoT firmware. Thus, multiple elements of the IoT ecosystem are considered in this category, such as the transfer media (WIFI, 4G), the protocols used, and the type of update (manual vs. automatic, online vs. offline).

3. Literature review

This section aims to conduct a detailed review of all IoT firmware security approaches related to software-based, hardware-based, and IoT update process-based studies. Additionally, a comparative analysis of the studies within each category is presented. However, before proceeding into the detailed review, the overall statistics concerning the number and type of studies related to IoT firmware security will be presented first.

Figure 3 presents the total number of publications that considered securing the IoT firmware and published in the IEEE database. The Figure exhibits a steady increase in the number of publications over the previous five years. This indicates that there is an increasing interest among scholars to study IoT firmware. Specifically, in 2014 and 2015 almost none of the researchers were concerned with analyzing the IoT firmware vulnerability and security issues. A limited number of studies, only 5, were published in 2016. Since then, the number of publications showed to increase steadily.

Figure 4 shows the breakdown of the categories of the surveyed studies that focused on IoT firmware security. As shown in the Figure, two-thirds of the approaches were found to be in the software-based category. The Figure exhibits an almost equal distribution of studies that focused either on the hardware or IoT firmware update process.

A. *Software based solutions*

Software-based solutions focus on analyzing the IoT firmware in order to assist in detecting the vulnerabilities and ensuring secure IoT device firmware. There are several approaches that examined the IoT firmware. The approaches can be categorized into general and fuzzing-based, machine learning, and blockchain approaches. Since the source code of the majority of IoT devices is not open, the software-

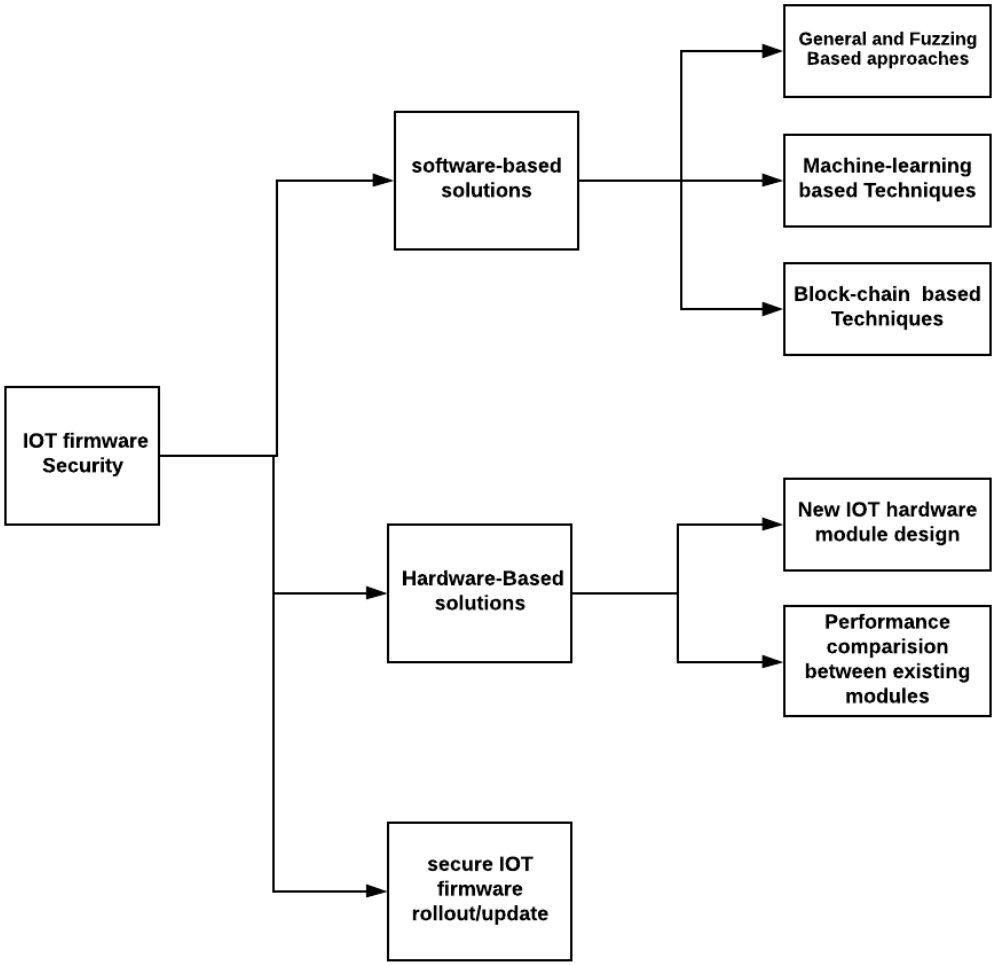

Figure 2. Overall taxonomy of IoT security studies.

based studies were found to focus on the examination of the compressed firmware from the perspective of analyzing the binary code of it. Other approaches utilized the file or directory structure to analyze the IoT firmware. In the following subsection, a review of the most relevant and recent studies will be provided.

1) General and Fuzzing Based approaches

This category comprises all of the work that examined the IoT firmware from a viewpoint different than machine learning or blockchain approaches. Xu et al. [9] proposed a manual search process for analyzing the IoT firmware for vulnerabilities. The proposed approach focuses on dissembling the IoT firmware and applying feature extraction and code search processes. Whereas Feature extraction is used to determine the functionalities of IoT devices, code search is used to identify the code and data fragments within the identified functions. Chen et al. [10] argued that IoT firmware vulnerabilities arise because of code reuse and therefore there is a need to identify homologs of IoT firmware. To do so, readable strings from firmware binaries were retrieved and a randomized algorithm called MinHash handled the similarity check process between different IoT firmwares. The work of Zhu et al. [11] goes beyond finding similarities between IoT firmwares to establishing an IoT firmware gene for the IoT devices. Thus, multiple firmwares can be compared based on the gene information. Their proposed system, called FCGES, extracts multiple features from the binary IoT firmware, processes them, and utilizes the hypothesis margin to generate the unique gene information.

More focused studies have utilized the control flow graph (CFG) to identify specific IoT firmware information that can be used for vulnerability analysis. Zhang et al. [12] followed a specific approach to identify only the version

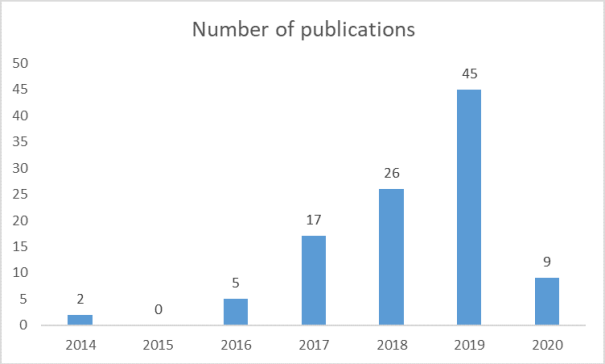

Figure 3. Number of publications concerning IoT firmware security by year.

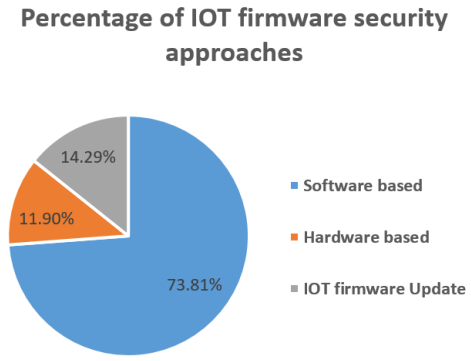

Figure 4. Breakdown of IoT firmware security approaches.

information of IoT firmware. Their study utilized the readable strings in the IoT firmware binary to detect version information and in case of any missing strings, String Recover Engine (SRE) handles the string recovery process. Besides version information, Sun et al. [13] focused on the embedded IoT controller binaries to identify information related to the control and state estimation algorithms being used in the IoT devices.

A well-known software vulnerability technology in traditional systems called fuzzing is also implemented in the field of IoT to detect possible vulnerabilities in the IoT firmware. Xie et al. [14] utilized static analysis along with fuzzing to detect the logical flaws in IoT firmware. Since logical flows do not cause system crashes as opposed to memory flows, their detection tends to be less likely to be discovered easily by the programs in Linux or Android. Specifically, two types of authentication bypass flaws were detected by the proposed method; CGIs without authentication protections and complex backdoors. Zheng et al. [15] focused on implementing the fuzzing technique to detect vulnerabilities in Linux-based IoT programs. In their work, static analysis is carried out to collect useful keywords and inputs that feed the fuzzier. Their proposed framework offers real-time monitoring of vulnerabilities while supporting accurate code coverage and new fork mode. Other approaches that utilized the vulnerability-oriented Fuzzing approach include [16],[17], and [18].

Table I shows a comparison between the studies mentioned in this section. As shown in the Table, almost all of the surveyed studies validated their approaches and frameworks by means of prototypes or evaluations. Additionally, the white-box Fuzzy based approach was the dominant fuzzing technique being used by the surveyed studies.

2) Machine learning based approaches

Supervised machine learning techniques were applied to the field of IoT firmware vulnerability detection and prevention. Miettinen et al. [19] proposed a system called IoT SENTINEL capable of identifying and preventing vulnerable IoT devices from affecting the other devices in the network. Specifically, IoT SENTINEL utilizes a classification model (random forest) to identify the device type and the IoT firmware version. Based on the identified information and by consulting a vulnerability database, the system will restrict the communication of IoT devices if it determines that the device is vulnerable. Lin et al. [20] asserted the issue of existing IoT vulnerability assessment approaches being restricted to support specific architecture types and proposed a more generic approach to overcome IoT firmware vulnerabilities. The proposed approach focuses on the function level of IoT firmware binary and thus detects the suspicious functions in other IoT binaries. A preliminary scanning phase utilizes Support Vector Machine (SVM) to identify possible suspicious functions. To increase the accuracy of the detection process, a graph similarity phase based on the attributed control flow graph is used. Their proposed approach showed to be effective in real-world scenarios.

Instead of focusing on a single machine-learning method, other researchers applied several machine-learning algorithms to assist in securing the IoT firmware. Lee et al. [21] experimented with four types of machine learning algorithms to identify IoT firmware type information like manufacturer, device type, and architecture type. The applied algorithms are Neural Network (NN), Random Forest(RF), Support Vector Machine (SVM), and Two Step Fusion Algorithm. The results showed varying degrees of accuracy with respect to device vendor identification. YU et al. [22] focused on utilizing two algorithms, Support Vector Machine (SVM) and Block-o-Matic (BoM) to identify the firmware information of IoT devices. Their approach was based on utilizing the already known weak passwords and login pages of IP-enabled IoT devices to scan them to get the firmware information required. According to YU et al. [22], their approach achieved an accuracy rate of 59.97%, which outperformed other approaches for detecting IoT device firmware information.

Deep learning and neural network techniques showed the potential to contribute to detecting IoT firmware vulnerabilities. The study of Wu et al. [23] was motivated

TABLE I. Comparison between the surveyed general and fuzzing Based approaches.

Study	Fuzzing based	Focus Level	Demonstration or evaluation	Fuzzing Type
XU et al.[9]	Search based (manual)	Function level	yes	N/A
CHEN et al. [10]	Minhash/ Randomized Algorithm	readable strings	yes	N/A
Zhang et al.[12]	String Recovering Engine based on CFG	readable strings (Only firmware version related strings)	yes	N/A
Sun et al.[13]	Abstract Syntax Tree Mapping and CFG	Functions and algorithm's operations.	yes	N/A
Xie et al.[14]	Fuzzing based	Logical flaws	No	White-box
Zheng et al.[15]	Fuzzing based	process monitor, system call monitor and context analyzer	yes	Grey-Box
Gui et al.[16]	Fuzzing based	Function based	yes	White-box
ZHU et al.[17]	Fuzzing based	Memory Flaws	Yes	White-box
Xiao et al. [18]	Fuzzing based	buffer overflow, heap overflow, DDOS and backdoor	yes	White-box

by the fact that most of the cross-architecture approaches to detect IoT vulnerabilities were based on Control Flow Graphs (CFG). Since the same IoT firmware can produce different assembly codes if different operating systems of compilation parameters are used, the practicality of CFG is doubted. Rather, a deep learning approach based on a neural network is proposed to reduce the differences between the produced assembly codes. Thus, cross-platform detection is enabled. Liu et al. [8] utilized an attention model to learn the high-level features from the extracted IoT firmware binary. A prediction model is constructed based on the obtained high-level features. Their results demonstrated superior performance and an ability to be employed in real-world scenarios.

Another work done by Wagn et al. [24] has reconfirmed the inefficiency of the control flow graph and simple feature matching in obtaining highly accurate vulnerability assessment results. To address the mentioned issues, a two-stage approach based on code similarity is proposed. Their approach utilized neural networks to analyze function similarities based on function embedding. In order to obtain more accurate results, the local call flow graphs of the functions are calculated in the second stage. Zhang et al. [25] utilized neural networks to provide an accurate prediction of the firmware version of an IoT device. As opposed to other approaches, their approach was based on the examination of the directory information of the IoT firmware to identify the version information. Both the timing and structural information were considered to get accurate predictions.

Table II summarizes the studies that utilized machine and deep learning to assist in securing the IoT firmware. As shown in the Table, it is clear that the majority of studies were focused on examining the binary information of the compressed firmware. Additionally, all of the studies applied some sort of evaluation to validate their models. Finally, the implementation of various machine-learning techniques reflects the ability and future of this discipline in supporting the detection of vulnerabilities and securing IoT firmware.

3) Block Chain based solutions

Blockchain technologies were also considered in the area of IoT firmware security. The main purpose of blockchain in the IoT firmware domain is to ensure the integrity and reliability of IoT firmware, especially during the update process [26]. Thus, security threats like rollback attacks or man-in-the-middle attacks can be avoided [27]. Additionally, relying on traditional client-server architectures for updating the IoT firmware is a security-risk-prone approach. In contrast, Blockchain techniques offer secure distributed architectures that do not suffer from the single point of failure issue [26], [28] . With this regard, several studies were motivated by the advantages of block-chain techniques and adapted them to propose block-chain-based IoT firmware security solutions.

Public blockchain offers a highly immutable, decentralized, and permission-less architecture [29] that can be used to secure IoT firmware. Lim et al. [30] proposed ChainVeri public blockchain firmware verification system. Their proposed system consists of a palette, trader, and three-way trade protocol process. The trader is responsible for verifying the firmware information and producing the

TABLE II. Comparison between the surveyed Machine Learning (ML) approaches.

Study	purpose	Machine learning method	Evaluation	Focus
Miettinen et al.[19]	Identify the IOT device type and firmware version. IF vulnerable, network communication is restricted	Supervised ML Classifications (Random Forest) & Tiebreaks (edit distance)	Yes	Binary
Lin et al.[20]	A cross architecture based system to identify IOT firmware vulnerabilities based on function level vulnerabilities	Support Vector Machine (SVM) and Attributed Control Flow Graph (ACFG)	Yes	Binary
Wu et al. [21]	Proposing a solution to identify the vulnerabilities of different binary assemblies of the same IOT firmware	Deep learning, Neural Network Method	Yes	Binary
Liu et al. [8]	an approach for automatic detection of IOT vulnerabilities from the binary file	Deep learning-based approach. Attention Model	Yes	Binary
Wang et al. [22]	an approach that deal of the deficiencies of simple feature and control graph matching by utilizing deep learning to analyze code similarities	Neural Network	Yes	Binary
Zhang et al. [23]	Focusing on the structural and timing information to predict the firmware version based on the examination of firmware file directories.	Neural Network	Yes	File (component based)
Chen et al.[24]	experimenting with several machine learning Algorithms to identify the IOT firmware type information	Random Forest (RF) algorithm, Support Vector Machine (SVM) algorithm, Neural Network (NN) algorithm, Two-step fusion algorithm (custom)	Yes	Binary
Yu et al. [25]	scanning the IOT devices connected and operating in a network by a tool and trying to hack the password to get firmware version	Support Vector Machine (SVM) and recurrent neural network algorithm (BAM)	Yes	Web-based Login page

palette that contains the verified information. All IoT devices connect to the traders while the triangular trade handles the actual communication and delivery of IoT firmware. Besides bitcoin technology, other approaches utilized the Ethereum public blockchain approach as in [28],[31].

Apart from public blockchain, other studies have adopted less decentralized but more efficient approaches such as Consortium and public blockchain techniques. The work of [32] has utilized multi-chain, private blockchain technology to preserve the confidentiality, integrity, and availability of IoT firmware. Each IoT device has to periodically scan for a newer firmware version through the blockchain nodes. The consensus protocol is used by the nodes to verify the firmware that has been recently updated and provided by the vendor. Whenever an IoT device requests an update from the vendor, a transaction for the update is created. Sun and Kim [27] acknowledged the possible low performance of public blockchains due to the use of Pow which creates disk storage problems. Therefore, they proposed a private blockchain solution. The proposed solution utilizes the hyper-ledger fabric to construct the private blockchain network. To protect the integrity of IoT firmware, the InterPlanetary File System (IPFS) is used. Additionally, the URL of IPFS is protected by the blockchain. Choi and Lee [[26]] proposed a private or consortium architecture that consists of registration nodes, retrieval nodes, and general nodes. While registration nodes are responsible for processing the registration of vendors' manifest and firmware files, retrieval nodes deal with downloading the manifest and firmware files to IoT devices.

Table **??** shows a comparison between the surveyed IoT firmware block-chain-based security solutions. It is obvious that the studies in general focus on either the vendor initiating the update process (PUSH method) or the IoT device requesting the newer firmware version (PULL method). Only one study considered both of them [28]. Additionally, many proposed approaches considered the requirement of supporting multiple IoT vendors. Thus, blockchain solutions have the potential to offer a universally applicable architecture.

B. Hardware based solutions

There is a number of studies found in the literature focusing on Hardware-level IoT security. Cyr et al. [33] argued that cloning electronic devices, including IoT devices poses serious security threats and thereby effective secure mechanisms have to be developed and implemented. Among the various threats, device cloning can cause transmitting secret information from unauthorized parties by utilizing the cloned devices. To overcome the cloning issue, the authors

TABLE III. Comparison between the surveyed IoT firmware block-chain-based security solutions.

Study	Using smart contracts	Multi-vendor support	firmware storage	Applied Blockchain platform	Performance evaluation case (Prototype implementation)	Update method
Choi and Lee [26]	no	yes	Distributed in blockchain network	private or consortium blockchain	No	pull
Boudguiga, et al. [32]	no	no	blockchain network	private blockchain	yes	Pull
Son and Kim [27]	no	Yes	dedicated peers in the blockchain network	private blockchain : Hyper ledger Fabric	yes	Pull
Lim, et al.[30]	no	yes	vendor repository	Proof-of-work (POW) based platform (bitcoin)	Yes	Pull
Yohan [28]	yes	yes	vendor and broker repositories	Ethereum public blockchain	no	pull + push
Pillai [31]	yes	yes	vendor repository	Ethereum public blockchain platform. Proof-of-work (POW) based platform	no	push method

have proposed a low-cost firmware obfuscation method. The proposed method focuses on hardware-level IoT security by swapping a subset of carefully chosen instructions from the firmware. Thus, the attacker will not be able to clone the correct firmware instructions. A small memory in the IoT device is dedicated to storing the swapped instructions and during the IoT device operation, this memory is used along with a PUF-generated identifier to reconstruct the original firmware.

Ronen et al. [2] described how devastating the effect of low security of IoT devices on the overall network and infrastructure of a country. In order to show such an effect, experimentation utilizing Philips smart lamps as a testing platform was carried out. Among the various activities of the experimentation, they demonstrated a setup of hardware attack to reverse engineering the IoT firmware and affect the over-the-air update process. Other Hardware focused studies were found measuring and evaluating the performance of hardware IoT cryptographic performance[34],[35] and networking operations[34].

C. *Secure firmware rollout approaches*

Zandberg et al. [36] argued that despite the strong positive economic impact of IoT devices, their weak security is a real threat that should be taken into consideration carefully. A specific area that concerned them is the absence of effective and secure update mechanisms. Such absence could make the majority of IoT devices unpatched and therefore very likely to be vulnerable to various attacks. In response to this limitation, a prototype for secure firmware updates of constrained IoT devices is proposed. The prototype is based on the existing open standards and open-source libraries. Based on the performance evaluation, their implementation is proven to be able to provide a state-of-the-art secure update process that supports the very resource-limited IoT devices .

A real-world implementation of a secure update process was carried out by Teng et al. [37]. In order to protect home routers from possible denial of service and other cyber-attacks, the authors proposed an efficient secure update process. The proposed process focused on one IoT device type, the home routers. The exact update process was explained by a detailed architecture that incorporates the routers as IoT devices, and the ISP's network management system and operation support systems as the coordinating and managing entities for firmware updates. The proposed process was applied to over 1 million routers connected to the largest IPS network in Taiwan. Over 96% of successful completion of updates over multi-vendor routers were achieved by the proposed process.

The work of [38] concentrated on proposing an object notation that can be embedded across the various protocols to support a secure firmware update process. Their proposed notation addresses several issues such as the inability to identify partial or delta updates of the IoT firmware and resuming the transfer of disconnected update process. Despite the benefits of such notation, the authors did not demonstrate the applicability of it in a real-world or proof-of-concept implementation. Other studies concerning the

firmware update security were found focusing on trust zone technology [39], combining IoT over-the-air update with JTAG security [40], proposing a generic unified firmware update scheme [41] and making a general survey of applicable secure update mechanisms [42].

4. Research challenges

There exist several challenges that have to be taken into consideration regarding the surveyed IoT firmware security studies. Many IoT security studies that proposed solutions for securing IoT firmware were found to lack practical and real-world implementations. In contrast, either a proof-of-concept (demonstration) or just an explanation of the method is provided. In fact, real word implementation is critical to assure the feasibility and interoperability of the recommended solutions as well as to enable finding any shortages in the current solutions. Additionally, many approaches were found targeting domain-specific [43] or architecture-specific IoT Types [33]. Since IoT devices operate in heterogeneous architectures, there is a need to consider the cross-architecture requirements and propose more generic approaches. The binary analysis of firmware without having the source code also poses its own challenges. Since this type of analysis depends on many factors including the type of architecture used to run/execute the binary code of the firmware, the accuracy of interpreting the results is not always optimum. Thus, binary analysis techniques face many challenges that were not present with static and dynamic analysis approaches of ordinary desktop-based software systems. For machine learning-based studies, improving the accuracy of the detection rate is a challenge as reported by [44]. Finally, while detecting the security issues of IoT firmware has received good attention from the surveyed studies, there are limited approaches that focused on recovering the attacked/corrupted IoT firmware, like in [45].

5. Future Opportunities

Based on the surveyed studies and in line with the current challenges, there exist several promising directions that can assist in improving IoT firmware security and dealing with the issues faced by current studies. First, more cross-platform, cross-domain approaches and architectures for securing the IoT firmware are required. With this respect, researchers can expand their existing work to support other devices, architectures as well as the domain of application. Alternatively, a more holistic approach to IoT firmware security can be adopted and implemented. With respect to machine learning approaches, there is a need to experiment with multiple approaches in order to improve the accuracy and efficiency of the vulnerability detection process of IoT devices.

Additionally, it is interesting to see future studies that combine multiple methods and approaches to secure IoT firmware. For example, combining blockchain technologies with binary-based firmware verification techniques. This scenario ensures safe IoT firmware distribution and update by blockchain as well as ensures that the vulnerable aspects of the firmware are detected by binary-based vulnerability analysis. Another direction for future work is to develop frameworks and approaches that consider managing and distributing IoT firmwares across cloud computing fog nodes. Finally, the beneficial use cases of Software Defined Networks (SDN) for managing IoT firmware security should be explored by future studies.

6. Conclusion

This paper reviewed the recent work done in the area of IoT firmware security. With this respect, it has been noticed that there is an increased interest among scholars to study and evaluate the various aspects of IoT security. Scholars have been motivated by the fact that IoT devices are penetrating all aspects of our lives and the lack of proper firmware security mechanisms could result in devastating consequences related to all domains. IoT firmware security approaches were found related to software-based approaches, hardware-based approaches, and approaches focusing on the IoT firmware update process. With this respect, it has been observed that the majority of surveyed studies were software-based approaches. Despite the various benefits of IoT devices, the outlined challenges set the direction of future studies that should tackle the existing issues.

References

[1] "The Internet Of Things 2020: Here's what over 400 IoT decision-makers say about the future of enterprise connectivity and how IoT companies can use it to grow revenue - Insider Intelligence Trends, Forecasts & Statistics." [Online]. Available: https://rb.gy/1s0h4

[2] E. Ronen, A. Shamir, A. Weingarten, and C. O'Flynn, "Iot goes nuclear: Creating a zigbee chain reaction," in *2017 IEEE Symposium on Security and Privacy (SP)*, 2017, pp. 195–212.

[3] A. Chacko and T. Hayajneh, "Security and Privacy Issues with IoT in Healthcare," *EAI Endorsed Transactions on Pervasive Health and Technology*, vol. 4, no. 14, 2018.

[4] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas, and Y. Zhou, "Understanding the mirai botnet," in *Proceedings of the 26th USENIX Conference on Security Symposium*, ser. SEC'17. USA: USENIX Association, 2017, p. 1093–1110.

[5] "Hackers Remotely Kill a Jeep on the Highway—With Me in It — WIRED." [Online]. Available: https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/

[6] "IoT Home Router Botnet Leveraged in Large DDoS Attack." [Online]. Available: https://blog.sucuri.net/2016/09/iot-home-router-botnet-leveraged-in-large-ddos-attack.html

[7] "Your Fridge is Full of SPAM: Proof of An IoT-driven Attack — Proofpoint US." [Online]. Available: https://www.proofpoint.com/us/threat-insight/post/Your-Fridge-is-Full-of-SPAM

[8] S. Liu, M. Dibaei, Y. Tai, C. Chen, J. Zhang, and Y. Xiang, "Cyber vulnerability intelligence for internet of things binary," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 2154–2163, 2020.

[9] Y. Xu, T. Liu, P. Liu, and H. Sun, "A search-based firmware code analysis method for iot devices," in *2018 IEEE Conference on Communications and Network Security (CNS)*, 2018, pp. 1–2.

[10] Y. Chen, H. Li, W. Zhao, L. Zhang, Z. Liu, and Z. Shi, "Ihb: A scalable and efficient scheme to identify homologous binaries in iot firmwares," in *2017 IEEE 36th International Performance Computing and Communications Conference (IPCCC)*, 2017, pp. 1–8.

[11] X. Zhu, Q. Li, P. Zhang, and z. chen, "A firmware code gene extraction technology for iot terminal," *IEEE Access*, vol. 7, pp. 179 591–179 604, 2019.

[12] W. Zhang, Y. Chen, H. Li, Z. Li, and L. Sun, "Pandora: A scalable and efficient scheme to extract version of binaries in iot firmwares," in *2018 IEEE International Conference on Communications (ICC)*, 2018, pp. 1–6.

[13] P. Sun, L. Garcia, and S. Zonouz, "Tell me more than just assembly! reversing cyber-physical execution semantics of embedded iot controller software binaries," in *2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2019, pp. 349–361.

[14] W. Xie, Y. Jiang, Y. Tang, N. Ding, and Y. Gao, "Vulnerability detection in iot firmware: A survey," in *2017 IEEE 23rd International Conference on Parallel and Distributed Systems (ICPADS)*, 2017, pp. 769–772.

[15] Y. Zheng, Z. Song, Y. Sun, K. Cheng, H. Zhu, and L. Sun, "An efficient greybox fuzzing scheme for linux-based iot programs through binary static analysis," in *2019 IEEE 38th International Performance Computing and Communications Conference (IPCCC)*, 2019, pp. 1–8.

[16] Z. Gui, H. Shu, F. Kang, and X. Xiong, "Firmcorn: Vulnerability-oriented fuzzing of iot firmware via optimized virtual execution," *IEEE Access*, vol. 8, pp. 29 826–29 841, 2020.

[17] L. Zhu, X. Fu, Y. Yao, Y. Zhang, and H. Wang, "Fiot: Detecting the memory corruption in lightweight iot device firmware," in *2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, 2019, pp. 248–255.

[18] F. Xiao, L. Sha, Z. Yuan, and R. Wang, "Vulhunter: A discovery for unknown bugs based on analysis for known patches in industry internet of things," *IEEE Transactions on Emerging Topics in Computing*, vol. 8, no. 2, pp. 267–279, 2020.

[19] M. Miettinen, S. Marchal, I. Hafeez, N. Asokan, A. Sadeghi, and S. Tarkoma, "Iot sentinel: Automated device-type identification for security enforcement in iot," in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, 2017, pp. 2177–2184.

[20] H. Lin, D. Zhao, L. Ran, M. Han, J. Tian, J. Xiang, X. Ma, and Y. Zhong, "Cvssa: Cross-architecture vulnerability search in firmware based on support vector machine and attributed control flow graph," in *2017 International Conference on Dependable Systems and Their Applications (DSA)*, 2017, pp. 35–41.

[21] S. Lee, J. Paik, R. Jin, and E. Cho, "Toward machine learning based analyses on compressed firmware," in *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, vol. 2, 2019, pp. 586–591.

[22] D. Yu, L. Zhang, Y. Chen, Y. Ma, and J. Chen, "Large-scale iot devices firmware identification based on weak password," *IEEE Access*, vol. 8, pp. 7981–7992, 2020.

[23] H. Wu, H. Shu, F. Kang, and X. Xiong, "Bin: A two-level learning-based bug search for cross-architecture binary," *IEEE Access*, vol. 7, pp. 169 548–169 564, 2019.

[24] Y. Wang, J. Shen, J. Lin, and R. Lou, "Staged method of code similarity analysis for firmware vulnerability detection," *IEEE Access*, vol. 7, pp. 14 171–14 185, 2019.

[25] W. Zhang, H. Li, H. Wen, H. Zhu, and L. Sun, "A graph neural network based efficient firmware information extraction method for iot devices," in *2018 IEEE 37th International Performance Computing and Communications Conference (IPCCC)*, 2018, pp. 1–8.

[26] S. Choi and J. Lee, "Blockchain-based distributed firmware update architecture for iot devices," *IEEE Access*, vol. 8, pp. 37 518–37 525, 2020.

[27] M. Son and H. Kim, "Blockchain-based secure firmware management system in iot environment," in *2019 21st International Conference on Advanced Communication Technology (ICACT)*, 2019, pp. 142–146.

[28] A. Yohan and N. Lo, "An over-the-blockchain firmware update framework for iot devices," in *2018 IEEE Conference on Dependable and Secure Computing (DSC)*, 2018, pp. 1–8.

[29] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *2017 IEEE International Congress on Big Data (BigData Congress)*, 2017, pp. 557–564.

[30] J. Lim, Y. Kim, and C. Yoo, "Chain veri: Blockchain-based firmware verification system for iot environment," in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, 2018, pp. 1050–1056.

[31] A. Pillai, M. Sindhu, and K. V. Lakshmy, "Securing firmware in internet of things using blockchain," in *2019 5th International Conference on Advanced Computing Communication Systems (ICACCS)*, 2019, pp. 329–334.

[32] A. Boudguiga, N. Bouzerna, L. Granboulan, A. Olivereau, F. Quesnel, A. Roger, and R. Sirdey, "Towards better availability and accountability for iot updates by means of a blockchain," in *2017 IEEE European Symposium on Security and Privacy Workshops (EuroS PW)*, 2017, pp. 50–58.

[33] B. Cyr, J. Mahmod, and U. Guin, "Low-cost and secure firmware obfuscation method for protecting electronic systems from cloning," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3700–3711, 2019.

[34] B. Pearson, L. Luo, Y. Zhang, R. Dey, Z. Ling, M. Bassiouni, and

X. Fu, "On misconception of hardware and cost in iot security and privacy," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, 2019, pp. 1–7.

[35] H. An, S. Yun, and O. Yi, "Performance evaluation of firmware cryptographic modules on various mcu environments," in *2016 International Conference on Platform Technology and Service (PlatCon)*, 2016, pp. 1–4.

[36] K. Zandberg, K. Schleiser, F. Acosta, H. Tschofenig, and E. Baccelli, "Secure firmware updates for constrained iot devices using open standards: A reality check," *IEEE Access*, vol. 7, pp. 71 907–71 920, 2019.

[37] C. Teng, J. Gong, Y. Wang, C. Chuang, and M. Chen, "Firmware over the air for home cybersecurity in the internet of things," in *2017 19th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, 2017, pp. 123–128.

[38] K. Doddapaneni, R. Lakkundi, S. Rao, S. G. Kulkarni, and B. Bhat, "Secure fota object for iot," in *2017 IEEE 42nd Conference on Local Computer Networks Workshops (LCN Workshops)*, 2017, pp. 154–159.

[39] R. Dhobi, S. Gajjar, D. Parmar, and T. Vaghela, "Secure firmware update over the air using trustzone," in *2019 Innovations in Power and Advanced Computing Technologies (i-PACT)*, vol. 1, 2019, pp. 1–4.

[40] S. K. Kumar, S. Sahoo, K. Kiran, A. K. Swain, and K. K. Mahapatra, "A novel holistic security framework for in-field firmware updates," in *2018 IEEE International Symposium on Smart Electronic Systems (iSES) (Formerly iNiS)*, 2018, pp. 261–264.

[41] B. Hong, J. Huang, T. Ban, R. Isawa, S. Cheng, D. Inoue, and K. Nakao, "Measurement study towards a unified firmware updating scheme for legacy iot devices," in *2019 14th Asia Joint Conference on Information Security (AsiaJCIS)*, 2019, pp. 9–15.

[42] A. Kolehmainen, "Secure firmware updates for iot: A survey," in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, 2018, pp. 112–117.

[43] P. Sun, L. Garcia, and S. Zonouz, "Tell me more than just assembly! reversing cyber-physical execution semantics of embedded iot controller software binaries," in *2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2019, pp. 349–361.

[44] W. Xie, Y. Jiang, Y. Tang, N. Ding, and Y. Gao, "Vulnerability detection in iot firmware: A survey," in *2017 IEEE 23rd International Conference on Parallel and Distributed Systems (ICPADS)*, 2017, pp. 769–772.

[45] M. Xu, M. Huber, Z. Sun, P. England, M. Peinado, S. Lee, A. Marochko, D. Mattoon, R. Spiger, and S. Thom, "Dominance as a new trusted computing primitive for the internet of things," in *2019 IEEE Symposium on Security and Privacy (SP)*, 2019, pp. 1415–1430.