A Large-Scale Analysis of IoT Firmware Version Distribution in the Wild

Frank Ebbers

Abstract—This paper examines the up-to-dateness of installed firmware versions of Internet of Things devices accessible via public Internet. It takes a novel approach to identify versions based on the source code of their web interfaces. It analyzes data sets of 1.06m devices collected using the IoT search engine *Censys* and then maps the results against the latest version each manufacturer offers. A fully scalable and adaptive approach is developed by applying the SEMMA data mining process. This approach relies on three data artifacts: raw data from *Censys*, a mapping table with firmware versions, and a keyword search list. The results confirm the heterogeneity of connected IoT devices and show that only 2.45 percent of the IoT devices "in the wild" run the latest available firmware. Installed versions are 19.2 months old on average. This real-world evidence suggests that the updating processes and methods used by engineers so far are not sufficient to keep IoT devices up-to-date. This paper identifies and quantifies influencing factors and captures the global and diverse distribution of IoT devices. It finds manufacturer and device type influence the up-to-dateness of firmware, whereas the country in which the device is deployed is less significant.

Index Terms—Internet of Things, IoT, embedded systems, firmware, version, patch, update, up-to-dateness, fingerprinting

1 Introduction

Today, Internet of Things (IoT) devices can be found in almost every area of life. Notwithstanding the benefits these devices bring to society and economy, they come with a plethora of security challenges. As most manufacturers prioritize rapid development over comprehensive security, the device firmware (FW) tends to be plagued by vulnerabilities [1], [2]. Criminals can exploit them, for example, to hijack the device for increasingly powerful botnets and distributed denial-of-service (DDoS) attacks [3].

Updates and patches (U&P) are an important and often the only way to fix FW vulnerabilities [2], [4]. While software patching is a common practice for computers and mobile devices, it is much less common for IoT devices, since it is comparatively complex to implement [3], [5].

While much software engineering (SE) research has been conducted on the continuous updating and patching of computer systems [6], such research into IoT devices is still underrepresented [7], even though IoT devices vastly outnumber other computer systems. It is therefore important that the SE domain contributes to improving the up-to-dateness of such devices. This publication presents a snapshot of the updating landscape in the wild, and brings a first understanding how SE can help to keep IoT devices up-to-date.

• *The author is with the Fraunhofer Institute for Systems and Innovation Research ISI, 76139 Karlsruhe, Germany. E-mail: frank.ebbers@isi.fraunhofer.de.*

Manuscript received 18 June 2021; revised 7 February 2022; accepted 28 March 2022. Date of publication 31 March 2022; date of current version 13 February 2023.
This work was supported by the SPARTA project which has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No 830892.
Recommended for acceptance by P. Pelliccione.
Digital Object Identifier no. 10.1109/TSE.2022.3163969

Researchers attribute the potential to protect devices to both users and manufacturers, and thus to software engineers [1], [8]. However, many manufacturers do not provide patches in a timely manner [9]. While IoT standards for providing FW updates exist [10], [11], these do not appear very institutionalized. Manufacturers prioritize fast time-to-market over implementing maintenance features. Further, these standards do not add to user experience.

For this reason, users tend to avoid installing FW patches even if these are available [12]. Detailed studies find that very few users perform FW U&P on IoT devices [13].

This highly unsatisfactory state of affairs has caught the attention of policymakers. An EU consumer protection directive came into force in 2022, which entitles IoT users (amongst others) to receive updates within a reasonable timeframe [14]. However, public policy efforts cannot keep pace with technical cybersecurity advances. This implies "a need for research that explicitly identifies the implicit values of IoT users, which can then be used to inform the policy development process" [15, p. 2]. From a strategic perspective, both regulators and software engineers could benefit from insights into FW distribution and the up-to-dateness of IoT devices "in the wild" in order to identify the factors hindering the dissemination of U&P. In addition, cybersecurity agencies and Internet service providers (ISP) could benefit, as they could identify how many unpatched devices are connected to their Internet exchange point, and thus anticipate possible impacts of DDoS attacks.

Although there are numerous studies analyzing IoT FW, all of them focus either on detecting vulnerabilities (see [16]) or on secure update processes (see [17]). None investigates the outcome (the actual up-to-dateness of devices in the wild) of updating processes. To do so, our study is based on data collected by the search engine Censys [18]. While most previous studies identify devices based on their

Internet traffic (see [19]), we analyzed the source code of the devices' web interface to identify the installed FW version.

We set out to determine the indicators of deprecated FW on IoT devices and, in particular, the extent to which IoT manufacturers or end users are responsible for this. Thus we contribute to [20, p. 706], who suggest "additional factors that influence the patch deployment process." To this end, our study investigated four specific questions:

- Which types of IoT device are prevalent "in the wild"?
- How up-to-date are these IoT devices "in the wild" with regard to their FW version?
- Does the patching level of IoT devices differ by device type and geography?
- Are U&P for devices provided frequently by the manufacturers and do users install them in a timely manner?

The remainder of this paper is structured as follows: first, we describe the need for FW U&P and the challenges for their deployment. We then explain our methodology and present the results, which we discuss in a subsequent chapter. We conclude our work with suggestions for improvement and outline further research.

2 Related Works

2.1 The Need for Updates and Patches of Internet of Things Devices

While security is considered a major challenge within the IoT, research in this area is still in its infancy [7], [13]. Various authors agree that IoT devices present easy targets for remote attacks (see [5], [21], [22], [23]). A 2018 study showed that "70 percent of the devices connected to the Internet are vulnerable to numerous attacks" [2, p. 1636]. These include children's toys or home medical devices [24], with some even disclosing the owner's home address [25]. Further, [26] found 40 percent of malicious devices are installed in critical infrastructure.

Most of these devices have very limited hardware capabilities to resist security threats [27]. This makes it easy for hackers to exploit known security vulnerabilities, intrude in even very sensitive areas (e.g., users' homes) and infect millions of connected devices at once to perform DDoS attacks [3], [21].

According to [21], the absence of control interfaces means attacks often go unnoticed, and poorly updated and patched devices turn zero-day into eternal vulnerabilities. The longevity of IoT devices exacerbates this effect. Typically, the FW of IoT devices is not a well-tested product, so security vulnerabilities typically exist in deployed code [2], [22]. From a technical perspective, vulnerabilities in IoT FW are mainly caused by input-independent errors (e.g., memory exhaustion) rather than input-dependent errors (e.g., invalid parameters) as is the case in PCs. [28] examines the firmware of 10 routers and IoT devices and finds 109 errors in total.

To fix these vulnerabilities in traditional computer systems, U&P are a common and important remedy [2], [12], since U&P tend to have a significant positive impact on IT system security [29]. That "[p]atching is necessary for security, but [. . .] difficult to manage systematically" [30, p. 49] is still very true today. For example, a survey of 3000 IT professionals worldwide found that 60 percent of data breaches could have been prevented by patching [31]. However, another survey found that 50 percent of the companies were unable to patch their systems within 72 hours, and an additional 15 percent were unable to do so even after 30 days [32].

The situation is quite similar in the field of IoT, where in many cases U&P are the only option to secure a device [4], [33]. A survey among 109 IoT manufacturers found that 50 percent see remote deploying of U&P as a key challenge [34]. Further, [35] found that 23 percent of 259 surveyed companies fear that IoT devices are not patchable. The importance of U&P has since attracted the interest of policymakers in the EU and US. In 2019, the European Parliament passed a consumer protection directive entitling IoT users (amongst others) to receive updates within a reasonable timeframe [14]. In 2020 the IoT Cybersecurity Improvement Act required federal government to improve the security of their used devices.

2.2 Software Engineering for Internet of Things' Patches and Updates

IoT security directly translates into several technical and non-technical domains of SE, with several chapters of the SWEBOK concerning U&P [36]. Patching personal computers (PC) and smartphones is a well understood process and has been researched from a user as well as a manufacturer perspective [3], [27]. For PCs, studies show that automatic updates greatly increase installation rates [12], but "patch management is the leading security challenge in the emerging IoT" [37, p. 41]. This challenge is due to both technical and socio-technical factors [6], for instance, that many IoT devices cannot be accessed physically by humans to conduct an update [38, p. 684].

In SE research, patching is often understood as traditional software maintenance and as an "essential feature [. . .] for bringing new functionality, or correcting discovered bugs" [39, p. 435]. However, patching is a not a trivial process [39]. Software engineers have developed a wide range of possible solutions and proof-of-concepts to improve IoT patching. These included dynamic patching [40], [41], incremental updates [39], firmware over-the-air (FOTA) [5], [27] and standardization [24]. From a technical perspective, typical firmware update procedures are fairly simple and adaptable to different devices. These include full firmware image updates using bootloaders, partial and incremental updates via dynamic or differential patching, or using scripts [11].

In practice, however, the situation is different, because an IoT device is highly dependent on Internet services, the technological and organizational infrastructure, and cannot be regarded as a standalone product [42]. Thus patch management is a "collaborative effort between multiple stakeholders" [6, p. 15]. FOTA updates are difficult to implement in IoT devices due to their limited and varying storage and processing power [5], [27]. "[L]ow downlink data rate, very short duty cycle and lack of firmware integrity verification" make it hard to implement sufficient FW update

mechanisms [2, p. 1667]. The need to take a device offline to install an update inhibits proper management [32]. However, most devices need to be shut down in order to perform a software or firmware update in order to allow a bootloader to load a new full FW image [11], [43]. Many devices are often in sleep mode, where memory and CPU intensive tasks, i.e., FW updates, are hardly possible [38].

In addition, low-cost manufacturers with proprietary software [7] provide very limited access to their systems and do not implement standards such as those of the Internet Engineering Task Force working group "Software Updates for Internet of Things" (SUIT) [10]. Nonetheless, [11, p. 71907] calls engineers to "use a standardized firmware update mechanism rather than having to design their own". Despite the myriad of solutions, a benchmark is still missing, as is an understanding of their effects. Such information could be interesting for SE.

Developers also focus too little on interfaces and methods to notify end users about new versions [37], [44]. In addition, there are "[d]ifferences between products with a low and high perceived security risk" [45, p. 440]. Products with a high perceived security risk include smart alarm systems or door locks. In the study by [45], users' buying preferences for these types of product without a guarantee for security updates was 6.6 percent. If a 6-year guarantee was included, this value increased to 53.4 percent [45]. Furthermore, messaging the importance of an update can influence the intention of updating [46], [47]. We therefore hypothesize that *there could be marked differences in the up-to-dateness of the FW for different IoT device types (H1).*

The processes and frequency for distributing patches varies between vendors. [3] and [24] find that some manufacturers assume a higher degree of responsibility for infected devices than others. Open-source developers, for example, provide updates within four weeks on average [48]. Further, "manufacturers certainly bear a large responsibility to 'bake in' the security as much as possible" [29, p. 519]. As many manufacturers consider U&P a key challenge [34], we hypothesize that *the installed base of latest FW versions differs between manufacturers, regardless of the device type (H2).*

Other research finds that manufacturers' "patch release behavior is an under investigated component of overall software quality and security" [49, p. 116]. Several studies, e.g., [9], find that there are significant delays between vulnerability disclosure and patch release. One good example is Android OS, where there are significant differences in the U&P frequency between phone manufacturers [50]. We transfer this observation to the field of IoT, and hypothesize that *there are IoT device manufacturers who do not provide regular updates or patches (e.g., in annual intervals) for devices that are still in use and online (H3).*

As users can also "take an active lead toward protecting their devices" [1, p. 21], we consider them another influencing factor. The "tragedy of the commons", a situation where one individual's behavior has an impact on many, is valid in cybersecurity as well. [12] observes that users often hesitate to install U&P due to "1) unanticipated user interface changes, 2) unused and unrecognized software, and 3) liking the current software" [12, p. 3216]. Further, users' individual benefits do not outweigh the installation costs [8]. [13] highlights that users do not feel responsible and [51, p. 3] finds that users "often choose to take on the consequences in favor of using the devices based on the utility they provide". After the Mirai botnet attack, vendors produced several patches. However most of them were never installed by end users [33]. A study of 2000 IoT owners found that around 40 percent never perform FW updates and a further 10 percent do not even know what FW is [52]. [20] finds differences between user groups regarding updating computer programs. Users tend to delay software updates for 80 days after they have been released by the manufacturer [53]. Thus, we hypothesize that *even if FW updates are available, many users do not install them within two months after release (H4).* For H4, we do not take into account other influencing factors, such as missing update notifications [21].

2.3 Firmware and Software Version Share

Several studies examine the distribution of different versions of web browsers or mobile and PC operating systems [54]. A recent study found that 55 percent of all installed PC programs worldwide are out-of-date [55]. Research efforts attempt to identify the FW and software versions of web servers by employing dedicated tools such as WhatWeb [56]. [20, p. 706] investigates the patch dissemination for popular software such as Adobe Flash, Reader and Firefox, and finds that "only 28 percent of the patches [. . .] reach 95 percent of the vulnerable hosts", even after a 5-year period.

Such statistics are scarce for IoT devices, due to their great variety [3]. [57, p. 2] states that "security updates are often rarely applied on a timely basis", but does not mention the reasons for this. IoTTracker by [58] scanned for specific open-source FW for routers, classified by vendor and country, but provided no insights into the installed FW version. For a specific printer, [59] found that only 1.08 percent run the latest FW. [60] analyzes IP cameras and found several running FW dated 2015. [37] found that a vast majority of Bluetooth devices implement a protocol that was outdated in 2013, and IoTInspector shows devices' outdated TLS versions [24]. However, to the best of our knowledge, so far, there has been no comprehensive study on the version share of IoT devices.

According to [61, p. 1169] "[d]evice types and manufacturer popularity vary dramatically across regions". IT security behavior also differs between cultures [62], [63], as demonstrated, for instance, by how frequently users install updates on PCs [53]. Also, there are significant regional differences in compromised IoT devices [1]. Our final hypothesis therefore states that *there are regional differences in the up-to-dateness of installed FW versions of IoT devices, which are independent of the device type and manufacturer (H5).*

2.4 IoT Device Discovery and Annotation

Several authors have researched discovering (malicious) IoT devices, e.g., through web crawling, natural language processing, data mining and traffic analysis [19], [58], [64]. The data sets also vary. For example, [61] uses proprietary data from an antivirus company. [19] investigates unused IP addresses and examine devices' traffic. [3] and [65] focus on devices compromised by the Mirai botnet.

It is crucial to identify the device type, manufacturer and model name to protect the IoT effectively [24], [64]. This is exemplified by [58], who found that devices produced by the same vendor or in the same series usually come with similar security vulnerabilities. In addition, [61, p. 1170] found that "90 percent of devices worldwide are produced by only 100 vendors". Thus, once a vulnerability is found in a manufacturer's firmware, the total number of potentially infected devices is huge. [65] finds that almost half of all Mirai infected devices worldwide are produced by only 9 manufacturers.

However, there are no uniform patterns or methodologies to gather such information on a large scale [1], [3], so researchers use different approaches. Some rely on users to label their device type and manufacturer [24], [61]. However, only 7.1 percent of the devices are labeled. Others rely on service banner data, DNS behavior or packet traces [1], [66], [67], but inferring device identities using such signals requires many very precisely predefined rules [24], and packet inspection generates a lot of traffic and overheads [67]. [68] and [69] use machine learning to annotate devices in a lab setting. However, it is unclear "whether the models would be equally effective if tested in real-world settings" [24, p. 4]. Deducing the manufacturer from a device's MAC address is possible, but less successful [24]. Finally, using the domain information a device corresponds with will not be suitable in future because of domain encryption. [24].

None of these approaches is suitable to annotate devices and extract FW information deployed "in the wild". Lab studies provide a full set of data, but are limited to a microscopic level, focusing on specific devices or contexts [19] or an organization's network [68]. Lastly, device traffic, i.e., IP and TCP headers do not include FW information. There are search engines that specialize in IoT devices. Two prominent ones are *Censys* [18] and *Shodan* [70]. Both scan IP addresses, different ports and protocols and grab banner data as well as other Meta data, e.g., the device's location. *Shodan* was released in 2009 and scans the IPv4 and IPv6 space. Its web crawlers generate a random IP address and test a random port for accessibility. If successful, the banner information is grabbed. This strategy ensures relatively uniform coverage. To limit traffic, increase search speed and avoid geographical bias, the crawlers are located in different countries [71]. *Shodan's* database supports queries with prefilters (e.g., constraining a search to a particular city). *Censys*, which was established in 2015, works in a similar way to *Shodan*, but its scans are limited to the IPv4 address space. Unlike *Shodan*, it is open source and can be used freely for academic purposes. The infrastructure and storage is provided by Google. *Censys* pings four billion devices each day. Data are aggregated on a daily basis and accessible via Google's BigQuery data warehouse, allowing historical searches that go back to 2015. Similar to *Shodan*, the data sets include the device type, manufacturer and device location. *Censys* provides two different data tables: an aggregated data table "banner" and a detailed "public" table.

2.5 Research Question and Hypotheses

Although SE have developed a myriad of different U&P solutions, it is generally accepted that IoT devices are a security risk because of missing updates and patches. To the best of our knowledge, however, there is no paper quantifying the problem and investigating the possible reasons and influencing factors for this observation using real-world data.

Our research question concerns *how up-to-date IoT devices in the wild are with regard to their installed FW version* and we aim to provide comprehensive qualitative data to answer this. We also investigate whether there is a difference between device types, the country in which the device is deployed and whether users or manufacturers are slowing FW distribution.

Our literature review suggests that studies of the frequency and up-to-dateness of FW versions are scarce and only cover specific devices, such as printers [59]. We did not find any comprehensive studies of the whole variety of IoT devices. A better understanding of the influencing factors and responsibilities for IoT FW U&P could help SE to identify factors hindering the dissemination of U&P in practice and thus help manufacturers improve their update strategy (e.g., mechanisms for nudging users to install available patches) and could guide policymakers in positioning regulatory levers. We address this gap with this large-scale study of IoT devices and their respective FW version. Our working hypotheses are:

H1: There could be marked differences in the up-to-dateness of the FW for different IoT device types.

H2: The installed base of latest FW versions differs between manufacturers, regardless of the device type.

H3: There are IoT device manufacturers who do not provide regular updates and patches (e.g., in annual intervals) for devices that are still in use and online.

H4: Even if FW updates are available, many users do not install them within two months after their release.

H5: There are regional differences in the up-to-dateness of installed FW versions of IoT devices, which are independent of device type and manufacturer.

3 Methodology

Our data mining approach (Fig. 2) followed the sequence of steps in the SEMMA process developed by the SAS Institute [72]. It has the advantage of being iterative, applicable across a variety of industries and offering methodologies for diverse business problems. We employed the storage and service facilities of the Google Cloud services, in particular, BigQuery. Our analysis is based upon three data artifacts: a list of keywords and regular expressions (regexp) to filter the data sets, real-world data from *Censys and Shodan*, and a mapping table containing device models and their corresponding firmware versions and dates as found on the manufacturers' websites.

Step 1: Sample Generation, Data Set and Selection of Search Terms

Although not specified in the SEMMA process, we first generated a data basis. We contacted top IoT manufacturers (Amazon, Bosch, Cisco, Google, HP, and Siemens). However, similar to [19], none were willing to provide information about the up-to-dateness of their devices. We therefore focused on the IoT search engines *Shodan* and *Censys* to ensure non-biased and good quality data. These are suitable

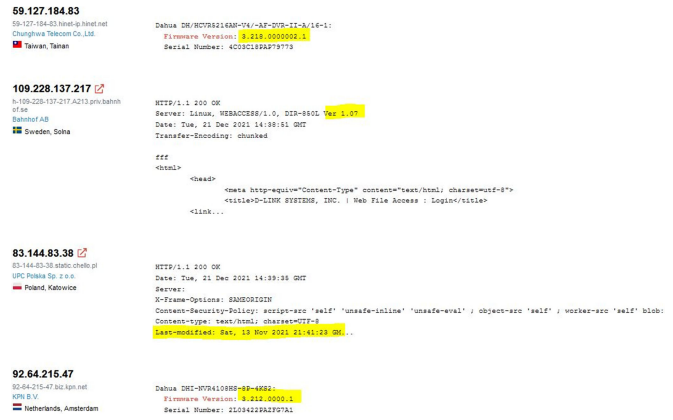

Fig. 1. Part of the search results on Shodan for keyword "firmware version" as of December 2021. Relevant information about the up-to-dateness highlighted.

for our analysis, as both databases represent the current real IoT distribution in the wild.

In order to create a list of suitable search terms that reveal information about the FW version, we applied an iterative process. We used *Shodan* for a first insight into possible keywords and devices, as it provides a simple search interface through the website (Fig. 1). A "small business" subscription was needed to download all the results of our search query "firmware version". We downloaded the 17k device data sets for this search, including banner data, the HTML source code of the Web-UI or telnet login screens. We conducted a keyword analysis of the source codes, using the Top-K Sequential Patterns algorithm by [73]. This yielded multiple keywords and HTML tags carrying information about installed FW versions (e.g., class = "fw-version", class = "fwv" or id = "fw_ver"), which we added to our list of search keywords. We also identified IoT manufacturers and device models known to have been hijacked in the past or widely accessible via public Internet [1], [3], [58]. These were also added to the list of search keywords. The final step was to translate the keywords into regexp, if applicable. We then used *Censys*, as it supports searches for regexp (which *Shodan* does not). Data are hosted in the Google Cloud, accessible via Google BigQuery. We carried out database searches in mid-April 2020 across the public and the banner data tables, which include the fields shown in Table 1.

Search terms were derived from the keyword list generated in the prior step, including some 80 regexp to infer the installed FW version (e.g., "FW ver. /d∗./d∗") or a FW date (e.g., "build date \d{4}\.\d{12}\.\d{12}"). This resulted in a total of 292541461 devices. Our hypotheses meant we were only interested in devices providing information about the device type, model, manufacturer or country. Of these 292m devices, 1.06m (0.31 percent) revealed information for at least one of the above-mentioned factors. These served as the base data set used in the subsequent processing steps described below. Duplicate devices were removed by calculating a hash value over all data fields.

We calculated the statistically representative sample size for N = 1.06m using standard sample calculation: 16384 devices at a confidence level of 0.99, with confidence interval = 0.1, and created it using the RAND-function in Google BigQuery.

Step 2: Explore - Initial Characterization of the Sample Data Set

In view of our working hypotheses, we first analyzed the manufacturer and device type fields of our sample data set (n = 16384). We found a small number of manufacturers to be strongly overrepresented (e.g., MikroTik, ZTE, D-Link) and that the majority of the devices are Internet routers. The manufacturer and device type field contained 68 percent null values, indicating that further analysis was needed to identify manufacturers and device types.

Most devices were installed in Venezuela (6.3 percent), Brazil (4.3 percent) and Russia (4.1 percent). The country field contained null values in 42 percent of all cases. The data fields "revision" and "version" did not contain any suitable information. Some five percent of the sample values did not contain any valid information at all.

Step 3: Modify - Adjustments to the Data Set

We introduced a number of modifications to the data fields as a next step. As we found that processing in BigQuery is very fast, we applied all modifications to the full data set of 1.06m devices.

We first created a unique identifier for each entry by calculating a cumulative hash value across all fields. We then focused on missing field values in *country*, *manufacturer*, *device model* and *type*. For example, we decided to use the *country-code* field instead of *country*, as this displayed only 7.1 percent null values. Next, we scanned the source code for manufacturer names and models using regexp, since the use of regexp had been successfully applied by [61] and [1]. Further, we examined the "metadata.description" field (see Table 1) and extracted the manufacturer, if applicable. In addition, we searched the source code for model names and derived the manufacturer based on the entries in our mapping table.

Again, we applied the Top-K Sequential Patterns algorithm to find additional patterns in the device source code. This allowed us to extend the search keyword list and identify device models. To amend missing values on manufacturer, device model and type, we consulted [58], whose classification tool helped to successfully categorize an additional 217836 devices.

In order to extract the FW version, we used our regexp and extracted patterns resembling FW version numbers, such as "(firmware ver\.∗\s∗\d+(\.∗\d∗[a-z]∗)+)". In a similar fashion, we developed patterns representing FW deployment dates in the source code, such as "(built date \s∗.{110}\s∗\d{4}\.\d{12}\.\d{12})".

We created a mapping table by manually searching for device models with their current as well as former FW versions, and their respective deployment dates from each manufacturer's website. We extracted this data manually, as automatic web crawling was not possible due to different website structures. We standardized vendor and model names and removed inconsistencies following a similar approach as [24]. Fig. 3 illustrates this procedure for two devices.

Step 4: Model - Mapping Devices with FW Dates

The final artifact of our SEMMA process was a table containing the following information:

- device type,
- device model,
- manufacturer,

TABLE 1
Description of the Fields Queried From the Censys Data Set

Field	Description
ip	ip address as a string
services.banner	source code of the web UI or telnet login screen
ipv4_public.country_code	ISO country code in public data set
ipv4_banners_public.country_code	ISO country code in banner data set
metadata.device_type	device type
metadata.product	product model
metadata.manufacturer	manufacturer
metadata.description	mostly including the manufacturer name and device series names (e.g., "Huawei Home Gateway")
metadata.revision	carrying a number
metadata.version	carrying a number
metadata.os	name of the OS, e.g., RouterOS or Ubuntu
metadata.os_version	version of the OS as a number
updated_at	Timestamp of latest update

all metdadata fields are derived by Censys

- installed FW version as a number (iFV) and
- installed FW date (iFD) as date derived from the raw data or extracted from the source code. It also included calculated data and data derived from the manufacturers' websites:
- latest FW version (lFV) found on the manufacturers' websites for each device model,
- latest FW date (lFD) as the release date of the corresponding version,
- age of the installed FW (AiF) as of April 2020,
- age latest FW (AlF) as of April 2020 and
- time between iFW and lFW (TB), which represents the main driver of the up-to-dateness.

As not all the information was provided for every device, we applied the rules presented in Table 2.

Step 5: Assess - Reliability

Usually a model's reliability is tested by applying it to a different set of data. However, IoT device deployment and

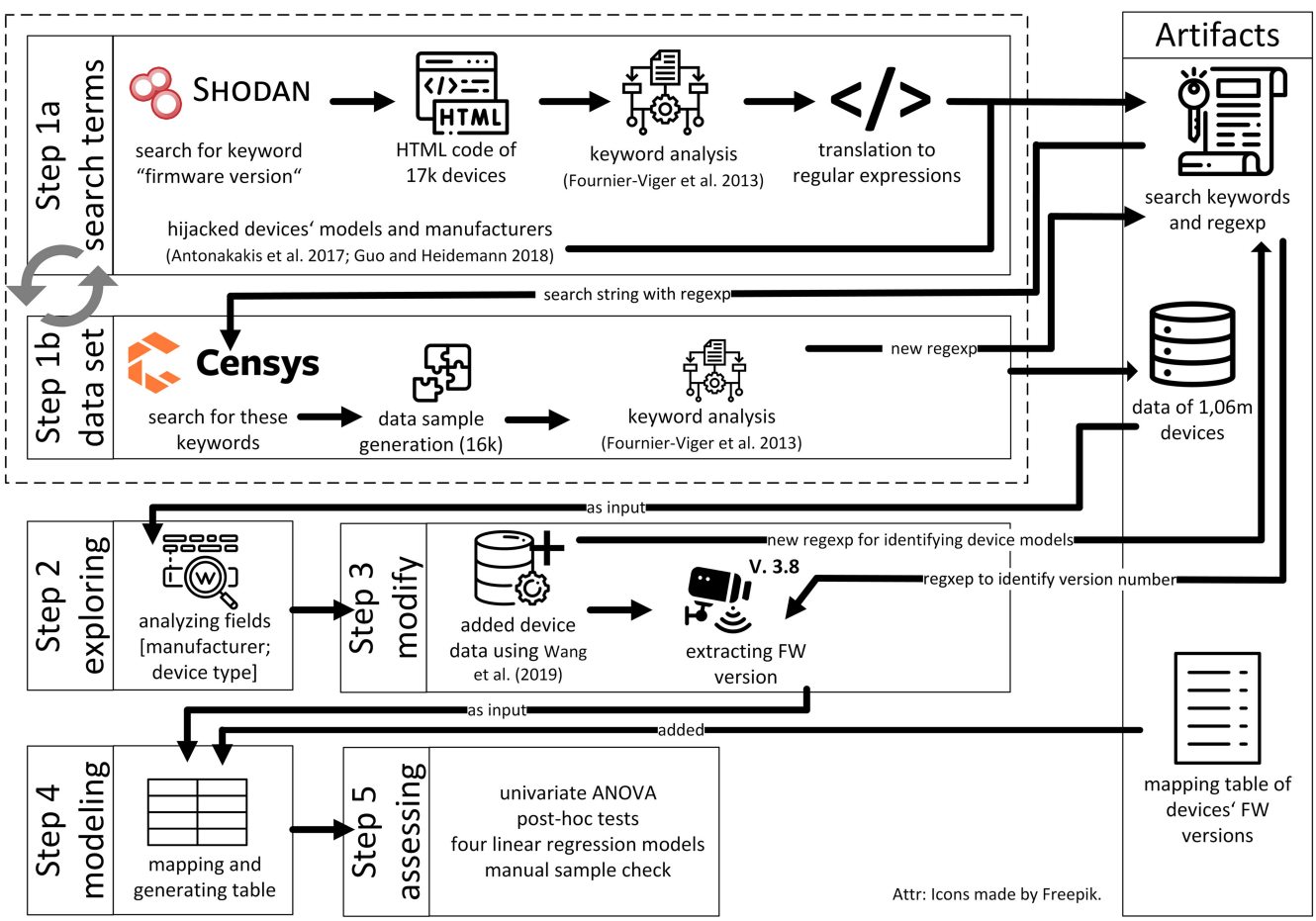

Fig. 2. Methodology.

Manufacturer_Export	Model_Export
/Dahua onvif://	ware/DHI-HCVR51
TP-Link	/TP-LINKTL-SC32

Device Type	Manufacturer	Model	FW version latest	FW date latest	FW version older	FW date older
Network video recorder	Dahua	DHI-HCVR51	V3.200.0001.30.R.20170807	22.09.2017		
Network cam	TP-Link	TL-SC3230	v1.140731	31.07.2014	v1.140401	01.04.2014

Fig. 3. Results of the regexp (top) and standardized mapping table with FW version information extracted from manufacturer website (bottom).

identification is highly volatile due to dynamic IP allocation [74]. Thus, it is impossible to examine the same set of IoT devices at another time to make a statement about AiF, for example. Therefore, we focused on the robustness of the results. We did so by conducting a univariate ANOVA to identify the influence of the independent variables (type, manufacturer, country), followed by a post-hoc test to identify influences within each variable. In addition, we conducted four linear regression models to validate the in-between group reliability and to control for confounding variables. As a final step, we checked a sample of the final data set manually to make sure device characteristics were set correctly. We did not find any allocation errors, but we did find that the version of the router device "H108N V2.5" corresponds to its model name and not to its firmware. We therefore excluded these devices from the analysis.

4 Results

4.1 Statistics for Full Sample

In total, 292541461 devices fit our regexp, of which 1061284 (0.31 percent) provide information about at least one of the following: device type, model, manufacturer or country. The results include 113 distinct models (e.g., "DIR-860L") from 63 distinct manufacturers, which can be assigned to 14 device types. More than half of all the devices found can be assigned to MikroTik, a manufacturer of routers (Fig. 4). No manufacturer was found for only a very small proportion of 0.46 percent. Most of those devices were "network device" (30.42 percent) or routers (27.94 percent), but roughly 15 percent were devices for collaboration and messaging. For printers, for example, only a very small proportion of 0.04 percent reveal such information (Fig. 5). However, slightly more than one quarter all all devices could not be assigned to a device type.

As Fig. 6 shows, most devices are installed in Venezuela (11.3 percent), followed by Brazil (8.3 percent) and Russia (5.0 percent). No country information (n/a) is provided for 7.3 percent.

TABLE 2
Modeling Rules

Data available:				Modeling rule:
iFV	iFD	lFV	lFD	
x	x	x	x	lFD - iFD = TB [months], if lFD = iFD: firmware is up-to-date
	x			Today - iFD = AiF [months]
		x	x	Today - lFD = AlF [months]
x				No calculation possible

4.2 Mapping Table

Our mapping table consists of 1899 device FW versions or dates for 401 distinct devices, of which 296 contain FW versions and dates. This includes current and former versions with their deployment dates. There are more distinct models in the mapping data set, because it includes all the available models of a device series. Using this mapping table, we were able to calculate the age of the most recent firmware available for a specific device (date difference between April 2020 and the deployment date of the FW image). This was successful for 99.4 percent devices in our database.

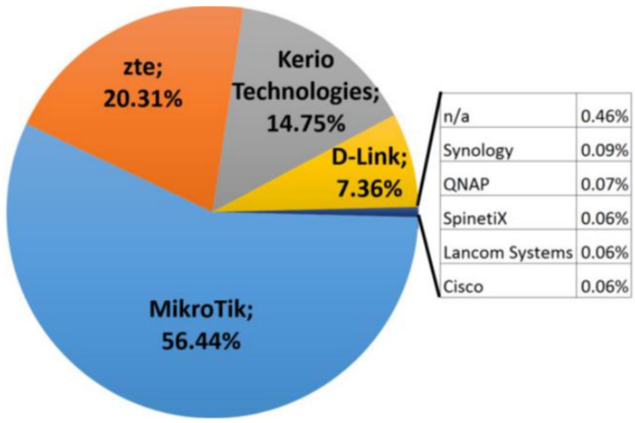

Fig. 4. Percentage of top 10 manufacturers.

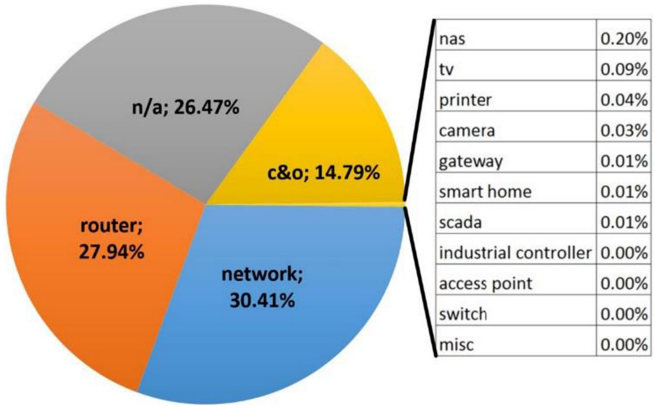

Fig. 5. Percentage of different device types (c&o = collaboration and messaging).

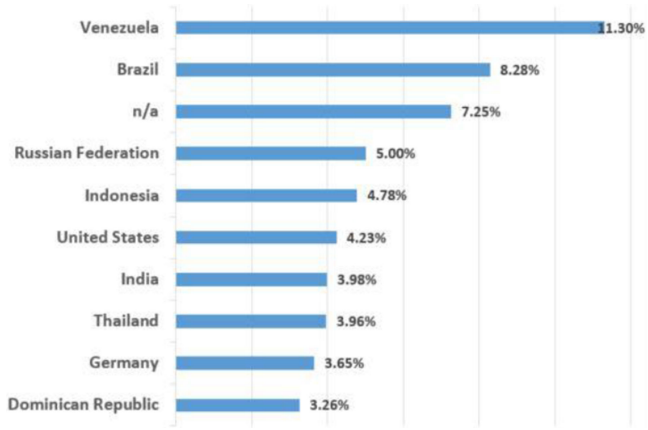

Fig. 6. Percentage of top 10 countries with most deployed devices.

Reading the **AIF** in the mapping table, the results show that the manufacturers Brother (43.5 months) and HP (37.2 months) offer U&P very infrequently, whereas MikroTik offers them very frequently (<1 month). The FW of smart home devices is the oldest (60.3 months), while manufacturers keep network devices most up-to-date (0.004 months). The **TB** varies between 0 and 102 months, with an average time of 18.88 months ($\sigma = 21.32$).

4.3 Statistics for Devices With Information About Firmware Version and Dates

Our approach extracted a FW version for 1055083 devices (again 99.4 percent), with a majority of devices from MikroTik (57 percent) and ZTE (20 percent). Most were "network device" (48.41 percent) or routers (33.20 percent), but roughly 19 percent were smart TVs. Only 6201 (0.58 percent) devices did not reveal any FW version information, despite fitting our regexp.

Mapping the firmware versions extracted from the web UI with the versions found in our mapping table resulted in 460773 devices (43 percent) in 12 device categories, with a majority for network devices (53 percent). 99 percent of these devices are produced by MikroTik.

The results show that the average age of the installed FW version (AiF) is 19.2 months, as of April 2020. Smart Home devices have the highest average age (77.0 months), whereas access points are relatively up-to-date (11.0 months) (Fig. 7). 226785 (49 percent) devices run the latest available FW version. Fig. 8 shows that Hikvision devices have the highest average age (96 months), while Algo and Cisco devices are relatively up-to-date. However, calculating an average age was only possible for 13 out of 63 manufacturers (21 percent), as the others did not provide any FW date information for their devices. Considering differences between countries, devices in the Bahamas are rarely updated (avg. 64.38 months), while devices in Haiti and Liechtenstein are updated every two months on average (Fig. 9). However, these countries only have a very small proportion of devices (Bahamas 0.013 percent, Liechtenstein 0.005 percent, Haiti 0.0004 percent). Egypt, on the other hand, accounts for 1.3 percent of all devices and is therefore more representative.

Descriptive statistics and a univariate ANOVA show the effect sizes and significance visualized in Fig. 10.

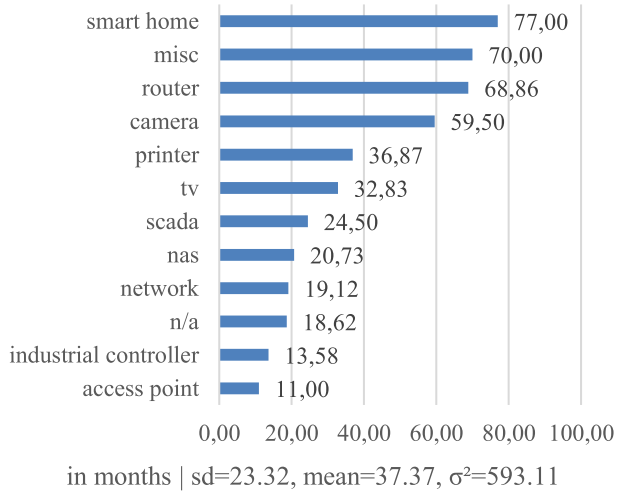

Fig. 7. Average age of FW versions, grouped by device type (switch, gateway and collaboration are missing due to lack of FW deployment dates).

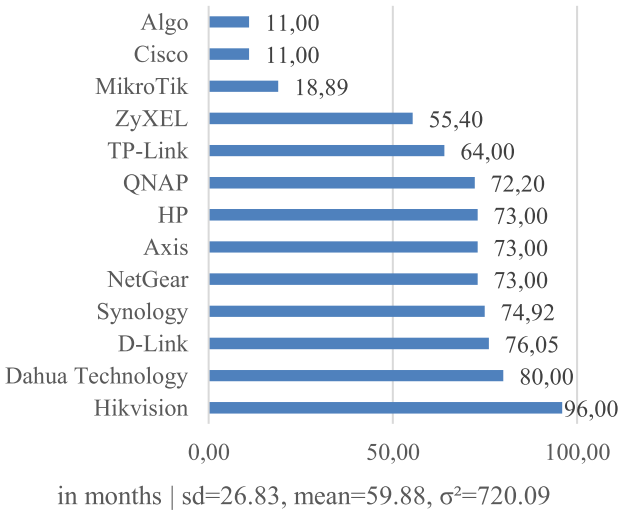

Fig. 8. Average age of FW versions, grouped by manufacturer.

Manufacturer ($F = 13.38$, $p<0.001$) seems to have the strongest significant influence on the FW age, and supports H2. However, it is likely that effects are not estimated correctly, as the number of devices within groups varies dramatically (e.g., 57 percent are MikroTik devices, whereas 30 manufacturers provide only 0.01 percent of all devices). Country ($F = 5.33$) is significant at a $p<0.05$ and device type ($F = 2.12$) at a $p<0.001$, which leads us to accept H1 and H5.

To gain more insights within groups, two post-hoc tests (Bonferroni and Tukey) were conducted. For the device type, 48 percent (Tukey) and 45 percent (Bonferroni) of the intra-group comparisons are significant at $p<0.001$. For example, the Tukey post-hoc analysis revealed a significant age difference ($p<0.001$) between routers and industrial controllers (55.30, 95 percent-CI[42.39, 68.20]. There are manufacturer intra-group differences in 51 percent of the cases. The results show that, e.g., MikroTik devices are more up-to-date than D-Link, Hikvision, HP, QNAP, Synology, ZyXEL at $p<0.001$. Only Algo devices from are more up-to-date, although this group comparison is not significant (Table 3, significant results with $p<0.01$ are shaded

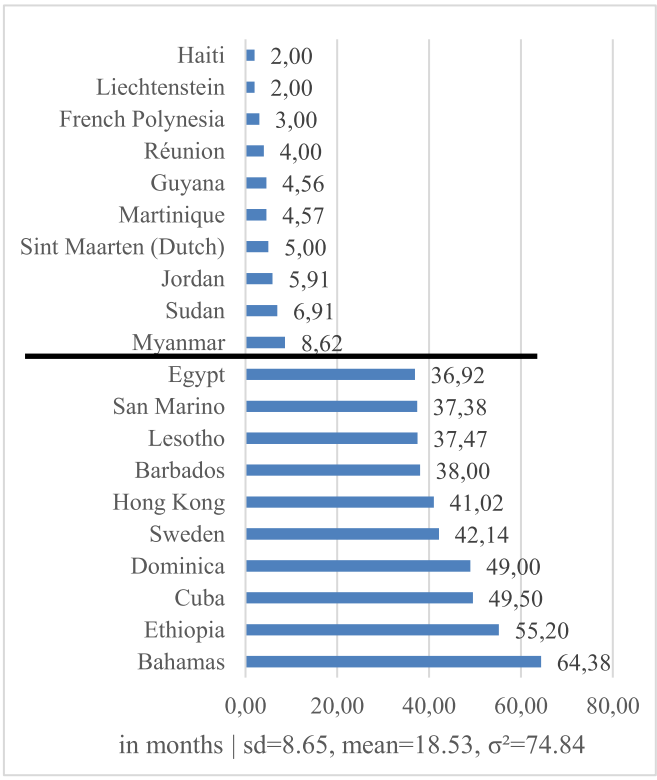

Fig. 9. Top and bottom ten countries with up-to-date IoT devices.

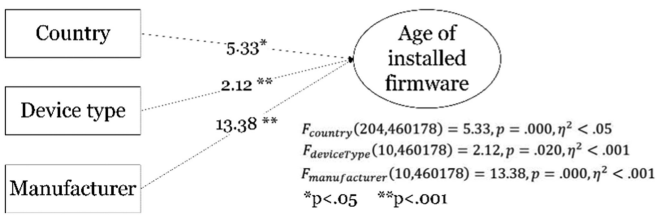

Fig. 10. Results of univariate ANOVA.

gray). The country in which the device is located seems to have less influence on the FW age, as both post-hoc tests are significant in only 21 percent of cases. These results support H1, but not H5.

In addition, a linear regression (LR) was conducted, as ANOVA is not robust in all cases, due to missing confounding variable control and its assumption of a normal distribution [75]. Four LR were conducted with age as the dependent variable and (1) device type, (2) manufacturer, (3) country and (4) all three as independent variables.

TABLE 3
Tukey Post-Hoc Test for Intra-Group Comparison With Manufacturer MikroTik

Mikrotik	Mean Difference	sd	Sig.
Algo	7.89	5.745	0.870
D-Link	-57,16*	0.410	0.000
Hikvision	-77,11*	5.178	0.000
HP	-54,11*	11.959	0.000
QNAP	-53,31*	9.263	0.000
Synology	-56,03*	5.745	0.000
ZyXEL	-36,51*	9.263	0.002

TABLE 4
Model Summary of Linear Regression

Model	R	R^2	Std. Error
(1) Device type	,184	,034	21.378
(2) Manufacturer	,198	,039	21.321
(3) Country	,222	,049	21.210
(4) = (1)&(2)&(3)	,290	,084	20823

For (1), the results show for $R^2 = 0.034$ (Table 4) that the device types camera, printer, and router ($p<0.001$), and smart home ($p<0.01$) all have a significant positive influence on the age of the installed FW (Table 5, gray shading indicates a significance of $p<0.01$). This means that the large number of months is due to these device types, which thus means that they are updated the least often.

Of these device types, routers (standardized Beta = 0.182) contribute the most. In contrast, access points and industrial controllers show a negative std. Beta, which means that devices of this type are frequently updated. However, this observation is not significant (sig = 0.171 in Table 5). The same applies to SCADA devices (sig of 0.314). Again, these findings support H1.

For (2), D-Link, HP, Hikvision, QNAP, Synology and ZyXEL have a significant influence on age ($p<0.001$, Table 5). However, only the influence of D-Link devices is strong compared to other manufacturers (std. Beta = 0.196, Std. err = 0.422).

For (3), 84 out of the 204 countries are significant at a $p<0.001$, and 94 at $p<0.01$. Table 6 shows a selection of countries. However, the strength of the correlation (standardized beta) is very low. Only eight countries have a std. Beta >0.05, including devices deployed in China and India (Table 6).

For (4), R^2 more than doubles (Table 4), which indicates that considering all three variables at the same time doubles the explanatory power for age variance.

5 Discussion and Contribution

Missing and uninstalled firmware of IoT devices is a serious security vulnerability [1], [4]. One current example is "Amnesia:33", which affects several IP cameras, sensors and smart home devices present in our results [76].

Despite a myriad of solutions developed by SE to improve U&P mechanisms, there is currently no analysis available whether and to what extend these solutions are used in practice.

Despite the calls already made in 2016 for an "efficient mechanism for the distribution and installation of updates" [77, p. 171] for IoT devices, our results reveal that outdated IoT devices are common around the globe and underline the severity of patch and update delays, leading to millions of vulnerable devices. We contribute to the ongoing discussion about the responsibility for outdated devices [1], [8]. Our results suggest that both users and manufacturers are accountable for outdated devices. These devices "in the wild" are diverse in type, manufacturer and country (cf. [1], [78], [24]). The results of our approach are quite similar for manufacturer distribution to those by [3], [1] and [58]. This

TABLE 5
Coefficients for Selected Device Types and Manufacturers

		Device type									Manufacturer*										
		access point	camera	industrial controller	NAS	printer	router	SCADA	smart home	tv	Algo	Axis	Cisco	D-Link	Dahua Technology	HP	Hikvision	NetGear	QNAP	Synology	TP-Link	ZyXEL
Unstandardized Coefficients	B	-8.123	40.377	-5.546	1.611	17.743	49.733	5.377	57.877	13.71	-7.889	54.111	-7.889	57.158	61.111	54.111	77.111	54.111	53.311	56.034	45.111	36.511
	Std. Error	5.929	6.761	4.193	1.337	2.76	0.396	5.345	21.378	8.728	5.914	21.321	21.321	0.422	21.321	12.31	5.33	21.321	9.535	5.914	21.321	9.535
Standardized Coefficients	Beta	-0.002	0.009	-0.002	0.002	0.009	0.182	0.001	0.004	0.002	-0.002	0.004	-0.001	0.196	0.004	0.006	0.021	0.004	0.008	0.014	0.003	0.006
t		-1.37	5.972	-1.323	1.205	6.428	125.641	1.006	2.707	1.571	-1.334	2.538	-0.37	135.444	2.866	4.396	14.466	2.538	5.591	9.475	2.116	3.829
Sig.		0.171	0.000	0.186	0.228	0.000	0.000	0.314	0.007	0.116	0.182	0.011	0.711	0.000	0.004	0.000	0.000	0.011	0.000	0.000	0.034	0.000

TABLE 6
Coefficients for Selected Countries

		Country																			
		United Arab Emirates	Bahamas	China	Cuba	Germany	Dominica	Ethiopia	United Kingdom	Guyana	Hong Kong	Haiti	Indonesia	India	Liechtenstein	n/a	French Polynesia	Russian Federation	Sweden	United States	Venezuela
Unstandardized Coefficients	B	-2.129	49.693	18.357	34.818	6.122	34.318	40.518	3.393	-10.122	26.335	-12.682	3.550	10.040	-12.682	4.331	-11.682	5.499	27.462	7.407	7.589
	Std. Error	0.503	7.500	0.191	14.998	0.511	21.211	9.486	0.587	2.201	0.782	21.211	0.140	0.158	21.211	0.188	8.017	0.148	0.887	0.203	0.534
Standardized Coefficients	Beta	-0.006	0.010	0.153	0.003	0.017	0.002	0.006	0.008	-0.007	0.049	-0.001	0.045	0.107	-0.001	0.037	-0.002	0.064	0.045	0.057	0.021
t		-4.235	6.626	95.917	2.322	11.975	1.618	4.271	5.784	-4.599	33.670	-0.598	25.405	63.455	-0.598	23.084	-1.457	37.137	30.971	36.506	14.213
Sig.		0.000	0.000	0.000	0.020	0.000	0.106	0.000	0.000	0.000	0.000	0.550	0.000	0.000	0.550	0.000	0.145	0.000	0.000	0.000	0.000

is somewhat surprising as it implies that all of these manufacturers reveal FW information on their devices' web UI.

We developed a novel approach to identify the FW version share of devices accessible via public Internet based on the devices' web UI source code. This method can analyze millions of devices in a short time without the need to connect to the devices' networks. Previous studies relied on web crawling, natural language processing, data mining and traffic analysis with mostly very limited data [19], [58], [64].

We show that tailored regexp can bring the identification rate up to 99.4 percent, notwithstanding that there is currently no uniform way of gathering FW information [3]. Identifying the correct FW for a device is rather complicated, as there are different FW versions for different device revisions and models sold in specific countries. Thus, we agree that "identifying technical information for Internet-wide IoT devices remains challenging" [19, p. 8]. In this regard, engineers should develop secure interfaces or provide ports to crawl such maintenance data.

The wide range in maturity of the installed FW versions by device type ($\sigma = 26.83$) suggests that some provide much better update and patching mechanisms than others, as surmised in H1. After researching how U&P are installed on different device types, the results show that 77 percent of all devices can only be updated manually (mainly IP cameras and routers). 15 percent provide a management interface, but the user has to start the update process manually (mainly TV devices). Only 8 percent provide automatic updates (mainly routers and access points). However, these are devices from the manufacturers AVM and Huawei, which are not covered by Fig. 8, as they do not show the release date of U&P. It seems likely that many manufacturers simply do not want to offer automatic updates, as these require secure transmissions, low downtime, rollback options and user notifications [11], [43], [77].

]The variance among different device types could also be attributed to a lack of interfaces [21], [44] and is supported by the higher update frequency of PCs and smartphones. In addition, some devices require less interaction with the user (passive usage), such as smart thermostats, while others need more frequent and intense interaction (active usage), such as smart speakers. There could also be marked differences in the intention to U&P, as passive devices, in particular, are often set up and forgotten [21]. Additionally, how users are notified about U&P might influence their intention to U&P [46], although other research indicates that users often ignore notifications [12], [27], so such influence might be limited or even negative (see the arrow label "0/-" in Fig. 11). Moreover, some devices are found mostly in industry, such as SCADA and industrial controllers. It is likely that their IT departments are responsible for keeping infrastructure up-to-date, which could explain the relative up-to-dateness in this area. In addition, most of these devices are high-end products, whose security could be of higher concern. At the same time, it is interesting to find such devices in our data set, as it means these devices are not properly configured and can be found via the public Internet. Finally, access point devices are very up-to-date. One possible explanation is that security concerns are the highest for these devices, as they serve as the entry point to a

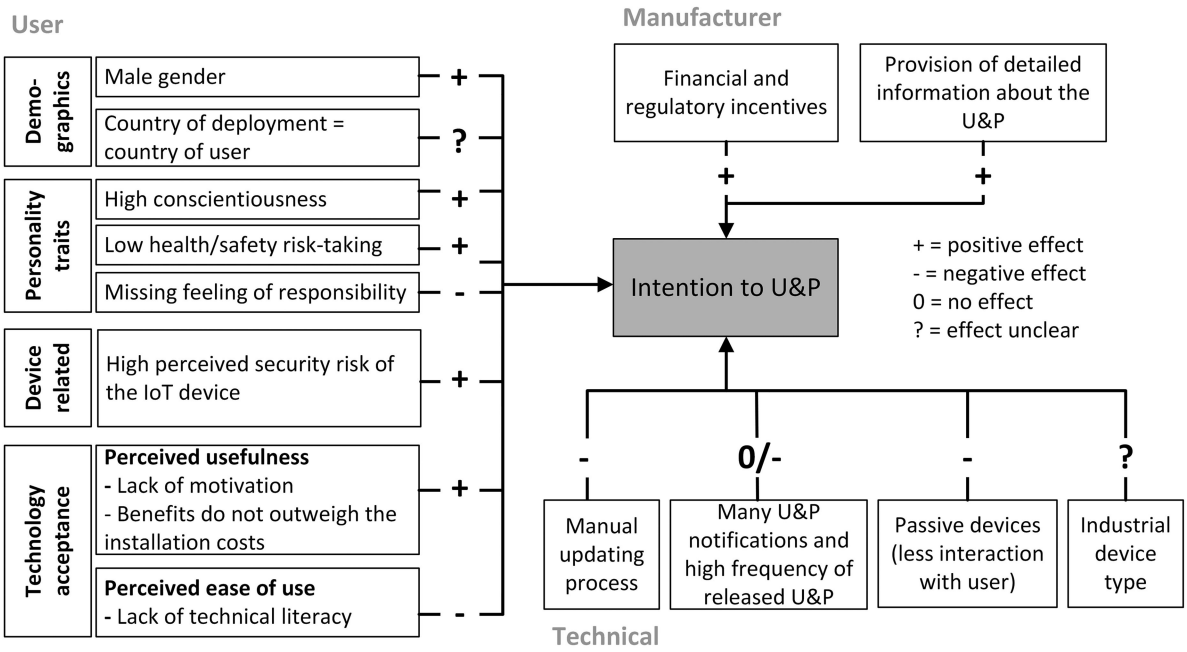

Fig. 11. First draft model of factors influencing the intention to U&P.

company's network. It should be noted that many devices are routers, hence, not IoT devices in the strict sense of the term. This observation is not surprising, because many IoT devices are deployed behind routers [24].

We find support for H2, as the average age of the latest available FW version varies strongly among manufacturers ($\sigma = 22.60$). Notably, manufacturer size appears insignificant, an observation often referred to as the "Patching Paradox". This coincides with [49, p. 126], who found that "vendor size is insignificant both statistically and economically" and [24], who found that even Amazon and Google use outdated TLS versions. One possible reason for this variance could be the method used to provide U&P. After researching how the manufacturers in Fig. 8 provide U&P, the results show that none of them support automatic or FOTA updates. This explains the overall long time between the latest and installed FW. 46 percent of the manufacturers provide a device management interface for updates, but the user has to start the search for an update manually. 38 percent provide updates only on their websites, and users have to download, extract and install them manually. The roughly even split between these different approaches also explains the differences in TB. This is especially true for TP-Link, ZyXEL and Hikvision, which show the highest TB. Their update process is quite elaborate and U&P have to be done manually. We suggest that automatic updates are the best solution for most devices to ensure they are up-to-date. Another factor not investigated in this study is the installation of proprietary and non-proprietary FW, as known from mobile OS like Android. For IoT devices, it is also conceivable that OEM produce the devices, but other vendors brand them and install a proprietary FW.

Further, we investigated whether there are manufacturers who do not provide regular U&P (H3). We were unable to fully test the validity of this hypothesis, as it is likely that some users run many end-of-life devices from manufacturers about whom we have no data. Nonetheless, the observation of irregular updates by manufacturers has been investigated in research and is often attributed to missing incentives [37], [45], [49]. Thus, financial or regulatory incentives for manufacturers could influence their intention to U&P. Further, [12] found that users tend to install U&P more frequently if manufacturers provide detailed information about the U&P, such as installation time, and ensure the FW download is easy to find.

The time between availability and installation of FW updates is an indicator of users' U&P involvement. Our results suggest that, on average, a FW update is installed 18.6 months after release, independent of the device type or manufacturer. As this is significantly higher than the installation delay of, e.g., MacOS upgrades of 80 days [53], our results confirm H4. This observation could be attributed to a "setup-and-forget" mentality [21] or to the fact that users' individual benefits do not outweigh the installation costs [8], [12]. Instead, users "choose to take on the consequences in favor of using the devices based on the utility they provide" [51, p. 3]. However, there might be other non-human factors playing a role here, such as missing or too frequent user notifications or complex or cumbersome update processes [21], [44].

Finally, the average FW age also varies between countries ($\sigma = 8.65$), which corresponds to findings about unpatched IoT vulnerabilities by [1]. The ANOVA finds a significance of $p<0.05$, but the post-hoc test for all countries is significant in only 21 percent of the cases. That system security is perceived differently in different countries is not a new phenomenon (see Global Cybersecurity Index (GCI) [63]). However, the update mentality does not correspond with the GCI. For example, devices deployed in Haiti are very up-to-date, even though they have a very low GCI of 0.046. This might be explained by the fact that GCI does not cover the consumer market. Additionally, most retrievable IoT devices are in Venezuela and Brazil, which have a medium GCI of 0.354 and 0.577. Whereas earlier studies

confirm our findings for Brazil [1], [3], [19], [78], the devices in Venezuela might have different web UIs that reveal more information about the FW version and thus coincide with more regexp. Another explanation could be country-specific factors if it is assumed that the devices deployed in a country are largely used by nationals of that country. We can employ Hofstede's cultural dimensions to understand the security behavior of users, as shown by [79] for security awareness. Comparing Jordan with Sweden, for example, reveals that there is a much higher "degree of individualism" in Sweden, a much shorter "power distance" and a three times higher "long term orientation". According to findings by [79], this results in a lower level of security awareness for individuals in Sweden. This is upheld by the finding that the FW version of Swedish devices is 42 months old compared to 5.9 months in Jordan. However, Hofstede's dimensions are not available for every country and some values, such as the one for uncertainty avoidance, is higher for Jordan than Sweden, which contradicts our findings.

There could also be technical explanatory factors, for example, the number of not yet allocated IPv4 addresses. This could explain why many devices are connected directly to the public Internet and not to a router. In some countries, there are very few IP addresses available, e.g., Bangladesh (5 per 1k person) [80]. However, Bangladesh still accounts for 1.63 percent of the devices found in our analysis.

In summary, we assume that the correlation between country and intention to U&P is unclear and reject H5. Researchers should investigate this in more detail in the future.

Our results show that manufacturer, device type and country together explain 8.4 percent of the model (Table 4). As this is the first work of its kind, we cannot assess the fit. As a caveat, we note that the results are prone to misinterpretation, because of the dominance of some values, e.g., there are many devices from MikroTik, whose average TB (18.9) is already close to the overall TB. This dominance is due to the fact that a) MikroTik devices show much information about the installed FW version on the device web UI and b) the manufacturer provides a large amount of FW version and date information on their website. Surprisingly, 49 percent of all devices run the latest available FW version. However, this finding is mainly driven by the router "H108N V2.5", whose version number corresponds to the edition rather than the FW version. If this device is excluded, the up-to-dateness rate drops dramatically to 2.45 percent, which seems more realistic.

Following the discussion of our descriptive findings, we tried to create a model showing the influences on the intention of users and manufacturers to U&P (Fig. 11). Factors were grouped into manufacturer-related, user-related and technical factors [6].

Manufacturer-related and technical factors were already discussed in the previous section including insights from the literature based on the findings. Although our quantitative results do not include human factors, these are addressed in the literature. [47] finds that a male gender, a high level of conscientiousness and a low level of health/safety risk-taking have a positive influence on the intention to update computer systems. The high perceived security risk of a product can also positively influence the U&P intention, as users are willing to pay more for IoT products that offer a six-year update guarantee [45]. [12] and [52] observe a lack of motivation, e.g., because users' individual benefits do not outweigh the installation costs [8]. Perceived usefulness and perceived ease of use [81] are additional possible explanations for the intention to U&P. Other factors include technical illiterateness [12], [52] or not feeling responsible [13], [51], both of which can negatively influence the intention to U&P.

This model represents a first step towards understanding the intention to U&P and does not claim to be collectively exhaustive. It aims to highlight possible research strands. We note that many of these conclusions and explanations were derived by analogy from computer security. Thus, the findings could be considered a grounded theory approach, with no claim to be true in the context of IoT updating behavior.

With our work, we want to draw attention to the growing need to pay more attention to U&P in practice. We contribute to the pressing problem of unpatched IoT devices [3], [82] and identify factors that influence their patch status. The results suggest that devices of particular manufacturers are updated much more frequently than those of others. This is a first step towards benchmarking different manufacturers of IoT devices, which is deemed to be useful information for vendors, engineers, policymakers, consumer protection agencies and ISP.

Policymakers can utilize the data to further reinforce the consumer protection directive [14], and use our methodology as a benchmark tool to evaluate the impacts of this directive over the next few years. Investigation factors, such as the AiF, can give valuable insights into the success of the directive over time. Such trend analyses are important. [1], for example, found that the number of vendors with vulnerable devices increased from 50 to 131 between 2018 and 2019. As our approach is fully scalable, further analysis iterations could be performed with little additional effort.

While policymakers could also choose to focus on standards, our results indicate that U&P standards, such as [10], are not implemented by low-cost manufacturers [7], and are not designed with a users focus.

For researchers, our work expands IoT-centric research by using large-scale, macroscopic and real-world IoT data to better understand the threat landscape specific to the IoT. Previous studies of U&P have mostly been limited to lab studies or specific devices. We showed that both users and manufacturers are responsible for outdated devices, which affirms the findings of [8]. We therefore pave the way for research into individual users' update behavior or manufacturers' update procedures.

Engineers can benefit from the insights into FW distribution and up-to-dateness "in the wild" in order to identify possible factors hindering the dissemination of U&P in practice. This shows how to gauge, on a global scale, to which extent end users, hardware limitations or engineering in terms of device type and manufacturer could be accounted responsibility for outdated IoT devices. At a technical level, this allows software engineers a deeper understanding of the infrastructure, their general interfaces, and the efficiency of their respective patch management processes. Engineers could use the data base of the results of this paper as a starting point for developing interfaces and processes to support or "nudge" users' U&P behavior.

Together with engineers, ISPs could anticipate possible infections. Once an infection of a device is found, they could prepare countermeasures for those with the same installed FW version, which are not infected yet.

Finally, we hope to encourage software engineers and a broad community of users to contribute regular expressions that identify IoT devices and to provide FW version and dates for devices. To do so, we plan to create a public database to collect such information. This helps to transfer research outcomes into industrial practice, which is needed due to the growing urgency associated with addressing U&P in practice.

Some limitations of our approach must be highlighted. First, our data are descriptive and quantitative in nature and serve only to obtain a first rough understanding of the status quo of FW U&P. The reliability of our results is limited due to overassessment and selection bias, as it is impossible to assess the full population. This is due to the sampling criterion that only devices that reveal FW information were added to our sample. As this is the first work of its kind, we cannot assess whether a R^2 of 0.084 (resp. explaining 8.4 percent of the model) is a good value or not. More variables could be added to improve the model.

Censys only finds devices that are accessible via public Internet. This means that constraint devices with no web UI cannot be found, but at the same time do not pose as great a security threat. This is the best source available for real-world data and our vendor and country distribution reflects prior results [3], [19]. We therefore assume that the bias in our input data does not have too much bearing on the results. To access more devices, input data from IoT manufacturers would be very helpful. However, as mentioned in the introduction, none of the contacted manufacturers agreed to disclose this information.

Further, we are optimistic that additional regexp and a more comprehensive mapping table will increase reliability. In addition, the majority of devices analyzed so far are not IoT devices in the narrow sense (i.e., end devices) but network components. However, this observation can be explained, as most IoT devices are installed behind a router and the number of IP addresses (IPv4) is almost exhausted. In order to address this problem, our process could be extended using methods of IoT fingerprinting.

6 Conclusion

Identifying outdated FW of IoT devices is a step towards improved overall IoT security. Our work indicates an increasing need to pay more attention to U&P. To the best of our knowledge, ours is the first comprehensive study of FW distribution and up-to-dateness of IoT devices. We presented an iterative and scalable approach to identify the FW version of IoT devices from *Censys* data. Our results suggest that the up-to-dateness of IoT FW is influenced by device type and manufacturer, whereas country of installation is less significant. Routers and devices from the manufacturer D-Link contribute the most. Our results show that these three variables explain 8.4 percent of the model. Due to its novelty, it is not yet possible to assess whether this value is a good fit, but we are optimistic that accuracy will improve as more devices are added to our mapping table. In summary, we found empirical support for hypotheses H1, H2 and H4, while H3 remains unanswered due to the lack of suitable data and H5 is rejected. We substantiated our quantitative results with a first model indicating factors that may influence users' intention to install updates.

7 Future Work

Future work could incorporate additional data from other sources such as Shodan, ZoomEye or IoTInspector as well as other variables in the data sets such as autonomous system number or TSL certificate. Further application and refinement of our data-mining approach is advisable.

The planned public database for regular expressions and FW versions should improve the overall reliability of the approach. In addition, software engineers could develop APIs to retrieve information about the installed FW version, or provide a standardized port to ping such information. This could anticipate possible future infections once a vulnerability has been detected. As up-to-dateness does not necessarily mean that the devices are more secure, future work could correlate the data with vulnerability databases.

Frequent analyses could help to reveal possible changes in the highly versatile IoT environment in a holistic manner (as direct device comparison is impossible due to dynamic IP address allocation). These could benchmark the impact of external events, such as the consumer protection directive [14] or workplace changes during the Covid-19 pandemic, and might encourage vendors to develop more effective and easy-to-use update mechanisms.

From a technical perspective, our analysis could be used to study the capabilities, components and configurations of the most prominent devices detected, which could help to assess the extent to which these devices fulfill U&P requirements.

We identified other factors that are worth evaluating in more detail as well. The relationships between the respective country/device type and the intention to U&P seem particularly promising. This could be done by multivariate testing in a multi-national study of IoT users. A choice based conjoint analysis could investigate the impact on the willingness to perform an update using three different methods: automatic (FOTA), semi-automatic and fully manual. Methods from human-computer interaction such as gamification could also be used to provide users with a prospective analysis of the possible effects of ignoring updates.

Acknowledgments

I thank Tobias Drexel for his advice when analyzing the data set as well as Xu Wang and his colleagues for classifying missing values. Further thanks go to Michael Friedewald, Dirk Kuhlmann and Sabine Preuß for their helpful feedback.

References

[1] A. Mangino, M. S. Pour, and E. Bou-Harb, "Internet-scale insecurity of consumer Internet of Things," *ACM Trans. Manage. Inf. Syst.*, vol. 11, no. 4, pp. 1–24, 2020, doi: 10.1145/3394504.

[2] I. Makhdoom, M. Abolhasan, J. Lipman, R. P. Liu, and W. Ni, "Anatomy of threats to the Internet of Things," *IEEE Commun. Surv. Tuts.*, vol. 21, no. 2, pp. 1636–1675, 2019, doi: 10.1109/COMST.2018.2874978.

[3] M. Antonakakis *et al.*, "Understanding the mirai botnet," in *Proc. 26th USENIX Secur. Symp.*, 2017, pp. 1093–1110. [Online]. Available: https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/antonakakis

[4] R. Yu, X. Zhang, and M. Zhang, "Smart home security analysis system based on the Internet of Things," in *Proc. IEEE 2nd Int. Conf. Big Data Artif. Intell. Internet Things Eng.*, 2021, pp. 596–599.

[5] T. Yu, V. Sekar, S. Seshan, Y. Agarwal, and C. Xu, "Handling a trillion (unfixable) flaws on a billion devices," in *Proc. 14th ACM Workshop Hot Top. Netw.*, 2015, pp. 1–7.

[6] N. Dissanayake, A. Jayatilaka, M. Zahedi, and M. A. Babar, "Software security patch management - A systematic literature review of challenges, approaches, tools and practices," *Inf. Softw. Technol.*, vol. 144, 2021, Art. no. 106771, doi: 10.1016/j.infsof.2021.106771.

[7] M. Fahmideh, A. A. Abbasi, A. Behnaz, J. Grundy, and W. Susilo, "Software engineering for Internet of Things," *IIEEE Trans. Softw. Eng.*, vol. 34, Jan./Feb. 2021, Art. no. 1, doi: 10.1109/TSE.2021.3070692.

[8] M. X. Ferreira, S. M. Weinberg, D. Y. Huang, N. Feamster, and T. Chattopadhyay, "Selling a single item with negative externalities," in *Proc. World Wide Web Conf.*, 2019, pp. 196–206.

[9] M. Shahzad, M. Z. Shafiq, and A. X. Liu, "Large scale characterization of software vulnerability life cycles," *IEEE Trans. Dependable Secure Comput.*, vol. 17, no. 4, pp. 730–744, Jul./Aug. 2019, doi: 10.1109/TDSC.2019.2893950.

[10] IEFT, "Software updates for Internet of Things," Accessed: Nov. 29 2021. [Online]. Available: https://datatracker.ietf.org/wg/suit/about/

[11] 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. 71907–71920, 2019, doi: 10.1109/ACCESS.2019.2919760.

[12] K. Vaniea and Y. Rashidi, "Tales of software updates: The process of updating software," in *Proc. 34th Annu. C.HI Conf. Hum. Factors Comput. Syst.*, 2016, pp. 3215–3226.

[13] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, and N. Ghani, "Demystifying IoT security: An exhaustive survey on IoT vulnerabilities and a first empirical look on internet-scale IoT exploitations," *IEEE Commun. Surv. Tuts.*, vol. 21, no. 3, pp. 2702–2733, Jul.–Sep., doi: 10.1109/COMST.2019.2910750.

[14] European Parliament, "Directive (EU) 2019/770," Accessed: Mar. 10 2020. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32019L0770

[15] K. J. Smith, G. Dhillon, and L. Carter, "User values and the development of a cybersecurity public policy for the ioT," *Int. J. Inf. Manage.*, vol. 56, 2021, Art. no. 102123, doi: 10.1016/j.ijinfomgt.2020.102123.

[16] D. He *et al.*, "Toward hybrid static-dynamic detection of vulnerabilities in IoT firmware," *IEEE Netw.*, vol. 35, no. 2, pp. 1–6, Mar./Apr. 2021, doi: 10.1109/MNET.011.2000450.

[17] N.-W. Lo and S.-H. Hsu, "A secure IoT firmware update framework based on MQTT protocol," in *Advances in Intelligent Systems and Computing*, L. Borzemski, J. Świątek, and Z. Wilimowska, Eds., 1st ed., Cham, Switzerland: Springer, 2020, pp. 187–198.

[18] Censys, "Censys," Accessed: Mar. 06, 2020. [Online]. Available: https://censys.io/

[19] M. S. Pour, E. Bou-Harb, K. Varma, N. Neshenko, D. A. Pados, and K.-K. R. Choo, "Comprehending the IoT cyber threat landscape: A data dimensionality reduction technique to infer and characterize Internet-scale IoT probing campaigns," *Digit. Investigation*, vol. 28, pp. S40–S49, 2019, doi: 10.1016/j.diin.2019.01.014.

[20] A. Nappa, R. Johnson, L. Bilge, J. Caballero, and T. Dumitras, "The attack of the clones: A study of the impact of shared code on vulnerability patching," in *Proc. IEEE Symp. Secur. Privacy*, 2015, pp. 692–708.

[21] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DDoS in the iot: Mirai and other botnets," *Comput.*, vol. 50, no. 7, pp. 80–84, 2017, doi: 10.1109/MC.2017.201.

[22] S. Ransbotham, R. G. Fichman, R. Gopal, and A. Gupta, "Special section introduction—Ubiquitous IT and digital vulnerabilities," *Inf. Syst. Res.*, vol. 27, no. 4, pp. 834–847, 2016, doi: 10.1287/isre.2016.0683.

[23] Z. B. Celik, E. Fernandes, E. Pauley, G. Tan, and P. McDaniel, "Program analysis of commodity IoT applications for security and privacy," *ACM Comput. Surv.*, vol. 52, no. 4, pp. 1–30, 2019, doi: 10.1145/3333501.

[24] D. Y. Huang, N. Apthorpe, F. Li, G. Acar, and N. Feamster, "IoT inspector: Crowdsourcing labeled network traffic from smart home devices at scale," *Proc. ACM Interactive Mobile Wearable Ubiquitous Technol.*, vol. 4, no. 2, pp. 1–21, 2020, doi: 10.1145/3397333.

[25] G. Acar, D. Y. Huang, F. Li, A. Narayanan, and N. Feamster, "Web-based attacks to discover and control local IoT devices," in *Proc. Workshop IoT Secur. Privacy*, Budapest Hungary, 2018, pp. 29–35.

[26] S. Torabi, E. Bou-Harb, C. Assi, M. Galluscio, A. Boukhtouta, and M. Debbabi, "Inferring, characterizing, and investigating internet-scale malicious IoT device activities: A network telescope perspective," in *Proc. 48th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw.*, 2018, pp. 562–573.

[27] S. Ray, A. Basak, and S. Bhunia, "Patching the Internet of Things," *IEEE Spectr.*, vol. 54, no. 11, pp. 30–35, Nov. 2017, doi: 10.1109/MSPEC.2017.8093798.

[28] P. Liu *et al.*, "IFIZZ: Deep-state and efficient fault-scenario generation to test IoT firmware," 2021. [Online]. Available: https://nesa.zju.edu.cn/download/liu_pdf_ifizz.pdf

[29] J. Shim, "Cyber-physical systems and industrial IoT cybersecurity: Issues and solutions," 2019. [Online]. Available: https://aisel.aisnet.org/amcis2019/info_security_privacy/info_security_privacy/4

[30] S. Liu, R. Kuhn, and H. Rossman, "Surviving insecure IT: Effective patch management," *IT Professional*, vol. 11, no. 2, pp. 49–51, 2009, doi: 10.1109/MITP.2009.38.

[31] ServiceNow, "Costs and consequences of gaps in vulnerability response," 2018. Accessed: May 04, 2021. [Online]. Available: https://www.servicenow.com/lpayr/ponemon-vulnerability-survey.html

[32] AimPoint Group, "Cyber hygiene report: Lessons learned from a survey of the state of endpoint patching and hardening," 2020. Accessed: Feb. 03, 2022. [Online]. Available: https://patch.automox.com/rs/923-VQX-349/images/Automox_2020_Cyber_Hygiene_Report-What_You_Need_to_Know_Now.pdf

[33] P. Anand, Y. Singh, A. Selwal, M. Alazab, S. Tanwar, and N. Kumar, "IoT vulnerability assessment for sustainable computing: Threats, current solutions, and open challenges," *IEEE Access*, vol. 8, pp. 168825–168853, 2020, doi: 10.1109/ACCESS.2020.3022842.

[34] Capgemini, "Securing the Internet of Things opportunity: Putting cybersecurity at the heart of the ioT," Accessed: Jan. 2021. [Online]. Available: https://www.capgemini.com/at-de/resources/securing-the-internet-of-things-opportunity-putting-cyber-security-at-the-heart-of-the

[35] IDG Research Services, "Studie Internet of Things," 2019, Accessed: Jan. 13, 2021. [Online]. Available: https://www.q-loud.de/hubfs/Kundendownloads/IDG-Studie_IoT_2018_2019.pdf

[36] IEEE, "Software engineering body of knowledge (SWEBOK)," Accessed: Jan. 31, 2022. [Online]. Available: https://www.computer.org/education/bodies-of-knowledge/software-engineering

[37] K. Fawaz and K. G. Shin, "Security and privacy in the Internet of Things: D," *Computer*, vol. 52, no. 4, pp. 40–49, 2019, doi: 10.1109/MC.2018.2888765.

[38] R. Tollefsen, I. Rais, J. M. Bjorndalen, P. H. Ha, and O. Anshus, "Distribution of updates to IoT nodes in a resource-challenged environment," in *Proc. IEEE/ACM 21st Int. Symp. Cluster Cloud Internet Comput.*, 2021, pp. 684–689.

[39] M. Stolikj, P. Cuijpers, and J. Lukkien, "Patching a patch - software updates using horizontal patching," *IEEE Trans. Consum. Electron.*, vol. 59, no. 2, pp. 435–441, May 2013, doi: 10.1109/tce.2013.6531128.

[40] L. Baresi, C. Ghezzi, X. Ma, and V. P. La Manna, "Efficient dynamic updates of distributed components through version consistency," *IEEE Trans. Softw. Eng.*, vol. 43, no. 4, pp. 340–358, Apr. 2017, doi: 10.1109/TSE.2016.2592913.

[41] Z. Zhao, Y. Jiang, C. Xu, T. Gu, and X. Ma, "Synthesizing object state transformers for dynamic software updates," in *Proc. IEEE/ACM 43rd Int. Conf. Softw. Eng.*, 2021, pp. 1111–1122.

[42] P. Pfister and M. Konstantynowicz, "Patching the Internet of Things: IoT software update workshop," 2016, Accessed: Jan. 4, 2022. [Online]. Available: https://www.ietf.org/blog/patching-internet-things-iot-software-update-workshop-2016/

[43] I. Mugarza, A. Amurrio, E. Azketa, and E. Jacob, "Dynamic software updates to enhance security and privacy in high availability energy management applications in smart cities," *IEEE Access*, vol. 7, pp. 42269–42279, 2019, doi: 10.1109/ACCESS.2019.2905925.

[44] S.-M. Cheng, P.-Y. Chen, C.-C. Lin, and H.-C. Hsiao, "Traffic-aware patching for cyber security in mobile ioT," *IEEE Commun. Mag.*, vol. 55, no. 7, pp. 29–35, Jul. 2017, doi: 10.1109/MCOM.2017.1600993.

[45] P. Morgner, C. Mai, N. Koschate-Fischer, F. Freiling, and Z. Benenson, "Security update labels: Establishing economic incentives for security patching of IoT consumer products," in *Proc. IEEE Symp. Secur. Privacy*, 2020, pp. 429–446.

[46] A. Forget *et al.*, "Do or do not, there is no try: User engagement may not improve security outcomes," 2016, pp. 97–111. [Online]. Available: https://www.usenix.org/conference/soups2016/technical-sessions/presentation/forget

[47] M. Gratian, S. Bandi, M. Cukier, J. Dykstra, and A. Ginther, "Correlating human traits and cyber security behavior intentions," *Comput. Secur.*, vol. 73, pp. 345–358, 2018, doi: 10.1016/j.cose.2017.11.015.
[48] GitHub, "Octoverse report 2020," Dec. 2020. Accessed: Nov. 23 2021. [Online]. Available: https://octoverse.github.com/static/github-octoverse-2020-security-report.pdf
[49] A. Arora, R. Krishnan, R. Telang, and Y. Yang, "An empirical analysis of software vendors' patch release behavior: Impact of vulnerability disclosure," *Inf. Syst. Res.*, vol. 21, no. 1, pp. 115–132, 2010, doi: 10.1287/isre.1080.0226.
[50] K. R. Jones, T.-F. Yen, S. C. Sundaramurthy, and A. G. Bardas, "Deploying android security updates: An extensive study involving manufacturers, carriers, and end users," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2020, pp. 551–567.
[51] Z. Singer and B. Jones, "The Internet of Things: The effects of security attitudes and knowledge on security practices," 2019. [Online]. Available: https://aisel.aisnet.org/amcis2019/info_security_privacy/info_security_privacy/29
[52] Canonical, "Taking charge of the iot's security vulnerabilities: White paper," 2017. Accessed: Apr. 08, 2020. [Online]. Available: https://ubuntu.com/engage/whitepaper-iot-security
[53] F. Vitale, J. McGrenere, A. Tabard, M. Beaudouin-Lafon, and W. E. Mackay, "High costs and small benefits," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2017, pp. 4242–4253.
[54] StatCounter, "Software version share," Accessed: Apr. 18 2020. [Online]. Available: https://gs.statcounter.com/
[55] Avast, "PC trends report," Accessed: Mar. 03, 2020. [Online]. Available: https://blog.avast.com/pc-trends-reports
[56] WhatWeb, "WhatWeb," Accessed: Mar. 06, 2020. [Online]. Available: https://www.whatweb.net/
[57] D. Privitera and L. Li, "Can IoT devices be trusted? An exploratory study," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2018. [Online]. Available: https://aisel.aisnet.org/amcis2018/Security/Presentations/44
[58] X. Wang, Y. Wang, X. Feng, H. Zhu, L. Sun, and Y. Zou, "IoTTracker: An enhanced engine for discovering Internet-of-Thing devices," in *Proc. IEEE 20th Int. Symp. A World Wireless, Mobile Multimedia Netw.*, 2019, pp. 1–9.
[59] A. Cui, M. Costello, and S. Stolfo, "When firmware modifications attack: A case study of embedded exploitation," in *Proc. 20th Annu. Netw. Distrib. System Secur. Symp.*, 2013, pp. 1–13, doi: 10.7916/D8P55NKB.
[60] P. Marrapese, "Abusing P2P to hack 3 million cameras," 2020. [Online]. Available: https://av.tib.eu/media/49779
[61] D. Kumar *et al.*, "All things considered: An analysis of IoT devices on home networks," 2019. [Online]. Available: https://www.usenix.org/system/files/sec19-kumar-deepak_0.pdf
[62] Y. Chen and F. M. Zahedi, "Individuals' internet security perceptions and behaviors: Polycontextual contrasts between the United States and China," *MISQ*, vol. 40, no. 1, pp. 205–222, 2016, doi: 10.25300/MISQ/2016/40.1.09.
[63] ITU, "Global cybersecurity index," 2018, Accessed: Jan. 19, 2021. [Online]. Available: https://www.itu.int/pub/D-STR-GCI.01
[64] X. Feng, Q. Li, H. Wang, and L. Sun, "Acquisitional rule-based engine for discovering Internet-of-Thing devices," in *Proc. 27th USENIX Secur. Symp.*, 2018, pp. 327–341.
[65] E. Rodríguez, A. Noroozian, M. van Eeten, and C. Gañán, "Superspreaders: Quantifying the role of IoT manufacturers in device infections," 2021. [Online]. Available: https://weis2021.econinfosec.org/wp-content/uploads/sites/9/2021/06/weis21-rodriguez.pdf
[66] R. Perdisci, T. Papastergiou, O. Alrawi, and M. Antonakakis, "IoTFinder: Efficient large-scale identification of IoT devices via passive DNS traffic analysis," in *Proc. IEEE Eur. Symp. Secur. Privacy*, 2020, pp. 474–489.
[67] A. Sivanathan *et al.*, "Classifying IoT devices in smart environments using network traffic characteristics," *IEEE Trans. Mobile Comput.*, vol. 18, no. 8, pp. 1745–1759, Aug. 2019, doi: 10.1109/TMC.2018.2866249.
[68] Y. Meidan *et al.*, "ProfilIoT," in *Proc. 32nd Annu. ACM Symp. Appl. Comput.*, 2017, pp. 506–509.
[69] J. Ortiz, C. Crawford, and F. Le, "DeviceMien: Network device behavior modeling for identifying unknown IoT devices," in *Proc. Internet Things Des. Implementation*, 2019, pp. 106–117.
[70] Shodan, Accessed: Mar. 06, 2020. [Online]. Available: https://www.shodan.io/
[71] J. Matherly, "Complete guide to shodan," 2018. [Online]. Available: https://leanpub.com/shodan
[72] S. A. S. Institute, "Introduction to SEMMA," Accessed: Apr. 09, 2020. [Online]. Available: https://documentation.sas.com/
[73] P. Fournier-Viger, A. Gomariz, T. Gueniche, E. Mwamikazi, and R. Thomas, "TKS: Efficient mining of Top-K sequential patterns," in *Proc. Int. Conf. Adv. Data Mining Appl.*, 2013, pp. 109–120.
[74] H. Guo and J. Heidemann, "Detecting IoT devices in the internet," *IEEE/ACM Trans. Netw.*, vol. 28, no. 5, pp. 2323–2336, Oct. 2020, doi: 10.1109/TNET.2020.3009425.
[75] H. Knapp, *Intermediate Statistics Using SPSS*. Los Angeles, CA, USA: SAGE, 2018. [Online]. Available: https://methods.sagepub.com/book/intermediate-statistics-using-spss
[76] Forescout Research Labs, Accessed: Nov. 23 2021. [Online]. Available: https://www.forescout.com/research-labs/amnesia33/
[77] M. Weisbach, N. Taing, M. Wutzler, T. Springer, A. Schill, and S. Clarke, "Decentralized coordination of dynamic software updates in the Internet of Things," in *Proc. IEEE 3rd World Forum Internet Things*, 2016, pp. 171–176.
[78] A10 Networks, "The state of DDoS weapons," Accessed: Apr. 07, 2020. [Online]. Available: https://www.a10networks.com/marketing-comms/reports/state-ddos-weapons/
[79] C. C. Chen, B. D. Medlin, and R. S. Shaw, "A cross-cultural investigation of situational information security awareness programs," *Inf. Manage. Comput. Secur.*, vol. 16, no. 4, pp. 360–376, 2008, doi: 10.1108/09685220810908787.
[80] A. P. I. WhoisXML, "A list of IP addresses by country," Accessed: Dec. 2021. [Online]. Available: https://ip-geolocation.whoisxmlapi.com/statistics
[81] F. D. Davis, R. P. Bagozzi, and P. R. Warshaw, "User acceptance of computer technology: A comparison of two theoretical models," *Manage. Sci.*, vol. 35, no. 8, pp. 982–1003, 1989, doi: 10.1287/mnsc.35.8.982.
[82] J. P. Shim, R. Sharda, A. M. French, R. A. Syler, and K. P. Patten, "The Internet of Things: Multi-faceted research perspectives," *CAIS*, vol. 46, pp. 511–536, 2020, doi: 10.17705/1CAIS.04621.

Frank Ebbers received the master's degree in "International Information Systems Management" with the University of Bamberg, Germany, in 2018. Since 2019, he is a research associate with the Fraunhofer Institute for Systems and Innovation Research ISI in Germany, and currently working toward the doctoral degree with the Chair of Information Systems and Information Management, Goethe University Frankfurt. During his studies he worked i. a. at Robert Bosch (SEA) in Singapore and Carl Zeiss in Germany. Besides his studies, he passed the certification as Scrum Master and started a small entrepreneurship.